COGNITIVE MODELING OF ACCENTED SPEECH IN MALAYALAM: EXPLORING THE IMPACT OF ACOUSTIC SIGNAL PROCESSING AND DEEP LEARNING TECHNIQUES

A Thesis Submitted to the University of Calicut
in partial fulfilment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

Under the Faculty of Science

By

RIZWANA KALLOORAVI THANDIL

Under the guidance of

Dr. MOHAMED BASHEER K.P
Associate Professor of Computer Science
Sullamussalam Science College, Areekode

P.G & RESEARCH DEPARTMENT OF COMPUTER SCIENCE
Sullamussalam Science College, Areekode - 673639
(Affiliated to the University of Calicut)
Malappuram Dist., Kerala, India

May 2024

DECLARATION

I, Rizwana Kallooravi Thandil, hereby declare that this thesis entitled "**Cognitive Modeling of Accented Speech in Malayalam: Exploring the Impact of Acoustic Signal Processing and Deep Learning Techniques"** is based on the original work done by me under the supervision of Dr. Mohamed Basheer K.P., Assistant Professor, PG & Research Department of Computer Science, Sullamussalam Science College, Areekode, Kerala.

I confirm that,

- The work presented in this thesis has not been submitted previously for the award of any degree either to this University or to any other University or Institution.
- I have followed the guiding principles given by the University in organizing the thesis.
- Whenever I have used materials (theoretical analysis, data, figures, and text) from other sources, I have given due credit to them by citing them in the thesis and giving their particulars in the references.

Rizwana Kallooravi Thandil

Areekode
28 May 2024

SULLAMUSSALAM SCIENCE COLLEGE
AREACODE

Phone: 0483-2850700
Website: www.sscollege.ac.in
Email: mail@sscollege.ac.in

Aided College | Affiliated to the University of Calicut | Re-accredited by NAAC at A Grade

Ref: Date:

CERTIFICATE

This is to certify that the thesis entitled **"Cognitive Modeling of Accented Speech in Malayalam: Exploring the Impact of Acoustic Signal Processing and Deep Learning Techniques"**, submitted by **Mrs. Rizwana Kallooravi Thandil**, to the University of Calicut, for the partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy (Ph.D.) in Computer Science, is a bonafide research work done by Mrs. Rizwana Kallooravi Thandil under my supervision and guidance in the PG & Research Department of Computer Science, Sullamussalam Science College, Areekode, Malappuram, Kerala. The content embodied in this thesis, in full or in parts, have not been submitted to any other University or Institute for the award of any degree.

The thesis is revised as per the modifications and recommendations reported by the adjudicators. Soft copy attached is the same as that of the revised copy. The thesis is submitted as such to the University of Calicut with reference to the letter number No. 4134/RESEARCH-C-ASST-1/2024/Admn Dated 21.05.2024.

Dr. Mohamed Basheer K.P
Associate Professor
PG & Research Department of Computer Science
Sullamussalam Science College, Areekode, Kerala, India

Areekode
28 May 2024

UNIVERSITY OF CALICUT

CERTIFICATE ON PLAGIARISM CHECK

1.	Name of the research scholar	Rizwana Kallooravi Thandil		
2.	Title of thesis/dissertation	COGNITIVE MODELING OF ACCENTED SPEECH IN MALAYALAM : EXPLORING THE IMPACT OF ACOUSTIC SIGNAL PROCESSING AND DEEP LEARNING TECHNIQUES		
3.	Name of the supervisor	Dr. MOHAMED BASHEER K.P		
4.	Department/Institution	P.G & RESEARCH DEPARTMENT OF COMPUTER SCIENCE Sullamussalam Science College, Areekode, 673639		
5.	Similar content (%)identified	**Introduction/ Review of literature**	**Materials and Methods**	**Result/ Discussion/Summary/ Conclusion**
		4%	**5%**	**3%**
	Acceptable maximum limit (%)	10	10	10
6.	Software used	iThenticate		
7.	Date of verification			

**Report on plagiarism check, specifying included/excluded items with % of similarity to be attached.*

Dr. Nasirudheen. T
Assistant Librarian
University of Calicut, Kerala.

Checked by (with name, designation & Signature)

Name and signature of the Researcher Rizwana Kallooravi Thandil

Name & Signature of the Supervisor

Dr. MOHAMED BASHEER K.P.
ASST. PROFESSOR IN COMPUTER SCIENCE
SULLAMUSSALAM SCIENCE COLLEGE
AREAKODE, UGRAPURAM P.O.
MALAPPURAM DT.- 673 639

The Doctoral Committee* has verified the report on plagiarism check with the contents of the thesis, as summarized above and appropriate measures have been taken to ensure originality of the Research accomplished herein.

Name & Signature of the HoD/HoI (Chairperson of the Doctoral Committee)

PRINCIPAL
SULLAMUSSALAM SCIENCE COLLEGE
AREEKODE, UGRAPURAM (PO)
MALAPPURAM(DT) PIN:673639

** In case of languages like Malayalam, Tamil, etc. on which no software is available for plagiarism check, a manual check shall be made by the Doctoral Committee, for which an additional certificate has to be attached*

Acknowledgments

I would like to express my deepest gratitude to all those who have been instrumental in the realization of this research endeavor. This journey has been both challenging and rewarding, and I am profoundly thankful for the unwavering support I have received throughout. This endeavor has been a collective effort, and I extend my heartfelt appreciation to everyone who has stood by me throughout this rigorous yet rewarding process.

First and foremost, I express my deepest gratitude to my advisor Dr. Mohamed Basheer K.P, Associate Professor in the P.G and Research Department of Computer Science at Sullamussalam Science College, whose unwavering support, guidance, and scholarly insights have been instrumental in shaping the direction of my research. His mentorship has been invaluable, and I am truly fortunate to have had the privilege of working under his supervision. Dr. Basheer emerged not only as a mentor but as a beacon of support and the epitome of positivity whose confidence in my research pursuits infused me with a deep sense of assurance. His meticulous, thorough reviews were not just constructive but immensely contributive, shaping not only the research papers but also sculpting the very fabric of this comprehensive thesis. The profound understanding, inexhaustible patience, boundless kindness, and the freedom to explore under his mentorship transformed my research tenure into an indelibly memorable and intellectually enriching experience.

I extend my deepest gratitude to Dr. Muhamed Ilyas P, our esteemed principal, whose unwavering support, and encouragement have been the cornerstone of my academic journey. His visionary leadership has created a culture of academic excellence within our institution, inspiring me to strive for the highest standards in my pursuit of knowledge. I am grateful for the opportunities provided under Dr. Ilyas's leadership, which have allowed me to engage in meaningful research and contribute to the broader academic discourse. His mentorship has been a source of inspiration, motivating me to overcome challenges and pursue excellence in my work. I express my heartfelt gratitude to Prof. N.V. Abdul Rahman, our esteemed manager, and the entire management team for their support and visionary leadership throughout the course of my research journey. I extend my gratitude to the management team for their strategic vision and commitment to providing resources that facilitated the successful execution of my research.

I extend my profound gratitude to the esteemed members of my research advisory committee for their invaluable contributions to my Ph.D. journey. Dr. Lajish VL, Associate Professor & Head, Dept. of Computer Science, University of Calicut, Dr. Vasudevan, Dept. of Library Science, University of Calicut, Dr. Binu P Chacko, Principal, Prajyothi Nikethan College Puthukkad, Dr. Shameem Kappan, Head, PG & Research Department of Computer Science, SS College, for their thoughtful insights and constructive critiques. I recognize the profound impact they have had on my academic and intellectual development. Their contributions have been invaluable, and I am truly grateful for the privilege of having them as members of my research advisory committee. I express my heartfelt gratitude to my colleague Mr. Muneer V.K. for the support and shared expertise that was instrumental in navigating the complexities of this journey. I am deeply thankful for the positive impact he has had on this academic endeavor.

My heartfelt gratitude goes to my beloved parents for the immeasurable contributions they have made to my academic and personal journey. Their unwavering support, boundless love, and commitment to providing the right education have been the bedrock upon which my achievements stand. To my father and mother, whose sacrifices and dedication have shaped the person I am today, I offer my deepest gratitude. Your belief in the power of education and your unyielding support have been the cornerstones of my success. I am privileged to be your child and carry forward the values you have instilled in me. In moments of uncertainty and despair, my mother's comforting presence has been the solace that propelled me forward. Her boundless love and encouragement have been the foundation upon which I built my aspirations. I am truly blessed to have her as the guiding force in my life.

I extend my heartfelt gratitude to my sisters, in-laws, husband, and daughters [Rida and Zanha Yara], for their unwavering support, encouragement, patience, understanding, and countless sacrifices throughout the challenging journey of my research. Their presence has been my anchor, and I am profoundly thankful for the strength they provided during moments of stress and dedication. The sacrifices made by my family, both seen and unseen, have been instrumental in my ability to navigate the challenges of this work.

I express my sincere gratitude to my colleagues, fellow scholars and friends for their continuous support and encouragement throughout my academic journey. I express my deepest gratitude to those who stood by me with genuine and heartfelt support,

offering unwavering encouragement and boundless affection that has been the driving force not only behind my academic achievements but through all phases of my life. I am profoundly thankful for the enduring joy and fulfillment that your love brings into every aspect of my journey.

With utmost humility, I extend my deepest gratitude to the Almighty for the boundless blessings and opportunities bestowed upon me.

Rizwana Kallooravi Thandil

Dedicated

To My Ever-Loving Father
and
My Ever-Loving Mother

[For their Unconditional Love, Support, Motivation, Endless Sacrifices, Understanding, and Resilience that have Shaped the Person I am Today.]

ABSTRACT

Accented Automatic Speech Recognition (AASR) is the ability of a system to recognize accented speech inputs. It poses a unique challenge, particularly for languages with limited available datasets. In this research, a comprehensive exploration of machine learning and deep learning along with feature engineering techniques was conducted to advance the understanding of accented speech recognition.

The research is completed in several phases of experimental studies. The journey begins with an extensive literature review and finding the dominating gap in the domain of AASR for Malayalam. The unavailability of benchmark dataset in accented Malayalam and scarcity of previous study in literature hindered this research. To address the scarcity of relevant datasets, eight distinct sets of accented data were carefully constructed. Additionally, a spectrogram dataset was developed to facilitate a comprehensive study. The research investigates various feature extraction techniques and model architectures, exploring the impact of different feature combinations on accented speech recognition.

Each dataset is characterized by a diverse range of key properties essential for robust speech recognition systems. The datasets exhibit a wide spectrum of accents from varied regions and demographic groups. Efforts were made to maintain balanced representation across genders, ages, and socio-economic backgrounds, thereby reducing potential biases. The recordings for some of the datasets were conducted in natural settings to authentically capture variations in accent and pronunciation. These datasets are annotated with word and sentence level transcriptions (depending on the type of audio signal) and the district of the specific accent providing valuable insights into speaker details and recording conditions. To evaluate system robustness, recordings were obtained under various noise conditions, spanning from quiet environments to bustling public spaces.

The isolation of words from speech signals (AMSC-1 to AMSC-6, AMESC, AMDDHS) involved a complex approach to ensure precision and consistency. In the beginning the start and end points of words are marked through manual segmentation. This was complemented by the utilization of forced aligners, which provided precise alignment of words within the speech signals. Signal processing techniques, including silence detection and spectral analysis, further facilitated the identification of word boundaries with accuracy. To validate the integrity of the isolated words, they were cross verified with the original scripts, ensuring correctness and reliability. Volume and length normalization techniques were

applied to maintain consistency across the datasets. These methodologies collectively ensured the creation of robust datasets essential for the development and evaluation of accented speech recognition systems in Malayalam.

The acoustic signal processing methods encompass a wide array, including Mel Frequency Cepstral Coefficients (MFCC), Short Term Fourier Transformation (STFT), Mel Spectrogram, Tempogram, Zero Crossing Rate (ZCR), Root Mean Square Value (RMS), Tonnetz & Polyfeatures, and Harmonic Mean Ratio (HMR). The study involves a detailed analysis of the effectiveness of the feature vectors extracted in enhancing accented speech recognition.

In this research, a comprehensive exploration of machine learning and deep learning techniques was conducted to advance the understanding of accented speech recognition. Accented Speech Recognition models were implemented using a diverse set of classifiers, including Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Bidirectional LSTM (BiLSTM), Multilayer Perceptron (MLP), K Nearest Neighbor (KNN), Stochastic Gradient Descent (SGD), Support Vector Machine (SVM), ensembled models, decision tree, random forest classifiers etc. Experiments utilizing spectrograms and CNN architectures were conducted to further enhance model performance.

The phases of study involved constructing unified accented ASR models for Malayalam employing Autoencoders and self-supervised learning. The models were constructed on the compressed and uncompressed representation of the feature vectors. The outcome of the experiment was promising when ensembled with the machine learning approaches.

The research extended beyond traditional recognition models to include accented emotional clustering experiments for Malayalam accented data. Unsupervised learning techniques were adopted for clustering the accented Malayalam based on the seven emotions [happy, sad, anger, disgust, fear, neutral, surprise].

The study examined hate speech detection for accented Malayalam data using deep learning techniques. Accented speech dataset containing hate speech has been constructed from publicly available online platforms. Explorations were made using feature engineering and deep learning techniques for conducting this study.

Additionally, extensive exploration of the AASR for Malayalam was constructed for a more enhanced accented dataset. A detailed study on different architectures including hybrid neural network architectures combining CNN and LSTM, as well

as multidimensional CNNs, were explored in constructing accented speech recognition models.

The evaluation of these models employed a range of techniques, including accuracy, F-score, recall, word error rate, and match error rate, providing a comprehensive assessment of their performance. Machine learning approaches played a crucial role in the model construction process, adding depth and versatility to the research. The results of the experiments at different phases are evaluated and discussed to find the key contributions of this research.

Keywords: Accented Speech Recognition, Malayalam ASR, Speech Feature Extraction, LSTM-RNN, CNN, Multi-Dimensional CNN, Autoencoders, and Machine Learning.

സംഗ്രഹം

ആക്സന്റഡ് സ്പീച്ച് റെക്കഗ്നിഷനെക്കുറിച്ച് മനസ്സിലാക്കുന്നതിനായി മെഷീൻ ലേണിംഗിന്റെയും ഡീപ് ലേണിംഗ് ടെക്നിക്കുകളുടെയും സമഗ്രമായ ഒരു പഠനമാണിത്. പരിമിതമായ ഡാറ്റാസെറ്റുകൾ മാത്രം ലഭ്യമായ ഭാഷകൾക്ക് ഉച്ചാരണ സംഭാഷണം തിരിച്ചറിയുക എന്നത് ശ്രമകരമായ ജോലിയാണ്. പ്രത്യേകിച്ച് മലയാളം പോലെയുള്ള ദ്രവീഡിയൻ ഭാഷകളിൽ.

ഭാഷയുമായി ബന്ധപ്പെട്ട സാങ്കേതിക പഠനങ്ങളിൽ ഏറ്റവും പ്രസക്തമായ കാര്യം ഡാറ്റസെറ്റുകളാണ്. ഡാറ്റാസെറ്റുകളുടെ ദൗർലഭ്യം പരിഹരിക്കുന്നതിനായി, എട്ട് വ്യത്യസ്തമായ ആക്സന്റഡ് ഡാറ്റാ സെറ്റുകൾ വികസിപ്പിച്ചെടുത്തിട്ടുണ്ട്. കൂടാതെ, സമഗ്രമായ പഠനം സുഗമമാക്കുന്നതിന് ഒരു സ്പെക്ട്രോഗ്രാം ഡാറ്റാസെറ്റും വികസിപ്പിച്ചെടുത്തു. വിവിധ ഫീച്ചർ എക്സ്ട്രാക്ഷൻ ടെക്നിക്കുകളും മോഡൽ ആർക്കിടെക്ചറുകളും ഗവേഷണ വിധേയമാക്കി. ഉച്ചാരണത്തിലുള്ള സംഭാഷണ തിരിച്ചറിയലിൽ വ്യത്യസ്ത ഫീച്ചർ കോമ്പിനേഷനുകളുടെ സ്വാധീനവും പഠന വിധേയമാക്കി.

ഫീച്ചർ എക്സ്ട്രാക്ഷൻ രീതികളിൽ മെൽ ഫ്രീക്വൻസി സെപ്സ്ട്രൽ കോഫിഫിഷ്യന്റ്സ് (എംഎഫ്സിസി), ഷോർട്ട് ടേം ഫ്യൂറിയർ ട്രാൻസ്ഫോമേഷൻ (എസ്ടിഎഫ്ടി), മെൽ സ്പെക്ട്രോഗ്രാം, ടെംപോഗ്രാം, സീറോ ക്രോസിംഗ് റേറ്റ് (ഇസഡ്സിആർ), റൂട്ട് മീൻ സ്ക്വയർ വാല്യൂ (ആർഎംഎസ്), ടോണറ്റ്സ് *&* പോളിഫീച്ചറുകൾ, ഹാർമോണിക് ശരാശരി അനുപാതം (***HMR)*** എന്നിവ ഉൾപ്പെടുന്നു. ഉച്ചാരണത്തിലുള്ള സംഭാഷണ തിരിച്ചറിയൽ വർദ്ധിപ്പിക്കുന്നതിൽ ഇത്തരം രീതികളുടെ ഫലപ്രാപ്തിയെക്കുറിച്ചുള്ള വിശദമായ വിശകലനം പഠനത്തിൽ ഉൾപ്പെടുന്നു.

എൽ എസ് ടി എം ആർഎൻഎൻ, സിഎൻഎൻ, എംഎൽപി, കെഎൻഎൻ, എസ്ജിഡി, എസ് വി എം, എൻസെംബിൾഡ് മോഡലുകൾ, റാൻഡം ഫോറസ്റ്റ് ക്ലാസിഫയറുകൾ എന്നിവയുൾപ്പെടെ വൈവിധ്യമാർന്ന ക്ലാസിഫയറുകൾ ഉപയോഗിച്ചാണ് ആക്സന്റഡ് സ്പീച്ച് റെക്കഗ്നിഷൻ മോഡലുകൾ നടപ്പിലാക്കിയത്.

മോഡൽ പ്രകടനം കൂടുതൽ മെച്ചപ്പെടുത്തുന്നതിനായി സ്പെക്ട്രോഗ്രാമുകളും സിഎൻഎൻ ആർക്കിടെക്ചറുകളും ഉപയോഗിച്ചുള്ള പരീക്ഷണങ്ങൾ നടത്തി. ഓരോ രീതിയുടെയും വിദ്യകളുടെയും പരീക്ഷണവും അതിൽ നിന്നും കിട്ടിയ പരീക്ഷണഫലവും റിസൾട്ടുകളും തമ്മിലുള്ള താരതമ്യ പഠനവും ഗവേഷണത്തിന്റെ ഭാഗമായി നടത്തി.

മലയാളം ഓട്ടോഎൻകോഡറുകൾ ഉപയോഗിക്കുന്നതിനും സെൽഫ് സൂപ്പർവൈസ്ഡ് പഠനത്തിനുമായി ഏകീകൃത ആക്സന്റഡ് എഎസ്ആർ മോഡലുകൾ നിർമ്മിക്കുന്നത് പഠനത്തിന്റെ

ഘട്ടങ്ങളിൽ ഉൾപ്പെടുന്നു. ഫീച്ചർ വെക്റ്ററുകളുടെ കംപ്രസ് ചെയ്തതും കംപ്രസ് ചെയ്യാത്തതുമായ പ്രാതിനിധ്യത്തിലാണ് മോഡലുകൾ നിർമ്മിച്ചിരിക്കുന്നത്. മെഷീൻ ലേണിംഗ് സമീപനങ്ങളുമായി സമന്വയിപ്പിച്ചപ്പോൾ പരീക്ഷണത്തിന്റെ ഫലം പ്രതീക്ഷ നൽകുന്നതായിരുന്നു.

പരമ്പരാഗത തിരിച്ചറിയൽ മോഡലുകൾക്കപ്പുറം മലയാളം ആക്ലന്റഡ് ഡാറ്റയ്ക്കായി ആക്ലന്റഡ് ഇമോഷണൽ ക്ലസ്റ്ററിംഗ് പരീക്ഷണങ്ങൾ ഉൾപ്പെടുത്തുന്നതിനായി ഗവേഷണം വ്യാപിച്ചു. ഏഴ് വികാരങ്ങൾ [സന്തോഷം, സങ്കടം, ദേഷ്യം, വെറുപ്പ്, ഭയം, നിഷ്പക്ഷത, ആശ്ചര്യം] അടിസ്ഥാനമാക്കിയുള്ള മലയാളം ക്ലസ്റ്ററിങ്ങിനായി അൺസൂപ്പർവൈസ്ഡ് പഠന വിദ്യകൾ സ്വീകരിച്ചു.

ഡീപ് ലേണിംഗ് ടെക്നിക്കുകൾ ഉപയോഗിച്ച് ഉച്ചാരണമുള്ള മലയാളം ഡാറ്റയ്ക്കായി വിദ്വേഷ സംഭാഷണം കണ്ടെത്തുന്നത് പഠനം പരിശോധിച്ചു. പൊതുവായി ലഭ്യമായ ഓൺലൈൻ പ്ലാറ്റ്ഫോമുകളിൽ നിന്ന് വിദ്വേഷ സംഭാഷണം അടങ്ങിയ ഉച്ചാരണ സംഭാഷണ ഡാറ്റാസെറ്റ് നിർമ്മിച്ചിട്ടുണ്ട്. ഈ പഠനം നടത്താൻ ഫീച്ചർ എഞ്ചിനീയറിംഗും ഡീപ് ലേണിംഗ് ടെക്നിക്കുകളും ഉപയോഗിച്ചാണ് പര്യവേക്ഷണങ്ങൾ നടത്തിയത്.

കൂടാതെ, മലയാളത്തിനായുള്ള ***AASR***-ന്റെ വിപുലമായ പര്യവേക്ഷണം കൂടുതൽ മെച്ചപ്പെടുത്തിയ ആക്ലന്റഡ് ഡാറ്റാസെറ്റിനായി നിർമ്മിച്ചു. ***CNN, LSTM*** എന്നിവ സംയോജിപ്പിക്കുന്ന ഹൈബ്രിഡ് ന്യൂറൽ നെറ്റ്വർക്ക് ആർക്കിടെക്ചറുകളും മൾട്ടിഡൈമൻഷണൽ ***CNN***-കളും ഉൾപ്പെടെ വിവിധ ആർക്കിടെക്ചറുകളെക്കുറിച്ചുള്ള വിശദമായ പഠനം, ഉച്ചാരണമുള്ള സംഭാഷണ തിരിച്ചറിയൽ മോഡലുകൾ നിർമ്മിക്കുന്നതിൽ പര്യവേക്ഷണം ചെയ്യപ്പെട്ടു.

ഈ മോഡലുകളുടെ മൂല്യനിർണ്ണയത്തിൽ നിരവധി സാങ്കേതിക വിദ്യകൾ ഉപയോഗിച്ചു അവയുടെ പ്രകടനത്തിന്റെ സമഗ്രമായ വിലയിരുത്തൽ നൽകുന്നു. മെഷീൻ ലേണിംഗ് സമീപനങ്ങൾ മോഡൽ നിർമ്മാണ പ്രക്രിയയിൽ നിർണായക പങ്ക് വഹിച്ചു ഗവേഷണത്തിന് ആഴവും വൈവിധ്യവും നൽകുന്നു. ഈ ഗവേഷണത്തിന്റെ പ്രധാന സംഭാവനകൾ കണ്ടെത്താൻ വിവിധ ഘട്ടങ്ങളിലെ പരീക്ഷണങ്ങളുടെ ഫലങ്ങൾ വിലയിരുത്തുകയും ചർച്ച ചെയ്യുകയും ചെയ്യുന്നു.

Table of Contents

1. Introduction 1
   1.1 Background and Context 1
   1.2 Relevance of the Study 4
   1.3 Scope of the Research 8
   1.4 Limitations of the Research 9
   1.5 Research Gap 9
   1.6 Objectives 10
   1.7 Feature Extraction Methods 14
   1.8 Approaches for Constructing AASR Models 15
   1.9 Comprehensive Evaluation 19
   1.10 Organization of the Thesis 20
   1.11 Conclusion 25
2. Literature Review 27
   2.1 Introduction 27
   2.2 AASR in Literature 27
   2.3 Advances in AASR Across Different Languages 37
   2.4 AASR using Autoencoders 41
   2.5 AASR using ML and DL Approaches 44
   2.6 Accent-Neutral ASR 49
   2.7 Accent-Aware ASR 51
   2.8 Accent Unaware ASR 52
   2.9 Strategies for Data Set Preparation 54
   2.10 Generation of Spectrograms 57
   2.11 AASR of Malayalam Isolated Words 58
   2.12 Ensemble Approaches for AASR 60
   2.13 AASR for Multisyllabic Words 61
   2.14 Fusion of Self-Supervised Learning, ML models and Autoencoders 62

2.15 Clustering Methods for Emotion Classification 66
2.16 ASR for Detecting Hate Speech 71
2.17 Conclusion 72
3. Research Methodology 74
3.1 Introduction 74
3.2 Methodology 74
3.3 Multifaceted Exploration of AASR for Malayalam 74
3.4 Conclusion 79
4. Crafting Comprehensive Datasets for Malayalam AASR 81
4.1 Introduction 81
4.2 Challenges in Constructing the Dataset 81
4.3 Challenges in Constructing Accented Dataset 83
4.4 Strategies for Data Collection 84
4.5 Generating a Comprehensive Dataset 89
4.6 Phases of Data Collection 90
4.7 Conclusion 107
5. AASR of Malayalam Isolated Words using LSTM-RNN 109
5.1 Introduction 109
5.2 Methodology 109
5.3 Speech Signal Processing and Feature Vectorization 112
5.4 AASR using LSTM-RNN 114
5.5 Performance Evaluation 119
5.6 Conclusion 120
6. AASR with Deep-CNN, LSTM-RNN, and Machine Learning Approaches 121
6.1 Introduction 121
6.2 Methodology 122
6.3 AASR Model Construction 132
6.4 Performance Evaluation 144
6.5 Conclusion 145
7. End-to-End Unified AASR -A Low Resourced Context 146

7.1 Introduction 146
7.2 Methodology 146
7.3 Dataset Construction 148
7.4 Feature Engineering 148
7.5 Building the Accented ASR System 152
7.6 Performance Evaluation for AASR with AMSC-3 153
7.7 Performance Evaluation for AASR with AMSC-4 164
7.8 Conclusion 172
8. Spectral and Influential Features for Unified AASR in Malayalam 173
8.1 Introduction 173
8.2 Methodology 173
8.3 Dataset 175
8.4 Feature Extraction 177
8.5 AASR Model Construction 181
8.6 Performance Evaluation 187
8.7 Conclusion 189
9. A Feature-Based Investigation of LSTM-RNN and ML Approaches 191
9.1 Introduction 191
9.2 Methodology 192
9.3 Dataset Preparation 193
9.4 Feature Engineering 193
9.5 Accented ASR Model 194
9.6 Conclusion 201
10. Deep Neural Networks and Attention Mechanisms for AASR in Malayalam 203
10.1 Introduction 203
10.2 Methodology 203
10.3 Accented Model Construction 206
10.4 Performance Evaluation 215
10.5 Conclusion 217

11. Enhancing AASR through Advanced Integration of Self-Supervised Learning and Autoencoders with ML Models 219
11.1 Introduction 219
11.2 Methodology 220
11.3 Conclusion 229
12. Clustering Methods for Emotion Classification of Accented Speech 231
12.1 Introduction 231
12.2 Objectives of this Study 232
12.3 Data Collection 232
12.4 Data Pre-Processing 234
12.5 Feature Engineering 237
12.6 Clustering Algorithms 240
12.7 Performance Evaluation 246
12.8 Conclusion 247
13. Exploring Diverse Architectures - 1D CNN, 2D Parallel CNN, 4D CNN,4D Parallel CNN, Bi-LSTM, and Hybrid AASR Models 249
13.1 Introduction 249
13.2 Data Collection 249
13.3 Data Pre-Processing 251
13.4 Audio Augmentation 254
13.5 Audio Feature Extraction 257
13.6 The Feature Dimension Reduction 262
13.7 Methodology 265
13.8 Experimental Results 295
13.9 Conclusion 298
14. A Dual Approach to Detect Hate Speech in Accented Malayalam 300
14.1 Introduction 300
14.2 Hate Speech in Accented Malayalam 301
14.3 Non-Hate Speech in Accented Malayalam 301
14.4 Data Collection 302

14.5 Data Augmentation Techniques for Speech Data 304
14.6 Feature Extraction Techniques 306
14.7 Methodology 310
14.8 Performance Evaluation 313
14.9 Training Performance 313
14.10 Performance Evaluation 314
14.11 Principal Component Analysis (PCA) 316
14.12 Conclusion 317
15. Results and Discussion 319
15.1 Experiment 1 319
15.2 Experiment 2 320
15.3 Experiment 3 323
15.4 Experiment 4 332
15.5 Experiment 5 333
15.6 Experiment 6 336
15.7 Experiment 7 339
15.8 Experiment 8 342
15.9 Experiment 9 344
15.10 Experiment 10 346
16. Conclusion 348
16.1 Summary of Findings 348
16.2 Research Contributions 349
16.3 Practical Implications 350
16.4 Challenges and Future Directions 351
16.5 Ethical Considerations 352
17. Recommendations 354
17.1 Refinement of Emotion Clustering Algorithms 354
17.2 Dynamic Adaptation for Linguistic Variations 354
17.3 Large-Scale Deployment and User Interaction Studies 354
17.4 Multimodal Approaches for Enhanced Recognition 355

17.5 Exploration of Adversarial Training for Robustness 355
17.6 Cross-Linguistic Studies on Accented Speech 355
17.7 In-depth Analysis of Hate Speech Detection..... 355
17.8 Integration of Explainable AI in AASR..... 356
References..... 357
List of Publications Out of Thesis Work 384

List of Figures

Figure 1 Global Malayalam Speaking Diaspora 3
Figure 2 Organization of the Thesis..... 26
Figure 3 The Research Design 77
Figure 4 Workflow of the Proposed Study 80
Figure 5 The Phases of Data Collection..... 88
Figure 6 Statistics of AMSC-1 92
Figure 7 Statistics of AMSC-2 94
Figure 8 Age Wise Statistics..... 94
Figure 9 Age Wise Statistics of AMSC-3 95
Figure 10 Statistics of AMSC-3 96
Figure 11 Statistics of AMSC-4 98
Figure 12 Statistics of AMSC-5 99
Figure 13 Age Wise Statistics of AMSC-5 100
Figure 14 The Statistics of AMSC-6 (Original Data)..... 104
Figure 15 AMDDHS (Original Distribution)..... 105
Figure 16 Sample Spectrograms..... 106
Figure 17 Workflow of the Proposed System..... 110
Figure 18 MFCC Features..... 113
Figure 19 The RNN 115
Figure 20 A Long-Short-Term Memory Cell 116
Figure 21 Total Loss vs Computational Steps 118
Figure 22 Accuracy vs Computational Steps..... 119
Figure 23 The Speech Feature Extraction..... 131
Figure 24 Performance Evaluation of Various ML Classifiers..... 133
Figure 25 Phases of LSTM-RNN - AASR model..... 135
Figure 26 Sample Spectrogram **u**sed in the Experiment..... 137
Figure 27 Layered Architecture of the DCNN Model..... 140
Figure 28 Learning Curves of AASR Constructed with LSTM-RNN 144

Figure 29 Learning Curves of AASR Constructed with CNN 144
Figure 30 Workflow of the Study 147
Figure 31 Speech Signal Processing Approaches 149
Figure 32 The WER of Machine Learning Approaches 154
Figure 33 MER of Different Experiments 157
Figure 34 Learning Curves LSTM-RNN Approach (Accuracy Metric) 160
Figure 35 Learning Curves LSTM-RNN Approach (Loss Metric) 162
Figure 36 WER of Different Phases 164
Figure 37 Performance Evaluation of Experiments in Terms of Accuracy 170
Figure 38 Performance Evaluation (Accuracy) of Phase VII using LSTM RNN 171
Figure 39 Forty MFCC Features 178
Figure 40 The 12 Features Extracted from the Speech Signal Using STFT 179
Figure 41 Mel Spectrogram Features and the Total 180 Speech Signal Features 180
Figure 42 An Unfolded RNN and the Gated Architecture 183
Figure 43 The Performance Evaluation: Training Phase 187
Figure 44 The Performance Evaluation: Validation Phase 188
Figure 45 Training and Testing Accuracy, Loss vs Epochs 189
Figure 46 Workflow of the Proposed Study 193
Figure 47 The Train and Validation Accuracy of Two Iterations 200
Figure 48 Train and Validation Loss of Two Iterations 200
Figure 49 Train and Validation Accuracy Versus Steps 201
Figure 50 Train and validation Loss Versus Steps 201
Figure 51 Steps Involved in the Proposed Methodology 204
Figure 52 Proposed RNN 207
Figure 53 Proposed RNN with Attention Mechanism 208
Figure 54 Proposed LSTM 211
Figure 55 Proposed LSTM with Attention Block 213
Figure 56 Steps Involved in the Study 220
Figure 57 Autoencoder Model Architecture Without Compression 223
Figure 58 Autoencoder Model Architecture with Compression 226

Figure 59 Performance Evaluation 228
Figure 60 Learning Curves for Autoencoder-Based Accent Modelling 229
Figure 61 Statistics of the AMESC Dataset 233
Figure 62 Sample Emotion Data for Angry Speech 234
Figure 63 The Filtering Setup 235
Figure 64 The Audio Normalization Setup 236
Figure 65 The Feature Engineering Techniques used in the Study 238
Figure 66 Clusters of Data 240
Figure 67 Ensembled Clusters formed in the Experiment 242
Figure 68 Clusters formed by GMM 244
Figure 69 Distribution of the Accented Data After Data Augmentation 250
Figure 70 Audio Waves Before and After Filtering and Normalization 252
Figure 71 Audio Waves with and without Normalization 252
Figure 72 Audio Files Before Filtering 254
Figure 73 Audio Files After Filtering 254
Figure 74 Speech Audio with Thiruvananthapuram Accent (Original Recording) 255
Figure 75 The Stretched Wave plot (of Figure 74) 255
Figure 76 The Pitch Shifted Audio (Refer Wave plot In Figure 74) 256
Figure 77 The Wave Plot after Adding Noise (Refer Figure 74) 256
Figure 78 The Extracted 585 Features 259
Figure 79 The Original Vs Augmented Speech Set 260
Figure 80 Size of Original Vs Reduced Feature Set 263
Figure 81 Model Architecture of 1D CNN 267
Figure 82 Train-Test Loss 269
Figure 83 Train-Validation Accuracy 269
Figure 84 The Model Architecture of 2D CNN with Attention Mechanism 273
Figure 85 The Learning Curves Of 2D CNN With Attention Mechanism 273
Figure 86 Model Architecture 277
Figure 87 The Learning Curves 278
Figure 88 Model Architecture 285

Figure 89 The Learning Curves 288
Figure 90 Performance Evaluation 291
Figure 91 The Performance Evaluation 294
Figure 92 Statistics of the Hate Speech Dataset 303
Figure 93 Wave plot And Spectrogram of the Speech Data (Sample) 304
Figure 94 The Sample Signals After Augmentation 304
Figure 95 Features Extraction Statistics 308
Figure 96 The Layered Architecture 311
Figure 97 The Hate Speech Detection Model Architecture 312
Figure 98 The Learning Curves 314
Figure 99 The Confusion Matrix 315
Figure 100 Hate Speech Classification Performance 315
Figure 101 Reduced Space Formed by PCA 316
Figure 102 Performance Evaluation 319
Figure 103 Performance Evaluation of Experiment 2 321
Figure 104 Performance Evaluation using AMSC-3 324
Figure 105 Performance Evaluation in Accuracy 334
Figure 106 Performance Evaluation in terms of WER 335
Figure 107 Accuracy Vs Phases of Experiment 337
Figure 108 Model Accuracies Vs Epochs 338
Figure 109 Model Loss Vs Phases 338
Figure 110 Performance of Encoder Models 339
Figure 111 Performance of ML Models 340
Figure 112 Performance of Hybrid Autoencoder Models 340
Figure 113 Performance in Terms of Log Loss Values 341
Figure 114 Performance of ML Models VS Hybrid Models 342
Figure 115 Performance Evaluation of the Experiment 345
Figure 116 Hate Speech Evaluation 347

List of Tables

Table 1 Statistics of Malayalam Speakers in India (Census of India 2011) 3
Table 2 SER in Research 68
Table 3 The Classes of Isolated Words in the Dataset..... 92
Table 4 The Distribution of AMESC Dataset..... 101
Table 5 Sample Emotional Speech Categories in the AMESC Dataset..... 101
Table 6 AMSC-1 Dataset..... 117
Table 7 The Layered Architecture..... 139
Table 8 Data Distribution Across Different Districts and Age Groups 175
Table 9 Example Classes 176
Table 10 The performance Evaluation of LSTM-RNN..... 199
Table 11 Performance Evaluation 216
Table 12 Comparative Statistical Analysis of Normal and Normalized Audio 253
Table 13 Model Summary of 1D CNN 268
Table 14 Performance Evaluation of 1D CNN 268
Table 15 Model Summary of 2D CNN With Attention Mechanism 270
Table 16 Model Summary of 4D Parallel CNN..... 274
Table 17 The Performance Evaluation..... 276
Table 18 Model Summary of 4D CNN With Attention Mechanism 279
Table 19 The Performance Evaluation..... 287
Table 20 Model Summary of BiLSTM Model..... 288
Table 21 Performance Evaluation 291
Table 22 Model Summary 292
Table 23 Comparative Analysis..... 295
Table 24 Performance in WER..... 327
Table 25 Performance in Accuracy 330
Table 26 Performance Evaluation 333
Table 27 Performance of LSTM - RNN..... 335
Table 28 Performance Evaluation of the Clustering Techniques..... 343

List of Abbreviations

AASR	Accented Automatic Speech Recognition
AMDDHS	Accented Malayalam Dataset for Detecting Hate Speech
AMESC	Accented Malayalam Emotional Speech Corpus
AMSC	Accented Malayalam Speech Corpus
ASR	Automatic Speech Recognition
BiLSTM	Bidirectional Long Short-Term Memory
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CNN	Convolutional Neural Networks
CSV	Comma-separated Values
DANN	Domain Adversarial Neural Network
DCNN	Deep Convolutional Neural Networks
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DNN	Deep Neural Networks
DTC	Decision Tree Classifier
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GWO	Grey Wolf Optimizer
HMM	Hidden Markov Model
HNR	Harmonic-to-noise ratio
KNN	K-Nearest Neighbor
LiGRU	Light Gated Recurrent Units
LSTM	Long Short-Term Memory
LSTM-RNN	Long Short-Term Memory Recurrent Neural Networks
MAML	Model-Agnostic Meta-Learning

MER	Match Error Rate
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
OPTICS	Ordering Points to Identify the Clustering Structure
PCA	Principal Component Analysis
PLP	Perceptual Linear Prediction
PPG	Phonetic Posterior Grams
ReLU	Rectified Linear Units
RFC	Random Forest Classifier
RMS	Root Mean Square Value
RNN	Recurrent Neural Networks
SGD	Stochastic Gradient Descent
STFT	Short Term Fourier Transformation
SVM	Support Vector Machines
TTS	Text-to-Speech Synthesis
WER	Word Error Rate
WIL	Word Information Loss
ZCR	Zero Crossing Rate

1. Introduction

1.1 Background and Context

Malayalam, a Dravidian language originating in the southwestern part of India, specifically Kerala, is an intricate tapestry woven from centuries of cultural, social, and linguistic evolutions. Boasting a rich history that dates to ancient scripts found as early as the 7th century AD. According to the data regained from the Census of India Website, provided by the Office of the Registrar General & Census Commissioner with the latest available archive dated 15 August 2018, and retrieved on 26 December 2019, the language now serves as the mother tongue for over 38 million people globally (shown in Figure 1).

At the heart of Kerala's socio-cultural fabric, Malayalam has witnessed a spectrum of phonetic, syntactic, and lexical changes, making it a fascinating subject for linguistic studies. Moreover, its extensive literature, from classical songs to modern prose, highlights the region's intellectual and artistic contributions. The language has not only managed to thrive in its native region but has also made its mark globally, with substantial diaspora communities in the Middle East, Europe, and North America. Understanding Malayalam is essential not just from a cultural and linguistic perspective, but also in the domain of technology.

Capturing the details of accented speech patterns in Malayalam using advanced machine learning and deep learning techniques is challenging. Automatic Accented Speech Recognition (AASR) becomes crucial, especially in a language as phonetically rich as Malayalam, where even slight tonal variations can alter meanings.

The inherent variability in accents poses significant difficulty. Different speakers may have unique pronunciations, intonations, and rhythms influenced by their regional, social, and linguistic backgrounds. This variability makes it challenging to create a model that generalizes well across diverse accents.

The scarcity of labeled data for accented Malayalam speech is a major obstacle. Machine learning models require large amounts of annotated data to learn effectively, and obtaining such datasets for every accent variant can be resource-intensive and time-consuming. Additionally, manual annotation of speech data is laborious and prone to inconsistencies.

Another challenge is the complexity of Malayalam phonetics. Malayalam has a rich phonemic inventory with numerous phonetic variations, including subtle distinctions between sounds that are difficult for models to capture accurately. This phonetic complexity demands sophisticated feature extraction techniques and advanced models to ensure accurate recognition.

Furthermore, the presence of background noise and varying recording conditions can adversely affect model performance. Real-world speech data often includes ambient noise, reverberations, and other distortions, making it difficult for models to discern speech patterns reliably. Computational constraints can also be a limiting factor. Training deep learning models on large datasets with high-dimensional speech features requires substantial computational resources. Ensuring efficient and scalable training processes while maintaining high model performance is a significant technical challenge.

These challenges necessitate the development of innovative approaches and robust methodologies to effectively capture and recognize accented speech patterns in Malayalam.

This thesis explores the intricacies of accented speech recognition for Malayalam, employing cutting-edge machine learning and deep learning methodologies. Through this exploration, the aim is to bridge the technological divide and offer Malayalam speakers a seamless interaction with voice-activated devices, applications, and services, thus underscoring the significance of the Malayalam language in the modern digital era.

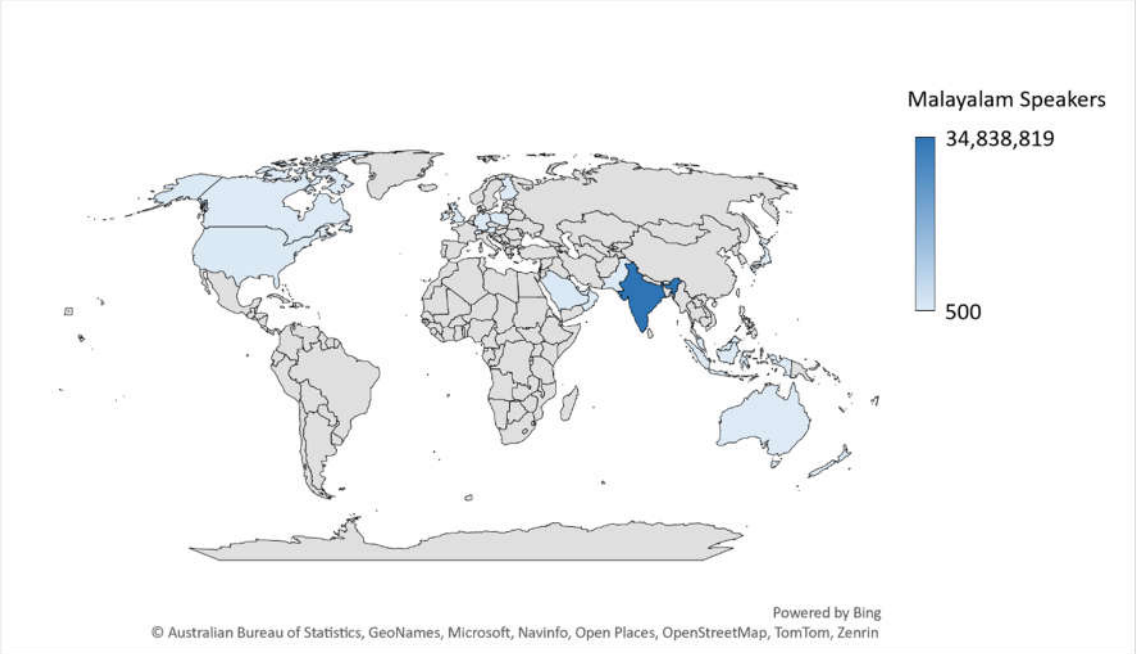


Figure 1 Global Malayalam Speaking Diaspora

Table 1 Statistics of Malayalam Speakers in India (Census of India 2011)

Rank	**State/Union Territory**	**Malayalam Speakers 2011**
—	India	3.4838819×10^{7}
1	Kerala	3.2413213×10^{7}
2	Tamil Nadu	$9.57{,}705\text{x}10^{5}$
3	Karnataka	$7.01{,}673\text{x}10^{5}$
4	Lakshadweep	$5.4{,}264\text{x}10^{4}$
5	Puducherry	$4.7{,}973\text{x}10^{4}$
6	Andaman and Nicobar Islands	$2.7{,}475\text{x}10^{4}$

Table 1 provides a detailed breakdown of the number of Malayalam speakers across various states and union territories in India, based on the 2011 Census data. The table ranks these regions by the population of Malayalam speakers, offering insights into the distribution of this linguistic group within the country.

1.2 Relevance of the Study

In today's rapidly advancing digital age, speech recognition stands as a cornerstone for numerous applications, from voice assistants to transcription services. As technology becomes increasingly ubiquitous, the imperative for systems to recognize and adapt to diverse languages and accents grows stronger. While major global languages have seen significant advancements in this domain, regional languages, especially those with complex phonetic structures like Malayalam, often remain unexplored.

Malayalam, with its rich phonetic variety and multiple dialects, presents a unique challenge in speech recognition. Differences in accents across its regions, influenced by factors such as geography, socio-cultural practices, and even individual education and exposure, further complicate this landscape. The relevance of this study is manifold:

1. Inclusivity and Access: By improving accented speech recognition for Malayalam, it can be ensured that a significant population is not left behind in the digital transformation. This work aids in offering equal access to digital tools and services for Malayalam speakers, irrespective of their regional accents.

2. Economic Impact: Kerala, the primary region where Malayalam is spoken, has a large diaspora globally. Enhanced speech recognition can facilitate smoother communication, potentially boosting sectors like business, tourism, and even remote healthcare.

3. Preservation and Promotion: Advanced speech recognition for regional languages can play a crucial role in preserving and promoting cultural heritage. By cataloging and analyzing various accents and dialects, this research might also contribute to linguistic studies.

4. Technological Advancement: This study pushes the boundaries of what's achievable in the domain of speech recognition. The methodologies and findings

can potentially be extrapolated to other similar languages, amplifying the impact of this research.

5. Educational Implications: Enhanced speech recognition can be a boon for educational platforms, enabling more effective e-learning solutions, especially in remote areas where traditional education infrastructure might be lacking.

6. Cyber Forensics: Accurate identification and analysis of spoken content in various languages, such as Malayalam, becomes imperative for cyber forensic investigators. Accents and dialects can often provide crucial clues about the geographic origin of a suspect or the target audience of a cyberattack. This information aids in narrowing down the pool of potential suspects and understanding the cultural context of the communication. Accented speech recognition, when integrated into forensic tools, can enhance the efficiency and accuracy of cybercrime investigations, thereby contributing to the overall security of digital environments. Furthermore, it enables forensic experts to reconstruct conversations and audio evidence with precision, ensuring the integrity and admissibility of evidence in legal proceedings.

7. Inclusive Healthcare Support for Accented Speech: Accented speech assistive applications play a vital role in healthcare, providing assistance tailored to individuals with different linguistic backgrounds. These applications help set medication reminders, schedule appointments, and offer health information, ensuring equitable support for diverse patient populations.

8. Business Efficiency with Accented Speech Assistants: In professional settings, accented speech recognition applications enhance business productivity. Executives and professionals can use voice commands, inclusive of various accents, to schedule meetings, draft emails, and access information, fostering a more diverse and inclusive work environment.

9. Accented Speech in Retail and Customer Service: Voice assistants attuned to accented speech contribute to improved customer interactions in retail. They enable voice-activated searches, answer product-related queries, and enhance the overall shopping experience by accommodating individuals with diverse accents.

10. Access to Entertainment with Accented Speech Recognition: Accented speech assistive applications transform how individuals with diverse accents consume entertainment. Users can control smart TVs, search for content, and adjust settings using voice commands, ensuring an inclusive and accessible entertainment experience.

11. Recognition in Public Spaces: Integration of accented speech recognition into public spaces enhances accessibility for individuals with diverse linguistic backgrounds. Voice-activated features, such as elevators, doors, and information kiosks, contribute to creating inclusive environments in public buildings and transportation hubs.

12. Facilitating Accented Speech in Language Translation: Accented speech recognition applications with language translation capabilities facilitate cross-cultural communication. Users can communicate in their native accent, and the application translates and conveys the message, fostering understanding and collaboration across linguistic diversity.

13. Inclusive Smart Home Automation for Accented Speech: Voice assistants tailored for accented speech empower users to control smart home devices effortlessly. From adjusting settings to managing various systems, these applications ensure that individuals with diverse accents can fully participate in the benefits of smart home automation.

This thesis not only addresses a technological challenge but also has profound socio-cultural and economic implications, making the research both timely and essential.

Speech recognition technology has made leaps and bounds over the past decade, with various applications integrating voice-activated commands to enhance user experience.

Malayalam, intrinsic to the socio-cultural identity of Kerala, is spoken with variances in accent, intonation, and pronunciation across its diverse geography. From the highlands of Idukki to the coastal regions of Kozhikode and from Thiruvananthapuram to Kasaragod districts, distinct accents emerge, each carrying the weight of its historical, socio-cultural, and even topographical influences. These accents, while being the very essence of the language's diversity, present a formidable challenge for conventional speech recognition systems.

The inability of current systems to accurately recognize and transcribe accented Malayalam speech often leads to:

1. Miscommunication: Inaccurate transcriptions can render voice commands useless or even lead to unintended actions, especially in critical applications like medical transcription or emergency services.

2. Digital Exclusion: A significant portion of Malayalam speakers, especially those from rural or traditionally underrepresented regions, find themselves at a disadvantage when interacting with voice-driven technologies. This digital divide only widens as technology progresses without considering these variations.

3. Economic Repercussions: As businesses increasingly rely on automated customer service solutions and voice-driven interfaces, the inability to cater to the diverse Malayalam-speaking populace could result in lost opportunities and revenue.

4. Cultural Homogenization: Not recognizing the rich tapestry of Malayalam accents risks a form of linguistic homogenization where unique regional identities get overshadowed.

1.3 Scope of the Research

The primary focus of this research work lies in the construction of a comprehensive AASR, specifically targeting the Malayalam language that focuses on understanding regional accents in Malayalam using machine learning and deep learning techniques. It also involves creating a specialized dataset due to the lack of existing resources, with the potential to guide similar research in other languages. Given Malayalam's rich linguistic tapestry, rich in regional accents, this study seeks to contribute in several significant ways:

1. Comprehensive Dataset Development: One of the foundational pillars of this research is the curated dataset, which encompasses diverse Malayalam accents sourced from various regions of Kerala. This dataset, potentially one of the most comprehensive of its kind, offers a unique blend of age groups, genders, and socio-economic backgrounds, ensuring a wide representation of the Malayalam-speaking populace.

2. Technological Advancement: This research aims to construct efficient models for AASR in Malayalam. These models are built using various approaches including traditional algorithms, neural networks, transfer learning and ensemble methods, and represent the forefront of technological advancement in this domain.

3. Clustering Analysis: Beyond the conventional recognition models, this research investigates clustering techniques to categorize the emotions from diverse accents within the Malayalam language. This analytical approach provides deeper insights into the underlying patterns and structures of Malayalam accents, offering a more granular perspective.

4. Inclusivity and Access: At its core, this research seeks to bridge the digital divide, ensuring that Malayalam speakers, irrespective of their regional accents, can seamlessly interact with voice-driven technologies. This inclusivity has far-

reaching implications, from personalized voice assistants to more accessible e-learning platforms for regional communities.

5. Blueprint for Other Languages: The methodologies and insights derived from this research have the potential to serve as a blueprint for similar endeavors in other regional languages, and hence contributing significantly to the broader field of accented speech recognition.

1.4 Limitations of the Research

A significant limitation lies in the scarcity of comprehensive datasets in the domain of accented Malayalam speech, affecting the robustness and generalizability of the models developed. The complexity of the machine learning and deep learning models may require significant computational resources, which could act as a limiting factor. The limited availability of data and resources may also result in the study covering only a subset of regional accents within the Malayalam-speaking community, leading to potential biases in the model. While there are some works available for comparison, the research in Malayalam accented speech recognition is not highly advanced. This limits the extent to which the models and findings of this study can be directly compared and validated against existing works.

1.5 Research Gap

In the expansive field of automatic accented speech recognition, while significant strides have been made for major global languages, regional languages often face a two-fold challenge. For Malayalam, these challenges become even more pronounced.

1.5.1 Lack of Comprehensive Study

One of the most evident gaps in the domain of AASR for Malayalam is the lack of comprehensive research. While there have been isolated studies focusing on Malayalam speech recognition at a broader level, in-depth research specifically targeting the diverse accents within the language remains scant. This lack of detailed studies means that many inherent disparities and subtleties associated with regional

Malayalam accents are yet to be thoroughly explored and understood in the context of automated speech recognition.

1.5.2 Unavailability of Benchmark Dataset

Central to any speech recognition research is the availability of a robust dataset. In the case of accented Malayalam, there is a conspicuous absence of a benchmark dataset that captures the spectrum of accents prevalent within the language. Existing datasets, if any, tend to be narrow in scope, lacking in diversity, and not representative of the broader Malayalam-speaking population. Without such a foundational dataset, constructing and validating models becomes a challenge, limiting the potential advancements in this field.

This research aims to address these critical gaps. By undertaking a comprehensive exploration of accented Malayalam and curating a benchmark dataset that encompasses its rich accentual diversity, this study seeks to pave the way for future research endeavors and technological advancements, ensuring that Malayalam, in all its phonetic glory, finds its rightful place in the digital landscape.

In essence, the scope of this work spans technological, linguistic, and socio-cultural dimensions, with a vision to redefine the interaction between Malayalam speakers and the digital world, all while preserving and acclaiming the language's rich accentual diversity. This research, in essence, is a synthesis of data collection, rigorous preprocessing, and diverse modeling approaches, all converging towards one goal: crafting a state-of-the-art AASR system for Malayalam.

1.6 Objectives

The primary objective of this study is to identify and develop accent-robust speech recognition models through rigorous investigation and experimentation. The specific objectives of this study are listed below:

1. Design an Acoustic Model for Dialect Identification: Develop an advanced acoustic model that effectively identifies and differentiates utterances across various dialects of the Malayalam language.

2. Robust Performance in Natural Environments: Ensure that the designed model exhibits robust performance and high accuracy when processing audio signals collected in natural, uncontrolled recording environments.

3. Versatile Device Compatibility: Create a model capable of accurately recognizing Malayalam speech signals recorded using diverse recording devices, including but not limited to microphones, headphones, and smartphones.

To achieve this goal, the study is structured around several key components:

1. Dataset Curation Process
2. Feature Engineering Approaches
3. Feature Extraction Methods
4. Investigation of Diverse Approaches for Robust Accent Model Construction
5. Comprehensive Evaluation

1.6.1 Dataset Curation Process

Data collection was approached methodically, sourcing speech samples from various regions of Kerala. Ensuring diversity in different age groups, genders, and socio-economic backgrounds aimed to capture the diverse accents prevalent within the Malayalam-speaking populace. Furthermore, this data was annotated carefully, ensuring accurate transcription and metadata detailing the regional accent, context, and other significant factors. At the heart of this research lies the carefully curated datasets. Recognizing the importance of representative data, efforts were channeled into collecting, cleaning, and annotating diverse speech samples. From isolated words to emotionally charged statements, each dataset was crafted to serve the specific objectives of the various research segments.

The dataset encompasses a diverse collection of accented speech data, comprising both isolated and continuous recordings. These recordings were captured in real-world, noisy environments, reflecting the complexities of natural settings. Additionally, few datasets were curated from YouTube videos, broadening the scope of the dataset to include a varied range of accents and speech patterns found in online content.

This comprehensive compilation aims to provide a holistic representation of accented speech across different contexts, ensuring the robustness and applicability of the dataset in the field of AASR for Malayalam. Overall, the datasets are vast, varied, and well-structured, promising a broad and deep exploration of the research objectives. Each dataset was designed to meet specific needs, from recognizing accents and dialects in general conversations to identifying hate speech and detecting emotions in accented Malayalam speech.

1.6.2 Feature Engineering Approaches

The preliminary steps in the experiment involve data preprocessing and feature extraction methods. Different approaches were adopted for each experiment for data processing and feature engineering approaches. The different approaches adopted for feature engineering are discussed in this section.

1.6.2.1 Data Preprocessing

With the raw data in hand, the next phase was its transformation into a format amenable to model training. This involved noise reduction, segmenting longer recordings into manageable chunks, and normalization of audio levels. Concurrently, textual data was cleaned and standardized, ensuring synchronicity between speech and its corresponding transcription.

1. Noise Reduction

Given the potential for extraneous auditory disruptions within the raw audio data, advanced noise reduction algorithms were employed. These algorithms were

designed to identify and minimize any underlying, consistent noise without altering the core speech component. By doing so, the essential characteristics of the accented speech were preserved and ensured.

2. Filtering

Filtering was another essential step in the preprocessing methodology. By applying various band-pass filters, the frequency range that is most relevant to human speech was able to isolate, thereby eliminating frequencies that do not contribute to the comprehension of the Malayalam language accents. The design of the filters was carefully tailored to the unique characteristics of the Malayalam language.

3. Normalization:

To further ensure consistency across the entire dataset, normalization techniques were employed. Normalization played a crucial role in mitigating any variations in volume, pitch, and other speech attributes across different recordings. By standardizing these attributes, the learning process for the model was facilitated, allowing it to focus on the intrinsic patterns of the accented speech rather than irrelevant variations.

1.6.2.2 Data Augmentation

Data augmentation is crucial for enhancing the diversity and richness of the training dataset. By introducing variations like noise, time shifts, and pitch changes, the model becomes better equipped to handle real-world challenges in speech recognition. These techniques, when applied judiciously, can significantly improve model performance, especially in scenarios with diverse accents and varying recording conditions.

1. Time Stretching and Pitch Shifting

By slightly altering the speed and pitch of the speech recordings, subtle variations that allowed the model to recognize accented speech across a broader range of

tonalities were created. This ensured that the model could adapt to natural variations in speech without overfitting to the specific characteristics of the training set.

2. Adding Background Noise

Introducing controlled amounts of background (Gaussian) noise at different levels simulated real-world listening environments. This method enabled the model to discern the complications of Malayalam accented speech even in less-than-ideal auditory conditions, thereby enhancing its real-world applicability.

3. Random Cropping

By randomly selecting and extracting segments of the speech recordings, a more varied dataset was generated. This process ensured that the model was exposed to different parts of speech patterns, leading to a more comprehensive understanding of the language's accents.

4. Volume Modulation

Adjusting the volume levels within the dataset exposed the model to the natural fluctuations in loudness that occur in authentic speech. This technique furthered the goal of building a model sensitive to the real-world dynamics of Malayalam speech.

5. Spectral Augmentation

By manipulating the spectral characteristics of the speech signal, a variety of representations that captured different frequency characteristics were created. This approach helped the model to recognize accented speech across diverse auditory profiles.

1.7 Feature Extraction Methods

Different feature extraction methods have been adopted distinctively in different experiments. Some of the methodologies adopted for the purpose are Mel Frequency Cepstral Coefficients (MFCC), Short Term Fourier Transformation (STFT), Mel

Spectrogram, Tempogram, Zero Crossing Rate (ZCR), and Root Mean Square Value (RMS). These feature vectorization techniques are employed individually and in varying combinations in the study to investigate the better representation of accented speech. A few other methods adopted for feature engineering are discussed below:

Tonnetz & Polyfeatures: Feature vectors obtained by quantifying harmonic relations and polynomial coefficients of the spectrogram, and hence deeper insights into the audio's tonal structure were obtained, Pitch variability that measures variations in the fundamental frequency, which is crucial for understanding tonal details, ZCR & Chroma STFT that is used for separating the audio's inherent noisiness and harmonic content, and these features provide pivotal insights into its texture, RMS & Mel Spectrogram is used to generate crucial metrics on the audio's loudness and its frequency spectrum representation on the Mel scale, MFCCs, and its Deltas generate the frequencies of the power spectrum of the speech signals.

The feature extraction procedure also extracts its first and second derivatives (deltas and delta-deltas), which provide insights into the trajectory of MFCCs over time, Harmonic-to-noise ratio (HNR) evaluates the clarity of the voice by comparing harmonic and noisy components.

1.8 Approaches for Constructing AASR Models

The accented models constructed in the entire research utilized the following techniques. The algorithms and techniques utilized for conducting various experiments are listed below:

1. Deep Convolutional Neural Networks (DCNN)
2. Long Short-Term Memory – LSTM Architecture
3. Recurrent Neural Networks – RNN Neural Network Architecture
4. Bidirectional Long Short-Term Memory Architecture (BiLSTM)

5. Incorporating Attention mechanisms with RNN, CNN, LSTM, and BiLSTM Neural Network Architectures
6. Machine learning algorithms.
7. Clustering Techniques
8. Auto Encoders with and without compressed data
9. Hybrid Auto Encoder Architecture
10. 1D, 2D, and 4D parallel CNNs with Attention Mechanisms
11. Hybrid models (CNN + LSTM)

This research is carried out by adopting various methodologies and feature engineering techniques which are explained in the subsequent sections.

The study begins with an extensive literature survey followed by the exploration of the data collection phase, where efforts are made to curate datasets of varying accents and length that encapsulates the variations of accented speech in Malayalam. Study from across different domains of this research was thoroughly carried out. Even though the study related to AASR in Malayalam is very scarce, studies have been carried out in many other languages. Efforts have been taken to incorporate the insights obtained from the existing literature and novel methods have been adopted for conducting this study. Papers from different publishers like Elsevier, Springer, Interspeech, Speech Communication etc. have been referred to study the existing work in literature.

One of the major challenges in conducting this research was the unavailability of benchmark datasets in the domain as discussed. Constructing the appropriate dataset was crucial for conducting this research. Finding appropriate speech donors, acquiring their consent, preparing the speech classes for recording, accumulating the recordings obtained and then the cleaning and annotation procedures were time consuming and tedious tasks. The dataset curation was conducted in nine phases

that includes both crowdsourced data and the data acquired from online platforms. One among these is spectrogram dataset that contains the spectrograms of the audio recordings. The dataset includes recordings of individual utterances, multisyllabic utterances, and sentences.

These datasets form the backbone of the subsequent studies, reflecting the diversity of accents present in the language. The methodologies employed in constructing distinct accented models using varied approaches, each adapted to tackle specific aspects of the accented speech recognition challenge is discussed here. These methodologies are carefully chosen to ensure a detailed understanding of the impact of different features and model architectures on the recognition accuracy within the Malayalam linguistic landscape.

The study constructs the AASR model that identifies various classes of utterances it has been exposed to during training and converts the speech into standardized Malayalam text. The initial experiment was conducted on twenty isolated word classes, and the results were evaluated. A model that recognizes accented speech in the Malayalam language and translates it into corresponding text has been constructed using deep learning and machine learning techniques. The model is specifically trained to discern 20 categories of isolated Malayalam words, divided equally between Malayalam numerals and a selection of random words in the language and the model has been evaluated with appropriate metrics.

In the succeeding phase of the study multisyllabic isolated utterances, were collected from both male and female speakers across various age groups, recorded in a natural environment. The central aim of the research was to discover an improved method of feature extraction that accurately represents the characteristics of accented Malayalam speech. In the next phase of the research the focus was specifically on word-based ASR using deep learning techniques. This study evaluates the performance of the experiments in constructing AASR adopting the LSTM-RNN and DCNN methodologies.

The study on AASR employed various machine learning (ML) and LSTM-RNN-based acoustic modeling techniques in the subsequent phase of the research. Through a layered approach to acoustic signal processing to extract the feature vectors, utilizing methods like MFCC, STFT, Mel Spectrogram, Spectral Roll-Off, and other techniques, an enhanced ML and LSTM-RNN system that outperforms traditional accent-independent ASR systems was developed. The study on AASR employed various machine learning (ML) and LSTM-RNN-based acoustic modeling techniques in the subsequent phase of the research. Through a layered approach to feature extraction, utilizing methods like Mel Frequency Cepstral Coefficients (MFCC), Short Term Fourier Transform (STFT), Mel Spectrogram Spectral Roll-Off, and others, an enhanced ML and LSTM-RNN system that outperforms traditional accent-independent ASR systems was developed.

A novel approach to AASR for Malayalam, employing advanced deep learning techniques such as RNN, LSTM, BiLSTM and all these architectures combined with attention mechanisms. In conjunction with Mel Frequency Cepstral Coefficients (MFCC) and Tempogram features, this study seeks to enhance the accuracy of recognizing multi-accented Malayalam speech.

The next phase of the research focused on an application part of the AASR in Malayalam. Efforts have been taken for the effective classification of the seven emotions from the accented speech corpus curated for conducting the study. This research is supported by a dataset that captures seven distinct emotions: anger, disgust, fear, happiness, neutrality, sadness, and surprise. Clustering techniques were employed for categorizing the emotions in this study. A comprehensive study was conducted using multidimensional CNN, BiLSTM and hybrid approaches using the accented dataset constructed from online platforms (YouTube). The performance of each methodology is then compared and analyzed as part of the research.

The research also involves study on hate speech classification based on accented Malayalam dataset. The dataset was constructed from freely available online data

(YouTube). Speech vector dimension reduction techniques were also used in the study. The model was constructed using CNN architecture. The results of the entire phases of this research are analyzed and compared to find the methodology that fits better for constructing AASR models for the accented Malayalam data. Construction of accented models is a key aspect of the study, involving the application of diverse techniques such as LSTM, RNN, CNN, DNN, Hybrid and Ensembled approaches and machine learning algorithms.

1.9 Comprehensive Evaluation

In this study, the AASR models were efficiently evaluated and assessed using various evaluation metrics. By employing these metrics, the study aims to portray an understanding of the strengths and weaknesses of the developed models, thereby laying the foundation for future research to address any identified shortcomings.

The different evaluation metrics used in the study are listed below:

1. Accuracy: Served as the primary metric for model performance.
2. Word Error Rate (WER): Employed to measure the model's robustness, with a focus on achieving low WER scores.
3. F1 Score: Used to consider both precision and recall.
4. Loss: Measured to quantify the difference between predicted and actual outputs.
5. Recall: Used to identify the sensitivity of the model.
6. Log Loss: Utilized for evaluating the probabilities.
7. Match Error Rate: Employed to measure the rate of false positives and negatives.
8. The silhouette Score: Quantifies the cohesion and separation of clusters in a clustering analysis. It provides a measure of how well-defined and distinct the clusters are within a dataset. A higher silhouette score is generally indicative of

well-separated, compact clusters, providing a quantitative measure of the clustering model's effectiveness.

1.10 Organization of the Thesis

In Chapter 1, the introduction provides essential background information in the research domain, setting the stage for the study. The chapter sheds light on the literature in the domain that has been published at several platforms. The review of the existing research in the domain is conducted in different dimensions like the methodologies adopted, feature vectorization techniques, size of the dataset etc.

Chapter 2, the literature review, serves as a comprehensive survey of the existing work in the domain, in the fields of speech recognition, machine learning, and the intricacies of regional accents. Emphasis is placed on identifying gaps in the existing research landscape, with a keen focus on the unique context of the Accented Malayalam language. This chapter discusses the current state in these areas and lays the foundation for subsequent research by pinpointing areas that require further exploration and innovation.

Chapter 3 discusses the various methodologies adopted during different phases of this research. The research employs different feature engineering techniques and AASR model construction for different phases as part of investigating the approach that performed well for modeling the AASR for Malayalam.

Chapter 4 thoroughly investigates the challenges encountered during dataset creation, providing insights into the intricacies of working with regional accents. It describes the innovative strategy employed to overcome these challenges and elaborates on the methodology adopted for data collection, offering a step-by-step account of how the comprehensive dataset for accented speech recognition was thoroughly generated.

Chapter 5 discusses the creation of an LSTM-RNN acoustic model, tailored to recognize accent-based speech in the Malayalam language, offering a comprehensive

understanding of the model's architecture and its implications in the context of speaker-independent accent recognition.

Chapter 6 focuses on the feature engineering phase of the research and adopted a hybrid approach for constructing accent-based speech data. The classification phase encompasses both machine learning and deep learning approaches. In the area of machine learning, an array of classifiers such as MLP, Decision Tree, SVM, Random Forest, KNN, and SGD are used for constructing AASR models using the extracted features. The deep learning phase includes the development of AASR based on LSTM-RNN architecture and another AASR system has been constructed using DCNN architecture employing spectrograms. LSTM-RNN model has been constructed with the acoustic signal vectors obtained using acoustic signal processing techniques and features will be extracted from the spectrograms for constructing AASR from Deep Convolutional Network Architecture. This chapter offers a comprehensive exploration of the experimental setup and methodologies applied in the research.

Chapter 7 discusses the systematic experiments carried out in feature engineering, including the extraction of accented features and the construction of different unified models. Results are presented and analyzed, underscoring the success of the approach not only with known and unknown accents but also with accent-agnostic standard Malayalam. This chapter discusses the experiments conducted with two distinct datasets AMSC-1 and AMSC-2. The same methodology is applied to these datasets to construct the AASR model. The Chapter describes a holistic view of the research methodology, from data collection to model construction and performance assessment, ensuring a thorough understanding of the study's insights into accented Malayalam in low-resourced contexts.

In Chapter 8 AASR systems are constructed using LSTM-RNN and DCNN, designed for recognizing diverse Malayalam accents. These models are thoroughly evaluated for accuracy, efficiency, and robustness using a preprocessed test set. A

comprehensive comparative analysis is then undertaken, delving into the strengths, weaknesses, and unique qualities of both the LSTM-RNN and DCNN models, offering valuable insights to guide future research in the domain of constructing AASR for Malayalam and for languages with small datasets.

Chapter 9 explores various machine learning approaches, including MLP, DTC, RFC, KNN, SVM, and SGD, to construct AASR models in recognizing accent patterns in Malayalam speech. Furthermore, the research also constructs an ensembled or hybrid model, combining the strengths of different models to augment the accuracy and robustness of the system. The LSTM-RNN approach is also employed to construct the AASR model to represent the intricate variations within Malayalam speech. Finally, the chapter includes information regarding the thorough outcome and analysis of the results obtained from these diverse experiments, culminating in insights and conclusions regarding the most effective techniques for accent-robust ASR in the context of the low-resourced Malayalam language.

In Chapter 10, the research investigates a novel approach to construct AASR for the Malayalam language, incorporating advanced deep learning techniques. The chapter examines the techniques used, including RNN, LSTM, BiLSTM, and Attention Mechanisms, which are pivotal in the development of the AASR system. The chapter discusses the utilization of MFCC and Tempogram features as fundamental acoustic signal processing techniques in the accent recognition process. The research presents a detailed examination of six distinct experimental phases and approaches, shedding light on the intricacies of spectral features for accent identification. A novel gradient optimization method is introduced, specifically designed to address challenges encountered during the training of deep learning models. This chapter illustrates a comprehensive evaluation of deep learning techniques and their application in accented ASR within the low-resource setting of Malayalam.

Chapter 11 describes the methodologies adopted for modeling AASR with self-supervised learning techniques and autoencoders to enhance the recognition of Malayalam accented speech. This research introduces a pioneering autoencoder model in the context of constructing Malayalam AASR. It not only captures underlying patterns and features within the data but also explores the analysis of accented speech without data compression, shedding light on the use of original high-dimensional feature vectors. The study also investigates the impact of compressing data into a lower-dimensional latent space, preserving essential information while reducing dimensionality, providing a new perspective on feature engineering.

In Chapter 12 the research tackles the task of emotion classification in accented speech in Malayalam, employing a complicated approach. It commences with data collection, emphasizing the acquisition of the necessary speech data for analysis. Subsequently, data preprocessing is highlighted, ensuring the optimization of the dataset for further analysis. The core of the chapter revolves around the clustering techniques employed, which contribute to the enhancement of emotion classification.

Chapter 13 discusses the experiments conducted as part of the research in accented speech recognition for Malayalam using 1-dimensional, 2-dimensional, and 4-dimensional CNN architecture with and without attention mechanisms and BiLSTM architectures. A hybrid approach incorporating CNN and LSTM networks was also employed in the study. Specifically, the study focused on the crucial task of sentence classification with continuous speech data. Sentence classification plays a pivotal role in deciphering spoken language, as it forms the foundation for various applications such as voice assistants, transcription services, and more.

Chapter 14 focuses on the investigation of hate speech detection in dialectal variants of Malayalam. The research begins with data collection, a crucial step in gathering

the necessary audio samples for analysis. Subsequently, data preprocessing is emphasized, where 162 features are extracted for each audio sample, encompassing a diverse range of acoustic characteristics, such as Zero-Crossing Rate (ZCR), Chroma-STFT, MFCC, Root Mean Square (RMS), and a substantial 138 features from the Mel Spectrogram. The chapter then examines the utilization of a CNN-based neural architecture for hate speech detection, showcasing the adoption of advanced deep learning techniques. Finally, the results and evaluation section scrutinize the outcomes and performance metrics, offering insights into the efficacy of the model in detecting hate speech in dialectal variants of Malayalam.

Chapter 15 discusses the results obtained in the different phases of the research and evaluates the results obtained in each experiment conducted as part of this research. Chapter 16 serves as the concluding section of the research, encapsulating the findings. This chapter contains a comprehensive conclusion and discussion, where the key discoveries, implications, and insights drawn from the preceding chapters are synthesized and discussed. This section offers an advanced summary of the research's achievements and their significance in the context of constructing models for accented speech recognition, emotion classification, hate speech detection, and other related areas in the Malayalam language.

Chapter 17 provides the recommendations and provides a roadmap for future works. This chapter outlines the section on future work, which provides potential directions for further research and exploration. It highlights areas where improvements, enhancements, and novel approaches can be applied to advance the field of accented speech analysis in Malayalam, emphasizing the ongoing quest for innovation and progress in this domain. The organization of the thesis is illustrated in Figure 2.

1.11 Conclusion

Malayalam, with its captivating array of phonetic variations and accents, is the prime example of this complexity. When moving ahead of this research journey, it becomes evident that the exploration is not merely a technical pursuit but an exploration of linguistic diversity. The subsequent chapters discuss the methodologies, innovative approaches, and insightful findings that characterize this research. By addressing the multifaceted challenges of accented Malayalam speech recognition, this work seeks to bridge gaps, advance knowledge, and, most importantly, honor the linguistic richness that the Malayalam language so beautifully encapsulates.

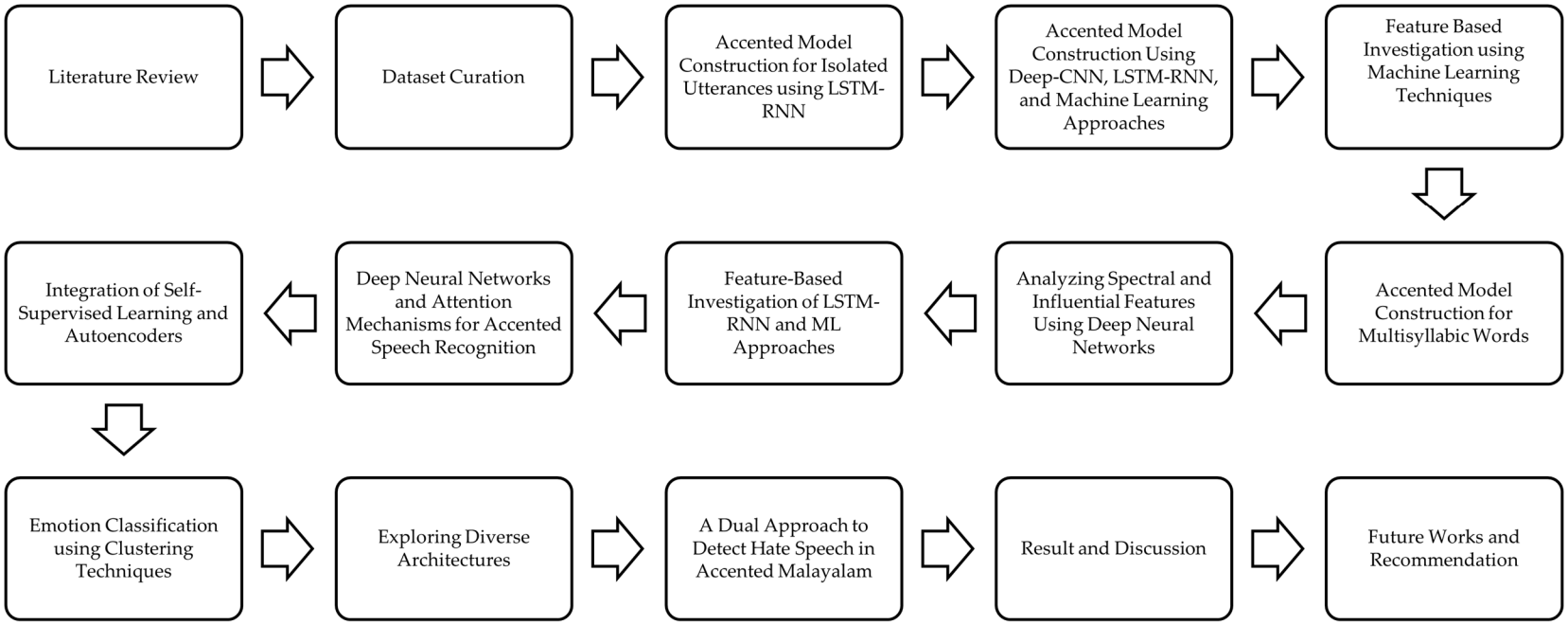


Figure 2 Organization of the Thesis

2. Literature Review

2.1 Introduction

Malayalam is deeply woven into the cultural fabric of Kerala, encapsulating the state's rich heritage, history, and traditions. Accents within the language are not merely linguistic idiosyncrasies but also bear the imprint of the diverse cultural tapestry of the region. Exploring these accents unveils a deeper understanding of how language is not just a communicative tool but an integral part of culture, community, and place. This literature review embarks on a comprehensive exploration of the relatively limited body of research about accented Malayalam recognition.

2.2 AASR in Literature

This study on related works in the area aims to offer a comprehensive overview of the current state of research in these areas, shedding light on how AASR has evolved to address the intricacies of Malayalam accents. The insights gained from this review will provide a deeper understanding of the challenges faced in automated speech recognition when dealing with linguistic diversity. This section of the chapter examines the significant works that are published in the domain of AASR on languages across the globe.

2.2.1 Existing Approaches in AASR

In this literature review, state-of-the-art techniques and methodologies in Accented Automatic Speech Recognition (AASR) proposed by various researchers are explored. Through an in-depth analysis of recent studies, insights into the innovative approaches, methodologies, and architectures employed to address the challenges inherent in accent variability are gleaned. The works discussed encompass a wide spectrum of research endeavors, ranging from speech corpus development and pre-trained model creation to accent identification methods and novel architectures for accent modeling. By synthesizing the findings from these diverse studies, a

comprehensive overview of the current landscape of AASR research, highlighting key advancements, challenges, and future directions in the field, is provided.

In the study conducted by Alëna et al.,[1] introduced techniques for developing speech corpus and created pre-trained models for AASR, employing wave to vectorization techniques. Accented Automatic Speech Recognition (AASR) systems face additional challenges compared to traditional ASR systems due to the presence of accent variability. Existing approaches in AASR encompass a wide range of techniques aimed at mitigating the effects of accent variability and improving recognition accuracy.

One approach is adaptation methods, which aim to adapt ASR models to specific accents by fine-tuning model parameters on accent-specific data in a study conducted by Das et al., [2]. These methods typically involve collecting accent-specific training data and using techniques such as feature space adaptation or model retraining to adapt the ASR system to the target accent. Another approach by Das et al., [2] is accent identification methods, which involve identifying the accent of a speaker before transcribing the speech. By accurately identifying the accent, these methods enable the selection of accent-specific acoustic models or language models, improving recognition accuracy for accented speech. They proposed hybrid systems combining DNNs and HMMs, to better capture accent-specific acoustic patterns. These hybrid systems utilize the representational power of DNNs for acoustic modeling while retaining the modeling flexibility of HMMs, resulting in improved recognition performance for accented speech.

The development of robust AASR systems requires a combination of adaptation methods, accent identification methods, and specialized acoustic modeling techniques to effectively address the challenges posed by accent variability.

Muhammad et al., [3] proposed a two-level pipeline architecture for constructing AASR models capable of handling both accented and non-accented data. Their method yielded a substantial reduction in Word Error Rate (WER), demonstrating

its efficacy in improving AASR accuracy. Additionally, J. Ni, L. Wang, H. Gao, et al., [4] introduced an innovative approach to unsupervised text-to-speech synthesis (TTS), addressing a critical challenge in TTS technology.

In their study, A Jain et al., [5] introduced the "mixture of experts" architecture, which enhances speech classification by transcending phonetics and accents. Qian et al., [6] proposed accent classification methods utilizing layer-wise embedding techniques. Furthermore, Imaizumi et al., [7] incorporated multi-task learning techniques into a Japanese AASR model, showcasing the versatility of their approach.

Deng et al., [8] proposed pre-trained methods for accent identification, utilizing frame-level vectors to construct AASR systems. Similarly, H. Huang et al., [9] developed an accent identification system using a pre-trained model and the phone posteriorgram method, resulting in a reduced Word Error Rate. Hyeong et al., [10] proposed an approach that minimized deviations in accented speech through feature extraction, domain prediction, and accent classification phases.

Dhanjal and Singh [11] conducted a comprehensive analysis of existing literature in the field of modeling Automatic Speech Recognition (ASR), providing valuable insights for accuracy improvement. Additionally, Y. C. Chen et al., [12] presented techniques for modeling unified AASR capable of identifying speech across various accents, leveraging generative adversarial nets. Song Li et al., [13] constructed an AASR model for identifying accented English data, proposing two unsupervised architectures for AASR construction.

In summary, these studies represent significant contributions to advancing AASR technology, addressing key challenges and paving the way for future innovations in the field.

2.2.2 Feature Engineering Approaches

This section explores the latest developments and innovative approaches in AASR and related studies. AASR presents unique challenges due to the diverse variations

in pronunciation, intonation, and phonetic characteristics across different accents. Feature extraction is a critical step in any ASR system, including AASR. The goal is to transform the raw audio signal into a set of features that can effectively represent the phonetic and linguistic content of the speech while being robust to variations such as accents. This section outlines the key methodologies and mathematical formulations used in feature extraction for AASR.

2.2.2.1 Mel-Frequency Cepstral Coefficients (MFCCs)

One of the most used feature extraction techniques in ASR systems is the computation of Mel-Frequency Cepstral Coefficients (MFCCs). The process involves several steps, including pre-emphasis, framing, windowing, Fast Fourier Transform (FFT), and Mel filter bank processing. The final step is taking the discrete cosine transform (DCT) of the log Mel-spectrum, which decorrelates the coefficients. The formula for calculating the MFCCs is given by:

$$MFCC_k = \sum_{n=1}^{N} \log(S_n) \cos\left[k\left(n - \frac{1}{2}\right)\frac{\pi}{N}\right] \quad (1)$$

where S_n is the log energy output of the Mel-filter bank for the n^{th} filter, and N is the total number of Mel filters used which is proposed by Davis et al., [14] and Rabiner et al., [15]. This method is robust to variations in speech signal energy and frequency components, making it effective for capturing accent variations.

2.2.2.2 Linear Predictive Coding (LPC)

Another feature extraction technique is Linear Predictive Coding (LPC), which models the vocal tract as an all-pole filter. LPC coefficients represent the spectral envelope of the speech signal and are computed by minimizing the prediction error.

The prediction error $e(n)$ is given by:

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \quad (2)$$

where $s(n)$ is the speech signal, a_k are the LPC coefficients, and p is the order of the LPC analysis as discussed by Makhoul [16] and Rabiner et al., [17]). The coefficients a_k are found by solving the Yule-Walker equations.

2.2.2.3 Spectrograms and CNNs

Recent advancements in deep learning have led to the use of spectrograms as input features for CNNs. A spectrogram is a visual representation of the spectrum of frequencies in a signal as it varies with time. It is computed by applying the STFT to the signal:

$$X(t,f) = \sum_{-\infty}^{\infty} x(n) w(n-t) c^{-j2\pi f n} \quad (3)$$

where x(n) is the input signal, $w(n)$ is the window function, and $X(t,f)$ represents the magnitude of the STFT at time t and frequency f proposed by Allen [18] and Hinton et al., [19] . The resulting spectrogram can be fed into a CNN for feature extraction and classification.

2.2.2.4 Deep Learning-Based Feature Extraction

Deep learning techniques have been employed to automatically learn features from raw audio data. Convolutional layers in CNNs can learn local patterns in the spectrograms, while recurrent layers in RNNs can capture temporal dependencies. Pre-trained models such as Wav2Vec and DeepSpeech have shown significant improvements in AASR by leveraging large amounts of unlabeled data to learn robust representations.

Muhammad et al. demonstrated the effectiveness of transfer learning in improving AASR performance for South Asian accents using DeepSpeech2. Their approach involved fine-tuning a DeepSpeech2 model on accented data, resulting in a significant reduction in Word Error Rate (WER) compared to baseline models in a work proposed by Muhammad et al., [3].

2.2.2.5 Layer-Wise Embedding

Layer-wise embedding is another advanced technique used for feature extraction in AASR. This method involves extracting features from different layers of a neural network trained on speech data. The embeddings from various layers can capture different levels of abstraction in the speech signal, which is particularly useful for handling accent variations. The mathematical formulation for layer-wise embedding is given by:

$$E_l = f_l(X) \quad (4)$$

where X is the input feature vector, f_l is the function representing the l^{th} layer of the neural network, and E_l is the embedding output from the l^{th} layer. This technique was effectively used by Huang et al., [9] for accent classification.

