Investigations on different Deep Learning Architectures for Telugu language Text-To-Speech Synthesis System

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

by

RAVI BOLIMERA

(Roll No. 715052)

Under the Supervision of

Prof. T. KISHORE KUMAR

Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL – 506004, T. S., INDIA

August – 2023

Dedicated to

God Almighty my Creator

and my beloved

Teachers, Parents, Children, and Wife

APPROVAL SHEET

This thesis entitled " **Investigations on different Deep Learning Architectures for Telugu language Text-To-Speech Synthesis System "** by **Mr. Ravi Bolimera** is approved for the degree of **Doctor of Philosophy**.

Examiners

_____

_____

Supervisor

_____

Prof. T. Kishore Kumar

Professor, Electronics and Communication Engineering Department,

NIT WARANGAL

Chairman

_____

Prof. D. Vakula

Head, Electronics and Communication Engineering Department,

NIT WARANGAL

Date:

Place:

DECLARATION

I, hereby, declare that the matter embodied in this thesis entitled " **Investigations on different Deep Learning Architectures for Telugu Language Text-To-Speech Synthesis System "** is based entirely on the results of the investigations and research work carried out by me under the supervision of **Prof. T. Kishore Kumar**, Department of Electronics and Communication Engineering, National Institute of Technology Warangal. I declare that this work is original and has not been submitted in part or full, for any degree or diploma to this or any other University.

I declare that this written submission represents my ideas in my own words and where other ideas or words have been included. I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ravi Bolimera

Roll No: 715052

Date: 16/08/2023

Place: Warangal

Department of Electronics and Communication Engineering
National Institute of Technology
Warangal - 506004, Telangana, India

CERTIFICATE

This is to certify that the dissertation work entitled " **Investigations on different Deep Learning Architectures for Telugu language Text-To-Speech Synthesis System** ", which is being submitted by Mr. Ravi Bolimera (Roll No. 715052), a bonafide work submitted to National Institute of Technology Warangal in partial fulfilment of the requirement for the award of the degree of Doctor of Philosophy to the Department of Electronics and Communication Engineering of National Institute of Technology Warangal, is a record of bonafide research work carried out by him under my supervision and has not been submitted elsewhere for any degree.

Dr. T. KishoreKumar
(Supervisor)
Professor, Department of ECE
National Institute of Technology
Warangal, India – 506004

ACKNOWLEDGEMENTS

I would like to thank a number of people who have contributed to my Ph.D. directly or indirectly and in different ways through their help, support, and encouragement.

It gives me immense pleasure to express my deep sense of gratitude and thanks to my supervisor **Prof. T. Kishore Kumar, (**NIT Warangal), for his invaluable guidance, support, and suggestions. His knowledge, suggestions, and discussions helped me to become a capable researcher. He has shown me the interesting side of this wonderful multidisciplinary area and guided me thoroughly.

I am thankful to the current Head of the Dept. of ECE, **Prof. D. Vakula**, and the former Heads, **Prof. P. Sreehari Rao**, **Prof. L. Anjaneyulu**, **Prof. N. Bheema Rao,** and **Prof. T. Kishore Kumar** for giving me the opportunity and all the necessary support from the department to carry out my research work.

I take this privilege to thank all my Doctoral Scrutiny Committee members, **Prof. D. M. Vinod Kumar**, Department of Electrical Engineering, **Prof. L. Anjaneyulu**, Department of Electronics and Communication Engineering, **Prof. C. B. Rama Rao**, Department of Electronics and Communication Engineering and **Dr. G. Arun Kumar**, Assistant Professor, Department of Electronics and Communication Engineering for their detailed review, constructive suggestions and excellent advice during the progress of this research work.

Special thanks to my seniors Dr. Sunnydayal V, Dr. Rakesh P, Dr. Sudeep Surendran, and Dr. Prasad N for their motivation, and suggestions during publishing papers and for being extremely supportive throughout my Ph. D. period.

I take this opportunity to convey my regards to my speech lab-mates, NIT Warangal, B Surekha Reddy, S Siva Priyanka, K Sunil Kumar, M Suneetha, M Manoj Kumar, and A Govind, for being by my side in my hour of need.

I thank my department co-scholars, especially Sahoo R, Santhosh Kumar V, and Shashank R for being very supportive.

I also appreciate the help rendered from teaching, non-teaching members, and fraternity of the deprtment of ECE of NIT Warangal. They have always been encouraging and supportive.

I acknowledge with gratitude my teachers and colleagues whose help enabled me to get through the thesis.

Finally, I appreciate my family members (my father Mr. B. Adamu, my mother Mrs. B. Mariyamma, my son B. Baruch, my daughter B. Benita and my beloved wife Naveena Priyadarsini) for being very supportive while giving me the mental support and inspiration that motivated me to complete the thesis work successfully. In particular, I thank my wife who has been supportive throughout my Ph.D. period.

RAVI BOLIMERA

ABSTRACT

Speech is an innate human capability for communication. An inanimate agent such as a machine, must know how to speak and listen well in order to interact with human beings. When a human being reads a text string, he or she may parse the text mentally, unconsciously predicting the pronunciation of each word and the manner in which they speak. By manner, we mean the pace, pause, intonation, and other things rather than pronunciation. Text-To-Speech synthesis (TTS) System can be implemented as a similar process to produce Synthetic speech.

Control over voice quality is a challenging task in speech synthesis. The existing techniques in building the TTS systems, Unit Selection Synthesis (USS) deals with issues of a) how to manage large numbers of units, b) how to extend prosody and timing control; conventional Hidden Markov Model (HMM) - based Statistical Parametric Speech Synthesis (SPSS) is inefficient to express complex context dependencies by decision trees that degrade quality. Therefore, there is a need to develop a high-quality TTS system with a small footprint. The researcher was motivated to model prosodic parameters by investigating different deep learning architectures for developing a quality TTS system.

Prosodic Modification (PM), Deep Neural Networks (DNN), simple Recurrent Neural Networks (Simple RNN), Gated RNN, and Elman RNN (ERNN) architectures were investigated in this thesis. Modifications in pitch, duration, and intensity of speech signals without impacting quality or naturalness are referred to as Prosody Modeling or Prosodic Modification. Natural speech signals have a low variation of pitch or fundamental frequency (F0) and intensity between the syllable boundary's left and right frames. With this advantage, we adopt a prosody adjustment process that involves comparing the parameters of surrounding syllables with join syllable in this study. This will give the advantage to create a USS system with a small footprint when there aren't enough syllables with differing pitch and amplitude in the database. This work performs and presents the adjustment of syllables prosody prior to concatenation in order to match neighboring syllables F0 and intensity. F0 and intensity of the current syllable last frames and next syllable first frames are used to determine the modification factor. The synthesized speech perceived discontinuity may be minimized as a result of reduced discontinuity in pitch and amplitude at the joining point.

The relationship between input texts and their acoustic realizations was modeled by incorporating deep architectures for mapping linguistic features to statistics of acoustic features and prosody for improving the naturalness of synthesized speech. During synthesis, the text provided is transformed into phonetic and acoustic notation, and Mel-cepstrum

(MCEP) coefficients are forecast using the said NN models. The test phrase database is used to obtain the durations of each phoneme and the fundamental frequencies (F0). STRAIGHT vocoder uses the original F0 and predicted MCEPs to produce speech. With a 5ms frameshift, 235 coefficient vectors are predicted for each 10ms frame size with 529 input textual features. The back propagation learning approach is used to train the NN model during a 20 iteration period.

For the English language, BDL & SLT US voice from CMU ARCTIC dataset was used. Out of 1132 utterances, 1075 utterances were used as a part of training and the remaining 56 utterances were used for testing and validation. For the Telugu language, IIIT Hyderabad and INDIC voices from the INDIC dataset were used. Out of 1000 utterances, 950 utterances were used as a part of training and the remaining 50 utterances were used for testing and validation. 50-dimensional Mel-general cepstral (MGC) features and 26-dimensional Band APeriodicities (BAP) were extracted with a frameshift of 5 ms for all the speech utterances along with their deltas and double-deltas. This feature extraction is followed from HTS-STRAIGHT demo available online.

DNNs are robust models, but their performance in parametric voice synthesis is limited by a few issues with network architecture and training methods. We call attention to the issues: Acoustic modeling, Contextual representation, and Over-smoothing that must be fixed for the model to perform well for NN based SPSS. We show that ERNNs can perform on par with gated RNNs despite being more complex RNN designs, such as Long Short-Term Memory (LSTM) and variants like Gated Recurrent Units (GRU) and Simplified LSTM (SLSTM). To prevent the vanishing gradients problem, RNNs are typically developed using either LSTMs or GRUs as RNN units. However, ERNNs are preferred over LSTM/GRU/SLSTM: (i) The number of parameters in gated recurrent architectures are significantly higher than ERNN for a given hidden state size, and the number of computational steps are also higher because the recurrent unit has more gates; and (ii) It is unclear which elements of the complex architectures are playing a crucial role. Deep Elman RNN-based SPSS system significantly improved the quality of synthesized speech with the use of DNN/ERNN as acoustically guided features but still affected with over-smoothing.

The novel paradigms : End to End TTS systems include a Sequence-to-Sequence Feature prediction network that maps the character vector to the Mel-Spectrogram, which is an efficient one to generate the qualitative speech directly from the characters. The experimental results indicated that the system performs well and gave a natural and intelligible speech as output. Need to investigate many aspects for further improvement of fastness and naturalness of the predicted speech.

List of Tables

List of Figures

List of Abbrevations

TTS	Text - To - Speech synthesis
USS	Unit selection synthesis
HMM	Hidden Markov Model
SPSS	Statistical Parametric Speech Synthesis
PM	Prosodic Modification
NN	Neural Networks
DNN	Deep Neural Networks
RNN	Recurrent Neural Networks
SRNN	Simple RNN
ERNN	Elman RNN
GRU	Gated Recurrent Units
LSTM	Long Short-Term Memory
SLSTM	Simplified LSTM
BLSTM	Bidirectional LSTM
ULSTM	Unidirectional LSTM
F0	Fundamental Frequency
MOS	Mean Opinion Score
DMOS	Degraded Mean Opinion Score
MCEP	Mel-cepstrum
MCD	Mel-Cepstral Distortion
LSP	Line-Spectral Pairs
HTS	HMM based TTS
MGC	Mel-General Cepstral
BAP	Band APeriodicities
C	Consonant
V	Vowel
RMSE	Root-Mean Square Error
VUV	Voiced/UnVoiced
CART	Classification and Regression Trees

LDMs	Linear Dynamical Models
PoEs	Product of Experts
SGD	Stochastic Gradient Descent
MLPG	Maximum-Likelihood Parameter Generation
CRL	Context Representation Learning
MSE	Mean Square Error
STFT	Short-Time Fourier Transform
BPTT	Back-Propagation Through Time
CI	Confidence Interval

List of Symbols

C_l^s	l^{th} coefficient of the synthesized utterance
C_l^n	l^{th} coefficient of the natural utterance
$\mathbf{M_i}$	i^{th} pitch period Modification factor
$\mathbf{I_i}$	Intensity contour of pitch period " i "
$\mathbf{E_i}$	Aaverage Energy of pitch period " i "
$\boldsymbol{\sigma}$	Sigmoid activation function
$\boldsymbol{\Theta}$	Element-wise multiplication
$\mathbf{W_i}$,	Input gate weight from the input state
$\mathbf{W_z}$	Unit-input weight from the input state
$\mathbf{W_f}$	Forget gate weight from the input state
$\mathbf{W_o}$	Output gate weight from the input state
$\mathbf{R_i}$	Input gate weight from the previous state
$\mathbf{R_z}$	Unit-input weight from the previous state
$\mathbf{R_f}$	Forget gate weight from the previous state
$\mathbf{R_o}$	Output gate weight from the previous state
$\mathbf{x_t}$	Input state at time **t**
$\mathbf{h_t}$	Hidden state at time **t**
$\mathbf{p_i}$	Input gate peep-hole connection
$\mathbf{p_f}$	Forget gate peep-hole connection
$\mathbf{p_o}$	Output gate peep-hole connection
$\boldsymbol{\lambda}$	Decay weight
$\boldsymbol{\alpha}$	Learning rate
$\boldsymbol{\beta 1}$ & $\boldsymbol{\beta 2}$	Exponential decay rates
$\mathbf{W_i}$	Input to hidden weights,
$\mathbf{W}$	Recurrent weight matrix of hidden layer
$\mathbf{b_h}$	Hidden bias,
$\mathbf{U}$	Hidden to output weight matrix
$\mathbf{b_o}$	bias vector for output layer

f	Nonlinear function at hidden layer
g	Nonlinear function at output layer
$\mathbf{x_t}$	Input at time **t**
$\mathbf{h_t}$	State at time **t**
$\mathbf{y_t}$	Output at time **t**
$\mathbf{h_{t-1}}$	Previous state time instant **t-1.**
$\mathbf{d_t}$	Desired signal at time **t**
$\mathbf{e_t}$	Error signal at time **t** at output layer
$\boldsymbol{\delta_t}$	Error signal at time **t** at hidden layer
f `	Derivative of the hidden layer activation function
T	Duration of the sequence.
$\mathbf{o_t}$	Target acoustic feature vector
h_{ij}	Activation at the i^{th} layer at the j^{th} frame
c_i	Current Generated Word

CONTENTS

Chapter-1

Introduction

Speech synthesis is the process of translating text into spoken language. The generated speech, called synthetic speech has been progressively improving over the previous few decades. A TTS system's purpose is to synthesize speech with intelligibility and naturalness from a given text. TTS systems have a variety of uses, providing a tool for interaction with dialogue systems, including public announcement systems, speech-to-speech translation, virtual personal assistants like "Apple's Siri", "Amazon's Echo", Samsung's talking refrigerator, and disabled people education using a TTS system with a screen reader [1], [2] as an assistive technology. Dozens of recorded hours of one professional speaker's speech in an acoustically perfect environment using a top-notch microphone are often needed to develop a high-quality voice. Additionally, in this day of globalisation, it is essential for all individuals to have access to language technological advances, such as professionals in the medical field, travelers, journalists, corporations, speech translation, those working in emergency response, and law enforcement.

The scope of the thesis is to develop a high-quality TTS system with a limited dataset that effectively models all the phonetic and complex dependencies in order to produce robust synthetic speech as natural as human speech. This chapter provides a brief introduction to

TTS system, and various types of TTS systems with evolution. The motivation for developing a robust TTS system with high quality and low data set followed by problem statement, objectives, contributions, and thesis organization is presented.

1.1. Introduction to Text – to – Speech System

Speaking to one another is a significant component of human contact. The dynamics and procedures involved in the creation and perception of speech have thus been the subject of centuries of investigation in the speech and hearing sciences. By changing both what is being said and how it is being uttered, a speaker encrypts the message into speech. Consequently, a listener decodes the information from the structure and content of the speech. To communicate with people, an inanimate agent like a machine needs to be able to talk and listen clearly. When a person reads a text string, they may mentally parse it, making unconscious assumptions about how each word will be spoken and how it will be used. Instead of word pronunciation, we refer to the tempo, pause, intonation, and other factors as manner. The speaker can direct the speech organs to speak based on the inferred information about the text. Similar steps can be taken to implement a TTS system.

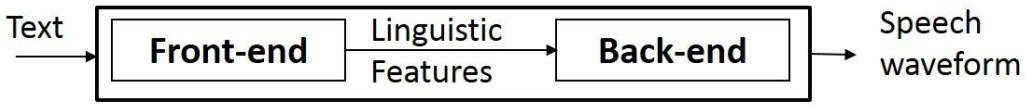

Figure 1.1. Block diagram of Basic TTS System

Text-to-speech synthesis, as seen in Figure 1.1, is the process of translating written language into spoken language. A front-end examines the text input and, among other things, decides what to say (pronunciation) and how to say it (speed, pause, and intonation). The speaking plan is subsequently transformed into a speech waveform by a back-end. The speaking strategy is organised and conveyed as linguistic elements in TTS systems. Two challenges are presented by TTS, one for the front end and one for the back end. First, it can be challenging to

decipher the text and determine linguistic aspects such as how to pronounce acronyms, which words to emphasise, whether to use a question or a sarcastic tone and so forth. This challenge cannot be avoided because not all of the information that determines what to say and how to say it is encoded in a text [3]. The second challenge is to translate those precise and sufficient linguistic characteristics into a speech waveform with a natural sound. A speech waveform is a physical signal, whereas linguistic attributes are a collection of symbols or numbers.

1.2. Fundamentals of Speech Production

The acoustic waveform is another name for speech, a dynamic, information-carrying signal. These waves are caused by the sound pressure that is created in the speaker's mouth as a result of a coordinated series of motions made by a number of structures in the human vocal system. Phonetics is the study of the characteristics and production of human sound. Both the speaker's acoustic wave and the listener's interpretation of the signal make up voice communication. The process of speech generation has been extensively studied and is now understood, whereas the mechanism of speech perception is still largely a mystery to science. Engineers must have a thorough understanding of the procedures involved in speech creation and perception in order to create appropriate ways for representing and transforming acoustic signals to produce the required outcomes.

1.2.1. Human Speech Production Mechanism

“Lungs, trachea (air flow pipe), larynx, vocal folds, pharyngeal cavity, oral cavity, nasal cavity, nostrils velum (soft palate), tongue, jaw, teeth, and lips” make up the human speech production system. The respiratory component of the mechanism is made up of the lungs and trachea. When the lungs exhale air into the trachea, this provides the energy needed for speech. The airflow that results goes via the larynx, which gives the system intermittent excitation to create the voiced sounds. Figure 1.2 depicts the cross-sectional view of the human vocal-tract system. “oral cavity (mouth), trachea (air flow pipe), lungs, larynx (voice production organ), pharyngal cavity (throat), and nasal cavity (nose)” are the significant parts of the system.

Typically, the nasal cavity is referred to as the nasal tract, while the pharyngeal and oral cavities are referred to as the vocal tract. The lungs supply air to the vocal tract, trachea, and vocal cords. The vocal tract and glottis are two different terms for the aperture between the vocal folds and the acoustic tube that extends above them.

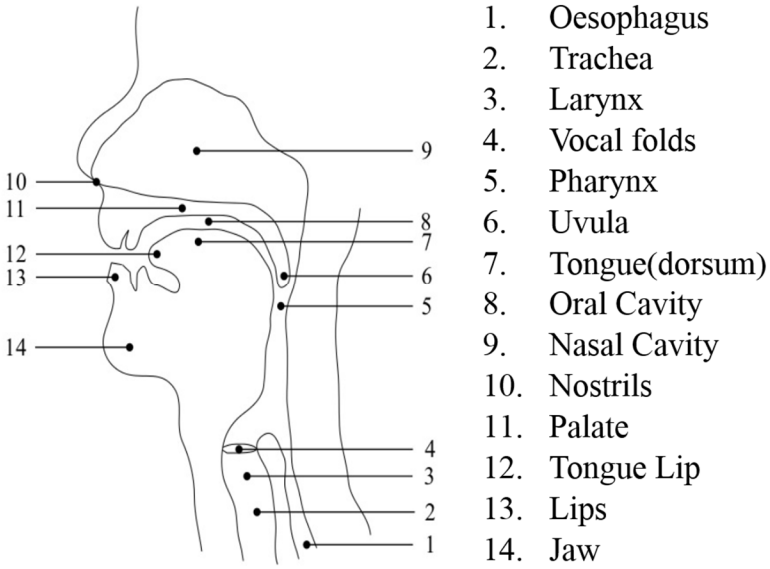

Figure 1.2. Cross-sectional view of the human vocal tract system

The system's three cavities might be referred to as the primary acoustic filter that modifies the generated sound. The articulators include the lips, tongue, jaw, teeth, and velum. These offer the more precise modifications needed to produce speech. Speech excitation can be divided into voiced, unvoiced, mixed, plosive, whispered, and silent types. To create a certain sound, any one or more can be blended together.

1.2.2. Speech Units Classification

While the physical articulation of various speech sounds was discussed in the previous section, within a hierarchy, it's crucial to be able to discern among various speech sounds.

Phonology : The examination of the nature of sound systems generally and of sound systems in particular in different languages. As such, it differs from phonetics.

In contrast to phonology, which aims to classify and categorise speech sounds more precisely and typically investigates linguistic components of languages, phonetics is only concerned with the speech sounds themselves, their characteristics, and ways of articulation.

Phoneme: The smallest distinct sound unit in a given language.

Phone: A speech sound which is identified as the realization of a single phoneme.

Syllable: A phonological unit consisting of a unit that can be produced in isolation.

Nucleus: The central element in a syllable.

The difference between these two concepts must be understood. When a phoneme is a sound unit, the word "phone" refers to the sound itself. Phonemes are smaller than syllables and comprise one or more phonemes, each of which can be produced separately.

The names "diphone," "triphone," and "pentaphone," which describe the sound of two phones, three phones, and five phones as they flow into one another, respectively, are also introduced here. Diphones, which are particularly relevant in voice synthesis and will be covered in more detail later in this study, capture the change between two phones.

A phoneme is a term used to define the linguistic meaning that a certain speech sound carries. According to figure 1.3, the 42 phonemes that make up American English can be divided into vowels, semivowels, diphthongs, and consonants (fricatives, nasals, affricatives, and whispers).

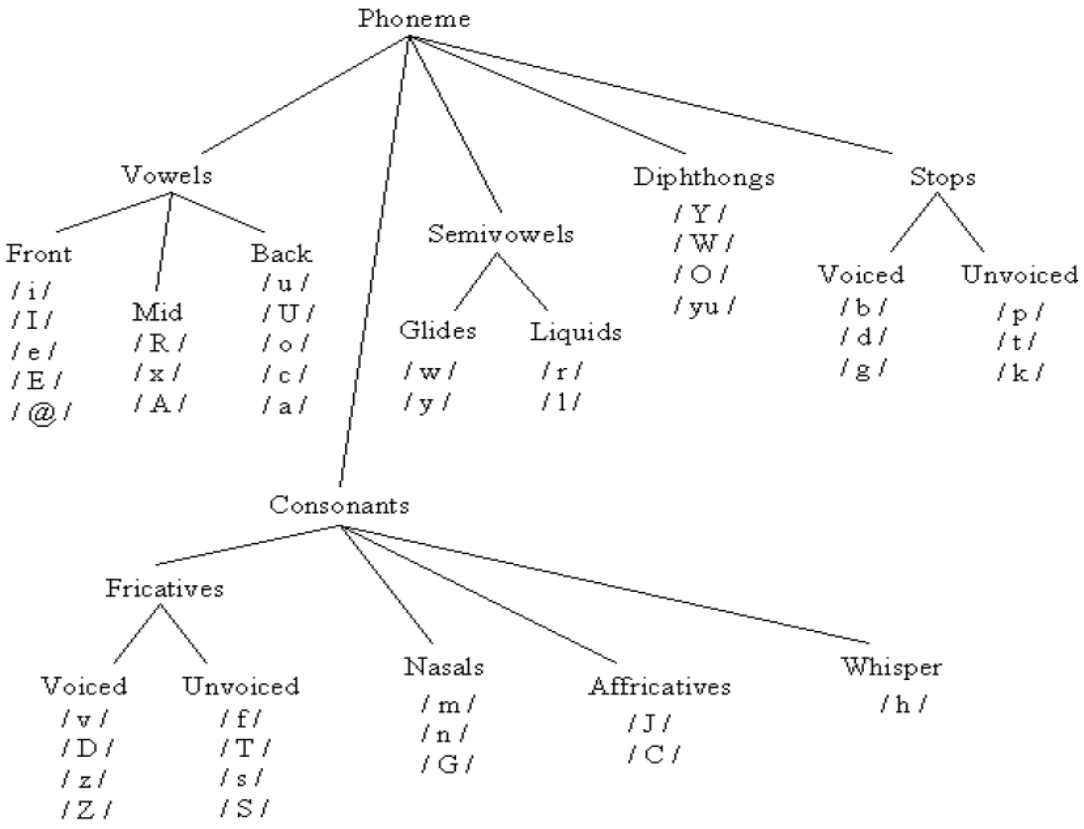

Figure 1.3. Phonemes in American English

Vowel: A basic speech sound produced through open approximation that typically constitutes the basis of a syllable. The vocal cords of the larynx periodically vibrate, producing vowels. The fundamental frequency or pitch of speech is the frequency at which the vocal cords vibrate. Formant frequencies or 'formants' are the resonant frequencies that the vocal tract generates. The vocal tract's length and shape affect the formants.

Monophthong: 'A vowel whose characteristics remain constant throughout a single syllable'.

Diphthong: 'A vowel whose quality noticeably shifts in one way during the course of a single syllable'.

Triphthong: 'A vowel that undergoes two subsequent changes in quality within a single syllable'.

Hiatus: ‘A separation between vowels from various words or syllables’.

Language understanding depends heavily on the different kinds of vowel transitions; in English, particularly important can be the distinction between a Diphthong and a Hiatus. A hiatus transition, for instance, occurs when the initial "i" and the final "o" in the word "iodine" belong to different syllables rather than transitioning inside the same syllable. The letters "o" and "u" in the word "noun" each contribute to a separate sound while remaining within the same syllable, forming a diphthong.

Consonant: A phonological component that contributes to a syllable in places other than its core.

The **fricatives** of air moving through the vocal tract's small constrictions produces the fricatives, which sound like random noises. By lowering the velum, the **nasal** cavity is acoustically coupled to the pharyngeal cavity, resulting in nasals. **Plosives** are made when pressure builds up in front of the vocal tract and is suddenly released.

Semivowel: A unit of sound that sounds phonetically like a vowel but occupies a consonant's position in the syllable structure. Semivowels are frequently classed with vowels in articulatory classification systems because of their phonetic similarity, but they can also be placed with consonants in a more phonological categorization. **Liquid**: A general name for 'r's and 'l's, particularly in languages where their phonological functions are comparable. **Glide**: A transition between sounds that can be heard, usually between semivowels.

Stop: Consonant that temporarily blocks airflow during articulation, such as [p] and [t] in pit. When air leaves the lungs, a stop is produced called **Plosive**, like the [t] in tea. In contrast, an **Affricate** is a stop consonant that is delivered at the same point of articulation as a fricative.