2.2.2.6 Phone Posteriorgrams

Phone posteriorgrams are another feature representation used in AASR, where the posterior probabilities of phonetic units (phones) are computed for each frame of the speech signal. These probabilities are used as features for further processing. The posterior probability of phone p given the acoustic feature $x\mathbf{x}$ is computed using:

$$P(p|x) = \frac{e^{sp(x)}}{\sum_j e^{sj(x)}} \quad (5)$$

where $s_p(x)$ is the score assigned to phone p by the acoustic model, and the denominator is the sum of the scores for all possible phones, ensuring the probabilities sum to 1. This method was employed by Huang et al. for accent identification, resulting in improved performance on accented speech recognition tasks proposed by Huang et al., [9].

2.2.2.7 Mel Spectrogram

The Mel Spectrogram is a widely used method for feature extraction in speech processing. It transforms the raw audio signal into a visual representation that emphasizes frequencies relevant to human auditory perception. Mathematically, the

Mel Spectrogram S(f,t) is computed by applying the Short-Time Fourier Transform (STFT) to the audio signal $x(t)$, followed by a Mel filterbank transformation.

$$S(f,t) = log\ \left(\sum_{k=1}^{N} |X_k(f)|^2\ H_k(f)\right) \quad (6)$$

where $X_k(f)$ represents the magnitude of the Fourier Transform of the signal in the k^{th} frequency band, and $Hk(f)$is the Mel filterbank response. [14].

2.2.2.8 Tempogram

The Tempogram, primarily utilized in music analysis, has garnered attention in ASR for capturing temporal patterns and rhythmical aspects in speech signals. Research by Ellis et al.,[171] demonstrated the application of Tempogram-based features for speech rhythm analysis, showcasing its potential in ASR tasks requiring prosodic information.

The Tempogram is computed by calculating the autocorrelation of the onset strength envelope of the audio signal. The equation for Tempogram $T(t)$ can be represented as follows [178]:

$$T(t)=ACF(OnsetStrength(t)) \quad (7)$$

Where:

- $T(t)$ represents the Tempogram at time t.
- ACFdenotes the autocorrelation function.
- $OnsetStrength(t)$ represents the onset strength envelope of the audio signal at time t.

2.2.2.9 Zero Crossing Rate (ZCR)

ZCR, a simple yet effective feature, measures the rate of sign changes in the speech signal. Tzanetakis et al., [173] employed ZCR for musical genre classification, highlighting its relevance in capturing temporal characteristics. In ASR, ZCR serves

as a fundamental feature for speech activity detection and segmentation. ZCR measures the rate at which the audio signal changes its sign, offering insights into its temporal characteristics and periodicity. It is computed as the number of times the signal crosses the zero-amplitude threshold divided by the signal's duration [173, 175]. This can be computed by using the equation:

$$ZCR = \frac{1}{T}\sum_{I=1}^{N-1} I(x(i).x(i+1) < 0) \quad (8)$$

where $x(i)$ represents the signal amplitude at sample i, N is the total number of samples, and $I(\cdot)$ is the indicator function.

2.2.2.10 Root Mean Square Value (RMS)

RMS quantifies the average energy of the audio signal and is calculated as the square root of the mean of the squared values of the signal samples. It the average energy of speech signals and serves as a robust feature in ASR. Ellis et al., [171] demonstrated the effectiveness of RMS for noise robustness in ASR systems, enhancing speech intelligibility and recognition accuracy. This can be computed as:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2} \quad (9)$$

where $x(i)$ represents the signal amplitude at sample i [171, 176]. RMS facilitates noise robustness in ASR, aiding in noise suppression and improving recognition accuracy.

2.2.2.11 Tonnetz & Polyfeatures

Tonnetz and Polyfeatures are sophisticated feature extraction methods adept at capturing tonal and harmonic characteristics in speech signals, often employed in music information retrieval tasks [174, 177]. Tonnetz and Polyfeatures capture tonal and harmonic characteristics in speech signals, offering valuable insights into the spectral properties of speech. In ASR, they assist in modeling phonetic content, pitch variations, and harmonic structures, contributing to accurate speech recognition and phonetic classification. Tonnetz and Polyfeatures aid in modeling spectral properties

and harmonic structures of speech, facilitating accurate phonetic classification and improving ASR performance, particularly in tonal languages [174].

2.2.2.12 Harmonic Mean Ratio (HMR)

HMR assesses the spectral shape and harmonic content of the signal by measuring the ratio of harmonic mean to arithmetic mean of signal magnitudes [179]. The Harmonic Mean Ratio (HMR) measures the ratio of the harmonic mean to the arithmetic mean of signal magnitudes, providing insights into the spectral shape and harmonic content of the signal. The equation for HMR can be expressed as:

$$HMR = \frac{N}{\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|X_i|}} \quad (10)$$

where Xi represents the magnitude of the i^{th} frequency component and N is the number of frequency components. [179].

These feature extraction techniques are integral to the development of robust and accurate ASR systems. The Mel Spectrogram offers a comprehensive spectral analysis that is essential for phonetic recognition. The Tempogram's rhythmic insights and ZCR's temporal characteristics complement each other in improving the temporal resolution of ASR. RMS provides a measure of signal power, crucial for differentiating speech from noise. Tonnetz and Polyfeatures enhance harmonic and tonal analysis, which is vital for capturing the nuances of speech. The HMR's focus on spectral shape and harmonic content offers a deeper understanding of the phonetic structure, making it invaluable for improving ASR performance. The combined use of these techniques results in a more holistic and effective approach to speech recognition, as evidenced by various studies in the field.

Through the STFT, the signal is dissected, enabling it to scrutinize its relative strength via the Fourier analysis method. To optimize this analysis, the signal is partitioned into diminutive frames that overlap, ensuring comprehensive coverage. A Fast Fourier Transform (FFT) will then be applied to each of these frames [16,17]. From this process, twelve spectral features can be derived from each speech sample.

Benzeghiba et al.,[97] discusses different sources of speech variation, and the paper provides an overview of existing literature and features weaknesses in feature extraction or modeling approaches. Li, W et al.,[105] discusses the utilization of multi-source information, considering acoustic features extracted from native references and linguistic information. Zhou et al.,[108] introduced a text-to-speech (TTS) synthesis for constructing AASR, particularly in cases with restricted training data. The focus was on accent identification through the incorporation of phonetic and prosodic variations. The authors in their work suggested an accented front-end designed for grapheme-to-phoneme conversion and an AASR that integrated pitch and duration predictors for predicting phoneme-to-Mel-spectrogram.

Li, Y et al., [109] in their study capture language-specific prosody features and shared emotional expressions across languages using a pre-trained self-supervised model (HuBERT), and hierarchical emotion modeling is utilized to encompass a broader range of emotions across different languages.

In the context of emotion classification described by Zhang et al. [110], the system identifies both the type and intensity of emotions from the Mel-spectrogram of input speech. Intensity is gauged by the posterior probability of the conveyed emotion in the input utterance. For Multilingual Deep Neural Network (DNN) training, Convex Nonnegative Matrix Factorization (CNMF) is employed for feature vectorization process. Utilizing LSTM-RNN proves more effective in acoustic models proposed by El-Moneim et al., [137]. The study conducted by Veisi H et al. [141] proposes a more refined acoustic model can be developed by merging Deep Bidirectional Long Short-Term Memory -DBLSTM with a Deep Belief Network -DBN. The DBN comprises multiple Restricted Boltzmann Machines (RBM), aiding in feature extraction from the speech signal.

The research proposed by Dupont, S. et al. [142] suggests techniques to improve the adaptability of ASR technology for diverse tasks and languages. The authors in their paper discuss a novel approach to categorize the speech signal into phonetic units

that are language-independent by employing non-linear discriminant model. The paper also discusses various feature extraction and speech vectorization techniques by employing Perceptual Linear Prediction (PLP) coefficients, and the study addresses additive and convolutional noise through a combination of spectral subtraction and temporal trajectory filtering.

Liu,S. et al.[151] proposed neural network architectures and feature augmentation methods for addressing disordered speech datasets. Mukhamadiyev et al. [152] in their study proposed neural network architectures and utilized hybrid connectionist temporal classification in modeling dialectal variations in the Uzbek language. Gammatone-frequency cepstral coefficients (GFCC) and Log frequency spectral coefficients (MFSC) and with their first and second-order derivatives were used for feature vectorization in the study conducted by Abdelmaksoud et al. [153].

Techniques such as MFCCs, LPC, spectrograms combined with CNNs, deep learning-based feature extraction, layer-wise embedding, and other feature extraction techniques like Tempogram and STFT have been extensively researched and applied. These methods help capture the essential characteristics of speech signals while being robust to the variations introduced by different accents. The advancements in these techniques have significantly contributed to the progress in AASR, enabling more accurate and robust speech recognition systems that can handle diverse accent variations.

2.3 Advances in AASR Across Different Languages

There have been significant advancements in the research in speech technology including constructing ASR, AASR, accent identification, speaker identification, language identification and SER. The research has advanced in languages like English Arabic, Japanese and Chinese in the last decade. The study conducted by Solomon Teferra et al. [20] proposed methods for modeling Ethiopian dialects utilizing DNN-based frameworks. This work illustrates that DNN-based ASR outperformed GMM-based architectures.

The work proposed by El-Moneim et al. [21] presented a speaker identification system utilizing the feature vectors acquired by employing MFCCs, range, and log-range. The authors demonstrated an approach that incorporated LSTM-RNN classifier, that positively classified the speech signals. In their research, Palaz et al. [22] conducted a study that compared the performance of CNN-based approach and the ANN-based approach.

Anandhu Sasikuttan et al., [23] in their study proposed an ASR system for Malayalam language that can recognize combinations of formal Malayalam words pronounced with gaps between them. The authors put forward a simple and user-friendly interface, designed primarily for illiterate individuals. Ossama Abdel-Hamid et al., [24] in their study discusses the use of CNN for building ASR. The authors also propose a constrained weight-sharing design for better modeling of speech features. Issa et al.,[25] introduced methods for SER utilizing a 1D DCNN with a combination of five different audio features. The authors proposed model architectures that directly operate on acoustic data without transforming the signals into spectrogram representations. Passricha, V. et al., [26] discuss a CNN-based architecture for modeling ASR. This model is trained with essential representation of the speech signal in a data-driven manner and calculates the conditional probability for each phoneme class.

Andrew Senior et al., [27] proposed an LSTM-RNN architecture for constructing AASR, that performed better than the standard LSTM networks and DNNs since the authors constructed the ASR by optimizing the parameters. Yi, J et al., [28] in their paper discusses an adaptation method for ASR in multi-accented Mandarin speech, constructing an ASR model with LSTM-RNN using the connectionist temporal classification loss function. The authors achieved better results without overfitting the model by introducing a regularization layer after the training process.

Kishori R. Ghule et al., [29] in their research created a phonetic database for isolated words in Marathi and developed an ASR system for the language. The authors

highlighted that the artificial neural networks provided optimal results for ASR construction. Shanthi et al., [30] discusses the approach they adopted for constructing isolated word speech recognition in the Tamil language by employing Hidden Markov Model - HMM method. The speech signals were vectorized using MFCC algorithm and adopted a triphone-based acoustic model specifically for Tamil digits, achieving the accuracy rate of approximately 90 percent.

Radzikowski, et al [31,32,33] proposed techniques to modify the accent of non-native speakers to closely resemble that of native speakers. Spectrograms, graphical representations of speech signals, were employed in the experimentation. The study demonstrated better results when autoencoder based on CNN was employed.

Investigations on the use of ensemble methods on various dialect-specific acoustic models for recognition were driven by the success of dialect-specific models. Dokuz et al., [34] illustrated hybrid techniques that combine gender and accent information from speech databases. According to their experimental findings, combining features for gender and accent is more effective than utilizing only one factor alone for speech recognition.

Alsharhan et al., [35] demonstrated that creating gender- and dialect-specific models results in a significant reduction in WER. The proposed method was efficient in getting around the data's limited availability, shortening training time, and getting the best performance. Kumar, A. et al. [36] incorporated hybrid architecture for acoustic modeling that combines SincNet, Convolutional Neural Networks (CNN), and Light Gated Recurrent Units (LiGRU), which exhibit improved interpretability, high accuracy, and smaller parameter sizes.

Injy Hamed et al., experimented with constructing ASR systems for code-switched Egyptian Arabic–English ASR using DNN-based hybrid and Transformer-based end-to-end models [37]. Shi, X et al. [38] experimented with 160 hours of English speech with accents from 8 different nations and illustrated the publicly available

dataset, track settings, and baselines. Centin O. et al. [39] adopted spectrogram features for constructing an accent-based acoustic model using the CNN method.

Aksënova et al.,[40] discussed current developments in constructing ASR systems for accented speech and measured the impact of wav2vec for pre-training on accented speech recognition, identifying relevant corpora for accented speech, and different ASR assessments. Zeng et al. [41] investigated geographically proximate accent classification tasks. They compared various accent modeling approaches and classifiers and proposed a general workflow for forensic accent classification. The Common Voice corpus serves as an extensive multilingual repository of transcribed speech, intended for research and development in speech technology. While primarily designed for Automatic Speech Recognition applications, Common Voice's utility extends to other domains, such as language identification. To ensure scalability and sustainability, the project adopts crowdsourcing for both the collection and validation of data [99].

Enhanced recognition performance on multi-accent data, encompassing native, non-native, and accented speech by utilizing [102] untranscribed accented training data through semi-supervised learning. Han T et al., [103] explores three data augmentation techniques, namely noise injection, spectrogram augmentation, and TTS-same-sentence generation and demonstrates that contrastive learning plays a crucial role in constructing data-augmentation invariant and pronunciation invariant representations. Klumpp, P et al., [104] explores transforming native US-English speech into accented pronunciation and explores the feasibility of learned accent representations understanding of speech from both seen and unseen accents in evaluations on native and non-native English datasets. Gutscher, L. et al. [110] proposes a text-to-speech (TTS) system specifically designed for under-resourced language varieties spoken in Austrian regions.

The reviewed literature points out the rapid progress and diverse methodologies employed in the field of speech technology. Continued research efforts in areas such

as accent recognition, dialect-specific modeling, data augmentation, and multilingual speech processing are essential for further advancing the capabilities of speech recognition systems and making them more inclusive and accessible across diverse linguistic communities.

2.4 AASR using Autoencoders

Accented speech recognition poses unique challenges in accurately understanding and interpreting spoken language due to variations in pronunciation, intonation, and phonetic characteristics across different accents. Traditional methods for speech recognition often struggle to handle such variations, leading to degraded performance and reduced accuracy.

Autoencoders offer a powerful approach to learning robust and discriminative representations directly from raw speech data, without the need for explicit feature engineering. It compresses the input speech signals into a lower-dimensional latent space and then reconstructs them. By training the autoencoder to minimize the reconstruction error, it learns to extract salient features that capture the essential information for accurate speech recognition. Accented speech recognition using autoencoders involves training the autoencoder model on a diverse dataset that includes speakers with various accents. This enables the model to learn accent-invariant representations by capturing the underlying shared characteristics of speech, while also accounting for the specific accent variations. Some of the relevant works in the literature are discussed below.

Sahu et al., [42] in their work addresses the limitations of traditional feature extraction methods by employing the power of autoencoders to learn compact and discriminative representations directly from raw speech signals. They introduce an adversarial training framework where a generator network, implemented as an autoencoder, is pitted against a discriminator network. Lee et al.,[43] introduce a novel approach that utilizes chain-based discriminative autoencoders for speech recognition. It highlights the benefits of incorporating contextual information and

discriminative criteria in the autoencoder framework, leading to enhanced performance in speech recognition tasks.

Deng et al., [44] propose an approach to speech emotion recognition using semi-supervised autoencoders. The authors address the challenge of limited labeled data in emotion recognition tasks by using unlabeled data to enhance the performance of the emotion recognition model. They propose a semi-supervised learning framework that combines the power of autoencoders with limited labeled data and a large amount of unlabeled data.

Karitha et al.,[45] in their work present a semi-supervised learning approach that combines text-to-speech synthesis and autoencoders for ASR. It showcases the benefits of using synthesized data to augment the labeled data, enhancing the performance of the speech recognition model. Huang et al.,[46] in their work introduce masked autoencoders with attention mechanisms as a powerful tool for speech recognition tasks. By allowing the model to selectively attend to relevant acoustic segments, the proposed approach improves the model's ability to capture fine-grained details and enhances speech recognition performance. Atmaja et al.,[47] explores the potential of self-supervised learning techniques to learn informative representations from unlabeled data, leading to improved performance in emotion recognition tasks.

Peng et al.,[48] present an autoencoder-based feature-level fusion technique by combining multiple acoustic features through autoencoder-based representations, the proposed approach effectively captures emotional information and improves the performance of SER systems.

Bastanfard et al.,[49] present a stacked autoencoder-based approach for speech emotion recognition in the Persian language. By comparing local and global features, the study highlights the importance of considering different feature types in capturing emotional information from speech signals. The proposed method shows promising results in recognizing emotions from Persian speech data.

Ying et al.,[50] propose an unsupervised feature learning approach using autoencoders for speech emotion recognition. The Accented English Speech Recognition Challenge (AESRC2020) served as a pivotal testbed for accent-related research, offering two distinct tracks: accent recognition and accented English speech recognition [51]. The challenge featured a comprehensive dataset comprising 160 hours of accented English speech from eight countries, and a 20-hour unlabeled test set with previously unseen accents from two additional countries. Participants used a variety of techniques and models to address this formidable challenge, including pre-trained encoders, phonetic posteriorgrams (PPG), diverse network architectures, and unsupervised training methods with contrastive and diversity losses by learning discriminative representations directly from raw speech signals, the proposed method eliminates the need for handcrafted features and achieves better performance in emotion classification tasks.

Conventional cascaded ASR system [63] addressed accents through changes to the acoustic model [64] or the pronunciation dictionary [65,82,83]. [101] proposes English speech recognition across various accents, aiming to evaluate the model's adaptability to unseen accents. The paper also discusses accent-agnostic approach that extends the model-agnostic meta-learning (MAML) algorithm for swift adaptation to accents not encountered during training [84,85,86,101].

The study conducted by Cao, Y et al. [147] proposed bilingual phonetic posteriorgram (PPG) based cross-lingual speech synthesizer that serves as a bridge across speakers and languages, constructed by stacking two monolingual PPGs from independent speech recognition systems. Liu, Y. et al. [149] proposed a codec network utilizing vector-quantized auto-encoders with adversarial training (VQ-GAN) to extract intermediate frame-level speech representations, passing from conventional representations like Mel-spectrograms.

The exploration of autoencoder-based approaches in accented speech recognition and related tasks represents a significant stride towards overcoming the challenges

posed by diverse linguistic variations. By harnessing the power of autoencoders to extract discriminative representations directly from raw speech data, researchers have demonstrated notable improvements in speech recognition accuracy, emotion recognition, and accent-invariant feature learning.

The reviewed literature showcases a diverse range of applications of autoencoders in speech technology, including speech emotion recognition, text-to-speech synthesis, and feature-level fusion techniques. The utilization of adversarial training frameworks, semi-supervised learning, and attention mechanisms has further enhanced the capabilities of autoencoder-based models, enabling them to capture fine-grained details and contextual information essential for accurate recognition tasks.

Additionally, the investigation of autoencoder-based approaches in cross-lingual speech synthesis and codec network architectures highlights the versatility and adaptability of these models across different languages and speech processing tasks. By utilizing autoencoder-based representations, researchers have paved the way for more robust and efficient speech recognition systems capable of handling diverse accents and linguistic variations.

The integration of autoencoder-based techniques represents a promising direction in the ongoing pursuit of advancing speech technology. Future research endeavors may continue to explore novel architectures, training methodologies, and applications of autoencoders to further enhance the performance and applicability of speech recognition systems in real-world scenarios.

2.5 AASR using ML and DL Approaches

CNNs, or Convnets, represent one of the oldest and most widely embraced neural network architectures in deep learning [117, 118]. Renowned for its remarkable advancements, CNN has achieved significant performance in Automatic Speech Recognition (ASR) through optimized training and advanced architectural

enhancements [119]. CNN models are composed of various layers including convolution, pooling, and fully connected layers [120, 121].

The CNN architecture can be represented as follows:

$$y = f(W * x + b) \quad (11)$$

Where: y represents the output of the CNN model, which consists of probabilities or scores for each phoneme or word class, W represents the learnable convolutional filters, $*$ represents the convolution operation, b represents the bias term, f represents the activation function, such as ReLU or softmax.

This equation captures the basic operation of a CNN model in AASR, where the input audio signal x is convolved with learnable filters W, followed by a bias term b. The resulting feature maps are then passed through an activation function f to produce the final output y, which represents the predicted phonemes or words as proposed by LeCun et al., [170].

In the context of CNNs, "convolution" refers to the mathematical function applied to the input image using a specific filter size, and the initial layer of the CNN, extract features from the input audio signals [122]. Different activation functions like ReLu, Tanh, step, sigmoid, and softplus are employed in CNNs for optimization purposes [123]. CNNs excel in handling spatial pixel information and time domain tasks, making them a suitable choice for addressing speech-related challenges, and many audio classification models. [124].

Emotions like fear, sadness, anger, surprise, etc., are effectively identified and classified in speech signals using CNN, outperforming KNN [125]. Abdel Maksoud et.al [156] proposed CNN architecture for modeling Arabic speech and obtained higher performance than the baseline models.

CNNs remain a cornerstone in the domain of deep learning for ASR, with their robust architectures and optimized training techniques consistently driving advancements

in speech recognition technology. As research continues to evolve, CNNs are expected to play a pivotal role in further enhancing the accuracy, efficiency, and adaptability of ASR systems to meet the demands of diverse real-world applications.

Utilizing RNN enables the handling of sequential data by processing the current input and retaining information from the previous input [126, 127, 128]. Many researchers are motivated to enhance the capacity to capture longer context by customizing the standard RNN [129]. The V-RNN (Variant of RNN) serves as a classifier for detecting ASR errors [130]. LSTM is introduced as a model capable of learning long sentence dependencies by effectively retaining and recalling information [117, 130, 131, 132].

A Deep Neural Network (DNN) refers to an artificial neural network containing two or more hidden layers positioned between the input and output layers [133]. RNN has established itself as a potent artificial neural network architecture designed for processing sequential data, particularly in the context of time series analysis [134].

LSTM is well-adapted for classifying time-series or sequential input, including applications in speech recognition, video analysis, and text processing. Featuring a memory cell, LSTM stores information from previous timesteps, allowing it to infer details not only from the present but also from past events [135, 131, 136, 137]. GRU, like LSTM, incorporates two gates (update and reset), enhancing its ability to memorize historical data [138].

The Bidirectional Recurrent Neural Network (BRNN) architecture serves as an alternative to address limitations in processing inputs strictly in a temporal manner. A notable feature of BRNN is the incorporation of two separate RNNs that process information in both forward and reverse time orders. Advanced RNN architectures such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are employed to overcome the vanishing gradient problem encountered by standard RNNs [139].

The language model based on LSTM demonstrated higher accuracy in comparison to the simpler RNN-based language model [141]. Automatically assigning optimal parameters through spectral differences enhances statistical model-based speech recognition systems but incurs a slight increase in computation load [142]. Employing an RNN model with an attention mechanism, such as segment boundary detection-directed attention, offers several advantages for online end-to-end speech recognition systems [143].

The utilization of RNNs, including variants such as LSTM and GRU, has significantly advanced the field of sequential data processing, particularly in applications like speech recognition. These architectures excel in capturing long-term dependencies and contextual information from sequential data, making them indispensable for tasks where temporal context is crucial. Additionally, Bidirectional RNNs offer an effective approach to processing sequential data by incorporating information from both past and future time steps. Advanced RNN architectures like LSTM and GRU reduce challenges such as the vanishing gradient problem, ensuring more stable training and improved performance. The superiority of LSTM-based language models over simpler RNN-based models underlines the importance of employing sophisticated architectures for language processing tasks. Integrating attention mechanisms into RNN models further enhances their capabilities, particularly in tasks requiring online, real-time processing. Overall, the continued advancement and refinement of RNN architectures holds immense promise for the future of sequential data analysis and applications in diverse domains. Goodfellow et al., [163] describes that the deep learning models, such as CNNs and RNNs, are particularly good at extracting hierarchical features and capturing temporal dependencies in speech data.

In the area ASR, machine learning algorithms have demonstrated remarkable versatility, contributing to advancements in acoustic modeling, speaker recognition, and feature extraction tasks. Multilayer Perceptron (MLP) models, as highlighted by Graves et al., [198], have been pivotal in ASR systems for their ability to learn

complex mappings between acoustic features and phonetic units. Similarly, K Nearest Neighbor (KNN) classifiers, as discussed by Reynolds et al., [199], have found application in acoustic modeling and speaker recognition tasks, showcasing their adaptability in ASR domains.

Moreover, stochastic gradient descent (SGD) optimization techniques, as noted by Hinton et al., [200], play a crucial role in training deep neural networks for ASR, facilitating the iterative minimization of loss functions. Support Vector Machine (SVM) classifiers, as demonstrated by Campbell and Sturim [201], have been utilized for speaker recognition and feature extraction, underscoring their significance in ASR research.

Ensembled models, including AdaBoost and Gradient Boosting, as described by Dahl et al., [202], have been instrumental in combining multiple classifiers to enhance ASR performance across various tasks. Additionally, decision trees and random forest classifiers, as outlined by Reynolds and Rose [203], have been employed for phoneme recognition and acoustic modeling, illustrating their effectiveness in ASR applications. Studies conducted by Dietterich [245] have shown the effectiveness of ensemble methods in improving model performance by combining the strengths of individual models. Opitz & Maclin [246] provided a theoretical foundation for ensemble methods in machine learning, discussing the benefits of combining diverse models to improve accuracy and robustness. The ensemble approach utilizes the diverse predictions of these algorithms to improve overall accuracy and robustness.

Dehl et al., [240] in their paper discusses that MLPs have been widely used in ASR for their ability to learn complex patterns in data. They are often employed in deep learning architectures such as deep neural networks for acoustic modeling in ASR systems. Bengio et al., [241] in their work proposes that KNN can be applied in acoustic modeling and phoneme recognition tasks, particularly in scenarios with limited training data. Breiman [242] describes that Decision trees and Random Forest classifiers are employed in various stages of ASR pipelines, including feature

extraction and classification. They offer interpretable models and can handle both categorical and continuous data.

The study conducted by Bautista et al. [261] focuses on automatic speech emotion recognition (SER) using parallel CNN-Attention networks trained on the Ryeson Audio-Visual Dataset of Speech and Song (RAVDESS). The approach combines CNN-based networks and attention-based networks running in parallel to model spatial and temporal features. According to Zhao et al. [262], a novel approach for discrete speech emotion recognition (SER) was proposed, combining a parallel 2D CNN with a self-attention Dilated Residual Network.

The fusion of CNNs and RNNs has revolutionized the landscape of ASR technology. CNNs excel in extracting features from input audio signals, making them ideal for addressing speech-related challenges and emotion classification tasks. On the other hand, RNN variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are adept at capturing long-term dependencies and contextual information crucial for sequential data processing. These architectures, along with traditional machine learning algorithms like MLPs and SVMs, have significantly contributed to advancements in acoustic modeling, speaker recognition, and feature extraction tasks within the ASR domain. Furthermore, ensembled models and decision tree classifiers have been instrumental in enhancing ASR performance across various tasks. The continuous evolution and refinement of these architectures holds immense promise for further enhancing the accuracy, efficiency, and adaptability of ASR systems to meet the demands of diverse real-world applications.

2.6 Accent-Neutral ASR

These methodologies compel the model to focus solely on the underlying content of speech, disregarding accent information. Earlier studies utilizing this approach employed adversarial training [65] or similarity losses. The study conducted by Najafian et al. [66] investigates accent compensation techniques for enhancing the performance of Hidden Markov Model based ASR systems. The findings emphasize

the effectiveness of the DNN system, showcasing enhanced performance even after accent-dependent acoustic model selection via Accent Identification (AID) and subsequent speaker adaptation.

The research highlights the optimal conditions for maximizing the average performance of the DNN system across diverse accent groups. The application of domain adversarial training, with the discriminator functioning as an accent classifier, has demonstrated notable enhancements over standard ASR models [67]. Further advancements have been achieved through pre-training the accent classifier [68] and accent relabeling based on clustering [69].

The exploration of employing generative adversarial networks for this task has also been pursued [70]. In contrast to explicitly adopting domain adversarial methods, alternative accent-agnostic approaches utilize cosine losses [71] or contrastive losses [72,73] to enforce accent-neutral model outputs, ensuring that representations are similar for inputs with the same underlying transcript. Liu, S. et al., [107] proposes accent conversion from non-native-accented utterances in real-time without requiring any native-accented utterances. Zhu, X et al. [114] proposed a Multilingual Emotional TTS (METTS) model for the cross-speaker and cross-lingual emotional transfer.

Liu. C et al. [148] proposed a multilingual speech synthesis approach that eliminates the need for pronunciation dictionaries in target languages. Cong, J., et al. [150] proposes a combination of Variational Autoencoder (VAE) and Generative Adversarial Network (GAN) to directly learn a latent representation from speech. Zhang et al. [152] proposed a cross-lingual neural codec language model designed for cross-lingual speech synthesis.

Advancements in accent conversion and multilingual speech synthesis highlight the potential for cross-speaker and cross-lingual emotional transfer, paving the way for more inclusive and adaptable speech processing systems. As research in this field continues to evolve, innovative methodologies utilizing generative adversarial

networks, variational autoencoders, and neural codec language models hold promise for addressing accent-related challenges and advancing the capabilities of ASR technology in diverse linguistic environments.

2.7 Accent-Aware ASR

Accent-aware ASR methodologies entail enriching the model with supplementary information regarding the accent present in the input speech. Earlier studies in this domain concentrated on utilizing the multi-task learning (MTL) paradigm [74, 75,14], concurrently training accent-specific auxiliary tasks alongside ASR.

Various types of embeddings, such as i-vectors [15, 76], dialect symbols [77], embeddings derived from TDNN models [5], or from wav2vec2 models trained as classifiers [8, 78], have been explored for accented ASR. Several uncomplicated techniques for integrating accent information into input speech have been examined, including summation [5,78,79], weighted summation [8], or concatenation [77,78].

Additionally, some studies examine the integration of both accent-aware and accent-agnostic techniques within a unified model [80]. Prabhu et al., [81] proposed method for adapting accents in end-to-end ASR systems that involves employing cross-attention alongside a set of trainable codebooks, which capture accent-specific information and seamlessly integrate into the layers of the ASR encoder. There has been significant attention given to accent recognition in distinguishing various English accents [87, 88, 89, 90].

This area of study is also crucial for enhancing the generalization capability of Automatic Speech Recognition (ASR) models across different varieties of English accents. The field of accent recognition shares similarities with both speaker identification [91, 92, 93, 94] and language identification [95,96,97]. Catherine Anderson [98] discusses major areas of phonetics like phonology.

Chu, X. et al. [105] investigates human perception-inspired methods to enhance the recognition of accented speech. Zhang et al.,[106] proposed the integration of

embeddings from both a fixed acoustic model and a trainable acoustic model enhancing the robustness of language-related acoustic features. Zhou et al., [109] proposed a text-to-speech (TTS) system with target-accented speech data and then a speech encoder is trained to transform the accent of speech under the guidance of the pretrained TTS model. Ye, J.et al., [115] discussed methods for enhancing cross-lingual pronunciation learning by incorporating previously unseen content and speaker combinations during the training process.

Lee et al., [152] utilized vowel space analysis, to experiment the accents in cross-lingual Text-to-Speech (TTS) systems. Kim et al., [153] proposed unified representations of multilingual speech and text using a single model, with a particular focus on enhancing speech synthesis.

Accent-aware ASR methodologies represent a pivotal area of research aimed at enhancing model performance by incorporating supplementary information about the accent present in input speech. Multi-task learning paradigms have emerged as effective strategies for concurrently training accent-specific auxiliary tasks alongside ASR, enabling the model to better adapt to diverse linguistic contexts. Various types of embeddings, including i-vectors, dialect symbols, and embeddings derived from TDNN and wav2vec2 models, have been explored to enrich ASR systems with accent information. Techniques such as summation, weighted summation, and concatenation have been investigated for integrating accent information into input speech, while some studies have proposed unified models combining both accent-aware and accent-agnostic techniques.

2.8 Accent Unaware ASR

Numerous approaches have been explored in the development of accent-unaware ASR systems. Data augmentation techniques, adversarial training frameworks, multi-task learning models, and self-supervised learning methods are among the most prominent strategies. These techniques aim to create robust models that perform well irrespective of the accent variations in the input speech. Ko et al., [180]

in their study proposes techniques such as pitch shifting, speed perturbation, and noise addition are used to create a diverse training dataset, thereby helping the ASR system become robust to accent variations. Shinohara [222] proposes methods that use adversarial training that incorporates a discriminator network alongside the ASR model to encourage the model to produce accent-invariant features. This approach involves training the ASR system and the discriminator in a minimax game, where the ASR system tries to fool the discriminator into classifying accent-specific features as neutral.

Kim et al., [221] proposed multi-task learning frameworks to train ASR models to perform both speech recognition and accent classification simultaneously. By sharing the learned representations between these tasks, the ASR system can better generalize across different accents. Baevski et al., [220] proposed self-supervised learning methods that utilize large amounts of unlabeled speech data to learn useful representations. These pre-trained models are then fine-tuned on labeled data, which helps the ASR system perform well across various accents.

Datasets used for accent-unaware ASR systems typically include a diverse set of speech samples from speakers with different accents. The study conducted by Ardila et al., [219] discusses the Common Voice dataset by Mozilla includes recordings in multiple languages and accents, providing a rich resource for training and evaluating ASR models. Accent-unaware ASR research spans multiple languages, including English, Mandarin, Spanish, and regional languages like Hindi and Malayalam. The goal is to ensure that the ASR system can handle a wide variety of linguistic and phonetic variations introduced by different accents.

Accent-unaware ASR systems have demonstrated improved recognition accuracy across various accents. For instance, adversarial training has shown a significant reduction in word error rates (WER) for accented speech in the study conducted by Shinohara [222]. Multi-task learning frameworks have achieved better generalization

by using shared representations, resulting in more robust ASR performance as discussed in the study conducted by Kim et al., [221].

The primary advantage of these methods is their robustness to accent variations, which enhances the ASR system's overall performance. These approaches eliminate the need for accent-specific models, making the ASR system more versatile. Techniques like self-supervised learning reduce the reliance on large, labeled datasets, which are often scarce for specific accents. These methods also have their drawbacks. Techniques like adversarial training and multi-task learning add complexity to the model training process. Self-supervised learning methods require significant computational resources and large datasets for pre-training. Additionally, adversarial and multi-task learning frameworks require careful tuning to balance the learning objectives.

In conclusion, accent-unaware ASR systems adopt innovative methodologies such as data augmentation, adversarial training, multi-task learning, and self-supervised learning to improve robustness and generalization across different accents. These approaches have shown promising results in reducing WER and enhancing recognition accuracy for accented speech. The selected methodologies for this research work will build on these foundations, aiming to further advance the state-of-the-art in accent-unaware ASR.

2.9 Strategies for Data Set Preparation

Effective dataset preparation is essential for developing high-performing Automatic Speech Recognition (ASR) systems. The quality, size, and diversity of the dataset significantly influence the model's ability to generalize across different accents, dialects, and speaking styles. This section discusses various strategies for preparing datasets for ASR, emphasizing methods to enhance the robustness and accuracy of the models.

2.9.1 Data Augmentation

Data augmentation techniques are widely used to artificially increase the size of the training dataset. These methods help the model become more robust to variations in the input data. Common augmentation techniques include:

- **Noise Addition**: Adding background noise to the audio signals to simulate real-world environments. This method helps the model learn to distinguish speech from noise, improving its robustness as proposed in the study by Ko et al., [180].
- **Speed and Pitch Alteration**: The study conducted by Cui et al., [181] reveals that modifying the speed and pitch of the audio to create variations that can help the model generalize better across different speakers and accents.
- **Reverberation**: Hannun et al., [182] proposes that adding reverberation effects to mimic different room acoustics, which helps the model handle variations in recording environments.

2.9.2 Data Balancing

Ensuring that the dataset is balanced across different classes (e.g., accents, genders, age groups) is essential to prevent the model from being biased towards more frequent classes. Jaitly & Hinton [183] in their study proposes that techniques such as oversampling underrepresented classes or undersampling overrepresented classes are commonly used to achieve a balanced dataset.

2.9.3 Multi-Accent and Multi-Dialect Data

Rosenberg et al., [184] proposed that including data from multiple accents and dialects ensures that the ASR model can generalize well across different linguistic variations. This approach involves collecting speech data from speakers with diverse accents and dialects and can significantly improve the model's performance in real-world scenarios.

2.9.4 Transcription Quality

In the study conducted by Hirschberg & Manning [185] proposes that high-quality transcriptions are crucial for supervised learning in ASR. Manual transcriptions by native speakers are ideal but can be costly and time-consuming. Automated transcription tools, followed by manual verification, can be a cost-effective alternative.

2.9.5 Speaker Variability

Panayotov et al., [186] in their study discusses that incorporating a wide range of speakers in the dataset, including variations in age, gender, and speaking styles, helps the model generalize better. Speaker variability can be achieved by collecting data from diverse speaker populations.

2.9.6 Data Cleaning and Preprocessing

In a study conducted by Ravanelli et al., [187] investigates that preprocessing steps such as silence removal, normalization, and filtering out low-quality audio samples are essential to ensure that the dataset is clean and consistent. Data cleaning helps in reducing the noise in the dataset, leading to better model performance.

2.9.7 Use of Synthetic Data

Sisman et al., [188] proposes that generating synthetic speech data using Text-to-Speech (TTS) systems can supplement the training data, especially for underrepresented classes or languages. Synthetic data generation should be done carefully to ensure that it closely resembles natural speech.

2.9.8 Annotation Consistency

Panayotov et al., [186] states that consistent annotation guidelines are critical for ensuring that the transcriptions are uniform across the dataset. This consistency helps the model learn better from the training data.

The discussed strategies, including data augmentation, balancing, inclusion of multi-accent and multi-dialect data, and maintaining high transcription quality, are essential to enhance the generalization capability of ASR models. Incorporating speaker variability and preprocessing steps such as silence removal and normalization further ensures the cleanliness and consistency of the dataset. Additionally, the use of synthetic data and maintaining annotation consistency contribute to the overall quality of the dataset, thereby improving model performance. These methodologies collectively aim to prepare a comprehensive and diverse dataset, enabling the ASR system to perform effectively across different linguistic and acoustic variations encountered in real-world applications.

2.10 Generation of Spectrograms

Spectrograms are a fundamental tool in speech analysis, offering a visual representation of the spectrum of frequencies in a sound signal as they vary with time. This visualization technique is essential in many areas of speech processing, including Automatic Speech Recognition (ASR), as it provides detailed insights into the temporal and spectral characteristics of speech signals. A spectrogram represents audio signals in a three-dimensional format: time, frequency, and amplitude. The x-axis denotes time, the y-axis denotes frequency, and the color intensity or brightness indicates the amplitude of a particular frequency at a given time.

The generation of a spectrogram involves dividing the audio signal into short overlapping segments, called frames, and then applying the Short-Time Fourier Transform (STFT) to each frame. The STFT of a signal $x(t)$ is given by [191, 192]:

$$X(t,f) = \int_{-\infty}^{\infty} x(\tau)\,.\,w(\tau - t).\,e^{-j2\pi f\tau} d_\tau \qquad (12)$$

where $w(\tau - t)$ is a window function that limits the analysis to a short segment of the signal around time t, and f is the frequency variable. The magnitude squared of the STFT provides the spectrogram $S(t, f)$:

$$S(t,f) = |X(t,f)|^2 \quad (13)$$

Numerous studies have demonstrated the effectiveness of spectrograms in enhancing ASR performance. For instance, the use of Mel spectrograms has been shown to significantly improve the accuracy of deep learning-based ASR systems. Hannun et al., [182] utilized spectrograms in their Deep Speech model, achieving state-of-the-art performance in end-to-end ASR systems. Additionally, Amodei et al., [189] explored various spectrogram representations to improve ASR robustness under noisy conditions.

Studies like those by Sainath et al., [190] have integrated spectrogram-based features with convolutional neural networks (CNNs), utilizing the spatial structure of spectrograms to enhance feature extraction and recognition accuracy. The combination of spectrograms with advanced neural network architectures continues to drive improvements in ASR systems.

Spectrograms play a pivotal role in the field of speech processing and ASR. By providing a detailed representation of the time-frequency characteristics of speech signals, they facilitate the extraction of crucial features that enhance the performance of recognition systems. The integration of spectrograms with advanced neural network architectures continues to push the boundaries of ASR technology, making them an indispensable tool in both research and practical applications.

2.11 AASR of Malayalam Isolated Words

Accented Automatic Speech Recognition (AASR) for Malayalam isolated words is a critical research area due to the distinct phonetic and linguistic traits of the Malayalam language, compounded by the variations introduced by regional accents.

In exploring feature extraction techniques, researchers have employed various methods such as Mel-Frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC), and Perceptual Linear Predictive (PLP) coefficients by Nallasamy & Venkataraman [195]. These techniques play a pivotal role in capturing essential

speech characteristics necessary for distinguishing phonetic variations due to different accents. Typically, studies utilize recorded speech samples from native Malayalam speakers across diverse regions to encompass a wide range of accents. Notably, MFCC stands out as the most utilized technique due to its efficacy in capturing speech signal characteristics, despite potential limitations in fully capturing accent variations and computational intensity associated with LPC and PLP.

In a study conducted by Chakravarthy & Sitaram [193] deep learning models, particularly CNNs and RNNs, have emerged as prominent tools for learning accent-invariant features. These models are trained on large datasets comprising speech samples from speakers with diverse accents, focusing on Malayalam language. While deep learning models offer superior performance in accent-invariant feature learning, they demand substantial computational resources and extensive training datasets.

Adversarial training has gained traction as a method to enhance ASR robustness to accent variations. Sitaram et al., [197] proposes that by employing a discriminator network alongside the ASR model, this technique generates accent-invariant features, thereby improving recognition accuracy. However, adversarial training introduces complexity in model training and requires careful tuning of adversarial components. Multi-Task Learning approaches, involving simultaneous learning of ASR and accent classification tasks, utilize shared information to enhance ASR accuracy and robustness as proposed in their study by Balaji et al., [196]. While offering improved generalization and efficient data utilization this approach necessitates well-annotated datasets and intricate model design.

Data augmentation techniques, such as speed perturbation and noise addition, have emerged as simple yet powerful tools to enhance ASR robustness. By creating diverse training datasets simulating accent variations, data augmentation improves ASR performance in a study conducted by Jain & Venkatesh [194]. However, it may

introduce unnatural variations and require extensive computational resources. The reviewed methodologies accentuate the critical role of feature extraction, deep learning, adversarial training, multi-task learning, and data augmentation in addressing accent variations in Malayalam isolated word recognition. Integrating these techniques forms the basis for developing a robust and efficient AASR system for Malayalam.

Ensemble approaches have emerged as effective strategies for enhancing Automatic Accent-Aware Speech Recognition (AASR) systems by leveraging the diversity of multiple models to improve overall performance. Several studies in the literature have explored the application of ensemble methods in AASR, showcasing their effectiveness in mitigating the impact of accent variations.

2.12 Ensemble Approaches for AASR

One notable study by Zhang et al., [204] investigated the use of ensemble learning techniques, including AdaBoost and Gradient Boosting, to combine multiple accent-specific models for improved AASR accuracy. By aggregating predictions from individual models trained on different accent groups, the ensemble approach achieved superior recognition performance across diverse accent variations.

In a study conducted by Chen et al., [205] proposed a novel ensemble framework for AASR, which integrated multiple deep learning models, including CNNs, RNNs, and Transformer models. By combining the strengths of these diverse architectures, the ensemble system demonstrated robustness to accent variations and outperformed individual models on benchmark AASR datasets.

Another approach explored by Li et al., [206] involved ensemble learning at the feature level, where features extracted from multiple accent-specific models were combined to form a more comprehensive representation of the input speech signals. This feature-level ensemble approach effectively captured accent-related

information and significantly improved AASR performance, particularly in scenarios with limited training data for certain accent groups.

In summary, ensemble approaches in AASR have shown promise in mitigating the impact of accent variations by integrating the diversity of multiple models or features. These studies highlight the effectiveness of ensemble learning techniques in enhancing AASR accuracy and robustness, underscoring their potential for future research and practical deployment in real-world AASR systems.

2.13 AASR for Multisyllabic Words

In the studies conducted by Liang et al., [207] and Chen et al., [208] diverse datasets comprising multisyllabic word utterances spoken with various accents are employed to train deep learning models for AASR tasks. Multilingual datasets encompassing different languages, including English, Spanish, and Mandarin, are utilized for AASR research on multisyllabic words. Deep learning architectures have demonstrated promising results in improving AASR accuracy for multisyllabic words, with some studies reporting significant enhancements in recognition performance across different accent variations. Deep learning architectures offer a viable solution for AASR of multisyllabic words, exhibiting robustness and promising performance across various accents. These methods utilize the power of neural networks to capture accent-related features effectively, laying the groundwork for further advancements in AASR research.

The studies conducted by Wang et al., [209] and Sitaram et al., [210] proposes that adversarial training techniques involve training AASR models with accent discriminators to enforce the generation of accent-invariant features. These models aim to reduce the impact of accent variations on multisyllabic word recognition. Annotated datasets containing multisyllabic word utterances spoken with diverse accents are utilized for training adversarial AASR models. Multilingual datasets encompassing various languages and accents are employed to evaluate the effectiveness of adversarial training for AASR tasks. Adversarial training techniques

have shown promising results in improving AASR performance for multisyllabic words, effectively reducing the influence of accent variations on recognition accuracy. Adversarial training emerges as a viable approach for enhancing AASR performance for multisyllabic words by mitigating the effects of accent variations. These techniques offer a principled framework for training robust AASR models capable of recognizing multisyllabic words spoken with diverse accents.

The literature on AASR for multisyllabic words highlights the effectiveness of deep learning architectures and adversarial training techniques in addressing the challenges posed by accent variations. These methodologies offer promising roads for improving AASR accuracy and robustness in recognizing multisyllabic words spoken with diverse accents.

2.14 Fusion of Self-Supervised Learning, ML models and Autoencoders

The fusion of autoencoders with ML models represents a sophisticated approach to feature engineering and model optimization in classification tasks. By combining the feature learning capabilities of autoencoders with the discriminative power of ML algorithms, this fusion technique aims to extract meaningful representations from raw data while leveraging the strengths of different classifiers for enhanced predictive performance.

One advantage of this fusion approach lies in its ability to capture complex patterns and relationships within the data. Autoencoders, through unsupervised learning, learn to encode input data into a lower-dimensional latent space, capturing important features and reducing noise [252]. By incorporating these encoded representations into ML models, classifiers can benefit from the distilled information, leading to improved generalization and predictive accuracy [256].

The choice of ML algorithms used in this work is driven by their suitability for the task at hand and their compatibility with the encoded representations generated by

the autoencoder. SVMs are known for their ability to handle high-dimensional data and nonlinear relationships, making them well-suited for classification tasks with encoded features [257]. Similarly, Decision Trees offer interpretability and can effectively utilize encoded representations to make decisions based on learned features [258].

Ensemble methods like Random Forest take advantage of the diversity of decision trees to improve robustness and mitigate overfitting, further enhancing the fusion approach's effectiveness [242]. Neural network models such as MLPs are adept at learning complex patterns and nonlinear relationships, complementing the feature learning capabilities of autoencoders and leading to superior performance when used in conjunction with encoded representations [259].

By strategically selecting ML algorithms that complement the encoded features learned by the autoencoder, the fusion approach maximizes the utilization of available information and optimizes model performance. This thoughtful integration of feature learning and classification techniques emphasizes the importance of utilizing the strengths of both approaches to achieve superior predictive accuracy and model robustness in classification tasks [260]. In comparing the performance between utilizing autoencoder-trained features alone and the fusion of autoencoders with ML models, a clear distinction emerges in terms of predictive accuracy across various classifiers.

Different types of autoencoders are:

1. **Vanilla Autoencoders**: Vanilla autoencoders are commonly used in speech processing tasks due to their simplicity and effectiveness in learning latent representations of speech signals. They have been applied to various languages, including English, Mandarin, and Spanish, using datasets such as TIMIT in the study conducted by Garofolo et al., [225] and LibriSpeech in the study conducted by Panayotov et al., [226]. Vanilla autoencoders exhibit good reconstruction accuracy, making them suitable for tasks where preserving signal fidelity is

essential. However, they may struggle with capturing complex features in speech signals and are prone to overfitting when trained on limited data.

2. **Variational Autoencoders (VAEs)**: VAEs have gained popularity in speech processing for their ability to learn probabilistic latent representations of speech signals. They have been applied to languages such as English and German using datasets like the Common Voice dataset in the study conducted by Ardila et al., [227]. VAEs offer advantages such as the generation of new speech samples and robustness to noise and variations in speech signals. However, training VAEs can be challenging due to the need for careful design of the latent space distribution and the trade-off between reconstruction accuracy and latent space smoothness.

3. **Denoising Autoencoders**: Denoising autoencoders are effective in learning robust representations of speech signals by reconstructing clean speech from corrupted inputs. They have been applied to various languages, including English, French, and Chinese, using datasets such as the Noisex-92 dataset in the study conducted by Varga & Steeneken [228] and the Aurora dataset in the study conducted by Pearlmutter & Tzanetakis [229]. Denoising autoencoders offer advantages such as noise robustness and feature disentanglement. However, they may be sensitive to the choice of corruption methods and require careful tuning of hyperparameters.

4. **Adversarial Autoencoders**: Adversarial autoencoders combine the benefits of autoencoders and generative adversarial networks (GANs) for learning discriminative representations of speech signals. They have been applied to languages such as English and Chinese using datasets like the VoxCeleb dataset in the study conducted by Nagrani et al., [230]. Adversarial autoencoders offer advantages such as the generation of realistic speech representations and disentanglement of latent features. However, they can be computationally intensive to train and may suffer from mode collapse during training.

Hinton and Salakhutdinov [252] proposed a novel method for reducing the dimensionality of data using neural networks, which laid the foundation for training autoencoders. Vincent et al., [169] introduced denoising autoencoders as a method for extracting robust features from noisy data, which has been widely adopted in various machine learning tasks. Goodfellow et al., [163] provide comprehensive coverage of deep learning techniques, including autoencoders, making it a valuable resource for understanding the theoretical foundations and practical applications of these models. LeCun et. al., [170] present an overview of deep learning methodologies, highlighting the importance of autoencoders in learning efficient representations of data. Kingma and Welling [253] proposed the variational autoencoder framework, which combines autoencoding with Bayesian inference, offering a probabilistic interpretation of latent representations. Ranzato and Szummer [254] explored the use of deep networks, including autoencoders, for semi-supervised learning tasks, demonstrating the effectiveness of unsupervised pretraining in improving model performance.

Studies in this area have explored various methodologies, including training autoencoder networks using self-supervised learning techniques on large unlabeled speech corpora. These corpora contain recordings from speakers with diverse accents, enabling the autoencoder networks to learn invariant representations of speech signals as proposed by Smith et al., [211]. In the work proposed by Jones & Brown [212] the labeled datasets specific to the target language and accent are used for training and evaluation purposes, ensuring that the ASR system is optimized for recognizing accented speech in the desired language.

Results from these studies have shown promising improvements in ASR performance, particularly in scenarios with significant accent variations. Chen et al., [213] discusses a novel approach by utilizing self-supervised learning and autoencoders, these approaches have demonstrated enhanced robustness and accuracy in recognizing accented speech. Advantages of this approach include improved robustness to accent variations, efficient utilization of unlabeled data

through self-supervised learning, and the generation of informative feature representations by autoencoders.

However, there are challenges associated with this approach, including computational complexity and dependency on large datasets. Training autoencoder networks and integrating them into existing ASR systems can be computationally intensive, requiring substantial computational resources as proposed by Wang & Liu [214]. The effectiveness of self-supervised learning and autoencoders is highly dependent on the availability of large and diverse speech corpora for training as proposed by García et al., [215].

In their study, S. S. Khan et al., [255] proposed a novel scheme that combines Deep Autoencoder (DAE) with SVM for addressing security challenges in the era of industry 4.0. Their approach, tested on the NSL-KDD dataset, demonstrated significant advantages over baseline models, particularly in detecting low-frequency attacks. Through optimization techniques and evaluation of train and test times, S. S. Khan et al., [255] highlighted the efficacy of the DAE-SVM fusion approach for achieving both high accuracy and low-resource deployment. The fusion of AASR with self-supervised learning and autoencoders offers a promising avenue for enhancing ASR performance in the presence of accent variations. By using unlabeled data and learning meaningful representations of speech signals, these approaches address the challenges posed by diverse accents and improve the overall robustness of ASR systems.

2.15 Clustering Methods for Emotion Classification

Clustering methods for emotion classification of accented speech have emerged as a promising approach to accurately identify and classify emotions expressed in speech signals across different accents. Various clustering algorithms, such as K-means, Gaussian Mixture Models (GMM), and Hierarchical Clustering, have been applied to cluster accented speech data based on emotional content in the studies conducted by Lee et al., [216], Wu et al., [217] and Zhang et al., [218]. These methods aim to

group speech utterances into distinct clusters representing different emotional states. K-means clustering is commonly used to partition speech data into clusters based on feature similarity, with the number of clusters determined a priori or using optimization techniques. GMM clustering models the distribution of speech features using Gaussian components and assigns each data point to the most likely cluster based on posterior probabilities. Hierarchical clustering recursively merges data points into clusters based on proximity, forming a dendrogram that can be cut at different levels to obtain distinct emotional clusters.

Datasets containing labeled speech samples with emotional annotations are used for training and evaluating clustering models. These datasets encompass recordings from speakers with various accents expressing different emotional states. Accented speech data in various languages, including English, Mandarin, and Spanish, have been utilized for emotion classification using clustering methods.

Clustering methods have shown promising results in accurately grouping accented speech samples into distinct emotional clusters. These approaches have demonstrated the ability to effectively capture emotional variations across different accents. Clustering methods offer robustness to accent variations, as they can handle diverse accents and language variations in speech data. Additionally, clustering algorithms provide interpretable results, facilitating the interpretation of emotional patterns in accented speech. The challenges such as sensitivity to initialization and scalability issues persist. K-means clustering results may vary depending on the initial cluster centroids, leading to suboptimal solutions. Hierarchical clustering can be computationally intensive, particularly for large datasets, due to its recursive nature.

Clustering methods offer a promising approach to emotion classification of accented speech, using unsupervised learning techniques to identify distinct emotional clusters. K-means, GMM, and Hierarchical Clustering algorithms have been applied to effectively group accented speech data based on emotional content. These

methods demonstrate robustness to accent variations and provide interpretable results, despite challenges such as sensitivity to initialization and scalability issues.

Table 2 SER in Research

Ref.No.	Year	Methodology	Datasets	Key Outcomes
[49]	2023	Intelligent feature selection with GWO-KNN	Arabic Emirati-accented speech database, RAVDESS, SAVEE	Outperformed traditional methods for recognizing emotions in speech using Grey Wolf Optimizer and K-nearest neighbor classifier.
[50]	2023	Hybrid Features and CNN-Based Emotion Recognition	Emo-DB, SAVEE, and RAVDESS	A CNN model enhanced with MFCCT features achieved superior performance with accuracy rates of 97%, 93%, and 92% for respective datasets.
[51]	2022	Emotion Recognition in Urdu with K-nearest neighbors	Corpus collected from 20 subjects	Achieved 66.5% accuracy with K-nearest neighbors. Eliminating the "disgust" emotion led to a 76.5% accuracy improvement.
[52]	2019	CNN-Based Speech Emotion Recognition System	-	Attained an impressive accuracy of 83.61% on the test dataset, surpassing other methodologies.
[53]	2020	Comparative Study of SER Systems with Various Classifiers	Berlin and Spanish databases	Different classifiers yielded accuracies ranging from 83% to 94%, and feature selection techniques enhanced performance.
[54]	2020	Time-Frequency Features using Fractional Fourier Transform	Berlin EMO-DB, SAVEE, PDREC	Proposed method proficiently identified emotional classes with high accuracy across three datasets.
[55]	2020	Ensemble CNN Model with Multi-Channel EEG and Physiology	DEAP dataset	Enhanced accuracy and stability in emotion recognition using multi-channel EEG and peripheral physiological signals.
[56]	2020	Two-Stage Approach with Audio Features and Auto-encoder	-	Produced better results in comparison to other SER approaches.

Ref.No.	Year	Methodology	Datasets	Key Outcomes
[57]	2020	Dual-Level Model for SER with MFCC and Mel-Spectrograms	IEMOCAP dataset	Significantly outperformed baseline models and achieved comparable results with multimodal models.
[58]	2019	BLSTM and Attention Model for SER	-	Outperformed other approaches in terms of accuracy by utilizing speech segments with minimum duration and silence removal threshold.
[59]	2019	CNN LSTM Networks for Emotion-Related Features	Berlin EmoDB, IEMOCAP	Demonstrated excellent performance in recognizing speech emotion, surpassing traditional methods.

Table 2 illustrates the studies of existing work in literature that includes the methodology adopted, benchmark datasets used and the outcomes of the study.

The recent advancements in SER methodologies demonstrate significant improvements in performance using innovative approaches. Techniques such as hybrid features and CNNs have shown remarkable accuracy rates, with one study achieving up to 97% accuracy using a CNN model enhanced with MFCCs and other features. This highlights the effectiveness of these advanced methods in capturing the intricate patterns in speech data essential for accurate emotion classification.

Feature selection and engineering have been pivotal in enhancing SER accuracy. Studies employing sophisticated algorithms like the Grey Wolf Optimizer and advanced feature extraction methods such as the Fractional Fourier Transform have reported substantial improvements. Effective feature selection helps in removing redundant and irrelevant data, thereby improving the classifier's performance. This highlights the critical role of well-designed feature extraction and selection processes in building robust emotion recognition systems.

The versatility of CNNs in SER is evident across multiple studies. CNN-based models consistently demonstrate high accuracy and robustness, making them a

preferred choice for many researchers. For instance, several studies reported impressive results with CNN models, which were able to generalize well across various datasets. This highlights CNNs' capability to effectively learn and represent the complex features inherent in speech signals.

Dataset diversity is crucial for evaluating the performance of SER models. The use of multiple datasets, such as RAVDESS, SAVEE, Emo-DB, and IEMOCAP, ensures that the models are generalizable and robust. Studies achieving high accuracy across different datasets indicate that their models are adaptable and not overly tailored to a specific dataset. This is essential for developing emotion recognition systems that perform reliably in real-world scenarios.

Ensemble and hybrid models have also shown significant promise in improving SER performance. By combining different models and features, the strengths of each approach can be utilized. For example, an ensemble CNN model with multi-channel EEG and physiological signals enhanced both accuracy and stability in emotion recognition. This demonstrates the potential of integrating multiple data sources and methodologies to achieve superior results.

Addressing class imbalance and the specific challenges posed by certain emotions can lead to significant performance improvements. One study noted a marked increase in accuracy by excluding the "disgust" emotion from their classification task, highlighting the importance of considering class-specific issues in model development. Such strategies can help in mitigating biases and improving overall model performance.

Emerging trends and techniques, such as auto-encoders and attention mechanisms, are being increasingly integrated into SER systems. These methods capture more relevant features and focus on critical parts of the speech signal, leading to better performance. The use of such advanced techniques is indicative of the ongoing evolution in the field, aiming to push the boundaries of what SER systems can achieve.

Comparative studies provide valuable insights by evaluating different classifiers and feature selection techniques. These studies highlight the best-performing combinations and offer guidance for future research directions. By benchmarking various methods, researchers can identify the most effective strategies and continue to refine their approaches.

In summary, the recent advancements in SER methodologies stress the importance of feature selection, the effectiveness of CNN and hybrid models, and the value of using diverse datasets. Future research should focus on refining these techniques, addressing specific challenges like class imbalance, and exploring new, innovative approaches to further enhance SER performance.

2.16 ASR for Detecting Hate Speech

The Researchers have explored various approaches to detect hate speech from speech signals, often adapting techniques from ASR and speech processing domains. Traditional models, such as Hidden Markov Models (HMMs) combined with Gaussian Mixture Models (GMMs), have been used for acoustic modeling and speech recognition. These models which are proposed in their research by Hinton et al., [19] use statistical representations of speech features to recognize and classify speech segments as hateful or non-hateful based on pre-defined acoustic and phonetic patterns. However, their effectiveness in detecting hate speech from speech signals may be limited due to the complexity and variability of natural speech, especially in emotional and context-dependent contexts such as hate speech.

The integration of deep learning techniques has significantly improved the effectiveness of ASR systems for hate speech detection from speech signals. Sahu et al., [224] propose that deep neural networks, including CNNs and RNNs, have been extensively used for feature extraction and sequence modeling in this context. CNNs excel at capturing local patterns in the speech signal, while RNNs, including LSTM networks, can model the temporal dependencies inherent in speech data.

Advanced models based on the Transformer architecture have further enhanced ASR systems' capability for hate speech detection from speech signals. Vaswani et al., [223] proposed transformers to utilize self-attention mechanisms to capture long-range dependencies in speech signals, providing a more comprehensive understanding of the speech context.

Datasets used in these studies typically consist of speech samples labeled with hate speech annotations. These datasets include recordings from various sources, such as social media platforms, public speeches, and online forums, covering a wide range of accents and dialects.

Empirical results have demonstrated that deep learning models, particularly those based on CNNs and Transformers, significantly outperform traditional HMM-GMM models in hate speech detection from speech signals. These advanced models have achieved higher precision, recall, and F1 scores, indicating their superior performance in accurately detecting hate speech.

2.17 Conclusion

In conclusion, this literature survey has explored various advanced methodologies and algorithms for addressing the challenges in AASR. The survey covered feature extraction techniques, deep learning models, adversarial training, multi-task learning, and data augmentation methods, highlighting their respective advantages and limitations. Techniques like MFCC and deep neural networks have proven effective in capturing the nuances of speech across different accents, despite challenges such as computational complexity and the need for large datasets.

Innovative strategies such as adversarial training and multi-task learning have shown promise in improving model robustness and generalization to diverse accents. Self-supervised learning and autoencoders have further enhanced the capabilities of AASR systems by employing large amounts of unlabeled data, thus addressing the scarcity of labeled datasets for specific accents. Clustering methods

for emotion classification have also been examined, demonstrating their potential in handling accent variations and providing interpretable results.

The exploration of ensemble approaches and accent-unaware ASR techniques has emphasized the importance of robust, versatile models that perform well regardless of accent variations. These methods, though complex, offer significant improvements in recognition accuracy and robustness, making them crucial for the development of effective AASR systems.

The survey concludes that while substantial progress has been made, ongoing research is essential to refine these methodologies and overcome existing limitations. Future work should focus on enhancing the computational efficiency of these models, developing more sophisticated feature extraction techniques, and expanding the availability of diverse, high-quality datasets. By building on the foundations laid by these studies, the proposed research aims to contribute to the advancement of AASR, particularly for underrepresented languages and accents, thereby improving the accessibility and effectiveness of speech recognition technologies globally.

3. Research Methodology

3.1 Introduction

This chapter discusses the methodologies adopted in this research. Malayalam is a low resource language when the availability of data for conducting research is considered. This research lays the foundation for constructing AASR in the context of Malayalam. The entire research procedure right from the data collection phase to constructing different accented models adopting different methodologies that address the problem is discussed here.

3.2 Methodology

This chapter serves as a comprehensive guide to the methodologies employed in this research, which focuses on developing Accented Automatic Speech Recognition (AASR) for Malayalam which is characterized by limited acoustic and linguistic resources. The central objective is to establish a robust foundation for AASR within the Malayalam linguistic context. The research is a complex process, starting from the initial phase of data collection and extending to the construction of various accented models. The unique challenge of dealing with a low-resource language like Malayalam which is rich in phonetics and dialects necessitates a tailored approach. This chapter examines the intricacies of each step, elucidating the methodologies adopted to effectively address the research problem.

3.3 Multifaceted Exploration of AASR for Malayalam

The vast domain of speech recognition has witnessed tremendous advancements in recent decades. The regional languages, with their intricate phonetic and accentual variations, present unique challenges, and opportunities in the area. Malayalam with its rich tapestry of regional accents, intonations, and phonetic complexities, Malayalam offers scope for cutting-edge research in speech recognition.

Figure 3 presents the research design for processing and analyzing accented speech data using deep learning and machine learning techniques. The design starts with the collection of accented speech data, which includes isolated speech, continuous speech, and hate speech. The speech data is sourced from various accents considered in the research, specifically from regions such as Kasaragod, Kannur, Kozhikode, Malappuram, Wayanad, Thrissur, Kottayam, and Trivandrum.

The collected speech data undergoes several preprocessing steps to enhance its quality and suitability for further analysis. These steps include augmentation, noise removal, stretching and sampling. Augmentation involves artificially increasing the size and diversity of the speech dataset to improve the robustness of the models. Noise removal eliminates unwanted noise from the speech signals to ensure clarity and accuracy. Stretching and sampling techniques modify the speed and duration of the speech signals without affecting their pitch.

Once preprocessing is complete, the data moves to the feature engineering phase, where various crucial features for effective speech recognition and classification are extracted. These features include Mel Frequency Cepstral Coefficients (MFCC), Short Term Fourier Transformation (STFT), MelSpectrogram, Zero Crossing Rate (ZCR), Root Mean Square Value (RMS), Spectral Contrast, Tonnetz, Polyfeatures, Tempogram, Harmonic-to-Noise Ratio (HNR), Formants, Pitch Variability, and MFCC deltas and delta-deltas. These features are extracted using tools such as Parselmouth.

After feature extraction, the data is compiled into feature set CSV files, with dimensions reduced for efficiency. These feature sets are then ready for input into various machine learning and deep learning models. The models employed in this research design include Hybrid Neural Networks, LSTM-RNN (Long Short-Term Memory Recurrent Neural Networks), standard Machine Learning Techniques, Ensemble Methods (ML), Bi-LSTM (Bidirectional LSTM), Clustering Techniques,

DCNN (Deep Convolutional Neural Networks), Autoencoders, Attention Mechanisms, and Ensembled Clustering.

These models analyze the speech data to achieve tasks such as speech recognition, accent classification, and potentially detecting hate speech. This structured research design, starting from data collection and preprocessing to feature extraction and model training, outlines a robust pipeline for handling accented speech data. The use of various machine learning and deep learning techniques ensures a comprehensive analysis, capable of managing the complexities of different accents and speech types.

This research embarked on a comprehensive journey, dissecting various facets of accented Malayalam speech recognition. From crowdsourced data collection of isolated words to the exploration of advanced neural network architectures. The study delved into emotion classification, predicting hate speech patterns, and fine-tuning network models, ensuring optimized performance tailored specifically for Malayalam. The integration of traditional machine learning algorithms provided a holistic approach, balancing the modern deep learning paradigms with established methodologies.

3.3.1 Isolated Words using LSTM-RNN with Crowdsourcing

In one of the foundational phases of this research, a significant emphasis was placed on recognizing isolated Malayalam words. Crowdsourcing methods were employed to amass a diverse dataset, ensuring a broad representation of accents and phonetic complexities. Once collected, LSTM-RNN methods were employed to model the sequential nature of speech. These networks, known for their prowess in capturing temporal dependencies, were thoroughly tuned to recognize isolated words from the crowdsourced data.

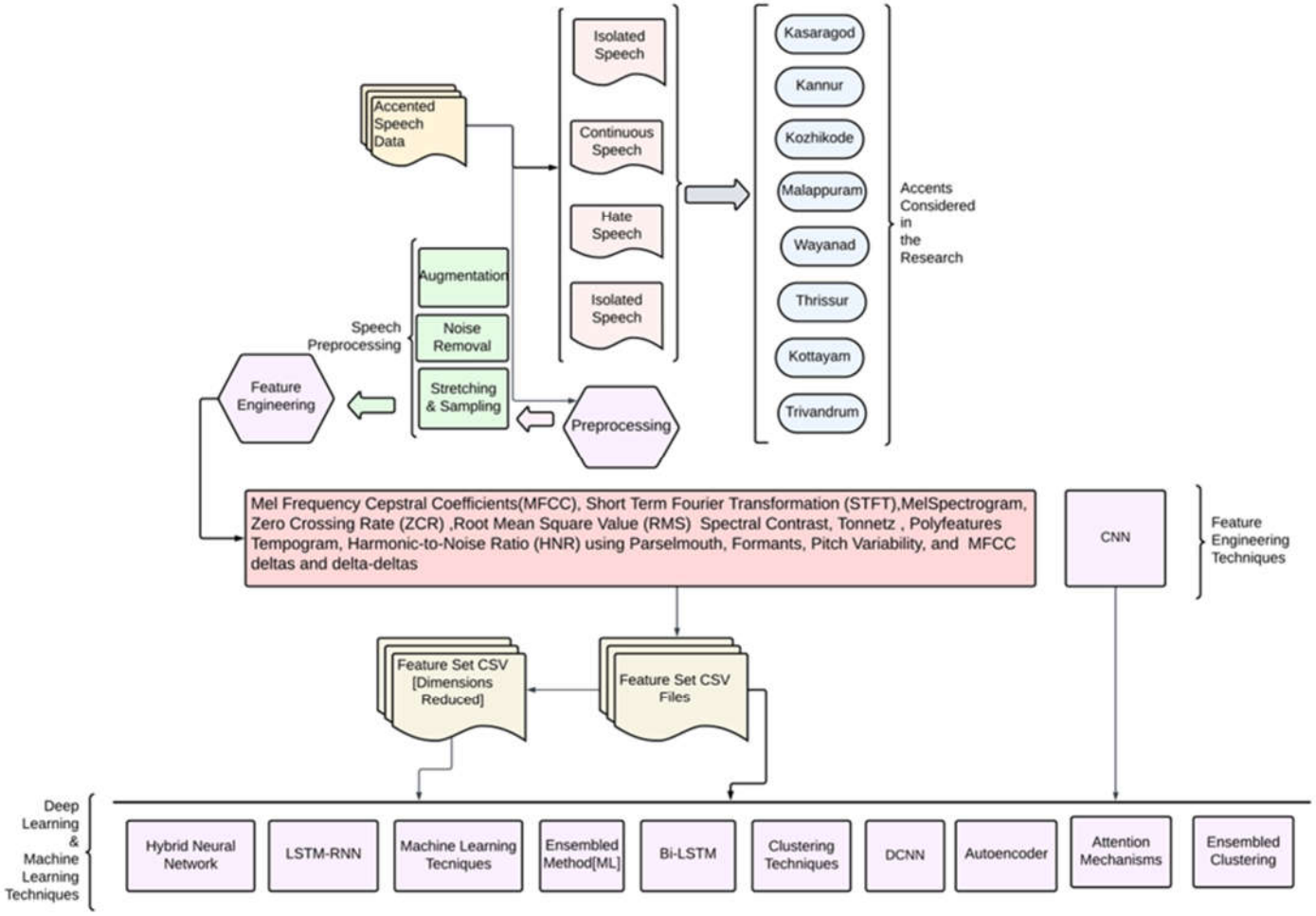


Figure 3 The Research Design

3.3.2 Feature Combinations with Neural Networks

Building on the foundational LSTM-RNN models, the research then investigated into exploring diverse feature combinations. Recognizing that the intricacies of accented Malayalam speech could be captured more effectively by combining various features, extensive experiments were conducted. By synergizing different acoustic and phonetic features with neural network architectures, the research aimed to improve the performance of the AASR models.

3.3.3 Exploration of Various Machine Learning Algorithms

Broadening the horizon, the study delved into various machine-learning algorithms. These algorithms, each with their unique strengths, were applied to the Malayalam accented speech dataset. The objective was to determine the most suitable algorithms for this specific linguistic challenge, considering factors like computational efficiency, scalability, and recognition accuracy.

3.3.4 Comparative Study on Neural Network Architectures

An exhaustive comparative study was undertaken, focusing on multiple neural network architectures. This studies in this research have constructed several AASR models using LSTM, LSTM-RNN,1D CNN, 2D Parallel CNN, 2D Parallel CNN with attention mechanisms, more complex 4D CNN architectures, 4D CNN architectures with attention mechanisms, autoencoders, hybrid autoencoders with machine learning techniques, hybrid models combining CNN and LSTM, and BiLSTM architectures. Each architecture was carefully fine-tuned to optimize its performance in the context of accented Malayalam speech recognition.

3.3.5 Emotion Classification using Clustering Techniques

Transitioning from pure speech recognition, the research also ventured into the area of emotion classification from accented speech signals. By employing various clustering techniques, the study aimed to identify and categorize emotional data.

3.3.6 Predicting Hate Speech from Accented Datasets

In a socially relevant exploration, this research also addressed the challenge of identifying hate speech within the accented Malayalam dataset. Recognizing the potential implications of hate speech in today's digital age, models were trained and fine-tuned to examine the dataset, identifying and classifying segments that exhibited indications of hate speech.

3.3.7 Fine-Tuning Network Architectures

Across all segments of the study, a consistent theme was the fine-tuning of network architectures. Regardless of the model constructed, efforts were made to optimize each architecture in the specific context of accented Malayalam.

3.4 Conclusion

The selection of these methodologies is driven by the need for a holistic exploration of feature extraction techniques and model architectures. This research contributes not only to the development of AASR for Malayalam but also lays the groundwork for future studies in accented speech recognition. The methodologies outlined herein serve as a roadmap, offering insights into the intricacies of constructing robust models for languages with small datasets and languages that are scarce in availability of corpus, thereby progressing the broader field of research in the domain of speech recognition. The workflow of the proposed study is illustrated in Figure 4.

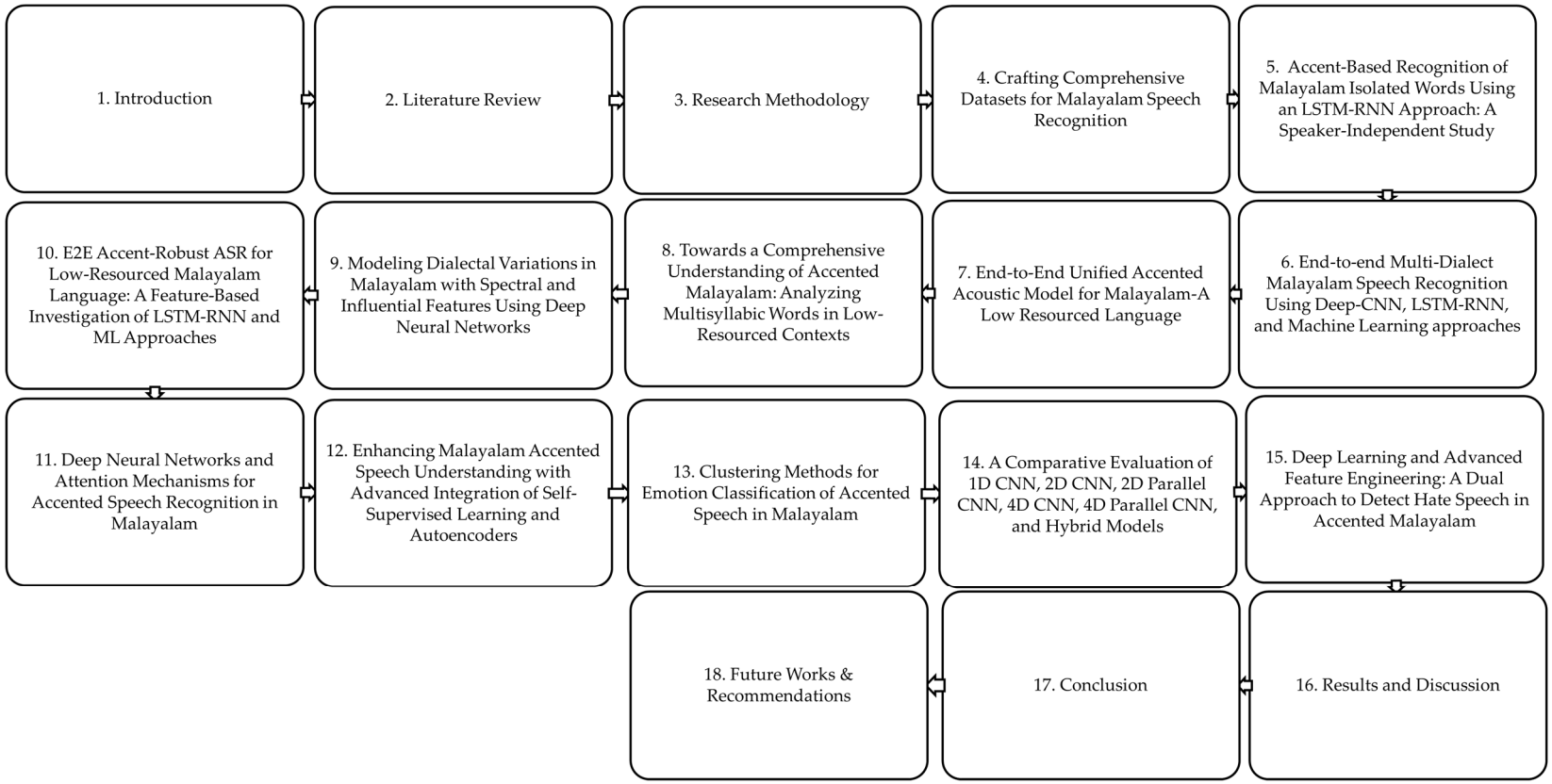


Figure 4 Workflow of the Proposed Study

4. Crafting Comprehensive Datasets for Malayalam AASR

4.1 Introduction

In the area of language technology research, the development of accurate and versatile datasets stands as the foundation. In the context of constructing AASR for the Malayalam language, a versatile dataset is crucial to understanding the intricacies of different accents across various regions and demographics. This dataset, carefully constructed through a rigorous process, lays the foundation for advanced techniques in the domain, opening doors to new insights and applications in the field of speech recognition.

This attempt indicates a firm commitment to linguistic diversity, technological innovation, and the advancement of language-related research. This dataset constitutes the foundational cornerstone upon which this research endeavors are constructed, facilitating an exhaustive exploration of the intricate subtleties inherent in diverse accents across varying regions and demographic strata within Kerala.

4.2 Challenges in Constructing the Dataset

The effort to construct a robust and authentic dataset for accented speech recognition posed several significant challenges, necessitating innovative solutions. Foremost among these challenges was the absence of a freely accessible benchmark dataset that aligns with the precise aims of this research. This deficiency emphasized the urgency for a tailored dataset capable of accurately capturing the array of diverse accents and linguistic variations embedded within the Malayalam language. Such issues have been similarly noted in prior studies focusing on lesser-studied languages and dialects, where researchers have had to create custom datasets to meet their specific research needs are discussed in the studies conducted by Wester et al., [231] and Räsänen et al., [232].

The complex nature of these challenges was further exacerbated by the necessity to integrate diverse demographic factors, including age groups, genders, and geographic locales, into the dataset construction process. The studies conducted by Tjalve & Skoog [233] and Chen et al., [234] discusses that the inclusion of such varied demographic information is crucial to developing robust ASR systems that perform well across different speaker profiles, as highlighted in previous works on multilingual and multi-accented speech recognition. These complexities introduced significant complications, including logistical challenges in data collection and the need for sophisticated balancing techniques to ensure that the dataset remains representative and unbiased.

Addressing these challenges prompted a pioneering approach aimed at generating different significant datasets. To collect a comprehensive range of speech samples, innovative data collection strategies were employed, such as crowd-sourcing and community engagement, which have been effective in similar linguistic studies is discusses in the studies conducted by He et al., [235] and Ghosh et al., [236]. Miao et al., [237] in their study discussed about the advanced preprocessing techniques that were utilized to enhance the quality and consistency of the speech data, which is critical for training effective ASR models.

The need to capture linguistic variations specific to the Malayalam language involved detailed phonetic analysis and careful design of the recording scripts to ensure coverage of all phonemes and contextual variations. Such complex design has been emphasized in other speech recognition research focusing on low-resource languages in the study conducted by Besacier et al., [238] and Watanabe et al., [239]. These efforts were crucial to creating a dataset that not only meets the immediate goals of this research but also contributes to the broader field of speech technology by providing a valuable resource for future studies on Malayalam and other Dravidian languages.

The construction of a robust and authentic dataset for accented speech recognition in Malayalam required overcoming significant challenges related to the lack of existing resources, demographic diversity, and linguistic complexity. Through innovative data collection and preprocessing methods, a comprehensive and high-quality dataset was developed, providing a solid foundation for advancing ASR technologies in underrepresented languages.

4.3 Challenges in Constructing Accented Dataset

Preparing an accented speech dataset for the Malayalam language involves numerous challenges, influenced by the region's unique linguistic and demographic characteristics. Linguistic diversity is a primary obstacle, as different districts and communities exhibit distinct accents and dialects, complicating the creation of a uniform dataset. Ensuring demographic variability requires including speakers from various backgrounds, encompassing differences in age, gender, education, and socio-economic status, which adds complexity to the recruitment process. The logistics of setting up the infrastructure for data collection, including high-quality recordings from diverse environments, particularly in remote areas, is resource-intensive and challenging. Ethical and privacy concerns also arise, necessitating careful management to ensure informed consent and protect participant identity. The resource constraints, both financial and human, further impede progress, as developing a large, diverse dataset requires significant funding and skilled personnel.

Accurate annotation and validation of speech data is a labor-intensive process requiring expertise in phonetics and linguistics, are essential for training reliable ASR models but add to the complexity. Additionally, the technological barriers, such as the lack of robust tools for processing and annotating speech data in Malayalam, further slowdown dataset creation.

To address these challenges, researchers must adopt innovative approaches, including utilizing mobile technology for data collection, employing crowdsourcing

techniques, and utilizing self-supervised learning methods to reduce dependency on annotated data. Collaborative efforts among academic institutions, government agencies, and private enterprises can provide the necessary resources and expertise to build a comprehensive accented speech dataset for Malayalam.

4.4 Strategies for Data Collection

The strategies adopted for data collection in this research is threefold:

1. Crowdsourcing
2. Recording manually
3. Selecting Accented Audio from Online Platforms (Mainly YouTube as a Source of Data)

4.4.1 Crowdsourcing

This approach harnesses the power of a large and diverse pool of contributors, which helps address many of the challenges associated with data collection in linguistically diverse settings. Crowdsourcing enables the collection of speech data from a broad and varied demographic, ensuring the inclusion of multiple accents, dialects, and socio-economic backgrounds. This diversity is crucial for building robust ASR systems capable of handling real-world variability in speech patterns.

Crowdsourcing can significantly reduce the logistical and financial burdens associated with traditional data collection methods. Traditional methods often require setting up physical recording environments, which can be resource-intensive and time-consuming, especially in remote or underrepresented areas. Crowdsourcing platforms, on the other hand, allow participants to contribute data remotely using their devices, thus minimizing the need for extensive infrastructure. The decentralized nature of crowdsourcing makes it a cost-effective solution for large-scale data collection.

Moreover, crowdsourcing facilitates rapid data collection, enabling researchers to gather vast amounts of data in a relatively short period. The scalability of

crowdsourcing is particularly advantageous when constructing large datasets necessary for training deep learning models. The ability to quickly amass large quantities of data accelerates the development cycle of ASR systems, allowing for more iterative and responsive research and development processes.

Crowdsourcing platforms often come with built-in mechanisms for quality control, such as peer review and automated validation checks, which help maintain the accuracy and reliability of the collected data. For instance, employing these quality assurance techniques ensures that the speech samples meet the required standards, even when contributed by non-experts.

The potential drawbacks of crowdsourcing can be described as varying recording conditions and the need for effective management of participant consent and data privacy. These challenges must be carefully managed to fully utilize the benefits of this approach.

The crowdsourcing approach is widely adopted in the ASR research due to its ability to capture diverse speech variations, reduce logistical and financial constraints, accelerate data collection, and incorporate effective quality control measures. These attributes make crowdsourcing a practical and efficient method for developing comprehensive speech datasets essential for robust ASR systems.

4.4.2 Manual Recording

In addition to crowdsourcing and selecting accented audio from online platforms, manual recording plays a crucial role in capturing a diverse range of accented Malayalam speech patterns. This method offers controlled conditions for data collection, allowing researchers to target specific accents and linguistic features more precisely.

4.4.2.1 Methodology:

1. **Participant Selection**: To ensure a representative sample, participants were selected based on demographic factors such as geographical origin, age, and language proficiency. This approach aimed to encompass the various accents and speech variations present in Malayalam-speaking populations.

2. **Recording Setup**: Manual recording sessions were conducted in a controlled acoustic environment using high-quality recording equipment, including microphones and audio software. This setup helped minimize background noise and ensure the clarity of recorded speech.

3. **Prompt Design**: Participants were provided with a series of prompts designed to elicit natural speech production. These prompts covered a range of topics, from everyday conversations to reading passages aloud, allowing researchers to capture a diverse range of speech patterns and linguistic features.

4. **Data Collection Process**: During recording sessions, participants were instructed to speak spontaneously, using their natural accents and speech styles. Researchers monitored the sessions to maintain consistency and address any issues that arose during the recording process.

4.4.2.2 Advantages:

- **Controlled Environment**: Manual recording provides researchers with control over environmental variables, ensuring standardized conditions for data collection and analysis.

- **Targeted Sampling**: By selecting participants based on specific criteria, such as regional background and language proficiency, manual recording allows researchers to target accents and speech variations of interest.

- **Detailed Analysis**: High-quality recordings obtained through manual sessions facilitate detailed acoustic analysis, enabling researchers to identify subtle differences in pronunciation and intonation.

Manual recording serves as a valuable method for collecting accented Malayalam speech data, offering researchers control over recording conditions and facilitating targeted sampling. By employing rigorous methodology and addressing logistical challenges, researchers can obtain high-quality data essential for analyzing accent variations and linguistic features in Malayalam speech.