The scripts used for the Indian languages have their origins in the ancient Brahmi alphabet. Writing systems start with characters, which are orthographic representations of speech sounds. vowel (V), consonant (C), and combinational sounds like CV, CCV, and CVC for characters that are close to syllables used in Indian scripts frequently. In Indian languages, there are roughly 18 vowels and 35 consonants [5].

The phonetic nature of Indian language scripts is a significant characteristic. What is written and what is spoken generally correspond in a one-to-one manner. The procedures needed to convert Indian languages' letters to sounds are remarkably simple. The scripts of all Indian languages are similar phonetic bases.

1.2.3. Features of Speech

We previously discussed phonemes, which enabled us to examine the acoustic features of discrete speech segments. The quality of speech that become significant when examining bigger phonetic groupings, such as those of various syllables within a word or various words within a sentence, are defined in this section. The term "suprasegmental" or "prosodic" components of speech is used to describe these. The prosodic components of speech are stress, pitch, loudness, timbre, length, and pause. These prosodic aspects of speech are primarily what determines the TTS systems' level of quality. Prosody is essential for transferring information to speech that goes beyond the literal meaning of the words being said.

A sentence's prosody might convey the speaker's mood, irony, sarcasm, or the need to emphasise certain points. In everyday speech, people will naturally modify these prosodic features to provide more meaning to the uttered word. In a subsequent section of this paper, this will be covered in more detail.

1.3. Types of TTS Systems

In this section we briefed on the history of various TTS systems that have evolved over a few decades of Research in TTS. They are classified as Limited domain and Unrestricted domain TTS systems based on the number of words and sentences which are used to build. Limited domain TTS systems are voice built specifically for an application, such are weather forecasting, speaking clock, Air/Rail travel information systems, agriculture information system which use a limited set of words or sentences. The unrestricted domain TTS systems

use a generic voice capable of reading everything, such as storytelling, news reading, desktop assistant, etc

Particularly over the past few decades, artificial speech has been created steadily. The formant [6], concatenative [7], [8], and articulatory [9] synthesis techniques are the three fundamental synthesis methods.

Formant synthesis, which is based on the modelling of vocal tract resonances, has likely been most widely employed in recent years. Concatenative synthesis, on the other hand, is more common and is based on playing prerecorded samples of actual speech. Arguably the most accurate method is articulatory synthesis, which closely resembles the "human speech production system" but it is also the most challenging strategy.

The most in-depth study has been done on the statistical parametric voice synthesis method so far [10]. As we can see, there are several ways to improve spoken output using SPSS synthesis. Contrary to USS, its more complex models offer comprehensive solutions without always needing recorded speech in any phonetic or prosodic settings.

To cover all of the necessary prosodic, phonetic, and aesthetic variants, the unit selection based voice synthesis requires very large datasets, which are challenging to collect and maintain. In contrast, because SPSS allows for model mixing and modification, it is not necessary to offer examples of any possible combinations of settings. TTS systems are also constrained by a number of characteristics that pose fresh difficulties for researchers. 1) The speech statistics that are currently accessible are not entirely accurate 2) There are inconsistent recording circumstances 3) The material's phonological balance is not optimal. The ability to quickly change the system with only a few phrases of data would seem to be an attractive study area. It can be noted that statistical parametric voice synthesis has a "processed quality" to it yet is still very understandable.

It is difficult for researchers to control quality of the voice interms of naturalness and , understandability, which is crucial for speech synthesis applications. The methodologies of unit selection and statistical parametric synthesis each have benefits and limitations. However, a third technique that can retain the benefits of corpus-based and Hidden Markov Model-based synthesis and make a synthetic speech that is extremely similar to genuine speech could be

created by properly combining the two approaches. To use the best aspects of the two methods, a more thorough review and analysis, as well as integration of HMM-based segmentation and labelling for database development and HMM-based search to pick best-suited units, will be helpful. A DNN-based system outperforms with comparable amounts of parameters, even if it increases computing cost, and offers an alternative method to get over the HMM's drawback (the inability of decision trees to effectively describe complicated context relationships).

1.4. Evaluation of TTS Systems

While building a speech synthesis system, speech data is segmented into training and held-out test set. The held-out test set is used to assess the model's performance after it has been built using training data. The quality of the TTS system is evaluated on subjective and objective metrics. Below, we discuss various metrics that are often used to measure the quality of speech systems.

1.4.1. Subjective Metrics

Mean Opinion Score [1]: The most popular subjective evaluation method is Mean Opinion Score (MOS). In this the listeners rate the synthetic voices' speech quality on a range of 1 to 5 using a 5-point scale listed in table 1.1.

In **Degraded MOS (DMOS)** tests, a natural sentence is played first, then its synthesised counterpart, followed by a brief pause. The listeners are then asked to rate the degradation of the synthesised sentence relative to the natural sentence on a scale of 1 to 5 listed in table 1.1.

Table 1.1 MOS and DMOS Scales

Scores	Quality scales	
	MOS	**DMOS**
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Extremely annoying

1.4.2. Objective Metrics

The metrics, a) Mel-cepstral distortion (MCD) for the spectrum, b) RMSE for F0, c) the percentage error in frames for “voiced/unvoiced”, and d) “Euclidean distortion” for “band aperiodicity” are among the objective metrics.

Spectral Distortion: Spectral distortion is computed between the synthesized speech spectra and the natural spectra of a held-out test set. Depending upon the representation of the spectrum, i.e., Mel-cepstrum (MCEP) or Line-spectral pairs (LSP) distortion is computed using a Euclidean-like distance measure. When using MCEP as the spectral representation, a popular measure known as MCD is computed to evaluate the systems and is given by :

$$\text{MCD} = (10 / \ln 10) * \sqrt{2 * \sum_{l=1}^{25} (C_l^n - C_l^s)^2} \quad (1)$$

where C_l^s and C_l^n are the synthesized and the natural utterances l^{th} coefficient respectively. Typically, a better synthesis system is implied by a lower MCD. The MCD of a typical excellent synthesis system will be between 4 and 5 dB.

Aperiodicity Distortion: When the speech signal is additionally parametrized by aperiodicity along with spectrum and *F0,* Euclidean distance is computed between synthetic and natural parametric sequences to measure aperiodicity distortion.

F0 **RMSE**: *F0* distortion is computed using the Root-Mean-Square Error (RMSE) between natural and synthesized *F0* sequences

VUV (% Error): Voiced/Unvoiced decisions are compared frame-wise with the actual ground truth VUV decision and the error (how many voiced frames are projected to be unvoiced, and vice versa) is calculated.

1.5. Motivation

In recent decades, hidden Markov models have become an accepted basis for statistical parametric speech synthesis. With only a little training database and a more flexible voice than concatenative systems, this approach has considerably created voices that sound reasonably genuine and adaptable. For the purely prosodic approach of unit-selection synthesis, very large datasets are required to encompass examples of all relevant prosodic, phonetic, and stylistic variants. Contrarily, because it enables models to be combined and modified, SPSS does not require instances of any hypothetical combinations of settings..

Although this method has had great success recently, there are still a number of unresolved issues in this expansive subject. For instance,

- The prosodic parameters like F0 estimation, duration, and voicing detection, could not be modelled efficiently to obtain high-quality intelligent speech [11], [12], [16].
- Large Footprints are required to build Unit selection synthesizers [13], [14].
- Although the operation of SPSS based on HMMs is outstanding, its naturalness (Buzzy Quality) still lags behind that of naturally occurring human speech. [10], [15].
- Complex context relationships cannot be accurately modelled using decision trees, leading to overfitting and degrading quality [15], [16].

- Making network architecture and training techniques impact the performance and leads to quality degradation in parametric voice synthesis [11], [16].

To overcome the above constraints and open challenges, there is a need to i) model prosodic features like F0 Estimation, duration, and Voicing detection to improve the quality of the TTS system with a small dataset, ii) investigate different deep learning architectures like DNNs, SRNNs, LSTM-RNNs, GRUs and Elman Recurrent Neural Networks for efficient acoustic modelling of complex context dependencies to build high-quality, small dataset TTS system. The Exploration of the same is the motivation for this research work.

1.6. Problem Statement

TTS systems that have to model the prosodic features like F0 Estimation, duration, and voicing detection to improve the quality with small dataset and to vary in speaking style, have to improve the intelligibility of the synthesized speech by incorporating different deep learning architectures in order to map the linguistic features on to statistics of acoustic features. And another issue is the modelling of complex context dependencies.

To increase the TTS system quality, as well as effective modeling of parameters such as context dependencies, etc., different deep learning architectures like DNNs, SRNNs, LSTM – RNNs, GRU, and ERNNs investigation are required.

1.7. Objectives

The objectives of the work are as follows

- Improving the naturalness of synthesized speech by modelling the Prosodic parameters.
- Implementing the statistical parametric methods for improving the intelligibility of synthesized speech.

- Developing a SPSS system for the Telugu language by incorporating different deep learning architectures for mapping linguistic features to statistics of acoustic features.

- Developing a small footprint, high-quality TTS for English and Telugu languages

1.8. Contributions of the thesis

The following are the thesis' contributions:

- The first contribution attempts to model the prosodic features for a high-quality unit selection based TTS system for the Telugu language; a unique Prosody Modeling for Improvement in the Telugu language TTS System [1**] quality with low database is provided.

- The second contribution focuses on different deep learning techniques for efficient acoustic modelling of complex context dependencies to produce high quality, small dataset TTS system. DNN, Simple RNN, Gated RNN, and LSTM-RNN architectures were focused. By adding these deep architectures for mapping linguistic elements to statistics of acoustic features and prosody to enhance the naturalness of synthesised speech, the association between input texts and corresponding acoustic realisations was modelled. The given text is converted into phonetic and acoustic notation during synthesis, and MCEPs are predicted using the aforementioned NN models. The lengths and fundamental frequencies of each phoneme are obtained from the test phrase database. STRAIGHT vocoder uses the original F0 and predicted MCEPs to produce speech. These architectures [2**] further improved the quality of speech. These models are also robust.

- The third contribution focuses on the limitations of the deep architectures investigated such as Acoustic modeling, Contextual representation and Over-smoothing in order to improve the quality further. The few problems that arose were with network architecture

and training methods. These must be fixed to perform better for NN-based SPSS. The use of Deep ERNNs [3**] can perform better than with gated RNNs despite more complex RNN designs like LSTM and variants like GRU and simplified LSTM.

- To further enhance the TTS system's quality, an investigation of encoder – decoder combination was done for the English language with Tacotron2.

The prosodic features are modified in PRAAT/Wavesurfer tool for adjusting the pitch and intensity contours. The different deep architectures are modeled in MATLAB for mapping linguistic features to statistics of acoustic features. With a 5ms frameshift, 235 coefficient vectors are predicted for each 10ms frame size with 529 input textual features.

1.9. Organization of thesis

The thesis presents a modification of prosodic features and investigations of different Deep Learning Architectures for efficient acoustic modelling of complex context dependencies to produce high quality, small dataset Telugu language TTS System. The thesis is organized into seven chapters. The following section gives a summary of the chapters.

Chapter 1 provides the introduction, motivation, objectives, problem statement, and thesis contributions.

Chapter 2 reviews the state-of-the-art of problem. History of text-to-speech synthesis systems with different speech features, languages and different databases in use.

Chapter 3 discusses the modeling of prosodic features: pitch and intensity contours at epoch level to enhance the quality of the unit selection synthesis system with less dataset.

Chapter 4 describes how DNN and various RNN based SPSS Synthesis systems out perform vis-à-vis conventional SPSS systems based on HMMs. The STRAIGHT vocoder is used to produce the speech with the original F0 and predicted MCEPs.

Chapter 5 DNNs are robust models, but their performance in parametric voice synthesis is limited by a few problems with network architecture and training methods. This chapter pays the attention to three issues (Acoustic modeling, Contextual representation, and Over-smoothing) that must be fixed to perform better for NN-based SPSS. The use of Deep ERNNs enables the technology to perform better than with gated RNNs despite more complex RNN designs like LSTM and variants like GRUs and simplified LSTM.

Chapter 6 explains the End-to-End TTS system, which uses an encoder-decoder structure with an attention mechanism and for alignment learning. It anticipates how long each input token will last and outputs an audio-aligned representation..

Finally, **in Chapter 7,** the conclusions of the thesis are presented based on the contributions, and brief discussion with regard to suggestions for future research is provided.

Chapter - 2

Literature Survey

This chapter provides the literature on conventional speech synthesis approaches, neural networks based SPSS systems, and novel paradigms such as end-to-end TTS systems. Initially, the development of basic TTS system is discussed. Then, a detailed description of basic approaches is provided, which will form the underlying theory of algorithms developed in later chapters. The modelling of prosodic parameters for improving the TTS system's quality is briefly described, and the currently used/trending methods such as deep learning based statistical parametric speech synthesis to produce high quality voice as natural voice are discussed. Finally, End-to-End TTS system framework is shown in brief. The drawbacks and limitations of each system are discussed briefly. The speech database description and the databases used in the thesis are also discussed in brief.

The issues identified from the literature survey and challenges in the development of high-quality TTS system are provided.

2.1. Introduction

The research on speech synthesis started a few decades ago. Speech synthesis is the process of translating text into spoken language. The generated speech, called synthetic speech has been progressively improving over the previous few decades. A TTS system's purpose is to synthesize speech with intelligibility and naturalness from a given text. TTS systems have a variety of uses, including “public announcement systems”, “speech-to-speech translation”, and People having voice disorders [17], like blind people education using a TTS system with a screen reader as assistive technology [18]. Various synthesizers were developed based on the three primary mechanisms for synthesis. These mechanisms are Formant synthesis (Synthesizers like OVE [19], advanced versions like DECTalk [20], MITalk [21]), concatenative synthesis (ART v-talk [22], CHATR [23]), and articulatory synthesis (Dynamic analog speech synthesizer [24]). Unit selection synthesis shows natural sounding synthetic speech but requires the large dataset. Statistical parametric speech synthesis [25], [26], [27], [28] is currently the most thoroughly researched method for speech synthesis. Neural network based systems [29], [30], [31], [32], are robust but increase computational cost. The novel paradigm: End-to-End TTS system [33], [34] is built with limited data and shown good quality more or less Unit selection system.

2.2. Types of Text-To-Speech Synthesis Systems

Several TTS technologies have evolved in the last few decades. The basic approaches are formant, articulatory and concatenative synthesis. The unit selection synthesis system is better and more natural but lacking in speaking style and large dataset usage are the limitation [35] and drawback respectively. To overcome these limitations, the hidden markov model based statistical parametric speech synthesis was evolved but this model is unable to model the context dependencies which leads to buzzing sound as output. HMMs were replaced with neural networks. NN models are robust and have more computational complexity. A detailed review of these technologies is given below.

2.2.1. Formant Synthesis

The very first synthesis method to be developed was formant synthesis, which dominated the field until the early 1980s. A common name is “synthesis by rule” for formant synthesis. The fundamental tenet of formant synthesis is to simulate formant frequencies and amplitudes to replicate the vocal tract transfer function. There are some prominent resonance frequencies in the vocal track [36]. Formants are the resonant peaks in the vocal track transfer function that change frequency as the vocal tract structure varies

The synthesis, which is based on mathematical models of the human speech organ, is a type of source-filter approach. Using the acoustic-tube principle, the formant synthesiser creates sounds from a source that is either periodic for voiced sounds or white noise for obstruent sounds. The vocal-tract model is then fed this fundamental source signal. To produce a speech pressure waveform, this signal travels via the oral cavity and nasal cavity before passing through a radiation component that simulates load propagation characteristics.

Speech that sounds unnatural and robotic is produced through formant synthesis technology. Even at very high rates, speech synthesised with formants is consistently understandable. Particularly for today's computing systems, formant synthesis does not require a lot of computational power [7]. The relative simplicity of formant synthesis and the tiny amount of memory required for the engine and its speech data make it effective. The primary benefit for embedded and mobile computing applications is this. The well-known TTS engines including DecTalk [20], MITalk [21], Apollo, Orpheus[38], and Eloquence[39] are the formant synthesisers

2.2.2. Articulatory speech synthesis

Creating speech using mechanical and auditory models of speech generation is often called Articulatory speech synthesis. A voice signal with preset acoustic qualities is produced via articulatory speech synthesis using a vector of anatomical or physiological data [40]. Based on mathematical representations of the structure ("Lips, Teeth, Tongue, Glottis & Velum") and

operations ("airflow movement through the supra-glottal cavities") of speech, it generates an entire synthetic output. Memory needs are minimal because of how computationally intensive this method is [41].

Many smaller uniform tubes are used in acoustic models to produce natural speech. These tubes are self-contained and autonomous. Natural movements in tubes can generate intricate speech patterns, avoiding the challenges of explicitly modelling complex formant trajectories. The motion of the tubes is controlled by a straightforward mechanism (mechanical dampening or filtering) in the interim stage of articulatory synthesis models, which is meant to simulate the fact that the articulators move at a specific inherent speed. Following that, the parameters for the specification-to-parameter component are provided using this motor-control space.

In articulatory synthesis, there are two problems that need to be solved: how to create the control parameters from the specification, and how to strike a compromise between a more realistic representation that is simple to construct and control and a very accurate model that closely resembles human physiology [42], [43].

2.2.3. Concatenative Synthesis

Speech signal analysis of natural speech databases is necessary for concatenative synthesis. The segmental database was created to reflect a language's main phonological characteristics. Concatenation approaches combine sequences of these small speech units, such as waveform data or acoustically parameterized data, to create either time varying acoustic parameters. A speech synthesiser must then process the time-varying acoustic data to produce a waveform. Festival speech synthesis system [44] is the well-known synthesiser for building the concatenative speech and there are many systems evolved in like-wise. The selection of the appropriate units and the connecting algorithms are key considerations in a concatenation system. In order to meet established prosodic schemes and to smudge unit transitions, it also does some signal processing. On these three essential types, concatenative synthesis depends.

a) Diphone synthesis

b) Domain synthesis and

c) Unit selection synthesis

The description of these methods is briefed in the following sections

2.2.3.1. Diphone Synthesis

The most common technique for generating a synthetic speech from recordings or voice samples of a specific individual is diphone synthesis [45]. A nominal speech database is employed. The database's diphone count is influenced by the language's phonotactics. The prosodic model and phrase or expression strength are both factors in diphone synthesis. Due to poor modelling, the speech could come off as somewhat monotonous. In languages with inconsistent pronunciation norms and in specific situations where letters are pronounced differently than they are generally, a diphone synthesis is not effective. For languages with strong pronunciation consistency, the diphone performs better.

Discontinuities at the intersection of two vowel bisects are the main issue with diphone synthesis [46]. In some instances, they produce a diphone boundary discontinuity and a bi-vocalic sound quality.

2.2.3.2. Domain based Synthesis

Using prepared vocabulary and axioms, domain-based synthesis creates whole utterances [40]. It is utilised in settings where output speech is constrained to a certain field, such as the transport schedules announcement and weather reports announcement. These systems can only combine vocabulary and axioms with preprogrammed content since they are constrained by the vocabulary and axioms in their databases.

2.2.3.3. Unit Selection Synthesis

USS is the most often utilised synthesis technique [1], [14], [35]. This method addresses how to manage huge numbers of units, prosody improvement and time control, and reduce

distortions brought on by signal processing. It is an extension of the second generation concatenative method.

In order to capture more natural variation and rely less on signal processing, USS employs a wide variety of speech [7]. A feature structure, which can be any combination of verbal and aural information, offers a full description of the specification and units. An algorithm selects one unit from the available options in an effort to find the best overall sequence of units that matches the specification. The carrier speech is used during synthesis to reduce issues with designing and recording a database that generates a unit for each feature value.

The length of the employed speech segments is where a unit selection and a diphone voice differ most. The unit database contains whole words and sentences. This means that compared to the database for diphone sounds, the database for unit selection voices is much larger. As a result, CPU usage is low and memory usage is high.

2.2.4. Statistical Parametric Speech Synthesis

Due to its scalability and scale, SPSS has increased in popularity in recent years. The primary elements for SPSS of speech are MCEPs, F0, and Duration. The development of parametric voice synthesisers, also known as synthesis-by-rule, began in the early 1980s. Careful parameter selection and a set of rules for parameter manipulation were used. Machine learning methods are used by Statistical Parametric Synthesis (SPS) to learn the parameters from the features retrieved from the speech signal [47]. The statistical parametric synthesis engines HTS [15] and CLUSTERGEN [48] use hidden Markov models and Classification and Regression Trees (CART) to learn the parameters from the speech data. In the SPS framework, harmonic noise models features, line spectral pairs, and cepstral coefficients are frequently used to describe spectral features. Strengths of the fundamental frequency and voicing are examples of excitation characteristics. Speech signal is produced using source-filter models using spectral and excitation properties. As we can see, statistical parametric synthesis provides a variety of methods for enhancing spoken output. Its more complex models allow for generic answers

without necessarily requiring recorded speech in any phonetic or prosodic contexts, in contrast to unit-selection synthesis.

2.2.4.1. HMM based SPSS

Hidden Markov model (HMM)-based voice synthesis is a SPSS that makes use of an HMM as its generative model. HMMs represent numerous language contexts in addition to phoneme sequences, which is similar to the unit selection approach. In order to produce a speech waveform, a vocoder, a simplified model of speech generation where speech is represented by vocal tract parameters and excitation parameters, is driven by acoustic parameters produced from HMMs selected in accordance with the linguistic specification [49], [50].

The probability densities of speech parameters given texts are commonly represented by decision tree-clustered context-dependent HMMs in traditional techniques to SPSS [47]. To create speech parameters, which are then utilised to reconstruct a speech waveform from the obtained parameters, probability densities are first used. The quality of the synthesised speech is one of its major drawbacks, though. Decision trees in HMM are ineffective in simulating intricate context dependencies, which results in overfitting and deteriorates performance. It showed successful in good voice quality with some busyness. To overcome these limitations of HMM based SPSS, an efficient architecture has to be modeled with proper training.

2.2.4.2. NN based SPSS

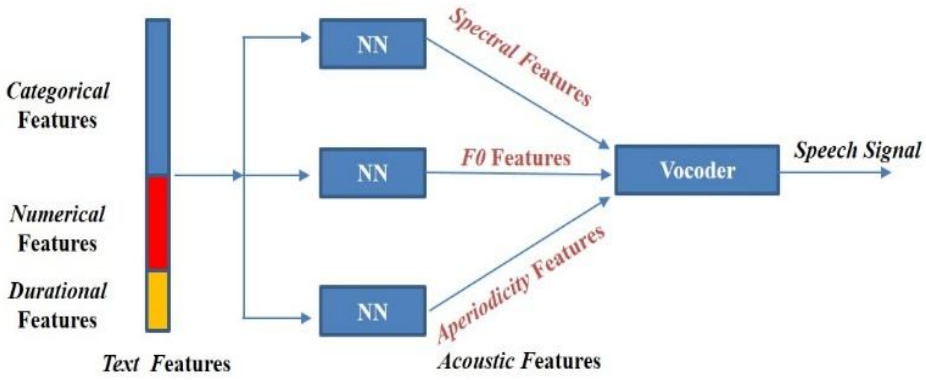

Figure 2.1: Simplified block diagram of Neural Networks based SPSS system

Figure 2.1 illustrates the simplified NN based SPSS system. In addition to one or more neural network models that are used to anticipate MCEPs, which are also included in the TTS system, it also includes a text-to-acoustic and phonetic analysis subsystem. The text provided is converted into phonetic and acoustic notation during synthesis, and MCEPs are predicted using the most recent models. For this work, the fundamental frequencies (F0) and durations of each phoneme are obtained from the test phrase database. STRAIGHT vocoder uses the original F0 and predicted MCEPs to produce speech.

Advancements in both software and hardware, training the different deep learning architectures for better acoustic modeling with good chunk of data got successful [51], [52]. The acoustic-articulatory inversion mapping and Speech recognition are two machine learning applications where deep neural networks have outperformed traditional methods. Despite the fact that DNNs are strong models, there are a few issues with network architecture and training techniques that affect how well they perform in parametric voice synthesis . We draw attention to three problems (Acoustic modeling, Contextual representation and Over-smoothing) that must be resolved in order to enhance the performance of neural network (NN) based SPSS in this research work. The use of ERNNs gave computational effectiveness without compromising quality [53] which are reviewed in the following section.

2.2.4.3. RNNs for SPSS

Various RNN units are briefly reviewed. Limitations of DNNs include poor Acoustic modeling, Contextual representation and Over- smoothing that must be fixed to perform better for NN based SPSS. To prevent vanishing gradients problem, RNNs are typically developed using either GRUs [56] or LSTMs [54], [55] as RNN units. But according to [57], ERNNs receive preference over LSTM/GRU/SLSTM: (1) For a given hidden state size, gated recurrent architectures have significantly more parameters than ERNN, and because the recurrent unit has more gates, there are also significantly more computational steps. (2) It is not clear which components of the complex architectures are crucial. We demonstrate that, despite more complicated RNN designs like LSTM [58] and variants like GRU [59], [60], and simplified LSTM [60], ERNNs can perform on par with gated RNNs. ERNNs [61] are networks that lack sophisticated gating mechanisms yet have hidden state[62].

2.2.5. Novel paradigms: End-to-End systems

Recently there have been attempts to substitute vocoding, text feature extraction, and duration modeling using sequence-to-sequence (seq2seq) neural networks leading towards end-to-end text-to-speech synthesis [63] [64] [34]. In [63], authors proposed Wavenet architecture which directly predicts the samples from text features there by replacing the vocoding in SPSS. Wavenet was shown to perform better than USS and SPSS. In [64], authors have proposed Char2Wav method that combines seq2seq and SampleRNN [65] architectures to predict samples from characters. Another seq2seq model (Tacotron) which can be trained from scratch taking characters as input and raw spectrogram as output was proposed in [34]. The main structure blocks in the Tacotron [34] (an end to end speech synthesis) are encoder, attention module, decoder, post- processing net and are shown in figure 2.2.

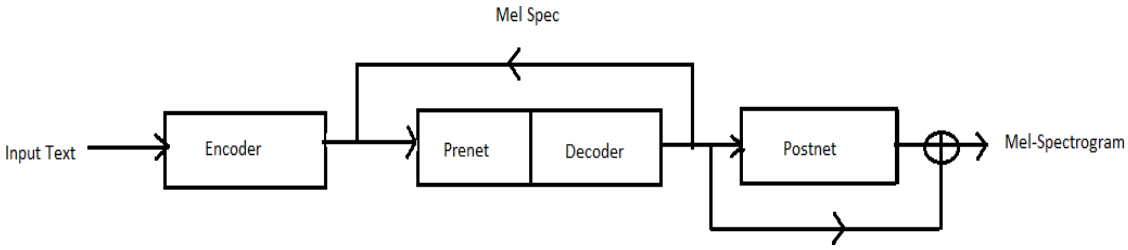

Figure 2.2: Simplified Encoder – Decoder Framework

Encoder: The encoder of tacotron extracts the robust and successional hidden representation from the input words. The input character sequence is also converted by the embedding layer. Some metamorphoses are applied inclusively called “pre-net” to each embedding. These pre-net products are now converted into final encoder representation by the CBHG module. A bank of 1-D convolutional filters, trace networks, and a bi-directional recurral neural network make up the CBHG module.In the CBHG module the input sequence first performs convolution operation with 1-D convolution filters. The reprocessed output data of the convolution layer is fed into a single bi-directional LSTM layer to induce isolated representation of input sequence.

Attention module : For each generated phoneme, an attention medium selects or weighs the signals produced by a trained feature extraction medium at potentially all of the time in the input sequence (speech frames The coming element in the output sequence is also conditionally generated using the weighted point vector. The portion of the encoder data that should be utilised at the current decoder step is defined by attention..

Decoder: The decoder takes the hidden representation of the input sequence from the attention module, the pre-net contains 2 completely connected layers, to this pre-net the former mel spectrogram frame is given as input while the output of this pre-net is concatenated with the former vectors and it is given to the decoder LSTM. In order to prognosticate the Mel spectrogram and stop token the output of the LSTM is given to 2 different linear protrusions. These spectrograms are fed to a post-net to upgrade the overall output.

Post processing net and waveform conflation: Transforming the processed data into the relevant waveforms is the primary goal of the post-processing network. In contrast to sequential to sequential, which always proceeds from left to right, it has both forward and

backward information to correct the vaticination error for each particular frame. It uses the Griffin Lim reconstruction for the phase frame. Post processing unit also has the CBHG unit. The CBHG module plays a pivotal part in the vocoder parameters and storage purpose.

Mel-Spectrogram: A Mel-Spectrogram differs significantly from a conventional spectrogram in two key ways. The latter graphs Frequency vs. Time. Mel Scale is used on the y-axis rather than Frequency.

Tacotron 2 uses Venila LSTM and convolutional layers in place of GRU recurrent layers and CBHG stacks in the encoder and decoder. Tacotron 2 doesn't use a "reduction factor"; instead, each decoder step is equivalent to a single spectrogram frame. Location-sensitive attention is utilised in place of additive attention.

The end to end TTS systems based on TACOTRON-2 [66], Deep voice-3 [67] and FASTSPEECH [68] showed good quality more or less Unit selection system with less data set for English language.

2.3. TTS for Indian Languages

The scripts used for the Indian languages have their origins in the ancient Brahmi alphabet. Writing systems start with characters, which are orthographic representations of speech sounds. Vowel (V), consonant (C), and combinational sounds like CV, CCV, and CVC for characters that are close to syllables used in Indian scripts frequently. In Indian languages, there are roughly 18 vowels and 35 consonants [5]. We need to segment the speech into syllable-like units to built the TTS system. According to perceptual data, which demonstrates that there are four options for speech units: syllable, diphone, phone, and half phone, the syllable unit performs better than all the others and is the better illustration for Indian languages. [1], [5], [69]. It has been noted that the spectral and prosodic patterns for each of the distinctive syllables change depending on where the syllable falls inside a word. The energy and pitch patterns of the syllables at the beginning of the word are rising, while those at the conclusion

of the phrase are gently tapering. The energy and pitch contour fluctuations for a syllable in a monosyllabic word are distinct from those for the same kind of syllables found in the starting, middle, and end positions.

The phonetic nature of Indian language scripts is a significant characteristic. What is written and what is spoken generally correspond in a one-to-one manner. The procedures needed to convert Indian languages' letters to sounds are remarkably simple. All Indian scripts share a shared phonetic foundation [14, 15].

Indian languages (Hindi, Gujrati, Telugu, Kannada, etc.) are not supported by the character sets (Windows default, UNICODE) that are currently available for computers. In order to operate with Indian languages, we need special typefaces (from several vendors, such as Ankit (Hindi) and Tikkana (Telugu)). Character sets like Windows default or UNICODE are still used by these fonts. Their graphical portrayal, however, will differ. For instance, the Telugu symbol "@" and the Hindi character "v" both stand in for the vowel V_A. The Unicode format for the telugu language is shown below in figure 2.3.

అ a	ఆ aa	ఇ i	ఈ ii	ఉ u	ఊ uu	ఋ rx	ఌ lx	ఎ e	ఏ ei
ఐ ai	ఒ o	ఓ oo	ఔ au	అం an:	అ: ah:				
క ka	ఖ kh	గ ga	ఘ gha	ఙ ng~a	చ cha	ఛ chha	జ ja	ఝ jha	ఞ nj~a
ట t:a	ఠ t:ha	డ d:a	ఢ d:ha	ణ nd~a	త ta	థ tha	ద da	ధ dha	న na
ప pa	ఫ pha	బ ba	భ bha	మ ma	య ya	ర ra	ఱ r:a	ల la	ళ l:a
వ va	శ sha	ష shha	స sa	హ ha	క్ష kshha				

Figure 2.3: UNICODE Format for Telugu from English

2.4. Evaluation of TTS Systems

While building a speech synthesis system, speech data is partitioned into training and held-out test set. Training data is used to build the model and the held-out test set is used to evaluate the performance of the model. We discuss various criteria that are often used to measure the quality of speech systems.

The objective metrics used to assess the effectiveness of TTS systems include MCD for spectrum, RMSE for F0, % error in frames for voiced/unvoiced, and euclidean distortion for band-aperiodicity [28], [70], [71], [72], [73].

The subjective measures are based on the listening tests with some number of listeners. There is natural speech signal as reference speech file is hidden amongst the test cases. The listeners aren't involved in developing any speech systems. The most popular subjective evaluation, the Mean Opinion Score, asks listeners to rate the synthesised voices' speech quality on a scale of 1 to 5 and is given in table 1.1. Additionally, MUSHARA ('MUltiple Stimuli with Hidden Reference and Anchor') is a subjective evaluation metric that asks the listener to rank how similar the test and reference systems sound in terms of naturalness. The range was 0 to 100, with 100 denoting a highly natural synthesis and 0 denoting very unnatural synthesis. A minimum of one test wavefile that sounded the most similar to the reference file should have received the highest rating from the listener.

2.5. Issues with the Existing Methods of TTS System

The major issues with existing TTS Systems identified from the literature survey are the following:

1. Synthetic speech has been studied for decades, yet it still does not sound convincingly natural [11]. Prosody is a unique flaw because it frequently has uninteresting, unsuitable, and ugly durations, intensity, and intonation. With USS, large datasets are also necessary. Due to these problems, synthesised speech cannot be used in otherwise enticing applications, such as audiobooks..
2. Hidden Markov models based SPSSS differs from concatenative speech synthesis, in that it can change voice characteristics [74], [75], has a smaller footprint [76], and is more

resilient [77]. This strategy has a few drawbacks, including the inefficiency of decision trees to express complicated context dependencies, leads to over fitting and degrades the quality.

3. According to [11], there are three key problems that reduce the quality of synthetic speech: vocoding, acoustic model correctness, and over-smoothing.
4. Making of network architecture and training techniques adversely affect the performance and leads to quality degradation in parametric voice synthesis [11], [16].

2.6. Framework for Research Work

From the issues, it is identified that, to solve the problems of existing methods of TTS Systems, there is a need to :

i) Model the prosodic features like F0 Estimation, duration and Voicing detection to improve the quality of TTS system with small dataset.

ii) Investigate different deep learning architectures like DNNS, varients of RNNs like S RNN, LSTM - RNN, Gated RNNs and Elman RNNs for efficient acoustic modelling of complex context dependencies to build high quality, small dataset TTS system.

iii) Subjective as well as objective measures, are used to evaluate the developed synthetic speech. Objective measures should be able to assess the performance of the developed methods with respect to the specific issues addressed. The developed algorithms could be used in a variety of applications like enhancing the pronounciation skills, mobile phones, hearing aids, communication systems, disabled people assistive technology etc.

2.7. Summary

In this chapter, the previous works, and the current advancements in the area of speech synthesis have been discussed. It provides the issues identified in the existing speech synthesis systems and the framework for the research work in the thesis. It is difficult to derive control parameters for articulatory synthesis from the specification, and it is challenging to strike the

correct balance between a very accurate model that closely mimics human physiology and a more practical representation that is simple to construct and control. Due to large dataset, and lack in style variation, the unit selection synthesis not used in much popular but better in quality, and intelligibility. There has always been an effort to develop speech synthsis algorithms, without compromising the quality and computational cost.

Deep learning (DNN, RNN) based systems that outperformed a similar number of parameters but increased computational costs [62], [79]. However, recent advancements in software (e.g. [80]) and hardware (e.g. GPU) allow us to train a DNN using a good chunk of training data. We illustrate that ERNNs can perform on par with gated RNNs, despite the fact that more complex RNN architectures, such as long short-term memory and variants like gated recurrent units and simplified LSTM, and gives computational effectiveness without compromising quality [51]. The end to end TTS systems based on TACOTRON-2 [66] and FASTSPEECH [68] are shown good quality more or less Unit selection system with less data set for English language.

Chapter - 3

Modeling Prosodic Features

3.1. Motivation

In this chapter, a significant focus is on modeling the prosodic features like pitch, duration and intensity of speech without affecting the quality or naturalness with small dataset. In most of the TTS systems, the quality of synthesized speech reduced due to poor modeling the prosodic parameters like F0 Estimation, duration, intensity and voicing detection. Hence there is a need to model prosodic parameters effectively [13], [45], [81]. The Unit Selection based Synthesis inventory contains a large number of fundamental units with differing prosodic and spectral features. Based on the target and concatenation costs, a USS system chooses relevant units from its inventory. We use the Festival framework to develop a USS system that is based on syllables. In any USS system, it's impossible to include all potential syllables in a local language like Telugu from all possible contexts. As a result, there is some discontinuity

in the synthesized speech at concatenation points, due to lack of units in the inventory that meet the target parameters. The fundamental frequency (F0), duration of the phones, and intensity of the speech signal are considered to be the most important parameters in determining prosody and emotion in the speech signal. In this work, to correlate pitch contour and intensity contours of adjacent syllables, prosody modeling of the syllables prior to concatenation is used. When there aren't many syllables with diverse prosodic and spectral properties available, this strategy leads to the creation of the USS system with a small footprint. After prosody correction, F0 and intensity discontinuity at the joint point is decreased, resulting in improvement of synthetic speech naturalness and intelligibility.

3.2. Prosodic Features of Speech

Prosody is essential for transferring information to speech that goes beyond the literal meaning of the words being said. A sentence's prosody might convey the speaker's mood, irony, sarcasm, or the need to emphasise certain points. In everyday speech, people will naturally modify these prosodic features to give more meaning to the uttered word. When considering larger phonetic groupings, such as those of various syllables within a word or various phrases within a sentence, some characteristics of speech become crucial. The term "suprasegmental" or "prosodic" components of speech is used to describe these.

Stress: A phonological characteristic that causes one syllable to be heard louder than others The different phonetic correlates include differences in length, perceived loudness, vowel quality, pitch, or any combination of these. In auditory terms, stress can also refer to any combination of these.

Pitch: The characteristic of sounds as heard by a listener that is similar to frequency in nature As a result, a vowel that, from the perspective of articulatory phonetics, is produced with faster vibration of the vocal cords will have a higher fundamental frequency from the perspective of acoustics and be perceived as having a higher pitch by the ear.

Loudness: The perceptual quality of sounds that somewhat matches to their decibel-based acoustic loudness or intensity.

Timbre: The auditory characteristics of sounds other than pitch and loudness; thus, vowel quality is sometimes used in phonetics.

Length: phonological or phonetic characteristic, particularly of vowels. It is possible for differences other than physical duration to fully or partially realise phonological distinctions that are presented as ones of length.

Pause: Any gap in speaking words and between parts of words.

In this chapter, we are concentrating on modification of F0 (Pitch) and intensity (Loudness) parameters in order to develop unit selection speech system with small footprint when there aren't enough syllables with differing pitch and amplitude in the database.

3.3. Prosodic Modeling Approaches

Prosody modelling, often known as prosody modification (PM), is the process of altering the pitch, length, intensity, and other prosodic aspects of speech signals without affecting their quality or naturalness. Parametric and non-parametric approaches are both available. Whether the alteration is made in the frequency or timedomain, another subdivision can be made inside the non-parametric technique group.

3.3.1. Time Scaling

The goal of time scale modification is to change the speaking rate without altering the original speech's spectral content. This implies that the time evolution of the vocal tract filter and the excitation signal must be time scaled when using the source-filter model of speech.The pace of time scale change can be altered evenly by applying a certain factor, or it can vary depending on the prosody or sound qualities of the various sections of speech. Time instants in the original signal (analysis instants) are mapped to equivalent time instants in the new signal (synthesis instants) using a time scaling function, or so-called time warping function. The

resulting time-scaled speech may be more understandable if the time scaling is not uniform [82]. Finding a mapping to improve comprehension is a difficult topic that has been addressed in [83], [84].

Concatenative speech synthesis, where the characteristics of the segments to be concatenated must be adjusted in accordance with linguistic restrictions, also uses non-uniform temporal scaling. Both uniform and non-uniform time-scaling can be accomplished using the methods for time scaling that are detailed in the following subsections.

3.3.2. Pitch Scaling

The fundamental frequency of the excitation signal and the fundamental frequency of pitch are related by the vocal chords' fundamental frequency of vibration.Only spoken or partially voiced speech is defined as having it. Pitchscale adjustment aims to change the fundamental frequency while leaving the signal's spectral envelope and formant structure untouched. Pitch scaling therefore requires splitting the voice signal into a source signal and the filter coefficients representing the vocal tract.You can scale pitches equally or unevenly. When pitch is scaled uniformly, it means that the pitch is changed by a fixed amount across the entire speech signal. As a result, the intonation stays the same. The intonation of the voice utterance can be altered by adjusting the pitch in a non-uniform manner. It mostly finds use in concatenative speechsynthesis wherein, depending on the language context, a specific tone is placed on the utterance

3.3.3. Algorithms for Prosodic Modifications

Following, we go over various time and pitch scaling techniques that have been put forth in the literature. Parametric methods and non-parametric methods are two categories into which the methods can be separated. The voice signal is represented by a collection of parameters in parametric approaches. By altering these parameters and synthesising the signal with them, pitch or temporal scaling can be performed. The speech itself is scaled for time and pitch in

non-parametric models. The modification can be applied to time-domain or frequency-domain speech signal fragments.

3.3.3.1. Non-parametric methods

Whether the alteration is made in the frequency or timedomain, another subdivision of the non-parametric approaches can be made. Due to the omission of the operation that transforms the signal into the frequency domain, time domain approaches are typically less computationally complex. [85, 86] provide a thorough discussion of many non-parametric techniques.

3.3.3.1.1. Time domain based methods

Time domain techniques act on overlapped-add synthesised portions of the original speech waveform. First introduced by [87], the idea of temporal scaling utilising overlap and add (OLA) synthesis. Concatenating brief windowed segments that have been cut from the original signal at time instants specified by the timewarping function is how it works. While concatenating segments that are not adjacent results in time compression, repeating the same segment during synthesis more than twice is similar to time stretching. The periodic structure of the signal, which corresponds to maintaining the local pitch of the speech signal, must be retained when the segments are concatenated. Different OLA approaches offer various approaches to deal with this issue.

The Synchronised Overlap and Add approach (SOLA) [88] places the segment in the synthesised signal at a location that maximises correlation with the segment that was previously synthesised. Based on the same theory, the Waveform Similarity Overlap and Add technique (WSOLA) [89] alters the point of extraction from the original signal to maintain periodicity. To prevent discontinuities, the duration of the synthesised segment is changed in [90]. The [88-90] methods are single-purpose techniques for stretching or compressing waveforms that operate directly on the waveform. Without the requirement for an explicit pitch estimation, they employ the periodic structure, or the local pitch.

The length of the segments in the time domain Pitch Synchronous Overlap and Add (TD-PSOLA) method [46] [91] is proportional to the local pitch period. The windowed segments are a multiple of local pitch period in length. In this way, the periodicity is maintained when overlapping the synthesised segments. Pitch markers that indicate the location of the pitch pulse within each pitch period must be used to classify the speech for PSOLA. The pitch markings are uniformly spaced and fixed for unvoiced passages. To obtain these pitch marks, a trustworthy and precise pitch estimate technique is required. The PSOLA approach is ideal for both time and pitch modulation, in contrast to the single purpose methods previously outlined. The pitch synchronous weighted segments are inserted with more overlap for higher pitch or less overlap for lower pitch for the purpose of pitch scaling. This process is applied to the speech signal for TD-PSOLA without dividing it into a residual signal and filter coefficients. Due to the pitch-synchronized processing, the formant structure is unaltered, and the waveform that results from the pitch pulse on the source signal that is filtered through the vocal tract is retained.

In contrast to frequency domain methods, time domain methods are less difficult and frequently produce high-quality speech modification for moderate time scaling factors. Stretching unvoiced sections introduces tone noise, which is a significant flaw in all OLA techniques [85]. Unwanted periodicity results from the repeating of unvoiced portions because it creates a long-term correlation. Therefore, it was advised in [91] to time reverse every other segment during unvoiced parts. When applied to speech areas where the source is a combination of a periodic signal and noise, such as voiced fricatives, this results in an improvement for solely unvoiced regions but is not applicable in those regions for voiced regions.

3.3.3.1.2. Frequency domain based methods

The phase vocoder serves as the foundation for or has significant effect over frequency domain techniques. A well-known instrument for modifying time and frequency scales is the phase vocoder. Flanagan and Golden first suggested the idea. The frequency domain of speech is the focus of techniques based on the phase vocoder, such as those in [92], [93]. Scaling for both time and pitch results in modifications to the short time Fourier spectra of the overlapping signal segments. As a result of the segments' overlap, the STFT signals' order must adhere to

consistency requirements. More specifically, one must ensure that the Fourier transform phases connected to each bin in the modified STFT are maintained throughout segments.

Portnoff [93] proposed a technique for time scaling based on the short time Fourier transformation. A somewhat altered and more straightforward method is provided in [85], in which time scaling is carried out in the frequency domain by altering the evolution of the amplitudes and frequencies of the signal's STFT. In order to prevent the shifted and weighted pictures of the window's main lobe from overlapping, the cut-off frequency for the analysis window w(n) is set to be less than half the pitch harmonic spacing. The narrow-band analysis situation is what this is known as [93]. The so-called instantaneous phase and instantaneous frequency, which specify the phase and frequency in a continuous manner for each channel, can be derived from the STFT from two close enough time instants that follow one another to conclude that the phase is a developing function gradually [85].

Due to the lack of difference between voiced and unvoiced sounds in this processing, tonal effects in unvoiced areas are produced for high stretching factors. In addition, chorusing effects (the perception that several persons are speaking at once) [85], reverberation, and transient smearing (a tiny loss of percussiveness) are frequently produced by phase vocoder time scaling. These occurrences are covered in length in [92], which offers insights into their causes as well as strategies to lessen their consequences. Pitch scaling in a timely manner According to the speech production model, the Fourier transform requires splitting the speech signal into a source signal, which represents the excitation, and a time-varying filter, which represents the vocal tract. The source audio is pitch-scaled via a resampling process in the phase vocoder. In order to get the spectra at the synthesis instants, the short-time Fourier coefficients provided at the analysis instants are linearly interpolated to do the resampling in the frequency domain. [85] provides a thorough explanation of how the phase vocoder for pitch scaling is implemented. Only used for pitch scaling, Frequency Domain Pitch Synchronous Overlap and Add (FDPSOLA) [46] uses the brief Fourier spectra of windowed pitch-synchronous segments (pitch cycles) to operate. By using STFT interpolation in the same manner as for the phase vocoder, the adjustment is made in the frequency domain of the residual. The phase does not need to be altered as it is with the phase vocoder because the STFT spectra are pitch synchronised. The pitch scaling causes a time scaling that must be offset using the TD-PSOLA time scaling technique, much like in the TD-PSOLA approach. For the sake of completeness,

it should be noted that phase vocoder techniques are common tools for modifying audio signals other than speech, such as music. They offer a potent tool for computer music spectral adjustments. Reverberation and other drawbacks like an artificial sound quality are not always undesirable in these applications; in fact, they are frequently sought.

3.3.3.2. Parametric methods

Based on a certain underlying model of speech, parametric approaches analyse speech. When modifying speech, the parameters are directly used and approximated from the original speech. The model's adjusted parameters are used in a synthesis procedure to create the modified voice signal. Since the underlying signal models for the majority of parametric methods often describe the signal in the frequency domain [86] [94], [95], these methods can be thought of as frequency domain-based methods.[94] presents a sinusoidal model that combines analysis by synthesis/overlap and add synthesis (ABS/OLA). A sum of short-time signals that overlap are used to express the signal; this total is represented as a sum of sinusoidal components. Conceptually, harmonic and noise models (HNM), first put forth in [95], are comparable to sinusoidal techniques. The use of concatenative synthesis and the harmonic plus noise model. [97] presents a parametric time-scaling method based on a nonlinear oscillator model acting in the time domain of speech. To extract the so-called glottis-signal, the speech is first subjected to LP analysis and additional low-pass filtering. This system's self-oscillation is used to synthesise the changed glottis signal, which is then processed through a high-pass filter and LP synthesis.

3.4. USS System for Indian Languages

Well-known synthesizer Festival Framework [44] supports a variety of synthetic speech production approaches, including unit level selection, and Statistical Parametric Synthesis. Unit level may be diphone, phone, half phone or a syllable. Syllables can be employed efficiently as unit level for languages in India, according to [4]. In Indian scripts, a character is similar to a

syllable and can take the form of consonant (C), Vowel (V), combinations like CV, CCV, and CVC. Indian languages contain approximately 35 consonants and 18 vowels [5].

Voice models are created for several Indian languages using a variety of methodologies, and the synthesized speech quality is determined by comparing it to a natural speech unit. There have been a number of previous papers in the literature that discuss syllable-based USS systems for Indian languages mainly Telugu, Hindi, Tamil, Bengali, Odia, Gujarathi, etc. Monosyllables and bi-syllables are employed as unit level in Tamil TTS, according to [69]. In [98], a syllable - based TTS for Bengali is built. [99] uses polysyllablic units to design Hindi and Tamil TTS. Poly syllabic comprises of monosyllable, bisyllable and trisyllable. However, employing the polysyllablic unit may result in a significant increase in inventory size. Some signal adjustments are performed in [100] such as syllable classification, energy prediction and modification, silence correction, and line spectral smoothing, to reduce the artefacts seen at the joining locations of syllable-based USS.

F0 (Pitch) and intensity (Loudness) are the parameters used to calculate concatenation cost and target cost in USS since they are significantly correlated with human sense of discontinuity. As a result, picking the most appropriate unit from the collection that perfectly suits the target parameters and has the minimum spectral and prosodic distance from its neighbours is also significant to the synthesised output. The inventory of unit selection based TTS encompasses large number of basic/fundamental units with differing prosodic and spectral features. To choose these units for synthesis, they should possess lowest weighted cost (concatenation cost and target cost). Target cost is the estimate of difference between the actual chosen unit from database and the target unit, where the target unit is the syllable to be synthesized and the concatenation cost is the distance between two successive selected units [8].

3.4.1. Speech Database used for this thesis work

IIIT Hyderabad Indic Telugu text corpus [101], was used to create unit selection based TTS for Telugu employing the syllable as the basic unit. The text selection guarantees that a language's syllables with the highest frequency are covered. However, due to the practical

impossibility of covering all possible syllables in a language with all potential situations, there is an audible discontinuity in synthetic speech at the concatenation point. Around 110 minutes of Telugu speech data was recorded by IIIT Hyderabad in a quiet recording room. Telugu has 33,417 realisations of 2,291 syllable units in its speech corpus of 1000 sentences. Festvox labeler was used to label the database at the phone level. A dynamic time warping approach was used by this labeler. The label boundaries were not appropriate since accurate duration knowledge was not provided for Indian language phones. emulabel (www.festvox.org/ emulabel) was used to manually adjust these label boundaries.

3.4.2. Natural and Synthesized Speech Comparison

The behavior of synthesized speech's pitch and intensity contours is explored and compared to real speech to figure out what causes the audible discontinuity at the syllable connecting point. The constructed TTS system is used to synthesize some sentences for this purpose. The same speaker also records the same sentences. These sentences aren't in the database that was used to create the TTS system. Through headphones, both the synthesized and natural utterances are urged to be carefully listened to indicate the terms that have audible discontinuity. Pitch and intensity measures taken when syllables are joined. These measures showed a significant discontinuity when examining these indicated words.

As a result, the current work looks at pitch and intensity as characteristics that might be tweaked to lessen syllable joining discontinuities. As an example we have taken recorded utterance // bommalu vaad:ei vidhaanan: peijiilanu oka saari chadavan:d:i // and plotted its spectrogram, pitch contour and intensity contour along with transcription labeling in figure 3.1. using Wave surfer software.