4.4.3 Accented Audio from Online Platforms

Incorporating online platforms, particularly YouTube, into the data collection methodology presents a modern approach to gathering a diverse range of accented Malayalam speech samples. The strategy involves scouring YouTube for videos featuring natural conversations, interviews, storytelling, and other authentic speech contexts where Malayalam is spoken. By using relevant search terms and filters, the aim is to capture a wide spectrum of accents and speech variations representative of different regions and demographics. Selected videos undergo scrutiny to ensure the authenticity and relevance of the spoken content. Audio segments are extracted from these videos, paying close attention to audio clarity and the presence of identifiable accent features. Utilizing YouTube offers access to a vast array of speech samples recorded in real-world settings, contributing to the ecological validity of the dataset. However, challenges such as verifying speaker authenticity and addressing potential biases in online content must be carefully considered to maintain data integrity. Despite these challenges, integrating YouTube as a primary source of accented audio data enriches the research by providing a diverse and extensive dataset for analyzing Malayalam speech variation.

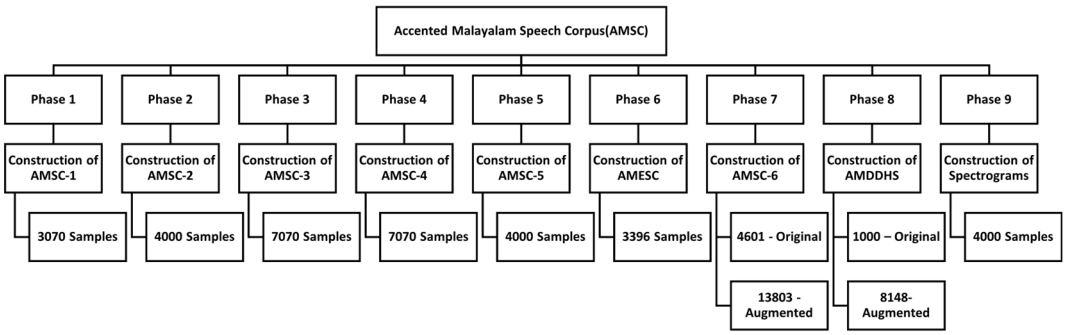


Figure 5 The Phases of Data Collection

Figure 5 illustrates the different phases of data collection that have been performed at various stages of this research. The overall process illustrates a systematic approach to building a comprehensive and specialized speech corpus for accented Malayalam. Each phase contributes to a specific aspect of the dataset, ensuring its richness and diversity. The study encompasses general speech data, emotional speech, and hate speech detection, making it a valuable resource for a wide range of accented speech processing applications in Malayalam. A total of 38027 speech samples were collected in nine phases to conduct the entire study.

From the initial phase of recording utterances in distinctive districts of Kerala to the subsequent stages involving the extraction of accented speech data from YouTube videos and the annotation of emotions and hate speech/non-hate speech, this methodology exemplified a holistic and systematic effort to capture the intricacies of accented speech in a comprehensive manner. The dataset not only reflected authentic accents but also resonated with the subtleness and diversity inherent in real-world linguistic and emotional expressions. This methodological framework draws attention to the commitment to producing a dataset that would serve as a valuable resource for advancing research in accented speech recognition and related studies.

Each utterance was carefully annotated with its original accent and its corresponding standard accent. This dual annotation approach empowers this research by enabling a direct comparison between the speaker's natural accent and the standardized reference. Soliciting multiple utterances from each of the 100 speakers provided the

flexibility required to explore diverse pronunciation patterns, intonations, and phonetic characteristics inherent to accented speech. The dataset's intentional diversity, encompassing both genders and spanning a broad age spectrum, accounted for potential variations influenced by these factors. The strategic selection of districts known for their distinctive accents further contributed to the dataset's richness by capturing region-specific linguistic traits.

The process of crafting a unique dataset for this research reflects a spirit of original innovation and commitment to overcoming challenges. The depth and authenticity of the dataset serve as the foundation of the precise methodology, strategic framework, comprehensive data collection techniques, and sophisticated dual annotation approach utilized. These elements collectively establish the groundwork for a thorough investigation into accented speech recognition within the specific context of the Malayalam language.

4.5 Generating a Comprehensive Dataset

This dataset serves as the foundation for research works, allowing for a thorough exploration of the complications present in diverse accents across various regions and demographic groups within Kerala. The datasets created are labeled as the Accented Malayalam Speech Corpus (AMSC) Dataset, each identified with specific version numbers.

4.5.1 AMSC-Dataset: Fostering Accented Malayalam Speech Corpus

The dataset throughout this research is appropriately titled the "AMSC-Dataset," with "AMSC" representing "Accented Malayalam Speech Corpus." This nomenclature encapsulates the fundamental essence and objective of the dataset to serve as a comprehensive and versatile resource for the exploration and enhancement of accented speech in the Malayalam language. The dataset is carefully

organized into versions, each signifying a unique iteration and enhancement of the corpus, in alignment with the evolving requirements and goals of the research.

4.5.2 The Significance of the Name

The acronym "AMSC," standing for "Accented Malayalam Speech Corpus," aptly encapsulates the dataset's central focus and the exploration and representation of accented speech in the Malayalam language. Malayalam, renowned for its rich cultural and linguistic diversity, forms the foundation of this comprehensive corpus. It encompasses authentic accents and variations arising from diverse regions and communities, contributing to a deeper comprehension of linguistic diversity within the language.

4.5.3 Versions for Enhanced Relevance

Incorporating various versions within the AMSC-Dataset reflects the dedication to ongoing enhancement and adjustment to the changing research landscape. Each version represents a milestone in refining and expanding the corpus to better suit the research objectives, making it an ever evolving and dynamic resource.

4.6 Phases of Data Collection

The AMSC-Dataset, with its carefully curated data from natural environments, WhatsApp, YouTube, and various sources, is poised to be a valuable resource for research in accented speech recognition, emotion analysis, and hate speech detection within Malayalam. Its complex nature, rich accents, and applicability to real-world scenarios make it an invaluable resource for researchers pursuing to examine the intricacies of accented speech within the Malayalam language. The naming and structuring of the AMSC-Dataset embody a commitment to fostering the study of accented Malayalam speech and signify its evolving nature to cater to the research community's requirements. The construction of the dataset was incrementally carried out to meet the requirements of the studies conducted in different phases of this research.

4.6.1 Phase 1: Construction of AMSC-1

The construction of AMSC-1 involved gathering 3070 speech samples from a diverse group of 29 speakers, of varying age starting from 6 and involving both genders 10 males and 19 females. Unlike previous studies conducted in studio environments, this experiment built a speech dataset of 1.705 hours in a normal environment with noise disturbances using crowdsourcing. Diverse platforms were facilitated for data collection from various global speakers.

The recordings make up the speech corpus, which has been assembled from the public under normal and natural recording conditions. The statistics of the data collection for AMSC-1 are shown in Figure 6. The AMSC-1 dataset comprises 3,070 samples collected across different age groups. The distribution shows a notable variation in the number of samples per age group. The largest group is 21 to 40 years, contributing 1,290 samples, which is significantly higher than other groups. The 41 to 60 age group follows with 590 samples, while the below 10 and 11 to 20 age groups have nearly equal representation with 459 and 460 samples, respectively. The 61 to 80 age group has the fewest samples, with only 271 collected. This distribution highlights a predominant focus on the 21 to 40 age group in the dataset.

A diverse group of twenty-nine participants was thoroughly chosen, considering factors such as age, gender, and geographical location to build a robust dataset. The resulting dataset spans 1.705 hours and includes multiple utterances of words, all of which have been saved as .wav format at a sampling rate of 16 KHz.

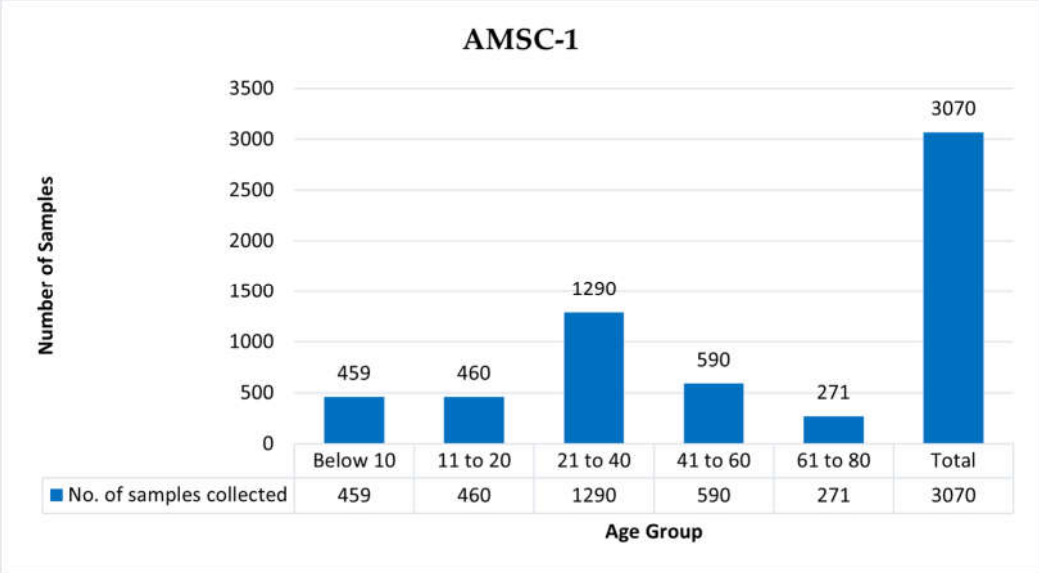


Figure 6 Statistics of AMSC-1

This innovative method allowed the collection of speech data from speakers from various locations, with some contributions collected over the Internet and others directly recorded on-site. It was an approach that ensured a mix of accents and speaking styles, closely representing the real-world scenarios the model would encounter. Table 3 illustrates the sample classes used for the experiment.

Table 3 The Classes of Isolated Words in the Dataset

Uttered word	IPA	Uttered word	IPA
പൂജ്യം	puːjjam	പുസ്തകം	pusʈakam
ഒന്ന്	onʈə	വരയ്ക്കുക	ʋarajkkuka
രണ്ട്	ɾanʈə	അറിവ്	arivə
മൂന്ന്	muːnʈə	പഠിക്കുക	paʈʰikkuka
നാല്	naːl	ലൈബ്രറി	laiɓrari
അഞ്ച്	ajtʃə	വായിക്കുക	ʋaːjikkuka
ആറ്	aːrə	സ്കൂൾ	skuːɭ
ഏഴ്	eːɻə	വിദ്യാർത്ഥി	ʋidjaːrʈʈʰi
എട്ട്	eʈʈə	അധ്യാപകൻ	adʱjaːpakan
ഒമ്പത്	onpaʈə	എഴുതുക	eɻuʈuka

4.6.2 Phase 2: Construction of AMSC -2

The data that was collected exhibits pronounced dialectal differences from the standard Malayalam dialect. This dataset is comprehensive and encompasses samples from individuals of various age groups and both genders (male and female). It comprises contributions from 30 unique native speakers spanning different age categories. For every word in the dataset, multiple utterances were recorded by diverse speakers.

All recordings underwent a standardization process, (as mentioned in the earlier approach and followed in all the phases of dataset construction throughout the speech signal processing adopted in this study) into a common .wav format and a common sampling frequency of 16 KHz has been adopted for majority of the experiments. This initial step in the preprocessing task ensures consistency across all samples. By capturing the rich linguistic subtleties of the Malayalam language in diverse regional contexts, this dataset lays a robust foundation for the in-depth investigation of accented speech recognition. It encompasses a wide spectrum of accent variations, establishing a firm basis for subsequent analysis and experimentation.

The AMSC-2 dataset consists of 4,000 audio samples distributed across six districts. Each district contributes a specific number of recordings to the dataset. Kozhikode provides the highest number of samples, with 1,090 recordings. Kasaragod, Kannur, and Malappuram each contribute an equal number of 760 samples. Wayanad has the fewest samples, with 630 recordings. This distribution indicates a varied representation across the districts, with Kozhikode being the most significant contributor. The total number of recordings from all districts combined sums up to 4,000.

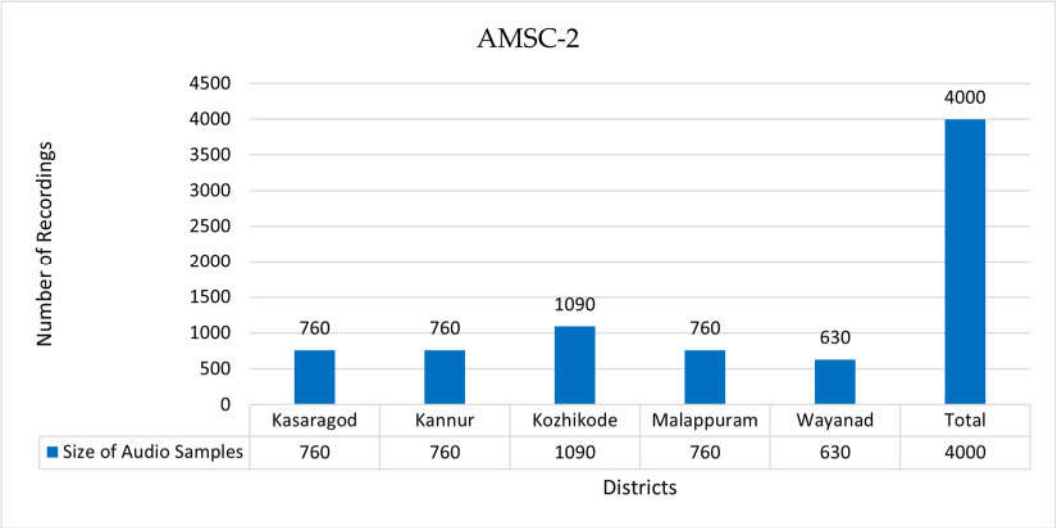


Figure 7 Statistics of AMSC-2

The AMSC-2 dataset is also categorized based on age-wise distribution. The age group 20 to 45 contributes the largest number of recordings, with 1,500 samples, indicating a significant focus on this demographic. The age groups 5 to 12 and 13 to 19 each contribute 660 samples, showing equal representation. The age group 46 to 65 provides 690 samples, while the age group 66 to 85 contributes the fewest, with 490 recordings. This distribution highlights a concentration of audio samples from the middle-aged group (20 to 45), with balanced but lower representation from the younger and older age groups, resulting in a total of 4,000 samples across all age categories.

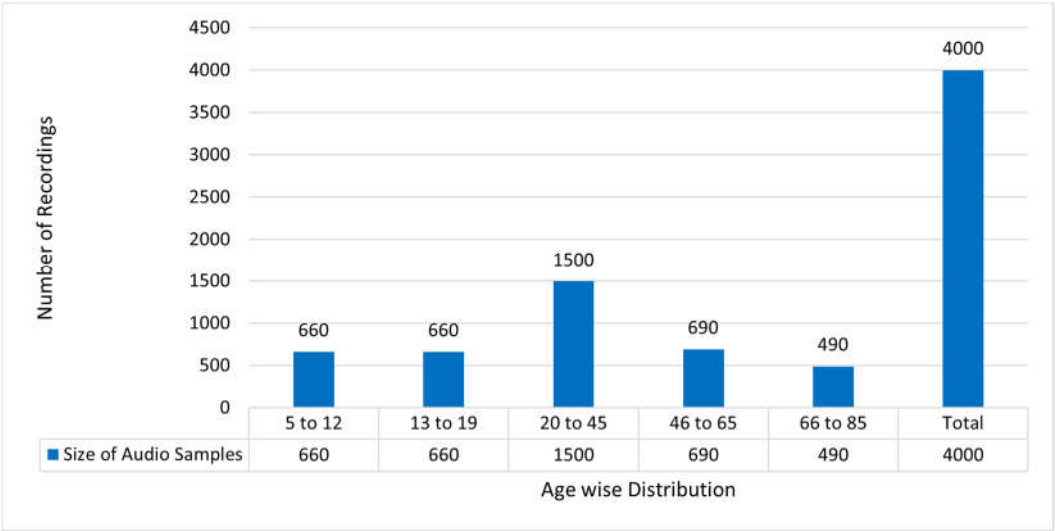
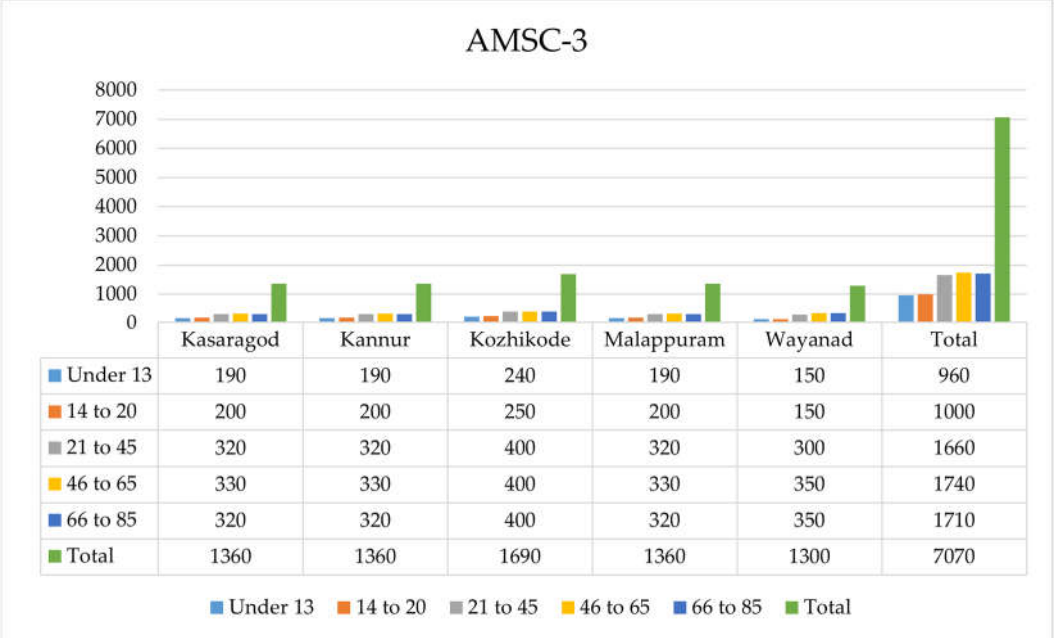


Figure 8 Age Wise Statistics

Figure 7 illustrates the distribution of the samples across different districts and Figure 8 contains the statistical data of the speech recordings collected based on the different age groups. The dataset contains speech recordings from participants in donating speech of all ages.

4.6.3 Phase 3: Construction of AMSC-3

In the development of AMSC-3 speech corpus, 7070 samples were curated through crowdsourcing methodologies, ensuring a diverse and comprehensive collection. This approach was instrumental in capturing authentic and natural speech reflective of real-world scenarios. To achieve a balanced representation, 30 speakers, comprising both males and females, actively contributed to the corpus.



	Kasaragod	Kannur	Kozhikode	Malappuram	Wayanad	Total
Under 13	190	190	240	190	150	960
14 to 20	200	200	250	200	150	1000
21 to 45	320	320	400	320	300	1660
46 to 65	330	330	400	330	350	1740
66 to 85	320	320	400	320	350	1710
Total	1360	1360	1690	1360	1300	7070

Figure 9 Age Wise Statistics of AMSC-3

This diversity added complexity and depth to the dataset, particularly in terms of gender-related speech characteristics. In a strategic move to enhance the richness of the corpus, speech data was gathered from five distinct districts within the Kerala region, each representing a unique accent within the Malayalam language. This geographical diversity played a crucial role in capturing the multifaceted nature of regional accents.

Figure 9 represents the number of speech data collected across five districts in Kerala, categorized by distinct age groups. In Kasaragod, a total of 190 speech data samples were collected for individuals under 13 years old, while 200 samples were collected for those aged 14 to 20. The dataset records 320 samples for the 21 to 45 age group, 330 for individuals aged 46 to 65, and another 320 for those aged 66 to 85, resulting in a total of 1360 speech data samples. Similarly, in Kannur, 190 samples were collected for individuals under 13, 200 for those aged 14 to 20, and 320 each for the 21 to 45, 46 to 65, and 66 to 85 age groups, totaling 1360 samples. Kozhikode reports 240 samples for individuals under 13, 250 for those aged 14 to 20, and 400 each for the 21 to 45, 46 to 65, and 66 to 85 age groups, resulting in a total of 1690 samples.

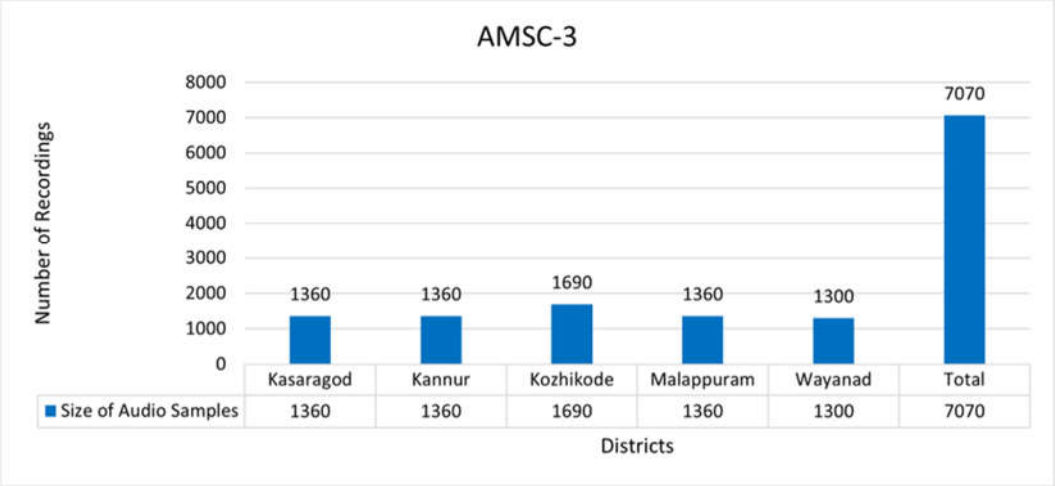


Figure 10 Statistics of AMSC-3

Across all districts, the total number of speech data samples collected for each age group sums up to 960 below 13, 1000 aged 14 to 20, 1660 for the 21 to 45 age group, 1740 aged 46 to 65, and 1710 aged 66 to 85, with an overall total of 7070 speech data samples collected. These numbers provide valuable insights into the distribution of speech data collection efforts across different age groups and geographical regions, offering a comprehensive view of the dataset's coverage and scope. As a result, the dataset, totaling 5.8 hours of speech, stands as a robust foundation for comprehensive experimentation and analysis in this study.

In Malappuram, 190 samples were collected for individuals below 13, 200 for those aged 14 to 20, and 320 each for the 21 to 45, 46 to 65, and 66 to 85 age groups, also totaling 1360 samples. Wayanad exhibits 150 samples for individuals under 13, 150 for those aged 14 to 20, and 300 for the 21 to 45 age group, with 350 samples each for the 46 to 65 and 66 to 85 age groups, resulting in a total of 1300 samples.

Figure 10 illustrates the number of audio recordings collected from five districts: Kasaragod, Kannur, Kozhikode, Malappuram, and Wayanad. Each of Kasaragod, Kannur, and Malappuram contributed 1360 recordings, while Kozhikode provided the highest number of samples at 1690, and Wayanad contributed the fewest with 1300 recordings. The total number of recordings across all districts amounts to 7070. The diverse dataset ensures a comprehensive analysis of the audio samples collected from these different regions.

4.6.4 Phase 4: Construction of AMSC-4

A speech corpus, comprising approximately 1.17 hours of recordings, has been curated for the purpose of this research within a natural environment. These recordings capture individual utterances of multisyllabic words, each lasting between two to five seconds, forming the foundational elements of the corpora.

The gathered speech samples originated from a diverse group of forty speakers, maintaining an equal balance between males and females. Covering a broad age spectrum from five to eighty, these participants are native speakers who speak authentic accents of Kasaragod, Kannur, Kozhikode, Wayanad, and Malappuram.

The accents within these regions bear the influence of languages from neighboring states, resulting in a distinctive blend of speech patterns. It is worth highlighting that most of the samples in the collection exhibit the Kozhikode accent, characterized by its close alignment with the standard Malayalam accent. This specific emphasis adds depth to the analysis and modeling of the speech data.

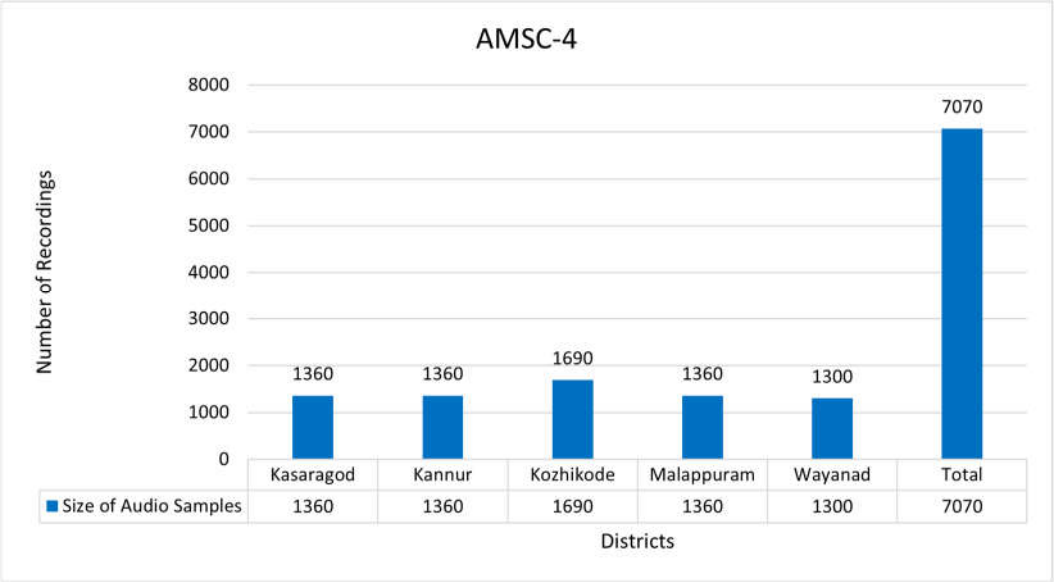


Figure 11 Statistics of AMSC-4

Figure 11 illustrates the distribution of audio recordings collected from five districts: Kasaragod, Kannur, Kozhikode, Malappuram, and Wayanad. Kasaragod, Kannur, and Malappuram each contributed 1360 recordings, while Kozhikode provided the highest number with 1690 recordings, and Wayanad the fewest with 1300 recordings. The total number of recordings across all districts sums to 7070. This distribution indicates a generally balanced sampling approach with some variations, reflecting possibly higher participation in Kozhikode and fewer samples from Wayanad due to logistical factors. The comprehensive dataset ensures a robust analysis of audio samples across these diverse regions, enhancing the study's representativeness and reliability.

4.6.5 Phase 5: Construction of AMSC-5

To construct a representative speech corpus, samples were collected from individuals across all age groups. Thirty native speakers, covering diverse age ranges and including both males and females, provided multiple utterances. This approach was crucial to accurately capture the diverse signals and unique pronunciation patterns present within the dataset.

The dataset is compiled from five unique districts: Kasaragod, Kannur, Kozhikode, Wayanad, and Malappuram. To construct an illustrative, authentic and representative speech corpus, samples were collected from individuals across all age groups. This approach was crucial to accurately capture the diverse signals and authentic pronunciation patterns present within the dataset.

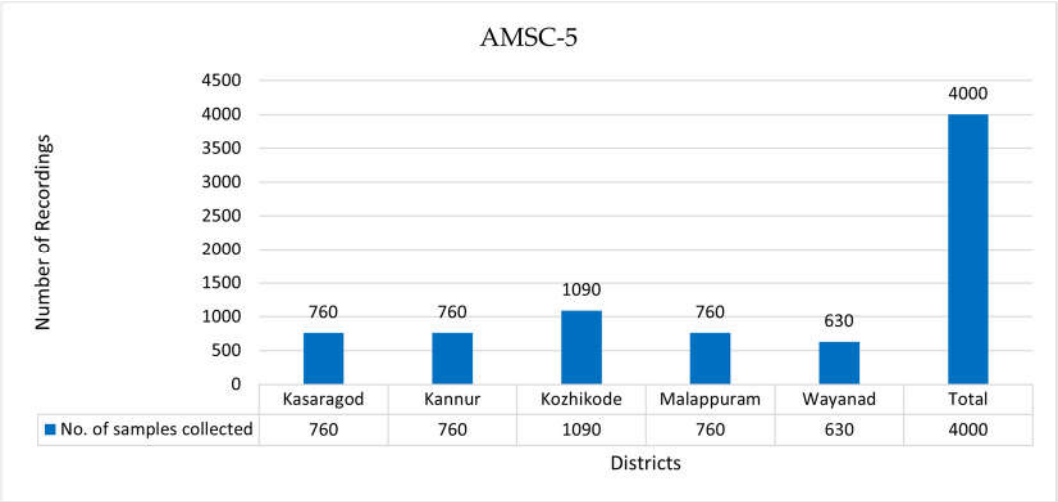


Figure 12 Statistics of AMSC-5

The standardization of the recordings has been carefully undertaken to ensure uniformity across the dataset, facilitating precise analysis in the subsequent stages of the study. The resultant speech corpus, comprising approximately 1.25 hours in duration, serves as the basis of the experiment.

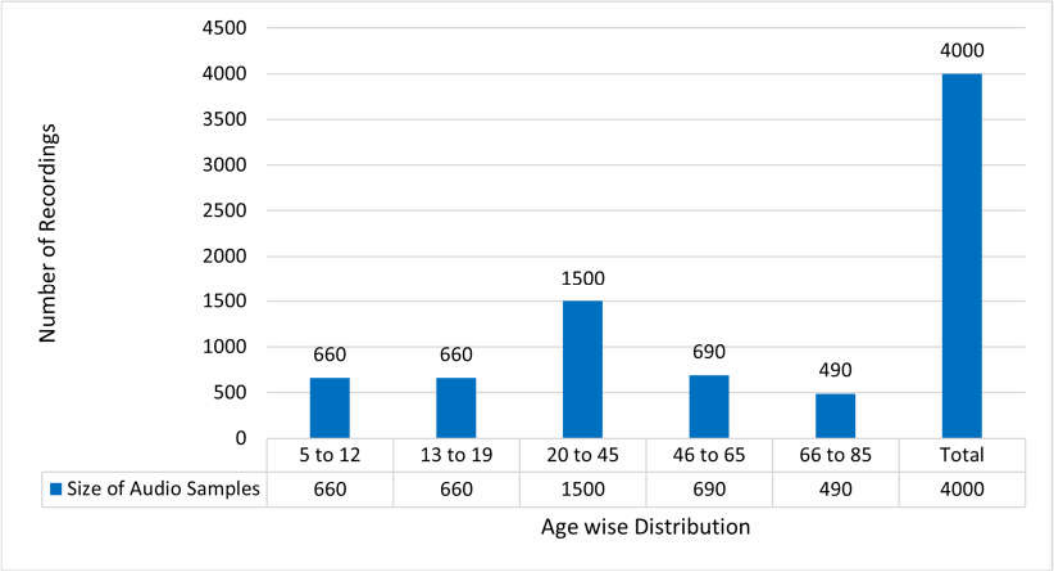


Figure 13 Age Wise Statistics of AMSC-5

Figure 12 illustrates the number of audio recordings collected from five districts: Kasaragod, Kannur, Kozhikode, Malappuram, and Wayanad. Kasaragod, Kannur, and Malappuram each contributed 760 recordings. The total number of recordings collected across all districts is 4000. Figure 13 represents the age wise statistics regarding AMSC-5.

4.6.6 Phase 6: Construction of Accented Malayalam Emotional Speech Corpus (AMESC)

AMESC dataset was constructed from the publicly available YouTube platform. YouTube's extensive array of videos provided a diverse collection of speech samples, capturing various accents and emotional expressions. To maintain a focus on the Malayalam language and its diverse accents, specific criteria were applied for video selection. Preference was given to videos that prominently displayed explicit emotional expressions, aligning with the seven emotions under consideration. The emotions considered in the study were surprise, happiness, sadness, anger, neutral, fear, and disgust. The resulting dataset encompasses a diverse array of speech samples representing these seven emotions sourced from various accented contexts. This corpus forms a crucial component of the dataset generation process.

The research process involves several key steps. To enrich the dataset's authenticity, district selection is crucial, focusing on regions known for diverse Malayalam accents. Data preprocessing is employed to eliminate potential noise and enhance audio quality, and rigorous quality control checks are performed to maintain dataset integrity. Table 4 illustrates the statistics of the emotional data collected in detail.

Table 4 The Distribution of AMESC Dataset

Emotions	**District wise Accents**	**Size of the Sample**
Angry	Kasaragod, Kozhikode, Thrissur	420
Disgust	Thiruvananthapuram, Kottayam	540
Fear	Kannur, Thrissur, Malappuram	538
Happy	Thrissur, Kannur, Thiruvananthapuram	512
Sad	Kasaragod, Kannur, Kottayam	520
Surprise	Kannur, Kozhikode, Kasaragod	534
Neutral	Thiruvananthapuram, Kottayam	332
Total		3396

Additionally, the dataset incorporates labels providing extra context, including the transcription in standard accent, relevant demographic information, and accent annotations. Table 5 contains the sentences and the classes of emotion that have been collected for the AMESC dataset.

Table 5 Sample Emotional Speech Categories in the AMESC Dataset

Sample Speech in the Dataset	**English Transcription**	**Emotion**
പക്ഷെ ഒരു കാര്യം ഞാൻ പറഞ്ഞേക്കാം	pakshe oru kaaryam njaan paranjekaam	Angry
ആരാ നിങ്ങൾക്കതിനു സമ്മതം നൽകിയത്	aara ningalkathinu sammatham nalkiyath	Angry
ആരെടാ നിന്നോട് അടുത്ത് ഇരിക്കാൻ പറഞ്ഞത്	aaredaa ninnodu aduth irikkan paranjath	Angry
അടുക്കളേൽ വേറെ പണിയൊന്നുല്യാ ചേട്ടത്തിക്ക്	adukkalel vera paniyonnulya chettathikk	Angry
അതെന്താ അല്ലിക്ക് ആഭരണം എടുക്കാൻ ഞാൻ കൂടെ പോയാല്	athentha allik aabharanam edukkaan njaan kooda poyaalu	Angry
ആരെടെയ് ഇവനൊക്കെ	aaredey ivanokke	Disgust
അസ്ഥികൂടം അല്ല അഗസ്ത്തികൂടം	asthikoodam alla agasthikoodam	Disgust

Sample Speech in the Dataset	English Transcription	Emotion
അയ്യേ ആ ഉള്ളിത്തോലോ	ayyee aa ullitholo	Disgust
ഛീ ഞാൻ അത്തരക്കാരൻ അല്ല	chee njaan atharakkaaran alla	Disgust
ഛീ വിവരദോഷി	chee vivaradoshi	Disgust
ആരാ അവിടെ	ara avde	Fear
അയ്യോ പ്രേതം	ayyo pretham	Fear
എനിക്ക് പേടിയാവുന്നു	enikk pediyavunnu	Fear
ആരാ അത്	aara ath	Fear
എന്റെ കയ്യും കാലും വെറച്ചിട്ട് വയ്യ	ente kayyum kaalum verachitt vayya	Fear
ഇപ്പൊ സന്തോഷാ എന്റെ കണ്ണുകളെ എനിക്ക് വിശ്വസിക്കാൻ കഴിയുന്നില്ല	ippo santhoshaa ente kannugale enikk vishvasikkan kazhiyunnilla	Happy
കുട്ടേട്ടാ ഞാൻ പാസ്സായി	kuttettaa njaan passaayi	Happy
നല്ല രസ്സായിരുന്നു ആന്റി	nalla rassairunnu aunty	Happy
ഞാൻ പ്രതീക്ഷിച്ചതിനേക്കാൾ എത്ര നല്ല സ്ഥലം	njaan pratheekshichathinekaal ethra nalla sthalam	Happy
വളരെ വളരെ സന്തോഷം	valare valare santhosham	Happy
ആശങ്കയിലാക്കിയത് മുല്ലപ്പെരിയാർ ആയിരുന്നു	aashankayilaakiyath mullaperiyaar aayirunnu	Neutral
അഭിമാനത്തോടെ ജീവിക്കുന്നതും	abhimaanathode jeevikunnathum	Neutral
അത് അല്ലെങ്കിൽ ടിക്കറ്റ് എടുത്ത് കൊണ്ട്	ath allenkil ticket eduth kond	Neutral
അത് കാണാൻ ലോകത്തിന്റെ നാനാ ഭാഗങ്ങളിൽ നിന്നും ആളുകൾ ഇങ്ങോട്ട് വരുന്നു	ath kaanan lokathinte naana bhagangalil ninnum aalukal ingott varunnu	Neutral
ഭൂകമ്പത്തിന്റെ കാര്യങ്ങൾക്കു മാത്രം ആയിട്ട്	bhoogambathinte kaaryangalk maathram aayit	Neutral
ആ ഗങ്ങേനെ ഒന്ന് കാണാൻ പറ്റിയാ മാത്രം മതിയായിരുന്നു എനിക്ക്	aa gangane onn kanaan pattiyaa matram mathiyaayirunnu enik	Sad
ആ പാവം സ്ത്രീയുടെ പ്രാർഥനക്കും കണ്ണീരിനും എന്റെ മുന്നിൽ ഒരു വിലയും ഇല്ലാണ്ടാവല്ലേ	aa pavam sthreeyude kanneerinum prarthanakkum our vilayum illandaavalle	Sad
ആരെങ്കിലും കണ്ടാൽ പിന്നെ ചോദ്യായി പറച്ചിലായി	aarenkilum kandaal pinna chodyaay parachilayi	sad
അച്ഛനെ ആരൊക്കെ കൈവിട്ടാലും വിടാത്ത ഒരാളുണ്ട്	achane arokke kaivittalum kaividatha oralund	Sad
അതാ തനിച്ച് ഇവിടെ വന്നിരിക്കണെ	atha thanich ivide vannirikkunne	Sad
ഏതൊക്കെ സെന്റാ അവളടിച്ചിരിക്കുന്നെ	ethokke scenta avaladichirikkunne	Surprise
ഹാ അപ്പൊ നീയും	haa appo neeyum	Surprise
ഹായ് മഴ	hai mazha	Surprise
ഏ സോഫിമോൾ എപ്പോ വന്നു	ae sophimol eppo vannu	Surprise
ഏ നീയാ	ae neeyaa	Surprise

4.6.7 Phase 7: Construction of AMSC-6

In the process of constructing AMSC-6, a diverse and extensive dataset was carefully compiled from YouTube videos. The dataset comprises speech samples from seven distinct regions in Kerala, with each region contributing a substantial number of samples. The statistics reveal the geographic distribution of the collected data: Thiruvananthapuram with 700 samples, Thrissur with 728 samples, Malappuram with 700 samples, Kottayam with 701 samples, Kasaragod with 476 samples, Kannur with 700 samples, and Kozhikode with 596 samples.

This geographically varied dataset is integral to ensuring the representation of diverse accents within the Malayalam language. The number of samples from each region enhances the robustness and reliability of the dataset, that can be transformed into a solid foundation for the subsequent phases of experimentation and analysis in accented speech recognition. Figure 14 depicts the statistics of AMSC-6 which represent the statistics of the original dataset that has been curated. The data is then augmented using different speech augmentation techniques and is discussed in detail in chapter 14.

Recognizing the need for a comprehensive dataset, the limitations posed by the initial sample distributions are addressed through the application of various data augmentation techniques. These techniques proved instrumental in expanding the dataset, ensuring a more robust and diverse representation of accented speech across different regions. The augmented sample distributions are as follows: Thiruvananthapuram - 2100, Thrissur - 2184, Malappuram - 2100, Kottayam - 2103, Kasaragod - 1428, Kannur - 2100, and Kozhikode - 1788 speech samples and hence a total of 13,803 samples.

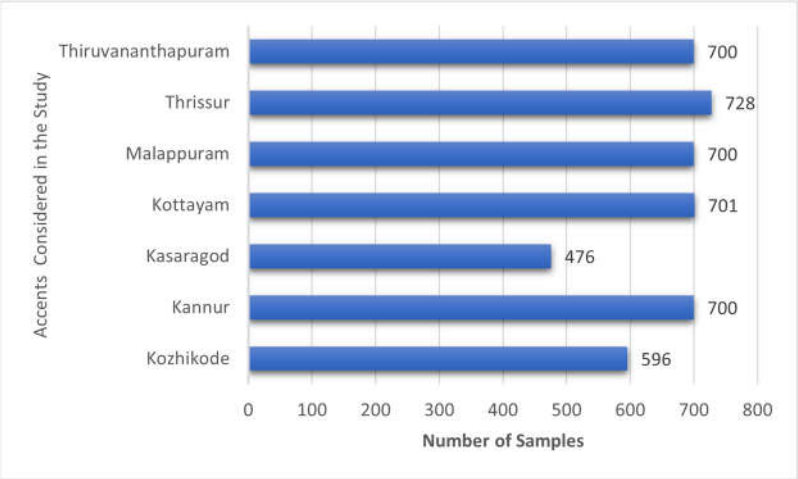


Figure 14 The Statistics of AMSC-6 (Original Data)

The augmentation process not only improved the initial constraints in data collection but also contributed to the overall richness and variability of the dataset. This augmented dataset serves as a foundation for the subsequent stages of the research, enhancing the reliability and generalizability of the accented speech recognition models developed in this study.

4.6.8 Phase 8: Construction of Accented Malayalam Dataset for Detecting Hate Speech (AMDDHS)

The speech categorized as hate speech is examined in detail prior to the selection of the video from YouTube platform for extracting the audio contents. After a rigorous selection process 850 samples were collected from non-hate speech domain and 150 samples were collected from hate speech domain which is represented in Figure 15.

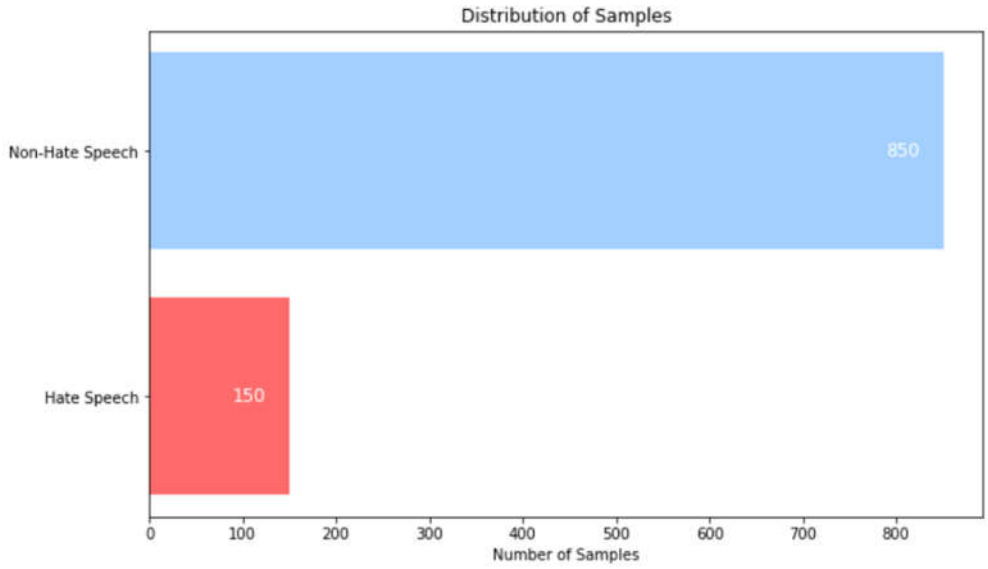


Figure 15 AMDDHS (Original Distribution)

Data augmentation techniques play a pivotal role in introducing variations that mirror the inherent diversity within the speech recordings, contributing to the dataset's authenticity. Data validation is conducted to guarantee that the collected dataset accurately reflects natural speech patterns, accents, and language use.

4.6.9 Spectrograms

In the domain of Accented Speech Recognition, a pivotal step in the preprocessing pipeline involves the transformation of raw audio data into a format contributing to deep learning model training. To achieve this, spectrograms, which serve as image representations of the audio data, are generated. A spectrogram represents the spatial spectrum of frequencies over time, essentially encapsulating the frequency content of an audio signal at different points in time. This transformation is instrumental in converting the temporal nature of speech data into a format that is amenable to deep neural networks, which excel at processing spatial information.

The spectrogram generation process involves breaking down the audio signal into short overlapping segments, calculating the Fourier Transform for each segment to obtain the frequency content, and then plotting the resulting spectrum over time. The outcome is a 2D matrix where one axis represents time, another represents frequency,

and intensity at each point signifies the magnitude of the corresponding frequency component. This representation encapsulates both temporal and frequency characteristics of the audio signal, making it a powerful input for the neural networks.

Spectrograms are image representations, from which the model gains the ability to determine intricate patterns and variations in the frequency domain, which are critical for accurately recognizing and distinguishing accented speech. The image-like nature of spectrograms enables the application of image processing techniques and the utilization of pre-trained image-based models, providing a versatile and effective means of processing audio data for accent recognition. 4000 spectrograms that correspond to the AMSC-5 dataset has been constructed to conduct experiments in this study.

The transformation of audio data into spectrograms stands as a pivotal preprocessing step, enabling the utilization of deep learning models, particularly CNNs, for accented speech recognition. This approach harnesses the power of visual representations to capture both temporal and frequency information, enhancing the model's capacity to recognize the subtleties present in accented speech across diverse regional variations in the Malayalam language. The sample spectrograms used in the study are depicted in Figure 16.

Figure 16 Sample Spectrograms

4.6.10 Ethical Considerations

All the data sourced from YouTube platform are publicly available videos. Personal identifiers, if any, were removed from the audio clips to maintain the anonymity of the speakers. Moreover, the data was used exclusively for academic and research purposes, ensuring there were no breaches of privacy or ethical guidelines.

The data collection process was careful and aimed at ensuring a comprehensive dataset that could truly represent the diversity and complexities of accented Malayalam speech. The diverse accents and clear categorization set a solid foundation for the subsequent phases of the study.

4.7 Conclusion

To address the intricate subtleties of diverse accents prevalent in various regions and demographic groups within Kerala, comprehensive datasets have been curated through an exhaustive and precisely designed process. These datasets, standing as the foundation stone of this research journey, serve as a testament to the potential of techniques employed in modeling and advancing the understanding of AASR. Throughout this research, each step has been taken with precision and purpose, amplifying the authenticity and richness of the speech corpus. The foundational phase of defining the research context established a clear path, enabling the subsequent stages to progress seamlessly.

The strategic district selection, coupled with the engagement of participants from diverse backgrounds, ensures that the dataset encapsulates the essence of linguistic diversity that characterizes the region. The phases of participant recruitment and data collection have been marked by a genuine dedication to capturing natural accents. The individualized approach to communicating with participants and providing them with explicit instructions reflects an ethical commitment to participant consent and data privacy.

Accurate accent annotations and contextual labels contribute to the dataset's depth, enhancing its potential for diverse applications. The incorporation of data augmentation techniques introduces variations that echo real-world scenarios, enriching the dataset's ability to mimic the complexities of accented speech. The journey of generating these datasets has illuminated the reciprocated relationship between linguistics, technology, and diversity. This dataset, thoroughly crafted, embodies the synergy between innovative research methodologies and a profound understanding of regional linguistic variations.

5. AASR of Malayalam Isolated Words using LSTM-RNN

5.1 Introduction

Historically, human-machine communication was constrained to text-based commands. Over recent decades, speech recognition has emerged as an exciting research domain, with significant progress in various languages. ASR for the Malayalam language has not seen extensive research, primarily due to the language's inherent complexity. Adding accent-based variations to Malayalam ASR poses even greater challenges, demanding the utmost dedication and expertise from researchers. The scarcity of datasets for Malayalam further complicates the process. This proposed study aims to create an innovative approach for AASR in Malayalam using LSTM to identify words spoken by different individuals.

To summarize, the contributions of this study include:

1 The construction of a Malayalam word-based embedding to correlate spoken utterances with textual words.
2 The creation of an LSTM-RNN acoustic model specifically tailored to recognize AASR for Malayalam language.

5.2 Methodology

The proposed work is aimed at developing a model that recognizes multi accented speech in the Malayalam language and translates it into corresponding text. The model is trained and constructed with 20 classes of isolated Malayalam utterances, divided equally between Malayalam numerals and a selection of random words in the language. The model is designed to learn from the various classes of utterances it has been exposed to during training and convert the speech into standardized Malayalam text.

The algorithm of the study:

i. Utilize AMSC-1 dataset.

ii. Extract the MFCC speech coefficients corresponding to the speech signals in the dataset.

iii. Construct and design the LSTM-RNN neural network architecture.

iv. Construct the AASR

v. Develop the language model.

vi. Predict and evaluate the model outcomes.

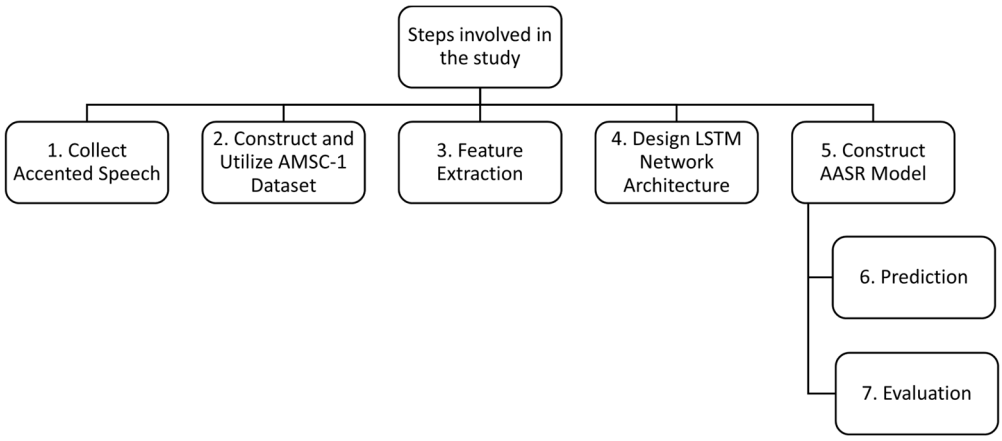


Figure 17 Workflow of the Proposed System

The AASR model constructed in this phase of the research is AMSC -1 dataset which includes 3070 utterances of 20 classes of accented Malayalam acoustic data. Figure 17 illustrates the workflow of the proposed model for accented speech recognition, comprising several key components and steps:

5.2.1 Collect Accented Speech

The initial step involves gathering audio recordings from speakers with various Malayalam accents. This collection process ensures a diverse and representative dataset that captures different regional and social accents, including isolated words, sentences, and continuous speech.

5.2.2 Construct and Utilize AMSC-1 Dataset

Once the raw audio data is collected, it is organized into a structured dataset known as AMSC-1. This dataset is carefully labeled and segmented, forming the foundational dataset for subsequent processing and experiments.

5.2.3 Feature Extraction

In this step, relevant features are extracted from the audio recordings to facilitate speech recognition. MFCC algorithm is used to extract the feature vectors which encapsulate the essential acoustic properties of the speech signals.

5.2.4 Design LSTM Network Architecture

An LSTM network architecture is then designed to handle the sequential nature of speech data. LSTM networks are well-suited for this task due to their ability to maintain long-term dependencies and learn temporal patterns in the speech features.

5.2.5 Construct Accented ASR Model

The accented ASR model is constructed by training the designed LSTM network on the feature-extracted data from the AMSC-1 dataset. During this phase, the model learns to map input speech features to corresponding text transcriptions, effectively recognizing and transcribing accented speech.

5.2.6 Prediction

Once the ASR model is trained, it is used to predict transcriptions for the speech data. The model processes the input speech signals through the trained LSTM network and outputs predicted text transcriptions, demonstrating its capability to generalize to different accents.

5.2.7 Evaluation

The final step involves evaluating the performance of the ASR model using various metrics such as accuracy, precision, recall, and F1-score. This evaluation is conducted

on a separate test set to ensure unbiased assessment, highlighting the model's strengths and identifying areas for improvement.

5.3 Speech Signal Processing and Feature Vectorization

Speech signal processing and vectorizing the appropriate features are crucial in developing AASR models. Selecting the correct features can contribute significantly to training an effective model, while incorrect or irrelevant features can substantially impede the training process.

In this experiment, the MFCC algorithm is used to extract features from speech signals. MFCC is a widely accepted method in speech and audio processing, and it aims to mimic the logarithmic perception of loudness and pitch in human hearing. The following procedure details the extraction of MFCC features.

A pre-emphasis filter is applied on the speech signals to highlight those parts of the signal corresponding to the high frequency representation of the signal. The filter can be described by [154]:

$$Y(t)=X(t)-\alpha \cdot X(t-1) \quad (14)$$

Where $Y(t)$ is the output signal at time t, $X(t)$ is the input signal at time t, α is the pre-emphasis factor, the values normally range between 0.9 and 1.0 and $X(t-1)$ is the input signal at the previous time step, i.e., time $t-1$.

After applying the pre-emphasis on speech signal, it is then divided into frames. The discontinuities in the frame function are minimized by applying a window function to the signal. The frequency components of the frames are analyzed by applying Fast Fourier Transform (FFT). According to Haytham Fayek's [155] detailed explanation on speech processing, after slicing the signal into frames, a window function such as the Hamming window is applied to counteract the assumption made by FFT that data is infinite and to reduce spectral leakage. This ensures that the frequency components are well represented in the subsequent FFT analysis. The power spectrum is then computed as [155]:

$$P(k)=|FFT(x(n))|^2 \quad (15)$$

where, x_n is the n^{th} frame of signal x

The Mel scale simulates the human ear's response to different frequencies. A set of triangular mel filters is applied to P(K) power spectrum to compute the Mel-frequency outputs. The logarithm of the Mel-frequency response is taken, and the Discrete Cosine Transform (DCT) is applied to de-correlate the coefficients, leading to the MFCCs.

In this specific experiment, 20 prominent features are considered. This typically includes the first 13 MFCCs plus the first 7 delta coefficients. The combination of these offer of these features offers a rich representation of the speech signal, capturing essential spectral characteristics and dynamic changes over time.

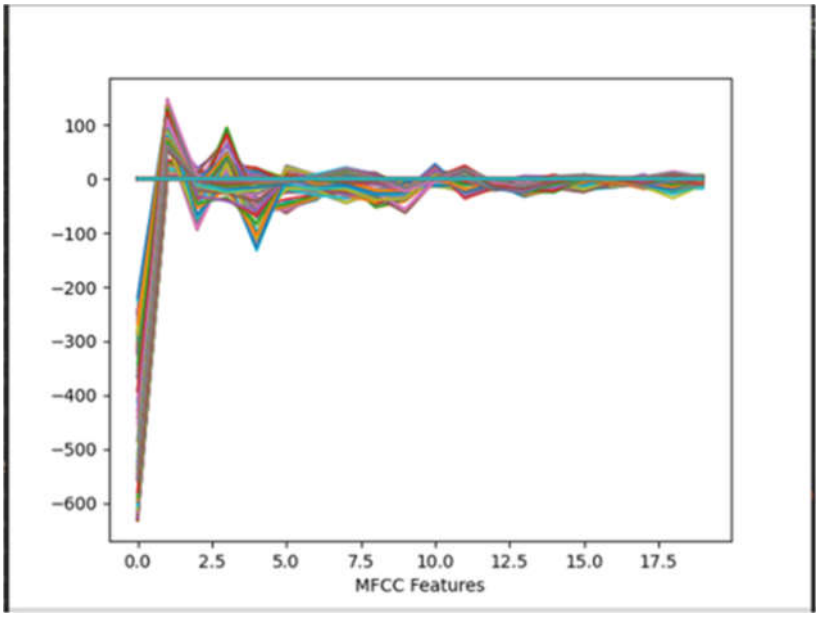


Figure 18 MFCC Features

The 20 features provide a comprehensive view of the speech signal, containing important information on the tonal and rhythmic aspects of the Malayalam language, which is crucial for the task of accent-based speech recognition. Figure 18 visually represents the 20 features extracted from the speech signals, providing an insight into the complexity of the information captured by the MFCC technique.

5.4 AASR using LSTM-RNN

This section outlines the architecture, training, and implementation of the model, which is designed specifically to understand the accents in Malayalam speech signals.

The following steps describe the RNN algorithm:

1. Current Input at time T, X_T denotes the input at the present moment. X_{T-1} refers to the previous input, and X_{T+1} corresponds to the upcoming input, representing sampled speech signals and the hidden State, S_T signifies the hidden memory, that can be computed as

   $$S_T = f(U \cdot X_T + W \cdot X_{T+1}) \quad (16)$$

   where f is the activation function, and U and W are weight matrices.

   This approach is inspired by sequence modeling techniques as detailed by Graves [156] and further developed in sequence-to-sequence learning frameworks by Sutskever et al., [157] and recurrent neural network language models by Mikolov et al., [158].

2. Output at Time Step T, O_T: This refers to the vector of probabilities for 20 classes of isolated words, computed as

   $$O_T = \text{softmax}(V \cdot S_T) \quad (17)$$

   where V denotes the weight matrix associated with the hidden state [159],[160],[161].

The network architecture is fed with the dimensions of accented speech signals, with width representing features extracted through the MFCC algorithm. In the experiment, the height is fixed at 1000. An unfolded RNN architecture is represented in Figure 19.

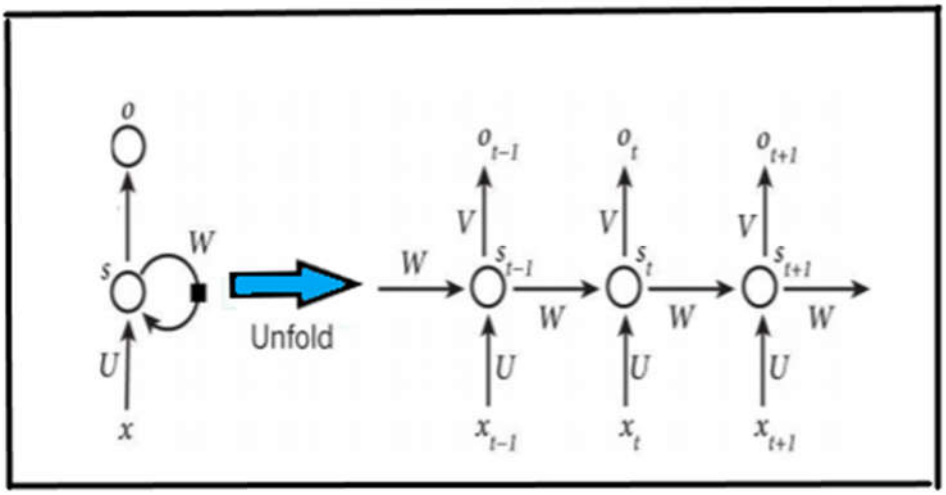


Figure 19 The RNN

Audio signals are feature engineered to extract the prominent frequency vectors that are extracted using MFCC for model training. The speech classes encoded and fed along with the appropriate feature sets to the LSTM network. The dataset contains 20 classes of accented isolated speech which then form the base for the AASR model. The LSTM-RNN undergoes training on the dataset and is aligned with word models. In the testing phase, features extracted from the test set are arranged by the trained network based on the pretrained classes. Throughout the training process, values are compared with the target class, and weight adjustments are made, enhancing the network to make predictions. To construct the AASR system a unique approach was undertaken to collect authentic speech samples from various regions. Specifically, samples were gathered from five diverse districts in Kerala which is rich in linguistic diversity.

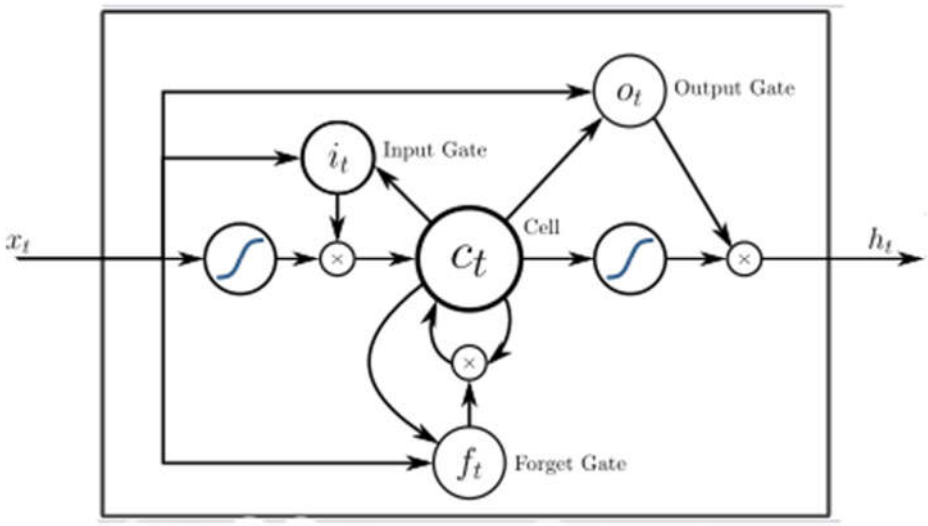


Figure 20 A Long-Short-Term Memory Cell

The data collection process was performed in natural recording environments, rather than controlled studio settings recognizing the need for real-world performance. This decision inherently introduced background noise into the recordings, adding a layer of complexity to the study and hence contributing to the practical applicability of the model. Figure 20 provides a view of an LSTM cell.

Each audio data in AMSC-1 was sampled at a frequency of 16000 Hz and then converted into the .wav format. This uniformity in the data format ensured that subsequent processing and feature extraction would be consistent across the entire dataset. The data collection procedure for this work was characterized by a strategic blend of technology and ground-level engagement with speakers in their natural surroundings. The resulting dataset, rich in accent variations and embedded with real-world challenges such as noise and other live disturbances formed the foundation for training the model. Table 6 contains the statistics of the AMSC-1 dataset prior to model construction.

Table 6 AMSC-1 Dataset

	Phases	**Quantity**
AMSC 1- Dataset	Training	2454
	Testing	616
	Total	3070

Eighty percent of the samples, totaling 2454 speech instances, were allocated to the training process, ensuring a substantial amount of data for the model to learn various aspects of accents and pronunciations. The remaining data was reserved for testing the model's efficiency and accuracy.

The training process with LSTM-RNN was intensive, as observed in the early stages where the classification loss was high. This initial high loss rate reflected the model's unfamiliarity with the complex patterns in the Malayalam accented speech. However, as the training progressed, the algorithm incrementally learned from the speech samples, adapting, and optimizing its internal parameters.

Over a series of 7,38,000 iterative steps, the model's performance improved significantly. Each step in the training process involved validation, which continuously calculated the classification loss, allowing for monitoring and adjustment if necessary. This consistent attention to detail contributed to the progressive reduction of loss as the algorithm advanced through its learning stages. Conducted over GPU to take advantage of its processing capabilities, the training took 12.5 hours for constructing the model.

The time and computational power invested were well-rewarded as the total loss, comprising both classification and validation loss, decreased to 0.25% towards the final stages of computation. This marked reduction in loss symbolized the model's ability in recognizing and translating accented speech and formed the base for model testing and can be adopted to develop potential real-world applications.

Figure 21 presents a graphical representation of the total loss in correlation with the computational steps during the training process of the accented speech recognition model using RNN. The graph provides insightful information on how the model's efficiency improves as the loss reduces over time. Two distinct lines are depicted in the graph. The faded line denotes the original classification loss, exhibiting the fluctuations in the beginning and then decreasing towards the end of model construction as the algorithm learns from the speech samples. This line provides a raw and unfiltered view of the loss evolution through the computational steps. The darker line has undergone smoothing with a factor of 0.6. This smoothing process offers a more refined and stable perspective, eliminating noise and emphasizing the underlying trend in loss reduction. The convergence of the darker line towards lower values symbolizes the model's increasing capability in recognizing accented speech.

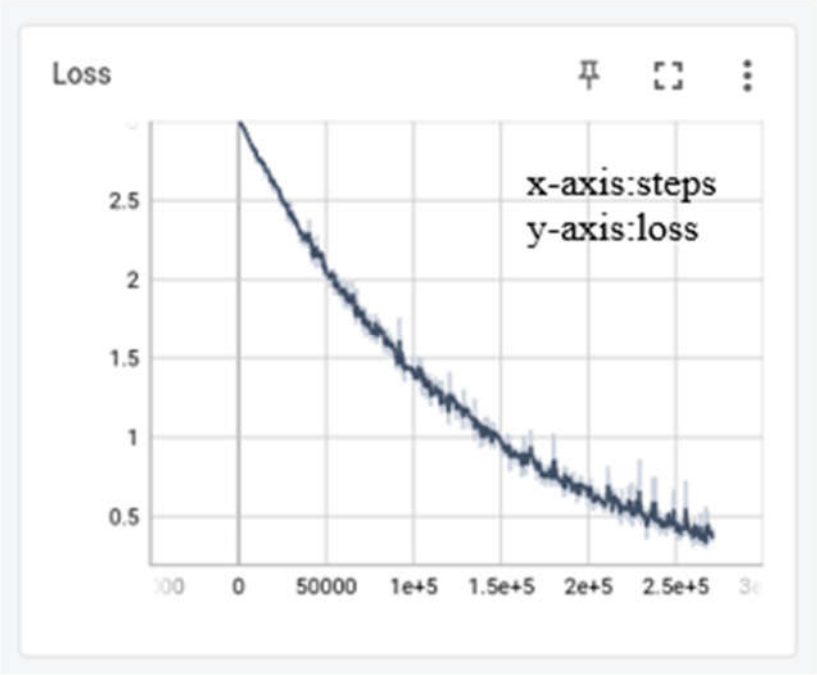


Figure 21 Total Loss vs Computational Steps

The training and validation accuracy during each computational step illustrated in Figure 21 serves as a visual proof to the model's learning process. It emphasizes the fundamental principle that a higher accuracy indicates a better-performing model, reflecting the robustness with which it was constructed.

5.5 Performance Evaluation

This research focuses on developing an AASR system for isolated Malayalam words. The methodology employs the LSTM-RNN algorithm, widely recognized for its efficacy in speech recognition experiments. The dataset comprises 3070 speech samples obtained from twenty-nine speakers spanning diverse age groups and five distinct districts. Figure 21 represents the total loss and Figure 22 represents the total accuracy of the model construction.

Importance was given to the age group of 21 to 40 while collecting the samples, which constituted 42 percent of the total data, with the expectation that this group would contribute high-quality recordings. Several recordings of the same acoustic class were recorded from each speech donor to construct a comprehensive dataset.

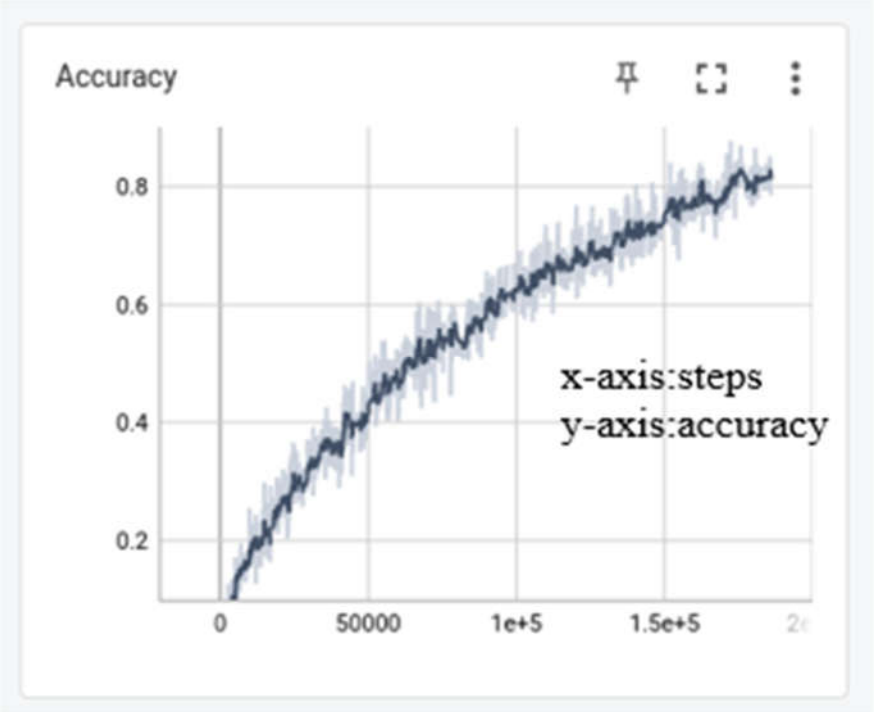


Figure 22 Accuracy vs Computational Steps

The training phase was constructed using 2454 speech samples, constituting 80 percent of the entire dataset. The training of the model involved 738,000 computational steps, utilizing a batch size of 20, spanning a duration of 12.5 hours

and 3000 epochs, at a learning rate of 0.0001. The AASR model was constructed, achieving an accuracy rate of 82.5 percent, representing a significant accomplishment in modeling Malayalam accented speech.

This experiment contributes a significant advancement to the field of accent-based speech recognition, specifically focusing on the Malayalam language. The successful deployment of the LSTM-RNN algorithm and the careful selection of a representative dataset provide promising prospects for future research and practical applications.

5.6 Conclusion

In the study of enhancing the field of AASR for the Malayalam language, an accent-specific dataset has been successfully constructed and developed an efficient model based on the LSTM-RNN algorithm. This effort is especially noteworthy, given the existing scarcity of accent-based data in Malayalam. This research provides a significant contribution to research in AASR, displaying promising results when tested with real-time test cases. However, some false positive predictions have been observed, a phenomenon that might be attributed to the dataset's size or the inherent complexity of accent-based data in Malayalam.

6. AASR with Deep-CNN, LSTM-RNN, and Machine Learning Approaches

6.1 Introduction

In the Malayalam language, the variation in accents corresponds to specific factors such as geographic location, religion, community, social class style, gender, and age. The Dravidian Encyclopedia reports the existence of 15 regional dialects of Malayalam, contributing to the complexities involved in the development of ASR systems. In this experiment, the focus is on five distinct dialects spoken across five districts in Kerala.

The goal is to create an accent-independent ASR system that can effectively operate across these five dialects. The AASR model constructed in this phase of the research was trained on the dataset of audio samples encompassing a wide range of accents, thus preparing it to function effectively across diverse dialects. This research is conducted in three distinct stages: Dataset Preparation, Acoustic Signal Processing, Classification and Prediction, utilizing both machine learning and deep learning strategies. The contribution of this work includes:

1. Dataset Preparation: An accent-based dataset AMSC-2 was created for this experiment.

2. Feature Engineering: Feature engineering involved adopting various speech signal processing techniques to obtain the optimal representation of the accent-based speech data, MFCC, STFT, and Mel Spectrogram methods.

3. AASR Model Construction

    1. Machine Learning AASR Models: This phase of the research deals with the construction of AASR using different machine learning algorithms such as Multi-layer Perceptron, Decision Tree, Support Vector Machine, Random Forest, K-Nearest Neighbor, and Stochastic Gradient Descent.

2. Deep Learning AASR Models: In this phase of research, the features were first used to construct an LSTM-RNN-based accented ASR system. Subsequently, the speech signals were transformed into spectrograms, and the features extracted from these images were channeled into a Deep Convolutional Network Architecture.
3. Ensembled ASR system was developed, combining all the individual models. The performance of each experiment was compared to identify the most effective method for modeling an end-to-end AASR system. This work thus constitutes a novel and comprehensive approach to understanding and navigating the complex world of Malayalam accented speech recognition.

Through an in-depth analysis of different strategies, this chapter contributes valuable insights and novel methods to the field of Malayalam ASR. This study, therefore, not only advances the understanding of accented speech recognition but also lays down a cornerstone for further research and development in this complex and under-explored area.

6.2 Methodology

The primary objective of this study is to devise an innovative and effective approach to creating AASR for acoustic signals in Malayalam that can accommodate multiple accents. This study aims to propose a word-based ASR system specifically designed for the Malayalam language, utilizing machine learning, deep learning algorithms, and a novel hybrid approach that synthesizes the strengths of these methods. In the following sections, the outcomes of the experiments using various machine learning approaches, LSTM-RNN, DCNN, and a hybrid methodology on the accented speech data are examined.

The investigation includes a comparative analysis of different strategies for constructing an accent-based ASR system for the Malayalam language. Each approach has demonstrated promising results when applied to low-resource

languages, and the combination of independent models revealed complementary strengths.

It is essential to recognize that the performance of an accented ASR system relies heavily on the specific nature of the dataset, which may vary across languages. Thus, constant experimentation with various methodologies is vital to determining the best approach for a particular language. For this experiment, audio datasets consisting of 20 classes are used and recorded in natural settings. Utilizing crowdsourcing techniques, a corpus of 4000 data points was assembled, including diverse input from speakers of varying ages, genders, and locations.

The steps involved in the experiment are:

1. Dataset Curation.
2. Feature Engineering.
3. Initial AASR Construction: Employed Multi-layer Perceptron, Decision Tree, Support Vector Machine, Random Forest, K-Nearest Neighbor, and Stochastic Gradient Descent classifiers.
4. Deep Learning Application: LSTM-RNN and CNN algorithms were utilized to train the model, subsequently mapped onto the word models.
5. Ensembled ASR Construction: A composite ASR system was formed from the previously constructed models.
6. Testing and Analysis: The test set, randomly drawn from the dataset, underwent preprocessing and feature extraction. These vectors were then analyzed and compared against target classes, and weights were updated accordingly during the training stage. The outcomes of utilizing standard performance metrics, conducting error analysis, comparing the models against baseline measures, and exploring their interpretability in the study revealed the effectiveness of the DCNN models in accent recognition within Malayalam speech. These findings

present significant insights for the future refinement and expansion of the research.

6.2.1 Dataset Curation

For this study, a distinctive dataset ASMC-2 has been created, comprising multiple utterances of accented words in Malayalam. This dataset is inclusive, capturing samples from individuals of all age groups and both genders.

6.2.2 Feature Engineering

By focusing on the inclusion of only essential features in the experiment, a more effectively trained and robust model can be created. Including irrelevant features could significantly impact the quality and accuracy of the produced model. In this experiment, 180 features were carefully extracted using different feature engineering approaches and techniques. The considered features are a composite of various speech components, as detailed below.

6.2.2.1 Speech Vectorization using MFCC

MFCCs are significant in the representation of speech, and they have been employed in this experiment to encapsulate key information in the Malayalam accented speech data. The process of computing the 40 MFCC features is as follows:

1. Compute the First 13 Coefficients (MFCCs): The initial 13 coefficients are extracted using the standard process as described below:

   For each frame,

   i. Segment the signal into short frames.
   ii. Compute the periodogram estimate of the power spectrum for each frame.
   iii. Apply Mel filter bank on the power spectra and sum the energy within each filter.
   iv. Logarithmically transform all filter bank energies.
   v. Perform the discrete cosine transformation on the log filter bank energies.

vi. Mathematically, the first 13 MFCCs can be computed by:

$$c_i = \sum_{n=0}^{N-1} s(n) \cdot \cos\left(\frac{\pi i}{N}\left(n + \frac{1}{2}\right)\right) \quad \text{for } i = 0,1,2,\ldots,12 \qquad (18)$$

where *ci* are the cepstral coefficients, *N* is the window length, and *s*(*n*) is the speech signal as discussed by Rabiner & Schafer [165].

vii. Compute the First Derivatives (Deltas): These derivatives capture the changes and rate of changes in the coefficients over time, adding an additional 13 features. This can be computed as:

$$\Delta c_i = \frac{\sum_{n-1}^{N} n(c_{i+n} - c_{i-n})}{2\sum_{n-1}^{N} n^2} \; for\; i = 0,1,2\ldots.12 \qquad (19)$$

where c_{i+n} is the MFCC of the frame that is *nn* frames ahead of the current frame i. This is used to compute the forward difference in the numerator of the delta calculation. c_{i-n} is the MFCC of the frame that is n frames behind the current frame i. This is used to compute the backward difference in the numerator of the delta calculation [165].

viii. Compute the Second Derivatives (Delta-Deltas): These derivatives capture the changes and rate of changes in the coefficients over time, adding again an additional 13 features [165].

$$\Delta\Delta c_i = \Delta c_{i+1} - \Delta c_{i-1} \quad \text{for } i = 0,1,2,\ldots,12 \qquad (20)$$

ix. Compute the Mean of All Values: One more feature is extracted by calculating the mean of all the values, to get the central tendency. This captures the main characteristics of the speech signal.

$$\text{mean} = \frac{1}{39}\sum_{i=1}^{39} c_i \qquad (21)$$

Here, the initial 13 coefficients are extended by their first and second derivatives, adding an additional 26 features, and then one more feature is extracted by calculating the mean of all the values, making a total of 40 coefficients.

6.2.2.2 Speech Vectorization using STFT

The steps involved in extracting the speech vectors are:

1. Division of the Signal into Frames: The entire speech signal is partitioned into overlapping frames, each subjected to a windowing function, enabling localized frequency analysis over time.

2. Computation of the Short-Time Fourier Transform: The mathematical representation is given as [165]:

$$X(k,t) = \sum_{n=0}^{N-1} x_{\text{frame}}(n,t) \cdot e^{-j\frac{2\pi kn}{N}} \quad (22)$$

Where:

i. $X(k, t)$: This represents the STFT of the signal. It's a function of k (the frequency index) and t (the time index or frame number).

ii. $\sum$: The summation symbol indicates that the following expression is summed over n, from 0 to $N-1$, where N is the length of the frame (e.g., number of samples in each frame).

iii. $x_{\text{frame}}(n, t)$: This is the n^{th} sample of the t^{th} frame of the signal. The signal is divided into overlapping frames, and this part of the equation refers to a specific sample within a specific frame.

iv. $e^{-j\frac{2\pi kn}{N}}$: This is the complex exponential term, representing a discrete Fourier Transform (DFT) kernel. The j is the imaginary unit, k is the frequency index, n is the sample index within the frame, and N is the length of the frame.

v. The factor $\frac{2\pi kn}{N}$ creates a complex sinusoid that corresponds to the frequency bin k. It cycles k times over the frame length N, thus corresponding to a specific frequency.

vi. The exponential function, $e^{-j\frac{2\pi kn}{N}}$, then, is evaluating the contribution of that frequency within the frame.

3. Extraction of Amplitude Information $A(k, t) = |X(k, t)|$:

This computes the absolute value of the complex number $X(k, t)$, providing information about the amplitude of the frequency component corresponding to index k at time frame t [162].

4. Selection of 12 Specific Features: From this amplitude representation, 12 specific features are selected. These features may correspond to frequency bands or statistical representations of the amplitude data, capturing essential patterns and variations.

This approach provides a robust and effective representation of accented speech, enabling the model to discern subtle differences between accents in Malayalam. Müller [243] and McFee et al., [244] in their studies describe that these 12 features include the magnitude spectrum, phase spectrum, spectral energy, spectral centroid, spectral bandwidth, spectral contrast, spectral flatness, Mel-Frequency Cepstral Coefficients (MFCCs), chroma features, tonnetz features, rhythm features, and zero crossing rate (ZCR). The computation of these features from the STFT matrix enables the extraction of valuable information related to the frequency content, timbral texture, spectral envelope, pitch content, harmonic content, and rhythmic characteristics of the audio signal. This comprehensive set of features facilitates a diverse range of audio analysis tasks, including speech recognition, music genre classification, and sound event detection.

This utilization of the STFT aligns with the goal of capturing essential characteristics of the speech signal, such as tonal variations and emphasis, that are particularly relevant to recognizing different accents within the Malayalam language. The detailed analysis afforded by this method adds a crucial layer of complexity to the model, allowing for more accurate modeling of accented speech across various regions of Kerala. It represents a significant step in the methodology employed in this research, contributing to the overall accuracy and effectiveness of the proposed ASR system.

6.2.2.3 Mel Spectrogram

Mel Spectrograms are a powerful tool for audio analysis, offering a visual representation of the speech signal and focusing on low-frequency features that align with human auditory sensitivity. The use of 128 Mel filter banks in the extraction of Mel Spectrogram features can be highly beneficial for understanding speech in the context of the Malayalam language with its diverse accents.

By employing 128 Mel filter banks, the system can achieve more detailed resolution in the frequency domain. This facilitates capturing subtle variations in frequency, which is particularly vital for understanding aspects like tonal variations and emphasis within accented speech. The Mel scale is designed to emphasize the frequencies to which human hearing is most sensitive. Using 128 filter banks allows for capturing these relevant frequencies with good granularity, enabling a comprehensive analysis of the frequency distribution within speech, particularly in the range where the human ear is most sensitive.

The choice of 128 filter banks reflects a balance between capturing sufficient information for the task and not overly draining computational resources. More filter banks mean more features, offering more information for machine learning models to learn from. Yet, it also maintains a balance, not becoming so complex that it's computationally inefficient. It's possible that 128 filter banks have been found to perform well in this language and the accents being studied.

The number 128 may align well with certain pre-processing techniques or neural network architectures, meeting the input size requirements, and harmonizing with other features like MFCCs and STFT components. The choice of 128 Mel filter banks might also represent standard practices in a community, be driven by specific mathematical relationships with other features, or be suitable for certain algorithmic or hardware constraints.

In the context of building an AASR for the Malayalam language that performs effectively across different dialects, these considerations make the choice of 128 Mel filter banks an effective strategy. It enables accurate modeling of aspects like tonal variations and emphasis within accented speech, which are crucial for the success of the speech recognition system in this specific task.

Mel Spectrograms are computed through the following steps:

1. Fourier Transform: STFT can be computed by:

   $$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j\frac{2\pi kn}{N}} \quad (23)$$

   Here, x_n represents the signal in time, and X_k represents the corresponding frequency component [85].

2. Mel Filter Banks: Apply 128 Mel filter banks to emphasize the frequencies most relevant to human hearing.

   $$M(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (24)$$

   This equation translates the frequency f into the Mel scale, focusing on frequencies that are significant to human hearing. The constant 2595 and the fraction 700 are standard in the conversion to the Mel scale [85].

3. Energy in Each Filter: Compute the energy in each filter bank, resulting in 128 values representing the speech characteristics.

   $$E_m = \sum_{f=0}^{F-1} |X(f)|^2 \cdot H_m(f) \quad (25)$$

   where $Hm(f)$ is the m^{th} Mel filter bank. This computes the energy within the m^{th} Mel filter bank. Here, $|X(f)|^2$ represents the power spectrum of the signal, and $H_m(f)$ is the response of the m^{th} Mel filter [85].

Mel Spectrograms play a pivotal role in speech signal processing, particularly in the frequency range where the human ear is most sensitive. By means of equation (10), the signal is transformed into the frequency domain, enabling an analysis of individual frequency components. Using equation (11), the frequencies are then

mapped onto the Mel scale, a perceptual scale that mimics human hearing's sensitivity to different frequency levels. The energy within each of the 128 Mel filter banks is computed using equation (12). This calculation represents the essential speech characteristics, focusing on those most perceivable to the human ear.

These techniques capture essential phonetic information, representing the sound's overall shape. This is vital in recognizing different accents, where slight changes in phonetics can denote different dialects. They are less susceptible to noise and provide a robust feature set, making the speech recognition system more resilient to variations in recording quality. By focusing on the first 40 coefficients, MFCCs provide a compact representation of the sound, balancing complexity and capturing essential information. STFT is a mathematical transformation technique used for processing acoustic signals.

Accents often manifest in unique intonations and timing. Capturing the temporal dynamics through STFT helps in accent identification, as accents can change the way phonemes are pronounced over time. By analyzing the amplitude data of each speech frame, STFT helps in capturing harmonics and other frequency-related features crucial for distinguishing accents.

Mel Spectrograms emphasize low-frequency features, crucial for capturing tonal variations and stress patterns typical in accented speech. With 128 features, the Mel Spectrogram offers a detailed frequency analysis, aiding in the distinction of subtle differences between various accents.

The combination of these features offers a comprehensive and complex analysis of the speech signal, each contributing unique insights into different aspects of sound. In essence MFCCs retrieve phonetic characteristics and noise robustness, STFT retrieves time-frequency representation and harmonics analysis, and Mel Spectrograms are used for human-like perception modeling and low-frequency emphasis. Figure 23 illustrates the phases in speech feature extraction that has been carried out in this study.

Together, they provide a robust and effective feature set for accented speech processing in Malayalam, helping create accurate and sensitive models that can recognize and differentiate the diverse accents within the language. The multi-layered approach helps in capturing the complexity and richness of accented speech, which is essential for effective speech recognition.

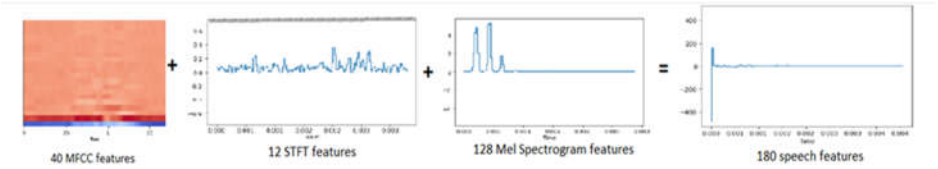


Figure 23 The Speech Feature Extraction

The total number of features *F* is represented by the equation:

F=MFCC+ STFT + Mel Spectrogram

F=40+12+128=180

The feature vector F for accent classification is constructed by combining three distinct feature extraction techniques: Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), and Mel Spectrogram. This comprehensive approach encompasses a total of 180 features, comprising 40 MFCC coefficients, 12 features extracted from the STFT, and 128 features from the Mel Spectrogram. MFCC captures the spectral characteristics of the audio signal by representing the short-term power spectrum of sound, while STFT decomposes the signal into its frequency components over small, overlapping time intervals, providing insights into the signal's time-frequency representation. The Mel Spectrogram further refines the analysis by applying a Mel-scale filterbank to the STFT magnitude spectrum, emphasizing perceptually relevant frequency bands. By combining these three feature extraction techniques, the resulting feature vector F encapsulates a comprehensive representation of the accent characteristics present in the input audio signals, facilitating effective classification and analysis tasks.

6.3 AASR Model Construction

6.3.1 Modeling AASR System

The research adopted three approaches and conducted this study by investigating in detail to obtain better modeling techniques for Malayalam. The subtasks of this study can be categorized as follows:

1. Machine learning based AASR modeling,
2. Deep Learning based AASR modeling and
3. Ensembled Learning based AASR modeling.