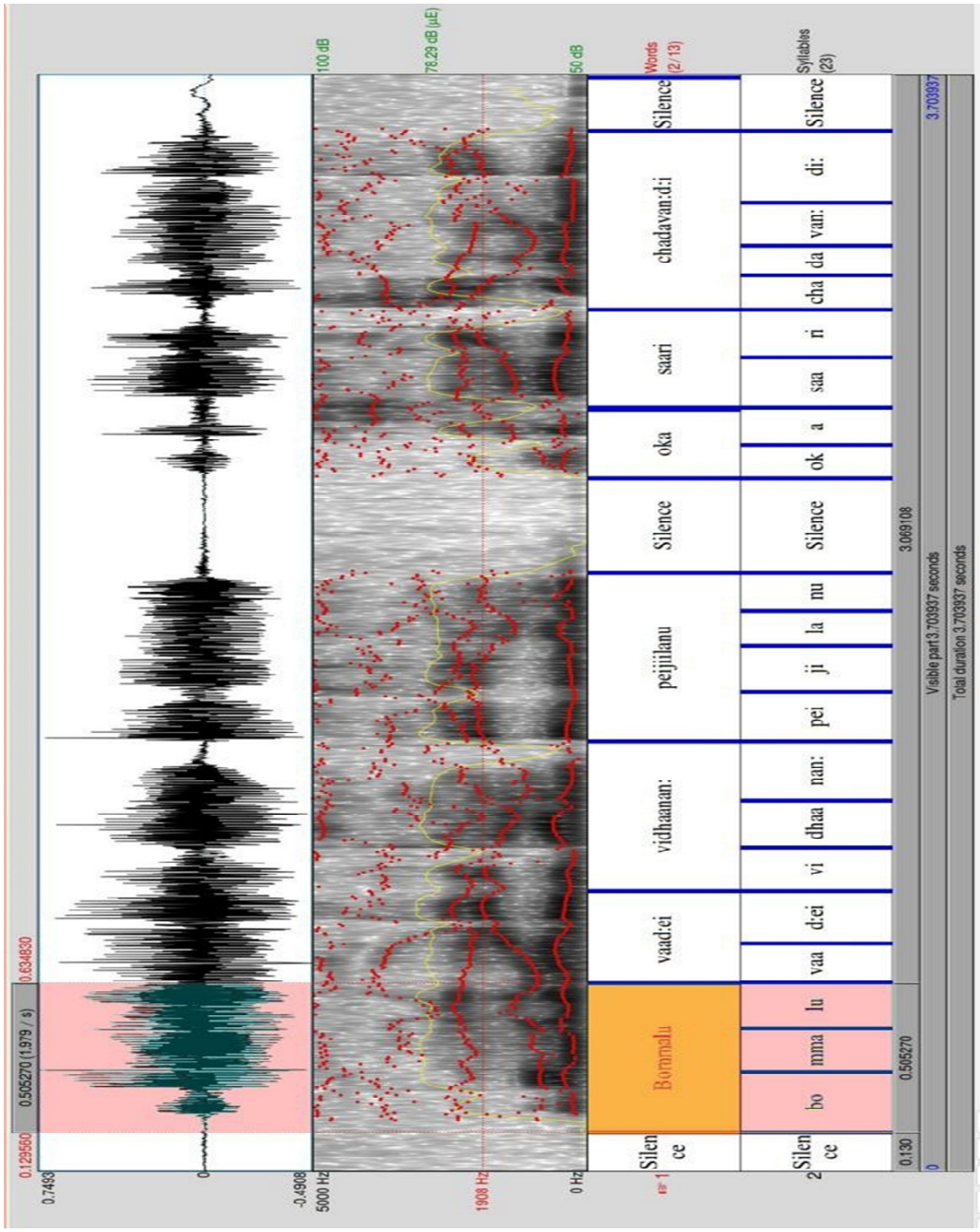

Figure 3.1: a) Recorded utterance of the sentence / bommalu vaad:ei vidhaanan: peijiilanu oka saari chadavan:d:i /, b) together with the associated pitch period contour, intensity contour, wide-band spectrogram and c) & d) transcription labelling units of words and syllables level

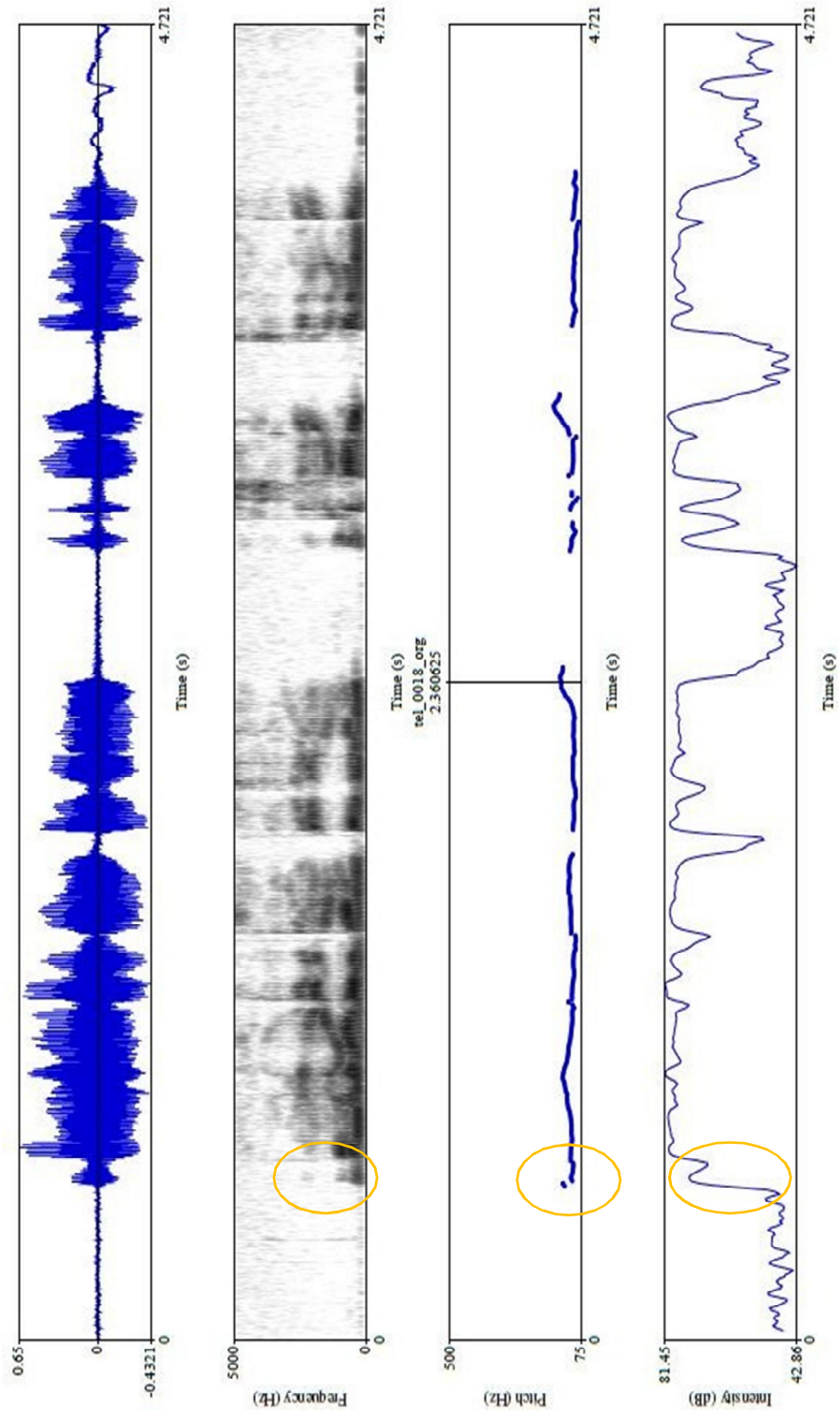

Figure 3.2: a) Recorded speech signal, b) spectrogram, c) pitch period contour and d) intensity contour of sentence

/bommalu vaad:ei vidhaanan: peijiilanu oka saari chadavan:d:i/

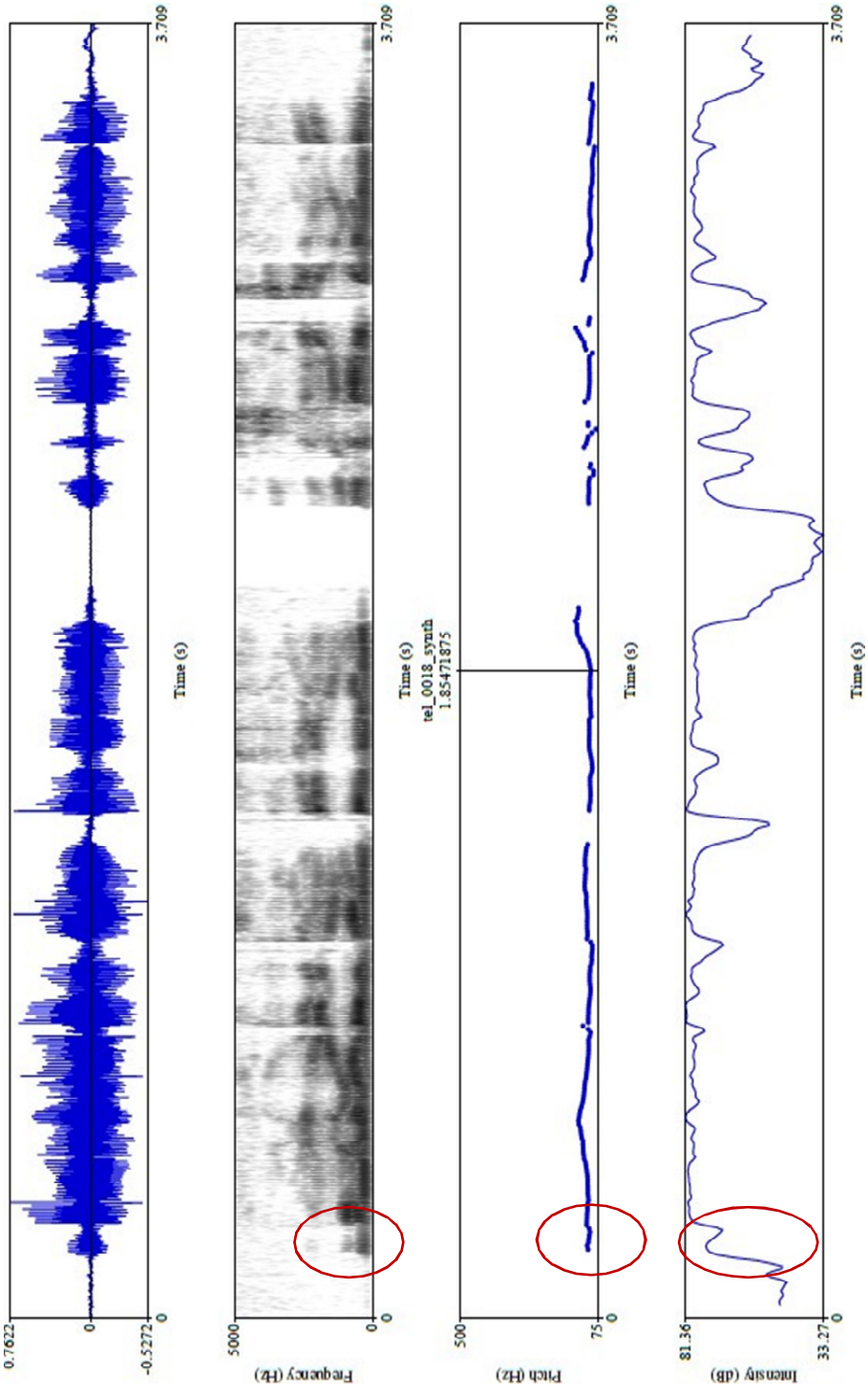

Figure 3.3: a) synthesized speech signal, b) spectrogram, c) pitch period contour and d) intensity contour of sentence

/bommalu vaad:ei vidhaanan: peijiilanu oka saari chadavan:d:i/

Figures 3.2 and 3.3 depicts the pitch period contour, intensity contour and wideband spectrogram of recorded speech and synthesized speech, respectively. The variation in synthesized speech pitch contour at the syllable boundary could be clearly visualized in the area inside the circle. The pitch period contour in real speech, on the other hand, is smoothly varying. This rapid change in pitch at the syllable boundary could be owing to a lack of appropriate target units in the TTS system's inventory. As a result, the synthesized speech has a perceived discontinuity that affects quality.

When observing the intensity contours and spectrograms in Figures 3.2 and 3.3, the synthetic sound shows an abrupt shift in intensity at the syllable boundary. In natural speech, however, there is never such great range in intensity within a word. While listening to the synthesized voice, this intensity change is easily distinguishable.

These illustrations can be used to make the following observations: In synthetic speech, there is a discontinuity around syllable boundaries. Natural speech's smooth pitch and intensity contours suggest that a limited number of frames can be used to obtain continuity information around syllable boundaries. This can be accomplished by prosody alteration. This paper presents methods for changing the pitch and intensity (energy equivalent) of synthetic voice samples around the syllable boundary

3.5. Proposed Prosodic Modeling

Natural speech signals have less variation of F0 and intensity in between the syllable boundary's left and right frames [37]. With this advantage, we adopt a prosody adjustment process that comparing the parameters of surrounding syllables with join syllable in this work. This given us an advantage to create USS system with small footprint when there aren't enough syllables with differing pitch and amplitude in the database. This work performs and presents the adjustment of syllables prosody prior to concatenation in order to match neighboring syllables F0 and intensity. As a result, no prior training is required to accomplish the prosody alteration. The F0 and intensity of the current syllable last frames and next syllable first frames

are used to determine the modification factor. The synthesized speech perceived discontinuity may be minimized as a result of the reduced discontinuity in pitch and amplitude at the join point.

To illustrate the efficacy of the strategy presented, we used FESTIVAL framework to create USS-based TTS for Telugu (a regional Indian dialect) employing the syllable as the basic unit. The work presented in this chapter was carried out using IIIT Hyderabad Indic Telugu text corpus [101]. The Festvox labeler is used to label the database at the phone level. A dynamic time warping approach was used by this labeler. The label boundaries were not appropriate since accurate duration knowledge was not provided for Indian language phones. Emulabel (www.festvox.org/ emulabel) was used to manually adjust these label boundaries.

3.5.1. Syllables Selection for Modification

We acquire insight of syllables selected from the database based on their total weighted cost function from the FESTIVAL framework by giving the text which is to be synthesized. The synthetic speech is created by concatenating these units. Before being concatenation, these units are imposed to pitch period adjustment and intensity adjustment respectively in this work. However, only some units which are selected from the inventory may be suitable for prosody alteration but not all. The syllables that begin with stop sounds have some stillness at the start, which is not suitable for change. The plosives' behavior is influenced by the silence preceding them. As a result, the syllable transition region is selected for modification, where the present syllable finishes with a voiced sound and the next syllable begins with voiced sounds not silence. An energy- based voice activity detection (VAD) technique is used to accomplish this. Only the region where the last frame of the current syllable and the first frame of the next syllable are recognized as voiced samples is picked for further modification.

3.5.2. Proposed prosody modeling method

Proposed prosody modeling method's procedure is as follows:

step 1. Make the given text into syllables and fine the target specifications.

step 2. Calculate the cost function by finding the weighted target cost and concatenation cost.

step 3. Choose the syllables whose cost is minimum. Assume that there are P selected syllables and that i = 1 at the beginning.

step 4. Perform the following steps for each of the P syllables:

a. Join i^{th} and $(i+1)^{th}$ syllable. Now find whether the final frame of the i^{th} syllable and the starting frame of the $(i+1)^{th}$ syllable are voiced or unvoiced by using "Voice Activity detection" (VAD).

b. For unvoiced frames, proceed to step 4. For voiced frames, the boundary region in between the i^{th} and $(i+1)^{th}$ syllables must be subjected to prosodic modification, and below steps must be followed.

c. Determine the epochs of the ith and $(i+1)^{th}$ syllables as per [102]. Now we need to modify speech samples few frames on either side of the syllable boundary. For example the syllable boundary will be at the m^{th} sample. Speech samples are modified from the $(m-n)^{th}$ to the $(m+n)^{th}$ epoch interval. We've used window size "n" in terms of epoch intervals as 3 in given example.

d. Determine the smoothed epoch period that fluctuates linearly between (m-n) and (m+n) epoch intervals. Now deduce the new epoch locations based on the modification.

e. Transfer the voice samples from each actual epoch interval to the reformed epoch interval that corresponds to it. Out of these voice samples, last 10% of voice samples are resampled in order to fit inside the modified epoch interval.

f. Next, calculate the intensity modification factor by calculating every epoch interval average energy. The scaling factor need to use to scale the samples of each epoch period correspondingly.

g. Now copy the modified epoch interval from $(m-n)^{th}$ to $(m+n)^{th}$ interval, to obtain the speech with pitch period modifications and intensity modifications.

step 5. Increase i and return to step 4.

3.5.3. Implementation

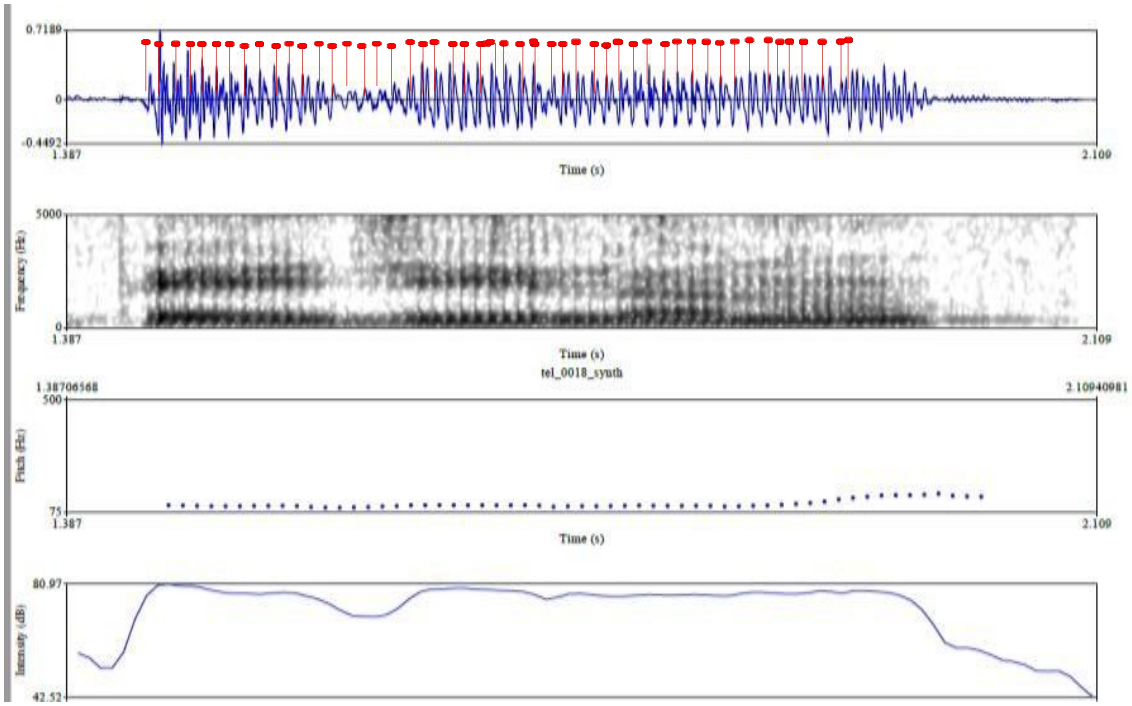

Figure 3.4: a) voice sample and appropriate epoch intervals, b) spectrogram, c) pitch contour and d) intensity contour of modified synthesized speech signal of the word / peijiilanu /.

The speech samples at the syllable boundary of the word “peijiilanu” from the same synthesized sentence as shown in Figure 3.3 are used to demonstrate the pitch period adjustment technique. Figure 3.4 shows the voice samples, appropriate epoch positions, pitch contour, and wide-band spectrogram. At the syllable boundary, both the spectrogram and the pitch contour indicate a discontinuity.

We developed a smoothly fluctuating pitch contour based on the proposed methodology, because a steady pitch contour makes speech sound artificial, and sudden changes in pitch can be mistaken for a shift in the speaker. To smooth the pitch contour at the syllable border, three pitch periods (epoch intervals) are used as the window size at the end of the current syllable and the start of the next one.. Which results in samples representing a total of 6 pitch periods around the boundary of syllable will be altered. The period of the pitch is calculated using the

zero frequency filtering method to acquire knowledge epoch positions. Based on the duration of the first and last epoch intervals, a smoothed pitch contour is generated in either ascending or descending order. Consider the six pitch periods M_1; M_2; M_3; M_4; M_5; M_6 where $M_1 + M_2 + M_3 + M_4 + M_5 + M_6 = T$ and $M_1 > M_6$.

The modified pitch periods P_1; P_2; P_3; P_4; P_5; P_6, for example, are obtained as $P_1 > P_2 > P_3 > P_4 > P_5 > P_6$ and $P_1 + P_2 + P_3 + P_4 + P_5 + P_6 = T$ where $P_1 = M_1$ and $P_6 = M_6$. We can get a smooth fluctuating pitch contour with the same duration by modifying the pitch periods. The new epoch positions are determined based on the new pitch period. The pre-modified speech's group of speech samples corresponding to each particular epoch interval is resampled and fitted into the changed epoch interval. Only the last 10% of the samples belonging to a given epoch interval are resampled, with the resampling factor being Pi 0:1_Mi where $1 < i < 6$. For each epoch interval, the process repeats again. These six-pitch-period-modified samples are copied to the matching index of pre-modified synthesized speech.

We used the same synthetic word seen in Figure 3.4 to demonstrate intensity modification, at the syllable boundary, where there is an abrupt change in intensity. To minimize it, the amplitude of the adjoining frames of the syllable border is scaled in a pitch synchronized manner. In speech samples, three pitch periods to the left (P_1; P_2; P_3) and three to the right (P_4; P_5; P_6,) of the syllable boundary, a total of six pitch periods (P_1 to P_6) are adjusted. First pitch period average energy to the last pitch period average energy is changed using a smoothly varying intensity contour (I_1; I_2; I_3; I_4; I_5; I_6,). The modification factor (M_1; ::: ;M_6) is calculated from the modified intensity contour as follows :

$$M_i = \frac{I_i}{E_i} \tag{2}$$

Where, I_i = Intensity contour and E_i = average energy of pitch period " i " respectively and i varies from 1 to 6. Factor M_i is used to scale the samples corresponding to the ith pitch period. The corrected and synthesized speech intensity contours are shown in figure 3.5. Figure 3.5 (a) and (b) indicate a significant reduction in discontinuity in corrected synthesized speech when compared to synthesized speech intensity contour.

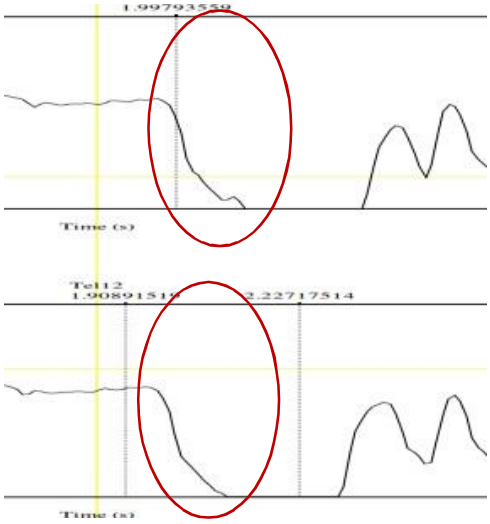

Figure 3.5: a) Intensity contour of synthesized speech signal, b) Modified intensity contour of synthesized speech signal of the word / peijiilanu /.

3.6. Experiments and Results

The effectiveness of the constructed TTS systems is demonstrated as follows. A total of six unit selection based TTS systems are developed in the festival framework, employing 5 hours, 2 hours and 1 hour of telugu dialect Indic database in order to evaluate the proposed technique performance. 1st and 4th TTS uses 5 hours of speech data, 2nd and 5th TTS uses 2 hours of speech data, while 3rd and 6th TTS uses only 1 hour of speech database.

3.6.1. Subjective Evaluation

TTS systems are used to collect and synthesize sentences from many domains. The sentences are between 10 and 15 words in length to give the listener, a degree of perceived quality. Subjective evaluation is performed using synthesized speech files generated from developed six TTS systems, where evaluation criterion is the Degraded Mean Opinion Score (DMOS) [1]. The tests were carried out in a quiet environment with headphones provided. Synthetic voice files and its recorded original speech samples of same speaker were played alongside at random. Twenty listeners are taken part in the tests. These listeners are not involved

in the development of any TTS voice. On a 5-point scale, listeners were asked to rate the system on the basis of audible discontinuities for naturalness, intelligibility, and overall quality. To prevent prejudice towards any method, every sentence has been coded. To eliminate the listener's prejudice, the scores are normalized against the scores given for the natural speech files given by each listener. Each method's DMOS is determined using the ratings from synthesised voice files. Table 3.1 shows the DMOS scores for each of the mentioned scenario above.

Table 3.1 Subjective evaluation scores of constructed USS based TTS systems

Method	**Speech data**	Subjective scores		
		Naturalness	***Intelligibility***	***Quality***
1st TTS	***5 hours***	3.5	3.78	**3.89**
2nd TTS	***2 hours***	2.9	2.99	**2.91**
3rd TTS	***1 hour***	2.49	2.58	**2.43**
4th TTS	***5 hours***	**4.02**	**4.19**	**4.32**
5th TTS	***2 hours***	**3.04**	**3.16**	**3.18**
6th TTS	***1 hour***	**2.88**	**2.92**	**2.9**

3.6.2. Comparison of proposed and conventional USS systems

The first three TTS systems use basic prosody modification as in festival with speech data of 5 hours, 2 hours and 1 hour respectively. The remaining three systems use the proposed prosody modification of speech data of 5 hours, 2 hours and 1 hour respectively; all six TTS systems employ the same modules during the training process.

Table 3.1 demonstrates that when 1st TTS and 4th TTS which are constructed with 5 hours of speech data are compared, the proposed method produces a significant improvement in the DMOS score. There is a slight improvement in 5th TTS over 2nd TTS. Both these uses 2

hours of speech data and there is much improvement in 6th TTS than 3rd TTS which uses only one hour of speech data. The intensity and pitch of synthesized speech syllables will vary and are smoothen using the proposed modifications in 4th, 5th and 6th TTS systems. All these proposed systems showed a better quality than the original systems and depends on selection of dataset. Care to be taken in selection of dataset. The dataset must cover all the possible context data.

3.7. Summary

We made an attempt to adjust the pitch and intensity contours of telugu synthesized speech around each syllable boundary in this work. Attempt was successful in developing a small foot print USS based TTS system with significant subjective scores. To some extent, the proposed prosody modification gave better perceived discontinuity. In the case of Large discontinuities, it cannot be totally avoided. To generate natural syllable boundary, future research could focus on spectral properties modification rather than pitch and intensity.

Chapter - 4

Neural Networks for Telugu TTS System

Concatenative speech synthesis requires huge recorded dataset to cover all the possible instances though its showed good quality as natural as specific human voice where as the quality of SPSS's traditional methods is impressive, but their level of naturalness still falls short of that of natural speech.. Due to inefficient modeling of complex context dependencies, the quality is degraded. Hence there is a need to make network architecture and training techniques efficiently to improve the naturalness of generated speech.

4.1. Introduction

The core of statistical parametric speech synthesis, which has gained in prominence over the past 20 years, are hidden Markov models [11]. It differs from concatenative speech synthesis, in that it can change voice characteristics [74], [75], has a smaller footprint [76], and is more resilient [77]. One of its key shortcomings is the quality of the synthesised speech, though. According to [10], there are three key problems that reduce the quality of synthetic

speech: vocoding, acoustic model correctness, and over-smoothing. HMM-based SPSS showed successful in good voice quality with some busyness.

The primary elements for statistical parametric synthesis of speech are MCEPs, F0, and duration. A set of guidelines for parameter modifications were applied, along with careful parameter selection. The popularity of statistical parametric synthesis has increased recently as a result of its scalability and scale. Statistical parametric synthesis employs machine learning techniques to learn the parameters from the features extracted from the voice data [47]. Using hidden Markov models and CART, respectively, the statistical parametric synthesis engines HTS and CLUSTERGEN learn the parameters from the speech data. In the SPS framework, harmonic noise models features, line spectral pairs, and cepstral coefficients are frequently used to describe spectral features. Strengths of the fundamental frequency and voicing are examples of excitation characteristics. Speech signal is produced using source-filter models using spectral and excitation properties.

4.1.1. Acoustic Modeling

One of the key techniques used in the back-end part is statistical parametric speech synthesis [10]. In this method, an acoustic model is used to depict how linguistic and acoustic information relate to one another, and a vocoder is used to create a speech waveform given acoustic features. Compared to concatenative speech synthesis, this approach offers a number of benefits, including lesser dataset [103], [104] and the flexibility to alter its voice characteristics [27], [75], [76]. The naturalness of the speech produced by SPSS, however, is not as good as that of the best samples produced by concatenative speech synthesisers. [10], [11] identified three key elements that potentially diminish naturalness: 1) the effectiveness of the vocoder, 2) the correctness of the acoustic model, and 3) the impact of oversmoothing.

Despite multiple attempts to develop a more realistic acoustic model for SPSS [26], [49], [74], [106], [107], [108], [109], [110], the HMM [105] is still the most used one. The term "HMM-based speech synthesis" refers to statistical parametric speech synthesis using HMMs [74]. “Trended HMMs” [106], “buried Markov models” [107], “trajectory HMMs” [108],

“polynomial segment models” [109], ‘linear dynamical models” (LDMs) [110], “autoregressive HMMs” [111], “product of experts (PoEs)” [112], “Gaussian process regression” [113], and “hidden trajectory model” [114] are just a few of the attempts to replace HMM with an alternative acoustic model. Some of them can generate dynamic feature constraints-free, smoothly evolving acoustic features.

Top-down decision trees are used in the acoustic models discussed above to reflect context dependency. Despite the fact that there are numerous varieties of acoustic models, they are actually best understood as sizable regression trees that translate a linguistic feature vector to the statistics (such as mean and variance) of acoustic features [115]. Enhancing the efficiency of decision trees itself is crucial because they serve as the primary regression model in various SPSS techniques. Additionally, there have been initiatives to enhance decision trees by using cross validation, outlier detection, boosting, and tree intersection [116].

Five different kinds of ANN-based acoustic models were developed in 2013 [12], [117], all of which were motivated by the advancements in machine learning and automatic speech recognition. A method that Zen et al. proposed [12] represents a mapping function from linguistic data to auditory features using a multi-layer artificial neural network. It opened up a new field of study and had a big impact on the research community. The popularity of this strategy has already skyrocketed despite its recentness, as seen in [12], [31], [118]. Here, language information (input) are mapped to auditory features (output) using an artificial neural network that has been trained [108]. The ability to synthesise speech with a natural sound has been demonstrated by ANN-based acoustic models [12], which provide an effective and distributed representation of complicated connections between linguistic and acoustic variables. It was later enhanced to predict the entire conditional multimodal probability distribution of auditory features rather than just conditional single predicted values.

Recurrent neural networks, particularly LSTM - RNNs, are used to simulate the speech's sequential nature, which has correlations between subsequent frames. This is a substantial extension. Modern LSTM-RNN-based SPSS greatly outperformed HMM and feed-forward deep neural network-based techniques in terms of subjective mean opinion scores. We tends to investigate these improvements to apply to the Telugu TTS system in this thesis.

The concept of employing neural networks for voice synthesis has been around for a while; in the 1990s, publications about using neural networks for speech synthesis first appeared. The availability of different algorithms used in HMM-based speech synthesis, such as high-quality vocoders (for example, STRAIGHT, Vocaine) and over-smoothing compensation techniques (for example, global variance, modulation spectrum), is one of the main differences between the current generation of neural network-based TTS systems and earlier generations.

4.1.2. Contextual Representation

Text features in a typical DNN-based SPSS system are made up of (a) categorical (e.g., pentaphone identities, identity of the vowel in the current syllable, etc.), (b) numerical (e.g., number of syllables in a word, number of words in a phrase, etc.), and (c) durational (e.g., frame index, duration of phone, etc.) features. Typically, 1-hot-k encoded representations are used for the category features. However, using continuous valued representations, as suggested in [66], would be a more appropriate kind of representation. Another strategy to be taken into consideration is learning embeddings for the phones, words, and utterances as in [67].

4.2. Neural Network for Speech Synthesis

The decision tree based clustered context-dependent acoustic models, as was discussed in the preceding section, can be thought of as large regression trees that translate language information to statistics of acoustic features. A different approach was put forth by Zen et al.

[12] that is based on a deep architecture [119] and substitutes a multilayer artificial neural network for the regression tree.

The following comparisons are made between the decision tree's and the artificial neural network's characteristics:

- XOR, the d-bit parity function, and multiplex issues are examples of complex functions of input features that decision trees are poor at expressing. Decision trees will be too big to represent such scenarios. On the other hand, an artificial neural network can compactly describe them [119].

- Decision trees use a different set of parameters for each region connected with a terminal node (also known as a local representation) and rely on partitioning the input space. This reduces the amount of data per location and makes generalisation difficult. The distributed representation provided by artificial neural networks is more effective than the local one in modelling data with componential structure and enables better generalisation because weights are trained using all training data. Additionally, they provide the option to incorporate high-dimensional, diverse features as inputs.

- The process by which information is transformed from the linguistic to the acoustic levels in the human speech production system is thought to include multiple layers [120]. A more accurate representation is provided by artificial neural networks with their layered hierarchical structure than by models with shallow designs (like regression trees).

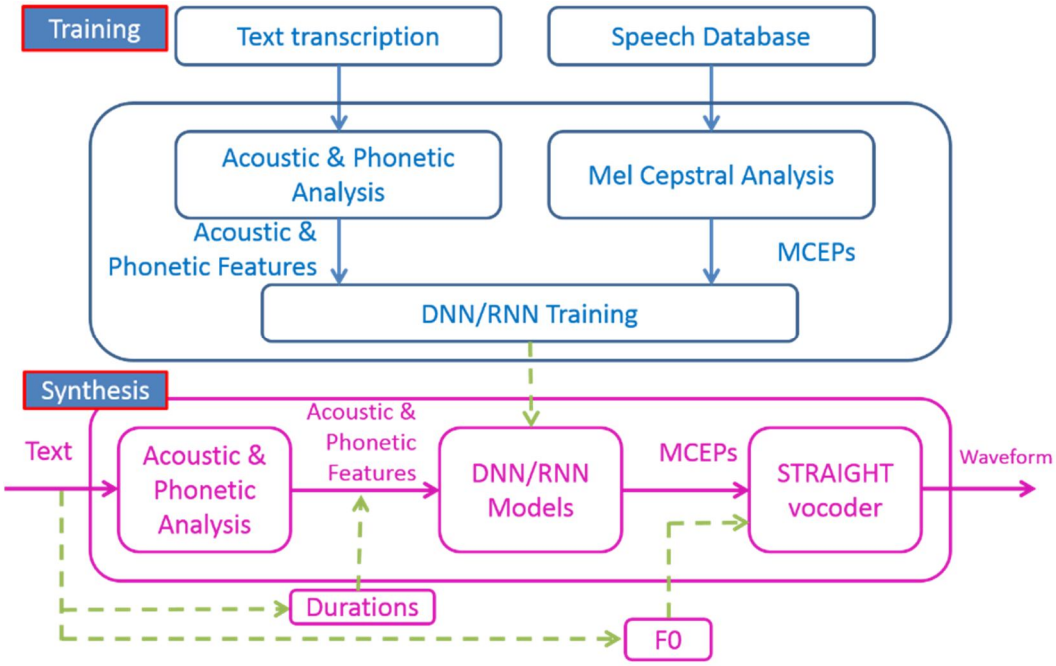

Figure 4.1. Block diagram of Neural Networks based SPSS system

Figure 4.1 illustrates the entire NN based SPSS system. A "text-to-acoustic and phonetic analysis subsystem", as well as one or more NN models that are used to anticipate MCEPs, are also included in the text-to-speech system.

The text provided is transformed into phonetic and acoustic notation during synthesis, and MCEPs are forecast using the most recent models. The test phrase database is used to obtain the durations of each phoneme and *F0* the for this example. STRAIGHT vocoder uses the original *F0* and predicted MCEPs to produce speech.

Interconnected processing nodes, each of which is a model of an artificial neuron, serve as the basic units of neural network models. Each connection between nodes has a weight. NN models with different topologies can recognise patterns in a variety of ways. For instance, although a feedback network can perform pattern association, a feedforward neural network can conduct pattern mapping. In addition to its aptitude for capturing intricate, nonlinear mapping, NN models are recognized for their capacity for generalisation. Speech synthesis requires a conversion from text (linguistic space) to speech (acoustic space). Using the pattern-

mapped capabilities of NN models, we perform a difficult and nonlinear translation of linguistic space to aural space to produce synthetic speech.

4.2.1. Deep Neural Networks

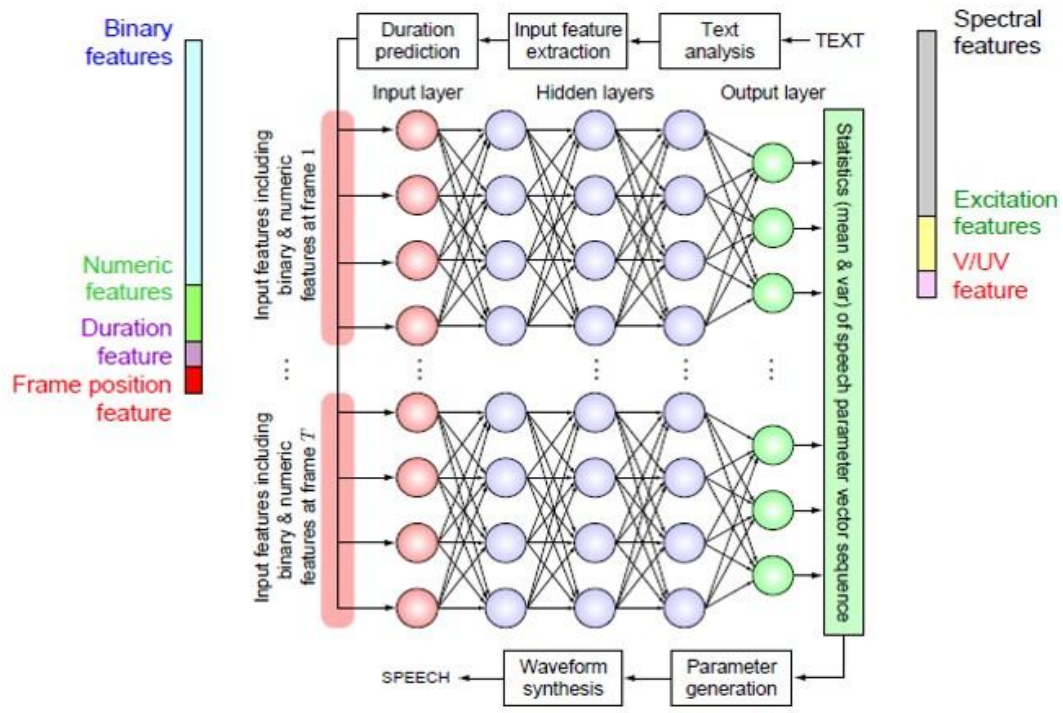

Figure 4.2. Framework of DNN-based parametric speech synthesis reproduced from [21]

The feed-forward, multi-layer artificial DNN-based architecture of feed-forward acoustic modelling for SPSS [12] is shown in Figure 4.2. An output acoustic feature vector is created right away from an input linguistic feature vector. Instead of using phoneme-level phonemes, this method makes advantage of input linguistic properties at the frame level. They include binary responses to questions about linguistic contexts, such as "is-current phoneme-vowel?," as well as phoneme-level numerical values (such as the number of words in the phrase and the length of the current phoneme), frame-level features (such as the relative position of the frame at the moment in the current phoneme), and frame-level features.

The vocal tract and source characteristics, as well as their dynamic features, are included in the target acoustic feature vector. Backpropagation is used to train the DNN weights utilising

pairs of input and target characteristics at each frame. A 3-layer DNN – based acoustic model and its dependency graph is shown in figure 4.3. h_{ij} denotes activation at i^{th} layer at j^{th} frame. The figure shows that there is no dependency between neighbouring frames. The absence of dependency causes a discontinuity between neighbouring frames. To solve this issue, Bulyko et al. [12] added dynamic acoustic characteristics to outputs before producing the final smoothly varying static acoustic features using the speech parameter generation technique.

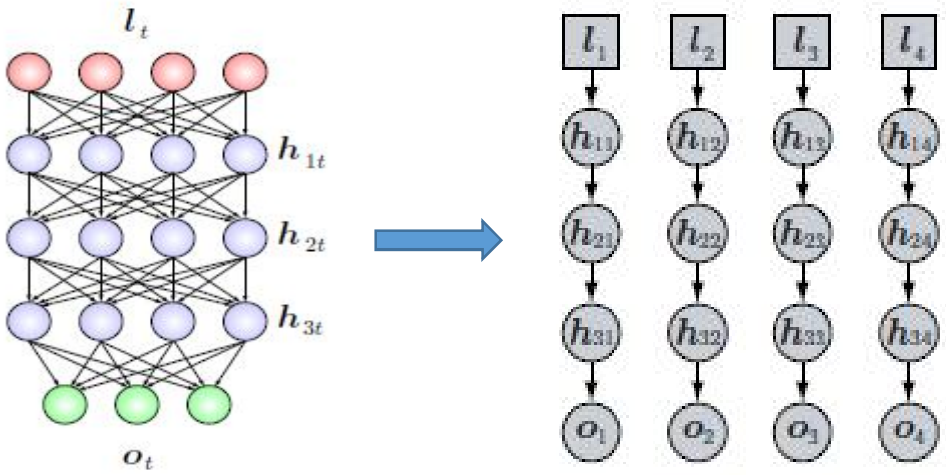

Figure 4.3. A 3-layer DNN – based acoustic model and its dependency graph

Deep Neural Networks has the following characteristics;

- **Inconsistent;** training does not use dynamic characteristics; they are only used in the synthesis stages.
- **No clustering** is used, and network weights rather than trees are used to map language feature vectors to auditory feature vectors.
- **Effective training**; Stochastic Gradient Descent (SGD) and back propagation are also suitable training methods. A collection of HMMs provides phoneme- or state-level boundaries, much as the LDM and trajectory HMM. During DNN training, these alignments are corrected..
- **High latency**

- **Slow synthesis;** complete utterance synthesis is computationally far more costly than HMM. The HMM searches through decision trees to discover statistics of acoustic features, but the DNN needs forward propagation, which includes matrix multiplication operations. Additionally, this procedure must run at every frame, unlike HMM, which is required at every state.
- Better **naturalness**; subjective score was greater to the standard HMM. [21].
- Decision trees are easier to understand than weights in a DNN. However, it would be beneficial to visualise the data and weights.

The feed-forward DNN-based acoustic model was further enhanced to predict the entire conditional multimodal distribution of acoustic features, as opposed to just conditionally single projected values using a mixture density output layer [29].

4.2.2. Recurrent Neural Networks

The feed-forward DNN-based acoustic modelling has the drawback of ignoring speech's sequential nature. The DNN-based technique makes the assumption that each frame is independent, even though there are undoubtedly relationships between consecutive frames in voice data. The sequential character of voice data should be incorporated into the acoustic model itself. Recurrent neural networks offer a beautiful method to represent sequential data that resembles speech and incorporates relationships between adjacent frames. It can forecast output features at each frame using all the available input features. While LSTM-RNN [121], which can detect long-term dependencies, was recently employed for audio modelling for SPSS [30, 31, 122], simple RNNs were still used for voice synthesis.

To describe temporal sequences and their long-term dependencies, the LSTM-RNN architecture was created [121]. It has unique building blocks called memory blocks. Figure 4.5 shows a simple recurrent unit on the left and a memory block in an LSTM-RNN on the right. A memory block has three special multiplicative units called gates that regulate the information flow, as well as a memory cell with selfconnections that stores the network's temporal state. These gates perform the role of differentiable random access memory (RAM); "input," "output," and "forget" gates protect accessing memory cells. Because of its architecture, LSTM-

RNNs can store data in memory cells for a lot longer than simple RNNs. These RNN units are briefly reviewed in the following section.

4.2.2.1. LSTM (Long Short-Term Memory)

Our use of LSTM is quite conventional, with forget gate and peephole connections included. Below are the forward pass equations for LSTM implementation, which is based on [123]:

$$z_t = g(W_z x_t + W_z h_{t-1} + b_z) \quad (3)$$

$$i_t = \sigma(W_i x_t + R_i h_{t-1} + p_i \odot c_{t-1} + b_i) \quad (4)$$

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + p_f \odot c_{t-1} + b_f) \quad (5)$$

$$c_t = i_t \odot z_t + f_i \odot c_{t-1} \quad (6)$$

$$o_t = \sigma(W_o x_t + R_o h_{t-1} + p_o \odot c_t + b_o) \quad (7)$$

$$h_t = o_t \odot h(c_t) \quad (8)$$

Where **"σ"** stands for the sigmoid activation function and **"g"** and **"h"** are tanh activation functions. At the input gate, unit-input, forget gate, and output gate, **Wi, Wz, Wf,** and **Wo** are the weights from the input, while **Ri, Rz, Rf,** and **Ro** are the weights from the previous state respectively. The peep-hole connections are $\mathbf{p_i}$, $\mathbf{p_f}$ and $\mathbf{p_o}$, while Θ stands for element-wise multiplication. The input and hidden state at time **t** are $\mathbf{x_t}$, $\mathbf{h_t}$.

4.2.2.2. GRU (Gated Recurrent Unit)

The GRU model was proposed in [59], and its forward propagation equations are shown below:

$$r_t = \sigma(W_r x_t + R_r h_{t-1} + b_r)) \quad (9)$$

$$z_t = g(W_z x_t + R_z h_{t-1} + b_z) \quad (10)$$

$$\hat{h}_t = g(W_c x_t + r_o \odot (R_c h_{t-1}) + b_c) \quad (11)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_{t-1} \quad (12)$$

GRU has shown in figure 4.4.

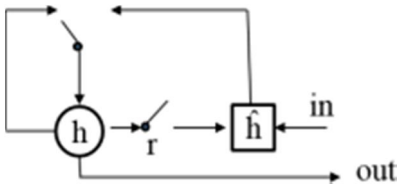

Figure 4.4. Gated Recurrent Unit (GRU).

4.2.2.3. SLSTM (Simple LSTM)

By removing the input, output, and peephole connections from the LSTM model, a new gated RNN known as SLSTM [56] was produced. The authors provide proof that SLSTM is faster and simpler than any previous gated recurrent designs while maintaining high-quality synthetic speech. The SLSTM model's forward propagation equations are listed below:

$$z_t = g(W_z x_t + R_z h_{t-1} + b_z) \quad (13)$$

$$f_t = \sigma(W_f x_t + R_f h_{t-1} + p_f \odot c_{t-1} + b_f) \quad (14)$$

$$c_t = i_t \odot z_t + f_t \odot c_{t-1} \quad (15)$$

$$h_t = h(c_t) \quad (16)$$

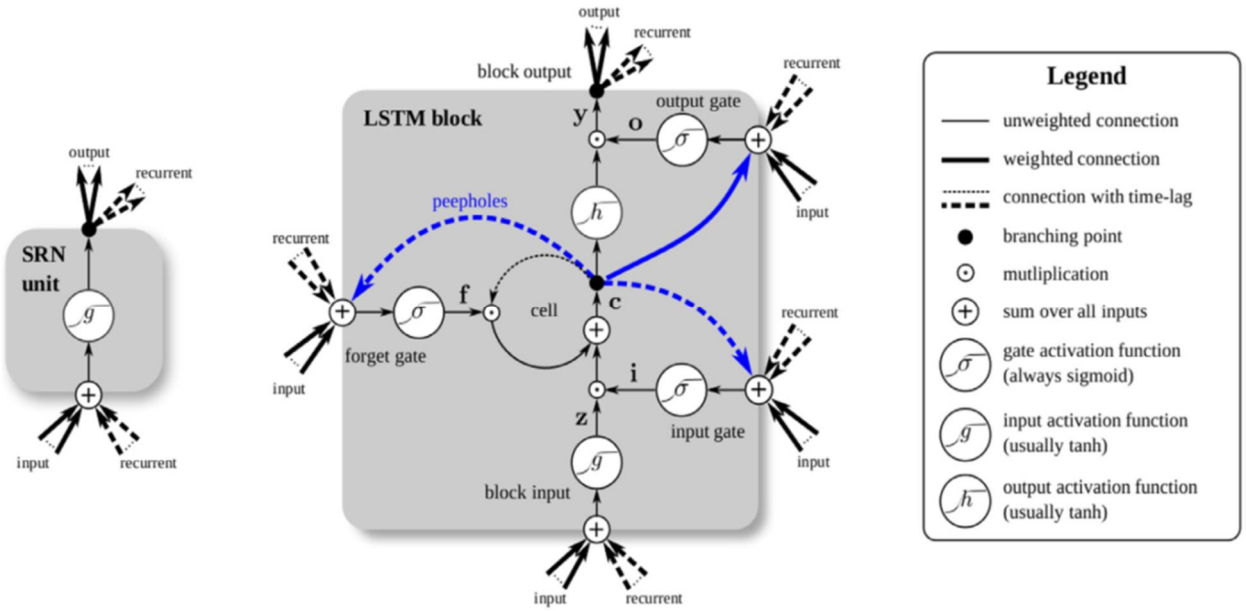

Figure 4.5. Simple recurrent units (SRN on left) and long short-term memory units (LSTM on right) are the two most popular forms of RNN units (reproduced from [123]).

An RNN's hidden layers often have feedback loops that are unidirectional, meaning that information only flows forward as soon as the input is transacted with left to right. Schuster proposed the bidirectional RNN architecture to incorporate both past and future inputs for prediction. It has feedback loops that allow information to move both forward and backward. Using inputs from the complete sequence, the network can predict outputs thanks to this architecture. Additionally proposed were the bi-directional LSTM-RNNs. Deep bi-directional LSTM-RNNs, which have access to input features at both current and future frames, were used by to increase the naturalness of their auditory models for SPSS [30]. The unidirectional LSTM-RNN, which can access input features up to the current frame, was used by [31] to create low-latency voice synthesis.

A 3-layer Unidirectional LSTM-RNN dependency graph with a recurrent output layer is shown in Figure 4.6. The graphic shows that it contains dependencies between neighbouring frames at both the hidden layer level and the output layer level. The naturalness was improved by the addition of a recurrent output layer, which produced acoustic qualities that varied more gradually.

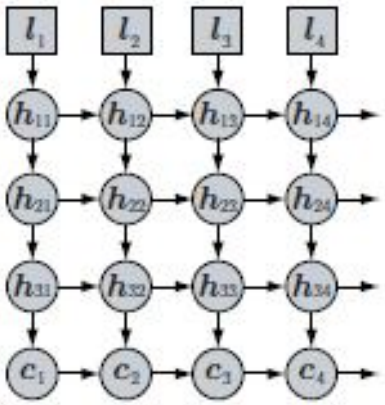

Figure 4.6. 3-layer unidirectional LSTM-RNN Dependency graph with recurrent output layer.

The LSTM-RNN has the following traits:

Consistent; dynamic features are not used at both training and synthesis stages

No clustering is used, and network weights rather than trees are integrated in the mapping, from a linguistic feature vector sequence to an auditory feature vector sequence.

Effective training; is trainable using SGD and backpropagation through time(BPTT). While the LSTM-RNN is being trained, the HMM's phoneme-level alignments are fixed.

Low latency; an LSTM-RNN-predicted acoustic feature sequence evolves smoothly [38], [44]. Frame-synchronous generation is offered by unidirectional LSTM-RNNs without the use of a look-ahead.

Complete utterance **synthesis is slower** than HMM, but less expensive computationally than DNN because the DNN's network depth can be smaller and the smoothing phase is not required**.**

Better naturalness; superior subjective rating to DNN [31].

Compared to decision trees and a DNN, weights in an LSTM-RNN are considerably more difficult to interpret. The dynamic nature of the network makes it more difficult to visualise the data and network weights.