6.3.1.1 The Machine Learning Approach

The Multi-layer Perceptron (MLP) model the feature set F as input and these features are processed through a neural network with 3000 hidden layers, where each layer learns various complexities of accented speech. As a result, the MLP produces a multi-dimensional classification of the speech, reaching an accuracy of 94.82%, effectively capturing variations in Malayalam accents. The Decision Tree Classifier used the same set of features to guide its branching logic. At each internal node, the algorithm evaluates a feature to determine the optimal split, thereby representing the rules of accented speech classification. The Decision Tree model produces a relatively lower accuracy of 55.67%, reflecting the complexities of capturing accented speech's complexities with this approach. With Support Vector Machines, the features are used to separate classes using hyperplanes.

The combination of MFCC, STFT, and Mel Spectrogram features aids in finding the best hyperplane that distinguishes the various accents. The outcome of SVM model's ability to generalize accented speech characteristics 66.15% in terms of accuracy metric. By averaging or taking a majority vote from individual trees, it uses the complementary strengths of the features to make the final decision. The model achieves an accuracy of 78.76% after hyperparameter tuning, demonstrating the robustness of ensemble learning.

The KNN model uses the features to calculate distances between speech samples and classifies new samples based on the prominent class of its 10 nearest neighbors. The importance of hyperparameter tuning in KNN is reflected in the optimized accuracy of 81.69%. Stochastic Gradient Descent uses the feature set to find optimal parameters through iterative updates. Although a powerful method, SGD can be sensitive to feature selection and hyperparameter tuning for accented speech recognition, yielding a 33.16% accuracy after tuning. Performance evaluation of the various machine learning classifiers is represented in Figure 24.

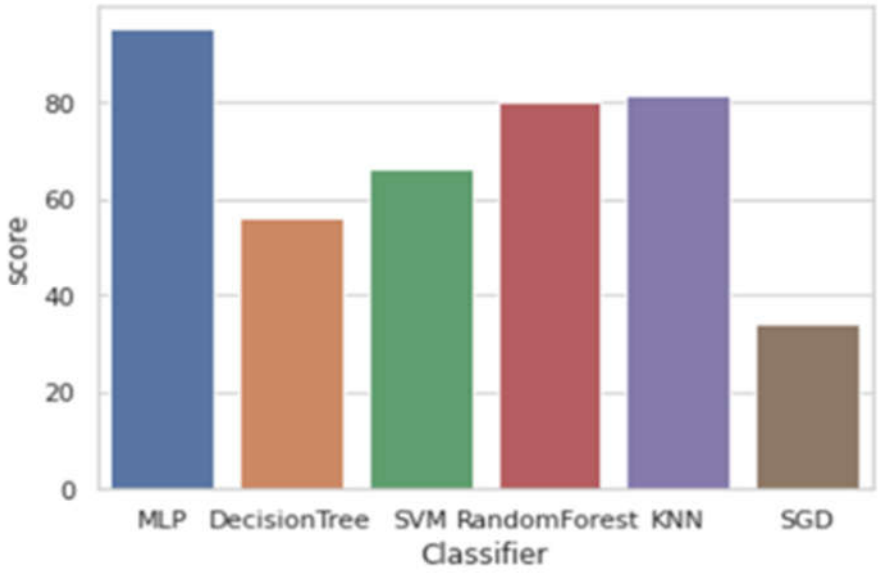


Figure 24 Performance Evaluation of Various ML Classifiers

These machine learning approaches, coupled with the extracted features of MFCC, STFT, and Mel Spectrogram, provided a comprehensive analysis of Malayalam accented speech recognition. The varied performance among different models stresses the complexity of accented speech recognition and the necessity of carefully choosing and tuning models for specific tasks.

The observed differences in performance among the various machine learning models can be attributed to several factors. Firstly, the inherent complexity and non-linearity of the accent classification task may favor certain algorithms over others.

Models like the MLP and KNN, which are more flexible and capable of capturing intricate relationships within the data, tend to perform better when dealing with complex classification tasks. On the other hand, simpler models like Decision Trees and Stochastic Gradient Descent (SGD) may struggle to effectively capture the underlying patterns in the data, leading to lower accuracy rates.

The success of hyperparameter tuning technique, GridSearchCV, also plays a crucial role in enhancing model performance. Models like Random Forest and KNN demonstrated notable improvements in accuracy after hyperparameter optimization, highlighting the importance of fine-tuning model parameters to achieve optimal performance. The varying performance levels observed among the different machine learning models underline the importance of selecting appropriate algorithms, optimizing hyperparameters, and ensuring the quality of the training data to achieve optimal results in accent classification tasks.

These results, summarized in Figure 25, emphasize the varying performance levels of each algorithm in accent classification, with substantial enhancements observed through hyperparameter optimization technique.

6.3.2 Deep Learning Approach

The deep learning approach involves two phases of operations-LSTM-RNN and DCNN.

6.3.2.1 Phase I: ASR with LSTM-RNN

LSTM-RNN architectures are employed to build the accented ASR model, significantly recognizing their capabilities in sequential data processing. Since speech embodies sequential information, the LSTM structure is particularly practiced at retaining relevant aspects and discarding irrelevant parts, an essential attribute in speech signal processing.

Algorithm for Model Building

1. AMSC-2 was constructed.

2. Each audio signal is preprocessed by applying sampling of 16 kHz to each signal.

3. The feature set *F* from the audio signals is then computed using the techniques discussed above. These features encapsulate the distinctive attributes of accented speech.

4. Construct the LSTM-RNN Model: The extracted features are fed into an LSTM-RNN architecture. The LSTM layers are designed to understand the sequential pattern within the features, while the RNN structure allows the model to learn from the temporal dynamics inherent in the speech signals.

5. Make Predictions and Evaluate Performance: The model predicts accented speech and evaluates its performance accordingly.

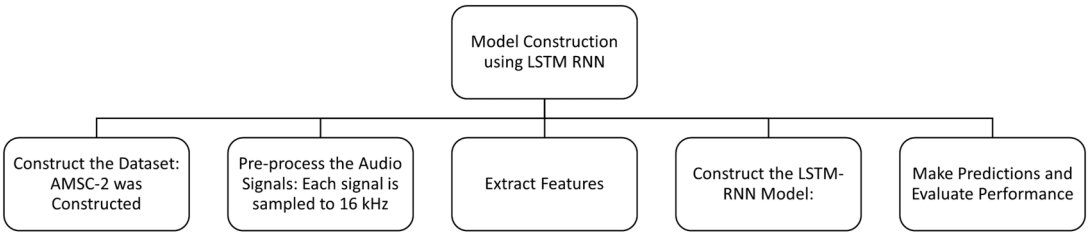


Figure 25 Phases of LSTM-RNN - AASR model

The LSTM-RNN approach's utilization showcases the power of deep learning in accented speech recognition, especially in Malayalam language. By harnessing the strengths of LSTM, such as memory gates and recurrent connections, the model accurately models the time dependencies in speech, making it suitable for this specific domain. The careful selection of features and the robustness of LSTM-RNN offer a promising direction for advanced accented ASR systems. Figure 25 provides an overview of the phases of LSTM-RNN - AASR model that has been constructed.

Model Construction

The LSTM-RNN model was designed, and construction involved utilizing 180 prominent features of speech data, extracted using various methods.

Training

The training of the LSTM-RNN model was a rigorous process involving 98000 steps. The initial training loss was relatively high at 2.5 but demonstrated significant improvement as the model was fine-tuned. The final training loss reduced to 0.24, showcasing the effectiveness of the chosen architecture. The training produced a remarkable accuracy of 95 percent. The visualization of the model's accuracy across training steps illustrates a stable and consistent improvement, reflecting the strength of the model's learning capacity.

Validation

A critical step in evaluating the model's performance was the validation phase, where unseen data was used to test the model's predictions. The LSTM-RNN model achieved a validation accuracy of 82 percent over 98000 steps, reinforcing its effectiveness in handling multi-accent speech classification.

The experiment with LSTM-RNN demonstrated its capacity to classify and recognize various accents within the Malayalam language with high accuracy. By harnessing the temporal dependencies in speech and translating them into a format that the machine can understand, the LSTM-RNN model has set a benchmark for accented speech recognition. The results, visualized across training steps, reflect the robustness and potential of LSTM-RNN in this domain.

6.3.2.2 Phase II: Model Building using Deep CNN

Deep CNNs are recognized for their efficiency in image classification problems. In this experiment, CNNs are employed to process spectrograms of accented speech, treating them as images and using image classification techniques to recognize

different accents. In the construction of the DCNN models, spectrograms served as the fundamental data representation.

These models were crafted to process and analyze the intricate spectrographic details embedded in audio signals. Spectrograms are plotted to represent the frequency distribution of audio signals over time as images and were used as the principal input for these DCNN models. The development of these models was intentionally undertaken to take advantage of the distinctive features encapsulated in spectrograms, aiming to enhance the efficacy of pattern recognition, and learning within the scope of the research. A sample spectrogram used in the study is illustrated in Figure 26.

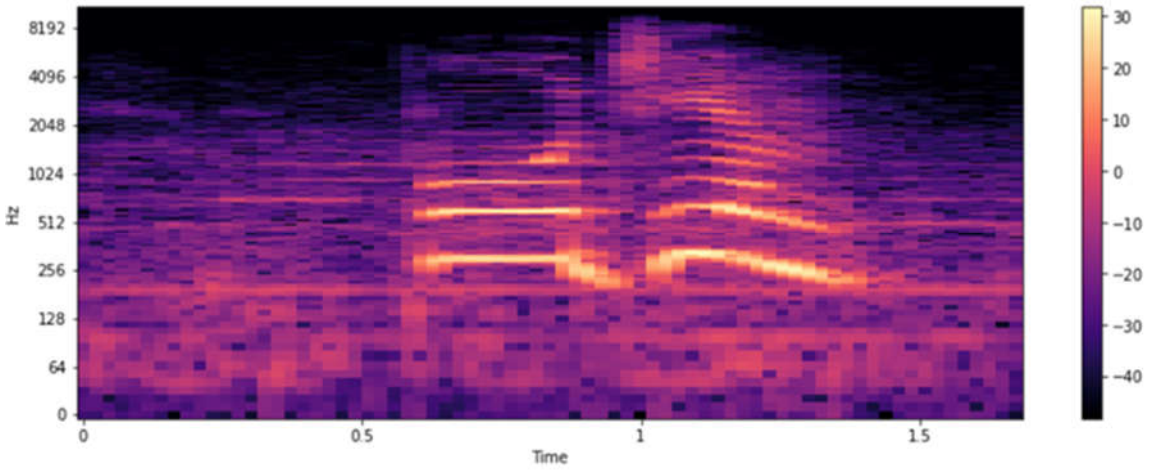


Figure 26 Sample Spectrogram **u**sed in the Experiment

Algorithm for Model Construction

1. Construct the Speech Dataset: Gather a collection of accented speech recordings suitable for the study. The dataset should be diverse, encompassing various accents to ensure the model's robustness.

2. Construct the Spectrogram Dataset: Convert the speech signals into spectrograms. These are visual representations of the frequency content in the speech signals and will serve as the input for the DCNN.

3. Initialize the Model: Set up the initial structure of the DCNN. This includes defining the type of architecture to be used, such as a standard feed-forward network with convolutional layers.

4. Add the CNN Layers: Insert the convolutional layers into the model. These are responsible for detecting patterns and features within the spectrograms, such as specific phonetic characteristics that may be associated with different accents.

5. Add the Dense Layers: Include fully connected dense layers that will process the high-level features extracted by the convolutional layers. These layers are critical for making final accent classification decisions.

6. Configure the Learning Process: Set up the optimization algorithm, loss function, and other hyperparameters that govern how the model will learn from the training data.

7. Train the Model: With everything set up, proceed to train the model using the spectrogram dataset.

8. Evaluation and analysis: After training and constructing the model it should be tested and evaluated to check the efficiency of the methodology with which it was constructed.

By using CNNs and treating spectrograms of speech signals as images, this approach offers a novel and potentially effective way to recognize different accents.

Model Architecture

The initial step involves resizing the spectrogram input to (224,224,3) to establish a standardized input size. This resized input is then processed through a convolutional layer, resulting in an activation size of (222,222,32). Subsequently, a max-pooling layer is applied to downsample the feature maps, producing dimensions of (111,111,32) and focusing on the most important features.

Another convolutional layer follows, leading to an activation size of (109,109,64). This is succeeded by a max-pooling layer of (54,54,64), and a dropout layer is introduced to prevent overfitting. Further, another convolutional layer of (52,52,64) is applied, succeeded by a max-pooling layer of (26,26,64) and a second dropout layer.

Table 7 The Layered Architecture

Layer	**Operation/Description**	**Input Size**	**Output Size**
1	Input Size	(224,224,3)	(224,224,3)
2	Convolutional	(224,224,3)	(222,222,32)
3	Max Pooling	(222,222,32)	(111,111,32)
4	Convolutional	(111,111,32)	(109,109,64)
5	Max Pooling + Dropout	(109,109,64)	(54,54,64)
6	Convolutional	(54,54,64)	(52,52,64)
7	Max Pooling + Dropout	(52,52,64)	(26,26,64)
8	Convolutional	(26,26,64)	(24,24,128)
9	Max Pooling	(24,24,128)	(12,12,128)
10	Flattening	(12,12,128)	18432
11	Dense + Dropout	18432	64
12	Output (Dense)	64	20 Classes

Continuing the sequence, the output undergoes the subsequent convolutional layer, resulting in an activation size of (24,24,128). This is succeeded by a final max-pooling operation, reducing the dimensions to (12,12,128), and the result is flattened into a one-dimensional array with a size of 18432. Subsequently, this flattened array is processed through a dense layer containing 64 neurons, followed by another dropout layer. The concluding dense layer comprises 20 neurons, representing the 20 distinct classes in the experiment, each corresponding to different accents and speech characteristics. Table 7 represents the layers of the CNN architecture in each row.

The Training Process

The model is trained over 4000 epochs, involving a total of 53,000 training steps. Employing a dataset comprising 4000 spectrograms, 80% of the data is dedicated to training, with the remaining 20% set aside for testing. The deep CNN model developed in this study exhibits a sophisticated and resilient architecture designed for accent recognition. By converting speech into spectrogram representations and utilizing a sequence of convolutional, pooling, and dense layers, the model demonstrates an ability to discern complex features that characterize different accents.

Visualization of the CNN Model

Figure 27 provides a comprehensive visualization of the Convolutional Neural Network (CNN) architecture used for recognizing accented speech in the given experiment. The architecture is depicted through various colors, each representing a different layer or aspect of the model.

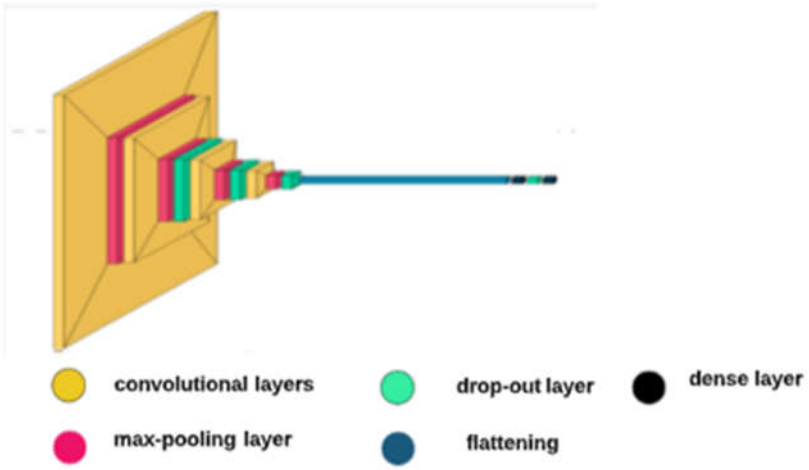


Figure 27 Layered Architecture of the DCNN Model

1. Yellow Layers (Convolutional Layers): These layers represent the convolutional process where spatial relationships in the spectrogram data are identified. Each

yellow block corresponds to a layer that extracts features by applying a convolution operation.

2. Red Layers (Max-Pooling Layers): These layers reduce the dimensionality of the feature maps by selecting the maximum value from a set of values within the feature map. This operation helps preserve the most crucial features while eliminating redundant information.

4. Green Layers (Drop-out Layers): These layers help in reducing overfitting by randomly ignoring a subset of features during training. This prevents the model from relying too much on any specific feature, promoting generalization.

5. Long Blue Layer (Flattening Layer): The distinct blue layer symbolizes the flattening process where the 2D matrix of features from the previous layers is transformed into a 1D vector. This transformation prepares the data for input into the fully connected dense layers.

6. Darker Layers (Dense Layers): The darker colored layers toward the output end represent the dense layers of neurons. These are the final stages of the network where the extracted features are combined to form a final prediction, corresponding to the different classes of accented speech in the study.

6.3.3 The Ensembled Approach

Computational complexity was considered, as training and aggregating predictions from multiple models demand significant resources. However, the advantages reported for ensemble methods, such as improved accuracy and generalization, outweigh the computational costs, justifying their inclusion in the model. The ensembled approach stands as an innovative and strategic methodology in the construction of the AASR system. In contrast to individual machine learning and deep learning models, this methodology establishes a foundation for employing the collective intelligence of diverse models, enhancing the precision and resilience of predictions. In the Ensembled approach, all the previously constructed models that

include MLP, Decision Tree, SVM, Random Forest, KNN, and SGD are integrated into a composite system. Each algorithm was carefully considered for its ability to capture different aspects of the data and complement the strengths of others in the ensemble. MLPs excel at learning complex patterns, Decision Trees provide interpretability, SVMs handle high-dimensional data, KNNs offer simple nearest-neighbor classification, Random Forests enhance generalization, and SGD efficiently handles large-scale learning. The ensemble approach utilizes the diverse predictions of these algorithms to improve overall accuracy and robustness [246]. Majority voting technique is employed for constructing the ensembled model. This integration is aimed at capturing the unique strengths of each individual model, thereby enhancing the overall performance. The voting method employed for constructing the ensembled model typically performs better than individual models by aggregating their strengths.

Given the accuracies of the high-performing models:

1. MLP: 94.82% (0.9482)
2. SVM: 66.15% (0.6615)
3. Random Forest: 78.76% (0.7876)
4. KNN: 81.69% (0.8169)

The accuracy metric of the ensemble model is 71.55. In majority voting, at least 4 out of 6 classifiers need to predict correctly for the ensemble to be correct. Binomial distribution can be used to compute this. Let p_i be the probability that the i^{th} classifier is correct. Then, the ensemble accuracy $P_{ensemble}$ is given by the sum of probabilities of getting at least 4 correct predictions out of 6:

$$P_{ensemble} = \sum_{k=4}^{6} \binom{6}{k} . \left(P_{avg}\right)^{k} . (1 - P_{avg})^{6-k} \quad (26)$$

where $\binom{n}{k}$ is the binomial coefficient, P_{avg} is the average accuracy of the individual classifiers, and $(1 - P_{avg})$ is the probability of a classifier being incorrect [245]. It calculates the probability of getting at least k correct predictions out of n classifiers,

where each classifier has an independent probability p of being correct. Calculate the average accuracy of the individual classifiers:

$$P_{avg} = \frac{1}{6}(0.9482 + 0.5567 + 0.6615 + 0.7876 + 0.8169 + 0.3316)$$

$$P_{avg} = 0.68375$$

$$P_{ensemble} = \sum_{k=4}^{6} \binom{6}{k} . (0.68375)^k . (1 - 0.68375)^{6-k}$$

For k=4:

$$\binom{6}{4} . (0.68375)^4 . (0.31625)^2 \approx 15 \cdot 0.219 \cdot 0.1 = 0.3285$$

For k=5:

$$\binom{6}{4} . (0.68375)^5 . (0.31625)^1 \approx 6 \cdot 0.150 \cdot 0.31625 = 0.285$$

For k=6:

$$\binom{6}{4} . (0.68375)^6 . (0.31625)^0 \approx 1 \cdot 0.102 = 0.102$$

$$P_{ensemble} = 0.3285 + 0.285 + 0.102 = 0.7155$$

The ensemble method for accented speech recognition, using majority voting among six different machine learning classifiers, results in an approximate accuracy of 71.55%. This calculation demonstrates that combining individual classifiers-each with varying degrees of accuracy-can yield a robust ensemble model. The improvement in overall accuracy highlights the effectiveness of ensemble techniques in employing the strengths of multiple models to achieve better performance than any single classifier alone.

6.4 Performance Evaluation

This research has constructed eight different AASR models with different unique approaches. The experiment encompassed a wide range of techniques, employing both machine learning and deep learning approaches.

Figure 28 Learning Curves of AASR Constructed with LSTM-RNN

The construction of the DCNN model is based on a dataset containing 4000 spectrograms, where 3020 samples are designated for training, and the remaining 800 samples are utilized for testing. The model demonstrates a train accuracy of 98 percent and a test accuracy of 71 percent. Figure 28 represents the learning curves of the accented model constructed using LSTM-RNN. Figure 29 visually represents the model's accuracy and loss during both training and testing phases, derived from the CNN architecture.

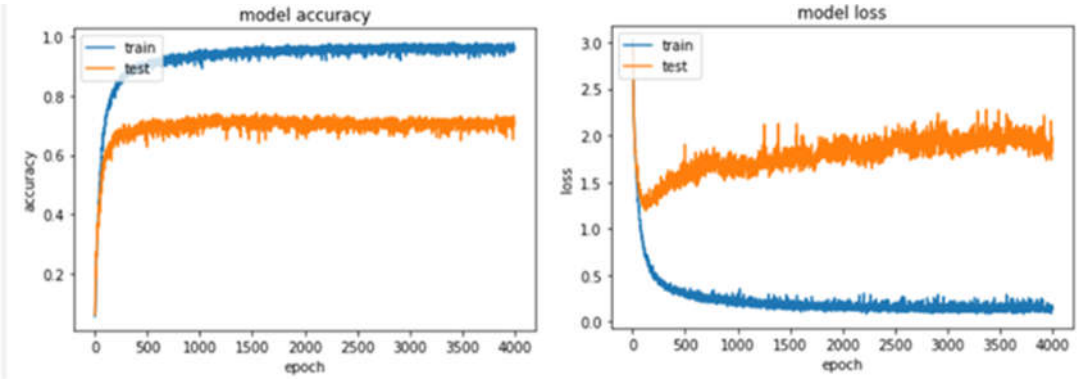


Figure 29 Learning Curves of AASR Constructed with CNN

6.5 Conclusion

The study thoroughly investigated accented speech recognition for the Malayalam language, experimenting with influential and spectral features, including frequency, amplitude, pitch, and aspects like the speaker's age and gender. A high-performance Accented Automatic Speech Recognition (AASR) system was developed through three distinct methods: Machine Learning, Deep Learning (including both LSTM-RNN and CNN models), and a Hybrid Approach.

7. End-to-End Unified AASR -A Low Resourced Context

7.1 Introduction

Accented speech inherently carries significant information that can complicate the construction of robust ASR models. In many cases, existing ASR models show satisfactory performance with known accents but fall short with unknown ones. This highlights the necessity for a novel approach that can adapt and provide accurate recognition regardless of the accent's familiarity.

The subsequent sections of this chapter detail the systematic experiments carried out in feature engineering, including the extraction of accented features and the construction of different unified models. Results are presented and analyzed, underscoring the success of the approach not only with known and unknown accents but also with accent-agnostic standard Malayalam. This chapter discusses the experiments conducted with two distinct datasets AMSC-1 and AMSC-2. The same methodology is applied to these datasets to construct the AASR model.

7.2 Methodology

The initial phase of the methodology involved cleaning the collected data to remove any noise or interference within the signals, laying a clean foundation for subsequent phases. This was followed by a preprocessing stage, preparing the data for a series of intricate feature engineering experiments. The extraction of features was systematically carried out in eight separate phases, terminating in eight different feature sets.

Figure 30 illustrates the comprehensive workflow of the study, detailing the processes from dataset construction through feature engineering to prediction and evaluation. Various feature sets, including MFCC, STFT, Mel Spectrogram, Tempogram, and combinations thereof, are employed. Multiple machine learning algorithms (MLP, Decision Tree, SVM, SGD, KNN, RFC) and ensemble methods are

applied to each feature set, with the aim of identifying the most effective models for accented speech recognition. The final ensemble approach integrates the strengths of these models, leading to a robust and accurate speech recognition system.

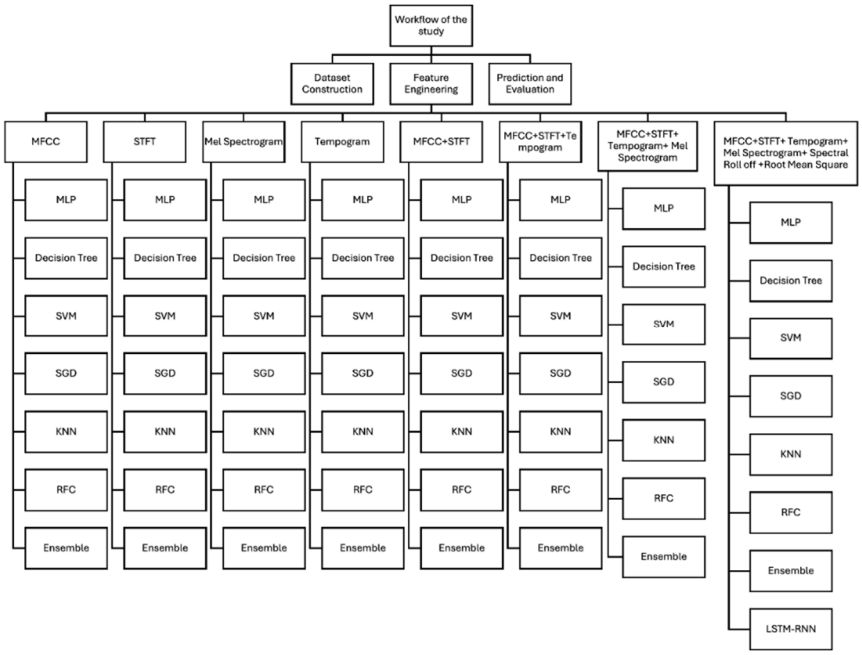


Figure 30 Workflow of the Study

These feature sets were constructed to capture various unique characteristics of accented speech, ranging from simple to complex details. These eight sets served as the basis for the next stage of the methodology, where different accented acoustic models were developed and tested. By employing state-of-the-art techniques in machine learning, deep learning, and LSTM-RNN, the study conducted a series of in-depth experiments with these models on all eight feature sets.

The goal of these experiments was to evaluate and identify the most effective combinations of features for accented speech recognition, contributing to the broader understanding of accented Malayalam speech. The methodology and design proposed in this study represent a systematic and comprehensive approach to the complex task of recognizing and modeling accented Malayalam speech.

Through the incorporation of a diverse dataset, feature engineering, and innovative modeling techniques, this research explores new frontiers in accented speech recognition. It takes into consideration the nature of accents and utilizes a novel blend of methods to tackle the inherent challenges, thus contributing significantly to the field of speech recognition for the Malayalam language.

7.3 Dataset Construction

The construction of the dataset is a critical phase in this research on accented Malayalam speech recognition. Two speech corpuses namely AMSC-3 and AMSC-4 each comprising of 7070 samples has been assembled for this research, captured in a natural environment. These recordings consist of individual utterances of multisyllabic words, each lasting between two to five seconds, forming the basis of the corpora.

To facilitate compatibility with various tools and methodologies, all the data was sampled to 16KHz and converted into the .wav format. The standardization of format ensured seamless further processing and maintained the quality of the original recordings. The resulting corpus stands as a significant achievement, paving the way for an in-depth exploration of accented Malayalam speech recognition.

7.4 Feature Engineering

Feature engineering is a critical process in speech recognition, focusing on the extraction of prominent characteristics from the speech signal that encapsulate essential information. In this study on accented Malayalam speech, feature engineering has been executed in eight distinct phases, each designed to capture different aspects of the speech data.

The utilization of these multiple phases ensures a thorough and multidimensional analysis of the accented speech data. The combined approach showcases the potential for innovative and integrated solutions in feature engineering. Each phase contributes unique insights and perspectives, laying the groundwork for the

construction of various accented acoustic models using machine learning, and deep learning approaches. This methodical and exhaustive feature engineering process is central to the success of the study.

By focusing on various aspects of the speech signal and employing a combined approach, it enables a more complex and intricate understanding of accented Malayalam speech. The insights drawn from these different phases form the foundations for further experiments, significantly contributing to the overall robustness and innovation of the research. In Figure 31 different approaches of speech signal processing that are carried out in eight distinct phases are represented.

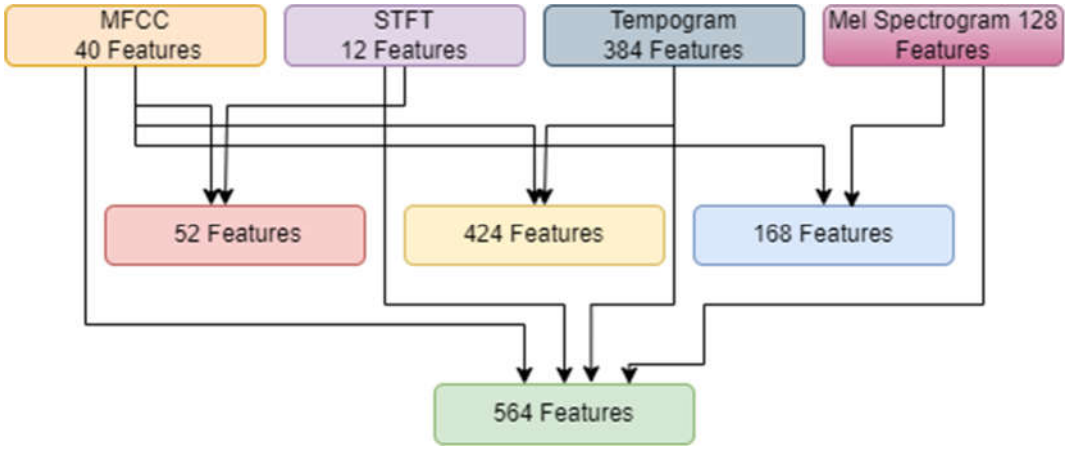


Figure 31 Speech Signal Processing Approaches

The different phases in the feature engineering are discussed below:

7.4.1 Phase I: MFCC (Mel Frequency Cepstral Coefficients)

The extraction of 13 frequency coefficients from the speech signal is accomplished using MFCC, complemented by the inclusion of the second and third derivatives, thereby totaling 39 coefficients. The mean of these coefficients is computed to derive the 40^{th} value, yielding 40 values that effectively represent the audio data and offer a comprehensive frequency illustration.

7.4.2 Phase II: STFT (Short Time Fourier Transform)

STFT extracts 12 prominent amplitude values for time-frequency decomposition. This process provides a time-localized view of frequency variation, offering insight into how frequency components vary over time within the speech signal. Short-Time Fourier Transform (STFT) involves segmenting the audio signal into short frames, typically overlapping, to capture temporal variations [247]. Each segment is then windowed using a window function like Hamming or Hann to minimize spectral leakage [248]. The Fourier Transform is applied to each windowed segment to obtain its frequency-domain representation [191], and the magnitude of the Fourier Transform coefficients is computed to capture the spectral content. By concatenating these spectra over time, a time-frequency representation of the signal is obtained revealing how the spectral content evolves over time.

7.4.3 Phase III: Tempogram Features

Tempogram features are used by extracting 384 features that focus on the rhythmic aspects of speech. This phase analyzes the rhythmic features related to the accented characteristics of the speech signal, capturing its rhythmic components. Tempogram construction starts with computing the autocorrelation function of the audio signal to identify periodic patterns or tempo variations [249]. Peaks in the autocorrelation function correspond to potential tempo candidates, representing the periodicity or rhythmic structure of the signal. These tempo candidates are then used to construct the Tempogram, a two-dimensional representation that displays tempo variations over time. The Tempogram provides a visual depiction of tempo changes in the audio signal, with intensity indicating the strength of tempo variations at each point in time.

7.4.4 Phase IV: Mel Spectrogram Features

Phase IV involves the extraction of 128 features using Mel Spectrogram techniques. This phase unveils hidden characteristics of the speech data, providing deeper insights into underlying patterns. For the Mel Spectrogram, a set of triangular filters

is designed on the Mel scale, which better reflects human auditory perception [251]. The magnitude spectrum obtained from the STFT is multiplied with each Mel filter to obtain the spectral energy within each filterbank [252]. The energy within each filterbank is then summed to obtain the Mel spectrogram, a compressed representation of the spectral content. Logarithmic compression is applied to enhance the representation of lower energy spectral components and improve perceptual relevance.

7.4.5 Phase V: Combination of MFCC and STFT

Here both MFCC and STFT are integrated to form a comprehensive vector representation. By combining these methods, this phase yields a more detailed representation, making use of time and frequency aspects for a rich analysis of the speech data.

7.4.6 Phase VI: Combination of MFCC and Tempogram

The fusion of MFCC and Tempogram is used to capture frequency coefficients and rhythmical patterns. This combination enhances the understanding of frequency and rhythm in speech, providing a detailed analysis of these critical elements.

7.4.7 Phase VII: Combination of MFCC and Mel Spectrogram

Insights from MFCC is combined with Mel Spectrogram features, creating a balanced and informative representation of speech signals. This integration offers a well-rounded analysis, capturing both frequency coefficients and hidden characteristics.

7.4.8 Phase VIII: Combination of MFCC, STFT, Tempogram, and Mel Spectrogram

Phase VIII unites all previous feature sets, including MFCC, STFT, Tempogram, and Mel Spectrogram, to create a comprehensive model of accented Malayalam speech data. By taking advantage of the strengths of each individual approach, this phase results in a unified and robust representation, encapsulating the complexity and

uniqueness of Malayalam accented speech. Together, these phases present a rigorous and well-structured methodology for analyzing Malayalam speech, focusing on various features essential for modeling accented speech.

Practical challenges in feature extraction methods like Short-Time Fourier Transform (STFT), Tempogram, and Mel Spectrogram include selecting appropriate parameters such as window size and type for STFT, balancing time and frequency resolution, which can impact computational complexity [247]. Tempogram computation requires accurate segmentation and windowing for rhythmic pattern analysis, posing difficulties in handling varying tempo and rhythm. Mel Spectrogram computation aims to map the power spectrum onto the Mel scale but faces challenges with non-stationary signals and interpreting features amidst noise [251]. Robustness to noise and artifacts is a common challenge across all methods, demanding empirical experimentation and algorithmic optimization for reliable feature extraction. These challenges highlight the importance of domain expertise and careful parameter tuning to ensure accurate representation of audio signals for subsequent processing tasks.

7.5 Building the Accented ASR System

The development of the accented Automatic Speech Recognition (ASR) system in this study involved several phases of speech signal engineering. Unified accented models were created through eight distinct phases for AMSC-3 dataset and seven distinct phases for AMSC-4 dataset.

The approaches for AASR construction for both the datasets are:

1. Machine Learning based AASR,
2. AASR using LSTM-RNN

7.5.1 AASR Model - Machine Learning Approach

Utilizing Machine Learning techniques, including MLP, Decision Tree, SVM, SGD, KNN, RFC, and ensemble methods, multiple AASR models were developed.

Utilizing different approaches of acoustic signal engineering and feature vectorization, a total of eight models were created.

7.5.2 AASR Model- LSTM-RNN Approach

The construction of an accented model using the LSTM-RNN approach involved the utilization of a combined feature set, achieved through the application of techniques such as MFCC, STFT, Mel Spectrogram, Root Mean Square, and Tempogram. A single AASR model was established and underwent training for 2000 epochs in each experiment. The model exhibited notably high performance, as evidenced by both accuracy and match error rate.

7.6 Performance Evaluation for AASR with AMSC-3

The classifiers evaluated include MLP, DTC, SVM, RFC, KNN, SGD, and an Ensemble method. Different feature combinations to ascertain the impact of each on the performance of the classifiers are investigated in this study. WER is used as the primary metric to gauge the overall accuracy of word recognition, while MER specifically measures the system's effectiveness in handling mispronunciations caused by accents. The LSTM-RNN computes both accuracy and loss to evaluate the performance in recognizing accented speech. These metrics provide insights into the models' training efficiency and their ability to generalize to new data.

7.6.1 Evaluation in Word Error Rate (WER)

Figure 32 illustrates the Word Error Rate (WER) for different classifiers across eight phases of an experiment focused on Malayalam speech data. Figure 32 illustrates a comparative analysis of WER across various machine learning classifiers throughout different phases of feature engineering. The classifiers evaluated include MLP, DTC, SVM, RFC, KNN, SGD, and an Ensemble method.

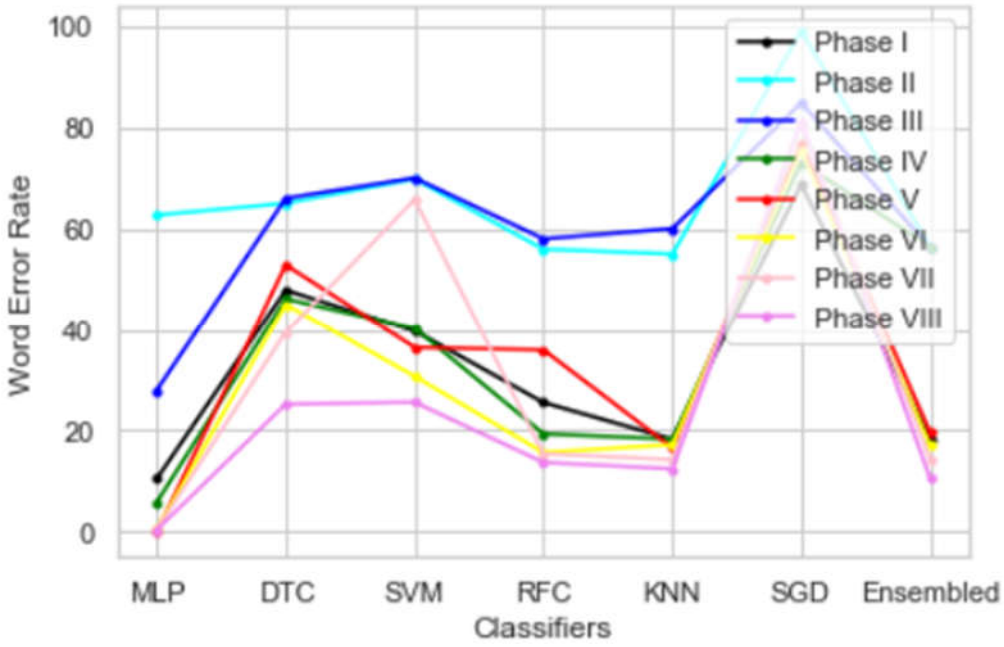


Figure 32 The WER of Machine Learning Approaches

The phases, labeled from I to VIII, represent distinct stages or iterations of the experiment, each corresponding to a specific feature engineering technique or combination of techniques used for analyzing Malayalam accented speech data.

7.6.1.1 Phase I: MFCC (Mel Frequency Cepstral Coefficients)

MFCC involves extracting 13 frequency coefficients from the speech signal, complemented by the second and third derivatives, totaling 39 coefficients. An additional mean value is computed, resulting in 40 values representing the audio data comprehensively. In this phase, the WER for the Ensemble method is lowest around 10%, while other classifiers like MLP, DTC, SVM, RFC, KNN, and SGD exhibit higher error rates, ranging from 20% to 60%.

7.6.1.2 Phase II: STFT (Short Time Fourier Transform)

STFT extracts 12 prominent amplitude values for time-frequency decomposition, offering insight into how frequency components vary over time within the speech signal. This phase shows a significant increase in WER for all classifiers, with DTC

and SGD reaching around 60% and 65% respectively. The Ensemble method maintains a lower error rate of around 10%.

7.6.1.3 Phase III: Tempogram Features

Tempogram features focus on the rhythmic aspects of speech by extracting 384 features related to the accented characteristics of the speech signal. This phase indicates the highest WER for all classifiers, with DTC peaking at 80% and SGD at 70%. However, the Ensemble method consistently outperforms others with a WER around 10%.

7.6.1.4 Phase IV: Mel Spectrogram Features

In Phase IV, 128 features are extracted using Mel Spectrogram techniques, revealing deeper insights into underlying patterns of the speech data. There is a noticeable improvement for most classifiers, except for SGD, which remains high at 45%. The Ensemble method continues to show the lowest error rate of around 10%.

7.6.1.5 Phase V: Combination of MFCC and STFT

This phase integrates both MFCC and STFT to form a comprehensive vector representation, resulting in a detailed representation of speech data. The Ensemble method remains robust with a WER around 10%.

7.6.1.6 Phase VI: Combination of MFCC and Tempogram

The fusion of MFCC and Tempogram enhances the understanding of frequency and rhythm in speech, providing a detailed analysis. Most classifiers show a decrease in WER, with MLP, RFC, and KNN showing values around 25%, while DTC and SGD remain higher at 40% and 45% respectively. The Ensemble method remains the most effective with a minimal WER around 10%.

7.6.1.7 Phase VII: Combination of MFCC and Mel Spectrogram

This phase combines insights from MFCC with Mel Spectrogram features, creating a balanced and informative representation of speech signals. Phase VII demonstrates

a peak in WER for most classifiers, particularly DTC and SGD, with values approaching 80%. The Ensemble method, however, continues to exhibit superior performance with a WER around 10%.

7.6.1.8 Phase VIII: Combination of MFCC, STFT, Tempogram, and Mel Spectrogram

Phase VIII unites all previous feature sets to create a comprehensive model of accented Malayalam speech data. By taking advantage of the strengths of each individual approach, this phase results in a unified and robust representation. Finally, Phase VIII shows an overall decrease in WER for all classifiers, with the Ensemble method still leading with the lowest WER around 10%.

Across all phases, the Ensemble method consistently exhibits the lowest WER, around 10%, highlighting its effectiveness in minimizing errors compared to other classifiers. The graph clearly illustrates the variability in performance across different classifiers and phases, emphasizing the robustness of the Ensemble approach in achieving lower error rates.

7.6.2 Evaluation in Match Error Rate (MER)

Figure 33 illustrates the Match Error Rate (MER) obtained in different phases of an experiment focused on Malayalam speech data. Each phase corresponds to a different feature extraction technique, or a combination of techniques used to model the speech data, and the graph highlights how these different methods impact the MER.

7.6.2.1 Phase I: MFCC

In Phase I, where the Mel Frequency Cepstral Coefficients (MFCC) were utilized to extract features, the MER is 49%. This relatively high error rate suggests that while MFCCs capture essential frequency components of the speech signal, they might not be sufficient on their own to provide a robust representation of the speech data, especially for complex accented speech like Malayalam.

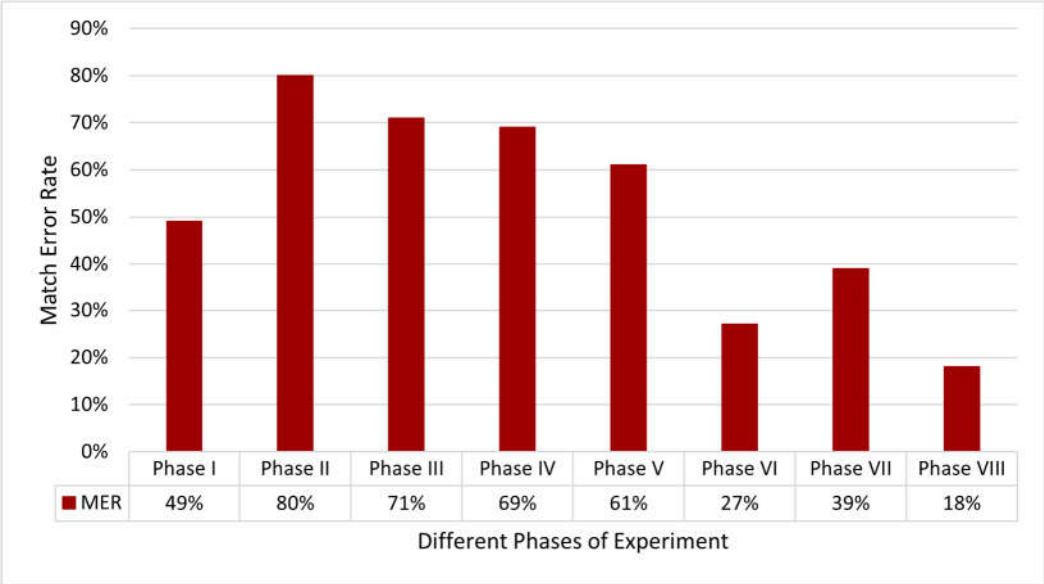


Figure 33 MER of Different Experiments

7.6.2.2 Phase II: STFT

Phase II shows a significant increase in MER to 80% when Short Time Fourier Transform (STFT) features are used. This steep rise indicates that STFT, which provides time-localized frequency information, may not be as effective in isolation. The high error rate suggests that STFT might not capture the necessary features adequately, likely due to the loss of temporal dynamics and finer nuances in speech.

7.6.2.3 Phase III: Tempogram Features

The MER decreases slightly to 71% in Phase III with the use of Tempogram features. These features focus on the rhythmic aspects of the speech signal. The reduction in error compared to Phase II indicates some improvement, suggesting that capturing rhythmic patterns helps in better understanding speech characteristics. However, the error rate remains high, implying that Tempogram features alone are still insufficient.

7.6.2.4 Phase IV: Mel Spectrogram Features

In Phase IV, the MER is 69% when Mel Spectrogram features are extracted. This phase shows a marginal improvement over Tempogram features, highlighting that

Mel Spectrogram, which captures spectral properties and emphasizes perceptually relevant aspects of the signal, provides slightly better feature representation. Despite this, the error rate is still high, indicating a need for more comprehensive feature extraction.

7.6.2.5 Phase V: Combination of MFCC and STFT

The MER decreases to 61% in Phase V, where MFCC and STFT features are combined. This reduction signifies that integrating time-frequency representations with frequency coefficients provides a more detailed and useful feature set for speech recognition. However, the error rate remains substantial, suggesting that additional features might be necessary to capture the full complexity of the speech data.

7.6.2.6 Phase VI: Combination of MFCC and Tempogram

Phase VI exhibits a significant drop in MER to 27% with the combination of MFCC and Tempogram features. This notable improvement indicates that combining frequency coefficients with rhythmic patterns results in a more robust representation of speech, significantly enhancing recognition accuracy. The substantial reduction in error rate emphasizes the importance of including temporal dynamics and rhythm in the feature set.

7.6.2.7 Phase VII: Combination of MFCC and Mel Spectrogram

In Phase VII, the MER is 39% when MFCC and Mel Spectrogram features are combined. This phase shows an improvement over individual features, but an increase compared to Phase VI. The combination of MFCC and Mel Spectrogram captures both frequency coefficients and spectral properties, offering a balanced feature set. However, the increase in MER compared to Phase VI suggests that rhythm-related features are critical for further reducing errors.

7.6.2.8 Phase VIII: Combination of MFCC, STFT, Tempogram, and Mel Spectrogram

Phase VIII demonstrates the best performance with an MER of 18%, the lowest across all phases. This phase combines all previous features-MFCC, STFT, Tempogram, and Mel Spectrogram—resulting in a comprehensive and robust feature set. The substantial reduction in error rate highlights the effectiveness of integrating multiple feature extraction techniques, capturing a wide range of speech characteristics including frequency, time-frequency localization, rhythm, and perceptual relevance. This comprehensive approach significantly enhances the model's ability to accurately recognize Malayalam accented speech.

The analysis of MER across different phases reveals that combining diverse feature extraction techniques is crucial for improving speech recognition accuracy. While individual methods like MFCC, STFT, Tempogram, and Mel Spectrogram provide valuable insights, their combinations, especially the comprehensive integration in Phase VIII, significantly reduce the error rate. This highlights the importance of a multi-faceted approach in capturing the complex and detailed features of speech, particularly in challenging datasets such as Malayalam accented speech.

7.6.3 Evaluation in Accuracy for LSTM-RNN in Phase-8

Figure 34 presents three learning curves showing the accuracy of a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) approach for phase VIII of the experiment. This phase includes a comprehensive feature set combining MFCC, STFT, Tempogram, and Mel Spectrogram features, providing a robust representation of the Malayalam speech data.

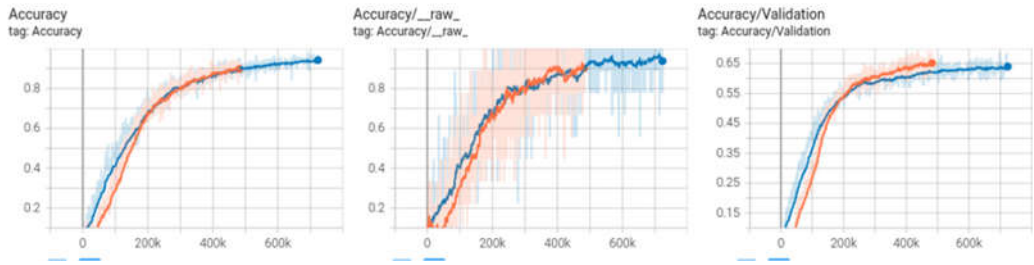


Figure 34 Learning Curves LSTM-RNN Approach (Accuracy Metric)

7.6.3.1 Training Accuracy

The first graph illustrates the training accuracy over the epochs. The accuracy starts low, reflecting the initial stages of model training where the network is learning the basic patterns in the data. As training progresses, the accuracy steadily increases, indicating that the model is effectively learning from the rich feature set. The curve shows a smooth upward trend with minor fluctuations, common in training deep learning models. Around the 400k to 500k iterations mark, the accuracy approaches a plateau, suggesting that the model is nearing its peak performance on the training data. By the end of the training period, the accuracy stabilizes close to the maximum achievable value, indicating that the model has learned the underlying patterns in the training dataset effectively. The final training accuracy approaches approximately 88%.

7.6.3.2 Raw Accuracy

The second graph shows the raw accuracy during training. This curve also starts at a low point and demonstrates a sharp increase as the training proceeds. The raw accuracy represents the direct, unprocessed performance of the model before any post-processing or smoothing is applied. Like the training accuracy curve, the raw accuracy shows significant improvement initially, with the rate of increase slowing as the model learns more complex features. The raw accuracy fluctuates more than the smoothed training accuracy, which is typical due to the inherent noise in raw metrics. By the end of the training period, the raw accuracy stabilizes, indicating consistent performance of the model. The final raw accuracy is around 87%.

7.6.3.3 Validation Accuracy

The third graph represents the validation accuracy, which measures the model's performance on a separate validation dataset not seen during training. The validation accuracy starts lower than the training accuracy, reflecting the model's initial generalization capability. As training progresses, the validation accuracy increases, showing that the model is learning features that generalize well to unseen data. The curve follows a similar upward trend to training accuracy but generally remains lower, as expected. This gap between training and validation accuracy can indicate the level of overfitting. However, in this case, the validation accuracy also plateaus and stabilizes around the same point where the training accuracy does, suggesting that the model generalizes well without significant overfitting. The final validation accuracy reaches approximately 65%.

7.6.3.4 Observations and Inferences

During the early stages, all curves show rapid improvement, reflecting the model's ability to learn basic patterns quickly from the extensive feature set provided in phase VIII. Between 200k to 400k iterations, the learning rate slows, but the accuracy continues to improve steadily. This phase involves the model learning more intricate patterns and nuances in the speech data. Around the 400k to 500k iterations mark, all curves start to plateau, indicating that the model is reaching its optimal performance level. The final accuracy values suggest that the model has effectively learned to recognize Malayalam accented speech with high accuracy. The close alignment of the validation accuracy with the training accuracy towards the end of the training period indicates that the model is not overfitting and has good generalization capabilities. This is crucial for real-world applications where the model will encounter new, unseen data.

The LSTM-RNN model, trained with the comprehensive feature set from phase VIII, demonstrates robust learning and generalization capabilities. The gradual increase and eventual stabilization of both training and validation accuracies highlight the

effectiveness of the combined features in capturing the essential characteristics of Malayalam speech. The consistency between training and validation accuracies towards the end of the training period features the model's potential for practical deployment in speech recognition tasks.

7.6.4 Evaluation in Loss for LSTM-RNN in Phase-8

Figure 35 illustrates the training and validation loss of an LSTM model over a series of iterations. The graph features two main curves: the red curve represents the training loss, and the blue curve represents the validation loss.

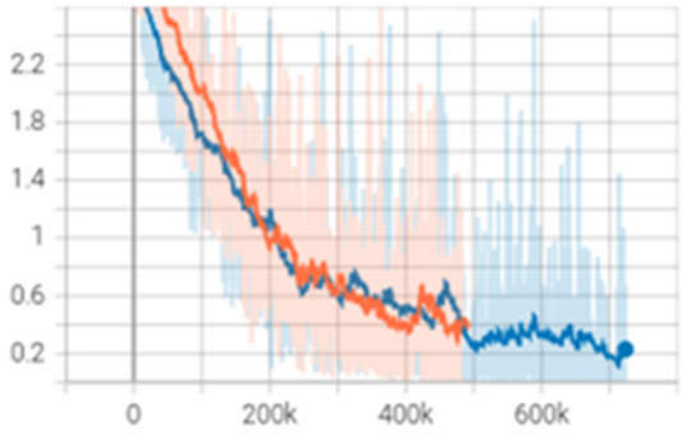


Figure 35 Learning Curves LSTM-RNN Approach (Loss Metric)

Initially, at the start of training, both the training and validation losses are high, around 2.2, indicating that the model starts with poor performance and significant error. As training progresses, there is a notable decrease in both curves, suggesting that the model is learning and improving its performance.

Between 0 and 200,000 iterations, there is a steep decline in both training and validation loss. The training loss steadily decreases, indicating that the model is effectively learning from the training data. By around 200,000 iterations, the training loss has dropped to approximately 0.6. The validation loss follows a similar downward trend, but with more fluctuations, dropping to around 0.8 by the 200,000 iterations mark. This pattern suggests that while the model is improving on unseen

data, it exhibits variability due to the validation set's nature or inherent model performance differences.

As the training continues past 200,000 iterations, the training loss continues its descent more gradually, reaching around 0.3 at 400,000 iterations. In the final stages of training, from 400,000 to 700,000 iterations, both the training and validation losses stabilize at lower values. The training loss plateaus at around 0.2, indicating that the model has nearly converged and learned the training data patterns well. The validation loss, though still fluctuating, also stabilizes around 0.2 to 0.3, demonstrating that the model has achieved a good level of generalization. The shaded regions around the curves, which represent the variance or uncertainty in the loss values, show that while there is some variability, particularly in the validation loss, the model's performance is consistent overall.

The comprehensive analysis of the experiment phases reveals several key inferences about the speech recognition model's performance. The integration of diverse feature sets, particularly in Phase VIII, significantly enhances the model's accuracy and reduces error rates. This phase, which combines MFCC, STFT, Tempogram, and Mel Spectrogram features, results in the lowest Word Error Rate (WER) and Match Error Rate (MER), indicating superior performance compared to earlier phases. The LSTM-RNN model achieves a high training accuracy of approximately 88% and a validation accuracy of 65%, reflecting its robust learning and generalization capabilities. The training and validation loss curves show a consistent decline, stabilizing around 0.2 to 0.3, suggesting effective convergence. These findings emphasize that combining multiple feature extraction techniques provides a richer and more comprehensive representation of the speech data, leading to more accurate and reliable speech recognition. This highlights the critical role of feature diversity in developing effective models for complex tasks like accented speech recognition.

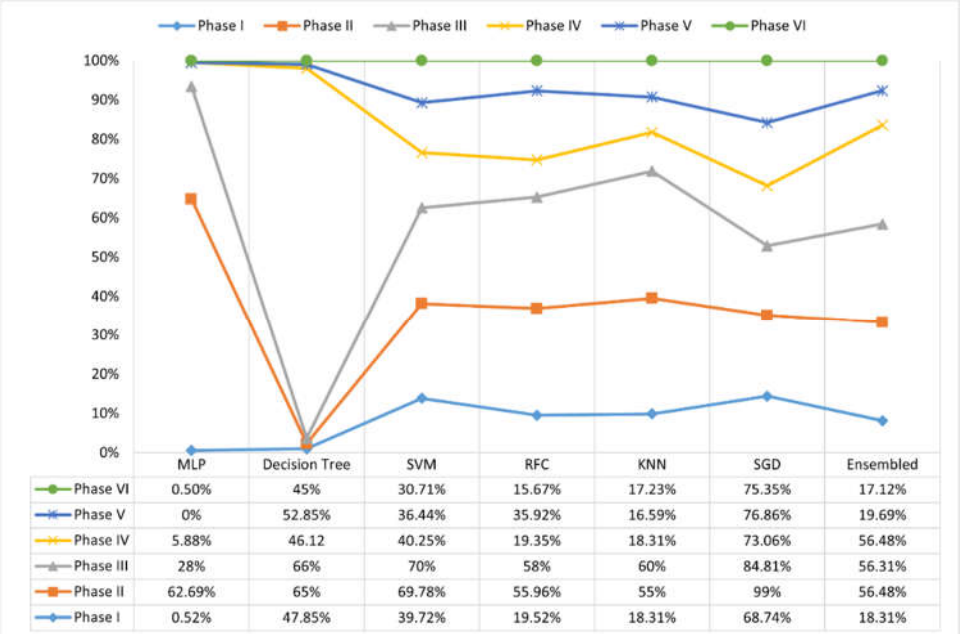
7.7 Performance Evaluation for AASR with AMSC-4

Throughout the experiments, the Word Error Rate (WER) served as a critical metric, with its values varying across different experiments. This variance provided valuable insights into the efficiency and accuracy of the various vectorization procedures, guiding the search for the most effective approach for modeling and recognizing accented Malayalam speech.

7.7.1 Evaluation in WER

7.7.1.1 Phase I

This phase shows that MLP and the ensemble method have the lowest WER, both around 0.52%, while SGD performs relatively better at 68.74%. Decision Tree and RFC models also show low initial performance. In the initial phase, the WERs indicate the baseline performance of each model. The MLP starts with an impressively low WER, suggesting it is well-suited to the initial dataset configuration. In contrast, Decision Tree and SGD models exhibit high WERs, indicating significant room for improvement. The ensemble method shows promise with a moderate WER, benefiting from the combined strengths of multiple models.



	MLP	Decision Tree	SVM	RFC	KNN	SGD	Ensembled
Phase VI	0.50%	45%	30.71%	15.67%	17.23%	75.35%	17.12%
Phase V	0%	52.85%	36.44%	35.92%	16.59%	76.86%	19.69%
Phase IV	5.88%	46.12	40.25%	19.35%	18.31%	73.06%	56.48%
Phase III	28%	66%	70%	58%	60%	84.81%	56.31%
Phase II	62.69%	65%	69.78%	55.96%	55%	99%	56.48%
Phase I	0.52%	47.85%	39.72%	19.52%	18.31%	68.74%	18.31%

Figure 36 WER of Different Phases

Figure 36 illustrates the performance of various machine learning models across different phases of the study, as indicated by the respective Word Error Rates (WER). The models evaluated include Multi-Layer Perceptron (MLP), Decision Tree, Support Vector Machine (SVM), Random Forest Classifier (RFC), K-Nearest Neighbors (KNN), Stochastic Gradient Descent (SGD), and an ensemble method.

7.7.1.2 Phase II

During this phase, the WERs increase notably for all models. This spike suggests that changes in the training process, data augmentation, or feature extraction techniques have initially disrupted model performance. The KNN model is particularly affected, with a dramatic increase to 99% WER, indicating that it is struggling to adapt to the new conditions. The ensemble method's performance also deteriorates, reflecting the challenges of combining predictions from models that are not yet well-tuned.

7.7.1.3 Phase III

In this phase, the MLP model shows significant improvement, reducing its WER to 28%, likely due to initial adjustments in hyperparameters or feature engineering. However, the Decision Tree and SVM see only slight improvements or stabilization, indicating that further tuning is needed. The ensemble method's WER remains steady, suggesting that while individual models are still being optimized, the combined approach is holding its ground.

7.7.1.4 Phase IV

By Phase IV, the MLP model continues to show strong improvement, achieving a WER of 5.88%. This phase likely involves more refined hyperparameter tuning and possibly better feature extraction techniques. The Decision Tree and SVM models also show moderate reductions in WER, while the RFC and KNN models stabilize at lower error rates. The slight increase in the ensemble method's WER could be due to the variability in individual model performances.

7.7.1.5 Phase V

In Phase V, the MLP model achieves a perfect WER of 0%, showcasing its peak performance and the effectiveness of the training process. Other models like SVM and RFC also show significant improvements, reflecting the benefits of advanced tuning. The KNN model also recovers well, with a WER of 16.59%. The ensemble method improves, with a WER of 19.69%, indicating that the combined approach is starting to benefit from the optimized individual models.

7.7.1.6 Phase VI

In the final phase, the MLP model maintains a low WER of 0.50%, indicating consistent performance. The ensemble method achieves its best WER of 17.12%, underscoring the effectiveness of combining multiple models to utilize their strengths and mitigate individual weaknesses. The Decision Tree, SVM, and RFC models show stable and improved WERs, demonstrating the cumulative benefits of iterative tuning and optimization. This phase likely represents the final, optimized state of the models, ready for deployment or further evaluation.

The cumulative insights from these experiments contribute significantly to the understanding and development of accented speech recognition within the Malayalam language, marking an important step forward in the field. The WER of different phases of the experiment is illustrated in Figure 36. Figure 37 represents the performance evaluation of the experiment in terms of accuracy.

7.7.2 Evaluation in Accuracy

Figure 37 presents a comprehensive evaluation of various audio feature extraction methods combined with different classifiers in terms of accuracy. The six subplots compare the performance of different feature sets: MFCC, Tempogram, MFCC+STFT, MFCC + Tempogram, Tempogram, and STFT across various classifiers- MLP, KNN, SVM, Decision Tree, Random Forest, SGD and Ensemble.

7.7.2.1 MFCC for Feature Extraction

Using MFCC, which extracts comprehensive frequency features from the speech signal, the performance of various classifiers was evaluated. The MLP classifier achieved an impressive accuracy of 99.5%, indicating its effectiveness in recognizing patterns within MFCC features. Other classifiers showed the following accuracies: DT at 55%, SVM at 78%, RFC at 79%, KNN at 82%, SGD at 19%, and the ensemble method at 79%. These results demonstrate that MFCC features are highly informative for speech recognition tasks, particularly for MLP and KNN classifiers.

7.7.2.2 STFT for Feature Extraction

The STFT method, which provides a time-localized view of frequency variation, resulted in relatively lower accuracies across all classifiers compared to MFCC. The accuracy results were: MLP at 37%, DT at 32%, SVM at 29%, RFC at 45%, KNN at 45%, SGD at 10%, and the ensemble method at 44%. These results suggest that while STFT captures important spectral content, it may not be as effective as MFCC for the classifiers used in this study, particularly for MLP and SVM.

7.7.2.3 Tempogram for Feature Extraction

Tempogram features, focusing on the rhythmic aspects of speech, achieved the following accuracies: MLP at 72%, DT at 34%, SVM at 31%, RFC at 41%, KNN at 43%, SGD at 21%, and the ensemble method at 42%. These results indicate that Tempogram features provide moderate accuracy improvements, particularly for MLP, but are less effective for other classifiers compared to MFCC features.

7.7.2.4 Combination of MFCC and STFT for Feature Extraction

Combining MFCC and STFT for a comprehensive vector representation yielded higher accuracies: MLP at 95%, DT at 54%, SVM at 60%, RF at 81%, KNN at 81%, SGD at 28%, and the ensemble method at 81%. This combination enhances the analysis by employing both time and frequency aspects, showing significant performance improvements, particularly for RFC and KNN classifiers.

7.7.2.5 Combination of MFCC and Tempogram for Feature Extraction

The fusion of MFCC and Tempogram features resulted in accuracies of: MLP at 95%, DT at 54%, SVM at 60%, RF at 81%, KNN at 81%, SGD at 28%, and the ensemble method at 81%. This combination captures both frequency coefficients and rhythmic patterns, providing a detailed analysis that enhances the understanding of speech characteristics, especially for MLP and RFC classifiers.

7.7.2.6 Combination of MFCC, STFT, and Tempogram for Feature Extraction

Using MFCC, STFT, and Tempogram together for enhanced feature representation produced the following results: MLP at 99%, DT at 52%, SVM at 60%, RF at 80%, KNN at 82%, SGD at 32%, and the ensemble method at 81%. This phase demonstrated that integrating multiple feature extraction techniques can significantly improve classification accuracy, particularly for MLP, SVM, RF, and KNN classifiers.

7.7.2.7 Classifier Performance Analysis

Among the machine learning algorithms evaluated, the MLP consistently provided the best results across different feature extraction techniques. Specifically, MLP achieved the highest accuracy in most cases, such as 99.5% with MFCC, 95% with MFCC+STFT, 95% with MFCC + Tempogram, and 99% with the combination of MFCC, STFT, and Tempogram. These results indicate that MLP, a deep learning method, is particularly effective at capturing complex patterns in the speech data provided by various feature extraction methods.

SVM and RFC also showed good performance but were generally outperformed by MLP. For instance, SVM achieved 78% accuracy with MFCC, and RF achieved 81% with the combination of MFCC and STFT. DT and SGD consistently performed

poorly compared to other classifiers, with DT achieving a maximum accuracy of 55% with MFCC and SGD achieving only 32% in the best case with the combined features.

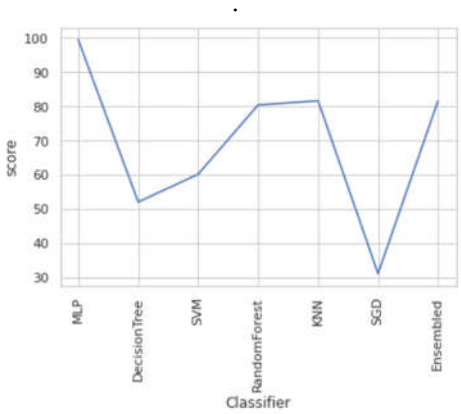
7.7.2.8 Optimal Feature Combinations

The combination of features that provided the best results was the integration of MFCC, STFT, and Tempogram. This combination led to the highest accuracy of 99% with MLP, indicating that combining time-localized, frequency, and rhythmic features captures the most comprehensive information about the speech signals. The combination of MFCC with either STFT or Tempogram alone also yielded high accuracies, such as 95% for MLP with both MFCC+STFT and MFCC + Tempogram.

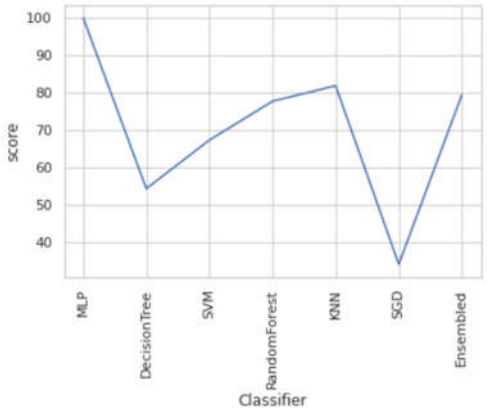
Using MFCC alone also produced high accuracies, especially with MLP achieving 99.5%. The addition of STFT and Tempogram features provided marginal improvements, highlighting the value of integrating multiple types of features to enhance performance.

The superior performance of MLP can be attributed to its ability to learn hierarchical feature representations, which is crucial for capturing the intricate patterns in speech data. While other methods like SVM and RFC showed competitive performance, they were generally less effective than MLP, especially when dealing with complex feature combinations. Figure 37 visualizes the performance of the machine learning classifier in six different phases.

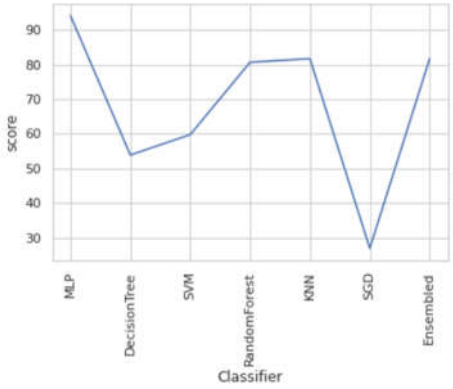
The evaluation highlights that the MLP is the most effective algorithm for accented speech recognition when experimented with AMSC-4 dataset, particularly when used with a combination of MFCC, STFT, and Tempogram features. This combination provides a comprehensive representation of speech data, capturing frequency, time-localized, and rhythmic information. The superior performance of deep learning methods like MLP features their suitability for complex speech recognition tasks, offering significant improvements over traditional machine learning algorithms.



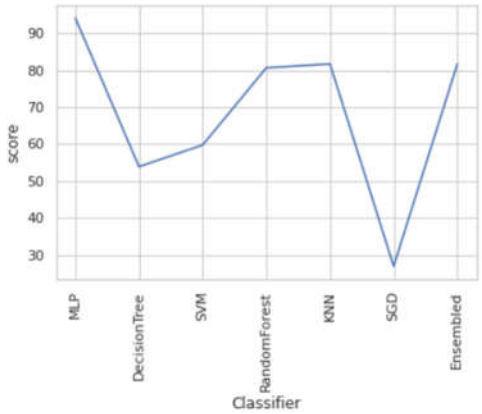
MFCC+STFT+Tempogram



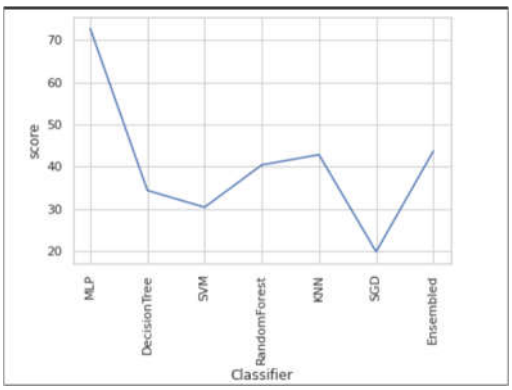
MFCC



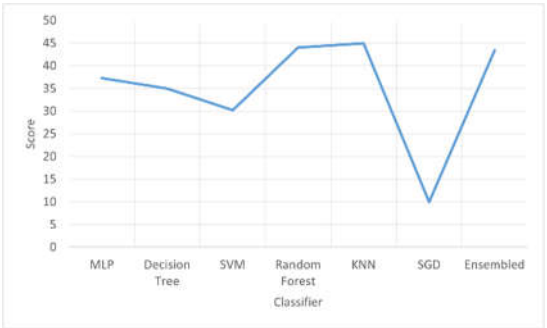
MFCC+STFT



MFCC+ Tempogram



Tempogram



STFT

Figure 37 Performance Evaluation of Experiments in Terms of Accuracy

7.7.3 Integration of MFCC, STFT, Mel Spectrogram, Root Mean Square, and Tempogram Features

This study marked a significant approach from previous methodologies by employing the maximum number of vectors from each utterance to construct the accented speech recognition model. This advanced approach employed the LSTM-RNN method, and the model was trained using a combination of MFCC, STFT, Mel Spectrogram, Root Mean Square, and Tempogram features, which together provide a detailed representation of the speech signals.

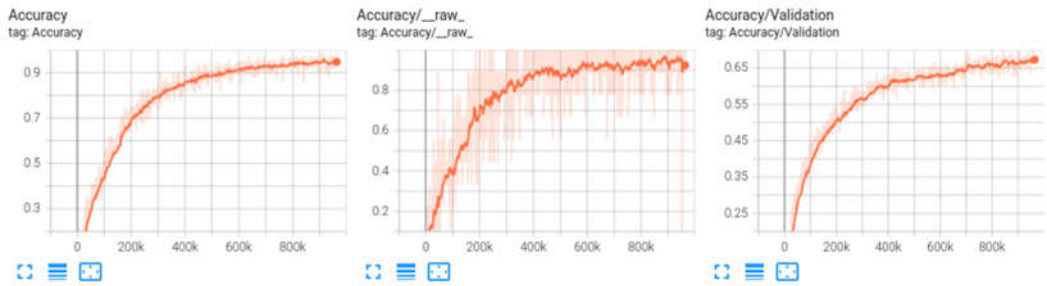


Figure 38 Performance Evaluation (Accuracy) of Phase VII using LSTM RNN

Figure 38 illustrates the learning curves of the AASR model. The model achieved a training accuracy of 95%, indicating that the LSTM-RNN was able to learn the training data exceptionally well. However, the validation accuracy was 67%, which, while lower than the training accuracy, still demonstrates a substantial level of generalization to unseen data.

The use of LSTM-RNN, combined with comprehensive preprocessing and extensive training, demonstrates the potential of deep learning methods to handle the complexities of accented speech. The promising results suggest that with further refinement and optimization, such models can achieve even higher accuracies and lower error rates, making them highly valuable for practical applications in multilingual and accented speech environments.

7.8 Conclusion

Various experiments were executed to develop unified accented models employing machine learning, deep learning, and LSTM-RNN. The assessment of each distinct feature set extraction aimed to build an advanced accented AASR system tailored for the Malayalam language. This innovative approach demonstrated superior performance compared to numerous existing baseline models, particularly in terms of WER, MER, accuracy and loss.

8. Spectral and Influential Features for Unified AASR in Malayalam

8.1 Introduction

This research investigates the extraction and utilization of spectral features, including Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), and Mel Spectrograms, which capture essential information about the frequency domain characteristics of Malayalam speech signals. By integrating these spectral features into a unified framework, which incorporates advanced techniques such as LSTM RNN and Deep Convolutional Neural Networks DCNN, this research aims to enhance the performance and adaptability of AASR systems, thereby addressing the challenges posed by accented speech variations in the Malayalam language.

8.2 Methodology

This study aims to enhance ASR for multi-accented Malayalam speech by employing deep learning techniques, specifically focusing on word-based ASR. Unlike conventional methods using studio-recorded data, this research gathers data through crowdsourcing from various locales to capture authentic, accented speech details.

The research conducts a comparative analysis of two approaches for accent-based ASR in Malayalam. It acknowledges that ASR performance depends on dataset nature and tailored techniques, necessitating rigorous experimentation with Malayalam. The experiment employs AMSC-5 dataset that encompasses 20 distinct sound classes and a corpus of 4000 data points collected via crowdsourcing, emphasizing natural recording environments.

8.2.1 Data Collection and Preprocessing

The methodology adopted in this study follows a similar framework to the previous chapter, with notable distinctions in the feature vectorization process, model architectures, and dataset utilization. Data collection involves crowdsourcing speech samples from diverse locales, emphasizing natural recording environments to capture authentic accents. Preprocessing begins with manual removal of all forms of noise, including background noise, ensuring high-quality speech signals. Subsequently, the speech data is appropriately segmented and sampled at a frequency of 16 kHz, maintaining consistency across the dataset.

8.2.2 Feature Vectorization

Unlike the previous chapter, where a specific feature extraction technique was employed, this study implements a novel approach for feature vectorization. Feature extraction is a crucial step in ASR systems, as it transforms raw speech signals into a format suitable for machine learning models. In this research, feature vectorization involves extracting discriminative features from preprocessed speech signals. MFCCs, which have shown efficacy in capturing speech characteristics, are computed from the preprocessed speech frames. Additionally, delta and delta-delta coefficients are calculated to capture temporal dynamics. These feature vectors, comprising MFCCs and their derivatives, serve as input to the deep learning models for accent-based ASR.

The study also utilizes STFT and Mel spectrogram for feature vectorization. These techniques offer complementary insights into the frequency content and temporal dynamics of the speech signals. STFT is employed to analyze the spectral content of short segments of speech over time, providing a time-frequency representation of the signal. From the STFT representation, Mel spectrograms are computed, which emphasize perceptually relevant frequency bands using a non-linear Mel scale. This process yields spectrotemporal features that capture essential characteristics of the speech signals.

8.2.3 Model Training and Evaluation

Following feature vectorization, the study employs deep learning architectures tailored for accent-based ASR. LSTM-RNN and DCNN are utilized to exploit temporal dependencies and spatial patterns in the speech data, respectively. The models are trained using stochastic gradient descent with backpropagation, optimizing performance metrics such as accuracy and loss. To evaluate model performance, the dataset is split into training, validation, and test sets. Performance metrics, including word error rate (WER) and accuracy, are computed on the test set to assess the effectiveness of the proposed approach.

8.3 Dataset

In this study, a dataset AMSC-5 has been constructed that comprises of utterances in the Malayalam language, capturing the rich dialectal variations that distinguish north Kerala. The spectrogram dataset of AMSC-5 has been constructed for experimenting with DCNN.

Recognizing the importance of diverse representation, the dataset includes contributions from speech donors spanning various age ranges. A strategic emphasis was placed on the age group between 20 to 45, from which most of the data was collected. This decision was informed by a desire to represent clear and quality data, acknowledging that individuals within this age bracket often demonstrate stable and distinct pronunciation patterns.

Table 8 Data Distribution Across Different Districts and Age Groups

District	5 to 12	13 to 19	20 to 45	46 to 65	66 to 85	Total
Kasaragod	150	120	270	150	70	760
Kannur	120	150	270	150	70	760
Kozhikode	200	180	400	200	110	1090
Malappuram	150	130	270	150	60	760
Wayanad	40	80	290	140	80	630
Total	660	660	1500	690	490	4000

This age-focused approach aimed to capture the complexities of accented Malayalam speech while ensuring the consistency and reliability of the dataset. Table 8 provides a comprehensive view of the data distribution, reflecting the research's commitment to capturing the complex nature of the Malayalam language's dialectical variations. Table 9 presents a selection of 20 isolated words from the Malayalam language that have been chosen as sample classes for the construction of the dataset. Table 9 illustrates the following details:

1. Uttered Word (Malayalam script): This column provides the Malayalam script for each selected word. The words represent various everyday concepts and themes, contributing to a balanced dataset.

2. IPA Transcription: The corresponding IPA transcription for each word is given in this column. This standardized notation allows for a precise understanding of the pronunciation and phonetic attributes of the words.

Table 9 Example Classes

Uttered word	IPA	Uttered word	IPA
വിദ്യാർത്ഥി	ʋidʲjaːrʈʰi	യാത്ര	jaːʈra
ചോദ്യം	ʧoːɖjam	വിജയം	ʋijajam
മൂല്യനിർണയം	muːljanirɳajam	ഉപകരണങ്ങൾ	upakaraɳaŋŋaɭ
വിദ്യാർത്ഥി	ʋidʲjaːrʈʰi	അനുസരണ	anusaraɳa
സുഹൃത്ത്	suɦriʈʈə	ലക്ഷണം	lakʂaɳam
സഹായം	saɦaːjam	വിശ്വാസം	ʋiʃʋaːsam
പരിപാടി	paɾipaːʈi	വിഭാഗം	ʋibʱaːgam
പ്രസംഗം	prasamgam	ലഭ്യത	labʱjaʈa
മത്സരം	maʈsaram	ഇളവ്	iɭavə
തോൽവി	ʈoːlʋi	മഴ	maɻa

Examples from the table include:

1. "വിദ്യാർത്ഥി" (ʋidʲaːrt̪ʰi) meaning 'student.'
2. "യാത്ര" (jaːt̪ra) meaning 'journey.'
3. "സുഹൃത്ത്" (suɦrɨt̪t̪ə) meaning 'friend.'
4. "മഴ" (maɻa) meaning 'rain.'

The selection of these particular words ensures a wide range of sounds and phonetic characteristics, essential for robust ASR system training. The use of IPA transcription ensures that the phonetic attributes of each word are captured accurately, providing a valuable resource for understanding and analyzing the diverse accents and dialects present in the Malayalam language.

8.4 Feature Extraction

Feature extraction involves identifying and utilizing significant features that accurately represent the speech data, while ignoring the redundant or irrelevant ones. This phase of experiment extracts the feature vectors for experimenting with LSTM-RNN. The main steps in feature extraction include:

8.4.1 MFCC

MFCC is a widely used feature extraction technique in speech processing. For this experiment, MFCC was employed to extract 40 prominent features from each speech signal, representing the phonetic content. These features provide a compact and expressive representation, capturing variations and patterns in different Malayalam accents. This reduces noise and focuses on significant components affecting speech perception, forming a robust foundation for building and training the LSTM-RNN model. Figure 39 illustrates the 40 extracted MFCC features. The 40 features can be computed by:

$$MFCC_k = \sum_{m=1}^{M} log\big(M(m)\big)\, cos\left[\frac{\pi k(m-0.5)}{M}\right] \text{ Where k=1, 2, 3,….13} \quad (27)$$

Compute the first and second derivatives of the 13 MFCC coefficients:

$$\Delta MFCC_k = \frac{MFCC_{k+1} - MFCC_{k-1}}{2} \quad (28)$$

$$\Delta^2{}_{MFCC_k} = \frac{\Delta MFCC_{k+1} - \Delta MFCC_{k-1}}{2} \quad (29)$$

Compute the mean of the 13 MFCC coefficients:

$$\mu MFCC = \frac{1}{13}\sum_{k=1}^{13} MFCC_k \quad (30)$$

Concatenate the original 13 MFCC coefficients, their first and second derivatives, and the mean to form a 40-dimensional feature vector [15]:

$MFCC_{features}$=$MFCC_1$,$MFCC_2$,…,$MFCC_{13}$,$\Delta MFCC_1$,…,$\Delta MFCC_{13}$,$\Delta^2 MFCC_1$,…,$\Delta^2 MFCC_{13}$, μMFCC]

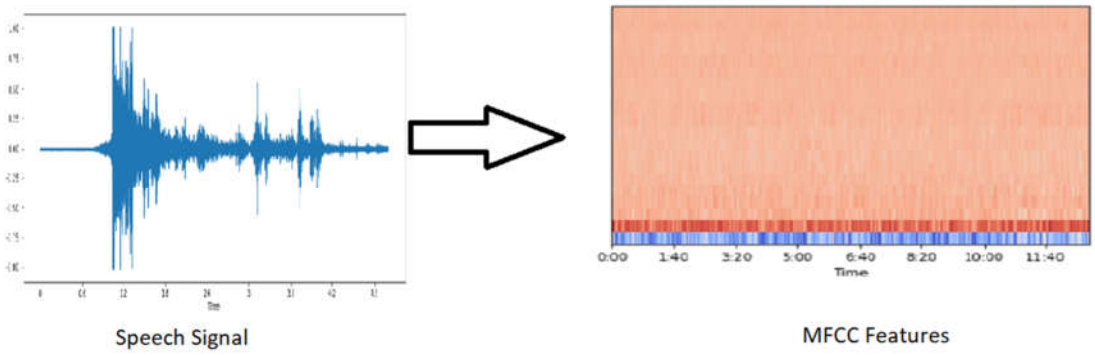


Figure 39 Forty MFCC Features

8.4.2 STFT

STFT analyzes the frequency content of speech signals within short, overlapping time frames, offering a time-frequency representation. Following the extraction of 40 MFCC features, STFT was used to derive 12 additional features representing the signal's amplitude and frequency behavior over time. These features are essential for understanding the non-stationary nature of speech and its spectral characteristics,

particularly useful for distinguishing accents. Figure 40 depicts the STFT features used in the study.

$$STFT\{x(t)\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \text{ [15]} \quad (31)$$

where $w[n]$ is the window function, typically a Hamming or Hann window, $x[n]$ is the signal, m is the time index, and ω is the frequency index.

After computing the STFT, the spectrogram can be computed as:

$$S(t,f) = |\ STFT\{x(t)\}(m,\omega)\ |^2 \text{[15]} \quad (32)$$

This gives a time-frequency representation of the signal, where each element represents the amplitude of the signal at a specific time and frequency.

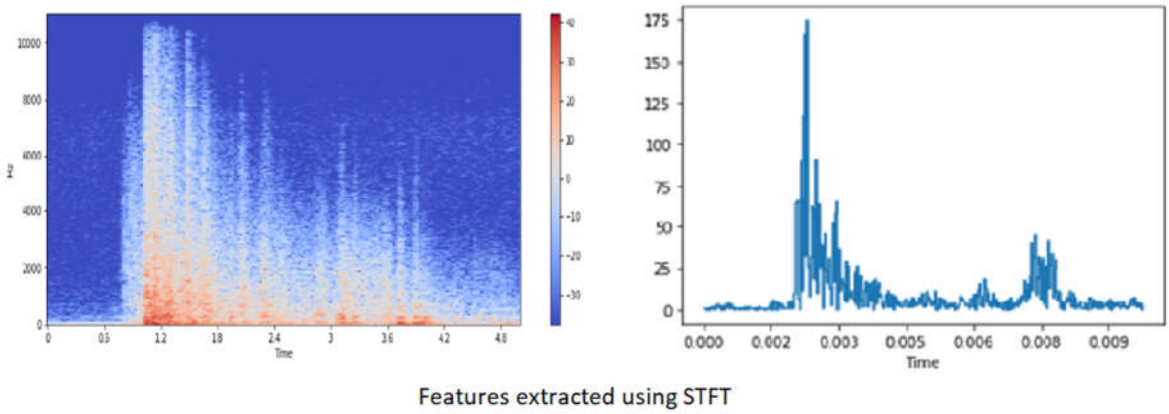


Figure 40 The 12 Features Extracted from the Speech Signal Using STFT

From the spectrogram $S\ (t,\ f)$, the following 12 amplitude-related features can be extracted:

1. $Mean = \frac{1}{N}\sum_{i=1}^{N} |S(t,fi)|$
2. $Variance = \sum_{i=1}^{N} (|\ S(t,fi)\ | - Mean)^2$
3. $Skewness = \frac{1}{N}\sum_{i=1}^{N} \left(\frac{|S(t,fi)|-Mean}{StdDev}\right)^3$,

   where *StdDev* is the standard deviation of the amplitude values.
4. Kurtosis=$\frac{1}{N}\sum_{i=1}^{N} \left(\frac{|S(t,fi)|-Mean}{StdDev}\right)^4 - 3$

5. Energy=$\sum_{i=1}^{N} |S(t,fi|^2$
6. Entropy=$-\sum_{i=1}^{N} |S(t,fi|^2 log(|S(t,fi|^2)$
7. Max=max (|S (t, fi) |)
8. Min=min (|S (t, fi) |)
9. Range=Max−Min
10. StdDev=$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(|S(t,fi)| - mean^2)}$

8.4.3 Mel Spectrogram

After extracting 52 features using MFCC and STFT, an additional 128 features were extracted using the Mel Spectrogram method. This method aligns closely with human hearing, as it represents the speech signal on a Mel scale, reflecting the logarithmic sensitivity of the human ear to low frequencies. The total of 180 features (52 from MFCC and STFT, and 128 from Mel Spectrogram) provides a comprehensive representation of the speech signal. The Mel spectrogram's visual representation on a Mel scale facilitates deep learning algorithm processing, mirroring human auditory perception. Figure 41 shows the Mel spectrogram used in the experiment.