4.3. Description of Speech Corpus

BDL & SLT US voice from CMU ARCTIC dataset [103] was used for English language. Out of a total of 1132 utterances, 1075 were used for training and the final 56 were used for testing and validation.

IIIT Hyderabad [101], and INDIC voices from the INDIC dataset were used for Telugu. Out of a total of 1000 utterances, 950 were used for training and the final 50 were used for testing and validation. A different data set from a portion of the “Blizzard Challenge 2015 database” was used for Telugu. “The Blizzard Challenge 2015 database” includes approximately 4 hours of speech data in Telugu, Hindi, and Tamil, as well as around 2 hours of speech data in Bengali, Marathi, and Malayalam, all of which were recorded by native professional speakers in high-quality studio settings. For our tests, we used a Telugu language dataset with 2355 utterances. The leaked speech recordings were 16 kHz samples. For acoustic feature extraction, the data were up-sampled to 48 kHz using the HTS-STRAIGHT demo versions 2 and 3. EHMM tool was used to perform alignments at the phone level [1] [44]. The feedforward network utilised in this work has a three layer architecture, or 359 L 250 N 235 L for English, and 529 L 250 N 235 L for Telugu. Along with that worked upto ten hidden layer architecture but shown more or less similar performance. Detailed experimentation given in following sections.

4.4. Experiments and Results

Figure 4.1 depicts the architecture model of NN-based SPSS systems. The next subsections provide a detailed discussion of the input/output features employed and the models developed.

4.4.1. Input Features

A careful representation is required at the input layer because we are working on a mapping from text input space to formant output space. In addition to the difficulty of this mapping, we anticipate that the model will offer modest variations in the formants and bandwidths for each frame.

For the purpose of training the NN model, the text's current, left and right phone articulatory, and syllable properties were extracted. There are also temporal features (the location of the frame within the current phone), current frame state information, the location of the current phone within the word, the location of the current word within the sentence, and these. Please be aware that state information is utilised into NN modelling to aid in differentiating the phone frame types.

4.4.2. Output Features

At the output layer, it is hoped that the network will forecast MCEPs. All of the spoken utterances' 26-dimensional band-aperiodicities, 50-dimensional Mel-general cepstral characteristics, and deltas and double-deltas were extracted with a 5 ms frame-shift. This feature extraction is based on the online HTS- STRAIGHT demo. F0 characteristics include binary voiced/unvoiced flags, F0, Δ F0, $\Delta\Delta$ F0, and VUV. The most effective threshold as determined by training data was used to set the voiced/unvoiced threshold during testing. With a 5ms frame shift, 235 coefficient vectors were predicted for each 10ms frame size. The back propagation learning approach was used to train the NN model during a 20 iteration period. The features of the input and output were adjusted for mean variance.

4.4.3. Models

As a starting point, we employ DNN with typical features. Early halting was employed to end the training for all networks. While no regularizer was employed for RNNs, For DNNs, a weight decay regularizer was applied with a decay weight (λ) = 0.00005.

4.4.3.1. Weight Initialization

For DNNs, we employed normalised initialization. A Gaussian distribution was used to select all of the non-recurring weight values. The Gaussian distribution was used to determine every parameter for LSTMs, GRUs, and SLSTMs, with the standard deviation set at 0.01. We used entire BPTT for training RNNs.

4.4.3.2. Hyper-Parameter Setting

We use "Adam" to automatically set the learning rate for each dimension when doing stochastic gradient descent. The hyper- parameters (α is the learning rate and $\beta 1$, $\beta 2$ are exponential decay rates) were chosen $\alpha = \{3e-3, 1e-3, 3e-4, 1e-4\}$, $\beta 1 = 0.9$ and $\beta 2 = 0.999$. Table 4.1 provides information on the models that were trained.

Table 4.1 Details of Hyper-Parameter Setting and Initialization of Models

Model (System)	**Activation Function**	**SGD Method**	**Initialization Method**
DNN	ReLU (R)	Adam	Normalized Initialization
LSTM / GRU / SLSTM	tanh (N)	Adam	Gaussian Initialization

4.4.4. Subjective Evaluation

Subjective listening tests called MUSHARA ("MUltiple Stimuli with Hidden Reference and Anchor") [61] were used to assess how natural speech sounded. To give the listener a sense of perceived quality, the sentences are between 10 and 15 words long. Advanced TTS systems' synthesised speech files are used for subjective assessment, with the Mean Opinion Score acting as the measuring criterion. The tests were conducted with provided headphones in a quiet

setting. The recorded original speech samples of the same speaker and the synthetic voice files (DNN, SLSTM, GRU, and LSTM) were played simultaneously at random. The exams involve twenty listeners in all. In the creation of any TTS voice, these listeners are not involved.

To ascertain which system (A or B) is more preferred for the experiment detailed in Section 4.5, we conducted a subjective preference test (ABX). The "No Preference" (X) option was available to participants if both systems were performing equally. And listed in table 4.4.

The Mean Opinion Scores of developed models are presented in table 4.3.

4.4.5. Objective Evaluation

“Mel-cepstral distortion for the spectrum, RMSE for F0, % error in frames for VUV, and Euclidean distortion for BAP features” are among the objective metrics. Global variance spectral augmentation was performed before the subjective listening tests but not when calculating MCD scores. MLPG smoothing, however, was utilised for all parameter tracks. The outcomes are shown in table 4.2.

4.5. Comparison of RNNs with DNN for SPSS

In this experiment, we use both objective and subjective measures to evaluate and contrast the performance of DNN alongside various RNN models. Five hidden layers, each with 500 ReLU units, were used to train all of the DNNs. We have tested the impact of breadth and depth on the acoustic modelling performance within the RNN models. With a depth of 1 and 2 hidden layers, namely, 250 and 500 hidden units were taken into consideration. It is obvious that for all models, increasing depth or breadth—or both—improves performance; however, it's vital to note that, in contrast to boosting depth, increasing the hidden state size from 250 to 500 causes a significant rise in parameters. Therefore, the size of the hidden state and the number of layers must be carefully chosen in order to have an appropriate amount of parameters in the model. Models with 1M parameters must be compared in order to compare the RNN models

below fairly. Table 4.2 lists the objective metrics for Telugu DNN and RNN-based SPSS systems.

Table 4.2 Objective Metrics of Telugu Language NN based SPSS Systems

System	Architecture	Parameters (in millions)	MCD (dB)	F0 (RMSE)	VUV (% error)	BAP (dB)
DNN (S1)	5 x 500R	1.3	4.21	32.13	6.22	28.20
SLSTM (S2)	250N	0.35	4.25	32.12	6.22	28.27
	500N	0.96	4.25	**31.98**	6.22	28.22
	2 x 250N	0.63	4.23	31.99	**6.20**	**28.20**
	2 x 500N	1.96	**4.21**	31.99	6.22	28.21
GRU (S3)	250N	0.5	4.24	33.52	6.23	28.42
	500N	1.38	**4.23**	**33.18**	**6.21**	**28.32**
	2 x 250N	0.88	4.24	32.29	6.22	28.36
LSTM (S4)	250N	0.65	4.23	33.13	**6.20**	28.28
	500N	1.83	**4.21**	**32.27**	6.21	**28.19**
	2 x 250N	1.18	4.22	33.22	6.23	28.33

The objective scores of RNN based SPSS systems gave more or less scores on par with DNN-based TTS System. The best scores are made bold in each model.

Table 4.3 Subjective Metrics of Telugu Language NN based SPSS Systems

System Model	Naturalness	Intelligibility	Quality
DNN (S1)	3.23	3.18	3.15
SLSTM (S2)	3.24	3.19	3.21

GRU (S3)	3.25	3.17	3.22
LSTM (S4)	**3.27**	**3.21**	**3.23**

The MOS scores of RNN based SPSS systems gave better than DNN model. 20 listeners were participated in testing the model and these were with zero knowledge people in developing a speech model. The listeners shown great interest in LSTM-RNN model. Systems S4 based on LSTM – RNN shown the naturalness improvement of 0.04 and 0.08 overall quality over the system S1 uses DNN as a synthesis network.

The results of the pairs ABX listening tests, which were used to ascertain subjective preference, are shown in table 4.4 below. The following system pairs were taken into consideration:

• DNN Vs GRU (2x250R)

• DNN Vs LSTM (2x250N)

• GRU (2x250R) Vs LSTM (2x250N)

Table 4.4 Subjective preference tests of Telugu Language NN based SPSS Systems

System	DNN	GRU	LSTM	No Preference
1	40.05	50.38	-	8.57
2	38.23	-	55.11	6.66
3	-	15.07	18.12	66.81

From Table 4.4, it is obvious that the listeners favour RNNs over DNNs greatly. It is evident from third row that the LSTM-RNNs and GRU are working equally well, which

explains why listeners typically select "No Preference" more frequently than 60% of the time. The RNN-based SPSS systems can be trained and tested quickly and efficiently

The proposed technique with the state-of-the-art results were validated with the available literature [78] [53]. We achieved better MCD, F0 and VUV but not Euclidian distortion for BAP for DNN, when validated with [78]. Whereas with [53], the results are vary for particular state-of-the-art techniques due to use of original F0 and Duration. There is slight reduction in MCD, F0 and VUV but there is abrupt increment in Euclidian distortion for BAP for DNN. This inturn motivated to go in depth with the RNNs, especially Elman RNNs.

4.6. Summary

In this chapter, we looked into the DNNs and RNNs for SPSS. Through objective and subjective evaluations, we proved that RNNs like SLSTMs, GRUs, and LSTMs can outperform DNNs in the Telugu language when the proper gradient clipping and weight initialization strategies are used. As a result, training and testing RNN-based SPSS systems may be done fast and effectively. Additionally, fewer parameters provide a storage advantage, which is very helpful for mobile apps. Given the advantages over the LSTM, the approach might make data even easier to port the SPSS system to mobile devices. Recently, an investigation using LSTMs for SPSS was been out. The LSTM-RNN is used as an acoustically guided feature in the DNN guided RNN-based SPSS system, which improves it. Because the recurrent unit in gated recurrent architectures contains more gates, increasing the computational steps also increases the number of parameters.

Chapter - 5

Deep Elman RNNs for Telugu TTS System

Deep learning (DNN, RNN) based systems outperformed a similar number of parameters but increased computational costs [78], [79]. However, recent advancements in software (e.g. [80]) and hardware (e.g. GPU) allow us to train a DNNs and RNNs using a good chunk of training data. Speech recognition and acoustic-articulatory inversion mapping are two machine learning applications where deep neural networks have outperformed traditional methods. Despite the fact that DNNs are strong models, but there are few issues with the network architecture and the training techniques that affect how well they perform in parametric voice synthesis. We draw attention to these problems that must be resolved in order to enhance the performance of “Neural Network based SPSS”. Elman RNNs are networks that lack sophisticated gating mechanisms yet have concealed state.

This chapter continues our investigation of Elman RNNs to improve various aspects of SPSS acoustic modelling and text-feature representation.

5.1. Introduction

DNNs are robust models, but their performance in parametric voice synthesis is limited by a few problems with network architecture and training methods. We call attention to three issues (Acoustic modeling, Contextual representation and Over- smoothing) that must be fixed to perform better for NN based SPSS. We show that ERNNs can perform on par with gated RNNs, despite the fact that more complex RNN architectures, such as LSTM and variants like GRUs and simplified LSTM, and gives computational effectiveness without compromising quality [53].

5.1.1. Acoustic Modeling

DNNs will only be useful for a small number of phones in the past and future when used as acoustic models, and they will also cause discontinuities in the projected parameters. Recurrent neural networks (RNNs) can use unconstrained and adaptive context to solve the first issue since they are able to remember the past. The latter issue results from the DNNs' training methodology. The framewise mapping of the textual and auditory representations is done independently for each frame. In contrast to the genuine speech parameter trajectories, this causes discontinuities in predicted parameters from frame to frame. By applying explicit dynamic constraints during post-processing (i.e., running the DNN predicted parameter trajectories through the maximum-likelihood parameter generation (MLPG) algorithm), the discontinuities can be smoothed down. However, this post-processing technique produces discrepancies between the settings for training and testing (synthesis). Two unique approaches to solving this issue are put out in the literature to reduce this discrepancy: (a) changing the cost function to take trajectory error into account rather than the commonly utilised framewise error as in [56] [125]; or (b) employing RNNs. Using MLPG is still advantageous, even though RNNs do predict temporally continuous parameter sequences (this conclusion was made in [31]).

We look into using deep Elman RNNs with SPSS. We demonstrate that ERNNs can perform similarly well in comparison to gated RNNs, despite the fact that more complicated RNN architectures, such as LSTM and variants thereof, such as gated recurrent units and

simplified LSTM, have been investigated for SPSS in previous chapters. This offers computational economy without sacrificing performance, and is theoretically easier, as will become clear in following parts.

5.1.2. Contextual Representation

The text features in a typical DNN-based SPSS system consist of (a) categorical features (such as pentaphone identities, identity of the vowel in the current syllable, etc.), (b) numerical features (such as number of syllables in a word, number of words in a phrase, and so on), and (c) duration features (such as frame index, duration of phone, etc.). Typically, 1-hot-k encoded representations are used for the categorical features. However, using continuous valued representations, as discussed in [117], would be a more appropriate kind of representation. Another strategy to be taken into consideration is learning embeddings for the phones, words, and utterances as in [126].

In order to add the hidden state as an acoustically guided contextual representation for the traditional text features, we train an ERNN for predicting acoustic features. The primary distinction between this approach and embeddings is that the former can be adjusted for each new test speech, whilst the latter, once learned, are fixed.

5.2. Elman RNNs for SPSS

Elman RNNs have long been recognised as effective models for tasks like sequence creation and prediction, but their use has been constrained due to the challenges associated with training them because of the exploding and disappearing gradients problem [58]. In the recent past, efforts have mostly been made in the following directions to avoid this issue: Finding more effective initialization techniques [127]; boosting learning speed and making stochastic gradient descent-based back-propagation through time (BPTT) [128] work similarly to a few second order methods (previously suggested to solve the issue [129]); incorporating heuristics like gradient clipping [11] whenever the gradients' norm crossed a threshold; and using

regularisation to prevent vanishing gradients problem [130]. However, ERNNs are preferred over LSTM/GRU/SL STM for two reasons: (1) Gated recurrent architectures have significantly more parameters than ERNN for a given hidden state size, and the number of computational steps is also significantly higher because the recurrent unit has more gates; and (2) It is ambiguous which components of the complex architectures are crucially important.

ERNNs are networks shown in figure 5.1 that lack sophisticated gating mechanisms yet have a hidden state. ERNN is formally defined as the following:

$$h_t = f\,(W_i\, x_t + W\, h_{t-1} + b_h) \tag{17}$$

$$y_t = g\,(U\, h_t + b_o) \tag{18}$$

Where **Wi** stands for the input to hidden weights, **W** for the recurrent weight matrix of hidden layer, b**h** for hidden bias, **U** for the hidden to output weight matrix and b**o** for the output layer's bias vector. At the hidden and output layers, **f** and **g** are nonlinear functions, respectively. At time t, the variables **xt, ht and yt** represent the input, state, and output, respectively. At time instant t-1, **ht-1** represents the state.

The error signal is computed at the hidden layer through BPTT as

$$e_t = d_t - y_t \tag{19}$$

$$\delta_t = f\,` * (W_t\, \delta_{t+1}\ \ U^T e_t) \tag{20}$$

Where **dt** represents at time **t,** the desired signal and $\mathbf{e_t}$, $\boldsymbol{\delta_t}$ stand for the error signal in the hidden layer and output layer, respectively, at time **t**. **f `** represents the derivative of the hidden layer activation function and **T** denotes the duration of the sequence. The network's parameters are adjusted in relation to the MSE loss function shown below:

$$E = \frac{1}{T}\sum_{t=1}^{T}(\,e_t\,)^2 \tag{21}$$

Back-propagation can be used to learn the output layer parameters (Eq. 18) and back-propagation through time (BPTT) can be used to learn the model parameters (Eq. 17).

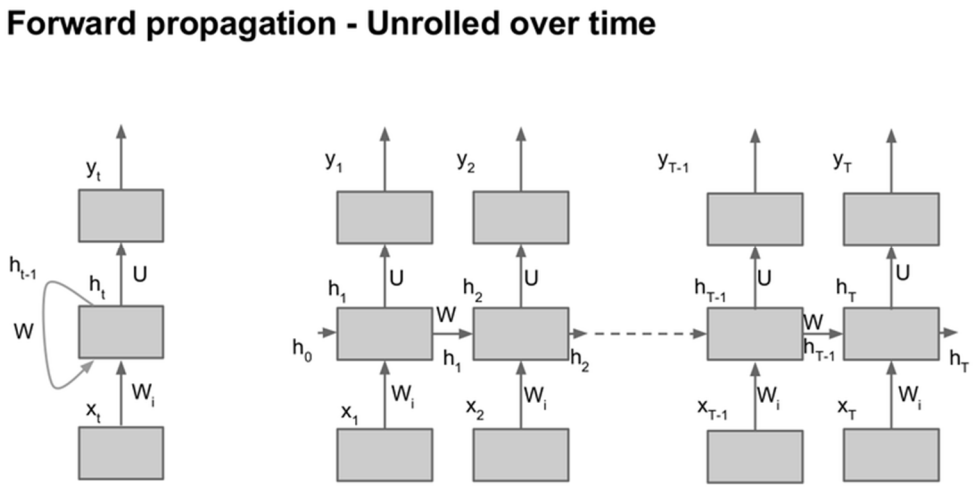

Figure 5.1. Elman RNN architecture (reproduced from [53]).

5.3. ERNN Guided DNN/ERNN based SPSS System

DNN and ERNN-based acoustic modelling enhances the performance by adding frame-wise contextual representation using a pre- trained ERNN's hidden state. First, we train the ERNN as a CRL system depicted in figure 5.2 utilising common text features as input and either spectrum or F0 as output features. The trained ERNN's hidden state is then added to the regular text features in order to train the synthesis system. Either ERNN-MCEP, ERNN-F0, or both networks may contain the concealed state. The resulting DNN and ERNN are trained to predict acoustic properties stream-wise.

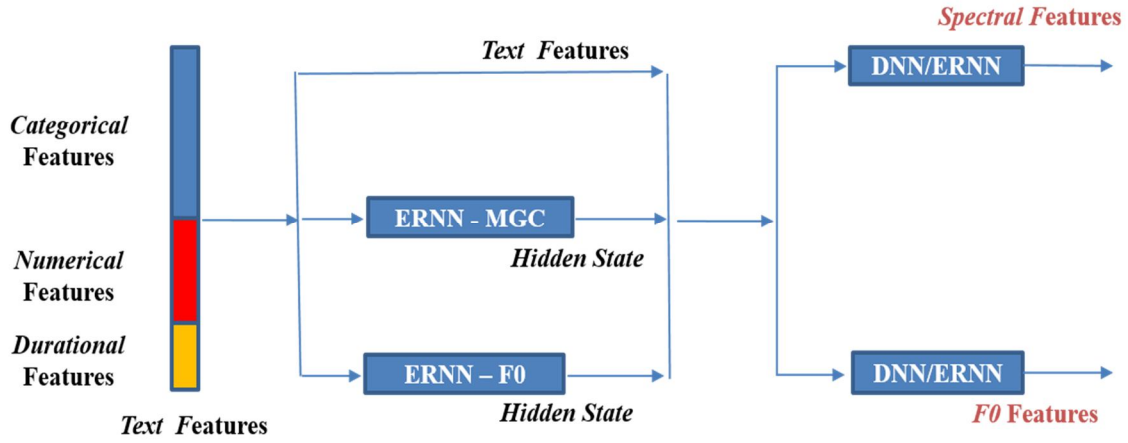

Figure 5.2. Framework of DNN/RNN based SPSS system with Pre-trained ERNN hidden state.

The ERNN-MCEP and ERNN- F0 are the pre-trained ERNNs, whose hidden states are fed as acoustically guided context information in addition to the baseline textual features to train another DNN/ERNN

5.4. Description of Speech Corpus

BDL & SLT US voice from CMU ARCTIC dataset [103] was used for English language. Out of a total of 1132 utterances, 1075 were used for training and the final 56 were used for testing and validation.

IIIT Hyderabad [101], and INDIC voices from the INDIC dataset were used for Telugu. Out of a total of 1000 utterances, 950 were used for training and the final 50 were used for testing and validation. A different data set from a portion of the “Blizzard Challenge 2015 database” was used for Telugu. It includes approximately 4 hours of speech data in Telugu, Hindi, and Tamil, as well as around 2 hours of speech data in Bengali, Marathi, and Malayalam, all of which were recorded by native professional speakers in high-quality studio settings. For our tests, we used a Telugu language dataset with 2355 utterances. The leaked speech recordings were 16 kHz samples. For acoustic feature extraction, the data were up-sampled to

48 kHz using the HTS-STRAIGHT demo versions 2 and 3. EHMM tool was used to perform alignments at the phone level [1] [44]. The feedforward network utilised in this work has a three layer architecture, or 359 L 250 N 235 L for English, and 529 L 250 N 235 L for Telugu. Along with that worked upto ten hidden layer architecture but shown more or less similar performance. Detailed experimentation given in following sections.

5.5. Experiments and Results

5.5.1. Input Features

A careful representation is required at the input layer because we are working on a mapping from text input space to formant output space. In addition to the difficulty of this mapping, we anticipate that the model will offer modest variations in the formants and bandwidths for each frame.

For the purpose of training the NN model, the text's current, left and right phone articulatory, and syllable properties were extracted. There are also temporal features (the location of the frame within the current phone), current frame state information, the location of the current phone within the word, the location of the current word within the sentence, and these. Please be aware that state information is utilised into NN modelling to aid in differentiating the phone frame types.

The MCEP and F0 features of ERNN are the pre-trained ERNNs features, whose hidden states are fed as acoustically guided context information in addition to the baseline textual features to train another DNN/ERNN.

5.5.2. Output Features

At the output layer, it is hoped that the network will forecast MCEPs. All of the spoken utterances' 26-dimensional band-aperiodicities, 50-dimensional Mel-general cepstral

characteristics, and deltas and double-deltas were extracted with a 5 ms frame-shift. This feature extraction is based on the online HTS- STRAIGHT demo. F0 characteristics include binary voiced/unvoiced flags, F0, Δ F0, Δ Δ F0, and VUV. The most effective threshold as determined by training data was used to set the voiced/unvoiced threshold during testing. With a 5ms frame shift, 235 coefficient vectors were predicted for each 10ms frame size. The back propagation learning approach was used to train the NN model during a 20 iteration period. The features of the input and output were adjusted for mean variance.

5.5.3. Models

As a starting point, we employ DNN with typical features. Early halting was employed to end the training for all networks. While no regularizer was employed for RNNs, For DNNs, a weight decay regularizer was applied with a decay weight (λ) = 0.00005.

5.5.3.1. Weight Initialzation

For DNNs, we employed normalised initialization. A Gaussian distribution was used to select all of the non-recurring weight values. The Gaussian distribution was used to determine every parameter for LSTMs, GRUs, and SLSTMs, with the standard deviation set at 0.01. We used entire BPTT for training RNNs. We used [60]'s diagonal initialization recommendation for ERNNs. During diagonal initialization, the recurrent weight matrices are initialised to scaled Identity matrices. As a result, the spectral radius can be chosen for stability and to benefit from orthogonality.

5.5.3.2. Hyper parameter Setting

We use "Adam" to automatically set the learning rate for each dimension when doing stochastic gradient descent. The hyper- parameters (α is the learning rate and $\beta 1$, $\beta 2$ are

exponential decay rates) were chosen as α = {3e −3, 1e −3, 3e−4, 1e −4}, β1 = 0.9 and β2 = 0.999. Table 5.1 provides information on the models that were trained.

Table 5.1 Details of Hyper-Parameter Setting and Initialization of Models

Model (System)	Activation Function	SGD Method	Initialization Method
DNN	ReLU (R)	Adam	Normalized Initialization
ERNN	ReLU (R)	Adam	Diagonal Initialization
LSTM / GRU / SLSTM	tanh (N)	Adam	Gaussian Initialization

5.5.4. Subjective Evaluation

Subjective listening tests called MUSHARA (“MUltiple Stimuli with Hidden Reference and Anchor”) [124] were used to assess how natural speech sounded. To give the listener a sense of perceived quality, the sentences are between 10 and 15 words long. Synthesised speech files produced by advanced TTS systems are used for subjective assesment, with the MOS serving as a measurement criterion. The tests were conducted with provided headphones in a quiet setting. The recorded original speech samples of the same speaker and the synthetic voice files (DNN, SLSTM, GRU, LSTM and ERNN) were played simultaneously at random. The exams involve twenty listeners in all. In the creation of any TTS voice, these listeners are not involved.

To ascertain which system (A or B) is more preferred for the experiment detailed in Section 5.6, we conducted a subjective preference test (ABX). The "No Preference" (X) option was available to participants if both systems were performing equally. And listed in table 5.4. The MOS Test Scores of 5 point scale are presented in table 5.3. Systems S6 and S7 are the context representation learning systems using DNN and ERNN respectively. The CRL system uses DNN as a synthesis network with improvement of 0.12 over DNN based SPSS system

(S1) and 0.1 over ERNN (S5) where as Deep Elman RNN as synthesis network improves 0.16 over DNN and 0.14 over ERNN based SPSS system respectively.

5.5.5. Objective evaluation

"Mel-cepstral distortion for the spectrum, RMSE for F0, % error in frames for VUV, and Euclidean distortion for BAP features" are among the objective metrics. Global variance spectral augmentation was performed before the subjective listening tests but not when calculating MCD scores. All parameter tracks, however, used MLPG smoothing. The results are displayed in table 5.2.

There is a noticeable improvement in the ERNN/ERNN based SPSS system compared to other systems, and the objective scores of the CRL system employing ERNN (S5) as the synthesis network gave better scores than other systems. As seen in preliminary synthesis studies, the projected BAP features from ERNNs were enough.

5.6. Comparison of Deep ERNNs with other RNNs and DNN

In this experiment, we use both objective and subjective metrics to assess and contrast the performance of several RNN models and DNN models. Five hidden layers, each with 500 ReLU units, were used to train all of the DNNs. We have tested the impact of breadth and depth on the acoustic modelling performance within the RNN models. With a depth of 1 and 2 hidden layers, namely, 250 and 500 hidden units were taken into consideration. It is obvious that for all models, increasing depth or breadth—or both—improves performance; however, it's vital to note that, in contrast to boosting depth, increasing the hidden state size from 250 to 500 causes a significant rise in parameters.

Therefore, the size of the hidden state and the number of layers must be carefully chosen in order to have an appropriate amount of parameters in the model. Models with 1M parameters must be compared in order to compare the RNN models below fairly. For instance, LSTM has 3.81M parameters while a 2-layer deep ERNN with 500 hidden units has 1.04M. Therefore, it

would be preferable to compare a 2-layer LSTM with 250 units each and 1.15M parameters to maintain the same level of model complexity. The objective scores of all models listed in table 5.2.

Table 5.2 Objective Metrics of Telugu Language NN based SPSS Systems (including ERNN)

System	Architecture	Parameters (in millions)	MCD (dB)	F0 (RMSE)	VUV (% error)	BAP (dB)
DNN (S1)	**5 x 500R**	**1.3**	**4.21**	**32.13**	**6.22**	**28.20**
SLSTM (S2)	250N	0.35	4.25	32.12	6.22	28.27
	500N	0.96	4.25	**31.98**	6.22	28.22
	2 x 250N	0.63	4.23	31.99	**6.20**	**28.20**
	2 x 500N	1.96	**4.21**	31.99	6.22	28.21
GRU (S3)	250N	0.5	4.24	33.52	6.23	28.42
	500N	1.38	**4.23**	**33.18**	**6.21**	**28.32**
	2 x 250N	0.88	4.24	32.29	6.22	28.36
LSTM (S4)	250N	0.65	4.23	33.13	**6.20**	28.28
	500N	1.83	**4.21**	**32.27**	6.21	**28.19**
	2 x 250N	**1.18**	4.22	33.22	6.23	28.33
ERNN (S5)	250N	0.22	4.23	33.12	6.22	28.27
	500N	0.62	4.21	32.98	6.19	28.22
	2 x 250N	0.33	4.22	32.39	6.20	28.23
	2 x 500N	**1.16**	**4.19**	**31.86**	**6.19**	**28.20**
DNN/ERNN (S6)	2 x 500N	**1.22**	**4.09**	**31.67**	**6.08**	**-**
ERNN/ERNN (S7)	2 x 500N	**1.04**	**4.05**	**31.54**	**6.03**	**-**

The objective scores of CRL system using ERNN (S5) as synthesis network gave better scores than other RNNs and compared to all RNN and DNN systems, the ERNN/ERNN-based

SPSS system (S7) has a significant improvement.. Results show the clear improvement in the quality of ERNN based TTS system when it is pretrained with acoustically guided ERNNs.

Table 5.3 Subjective Metrics of Telugu Language NN based SPSS Systems (including ERNN)

System Model	Naturalness	Intelligibility	Quality
DNN (S1)	3.23	3.18	3.15
SLSTM (S2)	3.24	3.19	3.21
GRU (S3)	3.25	3.17	3.22
LSTM (S4)	3.27	3.21	3.23
ERNN (S5)	3.25	3.28	3.32
DNN/ERNN (S6)	3.37	3.42	3.41
ERNN/ERNN (S7)	**3.43**	**3.48**	**3.46**

Systems S6 and S7 are the context representation learning systems using DNN and ERNN respectively. The CRL system uses DNN as a synthesis network with improvement of 0.14 over DNN based SPSS system (S1) and 0.12 over ERNN (S5) where as Deep Elman RNN (S7) as synthesis network improves 0.2 over DNN and 0.18 over ERNN based SPSS system respectively.

The results of the pairs ABX listening tests, which were used to ascertain subjective preference, are shown in table 5.4 below. The following system pairs were taken into consideration:

- DNN (5x500R) Vs GRU (2x250N)
- DNN (5x500R) Vs LSTM (2x250N)
- DNN (5x500R) Vs ERNN (2x500R)

• GRU (2x250R) Vs LSTM (2x250N)

• ERNN (2x500R) Vs LSTM (2x250N)

• ERNN (2x500R) Vs GRU (2x250N)

Table 5.4 Subjective preference tests of Telugu Language NN based SPSS Systems (including ERNN)

System	DNN	GRU	LSTM	ERNN	No Preference
1	40.05	50.38	-	-	8.57
2	38.23	-	55.11	-	6.66
3	18.23	-	-	76.02	5.75
4	-	15.07	18.12	-	66.81
5	-	13.03	-	16.76	70.11
6	-	-	12.01	20.23	67.76

The subjective listing test scores of system using ERNN (S5) as synthesis network gave better scores than other systems. The listeners strongly prefer ERNNs over DNN. As can be seen from the results table, the pairs of LSTM-RNNs and GRU, GRU and ERNN, LSTM-RNN and ERNN are doing more or less equally well, which explains why participants frequently select "No Preference" (more than 60% of the time). The ERNN-based SPSS systems can be trained and tested quickly and more efficiently with pretrained ERNNs.

5.7. Summary

We investigated deep Elman RNNs for SPSS in this chapter. We have demonstrated for the Telugu language through rigorous objective and subjective assessments that Deep gated

RNNs like ERNNs, SLSTMs, GRUs, and LSTMs may perform competitively to DNNs when the right weight initialization strategy and gradient clipping are utilised. As a result, RNN-based SPSS systems can be trained and tested quickly and efficiently. Less parameters also offer a storage advantage, which is extremely beneficial in mobile apps. Given the benefits over the LSTM-RNNs, our investigated method might make the process of transferring SPSS system to mobile devices even simpler. Subjective and objective scores clearly shows that the use of pretained ERNNs to guide the ERNN based SPSS systems greatly improves the quality as well reducing the computaional cost.

Chapter - 6

End To End TTS System Using Tacotron 2

Traditional speech synthesis techniques, like statistical parametric speech synthesis techniques, are made up of a number of independent parts, including a) a text frontend that extracts different linguistic features, b) a duration model, c) an acoustic feature prediction model, and d) a complex signal-processing-based vocoder. As a result, the overall system is challenging to modify and requires in-depth domain knowledge to design. End-to-End TTS systems built on ANNs have recently shown to have significant capacity in generating emotional and lifelike speech straight from raw text.

6.1. Introduction

Modern end-to-end voice synthesis technology that can accurately anticipate close-to-natural speech is given a lot of attention in this chapter. End-to-End TTS models are simpler to develop than SPSS models since they only require one model, and they also support rich conditioning that improves the controllability of synthesised speech. A character sequence is inputted into the Tacotron [34], an integrated “end-to-end generative TTS model”, which then

generates the associated spectrogram. More quickly than sample-level autoregressive approaches, it is frame-based. With random initialization and given text and audio pairs, Tacotron can be completely trained from scratch. Since phoneme-level alignment is not necessary, it scales well to the use of massive volumes of acoustic data along with transcripts. Tacotron produces sound of good quality using a straightforward waveform synthesis technique, surpassing the naturalness of a production parametric system but falling short of unit selection synthesis. However, it is clear that the synthesised speech produced by such models differs from actual speech, as synthesised speech sounds unnatural and rigid due to its excessive smoothness.

A cutting-edge end-to-end speech synthesis model, “Tacotron 2” [131], may produce speech directly from graphemes or phonemes. An encoder and a decoder with attention make up the network. It is Tacotron in a modified form. After a recurrent sequence-to-sequence melspectrogram prediction network, a modified WaveNet functions as a vocoder to produce time-domain waveforms from the anticipated spectrograms. Tacotron 2 uses less complex building blocks, Vanila LSTM and Convolutional layers, in the encoder and decoder in place of "CBHG stacks" and GRU recurrent Layers. Figure 6.1 depicts the Tacotron 2's architectural layout.

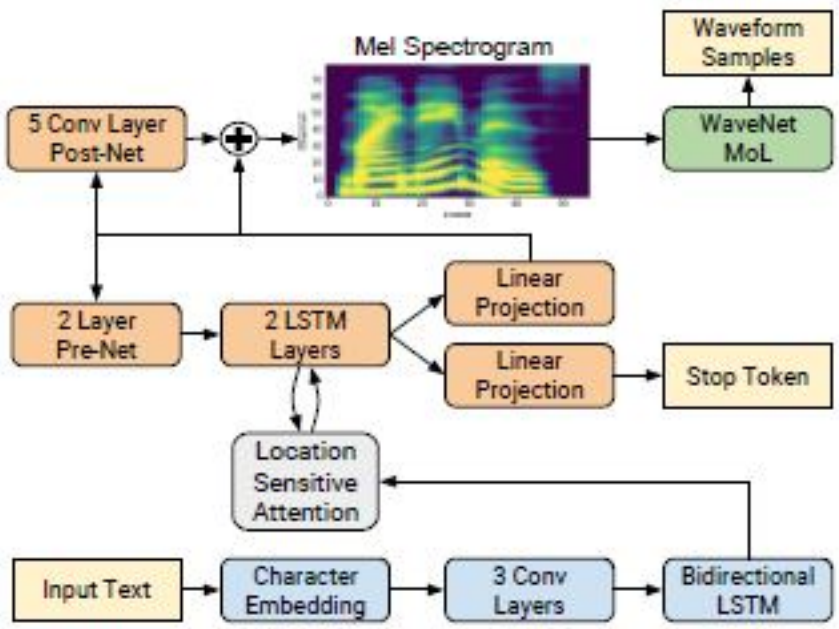

Figure 6.1. Tacotron 2 Architecture reproduced from [131]

6.2. Encoder – Decoder Framewok

The backbone of end-to-end systems is a seq2seq model with attention The main structure blocks in the tacotron or tacotron 2 are encoder, attention module, decoder, Pre-net and post- processing net. Detailed framework of tacotron 2 given in following sections.

6.2.1. Encoder

The “encoder’s” objective is to obtain reliable sequential representations of text. A character sequence is transformed by the encoder into a hidden feature representation that the decoder uses to forecast a spectrogram. A learned 512-dimensional character embedding is used to represent the input characters, which are then passed through a stack of three convolutional layers. Each convolutional layer five 512 filters in the shape of 5 x 1, i.e., one filter for every 5 characters, followed by batch normalisation and rectified linear unit activations. These convolutional layers, like Tacotron, simulate longer-term context (like N-grams) in the input character sequence. The encoded features are produced using a single bi-directional [132] LSTM [121] layer with 512 units (256 in each direction), which is fed the output of the final convolutional layer. The encoder's internal components are depicted in figure 6.2 below. With a probability of 0.5, dropout was used to regularise the network's convolutional layers, while azoneout [134] was used to regularise the LSTM layers.

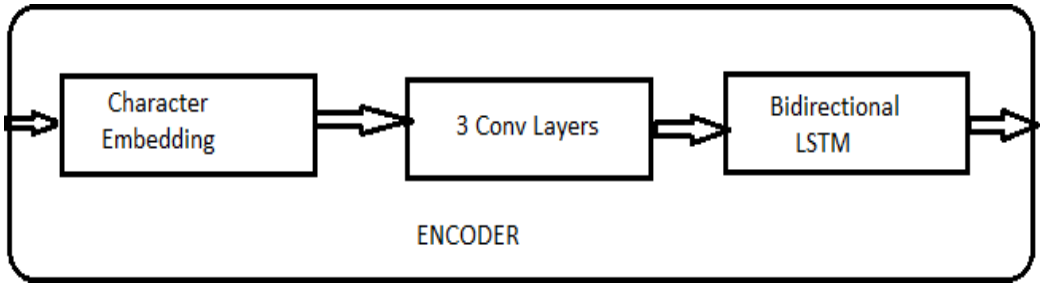

Figure 6.2. Encoder Internal Parts

6.2.2. Attention module

The mechanism of attention, a relatively recent and well-liked strategy, is another essential component of the system. "Attention" creates the context vector and modifies the attention weight at each decoding stage.The encoder and decoder stages of the encoder-decoder paradigm are connected by the location sensitive attention. When decoding the current generated word, $c_{i,}$ combines each unique ci to produce a result that is more accurate than the one before it. The portion of the encoder data that should be utilised at the current decoder step is defined by attention.

An “attention” network uses the encoder output and summarises the entire encoded sequence as a fixed-length context vector for each step of the “decoder” output. By extending the additive attention mechanism to include cumulative attention weights from earlier decoder time steps as an extra feature, the “location-sensitive attention mechanism” from [135] is used. As a result, the model is encouraged to proceed consistently through the input, reducing the likelihood that some subsequences would be repeated or ignored by the decoder and preventing probable failure modes.After projecting inputs and locationfeatures to 128-dimensional hidden representations, attention probabilities are calculated. 32 “1-D convolution filters” with a length of 31 are used to calculate location characteristics.

6.2.3. Decoder

The “decoder”, which predicts a mel spectrogram from the encoded input sequence one frame at a time, is an “autoregressive recurrent neural network”. It consists of a convolutional post net, a fully connected prenet, and a two-layer LSTM network. The previous time step's prediction is transmitted first through a tiny “pre-net” that has two fully connected layers and 256 hidden rectified linear units. We discovered that the “pre-net” serving as an information bottleneck was required for the learning attention. The “pre-net” output and “attention context vector” are concatenated and passed through a stack of 2 “uni-directional LSTM” layers with 1024 units. To forecast the target spectrogram frame, the “LSTM” output and the “attention context vector” are concatenated and projected using a linear transform. A 5-layer

convolutional post-net is used to predict a residual to add to the prediction and improve the overall reconstruction after the predicted mel spectrogram has been passed through it. Each post-net layer consists of 512 filters in the shape 5 x 1 with batch normalisation, followed by tanh activations on all but the final layer. Figure 6.3 depicts the decoder internal parts.'

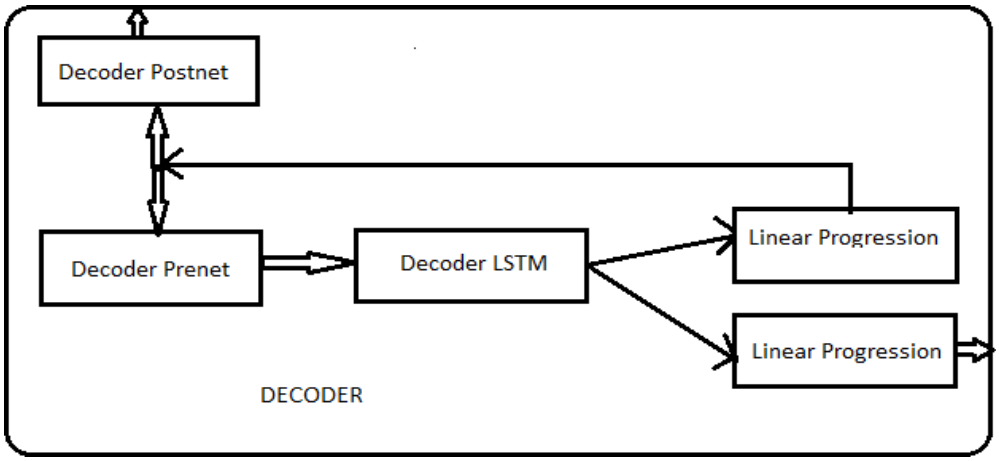

Figure 6.3. Decoder Internal Parts

To promote convergence, the summed “mean squared error” from before and after the post-net is minimised. To avoid assuming a constant variance over time, we also experimented with a “log-likelihood loss” by modelling the output distribution using a “mixed density network”, but we found that these were more challenging to train and did not provide better sounding samples.

The concatenation of the “decoder LSTM” output and the “attention context” were projected down to a scalar and sent through a “sigmoid activation” in parallel to spectrogram frame prediction in order to calculate the likelihood that the output sequence would be finished. The model was able to dynamically select when to stop generating rather than doing so for a predetermined period of time every time by applying this "stop token" prediction during inference. The generation is finished in the first frame for which this probability crosses the threshold of 0.5. Only the layers in the pre-net of the autoregressive decoder were subjected to dropout with a probability of 0.5 in order to induce output variance at inference time.

6.2.4. Mel-Frequency Spectrogram

The recurrent sequence-to-sequence feature prediction network and wavenet are connected by a mel-frequency spectrogram. A linear-frequency spectrogram, also known as the STFT magnitude, and a mel-frequency spectrogram are related. It is made by performing a nonlinear transform on the STFT's frequency axis, which compresses the frequency information into fewer dimensions and is based on observed responses from the “human auditory system”. The function of such an auditory frequency scale is to de-emphasize high frequency features, which are often dominated by fricatives and other noise bursts and do not require high fidelity modelling, while accentuating details in lower frequencies, which are crucial to speech intelligibility. These characteristics have led to the usage of features generated from the mel scale as an underpinning representation for speech recognition for many years.

While linear spectrograms lose information due to the discarding of phase information, methods like Griffin-Lim [136] are able to estimate this lost data, allowing for time-domain translation via the inverse short-time Fourier transform. Mel spectrograms eliminate even more data, posing a difficult inverse challenge. The mel spectrogram is a less complex, lower-level acoustic representation of audio data than the language and acoustic features employed in WaveNet. Therefore, it ought to be simple for a comparable WaveNet model trained on mel spectrograms to produce audio, effectively acting as a neural vocoder.

Tacotron 2 uses less complex building blocks than the original Tacotron, replacing "CBHG" stacks and GRU recurrent layers in the encoder and decoder with vanilla LSTM and convolutional layers. Since there is no "reduction factor" in use, each decoder step corresponds to a single spectrogram frame.

6.3. WaveNet Architecture

WaveNets are efficient audio decoders that can convert high-quality audio, including music and voice, to latent representations. WaveNet is an autoregressive model. A generative model that operates directly on the unprocessed audio waveform is used to model audio signals. A waveform's joint probability is factored as a result of conditional probabilities. Each audio

sample is consequently dependent on samples from all earlier timesteps. A stack of convolutional layers is used to model the conditional probability distribution. The model's output has the same time dimensionality as the input and lacks any pooling layers in the network. The model generates a softmax layer-based categorical distribution over the subsequent value xt. The model is enhanced to increase the data's log-likelihood with respect to the parameters. WaveNets can simulate the conditional distribution of the audio given an additional input h.

We modified the WaveNet architecture from [63] to translate the mel spectrogram feature representation into "time-domain waveform samples. Instead of forecasting discretized buckets with a softmax layer, we follow "PixelCNN++" and "Parallel WaveNet" and generate 16-bit samples at a rate of 24 KHz using a 10-component mixture of logistic distributions. To compute the logistic mixed distribution, the “WaveNet stack output” is processed through a “ReLU activation” followed by a linear projection to predict parameters (mean, log scale, and mixture weight) for each mixture component. The ground truth sample's negative log-likelihood is used to calculate the loss.

6.4. Description of Speech Corpus

This work made use of the LJ Speech Dataset for English language. A single speaker can be heard reading portions from seven non-fiction books in 13,100 brief audio samples that make up the public domain speech collection. Every footage comes with a transcription. The entire duration of the clips, which range in length from 1 to 10 seconds, is about 24 hours. The texts are in the public domain and were published between 1884 and 1964. IIIT Hyderabad [101], and INDIC voices from the INDIC dataset were used for the Telugu language.

6.5. EXPERIMENTS & RESULTS

6.5.1. Training Setup

We individually trained the feature prediction network first, and then we independently trained a modified WaveNet using the first network's outputs. We trained the feature prediction network on a single GPU using the popular maximum-likelihood training method with 64 batches. This technique, also referred to as "teacher-forcing," involves feeding the decoder side the actual output rather than the anticipated output. We use the Adam optimizer [29] with β_1 = 0:9; β_2 = 0:999; $\in$ = 10^{-6} and learning rate of 10^{-3} exponentially declining to 10^{-5} starting after 50,000 iterations. Additionally, weight 10^{-6} is used for L2 regularisation.

We next used the feature prediction networks ground truth-aligned predictions to train our modified WaveNet. In other words, the prediction network is used in teacher-forcing mode, where each predicted frame is dependent on the input sequence's encoding and the frame before it in the spectrogram of the ground truth. As a result, each forecast frame is guaranteed to perfectly line up with the target waveform samples. Averaging model weights across recent updates improves quality. The network parameters are therefore maintained as an exponentially-weighted moving average over update steps with a decay of 0.9999; this version is utilised for inference. By scaling the waveform targets by a factor of 127:5, we can hasten convergence by bringing the mixture of logistics layer's initial outputs closer to the final distributions.

On a dataset from LJ Speech involving a female speaker, we train all models. Each of our models is trained using normalised text..

6.5.2. Subjective Evaluation

The ground truth targets are unknown when producing speech in inference mode. Therefore, in contrast to the “teacher-forcing setup” employed for training, the expected outputs from the prior phase are supplied in during decoding. From the dataset's test set, we chose 100

fixed examples at random to serve as the evaluation set. Audio created with this set is transmitted to a human rating service like to “Amazon Mechanical Turk”, where each sample is scored subjectively by 20 raters using a scale from 1 to 5 with 0.5 point increments. These raters are not involeved in any speech research. Table 6.1 shows clearly that the end-to-end TTS architectute enhances the quality of th esynthesised speech as natural as.

6.6. Comparison of Tacotron 2 with USS and NN based SPSS

We conducted mean opinion score tests where the subjects were asked to judge the naturalness of the speech generated by the system on a 5-point scale in order to evaluate the effectiveness of our End-to-End speech synthesis. The experimental results indicated that our system performs well and gave a natural and intelligible speech as output. The results are compared with previous modelled systems and presented in Table 6.1. This end-to end model outperforms than the previos models.

Table 6.1 MOS scores with Confidence Interval of developed TTS systems

Model	**MOS ± CI**
Tacotron2	**3.99 ± 0.08**
Tacotron	3.82 ± 0.34
Parametric (ERNN/ERNN)	3.46 ± 0.10
Concatenative (USS)	**4.09 ± 0.11**
Ground Truth (Original speech)	**4.22 ± 0.97**

The end-to-end model will produce a voice that is as natural-sounding as a human voice, according to the MOS ratings. The resulting system can synthesise speech with WaveNet-level audio quality and tacotron-level prosody. In order to comprehend the essential elements of our model, we must carry out a few ablation investigations. It's challenging to compare models

using objective measurements because they frequently do not correlate well with perception, as is typical for generative models.

6.7. Summary

The quality of synthesised speech has improved significantly by using end-to-end TTS models. The inference is much quicker than sample-level autoregressive approaches because it is frame-based. The resulting system can synthesise speech with WaveNet-level audio quality and prosody. This technique may be trained directly from data without the use of intricate feature engineering, and it delivers modern sound quality that is comparable to that of natural human speech.

Numerous areas of our model still require investigation, and we are now striving to improve the speed and naturalness of the projected speech.

Chapter - 7

7.1. Conclusions

In this thesis, a significant focus is on modification of prosodic features and investigating diferrent deep learning architectures for increasing the quality of Telugu speech synthesis system with small foot print. Adjusting the prosodic features: pitch and intensity contours of Telugu synthesized speech around each syllable boundary, leads to develop the USS based TTS system with small foot print. Different deep architectures like “Deep Neural Networks”, “Simple Recurrent Neural Networks”, “Long Short Term Memory - Recurrent Neural Networks”, “Gated Recurrent Neural Networks” and “Elman Recurrent Neural Networks” can be incorporated for efficient acoustic modelling of complex context dependencies to build high quality, small dataset TTS system. Encoder-Decoder based an End to End TTS system was investigated to develop TTS System for English language. All these atrchitectures have been investigated, and used to develop a small foot print high quality TTS systems for Telugu and English languages in this report.

Chapter 3 proposes prosodic features adjustment in order to develop small foot print USS based TTS system with significant subjective scores. We made an attempt to adjust the

pitch and intensity contours of Telugu synthesized speech around each syllable boundary in this work. To some extent, the proposed prosody modification gave better perceived discontinuity. Large discontinuities, cannot be totally avoided. To generate natural syllable boundary, future research could focus on spectral properties modification rather than pitch and intensity.

In Chapter 4, demonstration of DNNs and RNNs for SPSS was done. We have demonstrated for the Telugu language through rigorous objective and subjective assessments that RNNs like "SLSTMs", "GRUs", and "LSTMs" may perform competitively to DNNs when the right "weight initialization" strategy and "gradient clipping" are utilised. As a result, RNN-based SPSS systems can be trained and tested quickly and efficiently. Less parameters also offer a storage advantage, which is extremely beneficial in mobile apps. Given the benefits over LSTM, this method might make the process of transferring the SPSS system to communication (mobile) devices even simpler. Recently, an exploration employing LSTMs for SPSS was done. The MCD of RNN-based SPSS system was improved over DNN based SPSS. Improvements were seen in mel-cepstral distortion. This increase is essentially identical to what would be gained by doubling the training data.

Chapter 5 investigated deep Elman RNNs for SPSS. we trained ERNNs as a context representation learning system using standard text characteristics as input and either spectrum or F0 as output features. The trained ERNN's hidden state is then added to the regular text characteristics in order to train the synthesis system. Either ERNN-MCEP, ERNN-F0, or both networks may contain the concealed state. The resulting "DNN" and "ERNN" is trained for stream-wise prediction of acoustic characteristics. This enhances the performance of System. Improvements were seen in objective metric - MCD.

In Chapter 6, a novel paradigm an End to End TTS system using Tacotron 2 was investigated for English language. The Seq-to-Seq Feature prediction network in this model of end-to-end speech synthesis transfers the character vector to the Mel-Spectrogram, a highly

effective tool for producing high-quality speech. The results of the trial showed that the system operates effectively and produces natural and understandable speech. The inference is much quicker than sample-level autoregressive approaches because it is frame-based. Numerous areas of our model still require investigation, and we are now striving to improve the speed and naturalness of the projected speech.