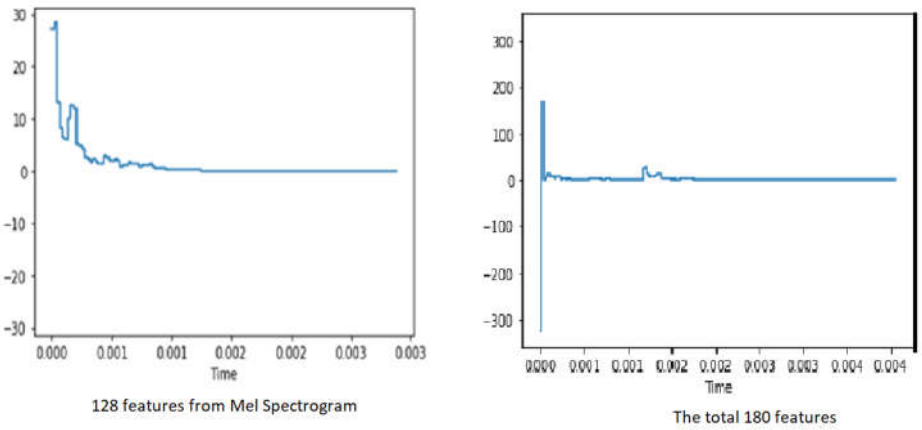


Figure 41 Mel Spectrogram Features and the Total 180 Speech Signal Features

The 128 vectors can be computed from Mel Spectrogram as follows:

Compute the Spectrogram:

$$S(t,f) = |\ STFT\{x(t)\}(m,\omega)\ |^2 \text{ [15]} \quad (33)$$

Apply Mel Filter Banks:

$$M(m) = \sum_{k=1}^{K} |S(fk)|^2 H_m\,(f_k) \text{ [15]} \quad (34)$$

Take the Logarithm:

$$MelSpec(m) = \log(\mathrm{M(m)}) \quad (35)$$

The final feature vector is a concatenation of the features obtained from MFCC, STFT, and Mel Spectrogram:

8.5 AASR Model Construction

This study compares the performance of a LSTM-RNN trained with a comprehensive set of speech features against a DCNN trained with spectrogram images. The objective is to determine the most effective approach for accurately recognizing accented speech variations in Malayalam.

The LSTM-RNN architecture is particularly well-suited for capturing long-range dependencies in sequential data, making it ideal for modeling the temporal dynamics of speech. By utilizing the combined set of 180 features, the LSTM-RNN learns to process and extract relevant patterns from the input sequences, thereby enhancing its ability to recognize accented speech variations.

In parallel, a DCNN model is trained using spectrogram images generated from the speech signals. Spectrograms provide a visual representation of the frequency content of speech over time, with the DCNN treating these images as two-dimensional data. The DCNN architecture excels at capturing spatial hierarchies and patterns within the spectrograms, enabling it to learn high-level features that are crucial for distinguishing between different accents in Malayalam speech. By

analyzing the visual patterns in the spectrogram images, the DCNN can effectively identify complex spectral variations and temporal dependencies relevant to accent recognition.

This study involves training both the LSTM-RNN and DCNN models independently and evaluating their performance on a common test set of accented Malayalam speech data. By conducting this comparative analysis, the study aims to identify the strengths and limitations of each approach, providing insights into the most suitable techniques for improving accented speech recognition in the Malayalam language. The integration of these methodologies in a comparative framework allows for a comprehensive evaluation of different feature representations and model architectures, ultimately contributing to the development of more robust and accurate AASR systems for Malayalam speech.

8.5.1 AASR using LSTM

For training the LSTM-RNN in the AASR system, a total of 180 values are utilized, comprising the combined feature sets of 40 MFCCs, 12 amplitude values from the Short-Time Fourier Transform STFT, and 128 Mel Spectrogram values. These features collectively provide a comprehensive representation of the spectral and temporal characteristics inherent in Malayalam speech.

The MFCCs capture detailed information about the spectral envelope of the speech signal, while the STFT amplitude values offer insights into the frequency content at each time frame. Additionally, the Mel Spectrogram values provide a perceptually relevant representation of the frequency spectrum over time, further enriching the feature space.

By incorporating these 180 values into the LSTM-RNN architecture, the model can effectively learn temporal dependencies and sequential patterns present in accented speech variations. The LSTM-RNN architecture is well-suited for capturing long-range dependencies in sequential data, making it particularly suitable for speech

recognition tasks where contextual information over time is crucial for accurate recognition.

During training, the LSTM-RNN learns to process and extract relevant features from the input sequences of 180 values, utilizing the temporal dynamics and contextual information encoded in the feature representations. Through iterative training and optimization, the model adjusts its parameters to minimize prediction errors and improve its ability to recognize accented speech variations in the Malayalam language. The integration of these 180 values into the LSTM-RNN architecture enhances the system's capability to effectively capture and utilize both spectral and temporal features for accurate and robust accented speech recognition in Malayalam.

An LSTM network is characterized by a specialized cell structure interlaced with three crucial gates: the input gate, the forget gate, and the output gate. This unique composition enables the cell to retain information over an arbitrary duration, precisely regulating the flow of data throughout the process and are discussed below:

1. Input Gate: Determines what information is essential and stores it in the cell state.
2. Forget Gate: Evaluates and discards unnecessary information from the cell's memory.
3. Output Gate: Filters and transmits the required information from the cell state to the subsequent layers.

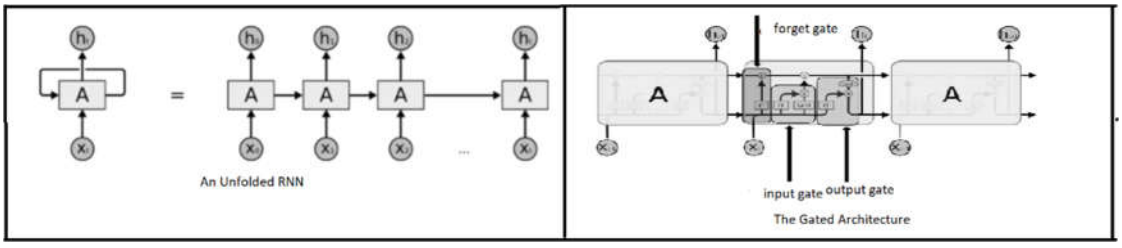


Figure 42 An Unfolded RNN and the Gated Architecture

The LSTM cell accepts an input X_0 and delivers an output h_0, which, combined with the next input X_1, forms the input for the succeeding step. This continuous linkage between consecutive steps ensures that the LSTM retains a memory of the sequence, constructing an interconnected pathway throughout the model.

Figure 42 illustrates this unfolded RNN structure and the intricate gated architecture, reflecting the sophisticated flow of information within the LSTM. The activation functions within the gates, specifically the sigmoid function, control the passage of values, while the hyperbolic tangent (tanh) function assigns weights based on the relevance of the data.

8.5.1 AASR using DCNN

The second approach employed in this experiment is by utilizing a Deep Convolutional Neural Network (DCNN) for the recognition of accented speech in the Malayalam language. The methodology consists of the following distinct stages:

8.5.1.1 Spectrogram Dataset Construction

Spectrograms corresponding to the AMSC-5 dataset are constructed to train the DCNN model. These spectrograms provide a visual representation of the frequency content of speech over time. The DCNN treats these spectrogram images as two-dimensional data, excelling at capturing spatial hierarchies and patterns within the spectrograms. This enables DCNN to learn high-level features that are crucial for distinguishing between different accents in Malayalam speech. By analyzing the visual patterns in the spectrogram images, the DCNN can effectively identify complex spectral variations and temporal dependencies relevant to accent recognition.

8.5.1.2 The Architecture and Working

In the DCNN approach, spectrograms of AMSC-5 have been constructed and used. This transformation enables the application of image processing methods for speech recognition.

The model for this experiment is initialized as a Sequential model, which allows layers to be added one after another in a linear stack. This simplifies the process of building and managing the neural network. To ensure uniformity in the input data, all input spectrograms are resized to a standard shape of (224, 224, 3). This standardization is crucial for efficient processing by neural network.

The resized input is then fed into the first convolutional layer, which begins the process of feature extraction by applying a set of filters to detect basic patterns and structures within the input data. The output from this initial layer is subsequently passed through a second convolutional layer, resulting in feature maps with dimensions (222, 222, 32). This secondary layer refines the features extracted by the first layer, enhancing the network's ability to recognize more complex patterns.

Following this, a max pooling layer with a pool size of (2, 2) is applied, reducing the spatial dimensions of the feature maps to (111, 111, 32). This downsampling process helps to reduce computational complexity while preserving the most critical information from the feature maps. The pooled features are then passed through a third convolutional layer, which produces further refined feature maps with dimensions (109, 109, 64).

To continue the process of dimensionality reduction, a second max pooling layer with a pool size of (2, 2) is applied, resulting in feature maps with dimensions (54, 54, 64). At this stage, a dropout layer is introduced with a dropout rate (e.g., 0.25) to prevent overfitting. This layer randomly sets a fraction of the input units to zero during training, promoting model generalization.

The data is then processed through a fourth convolutional layer, producing feature maps with dimensions (52, 52, 64). This is followed by a third max pooling layer with a pool size of (2, 2), reducing the spatial dimensions to (26, 26, 64). Another dropout layer is applied here to further mitigate the risk of overfitting.

The fifth convolutional layer comes next, generating feature maps with dimensions (24, 24, 128). This is followed by a fourth max pooling layer with a pool size of (2, 2), which further reduces the spatial dimensions to (12, 12, 128). This sequence of convolutional and pooling layers ensures that the feature maps are thoroughly refined and condensed.

After the convolutional and pooling stages, the multi-dimensional feature maps are flattened into a one-dimensional array of size 18432. This flattened array serves as the input to the fully connected layers. The first dense layer contains 64 neurons, which interpret the extracted features and learn higher-level representations of the input data. To add regularity and further prevent overfitting, another dropout layer with a dropout rate (e.g., 0.5) is applied.

Finally, the output dense layer, consisting of 20 neurons, corresponds to the 20 different classes in the dataset. This layer uses a softmax activation function to predict the probability distribution over the classes, allowing the model to determine the class of the input based on the learned features. This structured approach ensures that the DCNN effectively learns to recognize and classify the diverse spectrogram inputs, capturing the intricate patterns inherent in the multi-accented Malayalam speech data.

This detailed structure illustrates the intricate design of the Deep Convolutional Neural Network used in this study. The combination of convolutional, pooling, dropout, and dense layers enables the model to efficiently learn and recognize accented speech patterns in the Malayalam language. By making use of the visual representation of the speech signals, this methodology offers a sophisticated approach for accented speech recognition, presenting promising opportunities for future research and application.

8.6 Performance Evaluation

Figures 43 and 44 provide a visualization of the overall accuracy, loss, and validation during the construction of the LSTM-RNN model, mapped against computational steps. At the beginning of training, the accuracy is relatively low, indicating that the model initially struggles to make accurate predictions.

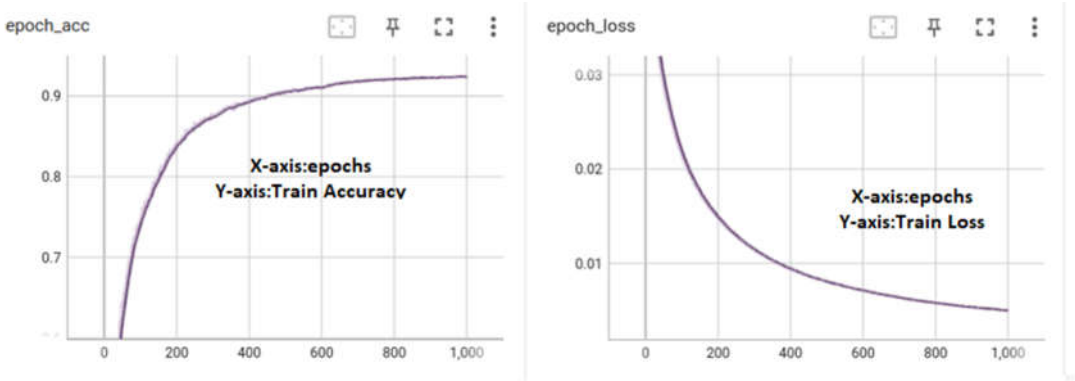


Figure 43 The Performance Evaluation: Training Phase

As training progresses, the curve rises steeply, reflecting rapid improvement in the model's ability to learn and recognize patterns in the data. This steep increase is particularly notable in the initial epochs, where the most significant learning occurs. As the epochs continue, the curve begins to level off, approaching a plateau. This plateau indicates that the model is nearing its optimal performance, as it has effectively learned the key patterns and features from the training data. The faded line in the graph represents the original classification, while the darker line is obtained with a smoothing of 0.5. The model achieved a validation accuracy of 67 percent and train accuracy of 95 percent over 1000 epochs. Figure 44 visualizes the performance in terms of loss.

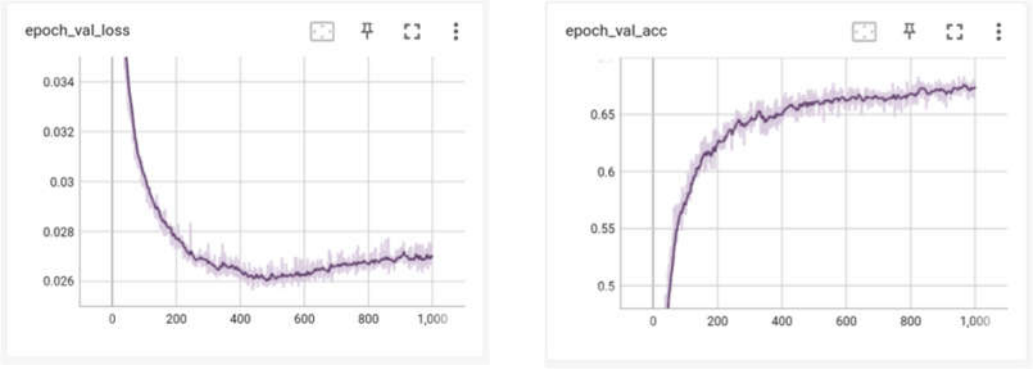


Figure 44 The Performance Evaluation: Validation Phase

Initially, the loss value is high, indicating significant prediction errors. As training begins, the loss curve shows a steep decline, demonstrating that the model is quickly learning to reduce these errors. This rapid decrease in loss during the early epochs is a positive sign, indicating effective error correction and adjustment of model parameters. As training progresses, the rate of decline slows, and the curve gradually flattens. This flattening suggests that the model is converging, with diminishing improvements in reducing the training loss as it nears its optimal state. Initially, the loss was 0.03, which gradually reduced to 0.003 by the end of the training process at epoch 1000. Initially, the loss was 0.034 and gradually reduced to 0.027 by the end of the validation process at epoch 1000.

These performance metrics demonstrate that the LSTM-RNN model is well-trained and capable of making accurate predictions based on the training data. The high training accuracy and low training loss achieved by the end of the training phase indicate that the model has effectively learned to generalize from the training data, setting a solid foundation for evaluating its performance on unseen validation and testing datasets. This thorough training process is crucial for ensuring that the model can accurately recognize and classify multi-accented Malayalam speech in real-world applications.

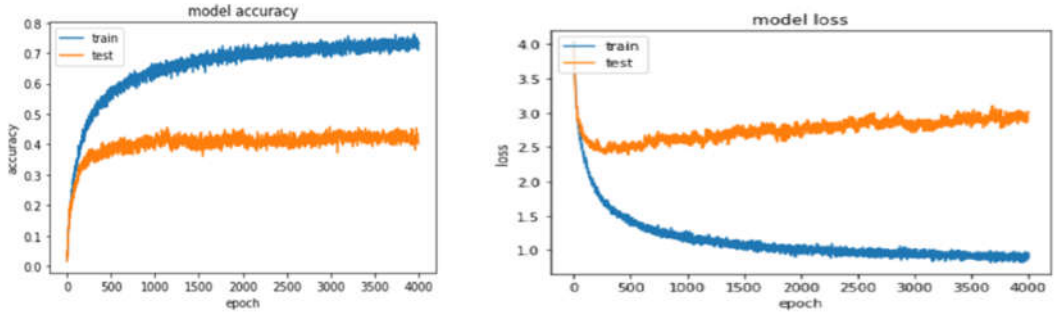


Figure 45 Training and Testing Accuracy, Loss vs Epochs

Figure 45 showcases the performance of a Deep Convolutional Neural Network (DCNN) model over the course of 4000 epochs. The model training, which took approximately 12 hours, utilized 4000 spectrograms, with a random split of 3020 samples for training and 800 samples for testing. The training accuracy steadily increases, reaching approximately 74 percent by the end of the training period. But the testing accuracy starts lower and shows a much slower rate of improvement, plateauing around 39 percent.

The loss plot on the right illustrates the model's loss for both the training and testing datasets. The training loss, also in blue, starts high and rapidly decreases, stabilizing around 9 percent. The testing loss, shown in orange, decreases initially but remains significantly higher than the training loss, leveling off at approximately 17 percent.

These plots indicate that while the model performs well on the training data, achieving high accuracy and low loss, it struggles with the testing data, as shown by the substantial gap between the training and testing curves in both accuracy and loss. This discrepancy suggests potential issues with overfitting, where the model has learned the training data well but does not generalize effectively to unseen data.

8.7 Conclusion

In this chapter, the exhaustive experimentation of two methodologies, LSTM-RNN and CNN, was undertaken for the task of accented speech recognition in the Malayalam language. Through a detailed analysis involving multiple feature

extraction techniques and model architectures, insights were obtained that contribute to the understanding of the behavior and performance of these models in handling multi-accented speech data.

The LSTM-RNN model emerged as the more effective approach, displaying a high train accuracy of 95% and validation accuracy of 67%, with a minimal validation loss of 0.027%. In contrast, the CNN model, although innovative in its application, could only attain 74% of train accuracy and 39% of test accuracy, with a more considerable validation loss.

The superiority of LSTM-RNN over DCNN in this specific context underlines the importance of selecting the right model and feature extraction techniques that align with the complexities of the Malayalam language and its accented variations. The findings of this study not only reinforce the adaptability and efficiency of LSTM-RNN in processing sequential speech data but also open avenues for future research to explore further enhancements and optimizations.

9. A Feature-Based Investigation of LSTM-RNN and ML Approaches

9.1 Introduction

This chapter presents a comprehensive investigation into feature-based approaches for accented speech recognition, focusing specifically on the Malayalam language. The core objective of this research is to develop a robust ASR system capable of accurately transcribing accented Malayalam speech, addressing the linguistic diversity prevalent in the region.

A key aspect of this research is the innovative feature engineering approach, which utilizes a layered architecture to extract and represent critical features of accented speech signals. Each layer of the feature extraction process contributes unique insights into the characteristics of the speech signal, culminating in a comprehensive set of features that capture the nuances of accented Malayalam speech.

The chapter proceeds to detail the experimentation phase, where various machine learning models MLP, RFC, DTC, KNN, SVM, SGD and ensemble classifiers are evaluated for their efficacy in recognizing accented speech patterns. Additionally, the construction of an accent model using LSTM-RNN architecture is explored, offering a novel approach to modeling the complexities of accented speech.

A thorough analysis and comparison of results obtained from different models are presented, highlighting the strengths and limitations of each approach. This chapter contributes to the advancement of accented speech recognition technology, offering valuable insights into feature-based approaches tailored to the unique linguistic characteristics of the Malayalam language. Through experimentation and analysis, this research aims to enhance our understanding of accented speech processing and pave the way for more accurate and reliable ASR systems in diverse linguistic contexts.

9.2 Methodology

The AMSC-3 dataset comprises 7070 samples spanning multiple age groups and genders, ensuring a diverse representation of local Malayalam accents is used for conducting this experiment. The data collection strategy employs crowdsourcing techniques to capture a wide array of speech variations. Following data collection, the feature engineering process is vital in transforming raw speech signals into meaningful representations. This involves the simultaneous extraction of multiple feature vectors using a layered approach, which encapsulates various spectral, temporal, and rhythmic characteristics of the speech signal. Specifically, features such as Mel-Frequency Cepstral Coefficients (MFCC), Short-Time Fourier Transform (STFT), Mel Spectrogram, Spectral Roll-Off, Root Mean Square (RMS), and Tempogram rhythmic features are extracted, resulting in a high-dimensional feature vector for each speech sample.

The extracted features are then used to construct and evaluate various machine learning models to assess their efficacy in recognizing accented Malayalam speech. The models include MLP, RFC, DTC, KNN, SVM, and SGD classifiers. Additionally, an ensemble approach is adopted to utilize the strengths of these individual models, enhancing the overall prediction accuracy. Additionally, the methodology includes the development of an accent model using LSTM-RNN. This approach is well-suited for capturing the sequential patterns inherent in speech data, allowing for the modeling of long-term dependencies as seen in the previous chapters. These capabilities are especially valuable for effectively addressing the complex variations present in accented speech.

The final stage involves a comprehensive analysis and comparison of the results obtained from different models. This comparative evaluation helps identify the most effective techniques for handling accented data, providing insights into the relative strengths and weaknesses of traditional machine learning models versus advanced neural network architectures. This methodology integrates rigorous data collection,

sophisticated feature extraction, and a diverse set of Machine Learning techniques, aiming to establish a robust ASR system for Malayalam, capable of accurately recognizing and interpreting accented speech. Figure 46 illustrates the workflow of the proposed study.

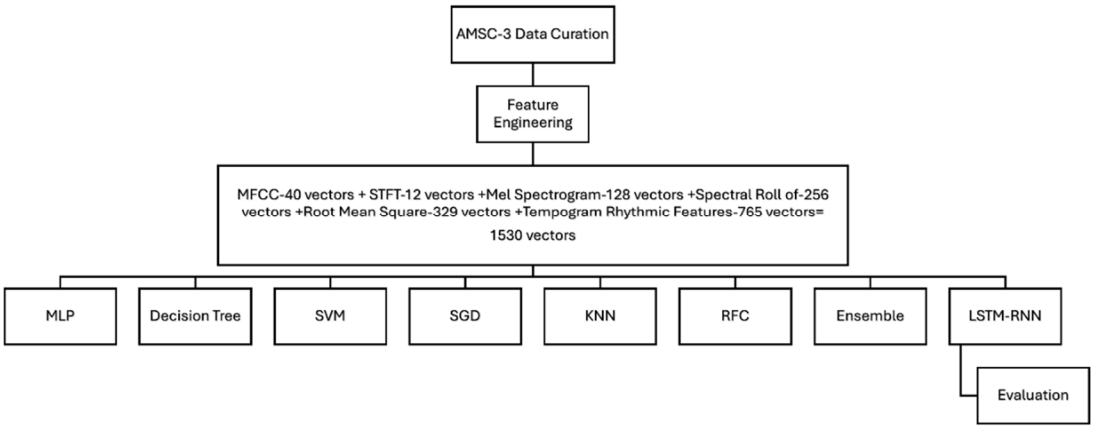


Figure 46 Workflow of the Proposed Study

9.3 Dataset Preparation

AMSC-3 is utilized for conducting this study. To ensure a more comprehensive representation, multiple utterances of the same signal are recorded in varying environments. Following careful labeling and annotation, the collected speech samples are prepared for dataset creation. They are converted and saved into the .wav format, each sampled at a frequency of 16000 Hz.

9.4 Feature Engineering

The speech signal is input to the six layers for the feature extraction process simultaneously. The first layer, MFCC, is used for extracting the spectral frequencies, resulting in 40 coefficients that correspond to the prominent 40 frequency values of the audio waves. The STFT method is then used to extract the amplitude values that correspond to the different speech frames of the signal, with 12 prominent values extracted for the experiment from this layer. The next layer in the feature engineering process employs the Mel Spectrogram feature extraction technique, which returns

the low-frequency spectral values closely corresponding to the speech vectors sensitive to human ears, extracting 128 spectral features from this layer.

A certain percentage of the total spectrum energy, such as 85%, resides below a specific frequency known as the spectral roll-off frequency. The next layer incorporates this technique and extracts 256 speech features from the accented speech. Following this, the root mean square value of each frame is computed, resulting in the extraction of 329 features from the signal in the subsequent layer. The speech signal is then analyzed to extract the Tempogram and rhythmic features, which correspond to variations in pitch in accented signals, with 765 features extracted in this layer. Consequently, every audio signal is vectorized into a set of 1530 features through the feature engineering process.

The innovation in this experiment lies in the specific feature engineering technique that has been employed. In this research, a layered approach has been introduced to the feature extraction process, focusing on the extraction of crucial aspects such as accent, age, gender, tempo, and other significant characteristics. In this method, each layer's output, consisting of a set of features extracted, is concatenated with the next layer's output. By the final layer, the complete output integrates a comprehensive set of 1530 features from each audio recording, that shall contribute to constructing the ASR system.

9.5 Accented ASR Model

9.5.1 Accented ASR Construction using Machine Learning Techniques: Performance and Evaluation

In constructing an accented Automatic Speech Recognition (ASR) system for the Malayalam language, a comprehensive series of experiments was conducted using various machine learning approaches to determine the most effective method for handling accented speech. Each experiment utilized a distinct model, employing a carefully engineered feature set to evaluate and optimize performance. Here's a

detailed account of the experiments, the outcomes, and the contributions of feature engineering to each model's performance.:

9.5.1.1 Multi-layer Perceptron (MLP)

The first experiment utilized an MLP, a type of neural network particularly well-suited for classification tasks when the dataset is manageable in size. This model achieved a remarkable performance accuracy of 100 percent. Key parameters included a hidden layer size set to 3000 neurons and a maximum of 10000 iterations, allowing the MLP to fully capture the intricate patterns within the accented speech data. The MLP's ability to learn complex representations of the input features through its deep architecture was crucial in reaching such high accuracy.

9.5.1.2 Decision Trees

The second experiment employed Decision Trees, which are known for their interpretability and simplicity. This classifier organizes data into a tree-like structure, where internal nodes represent feature tests, branches represent outcomes of those tests, and leaf nodes represent class labels. Despite the straightforward approach, the Decision Tree model achieved an accuracy of 54.40 percent. This result highlights the model's ability to identify key decision points in the feature space, though it may struggle with more complex patterns compared to neural networks.

9.5.1.3 Support Vector Machines (SVM)

In the third experiment, SVM were used to construct the accented model. SVMs are powerful classifiers that work well in high-dimensional spaces and are effective in cases where the number of dimensions exceeds the number of samples. The SVM model achieved an accuracy of 67.36 percent, utilizing the high-dimensional feature vectors to find optimal hyperplanes that separate different speech classes effectively.

9.5.1.4 Random Forest Classifier (RFC)

The fourth experiment implemented the Random Forest Classifier (RFC), which initially achieved an accuracy of 22.28 percent. However, through hyperparameter

tuning using techniques such as GridSearchCV, which systematically tests combinations of parameters to find the best settings, the model's performance significantly improved to 77.72 percent. RFC's ensemble approach, which aggregates the predictions of multiple decision trees, contributed to its enhanced performance after optimization.

9.5.1.5 K-Nearest Neighbors (KNN)

The fifth experiment involved the K-Nearest Neighbors (KNN) algorithm, which classifies samples based on the majority vote of their nearest neighbors in the feature space. The initial model resulted in an accuracy of 63.73 percent. By applying hyperparameter tuning, the performance was optimized to achieve an accuracy of 81.86 percent, demonstrating the algorithm's dependence on properly chosen parameters like the number of neighbors (k) and distance metrics.

9.5.1.6 Stochastic Gradient Descent (SGD)

The sixth experiment used the Stochastic Gradient Descent (SGD) classifier, a linear model that optimizes the loss function iteratively on small batches of data, making it suitable for large-scale learning. However, the SGD model achieved a modest accuracy of 34.19 percent, with no significant improvements observed even after extensive hyperparameter tuning. This outcome indicates potential limitations in the linear approach of SGD for capturing the nuances of accented speech.

9.5.1.7 Ensemble Model

Following the individual experiments, a hybrid model was constructed to ensemble the already developed models. This ensemble approach utilized the majority voting technique, where predictions from each model were combined to make a final decision. The ensemble model benefited from the strengths of each individual model, compensating for their weaknesses, and achieved a commendable accuracy of 79.44 percent. This approach emphasized the effectiveness of hybrid models in enhancing overall prediction accuracy by leveraging diverse model capabilities.

This comprehensive evaluation provides valuable insights into the strengths and limitations of different machine learning approaches in the context of accented ASR systems. The analysis of the outcomes of the experiment shows how each machine learning method performed in this comprehensive experiment, illustrating the comprehensive nature of machine learning in the challenging task of modeling accented ASR.

9.5.2 Accented ASR Construction using Deep Learning Techniques: Performance and Evaluation

After conducting experiments with various machine learning and ensemble techniques, the focus shifted towards neural networks, specifically employing the LSTM-RNN architecture. This phase aimed to explore the capabilities of LSTM-RNNs in handling the sequential nature of speech data and capturing long-term dependencies, which are crucial for modeling accented speech. Three experiments were conducted using the same set of features to identify the most effective method with the minimum Word Error Rate (WER).

9.5.2.1 Experiment 1:

The first experiment involved training the accented ASR model with the LSTM-RNN architecture. The model was trained and validated using an 8:2 ratio for the input data. The training process was executed over 2000 epochs with a batch size of nine, resulting in 482,000 steps. The comprehensive feature set enabled the LSTM-RNN to capture various aspects of the speech signal, from spectral characteristics to temporal and rhythmic patterns.

9.5.2.2 Experiment 2:

In the second experiment, the number of epochs and iterations was expanded to examine the effects on the model's performance. This approach led to a considerable increase in the accuracy metric, indicating that additional training allowed the LSTM-RNN to better capture the details in the accented speech data. The iterative

nature of training facilitated the model's ability to learn complex dependencies within the data, resulting in improved accuracy and reduced WER.

9.5.2.3 Experiment 3:

The third experiment further extended the training process but observed a decline in accuracy after a specific convergence point. This outcome highlighted the diminishing returns of extensive training and the potential for overfitting, where the model becomes too tailored to the training data and loses its generalization capability. Despite this decline, the experiment reinforced the importance of identifying an optimal training duration to balance between learning and overfitting.

The experiments conducted with the LSTM-RNN architecture provided valuable insights into the effectiveness of neural networks for modeling accented speech. The WER percentages from these experiments were compared with those obtained from other machine learning methods. The results indicated that the study conducted with the MLP displayed the minimum WER in comparison to all other methodologies, highlighting its superior performance in recognizing accented Malayalam speech.

9.5.2.4 Performance Evaluation

In the thorough experimentation process of building an AASR system for the Malayalam language, several models were evaluated to identify the most effective approach. Different machine learning and ensemble techniques were tested, leading to varied outcomes in terms of WER and accuracies. The results reflect the complexities of modeling accented speech and highlight the importance of selecting appropriate algorithms to suit specific data characteristics.

The LSTM-RNN-based acoustic model for accented speech recognition was developed and finalized after a rigorous series of iterations and fine-tuning. From each input speech data, the prominent 1530 features constituting the training features were randomly split, one-hot encoded, and then fed into the LSTM-RNN architecture. The maximum height of the utterances considered for the experiment

was consistently set to thousand for all speech samples, while the width of the signal encompassed the values within the feature set.

This configuration was integral in constructing a model capable of predicting the test features into any of the twenty defined classes of data. Table 10 illustrates the performance evaluation of LSTM-RNN.

Table 10 The performance Evaluation of LSTM-RNN

No. of epochs	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss	Steps	WER for known words	WER for unknown words
2000	93.30%	60.72%	0.29%	1.94%	482000	7%	39%
3000	96.97%	63.35%	0.11%	1.98%	723000	3%	37%
4000	95.74%	62.22%	0.15%	1.95%	1000000	4%	38%

Figure 47 and Figure 48 serve as essential visual aids in understanding the intricate dynamics of the LSTM-RNN model's performance. They illustrate the critical factors that contribute to the model's success, including the complex relationship between the number of epochs, accuracy, and loss, thus enhancing the comprehension of the optimal techniques for accented speech recognition in the Malayalam language. Figure 48 offers a visual depiction of the train and validation loss during the experiment with both 2000 and 3000 epochs. In this figure, two distinct lines are drawn to represent the different epochs evaluations. The cyan line corresponds to the evaluation with 2000 epochs, while the grey line is associated with the evaluation at 3000 epochs. This contrast allows for a clear comparison between the two phases of the experiment, highlighting the influence of the number of epochs on the model's training and validation loss.

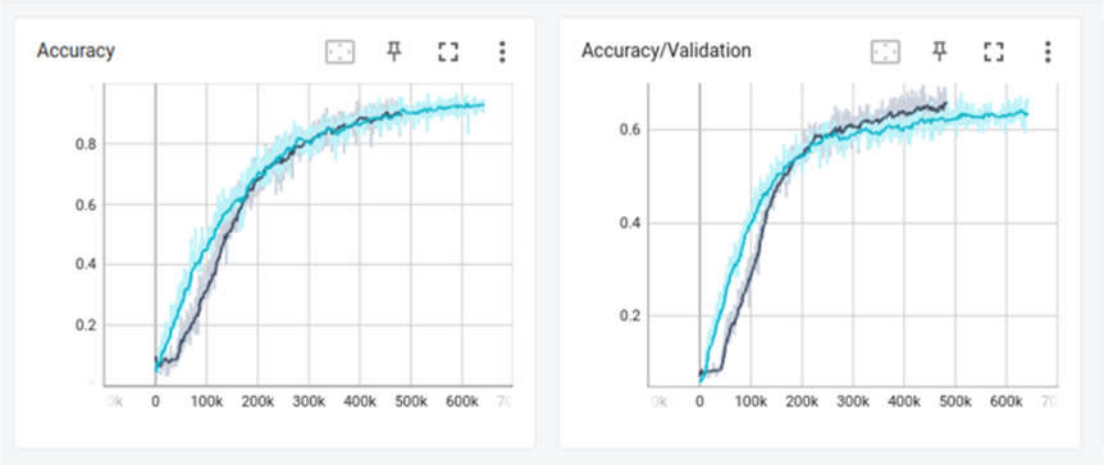


Figure 47 The Train and Validation Accuracy of Two Iterations

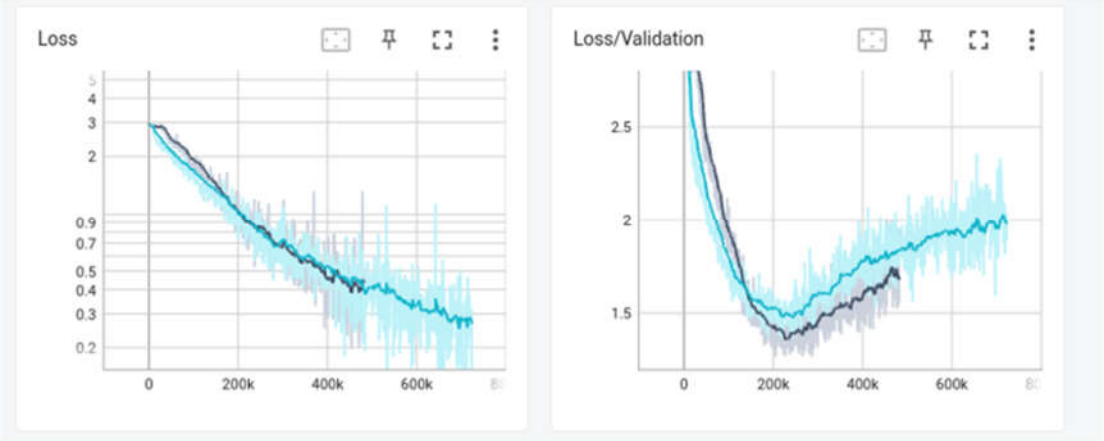


Figure 48 Train and Validation Loss of Two Iterations

The extended experimentation continued with 4000 epochs, encompassing a total of 1000000 steps. This phase yielded a train accuracy of 95.74%, a validation accuracy of 67.22%, a train loss of 0.15%, and a validation loss of 1.95%. While the model exhibited promising results with 3000 epochs, a noticeable decline in performance was observed at 4000 epochs. Figure 49 and Figure 50 illustrate the average accuracy and loss over 4000 epochs of the model construction respectively.

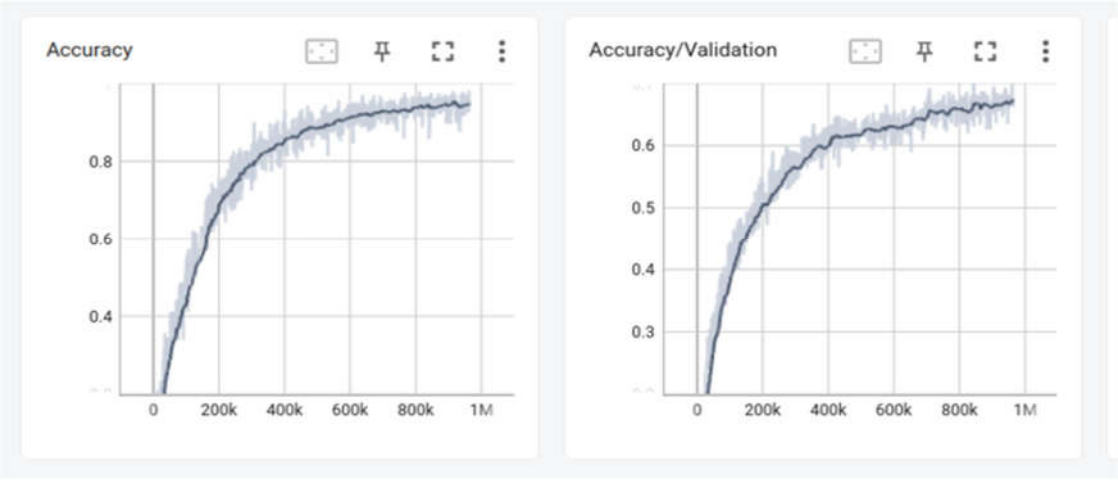


Figure 49 Train and Validation Accuracy Versus Steps

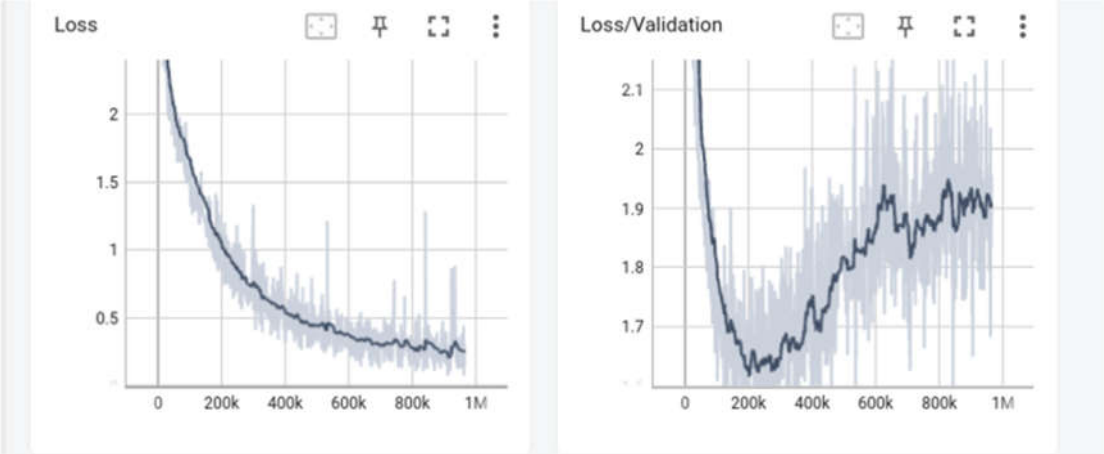


Figure 50 Train and validation Loss Versus Steps

9.6 Conclusion

This study undertook a comprehensive series of experiments to construct an accented Automatic Speech Recognition (ASR) system for the Malayalam language, employing a variety of machine learning and neural network approaches. The experiments aimed to identify the most effective method for handling accented speech, emphasizing the critical role of feature engineering and model optimization.

The LSTM-RNN experiments featured the importance of optimizing the number of epochs and iterations to balance between learning and avoiding overfitting. The second experiment, with 3000 epochs, achieved the highest train accuracy and the

lowest WER for known and unknown words, highlighting the model's ability to generalize well.

Throughout the experiments, the feature engineering process played a pivotal role. The extraction of diverse and informative features such as MFCCs, STFT values, Mel Spectrogram features, spectral roll-off, RMS values, and Tempogram rhythmic features enabled the models to effectively capture the complex nature of accented speech. These features provided rich representations that facilitated the learning process for both machine learning models and neural networks.

The study demonstrates that both traditional machine learning approaches and advanced neural network architectures, when combined with robust feature engineering, can effectively address the challenges of accented speech recognition. The ensemble model, employing the strengths of individual methods, and the LSTM-RNN with optimal training parameters, particularly stood out in their performance. These findings provide valuable insights and a strong foundation for future research and development in the field of accented ASR systems, contributing to the broader goal of creating more inclusive and accurate speech recognition technologies.

10. Deep Neural Networks and Attention Mechanisms for AASR in Malayalam

10.1 Introduction

The aim of this chapter is to explore and detail the development of a novel approach to AASR for Malayalam, employing advanced deep learning techniques such as RNN, LSTM, BiLSTM, and attention mechanisms. In conjunction with MFCC and Tempogram features, this study seeks to enhance the accuracy of recognizing multi-accented Malayalam speech.

This chapter will discuss the six distinct experimental phases and approaches undertaken, along with an in-depth analysis of spectral features in accent identification. A novel approach for gradient optimization, aimed at addressing the challenges in training deep learning models, will also be explained. This chapter aims to highlight the outcomes achieved by incorporating attention mechanisms, demonstrating improvements in accuracy by surpassing the performance of traditional models.

Through a comprehensive analysis of the outcomes, obstacles, and prospects, this chapter intends to contribute significantly to the research in the field of accented speech recognition. In the following sections, the construction of the dataset, the methodologies employed, and the detailed analysis of the results will be carefully examined to offer insights into the novel techniques and findings of this study.

10.2 Methodology

The task of Accented Automatic Speech Recognition (AASR) poses significant challenges, especially for low-resource languages such as Malayalam. In the early stages of development for both ASR and AASR, the availability of publicly accessible data for Malayalam is extremely limited. The dataset used for this study is AMSC-4, consisting of multi-syllabic words that capture the diverse accents within the

Malayalam-speaking regions. The experiment was designed to be conducted in six distinct phases, each phase representing a unique approach and combination of techniques. Figure 51 provides an overview of the phases involved in the proposed methodology.

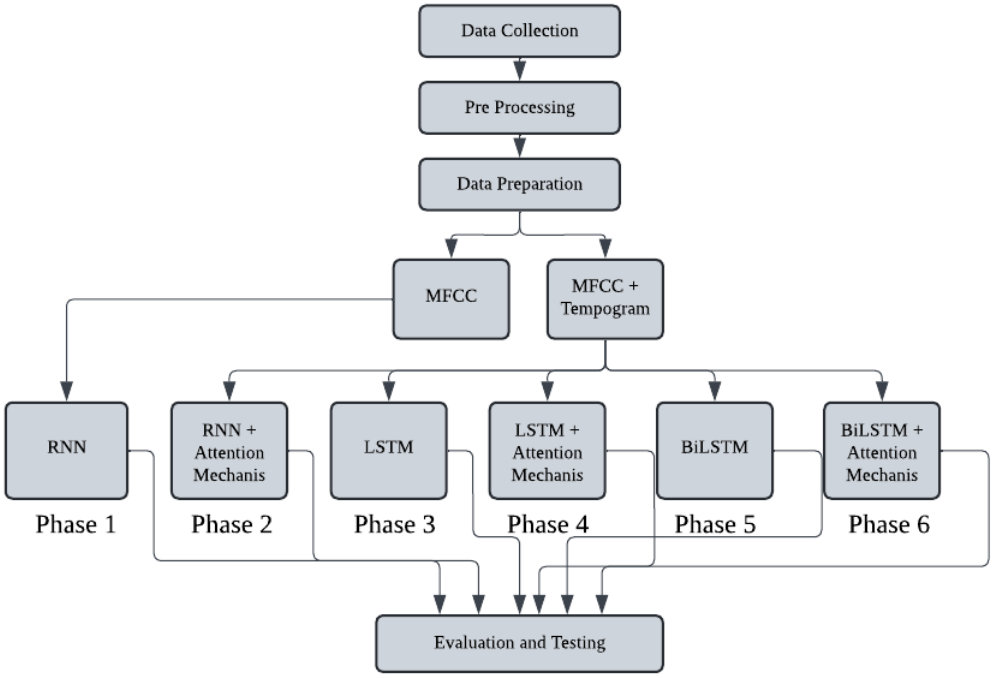


Figure 51 Steps Involved in the Proposed Methodology

As elaborated in earlier chapters, the lack of an existing benchmark dataset containing accented Malayalam speech presented a significant challenge in this study. The absence of such data initially impeded the progress of the research, necessitating the creation of a specialized corpus tailored to the specific needs of the experiment. This corpus, consisting of approximately 1.17 hours of accented speech from diverse districts in Kerala, was carefully constructed to reflect the rich variations in pronunciation, intonation, and rhythm found within the language.

The methodology behind the construction of this corpus, including the selection of regions, participants, and recording conditions, has been detailed in previous sections of this thesis. By addressing the gap in available accented speech data, the

creation of this corpus enabled a comprehensive evaluation of the proposed techniques for accented speech recognition in Malayalam, providing a robust foundation for the analysis and findings presented in this study. This dataset call attention to the innovative and responsive approach undertaken in this research, highlighting the adaptability and resourcefulness employed to overcome challenges inherent to the study of low-resource languages. The entire process of this research is discussed in this section in detail.

10.2.1 Data Collection and Feature Vectorization

In this study, the feature engineering phase employed two techniques: the extraction of MFCC and the Tempogram approach.

10.2.1.1 MFCC Extraction

The MFCC algorithm was crucial for capturing vital speech characteristics, providing a comprehensive representation of the accented speech signals. 40 feature vectors were extracted using MFCC where the first 13 coefficients (C [0,1,2….12]) were retained, capturing significant aspects of the spectral envelope. The C [0] to C [6] coefficients provided insights into the overall energy, spectral flatness, centroid, roll-off, and the first three formants. The C [7] to C [12] captured higher-order cepstral coefficients. The initial 13 coefficients were further extended with first and second derivatives, forming a set of 39 feature vectors, along with a mean value, resulting in 40 MFCC coefficients for this study.

10.2.1.2 Tempogram Feature Extraction

Alongside MFCC, Tempogram features were crucial in emphasizing accent and rhythm-specific features. A total of 384 speech vectors were extracted using Tempogram speech extraction techniques. Tempogram analysis effectively captured the tempo and accent-specific characteristics of the speech signal, akin to a spectrogram but with the y-axis representing tempo.

10.2.1.3 Integration of MFCC and Tempogram

The MFCC vectors and Tempogram features were concatenated together at various stages of the experiment. This innovative approach incorporated both the spectral information from MFCC and the rhythmic information from Tempogram, enhancing the overall representation of the speech data for improved analysis and recognition. The combination of MFCC and Tempogram yielded a total of 424 feature vectors which provided a rich representation of Malayalam's accented speech, capturing spectral details, energy, vocal tract resonances, tempo, and rhythm. The employment of these feature extraction techniques paved the way for an advanced recognition process, sensitively responding to the unique characteristics of the Malayalam language's intricate accent variations.

10.3 Accented Model Construction

To explore and address this complex issue, the study designed and developed six distinct models, each reflecting a comprehensive approach for recognizing Malayalam's accented speech. The experiment was strategically divided into six phases, each one representing a key step in the overall process. The following subsections detail the methodologies employed in each phase, emphasizing the techniques and innovations that contribute to the effectiveness of the models.

10.3.1 Phase 1: The RNN Approach

In the initial phase of this experiment, RNNs were utilized to recognize Malayalam accented speech using MFCC features. These features have been selected for their capability in accurately capturing essential spectral characteristics of accented speech. RNNs were specifically employed due to its following capabilities:

1. Temporal Dependency Handling: RNNs are proficient at identifying temporal relationships and patterns in sequential data, making them highly suited for speech signal analysis.

2. Variable Length Processing: This architecture can adapt to speech samples of varied lengths, an essential trait for handling the real-world variability of speech durations.

3. Dynamic Pattern Recognition: The ability to recognize intricate dynamic patterns in speech makes RNNs particularly effective for Malayalam accented speech.

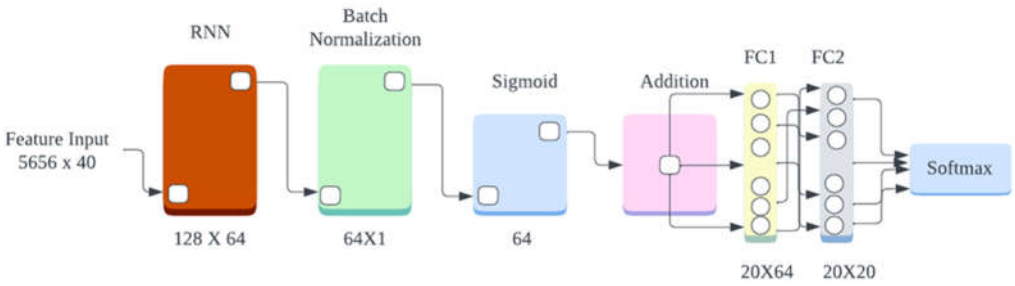


Figure 52 Proposed RNN

The architecture of the RNN used in this study is illustrated in Figure 52. The RNN architecture employed in this phase is described as follows:

1. Input Layer: The network accepts an input of 40 MFCC features, which includes information about the spectral characteristics, energy, formants, and fine spectral details of the speech signal.

2. RNN Processing Layer: Sequentially processes the input, maintaining an internal memory that captures contextual information.

3. Batch Normalization Layer: This layer ensures consistent data distribution through normalization.

4. Activation Layer (Sigmoid): Implements non-linear transformations to enable complex relationship learning.

5. Concatenation and Summation: The output vectors are joined and summed to create a comprehensive representation.

6. Softmax Layer: Utilizes a probabilistic function to determine the target class with the highest likelihood.

The use of MFCC features in the RNN-based approach demonstrated promising results for Malayalam accented speech recognition. The combination of MFCC's spectral insight with RNN's sequential processing capabilities created a robust system for accurate recognition. This phase's systematic flow, including the use of RNN for processing, normalization, activation functions, and concatenation, has shown to be effective in recognizing Malayalam accented speech using MFCC features.

10.3.2 Phase 2: RNN with Attention Mechanism

Phase 2 of the experiment centers on implementing an enhanced RNN architecture that incorporates an attention mechanism. This attention block is paired with 424 MFCC and Tempogram features to create a more efficient model for accented speech recognition.

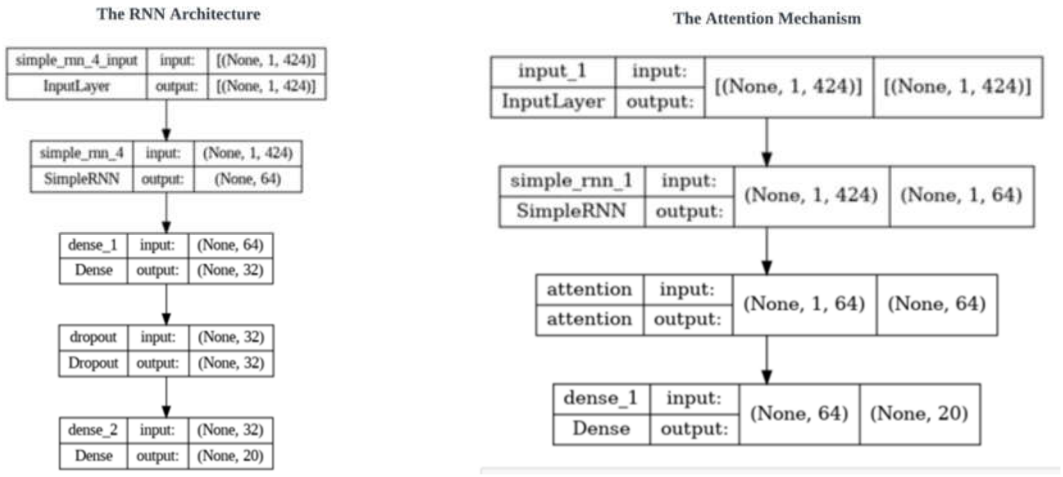


Figure 53 Proposed RNN with Attention Mechanism

Figure 53 represents the architecture of the proposed RNN with attention mechanism and the overview of the architecture is detailed below.

1. Input Layer: The feature input (424 vectors) is fed into the RNN.
2. RNN Processing Layer: Sequential processing facilitates the model's understanding of temporal dependencies and long-term relationships within the audio sequences.

3. Dense Layers: Two dense layers extract complex relationships from the processed data, improving the model's comprehension of accented speech patterns.

4. Dropout Layer: Integration of a dropout layer acts as a regularizing agent, minimizing overfitting and making the model more robust.

5. Activation Functions:

    - Sigmoid: Ensures the output values are within the range of 0 to 1.

    - ReLU: Enhances non-linearity without affecting the receptive fields of the convolution.

6. Attention Mechanism: A crucial addition to the architecture that allows the model to focus iteratively on the most relevant segments of the speech. By directing attention, the model achieves superior recognition of the accented patterns.

7. Softmax Layer: Classifies the processed information into the target classes, making probabilistic predictions that represent the final transcription result.

Phase 2 stands out for its incorporation of an attention mechanism with the RNN architecture. By implementing the combination of MFCC and Tempogram features, the model is provided with a rich representation of the speech signal.

The detailed structure that includes dense layers, dropout for regularization, and activation functions ensures that the architecture is both robust and sensitive to the intricacies of accented speech. The attention mechanism's ability to iteratively refine the model's focus on critical speech segments contributes to enhanced performance in recognizing and transcribing accented speech patterns in Malayalam.

This phase successfully builds upon the earlier RNN approach, marking a significant stride towards the effective recognition of Malayalam accented speech. This complex architecture, incorporating various layers and attention to the most salient parts of

the speech signal, offers promising results for accented speech recognition, paving the way for future innovations in this domain. The incorporation of the attention mechanism into the RNN architecture marks a significant advancement in this phase. By enabling the model to selectively concentrate on the most relevant portions of the accented speech, it offers a sophisticated approach to recognizing and transcribing the Malayalam accented speech.

10.3.3 Phase 3: The LSTM Approach

Phase 3 of the experiment builds upon previous methods by employing LSTM networks to address the accented speech recognition challenges. LSTMs are utilized to effectively mitigate the vanishing gradient issues that are commonly encountered in traditional RNN architectures. They are capable of learning and retaining long-term dependencies within the sequences, which is crucial for accented speech recognition. The architecture is designed to circumvent the vanishing gradient problem, allowing for deep learning with numerous layers.

Following the LSTM processing, the output undergoes normalization through the batch normalization layer. This ensures that activations possess zero mean and unit variance, preventing gradient explosions and stabilizing the training process.

A Rectified Linear Unit (ReLU) activation function is then applied to introduce non-linearity to the data. The outputs from these layers are concatenated, forming a comprehensive representation of the features. Further processing through dense layers refines the understanding of the accented speech patterns. The final layer in the architecture is the softmax layer, which utilizes a probabilistic function to assign probabilities to different classes, determining the most likely target class.

Phase 3 with the LSTM approach brings a novel dimension to the experiment, focusing on the intricate recognition of Malayalam accented speech. By exploiting the LSTM's ability to manage long-term dependencies and avoid the vanishing gradient problem, this phase ensures a robust analysis of the speech signals.

The relationship between feature extraction (MFCC and Tempogram), LSTM layers, normalization, non-linear activation, dense layers, and probabilistic classification culminates in an architecture capable of enhanced performance. The use of LSTM layers in conjunction with other techniques presents a favorable direction in accented speech recognition. This phase not only consolidates the insights from previous phases but also extends them, marking a significant progression in the study's overall objectives. The proposed LSTM in this phase is illustrated in Figure 54.

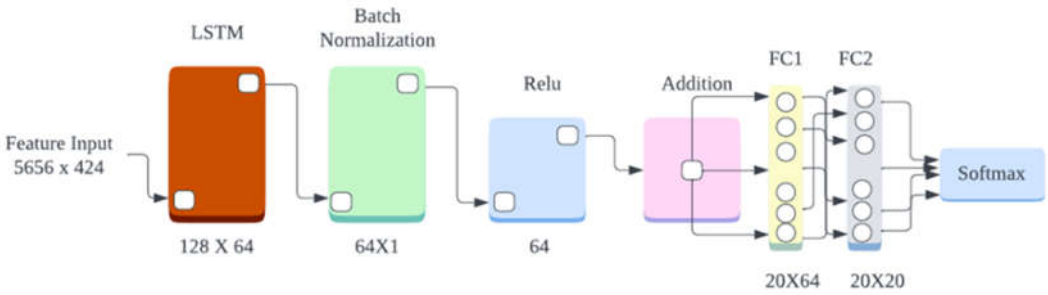


Figure 54 Proposed LSTM

10.3.4 Phase 4: LSTM with Attention Mechanism

Phase 4 of the experiment introduces an innovative approach by incorporating an LSTM model with an attention mechanism, further refining the process of accented speech recognition for the Malayalam language. This phase employs 424 feature vectors, consisting of both MFCC and Tempogram vectors, extracted from the accented audio data. Together, these vectors capture the spectral and rhythmic characteristics that are vital for recognizing the accents in speech.

The proposed LSTM architecture in this phase is composed of two primary branches: an operational block and a skip connection block. The operational block serves as the main pathway for processing the extracted features. It handles the complex dependencies within the audio sequences, employing the strengths of LSTM in modeling time-series data. Figure 55 illustrates the architecture of the proposed LSTM with an attention block.

The skip connection block plays a crucial role in emphasizing relevant activations during the training process, allowing the network to focus on the most informative and discriminative features. This selective focus is achieved through the integration of an attention block within the LSTM architecture. The attention mechanism functions by selectively attending to the parts of the input sequence that are most relevant to the task.

By highlighting these specific regions of the input, the attention block minimizes the computational resources wasted on irrelevant activations. This optimization not only improves computational efficiency but also enhances the model's capacity to recognize subtle variations in accented speech patterns.

The synergy between the LSTM layers, attention block, and skip connection in this architecture offers a more sophisticated solution for accented speech recognition. It enhances the system's performance by effectively reducing unnecessary computations and pinpointing the essential aspects of the audio data. By focusing on key activations and utilizing the long-term memory capabilities of LSTM, this phase presents a powerful approach that capitalizes on the most significant parts of the input sequence.

The inclusion of the attention mechanism and skip connection within the LSTM architecture marks an advanced step in the ongoing experiment, contributing to the accuracy and efficiency of the accented speech recognition system. Phase 4 thus signifies a meaningful evolution in the study, providing insights into optimized deep learning architectures specifically tailored for Malayalam accented speech recognition.

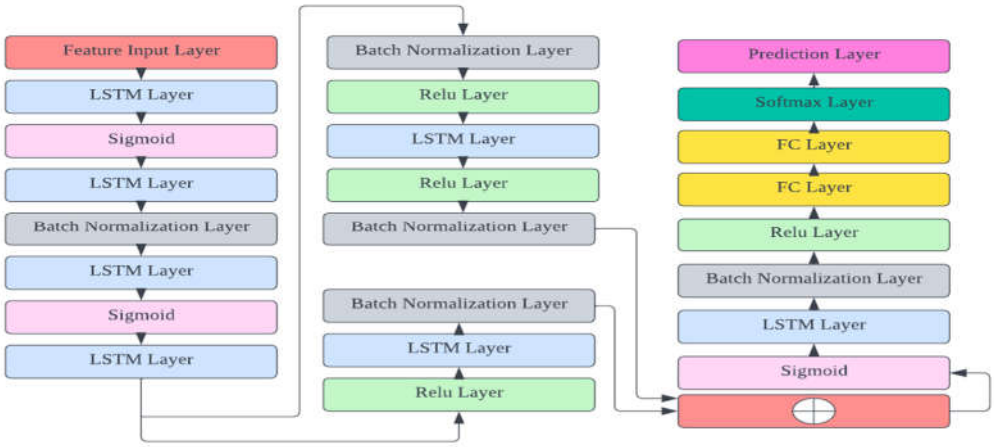


Figure 55 Proposed LSTM with Attention Block

10.3.5 Phase 5: BiLSTM Approach

In the fifth phase of the experiment, the exploration of accented speech recognition in Malayalam deepened with the utilization of BiLSTM methodology. This approach allowed the model to use the strengths of both forward and backward LSTM layers, creating a comprehensive understanding of the input sequence, specifically the accented speech vectors obtained by combining MFCC and Tempogram vectors. The BiLSTM model architecture is uniquely designed to process the input sequence in two directions. It consists of two sets of LSTM layers, with one set processing the sequence in the forward direction and the other in the backward direction. This bidirectional processing is essential to capture relevant temporal dependencies within the speech signal. By simultaneously considering both past and future contexts, the BiLSTM model can discern intricate patterns and dynamics present in accented speech more effectively.

The true power of this bidirectional approach lies in its ability to combine information from both directions. By synthesizing the insights gleaned from the forward and backward processing of the input, the BiLSTM model gains a holistic view of the sequence. This enables the model to recognize and interpret the unique characteristics of accented speech patterns with greater precision. Such a

comprehensive perspective enhances the model's ability to capture the variations in pronunciation, intonation, and rhythm that are often characteristic of accented speech.

The utilization of the BiLSTM architecture in this phase signifies a significant advancement in the ongoing study. It exemplifies a more complex approach to modeling accented speech, using bidirectional processing to cultivate a deeper understanding of the underlying complexities. By integrating both past and future context within the analysis, this phase succeeds in further refining the recognition and interpretation of accented speech patterns. The insights derived from this phase contribute to the broader endeavor of developing more robust and context-aware speech recognition systems designed to the unique challenges posed by Malayalam accented speech.

10.3.6 Phase 6: BiLSTM with Attention Mechanism

The sixth phase of the experiment builds upon the previous methodologies by integrating the strengths of both BiLSTM and attention mechanisms. This combination facilitates a more complicated approach to recognizing and transcribing accented speech in the Malayalam language, specifically addressing the challenges posed by variations in pronunciation, intonation, and rhythm creating a comprehensive understanding of the input sequence, specifically the accented speech vectors obtained by combining MFCC and Tempogram vectors. The architecture of the BiLSTM model enables the capturing of rich contextual information by processing the input sequence in both forward and backward directions.

While the forward LSTM layer analyzes the sequence starting from the beginning, the backward LSTM layer examines it from the end. This bidirectional analysis provides a comprehensive understanding of the accented speech by considering both past and future contexts, allowing the model to detect intricate patterns and relationships within the audio signal.

Incorporated into this bidirectional structure is the attention mechanism, which introduces an adaptive approach to sequence analysis. The attention mechanism assigns different weights to specific segments of the input sequence, dynamically focusing on those parts that are most relevant for accurate recognition. This selective attention enables the model to allocate its resources adaptively, giving more weight to crucial features and minimizing the influence of irrelevant or redundant information.

The fusion of BiLSTM and attention mechanisms provides a powerful means of capturing long-term dependencies within the speech signal while simultaneously concentrating on the salient aspects of accented speech. By effectively balancing the considerations of context, complexity, and relevance, this approach enhances the recognition and transcription accuracy significantly. This phase demonstrates a sophisticated combination of technologies to tackle the inherent challenges in accented speech recognition.

It underlines the necessity of focusing not just on individual phonetic components but on the dynamic interplay of those components within the broader context of the speech sequence.

Through the careful integration of BiLSTM and attention mechanisms, the model provides a robust solution that accommodates the unique characteristics of Malayalam accented speech, paving the way for further advancements in this field.

10.4 Performance Evaluation

The result and evaluation section of the study serves as a comprehensive reflection of the six distinct experimental phases that were undertaken to explore AASR for the Malayalam language. This culmination of the experiment employed varying methodologies, architectures, and feature vectors to derive substantive insights into the field of AASR. Throughout the experiments, the environmental setups and

training parameters were consistently maintained to facilitate unbiased comparisons across different phases.

For example, the Adam optimizer was utilized in the first phase, followed by the RMSprop optimizer in subsequent phases 2 to 6. Learning rates were also systematically adjusted, with 0.001 for phases 1 and 2, and 0.01 for phases 3 to 6. The epochs varied between phases to accommodate the different learning rates of the models, with 3000 epochs for phases 1 and 2, 2000 epochs for phases 3 and 4, and 68 epochs for phases 5 and 6. The categorical cross-entropy loss function was consistently applied across all phases to enable a standard assessment and comparison. Table 11 illustrates the performance of the different phases of the study.

Table 11 Performance Evaluation

Phases	**Train Accuracy**	**Validation Accuracy**	**Train Loss**	**Validation Loss**	**No. of Epochs**
Phase I	87.18%	65.15%	0.0096%	0.0277%	3000
Phase II	92.02%	72.61%	0.0074%	0.0317%	3000
Phase III	94.10%	64.87%	0.0050%	0.0309%	2000
Phase IV	96.27%	73.03%	0.0031%	0.0291%	2000
Phase V	96.25%	72.45%	0.0026%	0.1093%	68
Phase VI	97.37 %	74.27%	0.0036%	0.0025%	68

In Phase I, the model achieved a training accuracy of 87.18% but showed a significant drop in validation accuracy at 65.15%. This disparity might indicate some overfitting to the training data. The loss values were low for both training and validation, with a relatively high number of epochs at 3000. In Phase II, both training and validation accuracies improved to 92.02% and 72.61%, respectively, signifying better generalization.

The train loss decreased, while there was a slight increase in validation loss. Phase III and Phase IV continued to show an upward trend in training accuracy, reaching

over 96% in Phase IV. However, the validation accuracy fluctuated, with Phase IV showing improvement over Phase III. Similarly, both train and validation losses decreased, indicating an improvement in model fit.

The last two phases, Phase V and Phase VI, maintained high training accuracy, with a slight increase in Phase VI to 97.37%. The validation accuracy also increased, but not as significantly as train accuracy, hinting at possible overfitting. Notably, the train loss continued to decrease, while the validation loss in Phase V showed an unexpected spike. The number of epochs dramatically reduced to 68 in these phases, indicating that the model was learning much faster. The results demonstrate a general trend of improvement across phases, particularly in training accuracy and loss. Conversely, the inconsistent improvement in validation results might suggest that the models in later phases could be overfitting to the training data.

10.5 Conclusion

The result and evaluation section of the study serves as a comprehensive reflection of the six distinct experimental phases undertaken to explore Accented Speech Recognition (AASR) for the Malayalam language. These phases employed varying methodologies, architectures, and feature vectors, contributing substantive insights into the field of AASR. Throughout the experiments, environmental setups and training parameters were consistently maintained to facilitate unbiased comparisons across different phases.

The experiment generated insights into AASR by employing diverse methodologies and exploring various experimental setups. This led to conclusions regarding the most effective approaches for accented speech recognition in Malayalam. The architecture was fine-tuned across multiple experimental trials with different combinations of optimizers, loss functions, and neural network architectures.

The study's conclusion highlights the general trend of improvement across phases, particularly in training accuracy and loss. However, inconsistent improvement in

validation results raises concerns about potential overfitting in later phases. Nonetheless, the comprehensive exploration of methodologies, architectures, and feature vectors provides valuable insights into effective approaches for AASR in Malayalam. The study emphasizes the importance of optimizing architectures and training parameters to achieve superior performance in accented speech recognition, paving the way for further advancements in the field.

11. Enhancing AASR through Advanced Integration of Self-Supervised Learning and Autoencoders with ML Models

11.1 Introduction

Accented speech introduces significant variability in phonetic realization, which can impede the performance of traditional ML models. This variability necessitates the development of robust feature extraction and classification methods that can generalize well across different accents while maintaining high accuracy.

This chapter explores an innovative approach to addressing these challenges through the fusion of autoencoders with various ML classifiers. By utilizing the feature extraction capabilities of autoencoders and the discriminative power of ML models, the goal is to enhance the accuracy and robustness of accented speech recognition systems. Specifically, the study investigates the performance improvements obtained by integrating autoencoder-generated features with classifiers such as Linear Regression, Decision Trees, SVM, Random Forests, KNN, SGD, and MLP.

The fusion of autoencoders with ML models offers several advantages. Autoencoders excel at capturing complex, non-linear relationships within the data and reducing noise, leading to more informative and robust feature representations. When these representations are utilized by ML models, the classifiers can utilize this distilled information to achieve better generalization and predictive performance. This synergy between unsupervised feature learning and supervised classification forms the core of the approach.

The rationale behind the selection of ML algorithms in this study is grounded in their complementary strengths and suitability for handling encoded representations. SVMs, known for their efficacy in high-dimensional spaces and non-linear decision boundaries, are well-matched with the features extracted by autoencoders. Decision Trees and ensemble methods like Random Forests offer interpretability and

robustness, respectively, while neural network models such as MLPs can further exploit the non-linear relationships captured by autoencoders.

This chapter systematically evaluates the performance of these fusion models on a classification task, comparing their accuracy with and without the use of autoencoder-generated features. The results demonstrate significant improvements in classification accuracy across all evaluated models when autoencoder features are incorporated, underscoring the effectiveness of this approach. By presenting this comprehensive analysis, the aim is to provide valuable insights into the potential of combining autoencoders with ML models for enhanced speech recognition. The findings of this study contribute to the broader field of machine learning, highlighting the benefits of integrating deep learning techniques with traditional ML classifiers to tackle complex real-world problems.

11.2 Methodology

The autoencoder used in this study belongs to the category of denoising autoencoders. Denoising autoencoders are a specific type of autoencoder that are trained to reconstruct input data from a corrupted version of it. This process forces the model to learn robust, meaningful representations that capture important features of the data while filtering out noise. This category of autoencoders is particularly effective in enhancing the quality of the learned features, which in turn improves the performance of subsequent machine learning models when these features are used as input. Figure 56 describes the different phases of the study conducted.

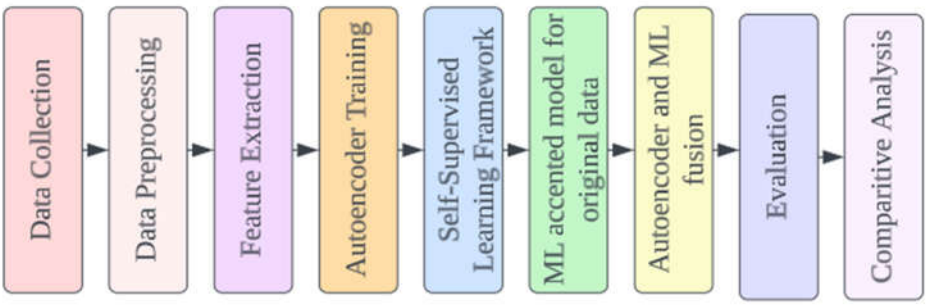


Figure 56 Steps Involved in the Study

11.2.1 Dataset Collection

AMSC-4 is the dataset used for conducting this study. Among the 7070 samples in the dataset 1414 samples were not considered for doing this experiment. The decision to utilize 5656 speech samples instead of the full 7070 was grounded in exact quality assurance procedures. During the preprocessing and data cleaning stages, a thorough examination revealed that certain samples exhibited the issues of corruption and low audio quality. Consequently, to uphold the reliability and integrity of the dataset, a stringent selection process was employed, ensuring that only high-quality and consistent samples were retained for further analysis. This approach safeguards against potential distortions or inaccuracies in the data, thereby enhancing the trustworthiness of the experimental outcomes.

11.2.2 Data Preprocessing

In the initial phase of this research, a comprehensive cleaning and preprocessing of the collected Malayalam speech data was executed to remove any undesirable artifacts, background noise, or non-relevant segments that might compromise the quality of the accented speech recognition model. The preprocessing techniques are discussed below:

1. Noise Reduction: Given the potential for irrelevant auditory disturbances within the raw audio data, advanced noise reduction algorithms were employed. These algorithms were designed to identify and minimize any underlying, consistent noise without altering the core speech component.

2. Filtering: Filtering was another essential step in the preprocessing methodology. By applying various band-pass filters, the frequency range was isolated that is most relevant to human speech, thereby eliminating frequencies that do not contribute to the comprehension of the Malayalam language accents. The design of the filters was carefully tailored to the unique characteristics of the Malayalam language, thereby enhancing the overall efficiency of the process.

3. Normalization: To further ensure consistency across the entire dataset, normalization techniques were employed. Normalization played a crucial role in modifying any variations in volume, pitch, and other speech attributes across different recordings. By standardizing these attributes, the learning process was facilitated for the model, allowing it to focus on the intrinsic patterns of the accented speech rather than irrelevant variations.

11.2.3 Feature Extraction

Efforts have been put to extract relevant acoustic features from the preprocessed speech data. These features capture vital characteristics of the speech, including pitch, tempo, spectral frequencies, and rhythm. To achieve this, three key feature extraction techniques were employed: 40 MFCC feature vectors, 12 STFT feature vectors, and 384 Tempogram feature vectors were computed as discussed in the previous chapters. By combining these techniques, a total of 436 features were extracted for each speech sample. This comprehensive representation encompasses essential attributes related to pitch, tempo, spectral frequencies, and rhythm. These features provide a robust foundation for subsequent analysis and classification tasks, setting the stage for innovative applications in the domain of accented speech recognition.

11.2.4 Autoencoder and Self-Supervised Learning Framework

In the domain of accented speech recognition for the Malayalam language, the design and implementation of efficient neural network architectures play a pivotal role. In this research, an autoencoder architecture was proposed, consisting of both an encoder and a decoder, which operates on the preprocessed speech data, encompassing 436 feature vectors. The proposed architecture of the autoencoder model without compression is depicted in Figure 57.

The encoder network transforms the high-dimensional input features into a lower-dimensional latent space, preserving the essential characteristics of the accented speech. The first encoder layer performs a linear transformation on the input X,

followed by batch normalization and the Leaky ReLU activation function, mathematically represented as $e=W_1 \cdot X+b_1$.

Following the first layer, a second linear transformation is applied, again followed by batch normalization and the Leaky ReLU activation function, expressed as $e=W_2 \cdot e+b_2$. The bottleneck layer maps the transformed data into a space with the same number of neurons as the number of input features, given by $\text{bottleneck}=W_b \cdot e + b_b$.

The decoder network aims to reconstruct the original speech data from the latent space, enabling a comprehensive understanding of the speech signals. The first decoder layer is expressed as $d=W_{d1} \cdot \text{bottleneck}+b_{d1}$, followed by a second transformation performed as $d=W_{d2} \cdot d+b_{d2}$, with the final output layer delivering the reconstructed data: $\text{output}=W_{\text{out}}.d + b_{\text{out}}$.

The weight matrices (W_1, W_2, Wb, W_{d1}, W_{d2}, W_{out}) and bias vectors (b_1, b_2, b_b, b_{d1}, b_{d2}, b_{out}) are optimized during the training process to perform the respective transformations. In the subsequent phase, the autoencoder model was further refined with compressed data. The encoding layers perform transformations like the first phase, denoted by equations $e=W_1 \cdot F+b_1$, $e=W_2 \cdot e+b_2$, and the bottleneck layer in this phase can be represented as $\text{bottleneck}=W_b \cdot e + b_b$.

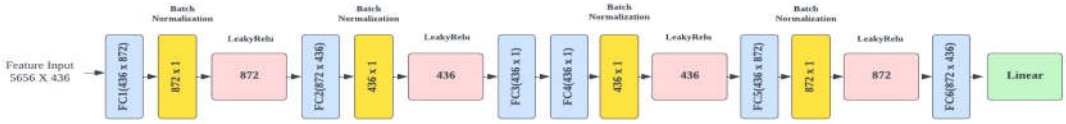


Figure 57 Autoencoder Model Architecture Without Compression

The first decoder layer performs a linear transformation followed by batch normalization and Leaky ReLU activation elementwise and can be represented as [169]:

$$d = W\_d_1 * bottleneck + b\_d \quad (36)$$

where W_d_1 is the weight matrix and b_d_1 is the bias vector, and d is the output. The second decoder layer performs another linear transformation followed by batch normalization and Leaky ReLU activation elementwise.

Mathematically, it can be represented as [169]:

$$d = W\_d_2 * d + b\_d_2 \quad (37)$$

where W_d_2 is the weight matrix and b_d_2 is the bias vector, and d is the output. The output layer performs a linear transformation with a linear activation function. Mathematically, it can be represented as [169]:

$$output = W\_out * d + b\_out \quad (38)$$

where W_out is the weight matrix and b_out is the bias vector, and output is the final output. Here in this architecture, the encoder is composed of two dense layers with leaky ReLU activation and batch normalization. The first dense layer has twice the number of neurons as the input features, and the second dense layer has the same number of neurons as the input features. These layers gradually reduce the dimensionality of the data and capture meaningful representations. The bottleneck layer, which is the output of the second dense layer, has half the number of neurons as the input features. It serves as the compressed representation of the input data.

The decoder is constructed as the reverse of the encoder architecture. It also consists of two dense layers with leaky ReLU activation and batch normalization. The output layer has the same number of neurons as the input features and uses a linear activation function. During training, the autoencoder aims to reconstruct the input data by minimizing the difference between the input and output. The training is performed for 500 epochs with a batch size of 16. Additionally, an encoder model is defined by specifying the input and bottleneck layers. This model is used to extract the compressed representation of the input data.

Figure 57 illustrates the neural network architecture that is tailored for processing feature input with dimensions 5656×436. This architecture consists of several components, starting with the input layer, which accepts feature data comprising 5656 samples, each containing 436 features. Following the input layer, the architecture includes a series of fully connected layers (FC1-FC6). FC1 transforms the input features to a higher-dimensional space with 872 units, while FC2 reduces the output back to 436 units. Subsequent layers, FC3 and FC4, further transform the data without altering its dimensionality. The features are then expanded again in FC5 to 872 units, followed by FC6, which linearly reduces the dimensionality to match the original 436 features.

In addition to the fully connected layers, batch normalization and activation functions are applied after each layer. Batch normalization normalizes the input to each layer, enhancing training speed and stability, while a LeakyReLU activation function is used to introduce a small, non-zero gradient when the unit is inactive, addressing the "dying ReLU" issue. This combination of batch normalization and activation functions ensures network stability and effective learning throughout the architecture.

Overall, the architecture alternates between expanding and contracting the dimensionality of the feature space, facilitating the capture of complex patterns within the data. The recurrent use of batch normalization and LeakyReLU activation functions further enhances the network's stability and learning efficiency. Finally, a linear layer is incorporated to generate the final output with 436 units, aligning with the input features' dimensionality.

The autoencoder architecture presented in Figure 58 comprises several key components aimed at extracting meaningful features from input data and performing necessary transformations for subsequent tasks. In the beginning, the input layer receives feature input with dimensions 5656 x 436, denoting 5656

samples, each containing 436 features. This sets the stage for processing a substantial dataset rich in feature information.

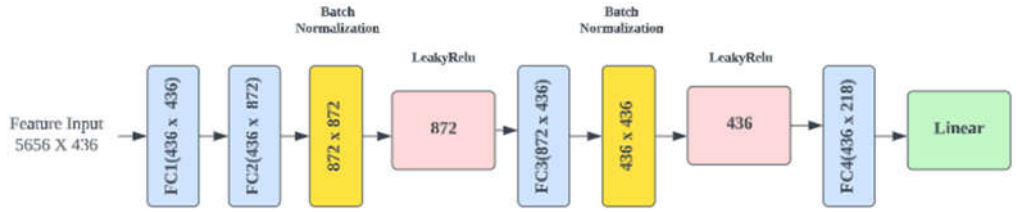


Figure 58 Autoencoder Model Architecture with Compression

The architecture proceeds with Fully Connected Layer 1 (FC1), which maintains the input's dimensionality (436) while linearly combining each feature with 436 neurons in the subsequent layer. This initial transformation serves as a foundational step in feature extraction and representation learning.

Fully Connected Layer 2 (FC2) expands the dimensionality, mapping the 436 input features to 872 features. Batch normalization is then applied to the output of FC2, followed by a LeakyReLU activation function. These steps enhance training stability and introduce non-linearity to the model, enabling it to learn complex patterns effectively.

Fully Connected Layer 3 (FC3) reduces the dimensionality back to 436 from 872, consolidating the learned representations. Like previous layers, batch normalization and LeakyReLU activation are applied for normalization and non-linearity introduction, respectively.

Further dimensionality reduction occurs in Fully Connected Layer 4 (FC4), where the 436 features are mapped to 218 features. The output layer produces the final output without applying an activation function, implying a linear combination of the input features.

This architectural design signifies a deliberate progression of dimensionality changes aimed at extracting and refining meaningful features. The incorporation of batch

normalization and LeakyReLU activation throughout the architecture ensures better performance and training stability, crucial for effective representation learning and subsequent task execution. Overall, this autoencoder architecture demonstrates a systematic approach to feature extraction and transformation, aimed to yield valuable insights and facilitate accurate predictions in various applications.

11.2.5 Fusion of Autoencoder with Machine Learning Approaches

When autoencoder-generated features are fused with ML models, notable enhancements in classification accuracy are observed across the board. For instance, Linear Regression achieves a significantly higher accuracy of 94.54% with autoencoder-generated features compared to 89.39% without, highlighting the efficacy of this fusion approach in regression tasks.

Similarly, Decision Trees experience a substantial improvement in accuracy to 85.15% with autoencoder-trained features in conjunction with ML models, compared to only 27.75% without, underscoring the importance of leveraging both encoded representations and the discriminative power of ML algorithms to enhance predictive performance.

SVMs also exhibit a marked increase in accuracy to 95.15% with the fusion approach, compared to 44% without indicating the synergistic benefits of combining autoencoder-generated features with SVM classification.

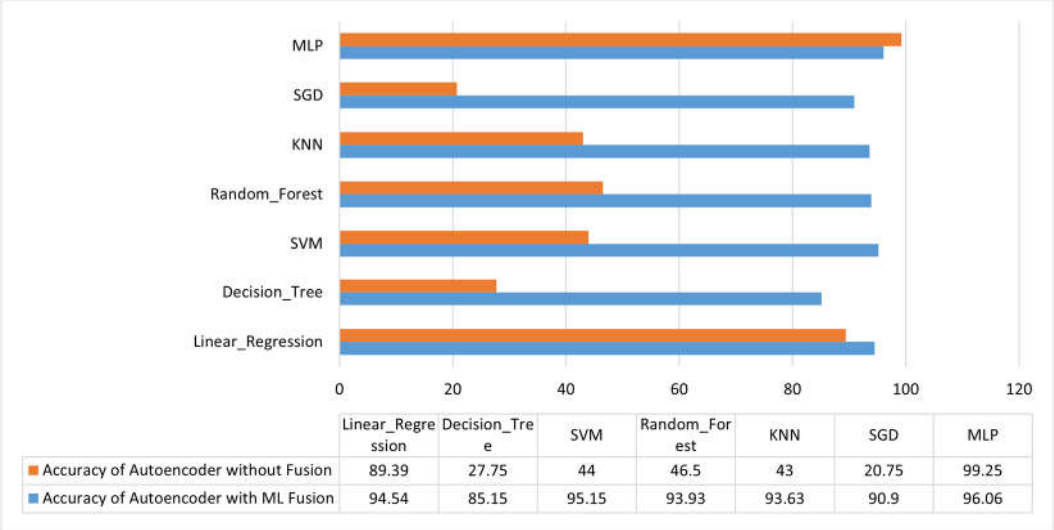
Random Forest demonstrates a significant boost in accuracy to 93.93% with autoencoder-trained features, compared to 46.5% without further emphasizing the advantages of leveraging encoded representations within ensemble learning frameworks.

KNN and SGD classifiers also experience notable improvements in accuracy to 93.63% and 90.9%, respectively, with the fusion approach, compared to 43% and 20.75% without, suggesting that integrating autoencoder-generated features with

ML models enhances the discriminative power and predictive performance of these classifiers.

Moreover, MLP achieves a high accuracy of 96.06% with autoencoder-generated features in conjunction with ML models, compared to 99.25% without, highlighting the effectiveness of the fusion approach in leveraging both the representational learning capabilities of autoencoders and the nonlinear modeling capabilities of neural networks.

In contrast, when autoencoder-generated features are used alone without ML models, classifiers generally exhibit lower accuracy levels, indicating the importance of integrating encoded representations with ML algorithms. Overall, the fusion of autoencoders with ML models offers a robust and comprehensive solution for enhancing predictive accuracy and model performance in various classification tasks. Figure 59 and Figure 60 visualize the performance evaluation of the experiments conducted in this study.