7.2. Future scope

This section provides future directions for work on small foot print high quality TTS systems. We made an attempt to adjust the pitch and intensity contours of Telugu synthesized speech around each syllable boundary in this work. We also developed a small foot print TTS systems with significant subjective scores. To some extent, the proposed prosody modification gave better perceived discontinuity. Large discontinuities cannot be totally avoided. To generate natural syllable boundary, future research could focus on spectral properties modification rather than pitch and intensity. Further we investigated different neural network architectures for efficient acoustic modeling. ERNN as a CRL system using standard text characteristics as input and either spectrum or F0 as output features. Then, in order to train the synthesis system, we append the pre-trained ERNN's hidden state to regular text characteristics. Either ERNN-MCEP, ERNN-F0, or both networks may contain the concealed state. The resulting DNN and ERNN is trained for stream-wise prediction of acoustic characteristics. We want to further our understanding of ERNN and use nonlinearities like exponential linear units to voice processing problems in our future work. The performance of the acoustic modelling will probably be improved by using these advanced nonlinearities. Recent attempts have been made to regularise ERNNs without the issue of vanishing or exploding gradients. We intend to use these methods with our ERNN-based SPSS solution. The subjective 5-scale mean opinion score for the End-to-End speech synthesis system was 3.99± 0.085. As a result of being frame-based, the inference is much quicker than sample-level autoregressive approaches. Several aspects of our paradigm remain to be investigated and we are at present working on further improvement of fastness and naturalness of the predicted speech.hybridzation of the exisiting end-to-end models surely results in more robust quality systems without any discontinuity.

Finally, many researchers have contributed to the development and evaluation of high quality speech synthesis system using different deep learning architectures. Numerous strategies and full algorithms have been put forth in English during the early 2000s. Unfortunately, local languages like Telugu (location specific) have rarely been used to report comparisons across approaches from various authors. Future research on a thorough comparison of cutting-edge techniques would be fascinating and helpful for the development of deep learning-based TTS systems for local languages.

References:

[1] Anila Susan Kurian, Badri Narayan, Nagarajan Madasamy, Ashwin Bellur, Raghava Krishnan, Kasthuri G., Vinodh M.V., Hema A. Murthy, Kishore Prahallad, "Indian Language Screen Readers and Syllable Based Festival Text-to-Speech Synthesis System" in Proceedings of the 2nd Workshop on Speech and Language Processing for Assistive Technologies, pages 63–72, Edinburgh, Scotland, UK, July 30, 2011.

[2] E. Raghavendra and K. Prahallad, "A multilingual screen reader in indian languages," in Communications (NCC), 2010 National Conference on, Jan 2010, pp. 1–5.

[3] Michael Alexander Kirkwood Halliday, Christian Matthiessen, and Michael Halliday. An introduction to functional grammar. Routledge, 2014.

[4] S. P. Kishore and A. W. Black, "Unit size in unit selection speech synthesis," in INTERSPEECH, 2003.

[5] S P Kishore, Rohit Kumar and Rajeev Sangal, "A Data Driven Synthesis Approach For Indian Languages using Syllables as Basic Unit," in Proceedings of Intl. Conf. on NLP (ICON) 2002, pp. 311- 316, Mumbai, India.

[6] W. Lawrence, "The synthesis of speech from signals which have a low information rate," Communication Theory, pp. 460–469, 1953.

[7] D. H. Klatt, "Review of text to speech conversion for english," The Journal of the Acoustical Society of America, vol. 82, no. 3, pp. 737– 793, 1987. [Online]. Available: http://scitation.aip.org/content/asa/ journal/jasa/82/3/10.1121/1.395275

[8] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on, vol. 1, May 1996, pp. 373–376 vol. 1.

[9] J. Benesty, M. M. Sondhi, and Y. A. Huang, Springer Handbook of Speech Processing. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[10] Zen, Heiga, Keiichi Tokuda, and Alan W. Black. "Statistical parametric speech synthesis" Speech Communication, Volume 51, Issue 11, November 2009, Pages 1039-1064 (ELSIVIER).

[11] Simon King, “Measuring a decade of progress in Text-to-Speech”, Loquens (Journal), 1(1), January 2014, e006. eISSN 2386-2637.

[12] Zen, Heiga, Andrew Senior, and Mike Schuster. "Statistical parametric speech synthesis using deep neural networks." In 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7962-7966. IEEE, 2013.

[13] Kiruthiga S and Krishnamoorthy K, “Annotating Speech Corpus for Prosody Modeling in Indian Language Text to Speech Systems”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No. 1, 2012, ISSN (Online):1694-0814.

[14] Hemant A Patil, Tanvina B Patel, Nirmesh J Shah, Hardik B Sailor, Raghava Krishnan, G R Kasthuri, T Nagarajan, Lilly Christina, Naresh Kumar, Veera Raghavendra, S P Kishore, S R M Prasanna, Nagaraj Adiga, Sanasam Ranbir Singh, Konjengbam Anand, Pranaw Kumar, Bira Chandra Singh, S L Binil Kumar, T G Bhadran, T Sajini, Arup Saha, Tulika Basu, K Sreenivasa Rao, N P Narendra, Anil Kumar Sao, Rakesh Kumar, Pranhari Talukdar, Purnendu Acharyaa, Somnath Chandra, Swaran Lata, Hema A Murthy, “A Syllable-Based Framework for Unit Selection Synthesis in 13 Indian Languages”, in Oriental COCOSDA held jointly with Intl Conf on Asian Spoken Language Research and Evaluation, Nov 2013, pp. 1–8.

[15] Ramani, B., S.Lilly Christina, G Anushiya Rachel, V Sherlin Solomi, Mahesh Kumar Nandwana, Anusha Prakash, Aswin Shanmugam S, Raghava Krishnan, S P Kishore, K

Samudravijaya, P Vijayalakshmi, T Nagarajan and Hema A Murthy, “A Common Attribute based Unified HTS framework for Speech Synthesis in Indian Languages”, 8th ISCA Speech Synthesis Workshop, pp. 291-296, 2013, Spain.

[16] N. P. Narendra · K. Sreenivasa Rao, “Robust Voicing Detection and F0 Estimation for HMM-Based Speech Synthesis”, Circuits Syst Signal Process (2015) 34:2597–2619. © Springer Science.

[17] Zulfiqar Ali, Ghulam Muhammad, Mohammed F. Alhamid, “An Automatic Health Monitoring System for Patients Suffering From Voice Complications in Smart Cities”, DOI: 10.1109/ACCESS.2017.2680467,3900-3908 VOLUME 5, 2017

[18] Idor Svensson, Thomas Nordström, Emma Lindeblad, Stefan Gustafson, Marianne Björn, Christina Sand, Gunilla Almgren/Bäck & Staffan Nilsson (2021) Effects of assistive technology for students with reading and writing disabilities, Disability and Rehabilitation: Assistive Technology, 16:2, 196-208, DOI: 10.1080/17483107.2019.1646821.

[19] J. Liljencrants, “The OVE III speech synthesizer,” IEEE Transactions on Audio and Electroacoustics, vol. 16, no. 1, pp. 137–140, Mar. 1968.

[20] D. H. Klatt, “The Klattalk text-to-speech conversion system,” in Proc. ICASSP, 1982, pp. 1589–1592.

[21] J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni, From text to speech: The MITalk system. Cambridge University Press, 1987.

[22] Y. Sagisaka, N. Kaiki, N. Iwahashi, and K. Mimura, “ATR n-talk speech synthesis system,” in Proc. ICSLP, 1992.

[23] A. W. Black and P. Taylor, “CHATR: A generic speech synthesis system,” in Proc. COLING, 1994, pp. 983–986.

[24] G. Rosen, “Dynamic analog speech synthesizer,” The Journal of the Acoustical Society of America, vol. 30, no. 3, pp. 201–209, 1958.

[25] A. W. Black, “CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling,” in Proc. ICSLP, 2006, pp. 1762–1765.

[26] T. Koriyama, T. Nose, and T. Kobayashi, “Statistical parametric speech synthesis based on Gaussian process regression,” IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 2, pp. 173–183, Apr. 2014.

[27] K. Shichiri, A. Sawabe, T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Eigenvoices for HMM-based speech synthesis,” in Proc. ICSLP, 2002, pp. 1269–1272.

[28] H. Zen, N. Braunschweiler, S. Buchholz, M. J. Gales, K. Knill, S. Krstulovic, and J. Latorre, “Statistical parametric speech synthesis based on speaker and language factorization,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 6, pp. 1713–1724, 2012.

[29] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the training aspects of deep neural network (DNN) for parametric TTS synthesis,” in Proc. ICASSP, 2014, pp. 3829–3833.

[30] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, “TTS synthesis with bidirectional LSTM based recurrent neural networks,” in Proc. INTERSPEECH, 2014, pp. 1964–1968.

[31] H. Zen and H. Sak, “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis,” in Proc. ICASSP, 2015, pp. 4470–4474.

[32] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” arXiv preprint arXiv:1402.1128, 2014.

[33] Guo, H., Soong, F. K., He, L., & Xie, L. (2019). A new GAN-based end-to-end TTS training algorithm. In Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH (Vol. 2019-September, pp. 1288–1292).

[34] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous,

"Tacotron: A fully end-to-end text-to-speech synthesis model," CoRR, vol. abs/1703.10135, 2017. [Online]. Available: http://arxiv.org/abs/1703.10135

[35] J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings, 1996 IEEE International Conference on, vol. 1, May 1996, pp. 373–376 vol. 1.

[36] T. Styger ,E Keller, "Format Synthesis", Fundamental of Speech Synthesis and Speech Recognition; Basi concept,State of the Art and future challenges (PP.109-128).

[37] B. Sharma and S. R. M. Prasanna, "Improvement of syllable based TTS system in assamese using prosody modification," 2015 Annual IEEE India Conference (INDICON), 2015, pp. 1-6, doi: 10.1109/INDICON.2015. 7443698

[38] https://speech.meridian-one.co.uk/orpheus.html

[39] https://blindhelp.net/software/eloquence

[40] L.R.Rainer,"Applications of Voice processing to Telecommunications", Proc.IEE,Vol.82,PP.199-228,1994.

[41] P. Birkholz, B. Kroger, and C. Neuschaefer-Rube, "Model-based reproduction of articulatory trajectories for consonant-vowel sequences," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 5, pp. 1422–1433, July 2011.

[42] B.Kroger, "Minimal Rules for Articulatory Speech Synthesis", Proceedings of EUSIPCO92, pp:331-334, 1992.

[43] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M.Wester, "Speech production knowledge in automatic speech recognition," The Journal of the Acoustical Society of America, vol. 121, no. 2, pp. 723–742, Feb. 2007.

[44] A. Black and K. Lenzo, "Building voices in the festival speech synthesis system," 2000.

[45] F. J. Charpentier and M. Stella, “Diphone synthesis using an overlap- add technique for speech waveforms concatenation,” in Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP‟86, vol. 11. IEEE, 1986, pp. 2015–2018.

[46] C. Hamon, E. Mouline, and F. Charpentier, “A diphone synthesis system based on time-domain prosodic modifications of speech,” in Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on. IEEE, 1989, pp. 238–241.

[47] Zen, Heiga. "Acoustic Modeling in Statistical Parametric Speech Synthesis-From HMM to LSTM-RNN." (2015), Research at Google.

[48] Black, A.W., 2006. CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling. In: Proc. ICSLP, 1762–1765.

[49] T. Yoshimura, “Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems,” Ph.D. dissertation, Nagoya Institute of Technology, Nagoya, Japan, 2002.

[50] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in Proc. ICASSP, 2000, pp. 1315–1318.

[51] Ling, Zhen-Hua, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M. Meng, and Li Deng. "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends." IEEE Signal Processing Magazine 32, no. 3 (2015): 35-52.

[52] B. Li and H. Zen, “Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis,” in Proc. INTERSPEECH, 2016, pp. 2468–2472.

[53] Achanta, S, Gangashetty, S.V., 2017. Deep Elman Recurrent Neural Networks for Statistical Parametric Speech Synthesis. Speech Communications, Vol 93, pp 31-42.

[54] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, “Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices,” in Proc. INTERSPEECH, 2016.

[55] S. Pascual and A. Bonafonte, “Multi-output RNN-LSTM for multiple speaker speech synthesis and adaptation,” in Proc. EUSIPCO, 2016, pp. 2325–2329.

[56] Wu, Z. , King, S. , 2016. Investigating gated recurrent networks for speech synthesis. In: Proc. ICASSP, pp. 5140–5144.

[57] S. Achanta, T. Godambe, and S. V. Gangashetty, “An investigation of recurrent neural network architectures for statistical parametric speech synthesis,” in Proc. INTERSPEECH, 2015, pp. 2524–2528.

[58] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in Proc. ICML, 2013, pp. 1310–1318.

[59] Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555.

[60] Le, Q.V., Jaitly, N., Hinton, G.E., 2015. A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941.

[61] S. Achanta and S. V. Gangashetty, “Deep Elman recurrent neural networks for statistical parametric speech synthesis,” Speech Communication, vol. 93, pp. 31–42, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167639316303818

[62] S. Achanta, R. Banoth, A. Pandey, A. Vadapalli, and S. V. Gangashetty, “Contextual representation using recurrent neural network hidden state for statistical parametric speech synthesis,” in Proc. Ninth ISCA Speech Synthesis Workshop, 2016, pp. 187–192.

[63] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for

raw audio," CoRR, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[64] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in Proc. ICLR, 2017.

[65] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model," arXiv preprint, arXiv:1612.07837, 2016.

[66] Liu, Y., & Zheng, J. (2019). Es-Tacotron2: Multi-task Tacotron 2 with pre-trained estimated network for reducing the over-smoothness problem. Information (Switzerland), 10(4).

[67] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: 2000-Speaker Neural Text- toSpeech," in Proc. ICLR, 2018.

[68] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech" 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada. arXiv:1905.09263v5 [cs.CL] 20 Nov 2019.

[69] S. Thomas, M. N. Rao, H. A. Murthy, and C. Ramalingam, "Natural sounding tts based on syllable-like units," Energy, vol. 2, no. 4, p. 6, 2006.

[70] S. Tiomkin, D. Malah, S. Shechtman, and Z. Kons, "A hybrid text-to-speech system that combines concatenative and statistical synthesis units," IEEE Transactions on Audio, Speech, and Language Processing, vol. 19, no. 5, pp. 1278–1288, July 2011.

[71] J. Yamagishi, T. Nose, H. Zen, Z.-H. Ling, T. Toda, K. Tokuda, S. King, and S. Renals, "Robust speaker-adaptive HMM based text-to-speech synthesis," IEEE Trans. Audio Speech Lang. Process., vol. 17, no. 6, pp. 1208–1230, 2009.

[72] Z. Wu and S. King, "Improving trajectory modelling for DNN-based speech synthesis by using stacked bottleneck features and minimum generation error training,"

IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 7, pp. 1255–1265, July 2016.

[73] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157–166, 1994.

[74] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM- based speech synthesis,” in Proc. Eurospeech, 1999, pp. 2347–2350.

[75] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi, “Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR,” in Proc. ICASSP, 2001, pp. 805–808.

[76] T. Nose, J. Yamagishi, T. Masuko, and T. Kobayashi, “A style control technique for HMM-based expressive speech synthesis,” IEICE Trans. Inf. Syst., vol. E90-D, no. 9, pp. 1406–1413, 2007.

[77] A. Gutkin, X. Gonzalvo, S. Breuer, and P. Taylor, “Quantized HMMs for low footprint text-to-speech synthesis,” in Proc. Interspeech, 2010, pp. 837–840.

[78] Srikanth Ronanki, Siva Reddy, Bajibabu Bollepalli, Simon King, “DNN-based Speech Synthesis for Indian Languages from ASCII text” in 9th ISCA Speech Synthesis Workshop, pp. 187–192, 2016, https://doi.org/10.48550/arXiv.1608.05374.

[79] Achanta, S., Godambe, T., Gangashetty, S.V., An investigation of recurrent neural network architectures for statistical parametric speech synthesis. In: Proc. INTERSPEECH, pp. 2524–2528, 2015.

[80] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, “Large scale distributed deep networks,” in Proc. NIPS, 2012.

[81] K. Rao and B. Yegnanarayana, “Prosody modification using instants of significant excitation,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 14, no. 3, pp. 972–980, May 2006.

[82] E. Janse, S. Nooteboom, and H. Quene, “Word-level intelligibility of time-compressed speech: prosodic and segmental factors,” in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 41, pp. 273-529, Oct. 2003.

[83] S. Lee, H. D. Kim, and H. S. Kim, “Variable time-scale modification of speech using transient information," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 2, pp. 1319 - 1322, Apr. 1997.

[84] M. Covell, M. Withgott, and M. Slaney, “MACH1: nonuniform time-scale modification of speech," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 1, pp. 349 -352, Apr. 1998.

[85] E. Moulines and J. Laroche, “Non-parametric techniques for pitch-scale and time-scale modification of speech," Speech Commun., vol. 16, pp. 175-205,Feb. 1995.

[86] E. Moulines and W. Verhelst, “Time-domain and frequency-domain techniques for prosodic modification of speech," in Speech Coding and Synthesis (W. B. Kleijn and K. K. Paliwal, eds.), pp. 519-555, Elsevier Science Publishers, 1995.

[87] R. Crochiere, “A weighted overlap-add method of short-time Fourier analysis/synthesis," IEEE Trans. Acoust. Speech, Signal Processing, vol. 28, pp. 99-102, Feb. 1980.

[88] S. Roucos and A. Wilgus, “High quality time-scale modification of speech," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 10, pp. 493- 496, Apr. 1985.

[89] W. Verhelst and M. Roelands, “An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. 2, pp. 554-557, Apr. 1993.

[90] J. Laroche, “Autocorrelation method for high-quality time/pitch-scaling," in Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoust., pp. 131-134, Oct. 1993.

[91] E. Moulines and F. Charpentier, “Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones,” Speech communication, vol. 9, no. 5, pp. 453–467, 1990.

[92] J. Laroche and M. Dolson, “Improved phase vocoder time-scale modification of audio," IEEE Trans. Speech Audio Processing, vol. 7, pp. 323-332, May 1999.

[93] M. R. Portnoff, “Time-scale modification of speech based on shorttime fourier analysis,” Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 29, no. 3, pp. 374–390, Jun 1981.

[94] E. George and M. J. T. Smith, “Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model,” Speech and Audio Processing, IEEE Transactions on, vol. 5, no. 5, pp. 389–406, Sep 1997.

[95] D. W. Griffin and J. S. Lim, “Multiband excitation vocoder," IEEE Trans. Acoust. Speech, Signal Processing, vol. 36, pp. 1223-1235, Aug. 1988.

[96] Y. Shen, J. Jia, and L. Cai, “Detection on psola-modified voices by seeking out duplicated fragments,” in Systems and Informatics (ICSAI), 2012 International Conference on. IEEE, 2012, pp. 2177–2182.

[97] G. Kubin and W. B. Kleijn, “Time-scale modification of speech based on a nonlinear oscillator model," in Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing, vol. i, pp. I/453 - I/456, Apr. 1994.

[98] N. Narendra, K. Rao, K. Ghosh, R. Vempada, and S. Maity, “Development of syllable-based text to speech synthesis system in Bengali,” International Journal of Speech Technology, vol. 14, no. 3, pp. 167–181, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10772-011-9094-4

[99] M. Vinodh, A. Bellur, K. Narayan, D. Thakare, A. Susan, N. Suthakar, and H. Murthy, “Using polysyllabic units for text to speech synthesis in indian languages,” in Communications (NCC), 2010 National Conference on, Jan 2010, pp. 1–5.

[100] Y. Venugopalakrishna, M. Vinodh, H. A. Murthy, and C. Ramalingam, “Methods for improving the quality of syllable based speech synthesis,” in SLT, 2008, pp. 29–32.

[101] Kishore Prahallad, E. Naresh Kumar, Venkatesh Keri, S. Rajendran and Alan W Black "The IIIT-H Indic Speech Databases", in Proceedings of Interspeech 2012, Portland, Oregon, USA.

[102] K. Murty and B. Yegnanarayana, “Epoch extraction from speech signals,” Audio, Speech, and Language Processing, IEEE Transactions on, vol. 16, no. 8, pp. 1602–1613, Nov 2008.

[103] J. Kominek and A. W. Black, “The CMU ARCTIC speech databases,” in Proc. 5th ISCA Speech Synthesis Workshop, 2004, pp. 223–224.

[104] S.-J. Kim, J.-J. Kim, and M.-S. Hahn, “HMM-based Korean speech synthesis system for hand-held devices,” IEEE Trans. Consum. Electron., vol. 52, no. 4, pp. 1384–1390, 2006.

[105] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257–285,1989.

[106] J. Dines and S. Sridharan, “Trainable speech synthesis with trended hidden Markov models,” in Proc. ICASSP, 2001, pp. 833–837.

[107] I. Bulyko, M. Ostendorf, and J. Bilmes, “Robust splicing costs and efficient search with BMM models for concatenative speech synthesis,” in Proc. ICASSP, 2002, pp. 461–464.

[108] H. Zen, K. Tokuda, and T. Kitamura, “Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic features,” Comput. Speech Lang., vol. 21, no. 1, pp. 153–173, 2007.

[109] J.W. Sun, F. Ding, and Y.H. Wu, “Polynomial segment model based statistical parametric speech synthesis system,” in Proc. ICASSP, 2009, pp. 4021–4024.

[110] C. Quillen, “Kalman filter based speech synthesis,” in Proc. ICASSP, 2010, pp. 4618–4621.

[111] M. Shannon, H. Zen, and W. Byrne, “Autoregressive models for statistical parametric speech synthesis,” IEEE Trans. Acoust. Speech Lang. Process., vol. 21, no. 3, pp. 587–597, 2013.

[112] H. Zen, M. Gales, Y. Nankaku, and K. Tokuda, “Product of experts for statistical parametric speech synthesis,” IEEE Trans. Audio Speech Lang. Process., vol. 20, no. 3, pp. 794–805, 2012.

[113] T. Koriyama, T. Nose, and T. Kobayashi, “Statistical parametric speech synthesis based on Gaussian process regression,” IEEE Journal of Selected Topics in Signal Process., vol. 8, no. 2, pp. 173–183, 2014.

[114] M.-Q. Cai, Z.-H. Ling, and L.-R. Dai, “Statistical parametric speech synthesis using a hidden trajectory model,” Speech Commun., vol. 72, pp. 149–159, 2015.

[115] S. King, “A reading list of recent advances in speech synthesis,” in Proc. ICPhS, 2015.

[116] K. Yu, H. Zen, F. Mairesse, and S. Young, “Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis,” Speech Commn., vol. 53, no. 6, pp. 914–923, 2011.

[117] Z.-H. Ling, L. Deng, and D. Yu, “Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis,” IEEE Trans. Acoust. Speech Lang. Process., vol. 21, no. 10, pp. 2129–2139, 2013.

[118] Y. Fan, Y. Qian, F. Soong, and L. He, “Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis,” in Proc. ICASSP, 2015, pp. 4475–4479.

[119] Y. Bengio, “Learning deep architectures for AI,” Foundations and Trends in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.

[120] D. Yu and L. Deng, “Deep learning and its applications to signal and information processing,” IEEE Signal Process. Magazine, vol. 28, no. 1, pp. 145–154, 2011.

[121] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[122] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, “Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural network,” in Proc. INTERSPEECH, 2014, pp. 2268–2272.

[123] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J., 2015. LSTM: a search space odyssey. arXiv preprint arXiv:1503.04069.

[124] N. Schinkel-Bielefeld, "Audio Quality Evaluation in MUSHRA Tests–Influences between Loop Setting and a Listeners’ Ratings," Paper 9779, (2017) May.

[125] Y. Fan, Y. Qian, F. K. Soong, and L. He, “Sequence generation error (SGE) minimization based deep neural networks training for text-to-speech synthesis,” in Proc. INTERSPEECH, 2015, pp. 864–868.

[126] O. Watts, Z. Wu, and S. King, “Sentence-level control vectors for deep neural network speech synthesis,” in Proc. INTERSPEECH, 2015, pp. 2217–2221.

[127] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in Proc. ICML, 2013, pp. 1139–1147.

[128] R. J.Williams and J. Peng, “An efficient gradient-based algorithm for on-line training of recurrent network trajectories,” Neural Computation, vol. 2, pp. 490–501, 1990.

[129] J. Martens, “Deep learning via Hessian-free optimization,” in Proc. ICML, 2010, pp. 735–742.

[130] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances in optimizing recurrent networks,” in Proc. ICASSP, 2013, pp. 8624–8628.

[131] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, Natural TTS Synthesis by vonditioning WAVENET on Mel Spectrogram Predictions Google, Inc., 2University of California, Berkeley, arXiv:1712.05884v2 [cs.CL] 16 Feb 2018

[132] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[133] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[134] D. Krueger, T. Maharaj, J. Kram´ar, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, H. Larochelle, A. Courville, et al., "Zoneout: Regularizing RNNs by randomly preserving hidden activations," in Proc. ICLR, 2017.

[135] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," arXiv preprint arXiv:1506.07503, 2015.

[136] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2):236–243, 1984.

List of Publications

International Journals:

[1] **Ravi Bolimera** and Kishore Kumar T, **"Prosody Modeling for Improvement in Telugu TTS System"** NeuroQuantology, Volume 20, Issue 1, Page 579-584, Jan 2022. doi: 10.14704/nq.2022.20.1.NQ22335. **(Scopus)**

[2] **Ravi Bolimera** and Kishore Kumar T, **"Artificial Intelligence and IoT-based Healthcare System using TTS Assistive Technology for Voice Disordered People"** NeuroQuantology, Volume 20, Issue 10, Page 3649-3655, 2022. doi: 10.14704/nq.2022.20.10.NQ55353. **(Scopus)**

[3] **Ravi Bolimera** and Kishore Kumar T, **"Development of High Quality Telugu Speech Synthesis system using Deep Elman Recurrent Neural Networks to Assist Children Education"** Volume 14, No 1, PP 3579-3587, 2022: International Journal of Early Childhood Special Education. DOI: 10.9756/INT-JECSE/V14I1.433 ISSN: 1308-5581 **(ESCI/Scopus)**

[4] **Ravi Bolimera** and Kishore Kumar T, "**Development and Evaluation of Neural based Speech Synthesis Systems for Telugu Language"** submitted to **Traitement du Signal (SCIE) (Under Review)**