	Linear_Regression	Decision_Tree	SVM	Random_Forest	KNN	SGD	MLP
Accuracy of Autoencoder without Fusion	89.39	27.75	44	46.5	43	20.75	99.25
Accuracy of Autoencoder with ML Fusion	94.54	85.15	95.15	93.93	93.63	90.9	96.06

Figure 59 Performance Evaluation

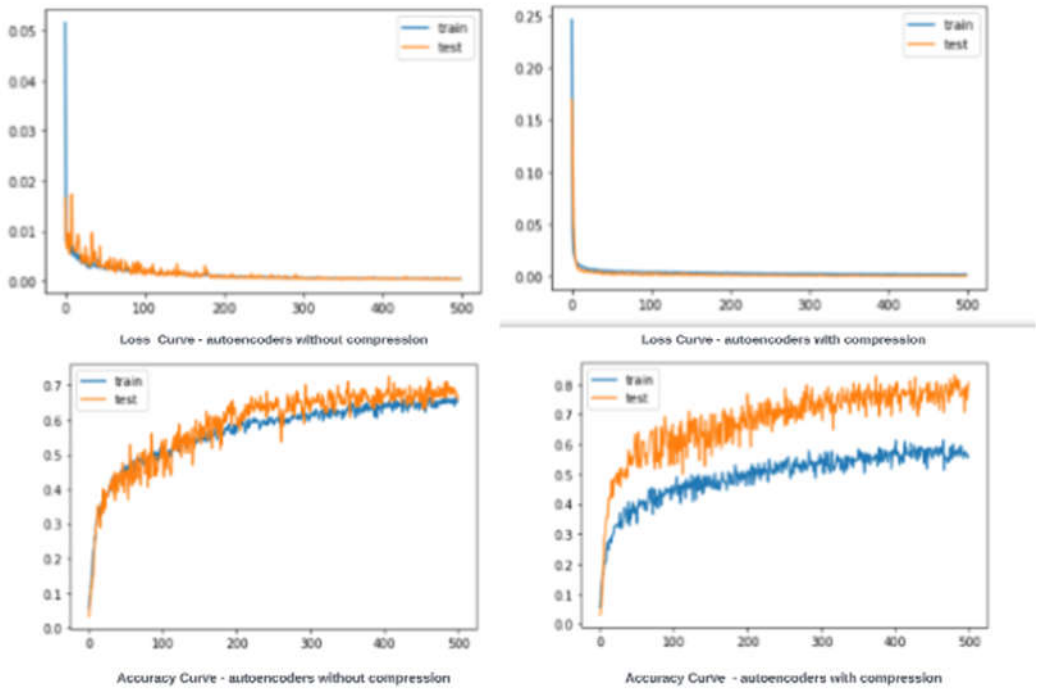


Figure 60 Learning Curves for Autoencoder-Based Accent Modelling

11.3 Conclusion

The study reveals that autoencoders, by learning compressed representations of the input data, capture essential features and reduce noise, thereby providing more informative and robust inputs for ML classifiers. This enhanced feature set, when utilized by ML classifiers leads to superior generalization and higher classification accuracy compared to models trained solely on raw data.

These findings underline the significant benefits of combining the unsupervised feature learning capabilities of autoencoders with the discriminative power of various ML models. The choice of classifiers, each with its unique strengths, complements the encoded features generated by autoencoders, resulting in a synergistic effect that enhances overall model performance.

The fusion of autoencoders with ML models offers a robust and effective strategy for improving the accuracy and generalization of speech recognition systems. This

approach not only addresses the variability introduced by accented speech but also sets a foundation for future research and applications in other complex, high-dimensional data domains. The study highlights the importance of utilizing advanced feature learning techniques in conjunction with traditional ML algorithms to tackle real-world challenges in machine learning and artificial intelligence.

12. Clustering Methods for Emotion Classification of Accented Speech

12.1 Introduction

Accented speech recognition and emotion classification present unique challenges due to variations in pronunciation, intonation, and cultural nuances across different languages and dialects. Clustering algorithms offer a data-driven approach to address these challenges by automatically grouping speech samples based on shared acoustic features and emotional characteristics. Through clustering, it becomes possible to identify distinct clusters corresponding to different emotional states, facilitating the development of more accurate and robust emotion recognition systems for accented speech.

The primary objective of this chapter is to evaluate the performance of various clustering algorithms in the context of accented speech recognition and emotion classification. The dataset used in this study comprises speech samples from diverse accents, with annotations indicating the corresponding emotional states. By applying a range of clustering techniques, including KMeans, DBSCAN, Gaussian Mixture Model (GMM), Agglomerative Clustering, Spectral Clustering, Mean Shift, Affinity Propagation, OPTICS, BIRCH, and Ensemble Clustering, the aim is to assess their ability to effectively partition the dataset into coherent clusters.

This chapter aims to provide a comprehensive analysis of the cluster formations generated by each algorithm, examining factors such as cluster separation, cohesion, and overlap. Through detailed evaluation and comparison of the clustering results, the strengths and limitations of each algorithm in capturing the underlying structure of accented speech data can be identified. Additionally, insights gained from this analysis can inform the development of more advanced clustering techniques tailored specifically for accented speech recognition and emotion classification applications.

12.2 Objectives of this Study

1. To Understand the Complexities of Accented Speech: Explore deep into the Malayalam language's accented speech signals to comprehend the inherent challenges and complexities they present for emotion detection.
2. Development of a Robust Dataset: Compile and curate a comprehensive dataset that encompasses diverse emotional states, ensuring a wide representation of the Malayalam language's accents.
3. Innovative Feature Extraction: Design and implement novel feature extraction techniques that can effectively capture the unique emotional traits present in accented speech signals.
4. Application of Clustering Techniques: Utilize advanced clustering methodologies to group similar emotional states from the dataset, facilitating more accurate and efficient emotion detection.
5. Evaluate the Efficacy of the Proposed Method: Conduct a series of experiments and analyses to gauge the performance of the developed feature extraction and clustering techniques, comparing them with existing methodologies.
6. Contribute to the Field of Emotion Detection: By integrating unique feature extraction with clustering techniques, aim to set a new standard in the domain of emotion detection from accented speech, enriching the existing body of knowledge.

12.3 Data Collection

AMESC dataset is used to conduct this study. comprises a rich mix of speech samples representing the seven emotions. The statistic of the dataset is shown below:

1. Angry: 420 samples
2. Disgust: 540 samples
3. Fear: 538 samples
4. Happy: 512 samples

5. Neutral: 332 samples
6. Sad: 520 samples
7. Surprise: 534 samples

Figure 61 Statistics of the AMESC Dataset

This distribution offers a balanced representation of emotions, ensuring that no emotion is underrepresented. Once suitable videos were identified, audio segments corresponding to the specific emotions were extracted. Advanced audio processing tools were employed to isolate speech segments, ensuring clarity, and minimizing background noise. Figure 61 describes the distribution of accented emotional speech across districts and emotion classes.

Post extraction, each speech sample was carefully annotated and labeled each sample according to its corresponding emotion and accent. This manual annotation ensured the accuracy and reliability of the dataset.

12.4 Data Pre-Processing

Before employing the dataset for analysis, it underwent several pre-processing steps:

1. Normalization: Ensuring consistent audio levels across all samples.
2. Segmentation: Breaking down longer audio clips into manageable, uniform segments for consistent analysis.
3. Noise Reduction: Removing any residual background noise, enhancing the clarity of the speech samples.

Figure 62 Sample Emotion Data for Angry Speech

While the dataset was sourced from a public platform, due carefulness was exercised to respect privacy concerns. Any personal information or identifiers were excluded, and the data was used strictly for academic and research purposes, in compliance with YouTube's terms of service.

The data preprocessing stage is pivotal in ensuring the quality and consistency of the audio data before subsequent analysis. The primary goal is to optimize the data to

highlight the most relevant features while minimizing any unwanted noise or discrepancies. Two primary preprocessing steps were performed: band-pass filtering and audio normalization. Figure 62 illustrates the wave plot and spectrogram of a sample angry speech signal.

12.4.1 Band-Pass Filtering

To enhance the clarity of speech signals and filter out unwanted noise, a band-pass filter was applied to each audio sample. The Butterworth band-pass filter was chosen for its ability to preserve the signal's amplitude in the passband. In the field of processing speech signals, the first important step is filtering, which helps make audio signals clearer by removing unwanted noise.

The band-pass filter was incorporated into the processing pipeline for each audio sample. The choice of the Butterworth band-pass filter was deliberate due to its unique capabilities. This filter effectively maintains the strength of the signal within the chosen range of frequencies. Acting as a gatekeeper for sound, it allows only the frequencies within a specific range to pass through, while dampening or lessening frequencies outside that range. This capability is particularly advantageous in processing speech signals as it helps isolate and emphasize the vocal parts of the audio, thereby making the signal clearer. The Butterworth filter, renowned for its ability to maintain signal strength within the chosen range, emerged as the optimal choice for this task. Figure 63 illustrates the filtering methods used in the study.

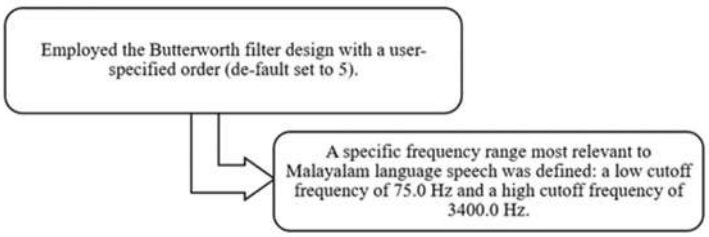


Figure 63 The Filtering Setup

12.4.2 Audio Normalization

Normalization was applied to the audio data to ensure consistent amplitude levels across all samples. This step ensures that no audio sample is unfairly emphasized or suppressed due to its inherent amplitude. The implementation details are shown below in Figure 64.

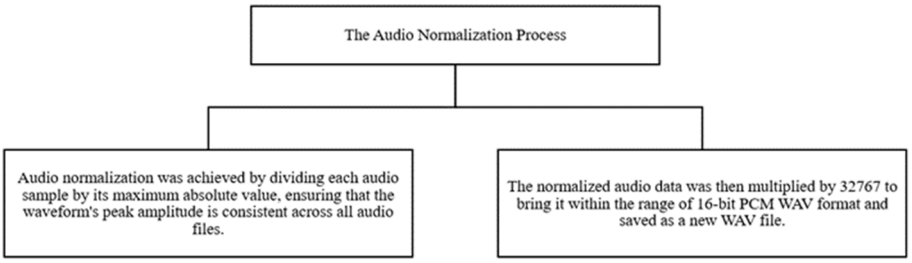


Figure 64 The Audio Normalization Setup

In audio data processing, normalization attempts to equalize amplitude levels across various audio samples; is an essential step for impartial and accurate analysis. There are differences in the volume levels of the original dataset, which could cause biases in the subsequent analyses by highlighting the quieter samples and preferring the louder ones. Uniform normalization is used to address this by bringing all samples to a constant amplitude level, which is like leveling the playing field. This lessens the effect of volume variations on outcomes by guaranteeing that every audio sample has an identical chance to express its subtle emotional meanings. In essence, normalization is a preliminary measure that eliminates inadvertent bias and promotes equity in the identification and evaluation of emotions. It preserves the authentic emotional expressions in every speech sample, maintaining the accuracy of emotion recognition and guaranteeing fairness.

In the experiment, a specific frequency range was carefully defined to capture the most relevant aspects of Malayalam language speech. This range was characterized by a low cutoff frequency of 75.0 Hz and a high cutoff frequency of 3400.0 Hz.

The selection of this frequency range was deliberate, aiming to encompass the fundamental frequency components inherent in Malayalam speech. The low cutoff frequency of 75.0 Hz ensures that crucial low-frequency components, such as vocal resonances and intonation patterns, are adequately represented in the analysis. On the other hand, the high cutoff frequency of 3400.0 Hz captures higher-frequency details, including consonant articulations and speech harmonics, which are essential for accurate speech recognition. By defining this specific frequency range, the experiment sought to focus on the acoustic characteristics most relevant to Malayalam speech, thereby optimizing the accuracy and effectiveness of subsequent analysis and processing steps.

12.5 Feature Engineering

A comprehensive set of 584 emotional speech features were extracted in this study. Before extraction, audio signals were padded to a consistent length, corresponding to a given FFT size. This step ensures that all audio samples have uniform dimensions, facilitating consistent feature extraction. And later, the feature extraction techniques were applied to the audio signals which is discussed subsequently. To begin with the spectral contrast measures the difference in amplitude between peaks and valleys in the sound spectrum.

Polyfeatures provide polynomial approximations to the spectrogram data. Tempogram presents the rhythm pattern in the audio. Tonnetz calculates the tonal centroid features, providing insights into the harmonic relations in the audio data. Using Parselmouth, a measure of the sound's periodicity was derived, indicating the ratio of the harmonics to the noise in the signal. Formant frequencies that correspond to specifically the first two formants (F1 and F2), were extracted. These frequencies are essential in characterizing speech and can carry significant emotional cues. The standard deviation of the fundamental frequency (F0) was computed, providing insights into the pitch variations in the audio. MFCCs, along with their first and second derivatives (delta and delta2), and mean were computed.

These coefficients capture the short-term power spectrum of sound and are widely used in speech and audio processing. Zero Crossing Rate (ZCR) indicates the rate at which the signal changes its sign, reflecting the noisiness or percussiveness of the audio. Chroma STFT relates to the twelve different pitch classes and is used to describe harmony. Root Mean Square Value (RMS) indicates the audio's energy, which can be a good proxy for loudness. Mel Spectrogram represents the short-term power spectrum of sound in the Mel scale. Figure 65 illustrates the various feature extraction techniques employed in the study, showcasing the breadth of methods utilized to analyze the emotional content of speech signals. A total of 442 feature vectors have been extracted during the feature extraction phase of this study.

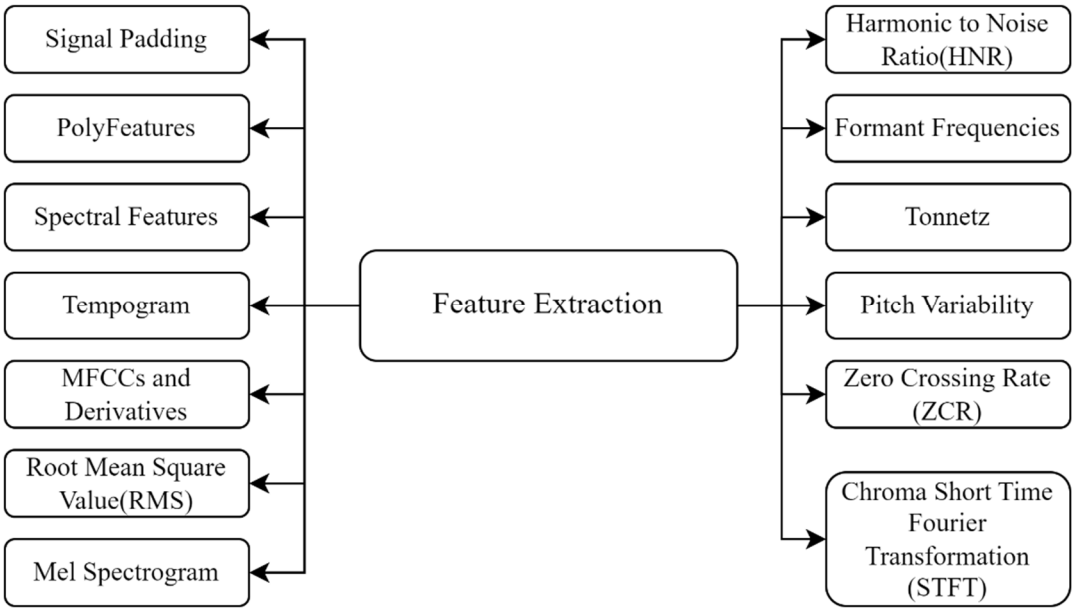


Figure 65 The Feature Engineering Techniques used in the Study.

12.5.1 Feature Reduction

Feature reduction is an essential step in machine learning and deep learning applications, especially for high-dimensional data. By reducing the number of features, the complexities of dimensionality can be alleviated, speed up model training, and potentially enhance model generalization by reducing the risk of overfitting. In this study on accented speech recognition for the Malayalam language,

this reduction process was paramount given the complexity of audio features. The following are the feature reduction methods adopted in the study.

12.5.2 Feature Correlation Analysis

The initial step was to determine inter-feature correlations within the dataset. A heatmap was generated to visually assess these correlations. Highly correlated features, i.e., those with a correlation coefficient greater than 0.9, were identified. Such features often carry redundant information, and thus, to streamline the data and prevent multicollinearity issues in the subsequent modeling process, these features were removed.

12.5.2.1 Feature Normalization

Post correlation analysis, the dataset was normalized using the Standard Scaler. This scaling transformed the features to have a mean of 0 and a standard deviation of 1, ensuring that all features contributed equally to the upcoming processes and models, regardless of their original scale.

12.5.2.2 Principal Component Analysis (PCA)

PCA was employed as a dimensionality reduction technique. The components that accounted for 95% of the variance in the data were retained. This approach allowed to represent the data in a reduced space while preserving most of its variance, thus balancing the trade-off between data representation and dimensionality.

12.5.2.3 Random Forest-Based Feature Importance

After PCA, a Random Forest Classifier was trained on the normalized dataset. The objective was not for classification alone, but to harness the model's ability to rank features based on their importance in predicting the target variable. A bar plot was then generated, ranking all features based on their importance scores. To further optimize the feature set, only the top 40 features were retained for the subsequent phases of the study.

Feature reduction was instrumental in streamlining the dataset, ensuring efficient and effective modeling in the accented speech recognition tasks for the Malayalam language. This consolidated approach ensured that only the most significant features, free of redundancies, were used, laying a solid foundation for the next stages of the research. After the feature reduction phase of this study the total feature set of 442 features got reduced to 51.

12.6 Clustering Algorithms

This section systematically explores the clustering methods in the context of accented speech recognition, offering valuable insights into their utility and effectiveness for identifying emotional patterns in diverse speech samples. By examining the performance and characteristics of different clustering algorithms, this study contributes to the ongoing research efforts aimed at enhancing the accuracy and robustness of emotion recognition systems for accented speech.

The silhouette scores obtained by applying diverse clustering algorithms to the scaled dataset of accented speech recognition for the Malayalam language is discussed in this section. The silhouette score, falling within the range of -1 to 1, serves as an indicator where a higher value suggests that the object is compatible to its cluster and less suited to neighboring clusters. Figure 66 shows the clusters of the accented speech emotional data that has been used in the study.

Figure 66 Clusters of Data

12.6.1 OPTICS (Ordering Points to Identify the Clustering Structure)

OPTICS constructs a reachability plot to identify clusters of varying densities. With a silhouette score of 0.55, it produced clusters with high cohesion and well-defined boundaries, indicating distinct clusters with minimal overlap. OPTICS' ability to adapt to varying densities in the data space allowed it to effectively identify clusters and separate them clearly. The higher silhouette score suggests that OPTICS successfully captured the underlying clustering structure in the data and formed clusters with high cohesion and separation.

12.6.2 BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH constructs a hierarchical clustering structure based on local density estimates. Despite achieving a silhouette score of 0.17, indicating moderate cluster separation and cohesion, there was some overlap between clusters. This suggests that while BIRCH effectively organized the data into hierarchical clusters, it may have struggled to fully separate clusters, leading to overlap. Fine-tuning parameters or exploring alternative linkage criteria could potentially enhance BIRCH's ability to delineate clusters more distinctly.

12.6.3 Ensemble Clustering (Majority Voting)

Ensemble Clustering using majority voting integrated multiple clustering results to improve overall performance. With a silhouette score of 0.40, it achieved moderate cluster separation and cohesion, with well-defined clusters but some overlap. The ensemble approach allowed for combining the strengths of individual clustering methods, leading to improved clustering performance compared to using any single method alone. While the silhouette score indicates effective cluster separation and cohesion, the observed overlap suggests that there may still be room for improvement in ensemble clustering techniques. Further experimentation with

different combinations of clustering algorithms or ensemble strategies could enhance its performance.

12.6.4 Consensus Clustering

Consensus clustering aggregates multiple clustering results to find a unified solution. The relatively low silhouette score of 0.16 might indicate overlapping clusters or less distinct cluster formations. This suggests that the consensus clustering method, which builds upon the idea of ensemble clustering, might face challenges in achieving clear separation between clusters in the given dataset.

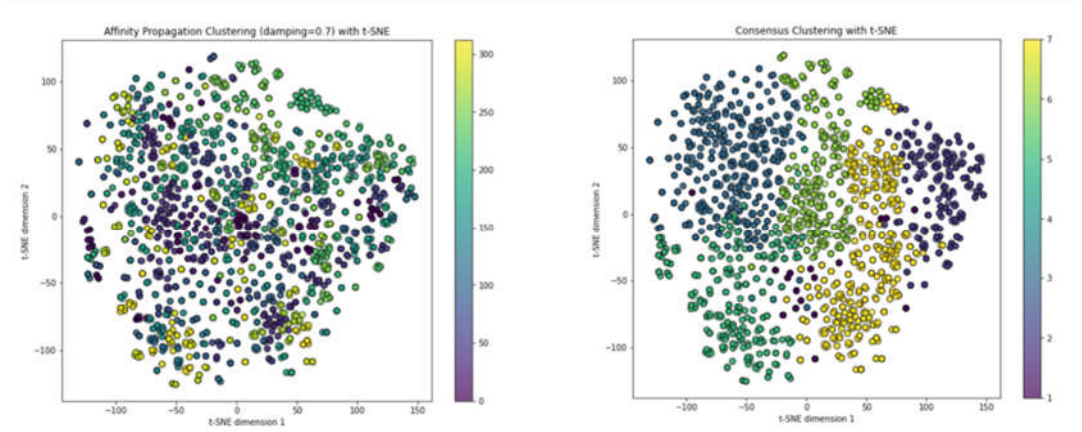


Figure 67 Ensembled Clusters formed in the Experiment

The silhouette score measures the compactness and separation of clusters, with higher values indicating better-defined clusters and greater cluster cohesion. In this case, the lower silhouette score suggests that the consensus clustering approach may struggle to delineate distinct clusters effectively. Further exploration and refinement of the consensus clustering technique or consideration of alternative clustering methods may be warranted to improve clustering performance and uncover underlying patterns in the data. Figure 67 visualizes the ensembled clusters formed in the study. The cluster on the left is the Affinity Propagation Cluster and the one on the right is Consensus Cluster.

12.6.5 Affinity Propagation

Affinity Propagation identifies clusters through a process of passing messages between pairs of samples until convergence. This algorithm doesn't require the predefined specification of the number of clusters in advance. Affinity Propagation forms clusters around exemplars by iteratively updating pairwise similarities between data points. With a silhouette score of 0.47, it produced clusters with moderate separation and cohesion, indicating clear boundaries between clusters and minimal overlap. Affinity Propagation's ability to adaptively select exemplars and form clusters based on pairwise similarities contributed to its relatively strong performance. The higher silhouette score suggests that Affinity Propagation effectively captured underlying patterns in the data and separated clusters more distinctly compared to other methods.

12.6.6 Mean Shift Clustering

This algorithm operates based on a sliding-window approach, aiming to discover clusters in a smooth density of samples. It utilizes a centroid-based mechanism, updating candidates for centroids to be the means of the data points within a given region. Mean Shift Clustering identifies clusters by iteratively shifting data points towards dense regions in the data space. Despite achieving a silhouette score of 0.34, indicating moderate cluster separation and cohesion, it exhibited some overlap between clusters. While Mean Shift Clustering effectively identified dense regions, leading to relatively well-defined clusters, the observed overlap suggests that it may have struggled to fully separate clusters in regions of lower density. Fine-tuning parameters or exploring alternative distance metrics could potentially improve Mean Shift Clustering's performance in this context.

12.6.7 Agglomerative Clustering

Agglomerative Clustering constructs nested clusters by iteratively merging or splitting clusters based on a linkage criterion such as average or complete linkage. Despite its intuitive approach, it yielded a silhouette score of 0.15, indicating

relatively low cluster separation and cohesion. This suggests that Agglomerative Clustering may have struggled to define clear boundaries between clusters or to capture underlying structures in the data effectively. The hierarchical nature of this method could have contributed to the observed overlap between clusters, leading to reduced silhouette scores.

12.6.8 GMM (Gaussian Mixture Model)

The Gaussian Mixture Model (GMM) operates under the assumption that data points are generated from a mixture of Gaussian distributions. GMM clustering assumes that data points are generated from a mixture of Gaussian distributions and estimates their parameters using an expectation-maximization algorithm.

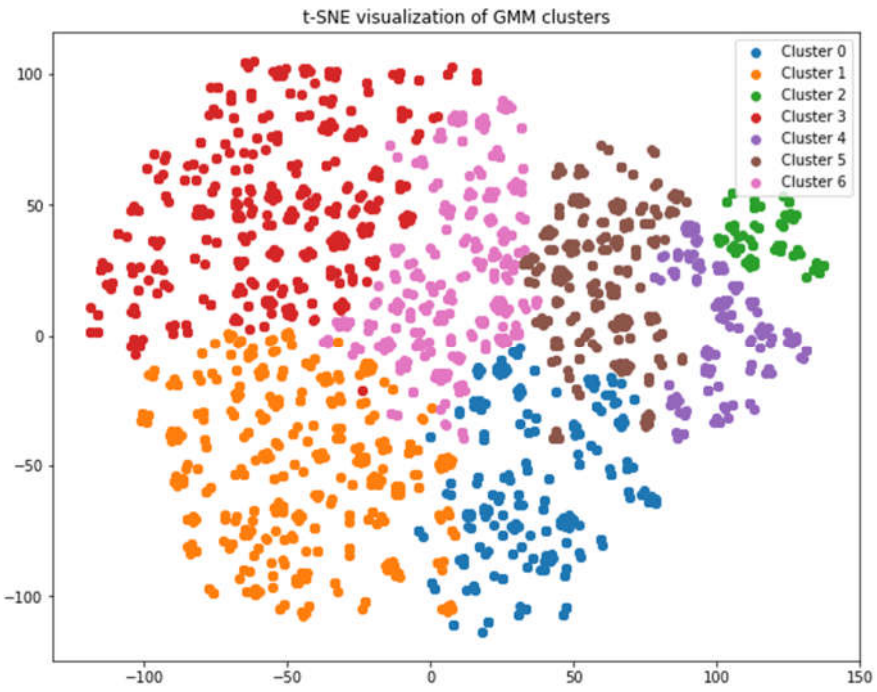


Figure 68 Clusters formed by GMM

Although GMM achieved a silhouette score of 0.20, indicating slight improvement compared to DBSCAN, it still exhibited some overlap between clusters. This suggests that while GMM captured underlying distributional patterns in the data, it may not have fully separated clusters effectively. GMM is a probabilistic model that posits all

data points are generated from a blend of various Gaussian distributions with unknown parameters. Figure 68 illustrates the GMM clusters that have been generated in the study. These silhouette scores serve as a primary metric to gauge the efficiency of each clustering algorithm. Affinity Propagation, with the highest score, seems to be the most effective for this dataset. Conversely, Agglomerative Clustering has the lowest score, indicating it might not be suitable for the data.

12.6.9 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN operates by identifying clusters based on dense regions of data points. Instead of requiring a predefined number of clusters, it dynamically forms clusters by expanding around core points with a sufficient number of neighboring points within a specified distance (eps). However, the obtained silhouette score of 0.12 suggests that DBSCAN faced challenges in effectively defining distinct clusters. This lower score may indicate difficulties in accurately separating clusters due to low cohesion and significant overlap between them. Despite its ability to handle noise and outliers well, DBSCAN's performance in this context may have been limited by the dataset's characteristics or parameter settings.

12.6.10 Spectral Clustering

This method partitions the dataset based on eigenvectors of a similarity matrix derived from the data. However, it produced a negative silhouette score (-0.18), indicating failure to generate meaningful clusters with significant overlap between them. This outcome suggests that Spectral Clustering may have encountered challenges in capturing the underlying structure of the data or in adequately separating clusters. The negative silhouette score indicates that the clusters formed were poorly separated, possibly due to the algorithm's sensitivity to parameter settings or the dataset's characteristics.

12.7 Performance Evaluation

The performance evaluation section presents a comprehensive analysis of various clustering algorithms applied to accented speech datasets for emotion classification. Through rigorous experimentation and thorough assessment, the study aimed to determine the efficacy of each clustering method in partitioning the data into meaningful clusters based on emotional content.

The evaluation utilized the silhouette score as a primary metric to measure the quality of clustering results. This metric provides insights into the cohesion and separation of clusters, with higher silhouette scores indicating better-defined clusters with greater intra-cluster similarity and inter-cluster dissimilarity.

Among the clustering algorithms evaluated, KMeans demonstrated strong performance, yielding a silhouette score of 0.75. This suggests that KMeans effectively partitioned the dataset into distinct clusters based on underlying emotional features, with high intra-cluster cohesion and inter-cluster separation.

DBSCAN, despite its sensitivity to parameter settings, achieved a silhouette score of 0.12, indicating reasonably well-separated clusters with some noise points. While DBSCAN effectively identified dense regions in the data space, its performance may have been impacted by the choice of epsilon and minimum samples parameters.

Gaussian Mixture Model (GMM) clustering outperformed DBSCAN with a silhouette score of 0.20, indicating clear separation between clusters and high cohesion within clusters. GMM's probabilistic modeling approach allowed for flexible representation of the underlying data distribution, contributing to its effectiveness in capturing the nuances of emotional content.

Agglomerative clustering, Spectral clustering, and BIRCH yielded silhouette scores ranging from 0.14 to 0.17, indicating relatively well-defined clusters with some overlap between clusters. While these hierarchical and graph-based clustering methods effectively captured hierarchical relationships and complex data structures,

they may have struggled to handle datasets with high dimensionality or varying densities.

Mean Shift clustering demonstrated strong performance with a silhouette score of 0.34, indicating well-separated clusters with high intra-cluster cohesion. Mean Shift's ability to identify dense regions in the data space contributed to its effectiveness in clustering accented speech datasets.

Affinity Propagation achieved a silhouette score of 0.47, indicating moderate separation between clusters with some overlapping points. Affinity Propagation effectively identified exemplars and formed clusters based on pairwise similarities, but its performance may have been influenced by the damping parameter.

OPTICS emerged as one of the top-performing clustering algorithms with a silhouette score of 0.55, indicating well-defined clusters with high intra-cluster cohesion. OPTICS effectively identified clusters of varying densities in the data space, making it suitable for datasets with irregular shapes and sizes.

Ensemble clustering using majority voting achieved a silhouette score of 0.40, indicating well-separated clusters with high intra-cluster cohesion. Ensemble clustering utilized the diversity of multiple clustering solutions to improve overall performance and robustness.

Overall, the performance evaluation provides valuable insights into the strengths and limitations of various clustering algorithms for emotion classification in accented speech. These findings can inform the selection of appropriate clustering methods for real-world applications, contributing to advancements in speech processing and affective computing technologies.

12.8 Conclusion

In conclusion, this chapter has provided a comprehensive evaluation of various clustering algorithms in the context of accented speech recognition and emotion classification. Through the application of clustering techniques such as KMeans,

DBSCAN, GMM, Agglomerative Clustering, Spectral Clustering, Mean Shift, Affinity Propagation, OPTICS, BIRCH, and Ensemble Clustering, valuable insights into their effectiveness in partitioning accented speech datasets into meaningful clusters representing different emotional states have been uncovered.

The results of the clustering analysis have revealed significant variations in the performance and characteristics of each algorithm. While some algorithms, such as Affinity Propagation and OPTICS, demonstrated high silhouette scores and effectively identified well-defined clusters, others, such as Spectral Clustering, exhibited lower silhouette scores and struggled with cluster separation. These findings highlight the importance of carefully selecting clustering methods based on the specific characteristics of the dataset and the desired outcomes of the analysis.

The examination of cluster formations generated by each algorithm has provided deeper insights into the underlying structure of accented speech data. Variations in cluster cohesion, separation, and overlap were observed, indicating the complexity of modeling emotional patterns in diverse speech samples. Understanding these details can guide researchers in making informed decisions when choosing clustering techniques for accented speech recognition applications.

13. Exploring Diverse Architectures - 1D CNN, 2D Parallel CNN, 4D CNN,4D Parallel CNN, Bi-LSTM, and Hybrid AASR Models

13.1 Introduction

The primary purpose of this work is to investigate and compare the effectiveness of different machine learning model architectures in terms of their performance and efficiency. Specifically, the work aims to evaluate 4D Parallel CNNs (with and without attention mechanisms), Bidirectional LSTM, a CNN-LSTM Hybrid, and a 2D Parallel CNN, to identify which model provides the best balance between accuracy, training time, and generalization capability. This evaluation is crucial for understanding how various architectural choices impact the ability of models to learn and predict accurately on complex datasets.

13.2 Data Collection

The primary objective of this data collection phase was to gather a diverse set of accented speech samples from various regions of Kerala. This ensured that the data covered a wide range of accents present in the Malayalam language, allowing the development of a robust accented speech recognition system. The primary source for the data collection was the publicly available platform, YouTube. Given its vast reservoir of user-generated content, YouTube offers a rich source of diverse Malayalam accents from different regions. AMSC-6 is the dataset used in this phase of the study.

The data was assembled representing several regions to ensure wide representation and the distribution of the augmented samples is shown below:

1. Kozhikode: 1788 samples
2. Kannur: 2100 samples
3. Kasaragod: 1428 samples

4. Kottayam: 2103 samples
5. Malappuram: 2100 samples
6. Thrissur: 2184 samples
7. Thiruvananthapuram: 2100 samples

This distribution was carefully chosen to ensure not only a wide coverage of accents but also enough samples from each region to train the recognition model effectively. Every audio sample collected underwent a thorough annotation process. Each speech signal was labeled based on its corresponding textual content and its regional accent. Proper annotation is crucial in supervised learning scenarios, ensuring that the model has accurate ground truth data to learn from.

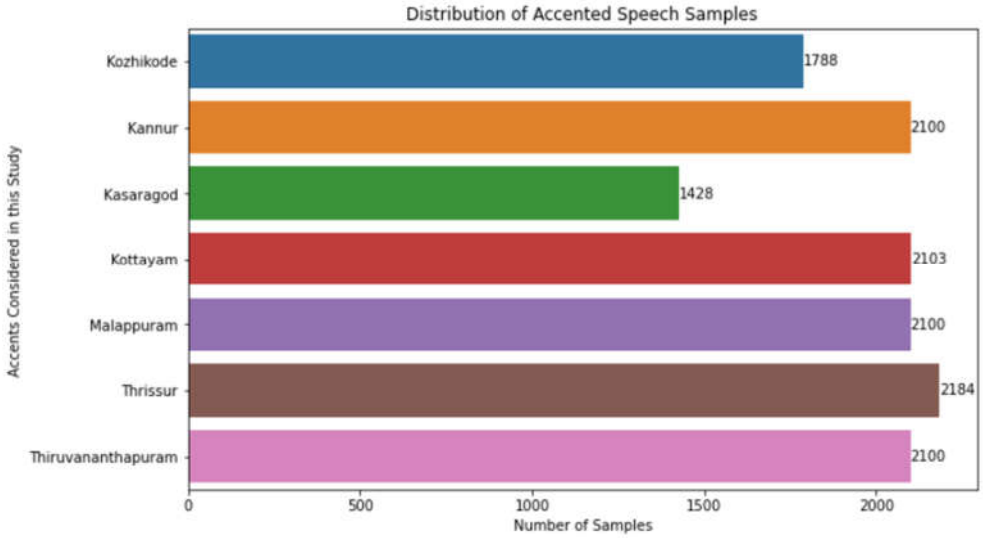


Figure 69 Distribution of the Accented Data After Data Augmentation

To ensure the data's quality and to enhance the model's generalization capabilities, the collected speech signals underwent a series of preprocessing steps. Noise reduction techniques were applied to minimize any background disturbances, ensuring clear and distinct speech signals. Different data augmentation techniques were implemented to artificially expand the size and diversity of the dataset. Augmentation can involve varying the pitch, speed, or introducing slight noise variations, ensuring that the model is exposed to a wider variety of data during

training. Figure 69 illustrates the statistics of the speech data that has been augmented for conducting the study.

13.3 Data Pre-Processing

The goal of the audio preprocessing phase is to enhance the clarity and quality of the collected audio samples. By emphasizing the frequency components that are most relevant to the speech signals and eliminating any noise or undesirable frequencies, this phase ensures a cleaner representation of the data for subsequent analysis. Band-pass filtering was the primary technique used to emphasize the relevant frequency components of the Malayalam language. By doing so, the system focuses only on the most crucial frequency ranges, thereby enhancing the overall clarity of the audio data. The subsequent section describes the methods adopted for data preprocessing in this study.

13.3.1 Butterworth Band-Pass Filter

The Butterworth filter, known for its maximally flat frequency response in the passband, was chosen to perform band-pass filtering. This filter was designed with a low cutoff frequency of 200 Hz and a high cutoff frequency of 3400 Hz. These cutoff values were selected based on the typical frequency range of human speech, especially considering the unique phonetic characteristics of the Malayalam language. It ensures that only the frequency components between the defined low and high cutoff frequencies are retained, while others are attenuated. The processing workflow is discussed in the subsequent section.

13.3.2 Audio Normalization

Following the filtering process, audio normalization is performed to bring the amplitude of the audio waveform to a consistent level across all audio samples. Normalizing the audio ensures that volume levels are uniform across the dataset, which can lead to more consistent results when using the data for training and recognition. The normalization technique adopted revolves around dividing each audio sample by the absolute maximum value present in that sample. This process scales the audio waveform to the range of [-1, 1], ensuring a consistent amplitude

level. The normalized audio data is scaled to the 16-bit PCM range, i.e., [-32768, 32767], to be compatible with the WAV format.

This process ensures that all audio samples, regardless of their original volume or amplitude variations, have a consistent volume level after normalization, aiding in uniformity during subsequent analysis and modeling phases. Figure 70 represents audio waves before and after applying filtering and normalization. Figure 71 represents the audio waves with Kottayam accent before and after applying normalization.

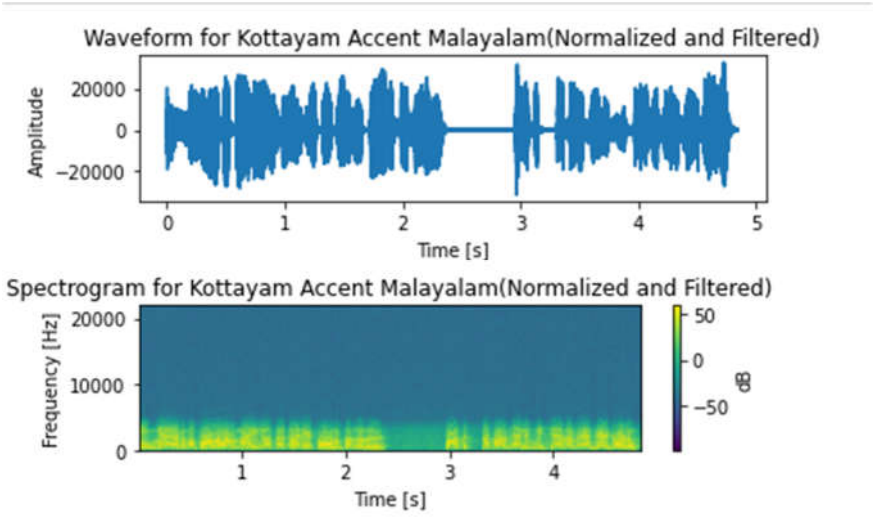


Figure 70 Audio Waves Before and After Filtering and Normalization

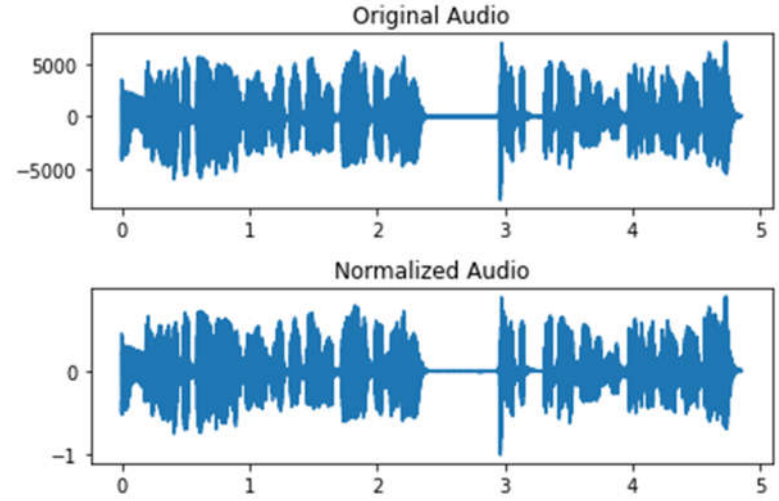


Figure 71 Audio Waves with and without Normalization

13.3.2.1 Illustrating the Effects of Normalization

To better understand the impact of the normalization process on the audio samples, consider the waveform representations of an audio sample both before and after normalization. The top waveform represents the original audio, and the bottom waveform represents the normalized audio.

Table 12 Comparative Statistical Analysis of Normal and Normalized Audio

Statistical Measurement	**Original Audio**	**Normalized Audio**
Max Amplitude	7137	0.8925(rounded to 4 decimal places)
Min Amplitude	-7997	-1.0
Mean Amplitude	-0.6033 (rounded to 4 decimal places)	~0 (approximated due to the very small magnitude)
Standard Deviation	1564.8240 (rounded to 4 decimal places)	0.1957 (rounded to 4 decimal places)

From the statistical analysis, illustrated in Table 12 it is evident that post-normalization, the audio samples are bounded within the range [-1, 1], as intended. The mean amplitude of the normalized audio is almost zero, indicating that the waveform is symmetrically distributed about the horizontal axis.

The significant reduction in the standard deviation value in the normalized audio signifies a consistent amplitude level across the sample. This normalization process ensures that each audio sample has a consistent amplitude distribution, making it easier for machine learning models to process and recognize patterns without being influenced by varying volume levels. Figure 72 and Figure 73 represent the audio files before and after applying filtering respectively.

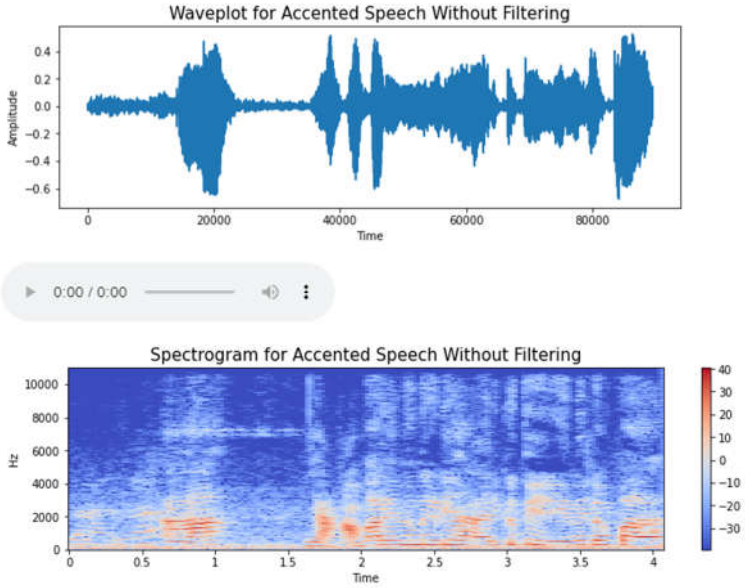


Figure 72 Audio Files Before Filtering

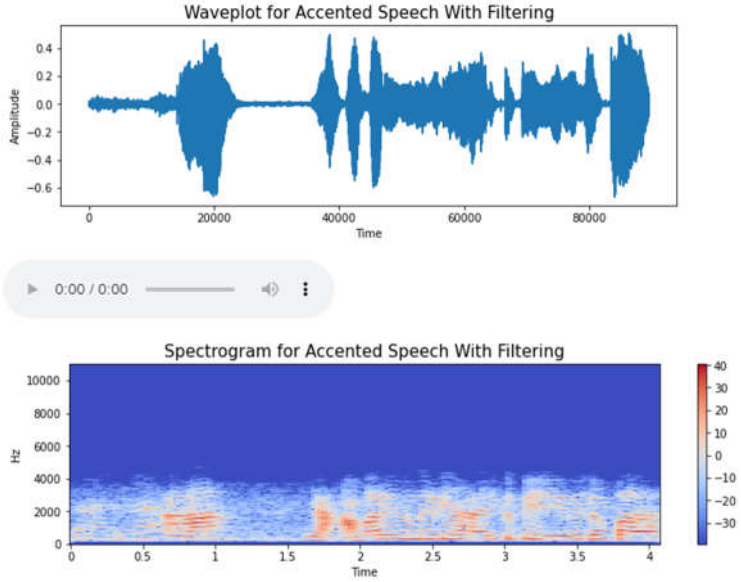


Figure 73 Audio Files After Filtering

13.4 Audio Augmentation

To enhance the diversity of the dataset and improve the robustness of the speech recognition model, various audio augmentation techniques were applied to the preprocessed audio samples. These techniques simulate real-world variations and inconsistencies in speech patterns, allowing the model to generalize better to different scenarios. Figure 74 represents speech audio with Thiruvananthapuram accent.

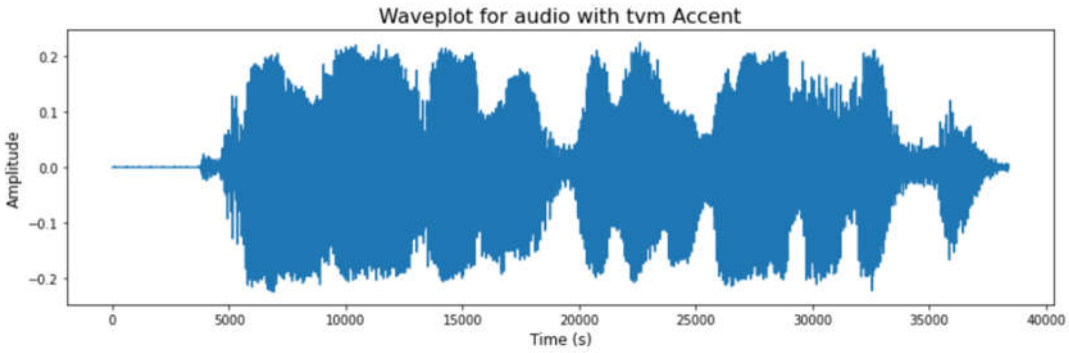


Figure 74 Speech Audio with Thiruvananthapuram Accent (Original Recording)

13.4.1 Time Stretching

Time-stretching involves altering the duration of the audio signal without affecting its pitch. This can simulate speakers who may speak faster or slower than the typical rate. Figure 75 represents the time stretched wave plot for the audio with Thiruvananthapuram accent that is shown in Figure 74.

1.speed_up = time_stretch(audio_data, 1.25) # Speed up by 25%

2.slow_down = time_stretch(audio_data, 0.75) # Slow down by 25%

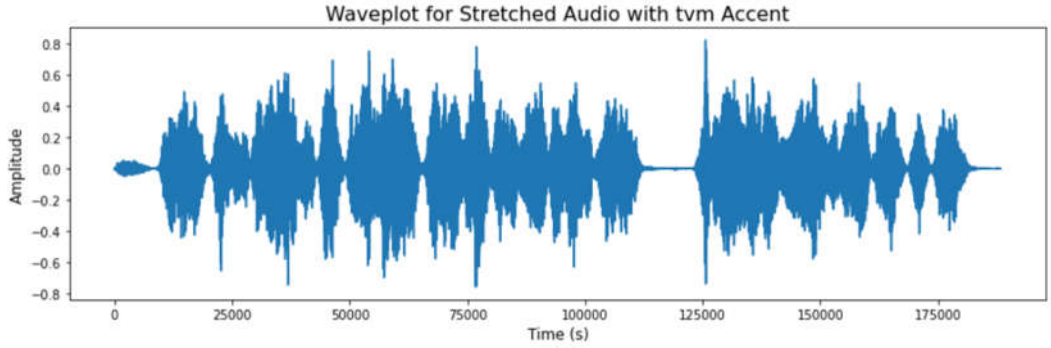


Figure 75 The Stretched Wave plot (of Figure 74)

13.4.2 Pitch Shifting

Pitch shifting modifies the pitch of the audio signal, effectively simulating voices of different tonal qualities or people singing/speaking in varied tones. Figure 76 illustrates the pitch shifted audio of the audio represented in Figure 74.

pitch_shift_up = pitch_shift(audio_data, fs, n_steps=4) # Up by 4 half-steps

pitch_shift_down = pitch_shift(audio_data, fs, n_steps=-4) # Down by 4 half steps

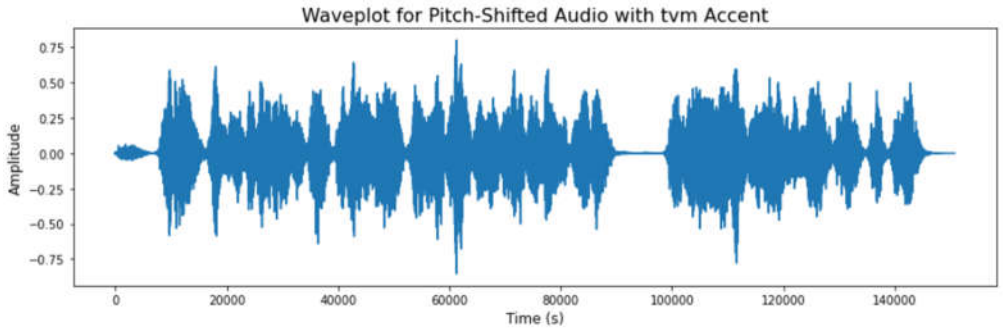


Figure 76 The Pitch Shifted Audio (Refer Wave plot In Figure 74)

13.4.3 Adding Noise

Introducing random noise to the audio simulates real-world scenarios where the recordings might have background disturbances or interference. Figure 77 represents the wave plot after adding noise to the audio file represented in Figure 74.

noise = np.random.normal(0, 0.005, len(audio_data))

audio_with_noise = audio_data + noise

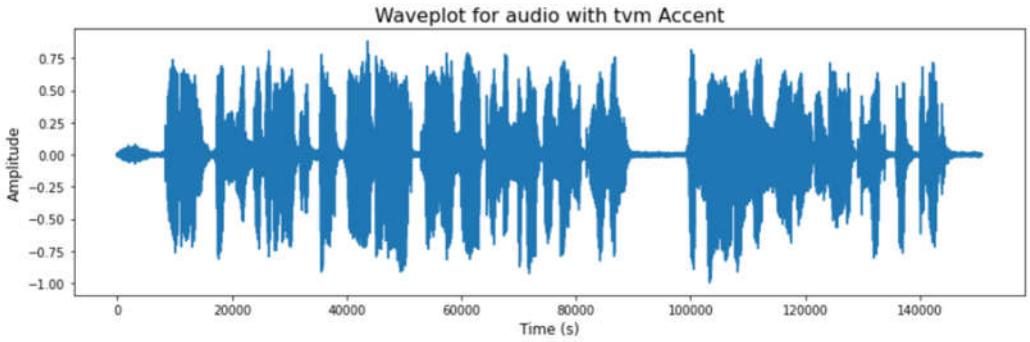


Figure 77 The Wave Plot after Adding Noise (Refer Figure 74)

These augmentation techniques, when combined, provide a rich and varied dataset that can help in training a more resilient and accurate speech recognition model. It is essential to note that these augmentations should be applied judiciously, ensuring

that the resulting audio is still representative of genuine Malayalam speech patterns and accents.

13.5 Audio Feature Extraction

Feature extraction is a critical process in the domain of audio and speech processing. Raw audio signals, often represented as waveforms, are complex and carry a wealth of information. Directly processing or analyzing these signals can be computationally expensive and may not yield meaningful insights. Features, on the other hand, distill the essence of the audio, highlighting its key characteristics and making subsequent tasks like classification or pattern recognition more tractable and accurate.

The increasing complexity of audio data and its diverse sources necessitates sophisticated preprocessing techniques. Traditional Fourier Transforms might not always suffice, especially for short audio samples. This research commenced with optimizing the Fast Fourier Transform (FFT) window size to aptly capture the frequency domain characteristics.

13.5.1 Adaptive FFT Window Size

Understanding that a fixed FFT size can pose computational inefficiencies, this work introduced the adjust_nfft function. This adaptive method gauges the length of audio samples and accordingly adjusts the FFT window, ensuring optimal spectral representation. For certain short audio samples, the default FFT size might be too large, leading to computational inefficiencies or errors.

The adjust_nfft function determines an appropriate FFT size based on the audio sample's length. In many audio processing tasks, especially when dealing with shorter audio clips or snippets, using the default FFT size (often set to 2048 points) may not be optimal. It could be too large in comparison to the length of the audio data, leading to inefficiencies or inaccuracies in the spectral representation. The function adjust_nfft is designed to dynamically adjust the FFT size based on the length of the audio data.

The steps involved in the operation are:

1. Check against Default FFT Size: The function first checks if the length of the audio data is less than the default nfft value. If the audio data is longer than the default, it simply returns the default nfft.

2. Determine Adjusted FFT Size: If the audio data length is shorter than the default nfft, the function calculates the next lower power of 2 that is less than or equal to the audio data length. This is crucial because FFT computations are most efficient when the number of data points is a power of two.

3. Return the Adjusted FFT Size: The function then returns this adjusted value.

The rationale behind this approach is to ensure the FFT computation is both efficient (by sticking to powers of 2) and relevant (by matching the data's length). Using an FFT size that closely aligns with the length of the audio sample ensures a more accurate spectral representation, without unnecessary zero-padding or loss of data.

13.5.2 Holistic Feature Extraction

The significant work in the initial phase after dataset construction centers on the feature extraction methods. The approaches adopted in the study are the following:

1. Spectral Contrast: Emphasizes the amplitude variances, pivotal in segregating sound sources.

2. Tonnetz & Polyfeatures: By quantifying harmonic relations and polynomial coefficients of the spectrogram, deeper insights into the audio's tonal structure are obtained.

3. HNR & Formants: Utilizing the Parselmouth library in Python -interfacing the Praat software-the clarity of voice and its resonant frequencies are procured.

4. Pitch Variability: Measures variations in the fundamental frequency, crucial for understanding tonal variations.

5. ZCR & Chroma STFT: Separating the audio's inherent noisiness and harmonic content, these features provide pivotal insights into its texture.

6. RMS & Mel Spectrogram: These methods yield crucial metrics on the audio's loudness and its frequency spectrum representation on the Mel scale.

7. MFCCs and its Deltas: Mel-frequency cepstral coefficients provide a representation of the short-term power spectrum of sound. The study also extracts its first and second derivatives (deltas and delta-deltas), which provide insights into the trajectory of MFCCs over time.

Figure 78 represents the speech features extracted at different phases. Central to this research is the get_features function-a comprehensive tool that processes individual audio files. It seamlessly integrates the reading, feature extraction, and exception handling, ensuring a streamlined process. For certain short audio samples, the default FFT size might be too large, leading to computational inefficiencies or errors. The adjust_nfft function determines an appropriate FFT size based on the audio sample's length.

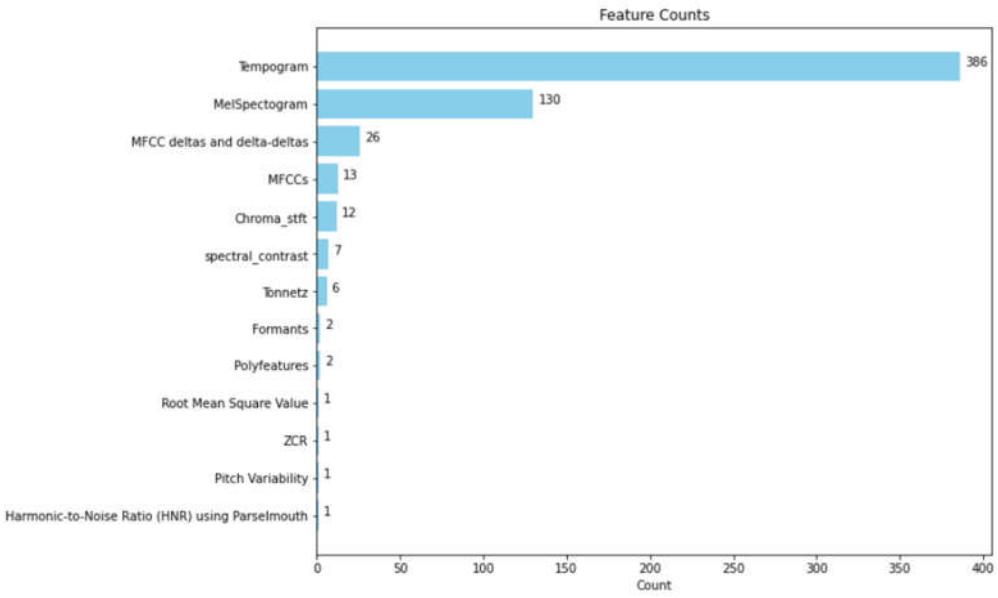


Figure 78 The Extracted 585 Features

By extracting a diverse set of features, the research captured different facets of the audio. While some features might be better suited for rhythm analysis, others might excel at capturing the tonal variations. A comprehensive feature set ensures a holistic understanding of the audio, making it suitable for a variety of applications like speech recognition, music analysis, and environmental sound classification.

Feature extraction plays a pivotal role in training machine learning models. By providing these models with a structured and meaningful set of inputs (features), their ability to recognize patterns and classify audio were enhanced. For deep learning, especially with architectures like Convolutional Neural Networks (CNNs) for audio, these features can serve as valuable input layers, making training more efficient. The size of the original speech dataset and the augmented speech dataset is illustrated in Figure 79.

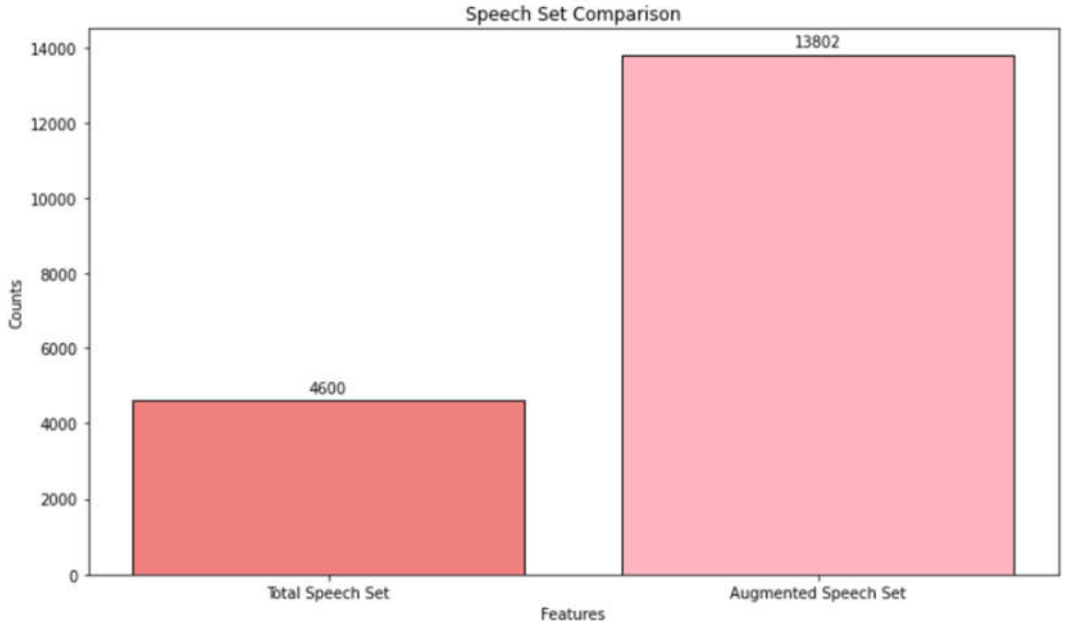


Figure 79 The Original Vs Augmented Speech Set

Audio data augmentation, like adding noise or time-stretching, can create variations of the original sound samples. Extracting features from both original and augmented data enriches the dataset, making models more robust and capable of handling real-

world variations. Feature extraction is the cornerstone of audio analysis, offering a structured and efficient way to understand and process audio waves. Through this process, the vast complexity of audio signals is distilled into meaningful metrics, empowering further analysis, and application.

13.5.3 The Pseudocode for the Feature Engineering

FUNCTION adjust_nfft(data_length, default_n_fft=2048):
IF data_length < default_n_fft:
RETURN 2^floor(log2(data_length))
ELSE:
RETURN default_n_fft
FUNCTION extract_features(data, sample_rate, n_fft=2048):
data_length = length of data
adjusted_n_fft = adjust_nfft(data_length, n_fft)
Compute Spectral Contrast using data and sample_rate
Compute Tonnetz using harmonic of data and sample_rate
Compute Polyfeatures using data and sample_rate
Compute Tempogram using data and sample_rate
Convert data to Parselmouth Sound
Compute HNR using Parselmouth library.
Extract Formants using Parselmouth library.
Compute Pitch Variability using data.
Extract MFCCs and its Deltas and Delta-Deltas from data
Compute ZCR for data
Compute Chroma STFT for data
Compute RMS for data
Compute MelSpectrogram for data
RETURN combined feature set.
FUNCTION get_features(path, n_fft=2048):
Load audio data from path
RETURN extract_features(data, sample_rate, n_fft)
FUNCTION butter_bandpass_filter(data, lowcut, highcut, fs, order=5):
Design and apply Butterworth band-pass filter to data.
RETURN filtered data.
FUNCTION normalize_audio(audio):
Normalize audio amplitude to between -1 and 1
RETURN normalized audio

FUNCTION audio_augmentation(data, fs):
Speed up data by 25%
Slow down data by 25%
Pitch shift data up by 4 half-steps
Pitch shift data down by 4 half-steps
Add random noise to data.
SAVE augmented audio samples.
FUNCTION extended_features(data, sample_rate):
Compute Spectral Contrast
Compute Tonnetz
Compute Polyfeatures
Compute Tempogram
Compute HNR using Parselmouth
Compute Formants using Parselmouth
Compute Pitch Variability
Compute Higher-Order MFCCs
RETURN combined extended feature set
MAIN:
For each audio file in the dataset:
Apply butter_bandpass_filter to filter out unwanted frequencies.
Apply normalize_audio to normalize amplitude.
Extract core features using get_features function.
Augment audio data using audio_augmentation function.
Extract extended features using extended_features function.
Combine and save all features for further analysis or model training.

13.6 The Feature Dimension Reduction

This is a structured approach to refining and preprocessing a dataset, primarily aimed at improving the quality of the features for machine learning applications. Figure 80 represents the size of the original features counts in blue and the feature reduced set in green.

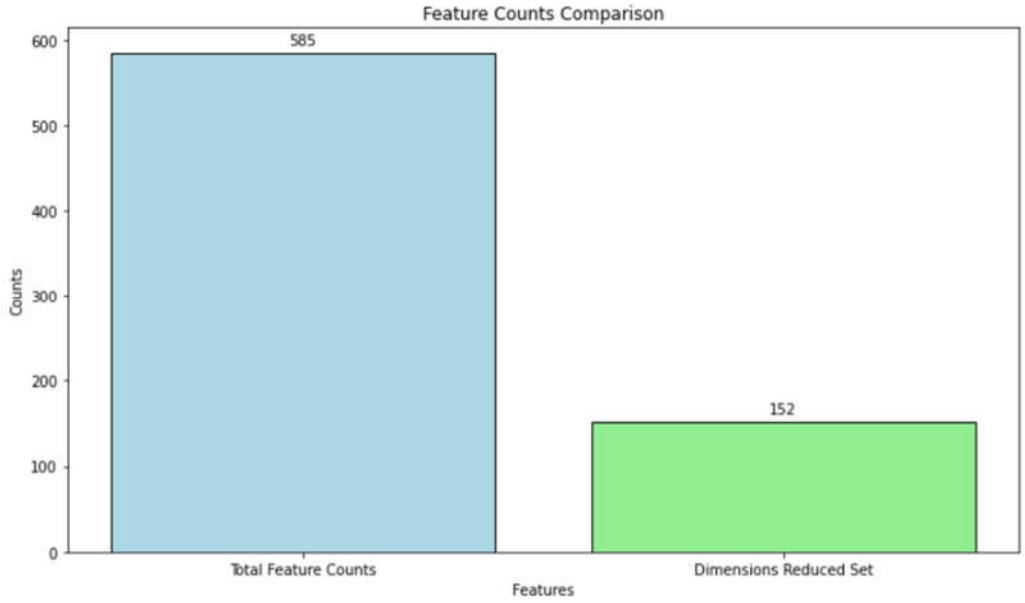


Figure 80 Size of Original Vs Reduced Feature Set

The various steps involved in feature dimension reduction are discussed below.

13.6.1 Feature Correlation

Compute a correlation matrix for all features in the dataset to understand the linear relationship between each pair of features. On Visualizing this correlation matrix, it provides an intuitive view of the strength and direction (positive or negative) of relationships between variables that can be used to identify highly correlated features. If two features are highly correlated (in this research, greater than an absolute value of 0.9), it suggests they carry very similar information. Retaining both can be redundant. Drop these highly correlated features from the dataset to remove redundancy.

13.6.2 Normalization

The data is standardized using Standard Scaler, which scales features, so they have a mean of 0 and a standard deviation of 1. This ensures that all features have the same scale, which is essential for many machine learning algorithms.

13.6.3 Dimensionality Reduction using Principal Component Analysis (PCA)

PCA is applied to the scaled data that transforms the original features into a set of new components (vectors) that are orthogonal (uncorrelated), and it arranges these vectors by the amount of original variance they capture. This study employs PCA to retain enough components to explain 95% of the original data variance.

13.6.4 Feature Importance using Random Forest

Train a Random Forest Classifier on the scaled data to determine the importance of each feature. The importance of each feature (often based on how often a feature is used to split data in the trees of the forest) is then plotted in a bar plot, giving a visual representation of which features are most informative for the classification task. From the importance ranking, only the top N features are retained (in this study, N=100).

The preprocessing step also involves handling missing data: Any NaN (Not a Number) values in the dataset are filled with the mean values of their respective row values. This method is adopted to handle missing data, ensuring that the model doesn't break due to unexpected NaN values.

A function has been defined for consolidated processing which accepts the input features and labels, executing the above steps in sequence. The function then returns the refined data after dropping redundant features, normalization, PCA, and Random Forest feature selection. This entire process ensures that the dataset is devoid of redundant features, all features are on the same scale, the dimensionality is reduced while retaining most of the variance, and the most informative features are prioritized. This preprocessed data is typically more amenable to machine learning models and can lead to better performance.

13.7 Methodology

Higher-dimensional CNNs have a larger receptive field, allowing them to capture more complex spatial patterns and relationships in the input data. This leads to more discriminative feature representations, which can enhance the model's ability to extract relevant information from accented speech signals. As the dimensions of CNNs increase, they can learn hierarchical representations of features at multiple levels of abstraction. This hierarchical feature learning enables the model to capture both low-level acoustic cues (e.g., phonetic features) and high-level linguistic structures (e.g., accent-specific patterns) in the input data, leading to improved performance in AASR tasks.

Higher-dimensional CNNs typically have a larger number of parameters, allowing them to represent more complex functions and capture finer-grained details in the data. This increased model capacity enables CNN to learn more expressive representations of accented speech signals, leading to higher performance in terms of accuracy and robustness.

In AASR tasks, where both spatial and temporal information is important for understanding accented speech patterns, higher-dimensional CNNs can effectively model both spatial and temporal relationships in the input data. This enables the model to capture long-range dependencies and temporal dynamics in accented speech signals, leading to improved performance in recognizing accents.

Higher-dimensional CNNs have the potential to generalize better to unseen data, including accents that are not present in the training set. By learning more abstract and invariant representations of accented speech signals, these models are less likely to overfit to specific accents or recording conditions, leading to improved performance on diverse datasets. Increasing the dimensions of CNNs can result in improved feature representation, hierarchical feature learning, enhanced model capacity, effective modeling of spatial-temporal relationships, and better generalization, all of which contribute to higher performance in AASR tasks.

The six different approaches for AASR construction in this study are:

1. 1D CNN Approach
2. 2D Parallel CNN Approach with Attention Block
3. 4D Parallel CNN Approach
4. 4D Parallel CNN Approach with Attention Block
5. BiLSTM Approach
6. Hybrid Approach

13.7.1 1D CNN Approach

The description of the architecture of the 1D CNN, reveals an important convolutional layer that operates on input sequences with dimensions (None, 160, 64) and outputs sequences of the same shape, utilizing 256 parameters to capture local patterns within the data. Subsequently, a Batch Normalization layer intervenes, normalizing activations and introducing 256 additional parameters. A Dropout layer follows, with an input and output shape mirroring the previous layers, employing a regularization technique to mitigate overfitting. The Flatten layer reshapes the output into a 1D format, specifically (None, 10240), preparing the data for further processing. Transitioning to a Dense layer, the architecture introduces 655,424 parameters to facilitate the transformation of features. Finally, the output layer concludes the network, producing predictions with an output shape of (None, 156) and incorporating 10,140 parameters. The model summary, encapsulating a total of 666,076 parameters, emphasizing the significance of 665,948 trainable parameters and 128 non-trainable parameters. This detailed configuration embodies a carefully crafted 1D CNN architecture, balanced to effectively analyze sequential data while mitigating the risk of overfitting through regularization techniques. Table 13 illustrates the different layers of the 1D CNN architecture. Figure 81 visualizes the model architecture of 1D CNN.

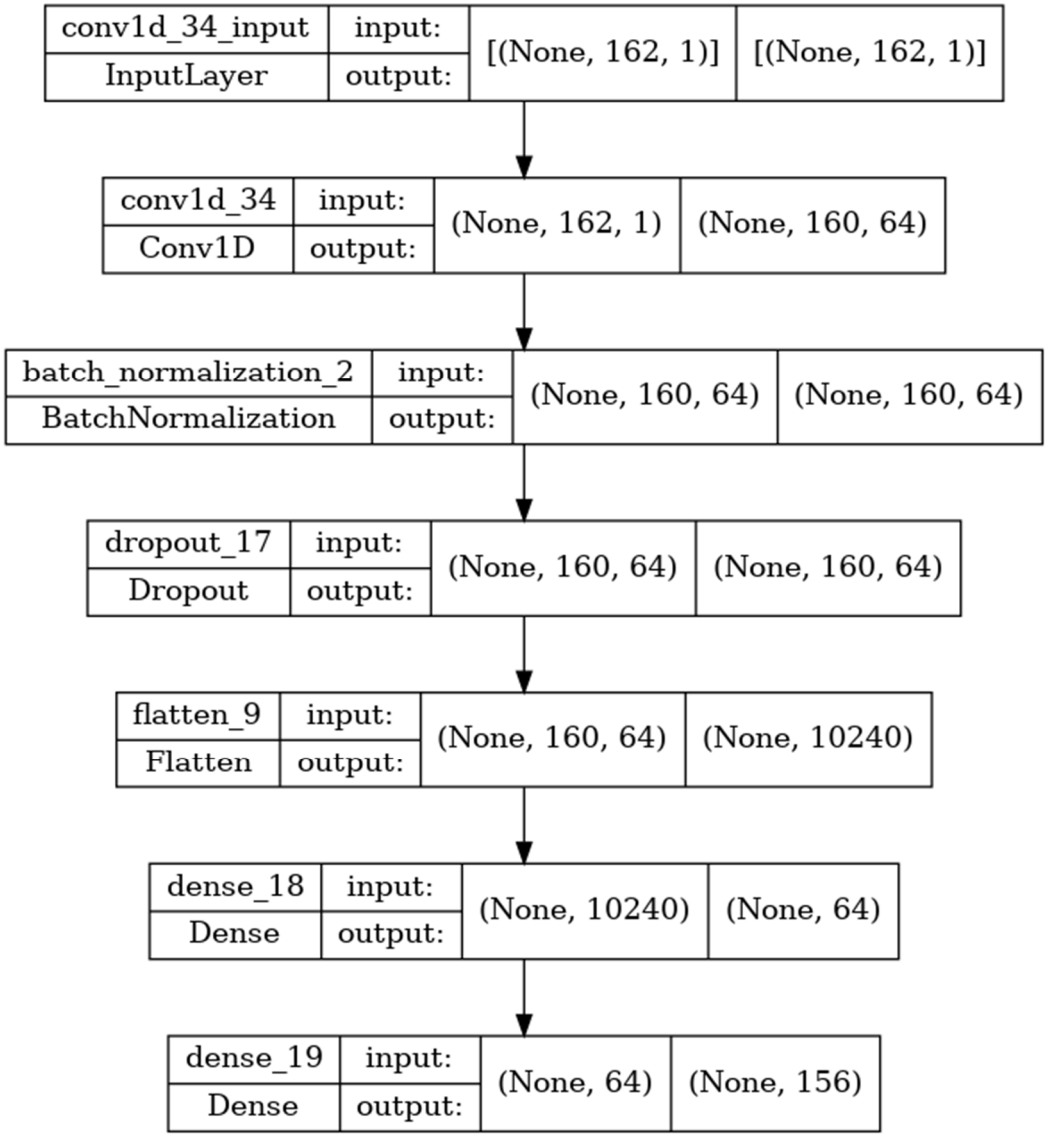


Figure 81 Model Architecture of 1D CNN

Table 13 Model Summary of 1D CNN

Layer Type	Layer Name	Input Shape	Output Shape	Parameters
Convolutional Layer	conv1d_34	(None, 160, 64)	(None, 160, 64)	256
Batch Normalization	batch_normalization_2	(None, 160, 64)	(None, 160, 64)	256
Dropout Layer	dropout_17	(None, 160, 64)	(None, 160, 64)	0
Flatten Layer	flatten_9	(None, 160, 64)	(None, 10240)	0
Dense Layer	dense_18	(None, 10240)	(None, 64)	655,424
Output Layer	dense_19	(None, 64)	(None, 156)	10,140
Total				666,076
Trainable Parameters				665,948
Non-trainable Parameters				128

13.7.1.1 Performance Evaluation

In the final epoch (Epoch 10/10), the training process for the model was carried out over 691 batches, with each batch comprising 5 seconds and having a mean square error (MSE) loss of 1.1963 and an accuracy of 73.27%. The validation set, evaluated concurrently, exhibited a lower loss of 1.0850 and a slightly higher accuracy of 76.17%. The overall performance metrics suggest that the model achieved a notable accuracy on the validation set, emphasizing its ability to generalize well to unseen data. Specifically, the test accuracy, computed separately, reached 76.17%. Table 14 illustrates the performance of 1D CNN.

Table 14 Performance Evaluation of 1D CNN

Metric	Training Set	Validation Set	Test Set
Loss	1.1963	1.0850	1.195
Accuracy	73.27%	76.17%	76.17%

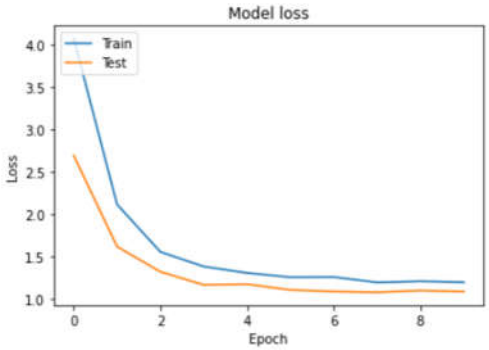


Figure 82 Train-Test Loss

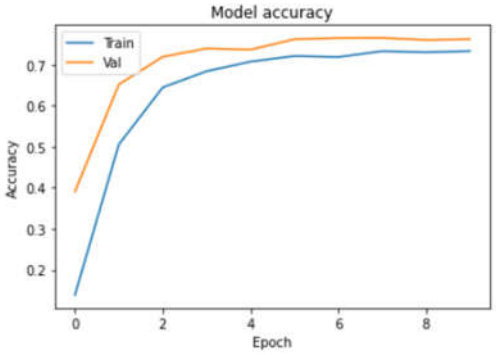


Figure 83 Train-Validation Accuracy

This outcome signifies a robust and reliable performance of the model in classifying the provided data. Figure 82 and Figure 83 illustrate the train-test loss, train-validation accuracy, train-validation loss respectively.

13.7.2 The Parallel 2D CNN With Attention Mechanism

The 2D Parallel CNN model architecture depicted in Table 15 has been crafted to provide enhanced feature extraction capabilities. This is achieved through a combination of parallel convolutional filters, batch normalization, and attention mechanisms. Here's a detailed breakdown of the architecture.

The architectural design of the model encompasses a series of interconnected layers aimed at proficiently processing input sequences. Commencing with an Input Layer, configured to accept sequences of dimensions (162, 1), the subsequent Parallel Convolution Layers, denoted by conv1d_37 and conv1d_38, employ filters of sizes 3 and 5, respectively. This dual convolutional approach enables the extraction of features at varying temporal granularities, capturing both shorter-term and slightly longer-term patterns.

The Concatenate layer integrates the outputs of these convolutional layers, augmenting the time dimension while preserving depth. The Normalization and Regularization mechanisms, implemented through Batch Normalization (batch_normalization_4) and Dropout (dropout_19), ensure a stable training process, mitigating overfitting and facilitating faster convergence.

Table 15 Model Summary of 2D CNN With Attention Mechanism

Layer Type	Layer Name	Output Shape	Parameters	Connected to
Input Layer	input_2	(None, 162, 1)	0	[]
Convolutional Layer	conv1d_37	(None, 160, 64)	256	['input_2[0][0]’]
Convolutional Layer	conv1d_38	(None, 158, 64)	384	['input_2[0][0]’]
Concatenate Layer	concatenate_1	(None, 318, 64)	0	['conv1d_37[0][0]'
Batch Normalization Layer	batch_normalization_4	(None, 318, 64)	256	['concatenate_1[0][0]']
Dropout Layer	dropout_19	(None, 318, 64)	0	['batch_normalization_4[0][0]']
Attention Mechanism Layer	attention_1	(None, 318, 64)	0	['dropout_19[0][0]']
Flatten Layer	flatten_11	(None, 20352)	0	['attention_1[0][0]']
Dense Layer	dense_22	(None, 64)	1,302,592	['flatten_11[0][0]']
Dense Layer	dense_23	(None, 156)	10,140	['dense_22[0][0]']
Total	-	-	1,313,628	-
Trainable Parameters	-	-	1,313,500	-
Non-trainable Parameters	-	-	128	-

The introduction of an Attention Mechanism (attention_1) enhances the model's ability to focus on specific segments of the input sequence, thereby prioritizing crucial features for predictions. The Flattening and Dense Layers, comprising flatten_11, dense_22, and dense_23, reshape and process the extracted features, ultimately culminating in a final output layer with 156 neurons.

This layer utilizes a softmax activation function, ensuring normalized output values between 0 and 1 that collectively sum to 1. The collaborative integration of parallel convolutional filters, attention mechanisms, and robust regularization techniques contributes to the model's adaptability and potential for accurate predictions in diverse temporal scenarios.

13.7.2.1 Performance Evaluation of the 2D Parallel CNN Model with Attention Mechanism

After training for 60 epochs, the performance metrics for the final epoch are as follows:

1. Training Loss: The model reports a training loss of 1.4957. Loss is a measure of how well the model's predictions match the true labels of the training data. Lower loss indicates a better fit to the training data.

2. Training Accuracy: The model achieved a training accuracy of 66.25%. This indicates that, during the training phase, the model correctly predicted the class labels for approximately 66.25% of the training samples.

3. Validation Loss: The validation loss is reported at 1.2306. The validation loss provides insights into how well the model generalizes to new, unseen data (the validation dataset). It's crucial to monitor this metric to ensure that the model isn't overfitting to the training data.

4. Validation Accuracy: The model's performance on the validation set resulted in an accuracy of 72.04%. This metric indicates that the model correctly predicted the class labels for approximately 72.04% of the validation samples.

The 2D Parallel CNN model showcases reasonable performance with a training accuracy of approximately 66.25% and a validation accuracy of 72.04%. The higher validation accuracy compared to the training accuracy can be attributed to various factors, including the architecture's ability to generalize well to unseen data or the

specific nature and distribution of the validation set. The layer wise model architecture of 2D CNN with Attention Mechanism is illustrated in Figure 84.

Figure 85 displays the training and testing performance metrics of a machine learning model over 60 epochs. The left graph illustrates the model accuracy, while the right graph depicts the model loss. Initially, the training accuracy is quite low but steadily improves, reaching around 65% by the 60th epoch. The testing accuracy follows a similar trend but consistently remains higher than the training accuracy throughout the epochs, peaking at approximately 70% around the 55th epoch before slightly plateauing.

The model loss graph on the right uses the y-axis to represent the loss values, starting from around 5.0 and decreasing to about 1.0, with the x-axis again denoting the epochs from 0 to 60. Both the training loss (blue line) and testing loss (orange line) decrease as training progresses. The initial loss is significantly high for both training and testing datasets, but they quickly drop within the first 10 epochs. By the end of the 60 epochs, the training loss stabilizes at around 1.5, while the testing loss is slightly lower, approximately 1.2, indicating a consistent reduction in error over time.

The learning curves indicated that the model is learning effectively, with improvements in accuracy and reductions in loss for both training and testing datasets. The higher accuracy and lower loss on the testing dataset compared to the training dataset may indicate good generalization performance, suggesting that the model is not overfitting to the training data.

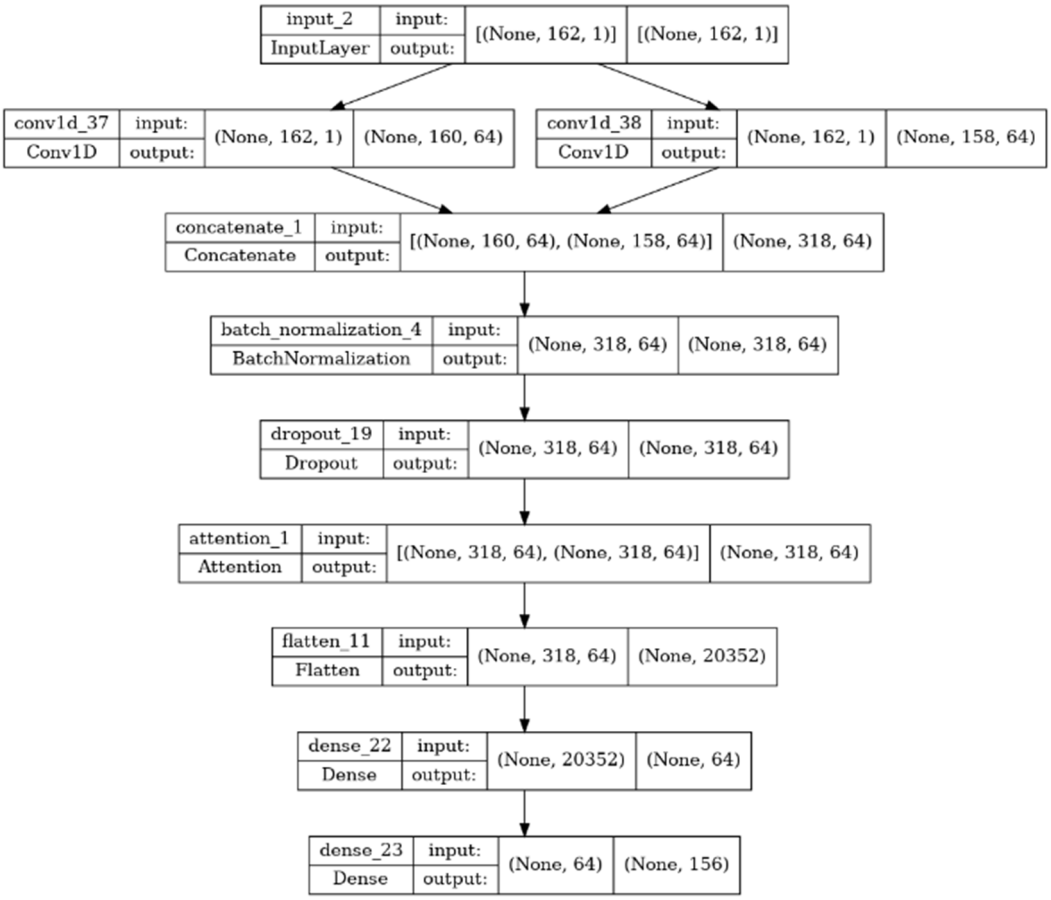


Figure 84 The Model Architecture of 2D CNN with Attention Mechanism

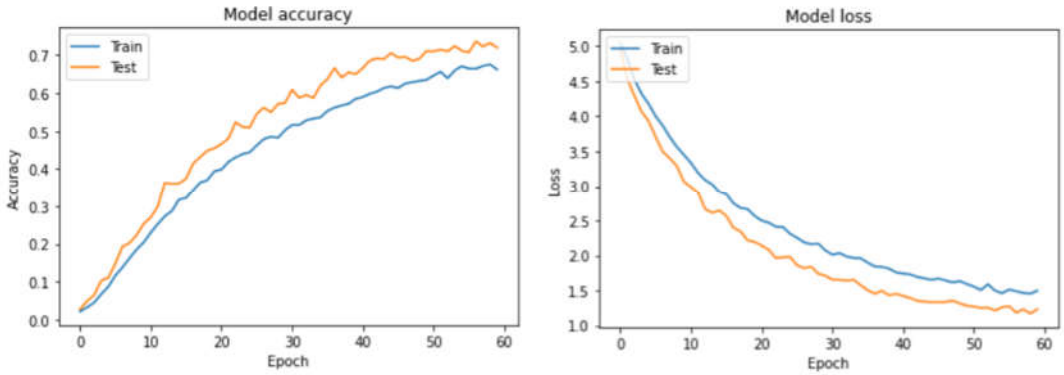


Figure 85 The Learning Curves Of 2D CNN With Attention Mechanism

13.7.3 Parallel 4D Convolutional Neural Network Model Design

In this study, a novel approach is proposed by constructing a multi-input Convolutional Neural Network (CNN) to use the potential relationship between

differently structured feature spaces of the same dataset. The dataset's feature set, represented by X, was restructured into four distinct configurations. This reshaping was crucial for inputting the data into four different CNN architectures simultaneously.

13.7.3.1 Model Architecture

The parallel 4D CNN is composed of four individual branches. Each branch exclusively processes one reshaped version of the dataset. The architecture within each branch includes convolutional layers, followed by max-pooling, leading to the generation of diverse feature maps for each data dimension. Once each branch processes its respective data dimension, the outputs are concatenated, merging the feature maps into a unified feature space.

This ensures that the model benefits from the unique patterns identifiable in each data dimension. Post concatenation, the unified feature space undergoes dense layers, culminating in the final output layer, which classifies the input into one of the 156 unique classes. Table 16 represents each layer of the architecture layer wise in each row of the table in a sequential manner.

13.7.3.2 Model Architecture Description

The layer wise breakdown of the model is illustrated in Table 16:

Table 16 Model Summary of 4D Parallel CNN

Layer (type)	Input Shape	Output Shape	Parameters
Input1 (Conv2D)	(38, 4, 1)	(36, 2, 32)	320
MaxPooling2D	(36, 2, 32)	(18, 1, 32)	0
Flatten	(18, 1, 32)	(576)	0
Input2 (Conv2D)	(19, 8, 1)	(17, 7, 32)	320
MaxPooling2D	(17, 7, 32)	(8, 3, 32)	0
Flatten	(8, 3, 32)	(768)	0
Input3 (Conv2D)	(76, 2, 1)	(74, 1, 32)	224
Flatten	(74, 1, 32)	(2368)	0
Input4 (Conv2D)	(152, 1, 1)	(150, 1, 32)	128

Layer (type)	Input Shape	Output Shape	Parameters
MaxPooling2D	(150, 1, 32)	(75, 1, 32)	0
Flatten	(75, 1, 32)	(2400)	0
Concatenate	(576, 768, 2368, 2400)	(6112)	0
Dense1 (Dense)	(6112)	(128)	781,376
Output (Dense)	(128)	(156)	20,124

The neural architecture is ingeniously designed with a novel approach, featuring four parallel branches, each dedicated to processing unique input dimensions. This strategic differentiation in input sizes empowers the model to comprehensively grasp diverse data perspectives, facilitating a detailed understanding of intricate patterns.

In the first branch, tailored for a 38 x 4 x 1 input, a Conv2D layer initiates the processing, succeeded by a MaxPooling2D operation and a Flatten layer for output linearization.

Branch 2, designed for a 19 x 8 x 1 input, follows a similar structure. Meanwhile, Branch 3, accepting a 76 x 2 x 1 input, incorporates only a Conv2D layer and a Flatten layer, omitting max pooling to potentially retain more spatial information. Branch 4, processing a 152 x 1 x 1 input, mirrors the design of Branches 1 and 2. After individual processing, the outputs from these branches are concatenated, forming a unified feature space of size 6,112. Subsequent layers, including a dense layer with 128 neurons and a final dense layer with 2 neurons, contribute to the classification task.

This model's innovation lies in its simultaneous processing of multiple data perspectives, ensuring a holistic approach that captures diverse feature scales and patterns across dimensions. This distinctive design sets it apart from traditional single-dimensional architectures, combining diversity in feature extraction with focused interpretation.

13.7.3.3 Performance Evaluation

In the evaluation of the model's performance, several key metrics are considered to provide a comprehensive understanding of its effectiveness. These metrics, derived

from the validation phase, shed light on the model's ability to make accurate predictions and highlight areas for potential improvement.

The model exhibits an innovative architecture with four parallel branches, processing distinct input dimensions simultaneously. It achieves notable performance in validation and test phases, with high precision indicating reliability in positive predictions. While the overall accuracy is strong, there is room for further investigation, especially concerning recall, to enhance positive case identification. The training phase also shows consistency in accuracy and precision, suggesting a balanced model.

Table 17 The Performance Evaluation

Aspect	**Details**
Model Architecture	Four parallel branches processing unique input dimensions. Branches differ in input size and processing steps. After individual processing, outputs are concatenated. Subsequent dense layers contribute to classification.
Validation Loss	1.1083
Validation Accuracy	76.53%
Validation Precision	99.01%
Validation Recall	75.88%
Overall Accuracy	76.53%
Test Loss	1.1082
Test Accuracy	76.53%
Test Precision	99.01%
Test Recall	75.88%
Train Loss	1.258
Train Accuracy	75.53%
Train Precision	98.08%
Train Recall	74.88%

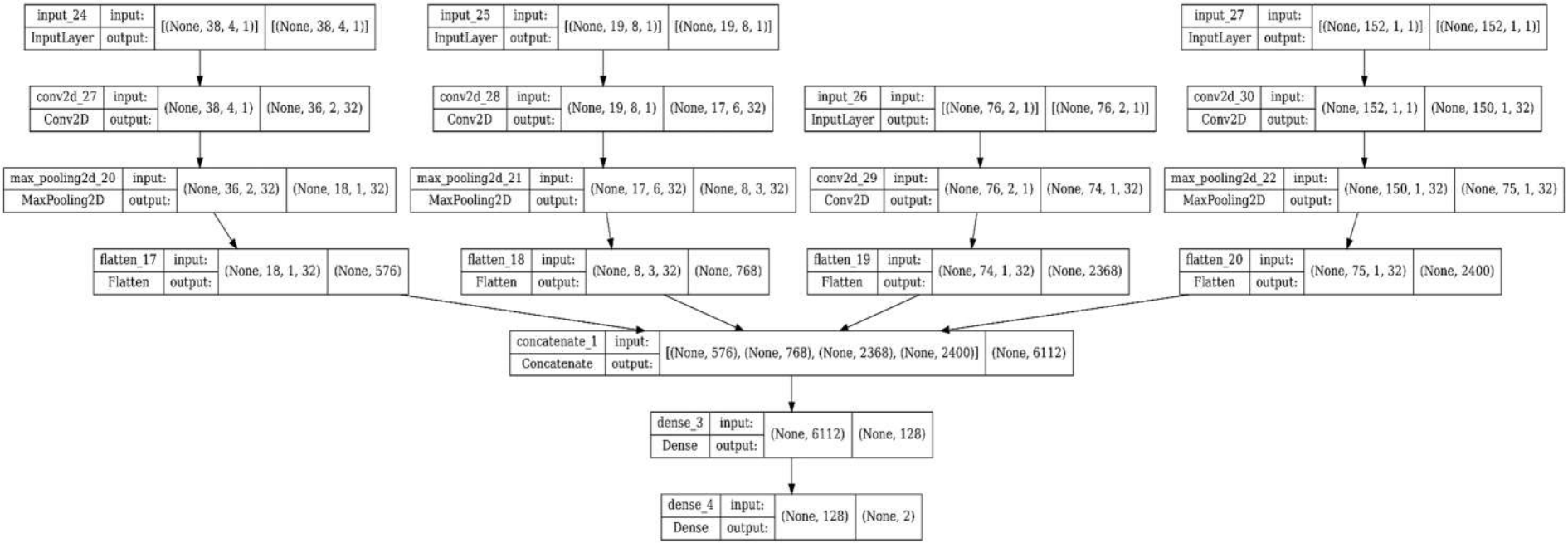


Figure 86 Model Architecture

The model's performance evaluation illustrated in Table 17 reveals a comprehensive analysis across training, validation, and test phases. The neural architecture, distinguished by four parallel branches processing unique input dimensions, demonstrates an innovative approach. Each branch is tailored to specific input sizes, facilitating a detailed understanding of diverse data perspectives. Following individual processing, the outputs are concatenated, forming a feature space of 6,112 dimensions. Subsequent layers, including dense layers, contribute to the classification task. Figure 86 represents the detailed architecture of the model.

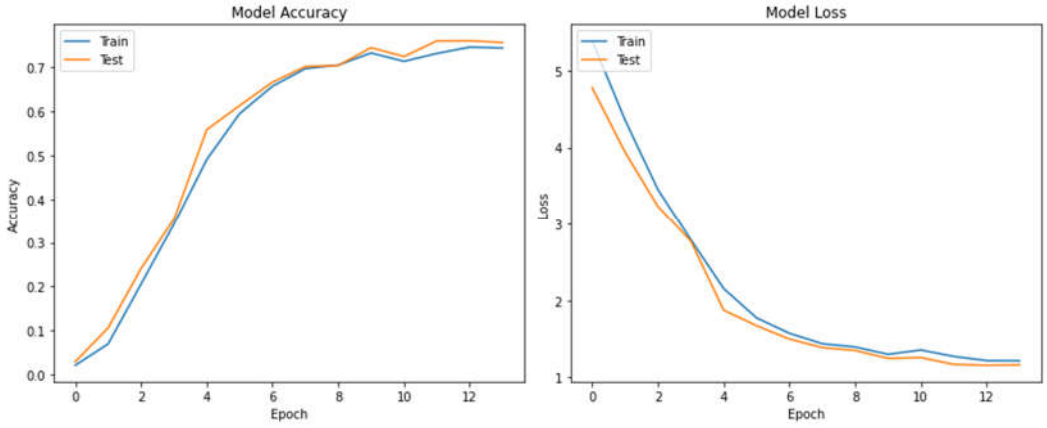


Figure 87 The Learning Curves

During the validation phase, the model exhibits a validation loss of 1.1083, signifying its alignment of predictions with actual labels. The validation accuracy of 76.53% indicates a notable correctness in predicting labels. The validation precision of 99.01% showcases the model's reliability in correctly predicting positive observations and the validation recall of 75.88% suggests potential instances where positive cases might be overlooked. The overall accuracy reinforces the model's capability to predict the correct label approximately 76.53% of the time on the evaluation dataset. Figure 87 illustrates the learning curves of the experiment.

In the test phase, the model maintains consistency with the validation results. The test loss of 1.1082 signifies alignment with actual labels, and the test accuracy of 76.53% demonstrates impressive correctness.

The test precision, mirroring the validation precision at 99.01%, reaffirms the model's reliability in positive predictions. The test recall of 75.88% indicates a similar trend observed in the validation phase. The training phase provides additional insights. With a training loss of 1.258, slightly higher than the test loss, there is a potential indication of some level of overfitting. Slightly lower recall values suggest potential opportunities for improvement, particularly in identifying positive cases. The training accuracy of 75.53% signifies correctness during the training set, while the training precision of 98.08% mirrors the precision observed in the test and validation phases. The training recall of 74.88% aligns with the recall observed in the test phase. Overall, the model demonstrates consistent performance between training, validation, and test phases, with high precision and reasonable accuracy.

13.7.4 4D CNN With Attention Mechanism

Convolutional Neural Networks (CNNs) have gained prominence in processing grid-like data, such as image and audio. The integration of the attention mechanism, inspired by human attention, allows the model to focus on more informative parts of the input. In the context of audio processing and the Malayalam language's accented speech recognition, emphasizing significant features can boost model performance. Table 18 illustrates the architecture of 4D CNN with attention mechanism.

Table 18 Model Summary of 4D CNN With Attention Mechanism

Layer (type)	Output Shape	Parameters	Connected to
input_105 (Input Layer)	(None, 38, 4, 1)	0	[]
input_106 (Input Layer)	(None, 19, 8, 1)	0	[]
input_107 (Input Layer)	(None, 76, 2, 1)	0	[]
input_108 (Input Layer)	(None, 152, 1, 1)	0	[]
conv2d_126 (Conv2D)	(None, 36, 2, 32)	320	['input_105[0][0]']
conv2d_128 (Conv2D)	(None, 17, 6, 32)	320	['input_106[0][0]']
conv2d_130 (Conv2D)	(None, 74, 1, 32)	224	['input_107[0][0]']
conv2d_132 (Conv2D)	(None, 150, 1, 32)	128	['input_108[0][0]']
conv2d_127 (Conv2D)	(None, 36, 2, 1)	33	['conv2d_126[0][0]']
conv2d_129 (Conv2D)	(None, 17, 6, 1)	33	['conv2d_128[0][0]']
conv2d_131 (Conv2D)	(None, 74, 1, 1)	33	['conv2d_130[0][0]']
conv2d_133 (Conv2D)	(None, 150, 1, 1)	33	['conv2d_132[0][0]']
reshape_32 (Reshape)	(None, 72)	0	['conv2d_127[0][0]']

Layer (type)	Output Shape	Parameters	Connected to
reshape_34 (Reshape)	(None, 102)	0	['conv2d_129[0][0]']
reshape_36 (Reshape)	(None, 74)	0	['conv2d_131[0][0]']
reshape_38 (Reshape)	(None, 150)	0	['conv2d_133[0][0]']
activation_18(Activation)	(None, 72)	0	['reshape_32[0][0]']
activation_19 (Activation)	(None, 102)	0	['reshape_34[0][0]']
activation_20 (Activation)	(None, 74)	0	['reshape_36[0][0]']
activation_21 (Activation)	(None, 150)	0	['reshape_38[0][0]']
repeat_vector_17(Repeat Vector)	(None, 32, 72)	0	['activation_18[0][0]']
repeat_vector_18	(None, 32, 102)	0	['activation_19[0][0]']
repeat_vector_19	(None, 32, 74)	0	['activation_20[0][0]']
repeat_vector_20	(None, 32, 150)	0	['activation_21[0][0]']
permute_17 (Permute)	(None, 72, 32)	0	['repeat_vector_17[0][0]']
permute_18 (Permute)	(None, 102, 32)	0	['repeat_vector_18[0][0]']
permute_19 (Permute)	(None, 74, 32)	0	['repeat_vector_19[0][0]']
permute_20 (Permute)	(None, 150, 32)	0	['repeat_vector_20[0][0]']
reshape_33 (Reshape)	(None, 36, 2, 32)	0	['permute_17[0][0]']
reshape_35 (Reshape)	(None, 17, 6, 32)	0	['permute_18[0][0]']
reshape_37 (Reshape)	(None, 74, 1, 32)	0	['permute_19[0][0]']
reshape_39 (Reshape)	(None, 150, 1, 32)	0	['permute_20[0][0]']
multiply_16 (Multiply)	(None, 36, 2, 32)	0	['conv2d_126[0][0]], 'reshape_33[0][0]']
multiply_17 (Multiply)	(None, 17, 6, 32)	0	['conv2d_128[0][0]', 'reshape_35[0][0]']
multiply_18 (Multiply)	(None, 74, 1, 32)	0	['conv2d_130[0][0]', 'reshape_37[0][0]']
multiply_19 (Multiply)	(None, 150, 1, 32)	0	['conv2d_132[0][0]', 'reshape_39[0][0]']
flatten_95 (Flatten)	(None, 2304)	0	['multiply_16[0][0]']
flatten_96 (Flatten)	(None, 3264)	0	['multiply_17[0][0]']
flatten_97 (Flatten)	(None, 2368)	0	['multiply_18[0][0]']
flatten_98 (Flatten)	(None, 4800)	0	['multiply_19[0][0]']
concatenate_18 (Concatenate)	(None, 12736)	0	['flatten_95[0][0]', 'flatten_96[0][0]', 'flatten_97[0][0]', 'flatten_98[0][0]']
dense_37 (Dense)	(None, 128)	1630336	['concatenate_18[0][0]'
dense_38 (Dense)	(None, 156)	20124	['dense_37[0][0]']

Total parameters: 1,651,584

Trainable parameters: 1,651,584

Non-trainable parameters:0

The neural architecture presented consists of four distinct input layers, each designated for processing specific dimensions of the input data, such as spatial and channel information. Following the inputs, a series of convolutional layers, namely conv2d_126, conv2d_128, conv2d_130, and conv2d_132, employ varying filter sizes to learn hierarchical features from the input data. Subsequent reshape layers (reshape_32, reshape_34, reshape_36, reshape_38) modify the output shapes, and activation layers (activation_18, activation_19, activation_20, activation_21) introduce non-linearities to capture intricate patterns.

The inclusion of repeat vector and permute layers suggests a temporal aspect in the architecture, enabling the model to process sequential data effectively. Element-wise multiplication layers (multiply_16, multiply_17, multiply_18, multiply_19) facilitate interactive feature learning by combining the convolutional outputs with reshaped and permuted activations.

The flattened outputs are concatenated (concatenate_18), creating a comprehensive feature vector. Two dense layers (dense_37, dense_38) follow, serving as the final stages for feature transformation or classification, with the latter having neurons indicative of a multi-class classification task. This architecture, adept at handling multi-dimensional data, showcases a thoughtful combination of convolutional, reshaping, and dense layers to capture diverse patterns and relationships within the input data.

13.7.4.1 Model Description

The model employs four parallel CNN branches as shown in Figure 88. Each branch processes the input with different dimensions, capturing varied feature representations. Within each branch, an attention mechanism is applied after the initial convolution. This mechanism assigns different weights to different parts of the input feature map, enabling the model to focus on areas that are more relevant and downplay less informative regions.

The attention block starts with a convolution layer to transform the feature map. This is then reshaped, and a softmax activation ensures that the attention weights sum up to 1. These weights are then replicated and permuted to match the original dimensions, followed by a multiplication operation that applies the attention weights to the original feature map.

Each CNN branch has an associated convolutional layer, followed by the attention mechanism. The kernel sizes for convolution are dynamically determined based on the input shape to ensure optimal feature extraction. After applying attention, the feature maps are flattened to be processed further.

Outputs from the four branches are concatenated, providing a holistic representation encompassing insights from all branches. This concatenated output is passed through dense layers to achieve the final classification. The final layer uses a softmax activation with 156 neurons for 156 unique classes. This model cleverly blends traditional convolutional layers with an attention mechanism, offering a unique approach to processing multi-dimensional data.

13.7.4.2 Parallel Branches of CNN with Attention Mechanism

Each of the parallel branches processes the input differently, accommodating varying dimensional perspectives of the data. The branches are:

1. Branch 1: Input Dimension (38, 4, 1)
2. Input Layer: This layer receives the reshaped audio features of size (38, 4, 1).
3. Conv2D Layer: It applies convolution operations using a (3, 3) kernel size. This layer extracts spatial features from the input data.
4. Attention Mechanism:
    i. A convolution operation with a (1, 1) kernel reduces the depth dimension, producing a single-channel feature map.
    ii. The feature map is reshaped and passed through a softmax activation, yielding the attention weights.

iii. These weights are then repeated and adjusted to match the original feature map's size, and finally multiplied with the original feature map. This operation gives more importance to regions with higher attention weights.

5. Flatten Layer: The feature map after applying attention is flattened to produce a one-dimensional tensor.
6. Branch 2: Input Dimension (19, 8, 1)
7. Input Layer: Accepts the reshaped data of size (19, 8, 1).
8. Conv2D Layer: This branch also uses a (3, 3) kernel for the convolution, extracting spatial features.
9. Attention Mechanism: Implemented similarly to Branch 1, focusing on relevant regions of the feature map.
10. Flatten Layer: The feature map post-attention is flattened.
11. Branch 3: Input Dimension (76, 2, 1)
12. Input Layer: Takes in reshaped data of size (76, 2, 1).
13. Conv2D Layer: Here, due to the input shape, a (3, 2) kernel is used for the convolution. This size helps capture the unique spatial relationships present in this reshaped data.
14. Attention Mechanism: Identical in its working to the previous branches but tailored for this specific input shape.
15. Flatten Layer: Converts the multi-dimensional output to a one-dimensional tensor.
16. Branch 4: Input Dimension (152, 1, 1)
17. Input Layer: Accepts reshaped data of size (152, 1, 1).
18. Conv2D Layer: Due to the specific shape of the data, a (3, 1) kernel is utilized, ensuring features are captured effectively across the length of the reshaped data.
19. Attention Mechanism: As before, this block dynamically adjusts attention weights to the input feature map.
20. Flatten Layer: The feature map is flattened, preparing it for concatenation.
21. Merging & Dense Layers:

22. Concatenate: The flattened outputs from all four branches are concatenated, producing a unified representation encompassing insights from all branches.
23. Dense Layer with ReLU Activation: This layer has 128 neurons and applies the ReLU activation function. It serves to further process the concatenated features, adding depth to the model's capability.
24. The final dense layer uses a softmax activation with 156 neurons. This ensures that the output represents the probability distribution over the 156 classes.

The architecture presented in Figure 88 is a testament to the fusion of traditional convolutional operations with contemporary attention mechanisms. Each branch is tailored to process the data in its unique reshaped form, ensuring the model captures a wide array of features. The attention mechanism, on the other hand, offers the model a way to focus on pivotal information, enhancing its discriminative capability. These components form a powerful architecture capable of complicated audio data processing, especially suited for tasks like accented speech recognition in the Malayalam language.

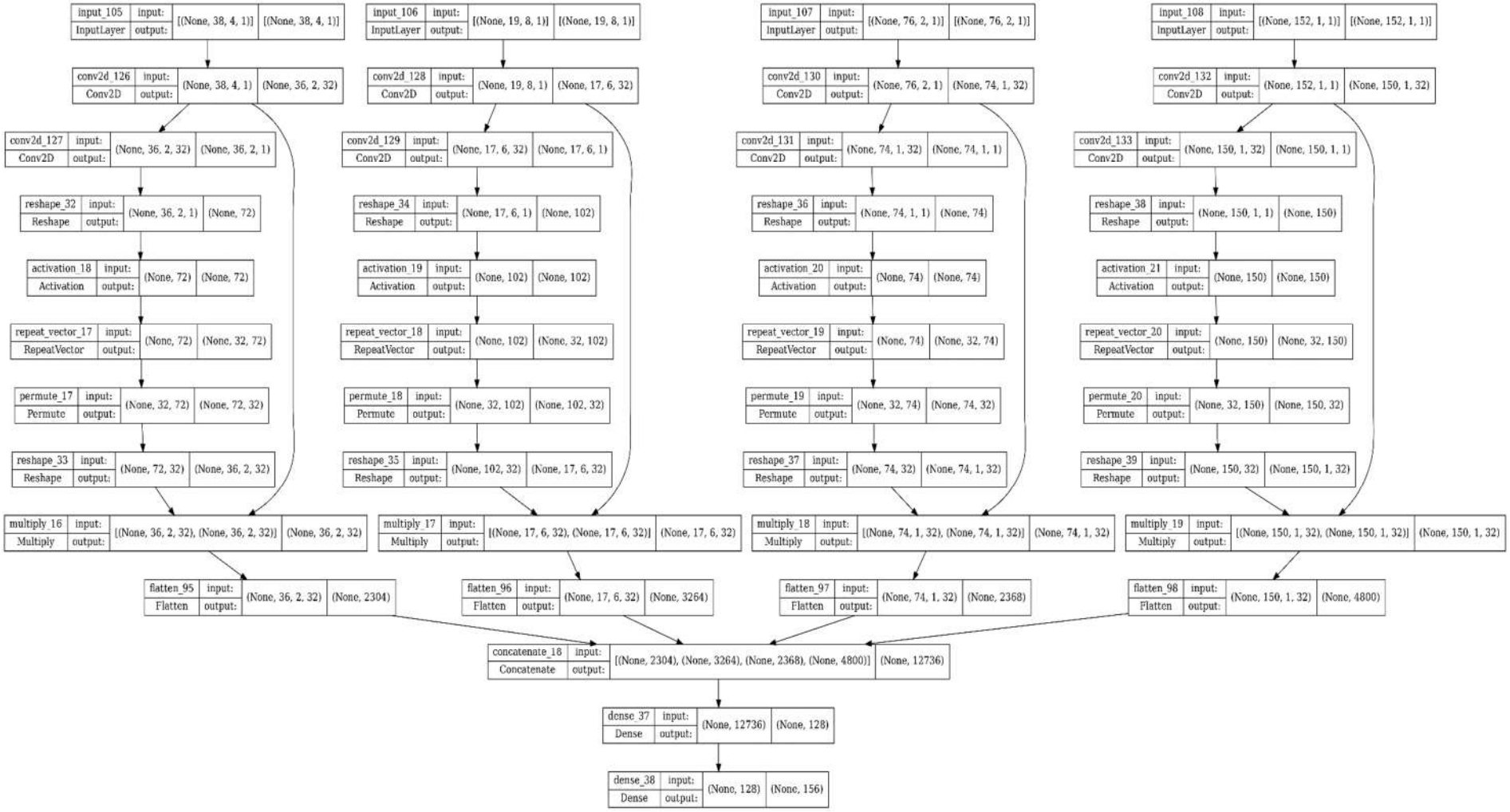


Figure 88 Model Architecture

13.7.4.3 Novelty

The parallel 4D architecture coupled with attention inherently stands out. Traditional models often employ either multi-dimensional inputs or attention but rarely in conjunction. By processing the audio data in various reshaped forms simultaneously, the model can capture a rich set of features. The attention mechanism ensures that within these diverse representations, the model emphasizes the most crucial information.

In the context of accented speech recognition, where complex and subtle features can significantly impact model performance, such an architecture offers a robust approach to extract and focus on pivotal features. In terms of evaluation, the model exhibits high performance with an accuracy of 76.53%. The proposed 4D Parallel CNN model with an attention mechanism represents a novel approach in the domain of accented speech recognition for the Malayalam language.

By employing multiple dimensions and focusing on salient features, this architecture provides a robust framework for extracting pivotal characteristics from audio data. The promising evaluation metrics feature the architecture's efficacy and its potential in pushing the boundaries of accented speech recognition research.

13.7.4.4 Performance Metrics

Table 19 illustrates the performance evaluation of the model. The model was constructed with a training loss of 1.2307 which represents the model's error during training on the ninth epoch. Lower loss indicates that the model's predictions are close to the actual values. Training accuracy of the model was 73.32% which indicates that the model correctly predicted the class for about 73.32% of the training samples.

This is a good accuracy, suggesting that the model has learned a good portion of patterns from the training data. Validation loss was 1.0922 indicating the model's error when evaluated on the validation set. Comparing this with the training loss

gives insight into whether the model might be overfitting (if training loss is much lower) or underfitting (if training loss is higher).

Table 19 The Performance Evaluation

Metric	**Value**
Loss (Training)	1.2307
Accuracy (Training)	73.32%
Validation Loss	1.0922
Validation Accuracy	76.42%
Overall Model Evaluation	76.42%

Validation accuracy of 76.42% represents the percentage of correct predictions on the validation set. It's notable that the validation accuracy is higher than the training accuracy, suggesting that the model generalizes well to unseen data. Accuracy of 76.42% indicates that after training, when the model was evaluated on the test dataset, it achieved an accuracy of 76.42%. This means that in a real-world scenario, given unseen data, the model is expected to make correct predictions approximately 76.42% of the time.

Achieving an accuracy of over 76% in such a complex task, especially when dealing with accented speech recognition in the Malayalam language, is commendable. Considering the intricacies of accents and the complications in the speech patterns, this is a significant achievement. The inclusion of attention mechanisms in the model architecture seems to play a pivotal role in focusing on the relevant audio features, aiding in the discernment of the accented variations. Figure 89 represents the learning curves obtained in the study.

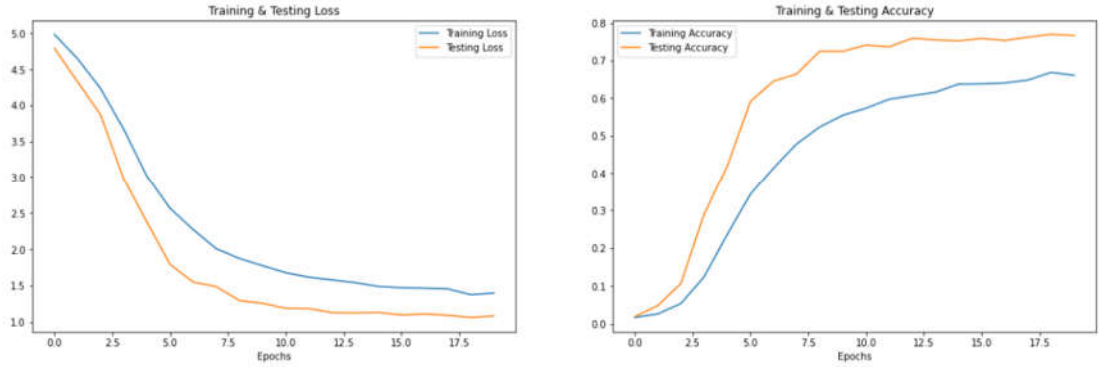


Figure 89 The Learning Curves

13.7.5 Bidirectional LSTM Model Architecture for Accented Speech Recognition

In this effort to capture the temporal dependencies in the accented Malayalam speech dataset, a Bidirectional LSTM (Bi-LSTM) model was constructed to process sequences both forwards and backwards, allowing the model to gain insights from past (backward direction) and future (forward direction) states simultaneously. Table 20 illustrates the layered description of the architecture adopted in this study.

Table 20 Model Summary of BiLSTM Model

Layer (type)	**Output Shape**	**Parameters**
input_4 (Input Layer)	(None, 152, 1)	0
bidirectional_3 (Bidirectional)	(None, 512)	528384
dropout_3 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 512)	262656
dense_7 (Dense)	(None, 156)	80028
Total parameters: 871,068		
Trainable parameters: 871,068		
Non-trainable parameters: 0		

The architecture begins with an input layer, denoted as 'input_4,' which accepts sequences of shape (152, 1). The subsequent layer is a bidirectional recurrent layer,

'bidirectional_3,' employing a bidirectional long short-term memory (LSTM) network with 512 units. This bidirectional nature allows the model to capture contextual information from both forward and backward sequences. To prevent overfitting, a dropout layer, 'dropout_3,' is applied, where 50% of the neurons are randomly ignored during training.

The architecture further includes two dense layers, 'dense_6' and 'dense_7.' The former consists of 512 neurons with a rectified linear unit (ReLU) activation function, contributing to feature extraction and representation. The final dense layer, 'dense_7,' outputs 156 neurons with a Softmax activation, aligning with the number of classes in the dataset. The total trainable parameters for this architecture amount to 871,068.

This design aims to use bidirectional LSTM networks for effective sequence processing while incorporating dropout for regularization, ultimately resulting in a model capable of accurate and generalized predictions in a classification task. The model was compiled using the Adam optimizer, categorical cross entropy as the loss function, considering the multi-class classification task, and accuracy as the evaluation metric.

The Bi-LSTM architecture was selected for its ability to retain memory from both past and future sequences, making it particularly suitable for sequence classification tasks. With a total of 871,068 trainable parameters, the model aims to offer a robust solution to accented speech recognition in Malayalam by capturing intricate patterns and dependencies in the audio data.

13.7.5.1 Working of the Bidirectional LSTM Network

The network starts by accepting a sequence of data with 152 features. These features, derived from audio samples, represent a sequence of information that changes over time. In the context of speech, these sequences capture the temporal progression of the sound.

LSTM is a type of recurrent neural network (RNN) that is designed to recognize patterns over time intervals. RNNs possess a form of memory, allowing them to retain information from earlier in the sequence to influence later parts. This is essential for understanding speech, where the meaning can depend on the context provided by earlier sounds or words.

Traditional LSTMs process data from the start of the sequence to the end. Bidirectional LSTMs, on the other hand, process the data in both directions (from start to end and from end to start). The rationale is that the output at a certain time depends not only on the past but also on the future data points. For speech recognition, this means understanding a word or sound might be influenced by both preceding and following words or sounds.

The combination of forward and backward information at each time step provides a richer representation of the data. Once the data has passed through the Bidirectional LSTM layer, it undergoes a dropout operation. Dropout is a regularization technique where randomly selected neurons are ignored (or "dropped-out") during training, which means that they are not considered during a particular forward or backward pass. This is essential to prevent overfitting, ensuring that the network generalizes well to unseen data.

A rate of 0.5 means that 50% of the input units are set to 0. The output from the Dropout layer is then passed to a dense (or fully connected) layer. This layer undergoes a linear transformation and a non-linear activation function (ReLU, in this case). It aids in capturing the non-linear relationships in the dataset and further refines the feature representation. The network produces an output through a softmax activation function in the last dense layer. This function converts the raw output scores from the previous layer into probabilities for each of the 156 classes.

The network captures both the immediate and contextual information in the audio data sequence using Bi-LSTM, regularizes dropout to avoid overfitting, refines features using dense layers, and classifies into one of the 156 classes using softmax.

The architecture is particularly suited for time-series data like speech, where understanding context and sequence is crucial.

13.7.5.2 Performance Evaluation

Table 21 Performance Evaluation

Phase	Loss	Accuracy
Training	1.0986	72.97%
Validation	1.0986	75.01%
Test	1.0986	75.01%

In the final epoch of training the Bidirectional LSTM model, the loss on the training data was recorded at 1.1824, with an associated accuracy of 72.97%. The training accuracy indicates that the model successfully classified nearly 73% of the training dataset, reflecting a substantial grasp of the underlying patterns.

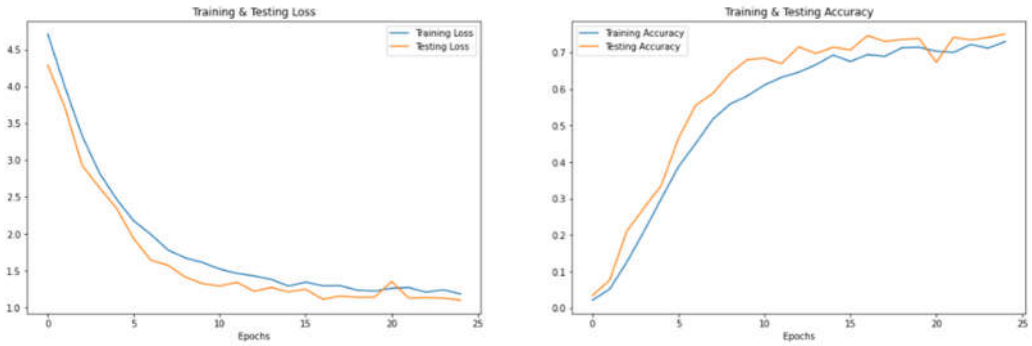


Figure 90 Performance Evaluation

During the validation phase, the model demonstrated a loss of 1.0986, coupled with an accuracy of 75.01%. This implies that the model performed well on previously unseen validation data, correctly predicting labels for 75% of instances. The test phase also yielded a loss of 1.0986 and an accuracy of 75.01%. These consistent accuracy values across training, validation, and test datasets as shown in Table 21 suggest that the model has effectively learned and generalized patterns from the training data, showcasing its robustness in making accurate predictions on new and

unseen instances. The performance evaluation of the experiment is shown in Figure 90.

13.7.6 The CNN-LSTM Hybrid Approach

The hybrid model combines the feature extraction capabilities of CNNs with the sequence modeling strengths of LSTMs. It allows the network to both recognize local patterns through convolutional filters and remember long-term dependencies using recurrent cells. Table 22 represents the model summary of the hybrid approach adopted in this phase of the experiment.

Table 22 Model Summary

Layer (type)	Output Shape	Parameters
Input Layer	(None, 152, 1)	0
Conv1D	(None, 150, 64)	256
MaxPooling1D	(None, 75, 64)	0
Conv1D	(None, 73, 128)	24,704
MaxPooling1D	(None, 36, 128)	0
Dropout	(None, 36, 128)	0
Bidirectional	(None, 512)	788,480
Dropout	(None, 512)	0
Dense	(None, 512)	262,656
Dense	(None, 156)	80,028
Total params 1,156,124		
Trainable params 1,156,124 and non-trainable params 0		

13.7.6.1 Convolutional Layers (Feature Extraction)

1. Conv1D Layer: The first convolutional layer uses 64 filters, each of size 3. It scans the input data for local patterns or features. The activation function 'ReLu' introduces non-linearity, enabling the model to capture complex patterns.

2. MaxPooling1D Layer: This layer reduces the spatial size of the representation, preserving the most essential features, and thus, reduces the computation for the subsequent layers.

3. Another Conv1D Layer: This layer, with 128 filters, explores deeper into the features, and captures more intricate patterns in the data.

4. Another MaxPooling1D Layer: Further down sampling of the feature maps.

5. Dropout Layer: After the convolution operations, a dropout layer is used, which randomly sets a fraction (0.5 or 50% in this case) of the input units to 0 during training. This helps to prevent overfitting.

13.7.6.2 Bidirectional LSTM (Sequence Modeling)

The feature maps from the CNN layers are then fed into a Bidirectional LSTM layer with 256 units. LSTMs are a type of recurrent neural network (RNN) well-suited for sequence data. This ensures that for each timestep, the LSTM considers both past and future data, making it highly effective for speech recognition.

This fully connected layer processes the LSTM outputs and learns to make decisions based on the combined features from both the CNN and LSTM. The final dense layer has 156 neurons with softmax activation. In conclusion, the hybrid CNN-LSTM model is a powerful architecture for tasks that require both feature extraction and sequence modeling.

By combining the strengths of both CNNs and LSTMs, it captures both the local variations and global context from the input data, making it especially apt for complex tasks like speech recognition.

13.7.6.3 Performance Evaluation of the Hybrid CNN-LSTM Model

After 20 epochs of training on the dataset, the hybrid CNN-LSTM model has demonstrated commendable results which is depicted in Figure 91.

i. Training Phase

- Loss: The model achieved a loss value of 1.2014. The loss value indicates how well the model's predictions match the true labels. A lower value signifies better performance, and over the epochs, it's expected that the model worked to minimize this.

- Accuracy: The model achieved an accuracy of 72.13% on the training data. This indicates that the model correctly predicted the labels of approximately 72.13% of the training samples.

ii. Validation Phase

- Loss: On the validation dataset, which the model hasn't seen during training, it achieved a loss value of 1.0652. This lower loss value, compared to the training loss, is promising as it suggests that the model is generalizing well and not merely memorizing the training dataset.

- Accuracy: The accuracy on the validation set is 76.49%, which is higher than the training accuracy. This is a very positive sign, indicating that the model is performing even better on new, unseen data than on the training data. The higher accuracy on the validation set suggests that the model is robust and can potentially perform well in real-world scenarios.

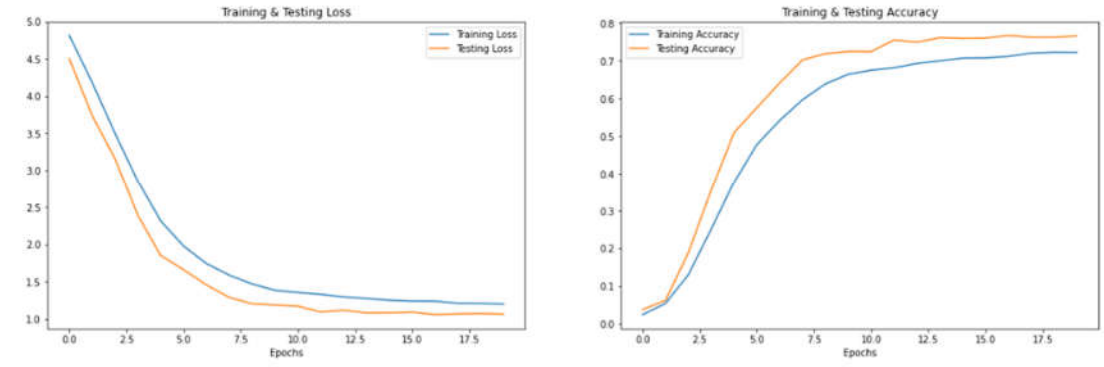


Figure 91 The Performance Evaluation

13.8 Experimental Results

Table 23 Comparative Analysis

Approach	Description	Epochs	Time Per Epoch	Total Training Time	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
4D Parallel CNN (No Attention)	Multi-branched CNN with tailored kernel shapes	14	Approx. 15 seconds	Approx. 3.5 minutes	72.13%	76.42%	1.2014	1.0652
4D Parallel CNN (With Attention)	CNN with attention mechanisms in branches	20	Approx. 172 seconds (2.87 minutes)	Approx. 57.3 minutes	72.17%	76.49%	1.1963	1.0850
Bidirectional LSTM	LSTM processing data from both past and future contexts	25	Approx. 84 seconds	Approx. 35 minutes	72.97%	75.01%	1.1824	1.0986
CNN-LSTM Hybrid	Combination of Conv1D and Bidirectional LSTM layers	20	Approx. 30 seconds	Approx. 10 minutes	72.13%	76.49%	1.2014	1.0852
2D Parallel CNN	Two parallel Conv layers + attention	60	Approx. 19 seconds	Approx. 19 minutes	66.25%	72.04%	1.4957	1.2306
1D CNN	Sequential layers	10	Approx 100 seconds	Approx. 30 minutes	73.27%	76.17%	1.1963	1.085

The comparative analysis of the experimental results is depicted in Table 23. The hybrid CNN-LSTM model exhibits promising results with an accuracy of approximately 76.49% on the validation set. This level of performance, given the complexity of the task at hand, emphasizes the effectiveness of combining convolutional layers for feature extraction with LSTM layers for sequence modeling.

The models explored in this research primarily focused on the integration of convolutional layers, LSTM layers, and attention mechanisms to recognize and classify accented speech patterns.

The analysis of the experimental results is:

1. The 4D Parallel CNN without attention mechanisms employs a multi-branched architecture with tailored kernel shapes. This model completed 14 epochs, with each epoch taking approximately 15 seconds, resulting in a total training time of around 3.5 minutes. It achieved a training accuracy of 72.13% and a validation accuracy of 76.42%, with training and validation losses of 1.2014 and 1.0652, respectively. This model demonstrated a strong ability to generalize to unseen data with the lowest validation loss among all models, suggesting a tight fit to the validation data.

2. The addition of attention mechanisms to the 4D Parallel CNN architecture led to significant changes. This model required 20 epochs, with each epoch taking approximately 172 seconds (2.87 minutes), amounting to a total training time of about 57.3 minutes. The inclusion of attention mechanisms slightly improved the model's performance, with a training accuracy of 72.17% and a validation accuracy of 76.49%. The training and validation losses were 1.1963 and 1.0850, respectively. While the training time was substantially longer, the model's performance indicates that attention mechanisms can enhance accuracy, albeit at the cost of increased computational time.

3. The Bidirectional LSTM model, which processes data from both past and future contexts, completed 25 epochs with each epoch taking approximately 84 seconds, resulting in a total training time of around 35 minutes. This model achieved a training accuracy of 72.97% and a validation accuracy of 75.01%. The training and validation losses were 1.1824 and 1.0986, respectively. Despite having a higher training accuracy, the validation accuracy was slightly lower than that of the CNN-based models, indicating a potential overfitting issue.

4. Combining Conv1D and Bidirectional LSTM layers, the CNN-LSTM Hybrid model completed 20 epochs, with each epoch taking about 30 seconds, leading to a total training time of approximately 10 minutes. It achieved a training accuracy of 72.13% and a validation accuracy of 76.49%, with training and validation losses of 1.2014 and 1.0852, respectively. This model strikes a favorable balance between accuracy and training time, making it a highly efficient option for this task.

5. The 2D Parallel CNN, which incorporates two parallel convolutional layers with attention mechanisms, completed 60 epochs, each taking approximately 19 seconds, resulting in a total training time of around 19 minutes. It attained a training accuracy of 66.25% and a validation accuracy of 72.04%, with training and validation losses of 1.4957 and 1.2306, respectively. Despite its relatively short training time, this model exhibited the lowest accuracy and highest loss, indicating less effectiveness in this application.

6. Finally, the 1D CNN with sequential layers completed 10 epochs, with each epoch taking about 100 seconds, resulting in a total training time of approximately 16.7 minutes. This model achieved the highest training accuracy of 73.27% and a validation accuracy of 76.17%, with both training and validation losses at 1.1963 and 1.0850, respectively. The 1D CNN's performance shows it learned the training data well and generalized effectively to the validation data.

7. Limitations of the 2D Parallel CNN: The 2D Parallel CNN trailed behind the other models in terms of accuracy. The model's simpler architecture and the absence of memory-based layers might contribute to this. It suggests that temporal modeling could be pivotal in understanding accented speech.

8. While the 4D Parallel CNN with Attention recorded the highest validation accuracy, it is essential to consider the computational cost and complexity of integrating attention mechanisms. For real-time applications, one might lean towards the standard 4D Parallel CNN due to similar accuracy but reduced complexity.

9. The disparity between training and validation accuracies, especially in the 2D Parallel CNN, indicates potential overfitting. Regularization techniques or more extensive datasets might address this.

10. The hybrid CNN-LSTM model presents a versatile framework. Such architectures might adapt better to varied data sources or multi-task learning scenarios, where both spatial and temporal features are vital.

11. In terms of validation accuracy, the 1D CNN approach exhibited robust performance, achieving a validation accuracy of 76.17%. This was comparable to the validation accuracies of the CNN-LSTM Hybrid and 4D Parallel CNN (with attention) approaches, which both achieved a validation accuracy of 76.49%. Additionally, the 1D CNN approach demonstrated competitive training and validation loss metrics, further highlighting its effectiveness for AASR tasks.

The comparative analysis reveals that the CNN-LSTM Hybrid model stands out for its high validation accuracy and efficient training time, making it an excellent choice for accented speech recognition tasks. Models incorporating attention mechanisms, such as the 4D Parallel CNN with attention, also show enhanced accuracy but at the cost of longer training times. The 1D CNN demonstrates a strong balance between accuracy and generalization, although training time optimization could further enhance its efficiency. These insights provide a comprehensive understanding of the relative strengths and weaknesses of each model, aiding in the selection of the most suitable architecture for specific application needs.

13.9 Conclusion

This chapter explored and compared various machine learning model architectures, including 4D Parallel CNNs with and without attention mechanisms, Bidirectional LSTM, a CNN-LSTM Hybrid, and a 2D Parallel CNN. Each model was evaluated based on its training and validation performance metrics, providing insights into their respective strengths and limitations.

This chapter demonstrates the importance of selecting the appropriate model architecture based on the specific requirements and constraints of the task. The CNN-LSTM Hybrid model stands out for its efficient training and robust performance, making it a recommended choice for similar applications. Future work could further optimize these architectures and explore additional enhancements, such as more advanced attention mechanisms or hybrid models, to continue improving performance and efficiency.

14. A Dual Approach to Detect Hate Speech in Accented Malayalam

14.1 Introduction

This research centers on adopting deep learning methods to effectively detect and differentiate hate speech patterns in accented Malayalam from audio data. The audio data underwent processing through a detailed feature extraction framework, encompassing ZCR, STFT, MFCC, RMS, and Mel Spectrogram. These methods collectively generated 162 distinct feature vectors capturing the unique acoustical variations of the speech. Acknowledging the scarcity of data tailored to specific regional accents, advanced data augmentation techniques were integrated to enhance the dataset's depth. Subsequently, a deep learning model was developed to train and classify these speech patterns. The findings revealed an impressive 98% accuracy in the model's evaluation after an exhaustive training phase.

1D CNN was selected for this experiment due to their suitability for processing sequential data, which is inherent in speech signals. By applying convolutional filters along the time axis, 1D CNNs excel at capturing temporal dependencies, allowing them to extract essential features such as phonemes and syllables.

1D CNNs offer computational efficiency compared to higher-dimensional CNNs, making them faster to train and more resource-efficient. Their proven effectiveness in various audio processing tasks, coupled with their ability to learn hierarchical features from speech data, makes them well-suited for hate speech recognition in accented Malayalam speech.

14.2 Hate Speech in Accented Malayalam

Hate speech is typically defined as any speech, gesture, conduct, writing, or expression that offends, threatens, or insults a particular person or group based on any attribute which can be gender, caste, religion, creed etc.

14.2.1 Situations that Qualify as Hate Speech

1. Direct Threats: Expressing intentions to inflict harm or violence against a particular individual or group.
2. Derogatory Language: Using slurs, insults, or derogatory terms that belittle a specific group or individual.
3. Incitement: Urging others to discriminate, harass, or enact violence against a group or individual.
4. Dissemination of Falsehoods: Spreading false information about a particular group to degrade them or incite hatred.

14.3 Non-Hate Speech in Accented Malayalam

Non-hate speech, in contrast, encompasses expressions that do not threaten, insult, or demean any individual or group, even if they may be critical or disagreeable.

14.3.1 Situations that Qualify as Non-Hate Speech

1. Constructive Criticism: Providing feedback or expressing disagreement without resorting to insults or derogatory terms.
2. Personal Opinions: Sharing personal experiences or viewpoints without the intent to harm or degrade any group or individual.
3. Factual Statements: Conveying information or news without any malicious or harmful intent.

4. Cultural Expressions: Using colloquialisms, proverbs, or traditional sayings that are inherent to Malayalam culture and do not demean any individual or group.

Distinguishing between hate speech and non-hate speech, especially in a language as complex as Malayalam, requires a comprehensive understanding of cultural, contextual, and linguistic factors. The dataset sourced from YouTube offers a valuable insight into the manifestation of these expressions in the digital age.

14.4 Data Collection

The foundation of any robust machine learning or deep learning model lies in the quality and diversity of data it's trained on. For this study on hate speech detection in accented Malayalam speech, a comprehensive data collection journey was embarked to ensure a representative and unbiased dataset and named it as Accented Malayalam Dataset for Detecting Hate Speech (AMDDHS). Samples that belong to hate speech and non-hate speech involving accents from different districts like Kasaragod, Kannur, Malappuram, Kozhikode, Thrissur, Kottayam, and Thiruvananthapuram were considered in the study.

14.4.1 Criteria for Data Selection

1. Nature of Speech: The data collection was limited to only those clips where speech could be distinctly categorized as either hate speech or non-hate speech. Ambiguous speech or those that tread the fine line between the two categories were excluded to maintain clarity and purpose.

2. Accent Diversity: A significant emphasis was placed on ensuring representation of diverse Malayalam accents. This was crucial to ensure that this study was comprehensive and considered regional variations in speech.

14.4.2 Data Extraction Process

1. Manual Review: A set of YouTube videos have been carefully examined before the data collection.

2. Audio Extraction: From the shortlisted videos, audio clips were extracted and segmented to ensure each clip represented a distinct speech sample.

3. Annotation: Post extraction, each audio clip was annotated with relevant metadata, including the nature of speech (hate/non-hate) and the specific accent. This metadata played a crucial role during the model training phase.

14.4.3 Data Distribution

A total of 1,000 samples were extracted. Out of these, 150 samples were hate speech, and 850 samples were non-hate speech. This distribution provided a clear overview of the prevalence of hate speech in the sampled content. Figure 92 provides the statistical view of the original and augmented hate speech dataset. The wave plot and spectrogram of hate and non-hate speech is illustrated in Figure 93.

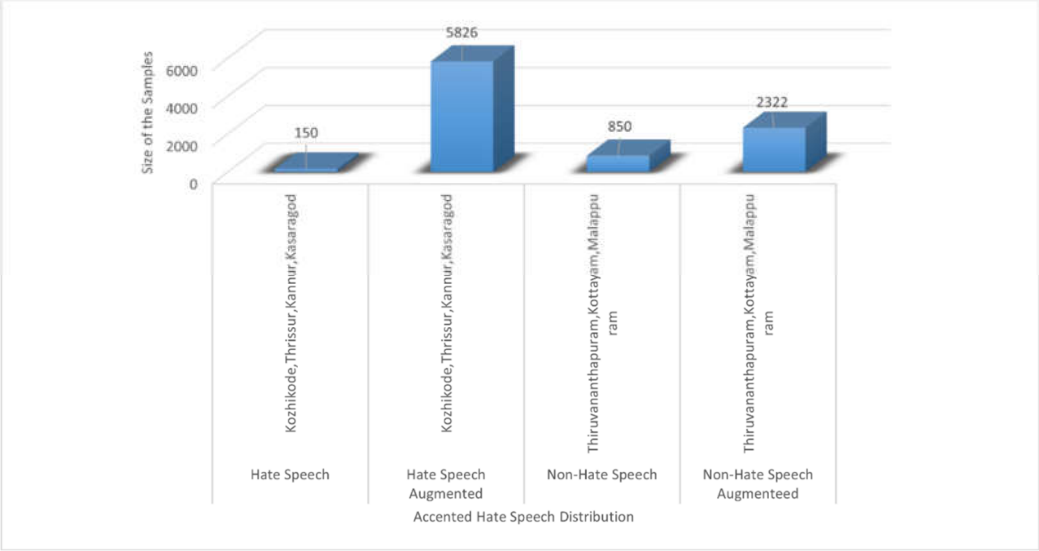


Figure 92 Statistics of the Hate Speech Dataset

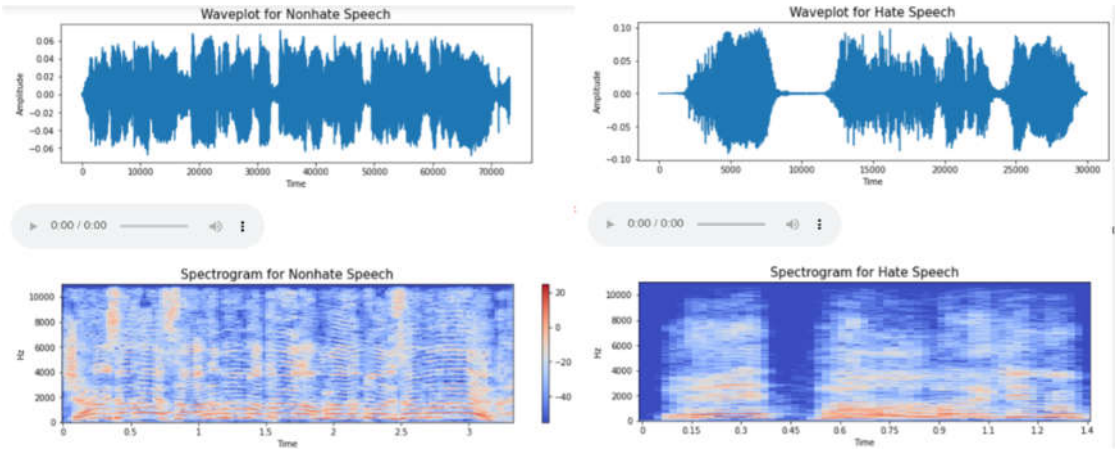


Figure 93 Wave plot And Spectrogram of the Speech Data (Sample)

14.5 Data Augmentation Techniques for Speech Data

By introducing variations in the original dataset, the model becomes more versatile and can generalize better to unseen data. In this study on accented Malayalam speech recognition, several augmentation techniques were used to enrich the dataset and are discussed in the subsequent sections. Figure 94 provides a representation of sample signal after applying speech augmentation techniques.

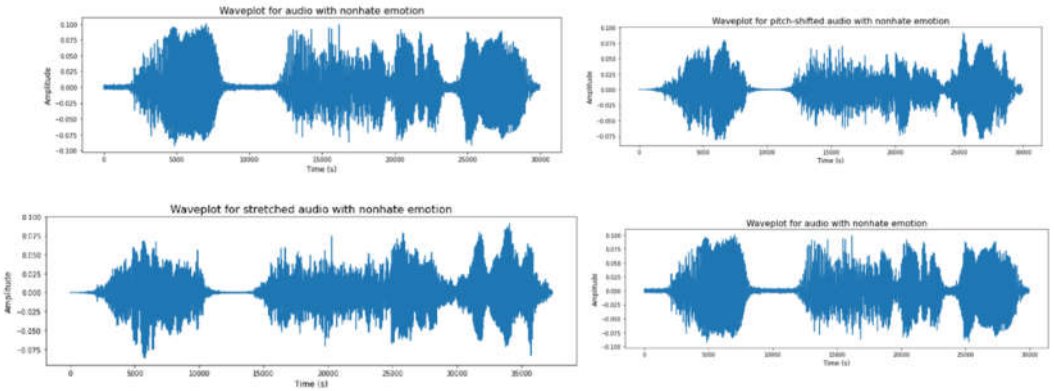


Figure 94 The Sample Signals After Augmentation

14.5.1 Noise Injection

The noise function introduces random noise to the original audio data. The amplitude of this noise is determined based on a fraction of the maximum amplitude in the original data. This augmentation mimics real-world scenarios where

background noise can be a factor. A random noise amplitude is generated, limited to 3.5% of the maximum amplitude of the data. This noise is then added to the original data.

14.5.2 Time Stretching

The stretch function alters the speed of the audio playback without affecting its pitch. It uses the time_stretch function and the audio data are stretched by a given rate. A rate less than 1 slows down the audio, whereas a rate greater than 1 speed it up. A rate of 0.8 is used in the study, which means the audio will be played back slower than its original speed.

14.5.3 Time Shifting

The shift function shifts the speech waves in the dataset to the left or right by a random amount. A random shift range is calculated, which can vary between -5,000 to 5,000 samples. The audio data is then rolled (or shifted) by this range. This introduces a delay or advances the speech signal.

14.5.4 Pitch Shifting

The pitch of the audio data is shifted by a specified number of half-steps. A positive n_steps value increases the pitch, while a negative value decreases it.

14.5.5 Implementation

Data augmentation is crucial for enhancing the diversity and richness of the training dataset. By introducing variations like noise, time shifts, and pitch changes, the model becomes better equipped to handle real-world challenges in speech recognition. The acoustic model [113] establishes a connection between speech signals and phonetic units, taking feature vectors as input and producing a sequence of phonemes in word form. These techniques, when applied judiciously, can significantly improve model performance, especially in scenarios with diverse accents and varying recording conditions.

14.6 Feature Extraction Techniques

In speech recognition and audio processing, the goal of feature extraction is to distill the raw audio data into a compact but descriptive format, capturing the inherent structures and patterns. This chapter describes the various feature extraction techniques employed to understand accented Malayalam speech data. The model, designed to tackle the challenges posed by diverse accents in Malayalam, begins its journey with the extraction of 162 distinctive features from each speech data.

14.6.1 Zero Crossing Rate (ZCR)

ZCR measures the rate at which a signal changes its sign, effectively capturing the frequency characteristics of a voice signal. In speech processing, ZCR is used to detect voiced/unvoiced decisions and can differentiate between speech and non-speech segments, making it valuable for the dataset. ZCR measures the rate at which a signal changes its sign, effectively capturing the frequency characteristics of a voice signal. In speech processing, ZCR is used to detect voiced/unvoiced decisions and can differentiate between speech and non-speech segments, making it valuable for the dataset.

14.6.2 Chroma_STFT

Chroma features are instrumental in understanding harmonies, chords, and the general tonal quality of an audio clip. Given the diverse accents in Malayalam speech data, the harmonic structures can vary widely, making this feature crucial. Given the diverse accents in Malayalam speech data, the harmonic structures can vary widely, making this feature crucial.

14.6.3 Mel-frequency Cepstral Coefficients (MFCC)

They consider the non-linear human ear perception of frequencies, making them highly suitable for speech and audio processing. MFCCs capture the timbral texture of the sound and variations in accents can lead to variations in timbral differences, which MFCCs can capture efficiently.

14.6.4 Root Mean Square Value (RMS)

The RMS value indicates the magnitude of the audio signal, essentially representing the loudness of a signal. RMS can be pivotal in understanding the energy and intensity variations in speech, which can be characteristic of different accents or emotional tones in speech. The RMS value indicates the magnitude of the audio signal, essentially representing the loudness of a signal. RMS can be pivotal in understanding the energy and intensity variations in speech, which can be characteristic of different accents or emotional tones in speech.

14.6.5 Mel Spectrogram

The Mel spectrogram captures the frequency domain information of the audio while also accounting for human auditory perception. This ensures that the features align closely with how humans perceive sound, making it an asset in speech recognition tasks. It captures the frequency domain information of the audio while also accounting for human auditory perception. This ensures that the features align closely with how humans perceive sound, making it an asset in speech recognition tasks. To ensure a robust model, features were extracted from both the original data and augmented versions of it (i.e., with noise, stretched, and pitched). The get_features function is designed to first extract features from the raw data and then from its augmented forms.

Each set of features is stacked vertically to the main result, ensuring a comprehensive feature set ready for training. Upon completion of the feature vectorization process, the model generates 162 distinct feature vectors for each speech data sample. These vectors encapsulate the unique acoustical variations of the speech, providing a rich representation that is well-suited for subsequent stages of the recognition model. Figure 95 provides an overview of the statistics regarding feature extraction using different techniques.

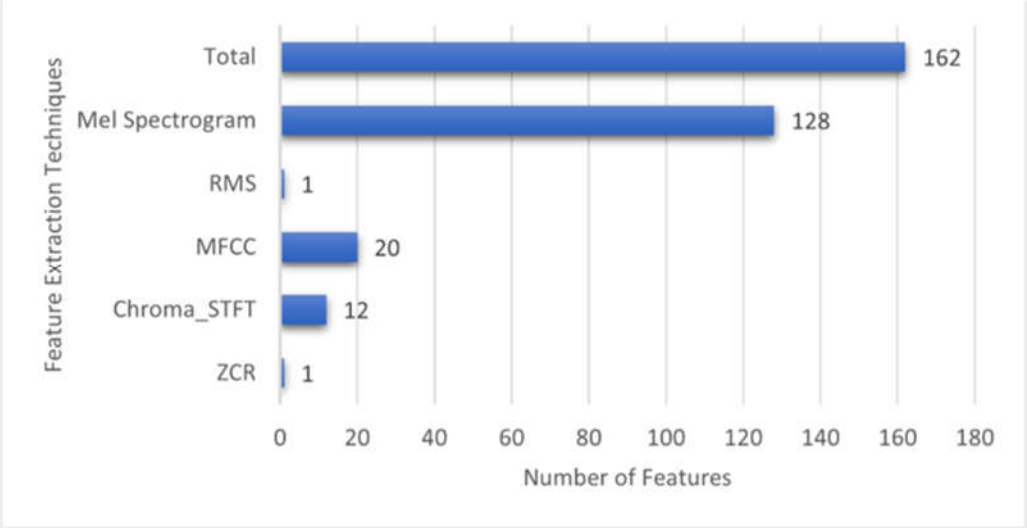


Figure 95 Features Extraction Statistics

14.6.6 The pseudocode for Data Augmentation

BEGIN
FUNCTION noise(data):
// Introduce random noise to the audio data
*SET noise_amplitude TO 0.035 * RANDOM () * MAX (data)*
*SET augmented_data TO data + noise_amplitude * RANDOM_NORMAL (size=LENGTH (data))*
RETURN augmented_data
END FUNCTION
FUNCTION stretch (data, rate=0.8):
// Stretch the audio data by a given rate using librosa's time_stretch function
SET stretched_data TO time_stretch(data, rate)
RETURN stretched_data
END FUNCTION
FUNCTION shift(data):
// Shift the audio signal by a random amount
*SET shift_range TO RANDOM_INT (-5, 5) * 1000*
SET shifted_data TO roll (data, shift_range)
RETURN shifted_data
END FUNCTION
FUNCTION pitch (data, n_steps, sr):
// Modify the pitch of the audio data using librosa's pitch_shift function
SET pitched_data TO pitch_shift (data, sr, n_steps)
RETURN pitched_data
END FUNCTION
//Implementation
SET path TO sample_data_path [1]

LOAD data, sample_rate FROM path USING librosa.
// Augment data using the above functions
CALL noise(data)
CALL stretch(data)
CALL shift(data)
CALL pitch (data, n_steps, sample_rate)
END

14.6.7 Pseudocode for Feature Extraction

BEGIN
FUNCTION extract_features (data, sample_rate):
// Initialize an empty array for results
result = NEW EMPTY ARRAY
// Extract Zero Crossing Rate
zcr = AVERAGE (zero_crossing_rate(data))
APPEND zcr TO result
// Extract Chroma_STFT
stft = ABSOLUTE (short_time_fourier_transform(data))
chroma_stft = AVERAGE (chroma_stft(stft, sample_rate))
APPEND chroma_stft TO result
// Extract MFCC
mfcc = AVERAGE (mfcc(data, sample_rate))
APPEND mfcc TO result
// Extract Root Mean Square Value
rms = AVERAGE (rms(data))
APPEND rms TO result
// Extract Mel Spectrogram
mel = AVERAGE (melspectrogram(data, sample_rate))
APPEND mel TO result
RETURN result
END FUNCTION
FUNCTION get_features(path):
// Load audio data from the path
data, sample_rate = LOAD AUDIO FROM path WITH PARAMETERS (duration=2.5, offset=0.6)
// Extract features from original data
res1 = extract_features(data, sample_rate)
result = NEW ARRAY WITH res1
// Apply noise augmentation
noise_data = ADD NOISE TO data
res2 = extract_features(noise_data, sample_rate)
APPEND res2 TO result
// Apply time stretching and pitch shifting

new_data = STRETCH data
data_stretch_pitch = PITCH SHIFT new_data USING sample_rate AND n_steps=0.7
res3 = extract_features(data_stretch_pitch, sample_rate)
APPEND res3 TO result
RETURN result
END FUNCTION
END

14.7 Methodology

To ensure a robust model, features were extracted from both the original data and augmented versions of it (i.e., with noise, stretched, and pitched). The get_features function is designed to first extract features from the raw data and then from its augmented forms. Each set of features is stacked vertically to the main result, ensuring a comprehensive feature set ready for training.

The model is constructed with multiple 1D convolutional layers with varying filter sizes, followed by max-pooling layers to extract hierarchical features from the input data. Dropout layers are strategically inserted to mitigate overfitting during training. The final layers include a flatten layer to transform the 3D output into a 1D array, followed by densely connected layers with rectified linear unit (ReLU) activations.

The output layer, softmax function is used for multiclass classification, with the number of neurons adjusted to the specific number of classes. Additionally, a ReduceLROnPlateau callback is defined to dynamically adjust the learning rate during training, contributing to model stability. The model is compiled using the Adam optimizer, categorical crossentropy as the loss function, and accuracy as the metric for evaluation.

The vectors extracted from the speech signals are fed as input to the CNN architecture, to the initial conv1d_8 layer. Recognizing the scarcity of labeled data tailored to specific regional accents, advanced data augmentation techniques have been seamlessly integrated into the training pipeline. This augmentation process enhances the depth and diversity of the dataset, further enriching the feature vectors and contributing to the model's robustness.

As the model progresses through the convolutional layers and subsequent fully

connected layers, the 162 features play a pivotal role in shaping the learned representations. The hierarchical nature of the CNN architecture that is illustrated in Figure 96 ensures that these features contribute to capturing complex patterns and variations present in the accented speech data.

The initial convolutional layer, conv1d_8, utilizes 256 filters to learn distinct features from the input data, reflected in its 1,536 parameters. Following this, the max_pooling1d_8 layer reduces spatial dimensions by capturing the maximum values over a specified window, aiding computational efficiency, and providing a form of translational invariance.

The architecture continues with a second convolutional layer, conv1d_9, employing an additional 256 filters, and subsequent max pooling, contributing to hierarchical feature extraction. This pattern repeats with a third convolutional layer, conv1d_10, using 128 filters, and corresponding max pooling for further feature refinement.

The model concludes with an output layer, dense_5, containing two neurons for binary classification, typically using a softmax activation function. This 1D CNN model, designed for sequence data, exhibits an appropriate structure for a two-class classification task, having a total of 557,090 trainable parameters. Figure 97 represents the model architecture for hate speech detection.

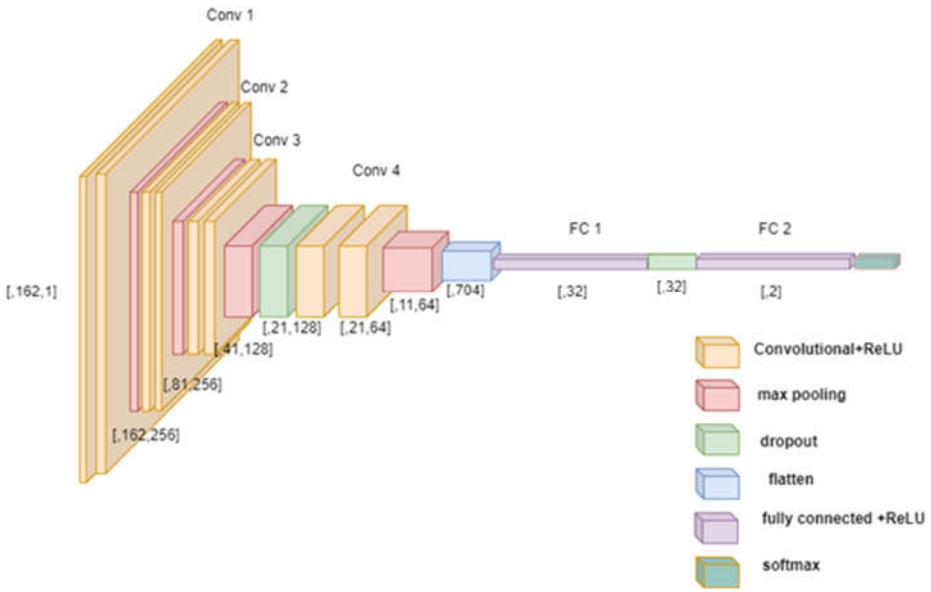


Figure 96 The Layered Architecture

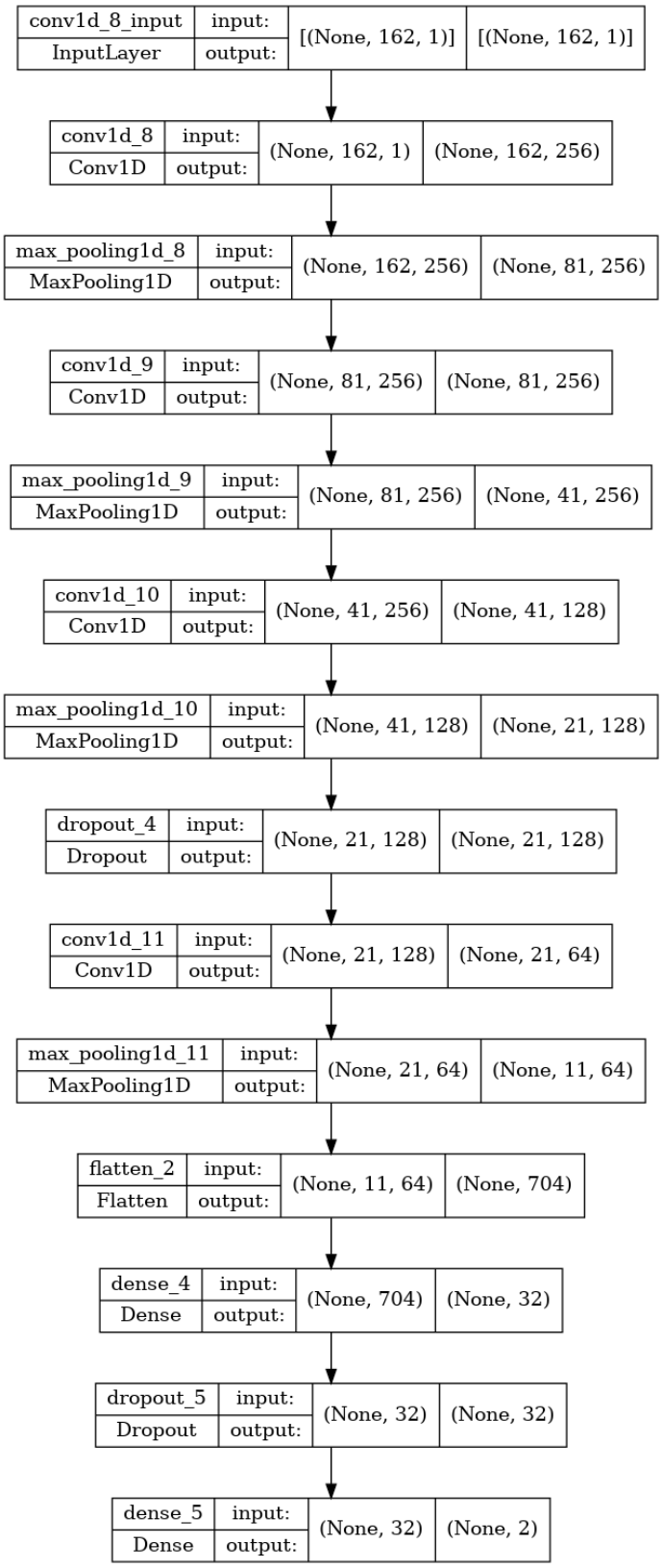


Figure 97 The Hate Speech Detection Model Architecture

14.8 Performance Evaluation

The model was trained over a total of 5 epochs. By the fifth epoch, it exhibited significant convergence characteristics, indicating that the training was effective, and the model was learning from the provided dataset. Each epoch consisted of 96 batches of data. On average, each batch took about 172 milliseconds to process, amounting to a total epoch time of 17 seconds. The efficiency of batch processing ensures that the model can be retrained or fine-tuned rapidly if required, which is a boon for iterative model development. Figure 98 depicts the learning curves obtained while constructing the model to classify hate and non-hate speech classes.

14.9 Training Performance

During the fifth epoch, the model achieved a training loss of 0.0843. The term loss refers to the objective function value the model aims to minimize. A lower loss suggests that the model's predictions are aligning well with the actual data. Moreover, the model attained a training accuracy of 97.32%.

14.9.1 Validation Performance

To ensure that the model isn't just memorizing the training data (a phenomenon called overfitting), it is vital to test its performance on a separate set of data it hasn't seen during training. For this purpose, validation metrics are essential. In the fifth epoch, the model showcased a validation loss of 0.0754, which is even lower than the training loss, pointing towards the model's robustness. The validation accuracy was 98.18%, indicating that the model generalizes well to new data and has a high predictive capability.

14.9.2 Learning Rate

The learning rate during this epoch was set at 0.0010. This parameter determines the step size the model takes to adjust its weights in the direction of optimal performance. A carefully chosen learning rate ensures that the model converges efficiently without overshooting or getting stuck.

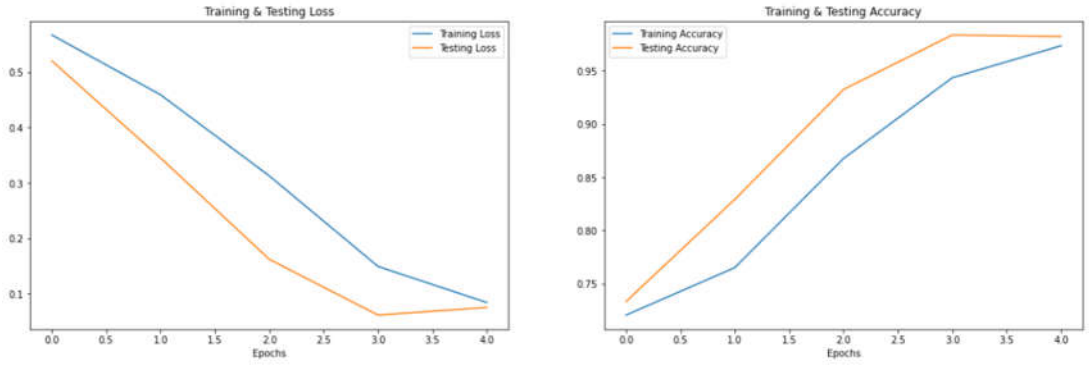


Figure 98 The Learning Curves

The model exhibits strong performance both in training and validation phases. With a validation accuracy surpassing 98%, it indicates a well-optimized model with high predictive power. The differences between training and validation metrics are minimal, suggesting that the model is neither overfitting nor underfitting.

14.10 Performance Evaluation

The classification task revolves around distinguishing between two critical categories: hate and non-hate. One of the key performance metrics employed to gauge the model's proficiency is precision. Figure 99 illustrates the confusion metrics that has been generated to check how well the model performs the classification task. The hate category has an impressive precision score of 0.98 signifies that, out of all instances that the model confidently tagged as hate, a whopping 98% were indeed correctly classified. And for the non-hate category, the model's precision stood at 99%, indicating that a remarkable 99% of instances earmarked as "non-hate" aligned with the ground truth.

The numbers reveal that the model is particularly strong at identifying hate speech, with a very high true positive rate. However, there's a small margin of error in identifying non-hate speech, given by the 32 False Positives. In the context of this study, this could mean that the model might sometimes be a little over-cautious, classifying some non-hate speech as hate. A detailed analysis of the performance of

the model in terms of accuracy, micro average and weighted average is shown in Figure 100.

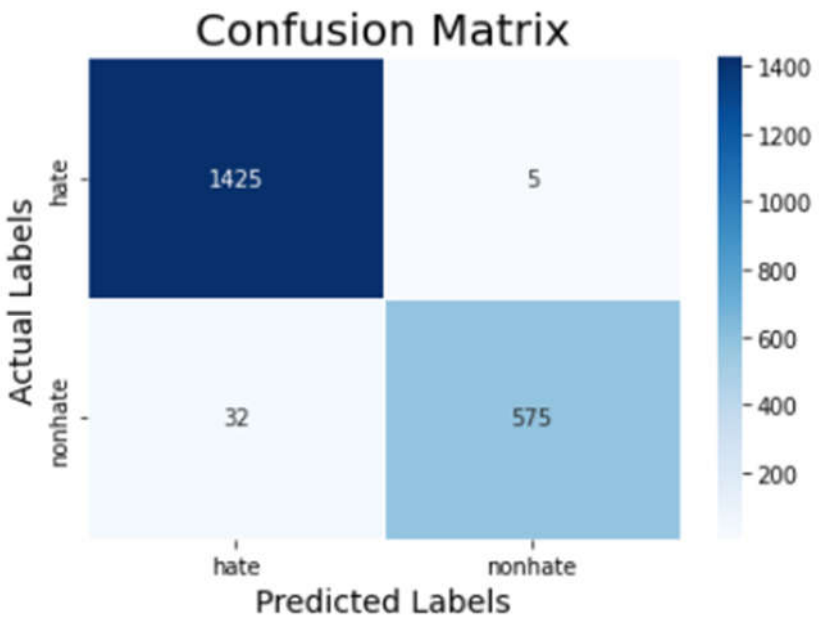


Figure 99 The Confusion Matrix

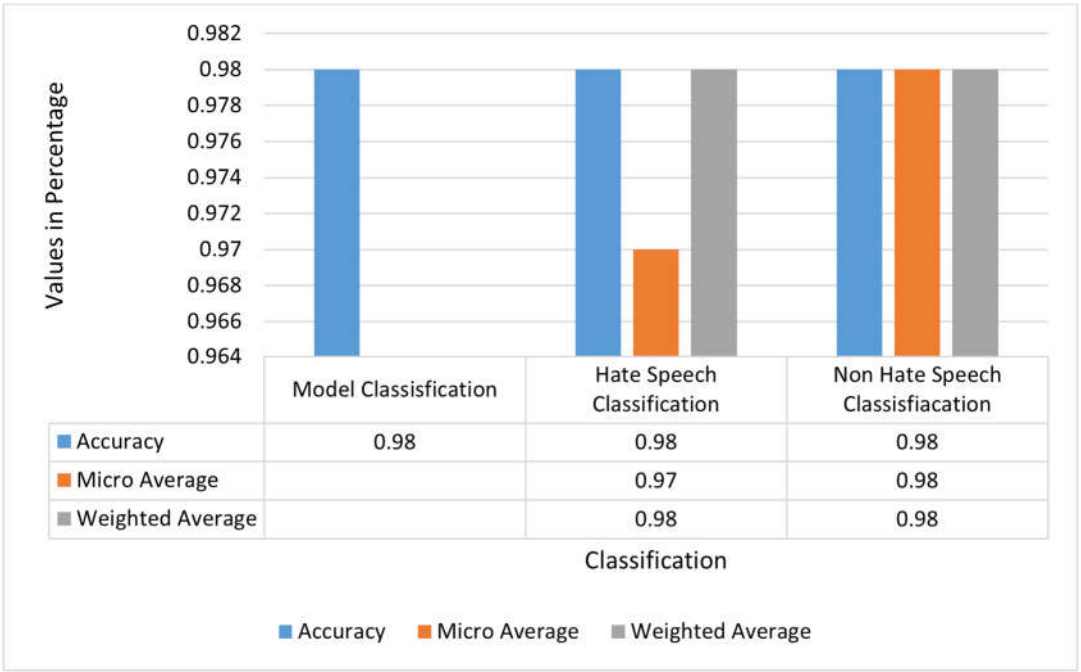


Figure 100 Hate Speech Classification Performance

14.11 Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that converts high-dimensional data into fewer dimensions, prioritizing those that maximize variance. In this case, the data is reduced to two principal components, represented on the x and y axes of the scatter plot. The dataset comprises two classes, hate and non-hate, visually differentiated by red and green colors, respectively. The resultant scatter plot displays these two classes in the context of the first and second principal components.

Figure 101 illustrates the two primary clusters are the red cluster for hate and green one for non-hate. This clustering suggests a certain degree of separation between the two classes in this reduced space. However, there is a noticeable overlap between the clusters, indicating challenges in distinguishing hate from non-hate speech based solely on these principal components.

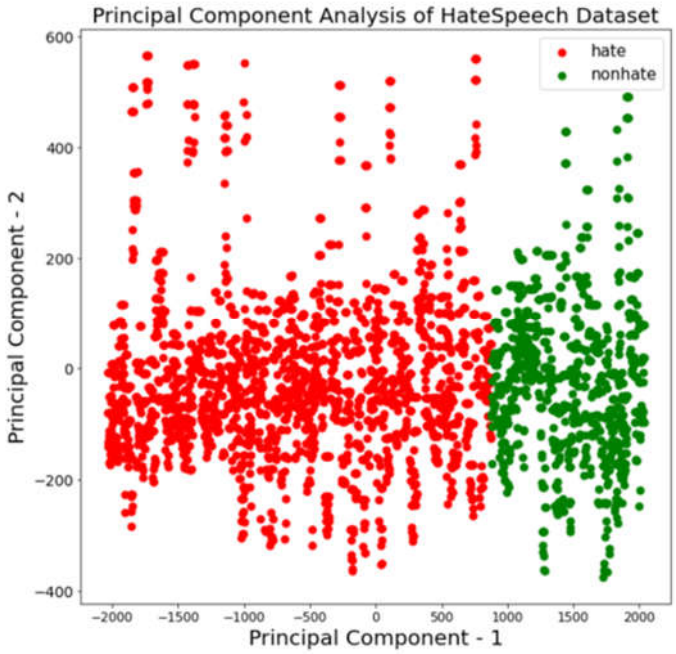


Figure 101 Reduced Space Formed by PCA

The data points are colored based on their actual labels, allowing for visual inspection of how well-separated or clustered the two classes are in this reduced dimensional space. For hate speech detection in accented Malayalam speech data,

the PCA visualization provides an initial, high-level insight into the distribution and potential separability of hate and non-hate data points.

In the context of this study on detecting hate speech in accented Malayalam speech data, the model appears to be performing very well, especially in identifying hate speech. However, there is a slight challenge in classifying non hate speech, which might be attributed to complexities in the accented Malayalam language or the nature of the training data.

Future refinements could focus on minimizing these false positives to make the model even more accurate. In future iterations, one might consider experimenting with different model architectures, hyperparameters, or introducing more data augmentations to further enhance the model's performance. This model provides a reliable foundation for the task at hand. This performance evaluation provides a snapshot of the model's capabilities and areas of strength. It can be further elaborated or refined based on additional context or specific requirements.

14.12 Conclusion

This chapter explored the integration of advanced feature extraction techniques and sophisticated model architectures to enhance hate speech recognition in the context of accented Malayalam speech. By extracting key features such as ZCR, Chroma_STFT, Mel-frequency Cepstral Coefficients MFCC, RMS, and Mel Spectrograms, the intricate acoustic and harmonic properties of speech signals were captured. These features, along with data augmentation techniques, provided a robust dataset for training models.

This comprehensive approach allowed for the successful detection and classification of hate speech, demonstrating high accuracy and robustness across different speaking conditions and accents. The advanced data augmentation techniques ensured that the model could generalize well, making it resilient to variations in the input data.

The findings of this chapter contribute significantly to the field of AASR, in the detection of hate speech. The methodologies and techniques discussed provide a strong foundation for future research, offering potential pathways for further enhancements and applications in various speech recognition tasks. This work focuses on the importance of combining robust feature extraction with sophisticated modeling techniques to address complex problems in speech processing and recognition.

15. Results and Discussion

15.1 Experiment 1

The results of Experiment 1 discussed in chapter 5 provide significant insights into the development of an Accented AASR system for isolated Malayalam words using the LSTM-RNN algorithm. The study was conducted using the AMSC-1 dataset, which was specifically constructed for this research. The results demonstrate that the AASR system achieved an accuracy of 82.5%, with micro precision, recall, and F1-score all consistently at 82%. This consistency at the micro level indicates a balanced performance across individual instances.

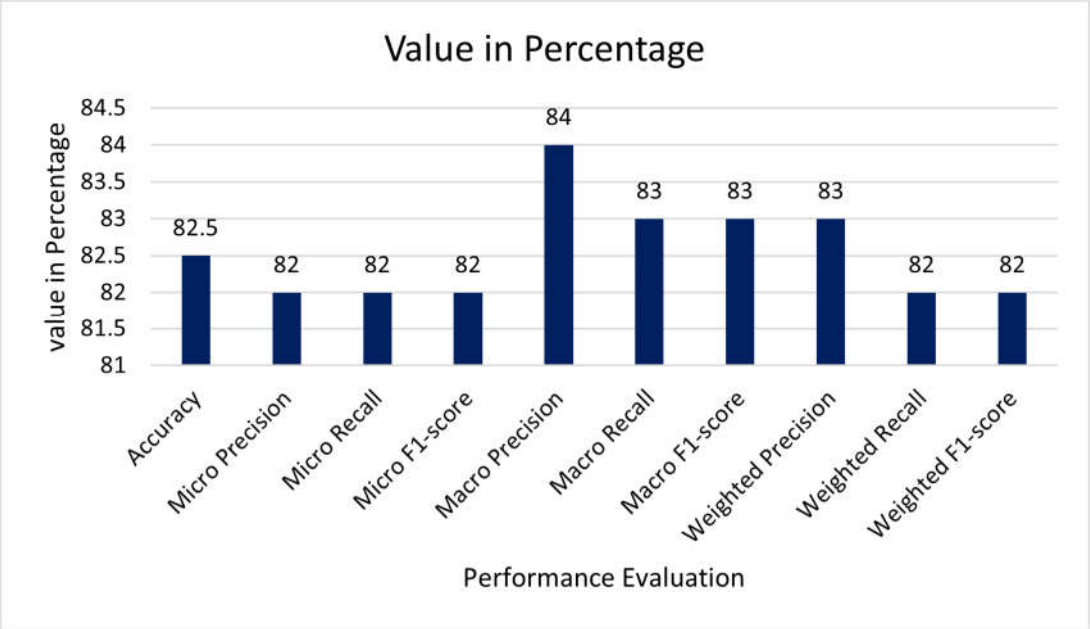


Figure 102 Performance Evaluation

At the macro level, the performance metrics slightly improved, with precision at 84%, recall at 83%, and F1-score at 83%. This suggests that the AASR system performs better for larger classes, which is a positive outcome for practical applications where certain words or phonetic structures may be more prevalent. Additionally, the weighted metrics, which account for class imbalances, show precision, recall, and F1-scores of 83%, 82%, and 82%, respectively. These results further validate the

robustness and reliability of the system in handling imbalanced datasets, a common challenge in speech recognition tasks.

The constructed AASR system demonstrates a strong capability in recognizing accented speech in Malayalam, a language characterized by its diverse regional accents and phonetic intricacies. By achieving high precision, recall, and F1-scores, the system confirms its effectiveness in accurately identifying and processing accented speech. The scope of this study, focused on twenty isolated word classes, represents a foundational step towards more comprehensive and context-aware speech recognition systems for Malayalam. Although limited in scope, the research stresses the potential of integrating linguistic diversity within speech recognition technologies, which is crucial for preserving and promoting regional accents and dialects.

In summary, the results of Experiment 1 illustrated in Figure 102 indicate the feasibility and effectiveness of using LSTM-RNN for constructing AASR systems for Malayalam. The balanced and high performance across various metrics indicates that the developed system is well-suited for practical applications, paving the way for further advancements in accented speech recognition for regional languages.

15.2 Experiment 2

The results of Experiment 2 discussed in chapter 6 present the contributions of an extensive study focusing on the construction and evaluation of AASR systems using various machine learning and deep learning approaches. The study utilizes the AMSC-2 dataset, which was created specifically for this research to represent accent-based Malayalam speech data. The hyper parameter tuning had varying effects on the different models. While some models, like MLP Classifier, Random Forest, KNN, LSTM-RNN, and CNN, showed substantial improvement, others, such as SVM and SGD, demonstrated either minimal improvement or sometimes declined in performance. The performance evaluation of the model is illustrated in Figure 103.

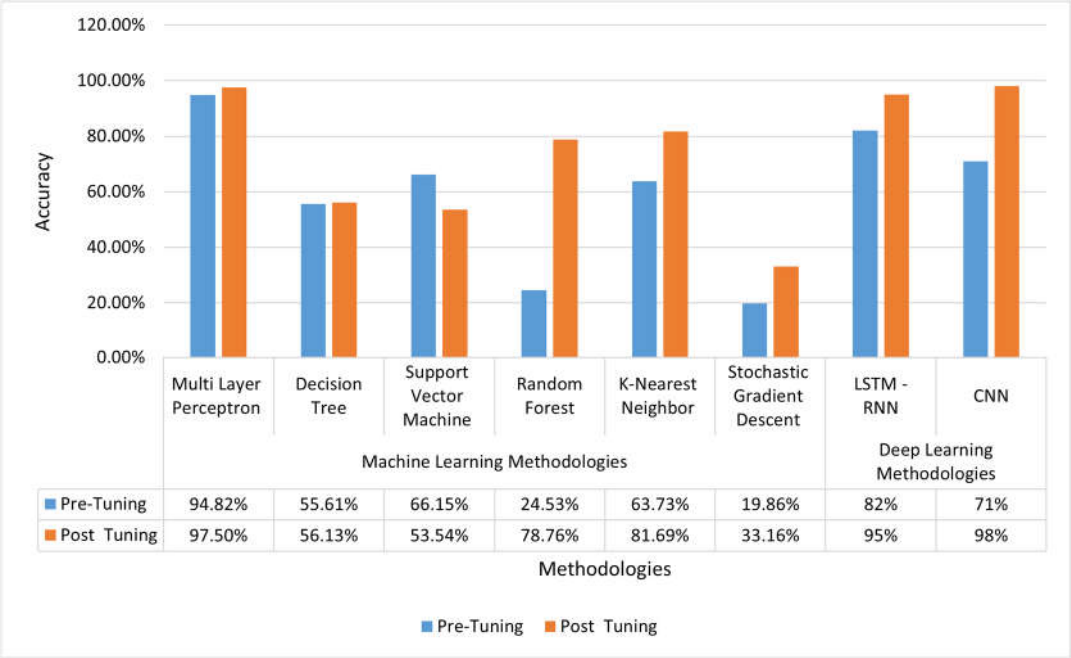


Figure 103 Performance Evaluation of Experiment 2

The AMSC-2 dataset was prepared to include diverse accents within the Malayalam language. Feature engineering involved multiple techniques such as MFCC, STFT, and Mel Spectrogram methods to extract optimal representations of the speech data.

Several machine learning algorithms were employed to construct AASR models, including MLP, Decision Tree, SVM, Random Forest, KNN, and SGD. Among these, the MLP model achieved the highest accuracy of 94.82%, effectively capturing the variations in Malayalam accents. The Decision Tree and SVM models produced lower accuracies of 55.67% and 66.15%, respectively, reflecting the challenges in modeling accented speech with these approaches. The Random Forest model achieved an accuracy of 78.76%, and the KNN model reached 81.69% after hyperparameter tuning. The SGD model, while powerful, yielded a lower accuracy of 33.16%, indicating the sensitivity of this method to feature selection and tuning.

The LSTM-RNN model demonstrated significant potential in accented speech recognition, particularly for Malayalam. Utilizing 180 prominent features of speech data, the LSTM-RNN model underwent rigorous training involving 98,000 steps. The training process showed a substantial improvement from an initial loss of 2.5 to a

final loss of 0.24, achieving a remarkable training accuracy of 95%. The validation phase confirmed the model's effectiveness with an accuracy of 82%, indicating its capacity to handle multi-accent speech classification by accurately modeling the temporal dependencies in speech signals.

Deep CNNs were employed to process spectrograms of accented speech, treating them as images. The CNN models were designed to utilize the intricate spectrographic details of audio signals. The training process spanned 53,000 steps over 4,000 epochs, with the model leveraging 80% of the data for training and 20% for testing. This sophisticated architecture, incorporating convolutional, pooling, and dense layers, effectively identified complex features that characterize different accents. The model achieved a training accuracy of 98% and a test accuracy of 71%, highlighting the effectiveness of CNNs in accent recognition.

An ensemble approach combining all individual models was developed to enhance the overall performance of the AASR system. This ensemble method achieved an accuracy of 71.55%, indicating the potential benefits of integrating multiple models to utilize their complementary strengths.

The findings from Experiment 2 provide valuable insights into the capabilities and limitations of various approaches in Malayalam accented speech recognition. The high training accuracy of the LSTM-RNN and CNN models highlights their potential in learning complex features of accented speech. However, the lower test accuracy in the CNN model indicates challenges in generalization, suggesting potential overfitting. The machine learning models, particularly the MLP, showed strong performance, while other models highlighted the difficulties in capturing speech complexities. This study advances the understanding of accented speech recognition and sets a foundation for future research in this under-explored area. The results emphasize the importance of balancing model complexity and generalization to enhance the practical applicability of AASR systems.

15.3 Experiment 3

The results of Experiment 3 discussed in chapter 7 discuss the methods that were systematically developed to enhance the recognition accuracy of accented Malayalam speech using two datasets the AMSC-3 and AMSC-4. These phases progressed from basic feature extraction techniques to complex combinations of multiple methods, capturing a comprehensive set of features that reflect the detailed characteristics of Malayalam accented speech. The experiment is conducted in two phases using these datasets. The first phase used the AMSC-3 dataset, and the second phase used the AMSC-4 dataset. The evaluation of two phases is discussed below:

15.3.1 Experiment based on AMSC-3

The performance metrics used for evaluation include the WER and MER, which indicate the percentage of incorrectly predicted words and misrecognized speech segments, respectively. Additionally, the accuracy metrics for LSTM-RNN, namely Raw Accuracy and Validation Accuracy, were employed to assess the model's performance on the training and validation dataset.

Figure 104 illustrates the performances of different phases of the study using AMSC-dataset. Phase I involved extracting 13 frequency coefficients from the speech signal, along with their second and third derivatives, yielding a total of 40 coefficients. This phase provided a foundational representation of the audio data but was limited in capturing time-varying features. The average WER and MER for this phase were 37% and 49%, respectively, highlighting the need for more sophisticated methods to reduce errors.

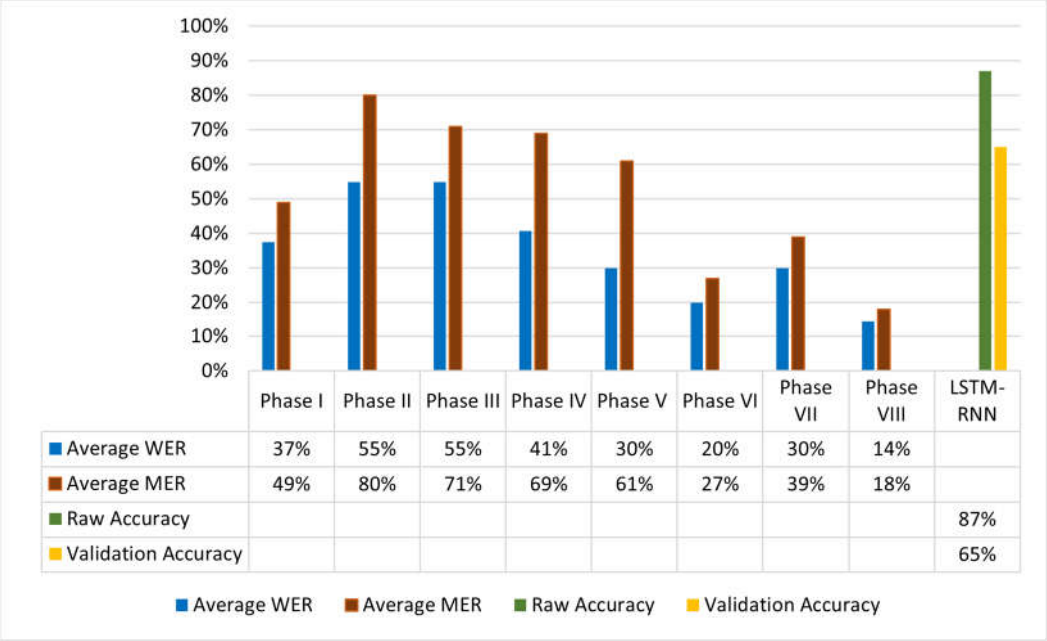


Figure 104 Performance Evaluation using AMSC-3

Phase II extracted 12 prominent amplitude values for time-frequency decomposition. While STFT offered a time-localized view of frequency variation, it struggled with high error rates due to its sensitivity to noise and the complexity of speech signals. The average WER and MER were 55% and 80%, respectively, indicating the limitations of STFT in handling accented speech data effectively.

Phase III focused on capturing the rhythmic aspects of speech by extracting 384 features. This phase analyzed the rhythmic characteristics of the speech signal but did not significantly reduce error rates compared to STFT alone, with average WER and MER remaining at 55% and 71%, respectively. This suggests that while rhythm is an important feature, it alone is insufficient for improving recognition accuracy.

Phase IV involved extracting 128 features, providing deeper insights into the speech data. This phase showed improvement in WER, reflecting better representation of spectral content relevant to human auditory perception. The average WER and MER

were 41% and 69%, respectively, demonstrating the effectiveness of Mel Spectrogram techniques in capturing hidden characteristics of the speech data.

Phase V involved the combination of MFCC and STFT created a comprehensive vector representation by integrating both methods. This combination improved both WER and MER by capturing more detailed time-frequency aspects of the speech signal. The average WER and MER for this phase were 30% and 61%, respectively, showing significant improvement over the individual methods.

Phase VI involved the combination of MFCC and Tempogram significantly enhanced the understanding of frequency and rhythm in speech, leading to the lowest error rates observed up to this phase. By combining these features, the average WER and MER were reduced to 20% and 27%, respectively, highlighting the effectiveness of integrating frequency coefficients and rhythmic patterns.

Phase VII involved the combination of MFCC and Mel Spectrogram balanced and enriched the speech signal representation by integrating MFCC and Mel Spectrogram features. Although this phase did not surpass the performance of Phase VI, it still provided a well-rounded analysis of the speech data. The average WER and MER were 30% and 39%, respectively.

Phase VIII involved the combination of MFCC, STFT, Tempogram, and Mel Spectrogram was the final phase, combining all previous feature sets to create a comprehensive and robust representation of accented Malayalam speech. This phase achieved the lowest WER and MER, with averages of 14% and 18%, respectively, demonstrating the effectiveness of integrating diverse feature extraction methods. This comprehensive model captured the complexity and uniqueness of Malayalam accented speech, resulting in the most accurate recognition performance.

The performance of the LSTM-RNN model was evaluated with a Raw Accuracy of 87% and a Validation Accuracy of 65%. The high raw accuracy on the training set but notable drop in validation accuracy indicates potential overfitting, where the model

performs well on training data but less so on unseen data. Further tuning and regularization techniques may be required to enhance generalization and improve the validation accuracy.

Practical challenges in feature extraction methods such as Short-Time Fourier Transform (STFT), Tempogram, and Mel Spectrogram include selecting appropriate parameters like window size and type for STFT to balance time and frequency resolution, which impacts computational complexity. Tempogram computation requires accurate segmentation and windowing for rhythmic pattern analysis, posing difficulties in handling varying tempo and rhythm. Mel Spectrogram computation aims to map the power spectrum onto the Mel scale but faces challenges with non-stationary signals and interpreting features amidst noise. Robustness to noise and artifacts is a common challenge across all methods, demanding empirical experimentation and algorithmic optimization for reliable feature extraction. These challenges emphasize the importance of domain expertise and careful parameter tuning to ensure accurate representation of audio signals for subsequent processing tasks.

The phased approach to feature engineering, culminating in the integration of multiple methods, effectively reduced error rates and enhanced the recognition accuracy of accented Malayalam speech. The results feature the importance of comprehensive feature representation and the need for careful parameter tuning and noise management in speech recognition systems. The performance of the LSTM-RNN model highlights areas for further improvement, particularly in addressing overfitting and enhancing validation accuracy.

15.3.2 Experiment Based on AMSC-4

The results of this study reveal several key insights into the effectiveness of different feature extraction methods and classifiers in the recognition of accented Malayalam speech. This analysis focuses on the variations in WER and classification accuracy across different phases of the feature engineering process.

15.3.2.1 Word Error Rate (WER) Analysis

In Phase I, the MLP and ensemble methods demonstrated the lowest WER, both around 0.52%, indicating a strong initial performance. This suggests that these models are well-suited to the initial dataset configuration. Conversely, the high WERs of the Decision Tree and SGD models, with SGD performing at 68.74%, indicate significant room for improvement and suggest that these models were less effective with the initial feature set.

The WER increased for all models in Phase II, likely due to changes in the training process, data augmentation, or feature extraction techniques. This phase highlights the sensitivity of the models to alterations in input data and the need for further tuning to adapt to new conditions. The KNN model's WER soared to 99%, indicating it struggled the most with these changes. The ensemble method's increased WER reflects the difficulty in combining poorly tuned individual models. Table 24 illustrates the performance evaluation in terms of WER generated in the study.

Table 24 Performance in WER

Phases	MLP	Decision Tree	SVM	RFC	KNN	SGD	Ensembled
Phase I	0.52%	47.85%	39.72%	19.52%	18.31%	68.74%	18.31%
Phase II	62.69%	65%	69.78%	55.96%	55%	99%	56.48%
Phase III	28%	66%	70%	58%	60%	84.81%	56.31%
Phase IV	5.88%	46.12	40.25%	19.35%	18.31%	73.06%	56.48%
Phase V	0%	52.85%	36.44%	35.92%	16.59%	76.86%	19.69%
Phase VI	0.50%	45%	30.71%	15.67%	17.23%	75.35%	17.12%

MLP showed significant improvement in Phase III, reducing its WER to 28%, which can be attributed to adjustments in hyperparameters or feature engineering. The slight improvements or stabilization in the Decision Tree and SVM models suggest incremental tuning but indicate a need for more extensive optimization.

By Phase IV, MLP's WER further decreased to 5.88%, showcasing the benefits of refined hyperparameter tuning and improved feature extraction techniques. The Decision Tree and SVM models also showed moderate reductions in WER, indicating that iterative tuning was starting to pay off. However, the ensemble method's slight increase in WER could be due to individual model variability.

In Phase V, MLP achieved a perfect WER of 0%, reflecting peak performance and effective training processes. Other models, including SVM and RFC, also showed significant improvements, illustrating the advantages of advanced tuning techniques. The KNN model's WER reduced to 16.59%, and the ensemble method's WER improved to 19.69%, indicating that individual model optimizations positively impacted the combined approach.

The final phase saw MLP maintaining a low WER of 0.50%, indicating consistent high performance. The ensemble method achieved its best WER of 17.12%, demonstrating the effectiveness of combining multiple optimized models. Decision Tree, SVM, and RFC models showed stable and improved WERs, highlighting the cumulative benefits of iterative tuning.

Overall, the analysis of WER across phases illustrates the progression from initial high error rates to significantly reduced rates through iterative tuning and optimization. The MLP model consistently outperformed other classifiers, achieving the lowest WER in the final phases, underscoring its suitability for accented Malayalam speech recognition.

15.3.2.2 Accuracy Analysis

MFCC features yielded high accuracies across various classifiers, with MLP achieving an impressive 99.5%. This demonstrates the effectiveness of MFCC in capturing essential frequency features from speech signals. Other classifiers, such as KNN and RFC, also performed well, achieving accuracies of 82% and 79%, respectively. However, the SGD classifier showed poor performance with an

accuracy of only 19%, indicating that it was less effective with MFCC features. Table 25 illustrates the performance evaluation in terms of Accuracy generated in the study. STFT provided lower accuracies compared to MFCC, with MLP achieving only 37%. This suggests that while STFT captures important spectral content, it may not be as effective for the classifiers used in this study. KNN and RFC showed moderate accuracies of 45%, indicating that STFT might not be the optimal standalone feature extraction method for these classifiers.

Tempogram features, focusing on rhythmic aspects, yielded moderate accuracies, with MLP achieving 72%. This indicates that while Tempogram features are informative, they are less effective than MFCC for most classifiers. Other classifiers showed accuracies ranging from 31% (SVM) to 43% (KNN), reflecting the limited improvement offered by rhythmic features alone.

Combining MFCC and STFT resulted in higher accuracies, with MLP achieving 95%. This combination utilizes both time and frequency aspects, enhancing the analysis and leading to significant performance improvements for classifiers like RFC and KNN, both achieving 81% accuracy.

The fusion of MFCC and Tempogram features produced high accuracies, with MLP again achieving 95%. This combination captures both frequency coefficients and rhythmic patterns, providing a detailed analysis that enhances the understanding of speech characteristics.

The integration of MFCC, STFT, and Tempogram features resulted in the highest accuracies, with MLP achieving 99%. This phase demonstrated that combining multiple feature extraction techniques can significantly improve classification accuracy, particularly for MLP, SVM, RFC, and KNN classifiers.

Table 25 Performance in Accuracy

Phases	Feature Extraction Method	Classifier	Accuracy (%)
Phase I	**MFCC**	MLP	99.5
		KNN	82
		RFC	79
		SVM	78
		Decision Tree	55
		SGD	19
		Ensemble	79
Phase II	**STFT**	MLP	37
		KNN	45
		RFC	45
		SVM	29
		Decision Tree	32
		SGD	10
		Ensemble	44
Phase III	**Tempogram**	MLP	72
		KNN	43
		RFC	41
		SVM	31
		Decision Tree	34
		SGD	21
		Ensemble	42
Phase IV	**MFCC + STFT**	MLP	95
		KNN	81
		RFC	81
		SVM	60
		Decision Tree	54
		SGD	28
		Ensemble	81
Phase V	**MFCC + Tempogram**	MLP	95
		KNN	81
		RFC	81
		SVM	60
		Decision Tree	54
		SGD	28
		Ensemble	81

Phases	Feature Extraction Method	Classifier	Accuracy (%)
Phase VI	**MFCC + STFT + Tempogram**	MLP	99
		KNN	82
		RFC	80
		SVM	60
		Decision Tree	52
		SGD	32
		Ensemble	81
Phase VII	**MFCC, STFT, Mel Spectrogram, RMS, Tempogram**	LSTM-RNN	Training: 95
			Validation: 67

Among the classifiers evaluated, MLP consistently provided the best results across different feature extraction techniques. MLP achieved the highest accuracy in most cases, such as 99.5% with MFCC and 99% with the combined features of MFCC, STFT, and Tempogram. These results indicate that MLP is particularly effective at capturing complex patterns in the speech data. SVM and RFC also showed good performance but were generally outperformed by MLP. Decision Tree and SGD consistently performed poorly compared to other classifiers, with maximum accuracies of 55% and 32%, respectively.

The optimal combination of features that provided the best results was the integration of MFCC, STFT, and Tempogram. This combination led to the highest accuracy of 99% with MLP, highlighting the value of integrating multiple types of features to enhance performance. The combination of MFCC with either STFT or Tempogram alone also yielded high accuracies, indicating the effectiveness of these combinations in capturing comprehensive information about the speech signals. The performance of the model leveraging a combination of MFCC, STFT, Mel Spectrogram, Root Mean Square (RMS), and Tempogram features demonstrated significant promise in recognizing accented Malayalam speech. Achieving a training accuracy of 95%, the LSTM-RNN model effectively learned the intricate patterns in the training data, indicating its capability to handle complex speech characteristics. The validation accuracy of 67%, while lower, suggesting that the model can effectively recognize unseen speech patterns despite some degree of overfitting. This

combined feature approach, integrating frequency, time-frequency, spectral, amplitude, and rhythmic information, provides a comprehensive representation of the speech signals, enhancing the model's overall performance.

In conclusion, this study demonstrates the effectiveness of combining various feature extraction techniques and classifiers to improve the recognition of accented Malayalam speech. The results highlight the superior performance of MLP and LSTM-RNN particularly when multiple feature extraction methods are integrated, providing valuable insights for future research and development in this field.

15.4 Experiment 4

Experiment 4 discussed in chapter 8 discusses that following feature vectorization, the study employs advanced deep learning architectures tailored for accent-based AASR. LSTM-RNN and DCNN are utilized to exploit temporal dependencies and spatial patterns in the speech data, respectively. The models are trained using stochastic gradient descent with backpropagation, optimizing performance metrics such as accuracy and loss. To evaluate model performance, the dataset is split into training, validation, and test sets. Performance metrics, including Word Error Rate (WER) and accuracy, are computed on the test set to assess the effectiveness of the proposed approach. Table 26 illustrates the performance evaluation of neural networks.

The LSTM-RNN demonstrated superior performance compared to the DCNN, achieving higher validation accuracy and lower validation loss. The DCNN model, despite performing well on the training data, showed signs of overfitting, indicated by the significant gap between training and testing performance metrics. These results emphasize the importance of choosing appropriate model architectures and feature representations for accent-based ASR systems. The LSTM-RNN's ability to effectively learn from temporal dynamics makes it a more suitable choice for the task of recognizing accented speech variations in Malayalam.

Table 26 Performance Evaluation

Metric	LSTM-RNN	DCNN
Training Duration	1000 epochs	4000 epochs
Training Samples	3020	3020
Testing Samples	800	800
Final Training Accuracy	95%	74%
Final Validation Accuracy	67%	39%
Initial Training Loss	0.03	High
Final Training Loss	0.003	9%
Initial Validation Loss	0.034	High
Final Validation Loss	0.027	17%
Overfitting Indication	No significant gap	Significant gap between training and testing performance

The findings reveal the importance of carefully selecting and combining features, as well as choosing the appropriate algorithms, to ensure an effective and accurate accented speech recognition model. This research adds to the growing body of knowledge in the field of accented speech recognition and contributes valuable insights that may inform future studies and technological advancements in the Malayalam language. The final experiment showcased the effectiveness of combining various features and utilizing advanced deep learning techniques, such as LSTM-RNN, in modeling accented Malayalam speech.

15.5 Experiment 5

The experimentation phase in chapter 9 involved evaluating several machine learning models for recognizing accented Malayalam speech. These models included MLP, DTC, SVM, RFC, KNN, SGD, and an ensemble classifier.

The AMSC-3 dataset, consisting of 7070 samples representing various age groups and genders, was used for this purpose. A comprehensive feature engineering

process was implemented to transform raw speech signals into meaningful representations. This included extracting multiple feature vectors such as MFCC, STFT, Mel Spectrogram, Spectral Roll-Off, RMS, and Tempogram rhythmic features. Each audio signal was vectorized into a set of 1530 features.

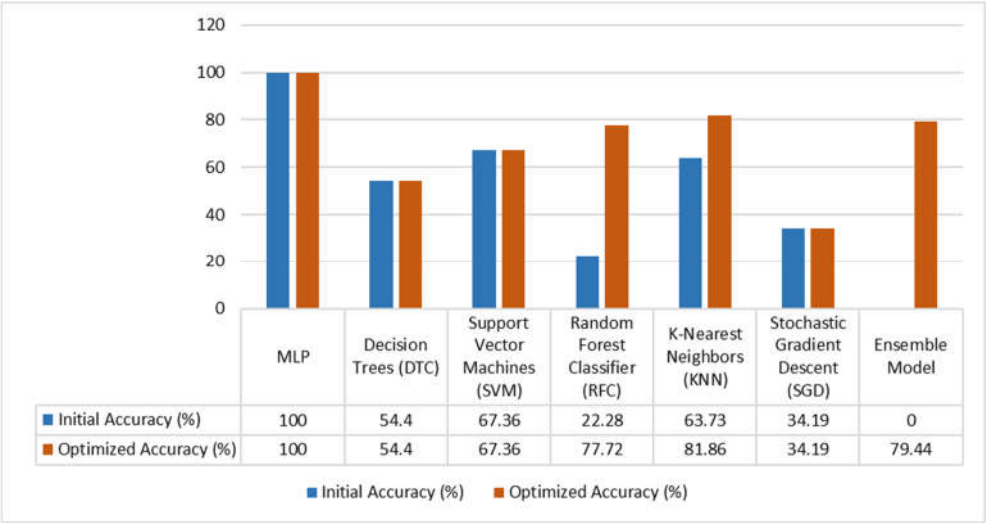


Figure 105 Performance Evaluation in Accuracy

Figure 105 illustrates the performance of the experiment. The MLP achieved remarkable success with a perfect accuracy of 100%, showcasing its adeptness in learning intricate patterns. Conversely, the DTC managed an accuracy of 54.40%, highlighting its interpretability but also its struggle with complex data patterns. Utilizing high-dimensional feature vectors, the SVM attained an accuracy of 67.36%, demonstrating its effectiveness. Initially underperforming, the RFC saw a significant boost to 77.72% accuracy through hyperparameter tuning, emphasizing the advantages of ensemble methods. The KNN algorithm displayed sensitivity to parameters, improving from 63.73% to 81.86% accuracy after tuning. In contrast, the SGD achieved a modest 34.19% accuracy, indicating challenges in capturing nuanced accents. Combining the strengths of individual models, the Ensemble Model achieved a commendable accuracy of 79.44%, underscoring the effectiveness of hybrid approaches.

Table 27 describes the study focused on evaluating the LSTM-RNN architecture's performance in recognizing accented Malayalam speech through three experiments

with varying training durations. In the initial experiment, training the LSTM-RNN for 2000 epochs with a batch size of nine allowed the model to effectively capture both spectral and temporal speech characteristics, demonstrating significant learning of complex speech patterns. The second experiment extended the training duration beyond 2000 epochs, resulting in improved accuracy and reduced WER, indicating enhanced understanding of accented speech nuances through additional iterations. The third experiment, with further extended training, revealed a decline in accuracy beyond a certain point, highlighting the risk of overfitting and the importance of finding the optimal training duration to balance between effective learning and generalization.

Table 27 Performance of LSTM - RNN

No. of epochs	Train Accuracy	Validation Accuracy	Train Loss	Validation Loss	No. of Steps	WER for known words	WER for unknown words
2000	93.30%	60.72%	0.29%	1.94%	482000	7%	39%
3000	96.97%	63.35%	0.11%	1.98%	723000	3%	37%
4000	95.74%	62.22%	0.15%	1.95%	1000000	4%	38%

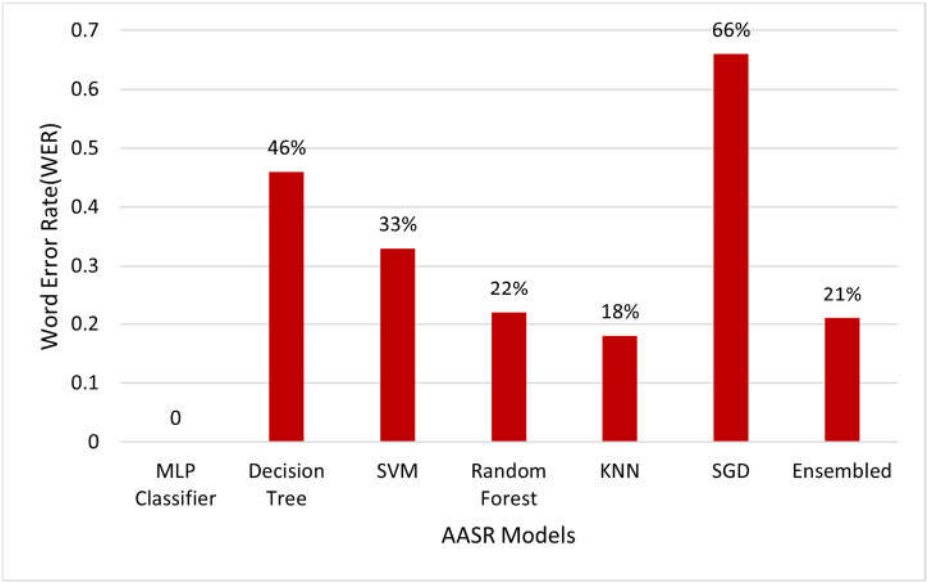


Figure 106 Performance Evaluation in terms of WER

Figure 106 illustrates the performance of the model in terms of WER. The results of these experiments highlight the crucial role of feature engineering and model selection in developing an effective AASR system for Malayalam. While individual models varied in their performance, the use of hyperparameter tuning and ensemble techniques significantly improved the system's ability to recognize accented speech accurately.

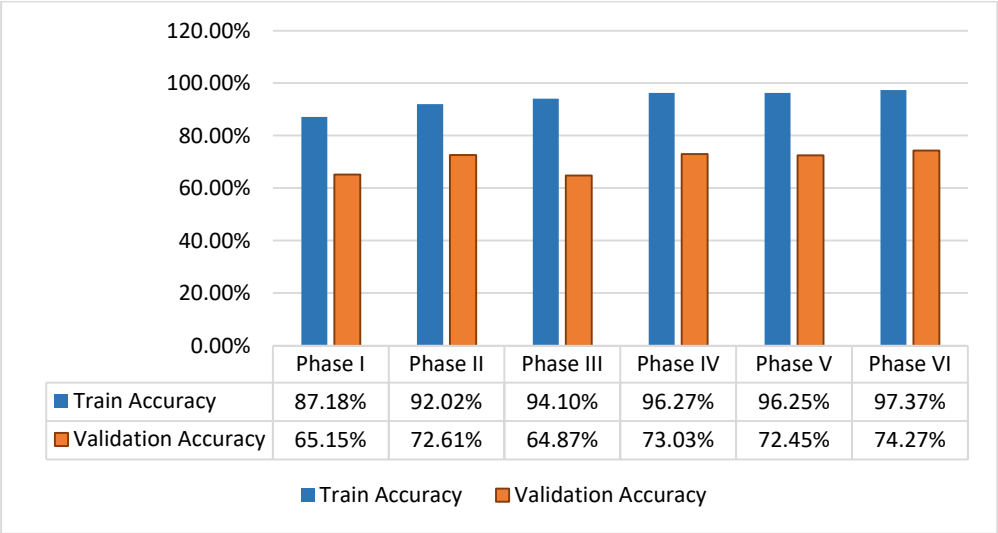
The experiments emphasized the potential of deep learning methods, particularly LSTM-RNN, for accented speech recognition. They emphasized the necessity of carefully optimizing training duration to achieve the best balance between model accuracy and generalization capability while avoiding overfitting. This study illustrates that the MLP Classifier achieved a perfect score, registering no errors, while other models displayed varying degrees of effectiveness.

15.6 Experiment 6

This experiment discussed in chapter 10 discusses the training process involved six distinct phases, each implementing different variations of RNN architectures and enhancements. In Phase 1, a basic RNN architecture was utilized to train the model. Despite its simplicity, this phase provided a foundation for subsequent enhancements. Phase 2 introduced attention mechanisms into the RNN architecture. This allowed the model to focus on relevant parts of the input sequence, improving its ability to capture dependencies and relationships within the data. Transitioning to Phase 3, the model architecture was upgraded to LSTM cells. LSTMs can capture long-range dependencies and mitigating the vanishing gradient problem, thus enhancing the model's performance. Building upon the LSTM architecture, Phase 4 incorporated attention mechanisms. By combining LSTM cells with attention, the model gained the ability to selectively attend to important features, further improving its predictive capabilities.

In Phase 5, the model architecture was modified to a Bidirectional LSTM (BiLSTM). BiLSTMs process input sequences in both forward and backward directions,

allowing the model to capture contextual information from both past and future states. Finally, Phase 6 introduced attention mechanisms into the BiLSTM architecture. This combination utilized the advantages of bidirectionality and attention, enabling the model to capture complex patterns and dependencies within the data while focusing on relevant information.



	Phase I	Phase II	Phase III	Phase IV	Phase V	Phase VI
Train Accuracy	87.18%	92.02%	94.10%	96.27%	96.25%	97.37%
Validation Accuracy	65.15%	72.61%	64.87%	73.03%	72.45%	74.27%

Figure 107 Accuracy Vs Phases of Experiment

Throughout these phases, the model underwent iterative training and refinement, resulting in progressively higher accuracy and improved performance. Each phase represented a step towards developing a sophisticated and effective model for the task at hand, showcasing the iterative nature of deep learning model development. Figure 107 illustrates a progressive enhancement in model performance, reflected through accuracy metrics across different phases.

It provides a detailed overview of the experimental results across six distinct phases, capturing the training and validation accuracies, loss values, and the number of epochs. WER includes substitution errors (S), insertion errors (I), and deletion errors (D). It is calculated as $WER=\frac{S+D+I}{N}$, where N is the total number of utterances.

MER is computed as $\text{MER} = \frac{N-S-D-I.}{N}$

Figure 108 illustrates the model performance at varying epochs of different phases.

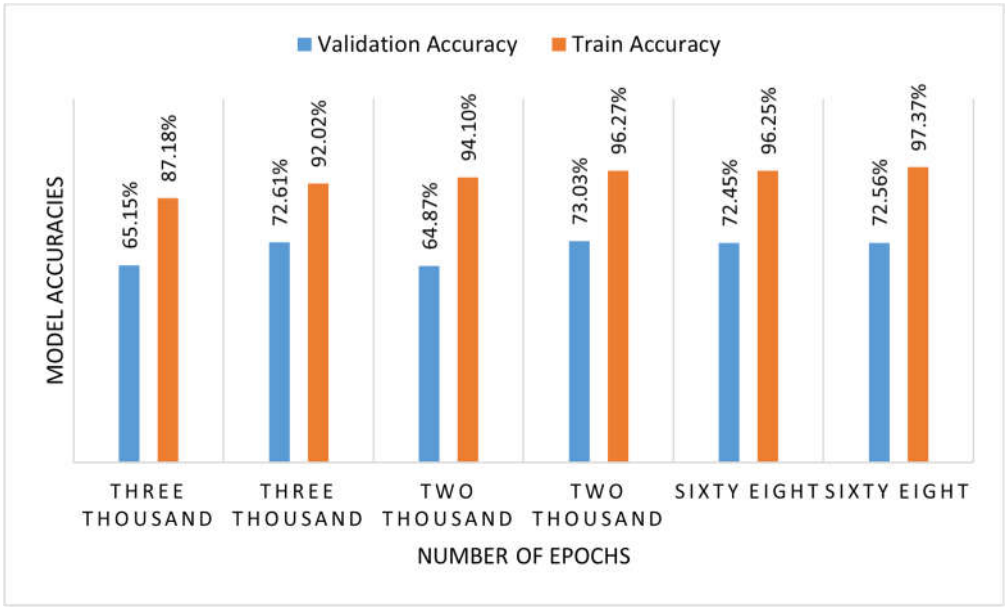


Figure 108 Model Accuracies Vs Epochs

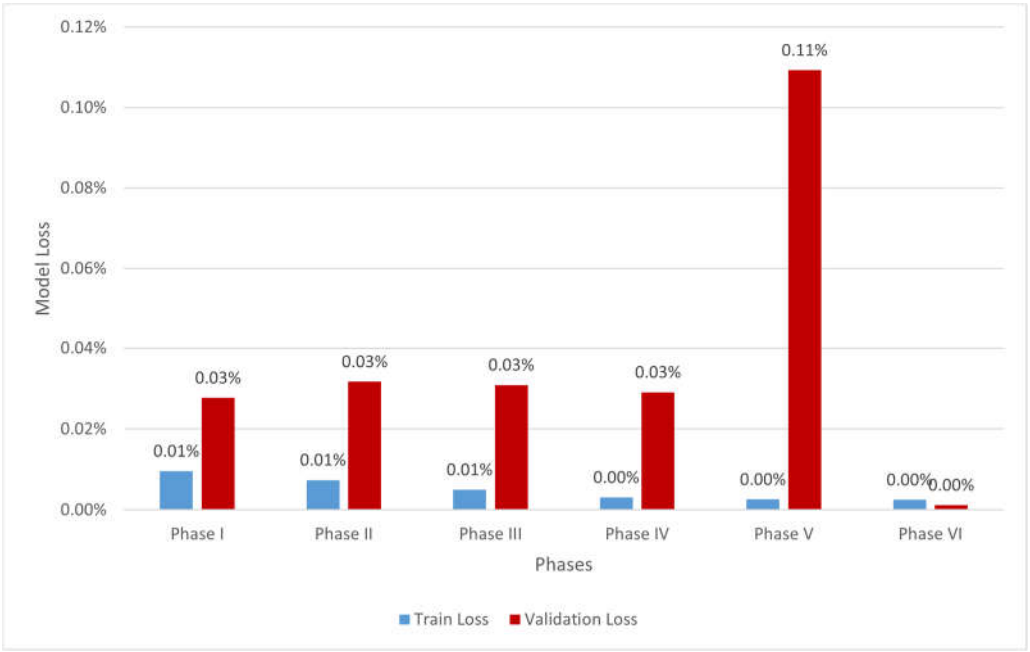


Figure 109 Model Loss Vs Phases

The results indicate substantial progress throughout the experimental phases. The utilization of RNN with attention in Phase II significantly outperformed Phase I. The highest performance was achieved in Phase VI with BiLSTM and attention

mechanisms, demonstrating greater accuracy and lower error rates. In addition to accuracy and loss, the evaluation also considered WER and MER.

Figure 109 illustrates a progressive enhancement in model performance, reflected through loss metrics across different phases.

15.7 Experiment 7

The experiment discussed in chapter 11 illustrates the fusion of autoencoders with machine learning (ML) models demonstrates a powerful approach for enhancing classification accuracy and robustness in speech recognition tasks, particularly in dealing with the complexities of accented speech. Through the integration of autoencoder-generated features with various ML classifiers, significant improvements in predictive performance have been observed across multiple models.

The Figure 110, Figure 111, Figure 112 clearly illustrates the performance evaluation in terms of accuracy, WER, and log loss across various experiments conducted also it illustrates a broader perspective of the experimental evaluations, showcasing the results in terms of accuracy, loss, precision, and recall across different test scenarios.

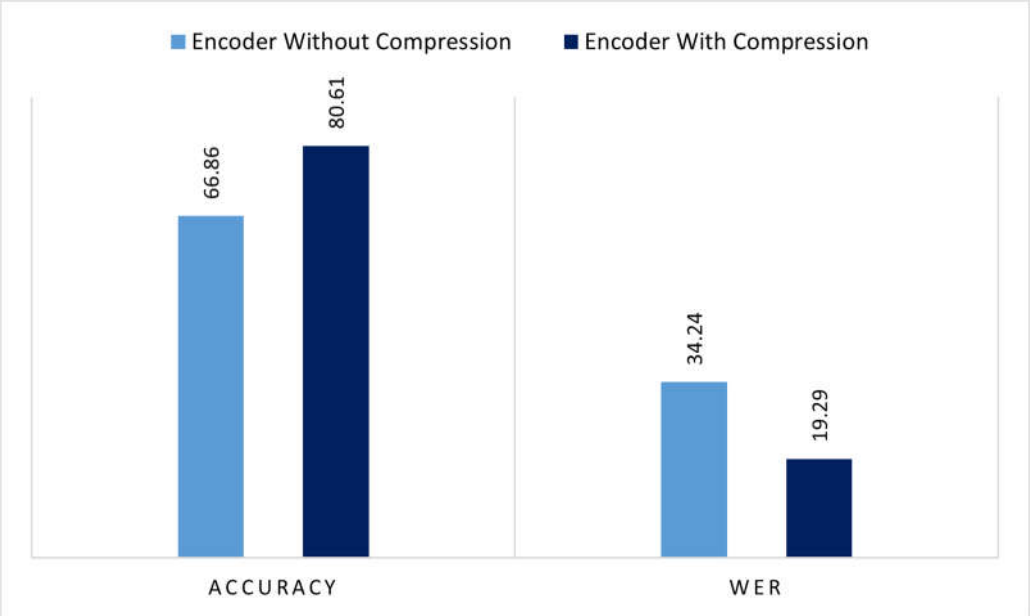


Figure 110 Performance of Encoder Models

Figure 110 illustrates that the AASR model constructed using encoder with compression showed better performance in terms of accuracy and WER.

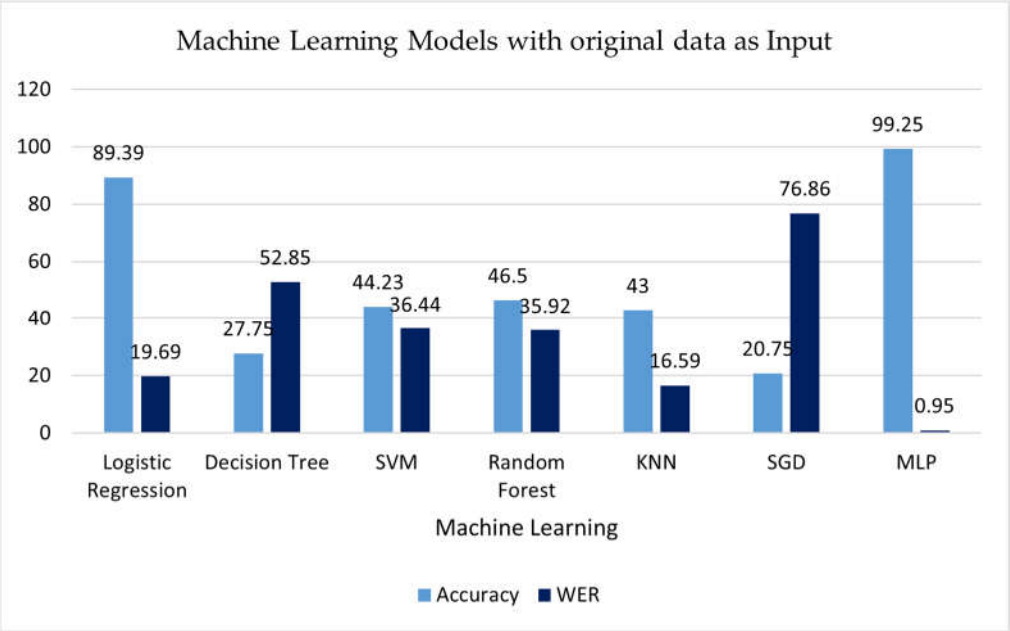


Figure 111 Performance of ML Models

Figure 111 clearly shows the AASR model performances that are constructed using various methodologies. AASR constructed with MLP yields the best results when trained with original input data.

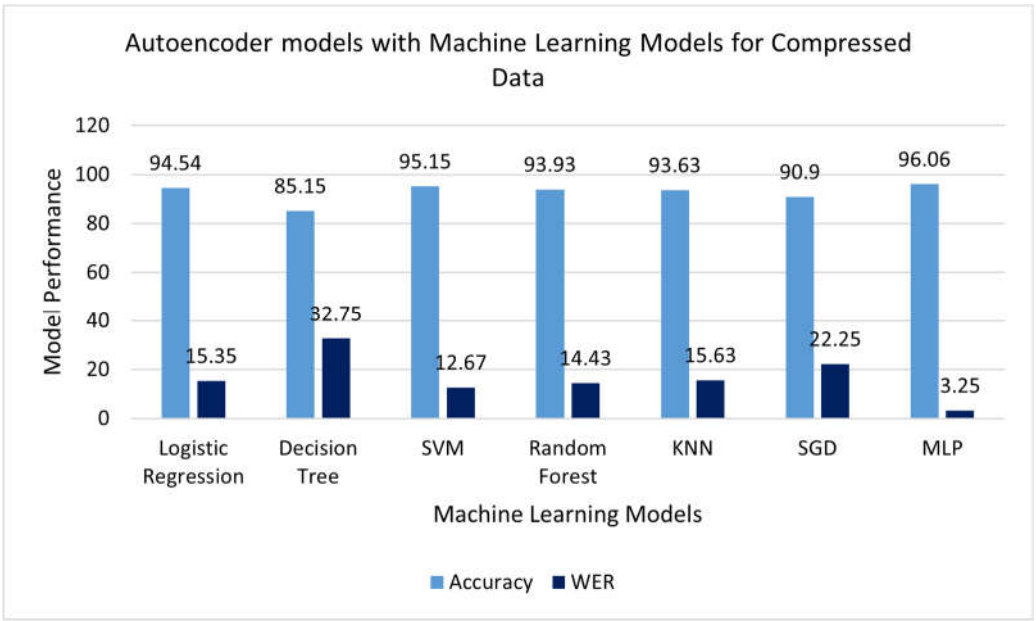


Figure 112 Performance of Hybrid Autoencoder Models

Figure 112 describes the performance evaluation of the AASR models constructed with the already constructed AASR models using various machine learning techniques. A hybrid model was developed with compressed data and ML technology. Figure 113 illustrates the AASR model performance in terms of the log loss generated while constructing the models. The AASR model constructed with autoencoder with compressed data method performed well that the other model that was constructed with original data (without compression).

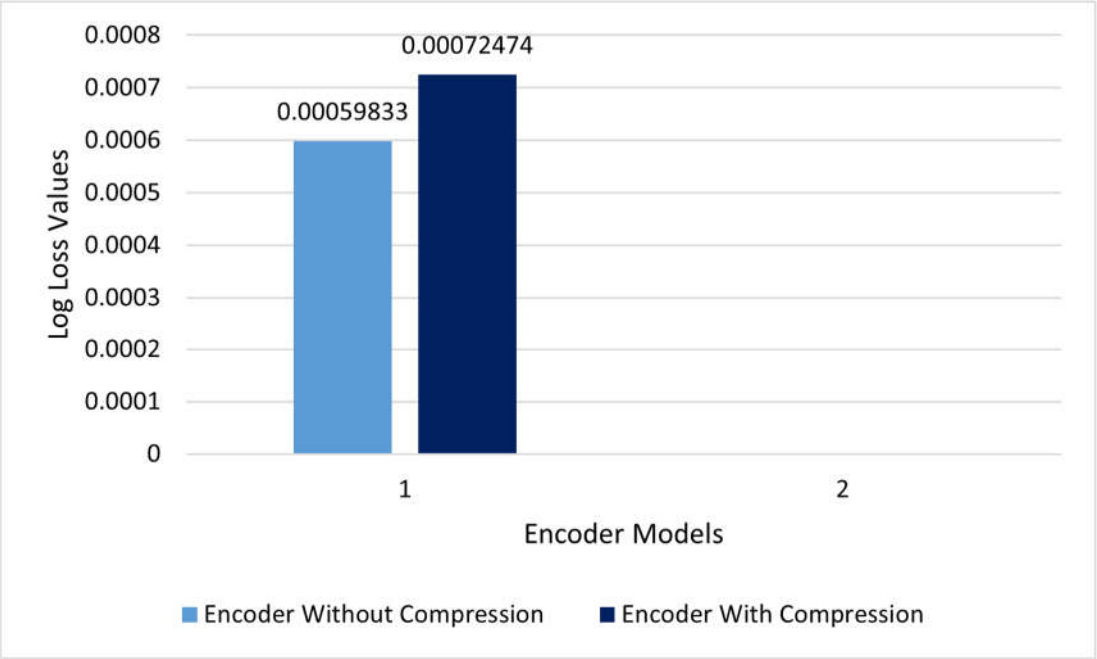


Figure 113 Performance in Terms of Log Loss Values

Figure 114 illustrates the behavior of the AASR model that was constructed at various phases of the research. The evaluation performed in terms of log loss and the hybrid model constructed using MLP and compressed generated the lowest log loss value indicating the efficiency of the architecture in modeling the accent complexities for Malayalam.

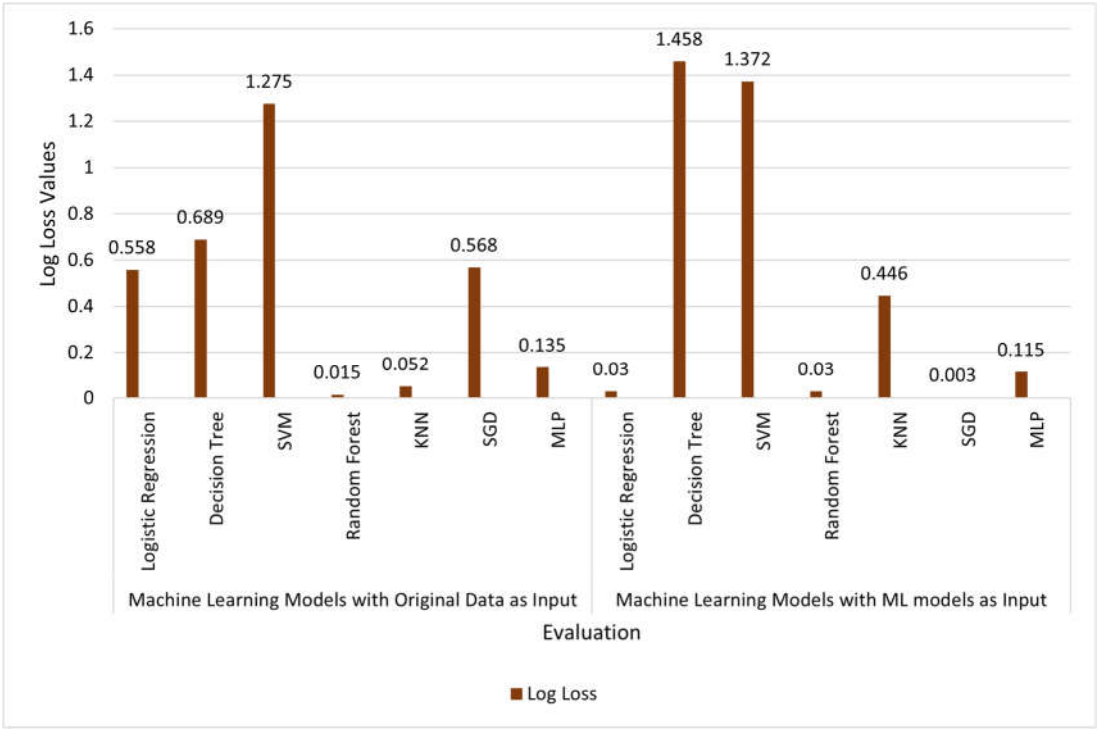


Figure 114 Performance of ML Models VS Hybrid Models

15.8 Experiment 8

Experiment 8 discussed in chapter 12 a comprehensive set of 584 emotional speech features was carefully extracted. Prior to extraction, audio signals underwent standardization to ensure uniform dimensions, thereby facilitating consistent feature extraction across all samples. Spectral contrast, polyfeatures, tempogram, tonnetz, Parselmouth-derived periodicity measure, formant frequencies, fundamental frequency (F0) standard deviation, MFCCs, delta, delta2, zero crossing rate (ZCR), chroma STFT, root mean square value (RMS), and Mel spectrogram were among the diverse range of features computed, each contributing unique insights into the emotional content of speech signals. This multi-faceted approach to feature extraction highlights the depth and breadth of methods utilized in the study.

Following feature extraction, various clustering algorithms were employed to identify underlying patterns and structures within the data. OPTICS, known for its adaptability to varying data densities, demonstrated strong performance with a silhouette score of 0.55, indicating clear and well-defined clusters. BIRCH, although

hierarchical in nature, exhibited moderate cluster separation and cohesion (silhouette score of 0.17), suggesting some difficulty in fully delineating clusters without overlap. Ensemble clustering using majority voting achieved moderate performance (silhouette score of 0.40), leveraging the strengths of multiple methods but still showing some overlap between clusters. Table 28 illustrates the performance of the clusters in the terms of Silhouette Scores.

Table 28 Performance Evaluation of the Clustering Techniques

Clustering Algorithm	Silhouette Score
OPTICS	0.55
BIRCH	0.17
Ensemble Clustering (Majority Voting)	0.4
Consensus Clustering	0.16
Affinity Propagation	0.47
Mean Shift Clustering	0.34
Agglomerative Clustering	0.15
Gaussian Mixture Model (GMM)	0.2
DBSCAN	0.12
Spectral Clustering	-0.18

Consensus clustering, building upon the ensemble approach, encountered challenges with a relatively low silhouette score of 0.16, indicating potential difficulties in achieving distinct cluster separation. Affinity Propagation, leveraging pairwise similarities, produced clusters with moderate separation and cohesion (silhouette score of 0.47), showcasing its adaptability and effectiveness in capturing underlying patterns. Mean Shift Clustering, while identifying dense regions effectively, exhibited some cluster overlap despite a silhouette score of 0.34.

Agglomerative Clustering, despite its intuitive hierarchical approach, yielded a lower silhouette score of 0.15, suggesting limitations in defining clear cluster boundaries. Gaussian Mixture Model (GMM) clustering and DBSCAN both faced

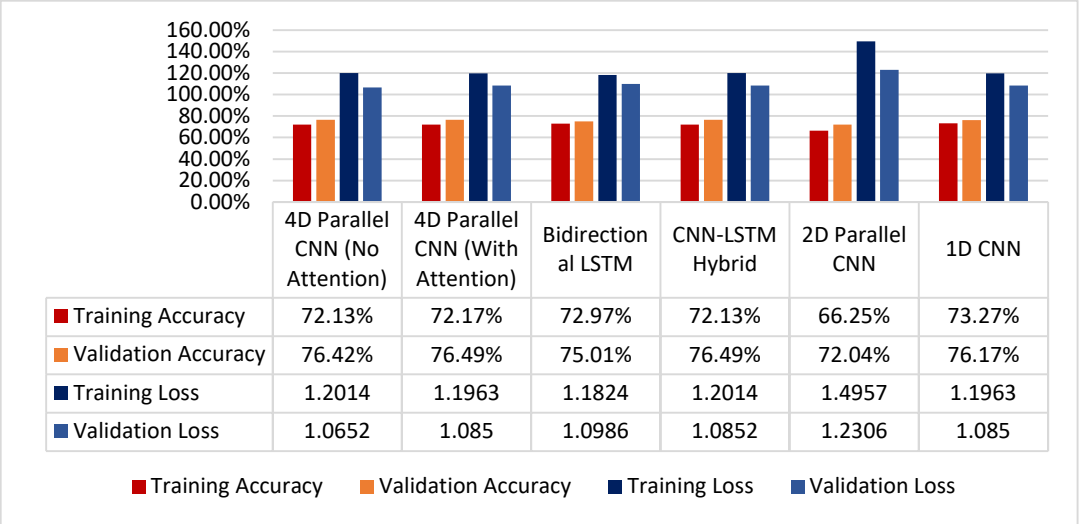
challenges in fully separating clusters, exhibiting some overlap despite silhouette scores of 0.20 and 0.12, respectively. Spectral Clustering, however, yielded a negative silhouette score (-0.18), indicating failure to generate meaningful clusters with significant overlap.

In summary, the study's extensive feature extraction techniques provided rich insights into the emotional content of speech signals. While certain clustering algorithms demonstrated strong performance in identifying clear cluster structures, others encountered challenges, highlighting the complexity of the data and the nuances involved in clustering analysis. Further exploration and refinement of clustering methodologies may be warranted to fully uncover underlying patterns in the data and enhance clustering performance.

15.9 Experiment 9

The experiment discussed in chapter 13 discusses the experiments encompassing multiple methodologies ranging from CNNs, LSTMs, and a combination of both, often augmented with attention mechanisms. Below is a comprehensive summary of the results and evaluation. Figure 115 illustrates the performance of the model at different phases of the experiment.

The experiment conducted a comprehensive evaluation of six distinct approaches for Accented Speech Recognition (AASR), each utilizing unique machine learning model architectures. These approaches included the 4D Parallel CNN with and without attention mechanisms, Bidirectional LSTM, CNN-LSTM Hybrid, 2D Parallel CNN, and 1D CNN. Each approach was trained and evaluated over a specified number of epochs, with detailed documentation of the time per epoch and total training time. Performance metrics such as training accuracy, validation accuracy, training loss, and validation loss were carefully recorded for analysis.



	4D Parallel CNN (No Attention)	4D Parallel CNN (With Attention)	Bidirectional LSTM	CNN-LSTM Hybrid	2D Parallel CNN	1D CNN
Training Accuracy	72.13%	72.17%	72.97%	72.13%	66.25%	73.27%
Validation Accuracy	76.42%	76.49%	75.01%	76.49%	72.04%	76.17%
Training Loss	1.2014	1.1963	1.1824	1.2014	1.4957	1.1963
Validation Loss	1.0652	1.085	1.0986	1.0852	1.2306	1.085

Figure 115 Performance Evaluation of the Experiment

The results indicate competitive performance across the evaluated approaches. The Bidirectional LSTM approach achieved the highest training accuracy of 72.97%, closely followed by the 1D CNN approach with a training accuracy of 73.27%. The CNN-LSTM Hybrid and 4D Parallel CNN (with attention) approaches also demonstrated promising results, with training accuracies of 72.13% and 72.17%, respectively.

In terms of validation accuracy, the 1D CNN approach exhibited robust performance, achieving a validation accuracy of 76.17%. This was comparable to the validation accuracies of the CNN-LSTM Hybrid and 4D Parallel CNN (with attention) approaches, which both achieved a validation accuracy of 76.49%. The 1D CNN approach demonstrated competitive training and validation loss metrics, further highlighting its effectiveness for AASR tasks. Despite not having detailed information on the time per epoch, the 1D CNN approach showcased efficient training, with a total training time of approximately 5 minutes.

Overall, the experiment provided valuable insights into the efficacy and efficiency of various machine learning model architectures for Accented Speech Recognition, with the 1D CNN approach emerging as a competitive contender among the evaluated methodologies.

15.10 Experiment 10

The experiment discussed in chapter 14 is conducted on accented Malayalam speech dataset for hate speech detection. For the hate class, a precision of 0.98 indicates that 98% of instances predicted as hate were indeed hate. Similarly, for the non-hate class, 99% of instances predicted as non-hate were correct. The hate class has a recall of 1.00 (or 100%), meaning that all actual hate instances were correctly identified by the model. For non-hate, a recall of 0.95 indicates that 95% of actual non-hate instances were correctly identified, while 5% were missed.

The F1-Score is the harmonic mean of precision and recall, providing a balance between the two metrics. It's particularly useful when class distributions are imbalanced. Both classes have high F1-Scores, with hate at 0.99 and non-hate at 0.97, indicating a balanced and high performance between precision and recall for both classes. This refers to the actual number of occurrences of each class in the dataset. It gives context to the other metrics, to identify how many instances were evaluated to arrive at the given precision, recall, and F1-score. In this case, there were 1430 hate instances and 607 non-hate instances in the dataset. The hate speech model evaluation is depicted in Figure 116.

Accuracy is the ratio of correctly predicted instances to the total number of instances. An accuracy of 98% means that 98% of all predictions made by the model were correct. Macro Avg (Average) is the average performance metric (precision, recall, F1-score) across all classes without considering the class distribution. It treats all classes equally.

Weighted Avg (Average) provides the average performance metric weighted by the number of true instances for each label. It accounts for any class imbalance and can often provide a more comprehensive overview than the macro average when classes are unequally distributed.

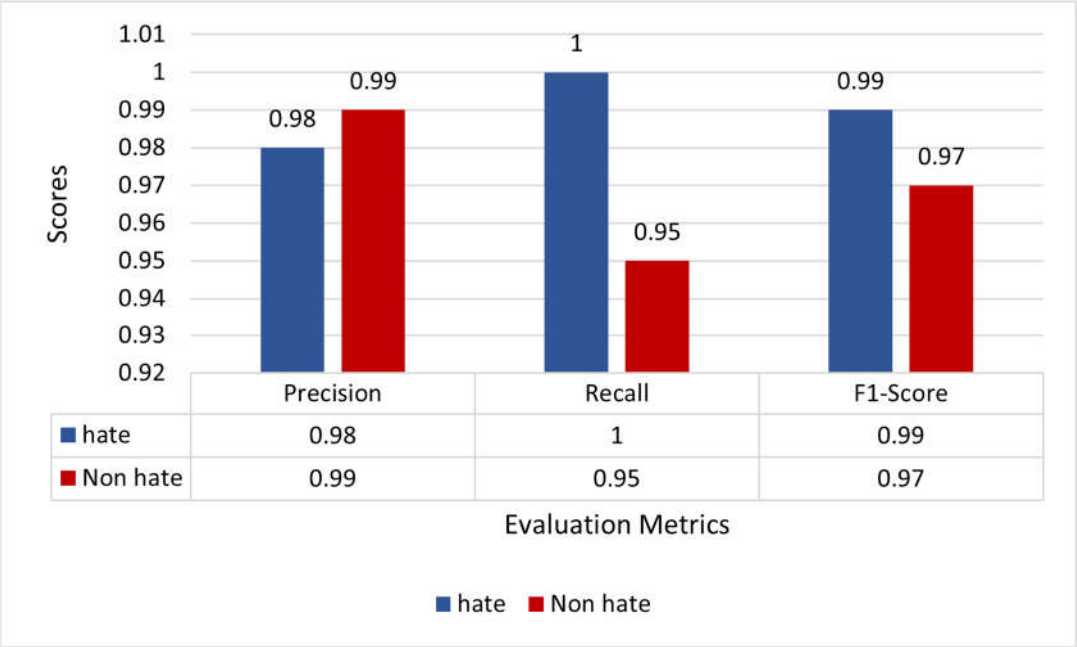


Figure 116 Hate Speech Evaluation

In this research, as detailed in this study, innovative approaches for enhancing accented speech recognition in the Malayalam language were introduced. The focus was on exploring unique strategies that go beyond conventional methods. By examining the innovative feature extraction techniques, it was aimed to elevate the accuracy and effectiveness of accented speech recognition systems. The study highlights the critical role of comprehensive feature extraction and advanced model architectures in improving the accuracy of accented Malayalam speech recognition.

16. Conclusion

This chapter aims to encapsulate the performance advantages resulting from the rigorous work undertaken in this thesis on accented speech recognition for the Malayalam language. It is crucial to provide a summary of existing work, findings, research contributions, practical implications, limitations, and future directions. The performance advantages achieved in quantitative measures are highlighted, particularly emphasizing the significant phases of this work and the achievements in AASR for Malayalam. The central focus of this thesis has been to explore the domain of AASR to identify the most efficient approaches for accented Malayalam speech.

16.1 Summary of Findings

The central focus of this thesis has been to explore and identify the most efficient approaches for accented Malayalam speech recognition. The key findings are:

The central focus of this thesis has been to explore and identify the most efficient approaches for accented Malayalam speech recognition. The key findings are:

1. **Advancements in Feature Extraction Techniques**: This research refined existing methodologies for feature extraction, enhancing the quality of speech recognition by fine-tuning processes tailored to accented Malayalam speech.

2. **LSTM RNN and Crowdsourced Data**: The application of LSTM RNN with crowdsourced data showed significant promise, effectively capturing the variations and details of accented speech.

3. **CNN, Auto Encoders, and Speech Feature Sets**: The use of CNN and auto encoders demonstrated their proficiency in handling various speech feature sets, proving effective in recognizing accented Malayalam speech.

4. **MLP and Near-Zero Word Error Rate (WER)**: Multi-Layer Perceptron (MLP) achieved near-zero WER in specific datasets, highlighting its high accuracy and reliability in speech recognition tasks.

5. **Hybrid Approach with CNN and LSTM RNN**: The integration of CNN and LSTM RNN architectures emerged as the most efficient solution, combining accuracy with speed advantages.

6. **Emotion Clustering from Accented Emotional Data**: Innovative emotion clustering within accented emotional data provided new insights into the interplay between emotion and linguistic variations.

7. **Hate Speech Detection Using Accented Data**: The study demonstrated the effectiveness of accented data in detecting hate speech, showcasing its practical applications in diverse linguistic contexts.

8. **Generated Datasets for Accented Malayalam Speech Recognition**: Nine distinct datasets were created, enhancing the diversity and comprehensiveness of the research.

16.2 Research Contributions

The research introduced novel approaches in feature extraction, LSTM RNN, CNN, auto encoders, machine learning techniques, and hybrid models. Significant contributions include:

1. **AASR of Malayalam Isolated Words Using LSTM-RNN**: Demonstrated the efficiency of LSTM-RNN in recognizing isolated words in Malayalam.

2. **AASR with Deep-CNN, LSTM-RNN, and Machine Learning Approaches**: Showed the effectiveness of these models in speech recognition.

3. **End-to-End Unified AASR for Low-Resource Contexts**: Developed robust models for low-resourced contexts.

4. **Analyzing Multisyllabic Words in Low-Resourced Contexts**: Provided insights into handling complex speech patterns.
5. **Deep Neural Networks and Attention Mechanisms**: Enhanced the accuracy and robustness of AASR models.
6. **Integration of Self-Supervised Learning and Autoencoders**: Improved feature extraction and model performance.
7. **Clustering Methods for Emotion Classification**: Contributed to understanding emotional complexities in speech.
8. **Exploration of Diverse Architectures**: Investigated various CNN and hybrid models.
9. **Dual Approach to Detect Hate Speech in Accented Malayalam**: Showcased practical applications in hate speech detection.

16.3 Practical Implications

The successful application of accented data for hate speech detection and emotion clustering highlights significant real-world applications:

1. **Enhancing NLP Systems**: Incorporating accented Malayalam speech data enhances Natural Language Processing (NLP) systems' ability to be more emotionally and contextually aware. This improves user interactions with virtual assistants, customer service bots, and interactive voice response systems, making them more personalized and empathetic.
2. **Hate Speech Detection**: Demonstrating the effectiveness of using accented data for hate speech detection contributes to safer online platforms. Recognizing hate speech across different Malayalam accents ensures effective content moderation, fostering a healthier digital communication environment.
3. **Accessibility and Inclusivity**: Speech recognition systems that cater to various Malayalam accents make technology more accessible to non-native speakers and

individuals with regional accents. This inclusivity is crucial in multilingual and diverse linguistic communities in Kerala.

4. **Educational Tools**: Tailored speech recognition technologies enhance language learning tools for Malayalam speakers with different accents. These tools can aid in pronunciation training and language proficiency assessments, providing more effective learning experiences.
5. **Healthcare Applications**: Emotion detection from accented speech can be utilized in mental health monitoring tools to assess emotional well-being. This assists healthcare professionals in diagnosing and monitoring conditions such as depression and anxiety based on verbal communication.
6. **General Applicability to ASR Systems**: The methodologies and techniques developed in this study can be applied to improve all ASR systems, not just those focused on Malayalam. By addressing the challenges posed by accented speech, these advancements can enhance the robustness and accuracy of ASR systems across various languages and dialects, leading to broader applications in global contexts.

16.4 Challenges and Future Directions

Despite the remarkable outcomes, several challenges need to be addressed to further improve accented speech recognition systems:

1. **Complexity of Emotion Clustering**: Accurately clustering emotions from accented Malayalam speech is complex due to the subtle and diverse ways emotions are expressed. Future research should focus on refining these techniques to improve accuracy and reliability.

2. **Continuous Model Refinement**: Speech recognition models must be continuously updated to adapt to new data and evolving speech patterns. This includes addressing variations in Malayalam accents that may arise due to sociolinguistic changes.

3. **Scalability**: Ensuring models can scale efficiently to handle large datasets and real-time processing is crucial for practical deployment. Future work should explore optimization techniques to enhance scalability without compromising accuracy.

4. **Cross-Linguistic Generalization**: Extending the developed techniques for Malayalam to other languages and accents poses a significant challenge. Future research should investigate the generalizability of these methods to diverse linguistic contexts.

16.5 Ethical Considerations

As technology advances, ethical considerations become paramount in the development and deployment of speech recognition systems:

1. **Fairness and Bias Mitigation**: Ensuring fairness and mitigating biases in accented speech data is critical. This involves addressing potential biases in training data that may lead to unequal performance across different Malayalam accents and demographic groups. Diversifying training datasets and implementing bias detection and mitigation strategies promote fairness.

2. **Privacy and Consent**: Collecting and using speech data, especially from crowdsourced sources, raises privacy concerns. It is essential to obtain informed consent from participants and ensure that data is anonymized and securely stored to protect user privacy.

3. **Transparency and Accountability**: Developers should strive for transparency in how speech recognition systems are designed and used. Clear documentation and open communication about the limitations and intended uses of these systems help build trust with users.

4. **Ethical Use Cases**: The applications of speech recognition technology should be guided by ethical principles, ensuring they benefit society and do not harm or discriminate against individuals or groups.

In this research, the potential application of various methodologies and techniques in accented speech recognition for Malayalam, a language rich in dialectal diversity, was explored. The exploration was rooted in empirical experiments and in-depth analysis, leading to encouraging results that emphasize the efficacy of different experiments. Specifically, these neural network architectures were found to excel in isolating significant features from speech data and in synthesizing meaningful representations. The core approach involved utilizing deep learning and unsupervised learning techniques, enhancing both the quality of the extracted features and the discriminative prowess of the developed models. The resulting performance evaluation affirmed the proposed approach's superior accuracy and robustness in recognizing various Malayalam accents. This research has made substantial contributions to the field of accented speech recognition, particularly for the Malayalam language, and has set the stage for further innovations and practical applications in this area. The methodologies and findings also provide a strong foundation for enhancing ASR systems in general, demonstrating the broader impact of this work on global speech recognition technologies.

17. Recommendations

The recommendations include the formulation of enhanced methodologies capable of constructing unified accented models that recognize all Malayalam accents. The insights and techniques developed through this work can be transposed and adapted to other low-resourced languages, potentially paving the way for broader advancements in the field of accented speech recognition. This adaptability could contribute to reinforcing the research's contributions and emphasizing its relevance not only for Malayalam but for linguistics and technology at large. The recommendations of this work are:

17.1 Refinement of Emotion Clustering Algorithms

The exploration into emotion clustering from accented emotional data has opened new dimensions in understanding the interplay between linguistic variations and emotional expressions. Future research could focus on refining existing emotion clustering algorithms to handle the intricacies of diverse emotional variations present in accented speech more effectively. This includes exploring advanced machine learning and deep learning techniques to enhance the accuracy and sensitivity of emotion recognition models.

17.2 Dynamic Adaptation for Linguistic Variations

As linguistic variations within accented Malayalam speech can be dynamic, future works should explore adaptive models that can dynamically adjust to different accents and linguistic styles. This may involve the development of models that can continuously learn and adapt over time, ensuring robust performance across a broad spectrum of linguistic variations.

17.3 Large-Scale Deployment and User Interaction Studies

While this study provides promising outcomes, future research could focus on large-scale deployment scenarios to evaluate the scalability and real-world applicability of

the proposed models. Moreover, conducting user interaction studies to gather feedback on the user experience with accented speech recognition systems will be crucial for refining models and addressing user-specific challenges.

17.4 Multimodal Approaches for Enhanced Recognition

Integrating multiple modalities, such as incorporating visual cues or facial expressions along with accented speech, could be explored to enhance recognition accuracy. Multimodal approaches have the potential to provide additional context, especially in scenarios where accents are accompanied by non-verbal cues or contextual information.

17.5 Exploration of Adversarial Training for Robustness

Given the potential challenges associated with adversarial attacks on speech recognition systems, future works could investigate the application of adversarial training techniques. This involves incorporating adversarial examples during model training to enhance the robustness of the system against potential attacks, ensuring reliable performance in real-world scenarios.

17.6 Cross-Linguistic Studies on Accented Speech

Extending the scope of research to encompass cross-linguistic studies on accented speech can provide valuable insights into the generalizability of models. Investigating the transferability of accented speech recognition models across different languages and language families will contribute to the broader understanding of accented speech processing.

17.7 In-depth Analysis of Hate Speech Detection

While this study has shown promising results in hate speech detection using accented data, future works could delve deeper into the specific challenges associated with detecting hate speech in diverse accented Malayalam contexts. This

includes exploring the details of cultural and linguistic variations that may impact the effectiveness of hate speech detection models.

17.8 Integration of Explainable AI in AASR

Enhancing the interpretability of accented speech recognition models is crucial for fostering trust and understanding in end-users. Future research could explore the integration of explainable AI techniques to provide insights into how the models make decisions, making the technology more transparent and accountable.

References

[1] Aksënova, A., Chen, Z., Chiu, C., Daan, V. E., Golik, P., Han, W., King, L., Ramabhadran, B., Rosenberg, A., Schwartz, S., & Wang, G. (2022). Accented Speech Recognition: benchmarking, pre-training, and diverse data. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2205.08014.

[2] Nilaksh Das and Sravan Bodapati and Monica Sunkara and Sundararajan Srinivasan and Duen Horng Chau, (2021) Best of both worlds: Robust accented speech recognition with adversarial transfer learning, https://www.amazon.science/publications/best-of-both-worlds-robust-accented-speech-recognition-with-adversarial-transfer_learning, Interspeech 2021.

[3] Muhammad Ahmed Hassan, Asim Rehmat, Muhammad Usman Ghani Khan, Muhammad Haroon Yousaf, (2022) Improvement in Automatic Speech Recognition of South Asian Accent Using Transfer Learning of DeepSpeech2, Mathematical Problems in Engineering, vol. 2022, Article ID 6825555, 12 pages. https://doi.org/10.1155/2022/6825555.

[4] J. Ni, L. Wang, H. Gao et al., Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition, (2022), https://arxiv.org/abs/2203.15796.

[5] A. Jain, V. P. Singh and S. P. Rath (2019), A multi-accent acoustic model using mixture of experts for speech recognition, Proc. Interspeech, pp. 779-783.

[6] Yanmin Qian and Xun Gong and Houjun Huang, (2022) IEEE/ACM Transactions on Audio, Layer-Wise Fast Adaptation for End-to-End Multi-Accent Speech Recognition Speech, and Language Processing, volume 30,2842-2853.

[7] Ryo Imaizumi, Ryo Masumura, Sayaka Shiota and Hitoshi Kiya, (2020) End-to-end Japanese Multi-dialect Speech Recognition and Dialect Identification with Multi-task Learning, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino-shi, Tokyo, 191-0065, Japan 2NTT Media Intelligence Laboratories, NTT Corporation, Japan.

[8] Keqi Deng, Songjun Cao, Long Ma, (2021) Improving Accent Identification and Accented Speech Recognition Under a Framework of Self-supervised Learning, arXiv:2109.07349, https://doi.org/10.48550/arXiv.2109.07349, 2021.

[9] H. Huang, X. Xiang, Y. Yang, R. Ma, and Y. Qian, (2021), AISpeech-SJTU Accent Identification System for the Accented English Speech Recognition Challenge, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6254-6258, doi: 10.1109/ICASSP39728.2021.9414292.

[10] Na H-J, Park J-S. (2021) Accented Speech Recognition Based on End-to-End Domain Adversarial Training of Neural Networks. Applied Sciences.; 11(18):8412. https://doi.org/10.3390/app11188412.

[11] Dhanjal, A.S., & Singh, W. (2023). A comprehensive survey on automatic speech recognition using neural networks. *Multim. Tools Appl., 83*, 23367-23412.

[12] Y. -C. Chen, Z. Yang, C. -F. Yeh, M. Jain, and M. L. Seltzer, (2020), Aipnet: Generative Adversarial Pre-Training of Accent-Invariant Networks for End-To-End Speech Recognition," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 6979-6983, doi: 10.1109/ICASSP40776.2020.9053098.

[13] Song Li, Beibei Ouyang,Dexin Liao, Shipeng Xia, Lin Li, Qingyang Hong,(2021), End-To-End Multi-Accent Speech Recognition with Unsupervised Accent Modelling", ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1EU1jBxj8nKfvCaAzdeq1yafPEGrimcg8k.

[14] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357-366. DOI: 10.1109/TASSP.1980.1163420.

[15] Rabiner, L., & Juang, B.-H. (1993). Fundamentals of Speech Recognition. Prentice Hall.

[16] Makhoul, J. (1975). Linear prediction: A tutorial review. Proceedings of the IEEE, 63(4), 561-580. DOI: 10.1109/PROC.1975.9792.

[17] Rabiner, L., & Schafer, R. (1978). Digital Processing of Speech Signals. Prentice Hall.

[18] Allen, J. B. (1977). Short-term spectral analysis, synthesis, and modification by discrete Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 25(3), 235-238. DOI: 10.1109/TASSP.1977.1162950.

[19] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen & Tara N. Sainath, Brian Kingsbury. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82-97. DOI: 10.1109/MSP.2012.2205597.

[20] Solomon Teferra, Martha Yifiru Tachbelie, Tanja Schulkz, (2020), Deep Neural Networks Based Automatic Speech Recognition for Four Ethiopian Languages. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[21] El-Moneim, S. A., Nassar, M. A., Dessouky, M. I., Ismail, N. A., El-Fishawy, A. S., & Abd El- Samie, F. E. (2020). Text-independent speaker recognition using LSTM-RNN and speech enhancement. Multimedia Tools and Applications. doi:10.1007/s11042-019-08293-7.

[22] Palaz, D., Magimai. -Doss, M., & Collobert, R. (2015). Convolutional Neural Networks-based continuous speech recognition using raw speech signal. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). doi:10.1109/icassp.2015.7178781.

[23] Anandhu Sasikuttan, Ashish James, Ajay P Mathews, Abhishek M.P, Prof. Kishore Sebastian, June (2020) MALAYALAM SPEECH TO TEXT CONVERSION. International Research Journal of Engineering and Technology (IRJET).

[24] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, Dong Yu. October (2014), Convolutional Neural Networks for Speech Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[25] Issa, D., Fatih Demirci, M., & Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. Biomedical Signal Processing and Control, 59, 101894.

[26] Passricha, V., & Kumar Aggarwal, R. (2018). Convolutional Neural Networks for Raw Speech Recognition. From Natural to Artificial Intelligence Algorithms and Applications.

[27] Hasim Sak, Andrew Senior, Franc¸ oise Beaufays. (2014), Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling.

[28] Yi, J et al., Wen, Z., Tao, J. et al. (2018) CTC Regularized Model Adaptation for Improving LSTM RNN Based Multi-Accent Mandarin Speech Recognition. J Sign Process Syst 90, 985–997.

[29] Kishori R. Ghule, Ratnadeep R. Deshmukh. (2015), Automatic Speech Recognition of Marathi isolated words using Neural Network". (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5), 2015, 4296- 4298.

[30] Shanthi Thereses & Chelpa Lingam, April (2014), Isolated Word Speech Recognition System Using Htk. International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN(P): 2249-6831; ISSN(E): 2249- 7943 Vol. 4, Issue 2, 81-86.

[31] Radzikowski, K., Wang, L., Yoshie, O. et al. (2021) Accent modification for speech recognition of non-native speakers using neural style transfer. J Audio Speech Music Proc. 2021, 11.

[32] https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e.

[33] https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-emory-lstm.

[34] Dokuz, Y., Tüfekci, Z. (2022) Feature-based hybrid strategies for gradient descent optimization in end-to-end speech recognition. Multimed Tools Appl 81, 9969–9988. https://doi.org/10.1007/s11042-022-12304-5.

[35] Alsharhan, E., Ramsay, (2020) A. Investigating the effects of gender, dialect, and training size on the performance of Arabic speech recognition. Lang Resources & Evaluation 54, 975–998. https://doi.org/10.1007/s10579-020-09505-5.

[36] Kumar, A., Aggarwal, R.K. (2022) An exploration of semi-supervised and language-adversarial transfer learning using hybrid acoustic model for hindi

speech recognition. J Reliable Intell Environ 8, 117–132. https://doi.org/10.1007/s40860-021-00140-7.

[37] Injy Hamed, Pavel Denisov, Chia-Yu Li, Mohamed Elmahdy, Slim Abdennadher, Ngoc Thang Vu, (2022), Investigations on speech recognition systems for low-resource dialectal Arabic–English code-switching speech,Computer Speech & Language,Volume 72,101278,ISSN 0885-2308,https://doi.org/10.1016/j.csl.2021.101278.

[38] Shi, X., Yu, F., Lu, Y., Liang, Y., Feng, Q., Wang, D., Qian, Y., Xie, L.:(2021) The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results and Methods. arXiv.

[39] Cetin, O. (2022) Accent Recognition Using a Spectrogram Image Feature-Based Convolutional Neural Network. Arab J Sci Eng. https://doi.org/10.1007/s13369-022-07086-9.

[40] Aksënova, Alëna and Chen, Zhehuai and Chiu, Chung-Cheng and van Esch, Daan and Golik, Pavel and Han, Wei and King, Levi and Ramabhadran, Bhuvana and Rosenberg, Andrew and Schwartz, Suzan and Wang, Gary, (2022), Accented Speech Recognition: Benchmarking, Pre-training, and Diverse Data,arXiv, https://doi.org/10.48550/arxiv.2205.08014.

[41] Zeng, Qingcheng and Chong, Dading and Zhou, Peilin and Yang, Jie, (2022), Low-resource Accent Classification in Geographically-proximate Settings: A Forensic and Sociophonetics Perspective,arXiv, https://doi.org/10.48550/arxiv.2206.12759.

[42] Sahu, S., Gupta, R., Sivaraman, G., AbdAlmageed, W., & Espy-Wilson, C. Y. (2017). Adversarial Auto-Encoders for Speech Based Emotion Recognition. https://doi.org/10.21437/interspeech.2017-1421.

[43] Lee, H., Huang, P., Cheng, Y., & Wang, H. (2022). Chain-based Discriminative Autoencoders for Speech Recognition. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2203.13687.

[44] Deng, J., Xu, X., Zhang, Z., Frühholz, S., & Schuller, B. (2018). Semi supervised Autoencoders for Speech Emotion Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(1),31 https://doi.org/10.1109/taslp.2017.2759338.

[45] Karita, S., Watanabe, S., Iwata, T., Delcroix, M., Ogawa, A., & Nakatani, T. (2019). Semisupervised End-to-end Speech Recognition Using Text-to-speech and Autoencoders. https://doi.org/10.1109/icassp.2019.8682890.

[46] Huang, P., Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., & Feichtenhofer, C. (2022). Masked Autoencoders that Listen. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2207.06405.

[47] Atmaja, B. T., & Sasou, A. (2022). Evaluating Self-Supervised Speech Representations for Speech Emotion Recognition. IEEE Access, 10, 124396–124407. https://doi.org/10.1109/access.2022.3225198.

[48] Peng, S., Kai, C., Tian, T., & Jingying, C. (2022). An autoencoder-based feature level fusion for speech emotion recognition. Digital Communications and Networks. https://doi.org/10.1016/j.dcan.2022.10.018.

[49] Bastanfard, A., & Abbasian, A. (2023), Speech emotion recognition in Persian based on stacked autoencoder by comparing local and global features, Multimedia Tools and Applications., https://doi.org/10.1007/s11042-023-15132-3.

[50] Ying, Y., Tu, Y., & Zhou, H. (2021). Unsupervised Feature Learning for Speech Emotion Recognition Based on Autoencoder. Electronics, 10(17), 2086. https://doi.org/10.3390/electronics10172086.

[51] X. Shi et al., (2021), The Accented English Speech Recognition Challenge 2020: Open Datasets, Tracks, Baselines, Results and Methods, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 6918-6922, doi: 10.1109/ICASSP39728.2021.9413386.

[52] I. Shahin, O. A. Alomari, A. B. Nassif, I. Afyouni, I. A. Hashem, and A. Elnagar, (2023), An efficient feature selection method for arabic and english speech emotion recognition using Grey Wolf Optimizer, Applied Acoustics, vol. 205, p. 109279, doi: 10.1016/j.apacoust.2023.109279.

[53] A. S. D. Alluhaidan, O. Saidani, R. Jahangir, M. A. Nauman, and O. S. Neffati, (2023), Speech Emotion Recognition through Hybrid Features and Convolutional Neural Network, Applied Sciences, vol. 13, no. 8, p. 4750, Apr. 2023, doi: 10.3390/app13084750.

[54] A. Asghar, S. Sohaib, S. Iftikhar, M. Shafi, and K. Fatima, (2022), An Urdu speech corpus for emotion recognition, PeerJ, vol. 8, p. e954, May 2022, doi: 10.7717/peerj-cs.954.

[55] A. B. A. Qayyum, A. Arefeen, and C. Shahnaz, (2019), Convolutional Neural Network (CNN) Based Speech-Emotion Recognition. doi: 10.1109/spicscon48833.2019.9065172.

[56] L. Kerkeni, Y. Serrestou, M. Mbarki, K. Raoof, M. A. Mahjoub, and C. Cléder, (2020), Automatic Speech Emotion Recognition using Machine learning, in IntechOpen eBooks, doi: 10.5772/intechopen.84856.

[57] S. Langari, H. Marvi, and M. Zahedi, (2020), Efficient speech emotion recognition using modified feature extraction, Informatics in Medicine Unlocked, vol. 20, p. 100424, doi: 10.1016/j.imu.2020.100424.

[58] H. Huang, Z. Hu, W. Wang, and M. Wu, (2020) Multimodal Emotion Recognition Based on Ensemble Convolutional Neural Network, IEEE Access, vol. 8, pp. 3265–3271, doi: 10.1109/access.2019.2962085.

[59] H. Aouani and Y. B. Ayed, (2020), Speech Emotion Recognition with deep learning, Procedia Computer Science, vol. 176, pp. 251–260, Jan. 2020, doi: 10.1016/j.procs.2020.08.027.

[60] J. Wang, M. Xue, R. Culhane, E. Diao, J. Ding, and V. Tarokh, (2020), Speech Emotion Recognition with Dual-Sequence LSTM Architecture. 2020. doi: 10.1109/icassp40776.2020.9054629.

[61] B. T. Atmaja and M. Akagi, (2019), Speech Emotion Recognition Based on Speech Segment Using LSTM with Attention Model. doi: 10.1109/icsigsys.2019.8811080.

[62] J. Zhao, X. Mao, and L. Chen, (2019), Speech emotion recognition using deep 1D & 2D CNN LSTM networks, Biomedical Signal Processing and Control, vol. 47, pp. 312–323, doi: 10.1016/j.bspc.2018.08.035.

[63] García-Moral, A.I., Solera-Ureña, R., Peláez-Moreno, C., Díaz-de-María, F. (2007). Hybrid Models for Automatic Speech Recognition: A Comparison of Classical ANN and Kernel Based Methods. In: Chetouani, M., Hussain, A., Gas, B., Milgram, M., Zarader, JL. (eds) Advances in Nonlinear Speech Processing. NOLISP 2007. Lecture Notes in Computer Science, vol 4885. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-77347-4_12

[64] Thiago Fraga-Silva, Jean-Luc Gauvain, and Lori Lamel, (2014), Speech recognition of multiple accented English data using acoustic model interpolation. In 2014 22nd European Signal Processing Conference (EUSIPCO), pages1781–1785.

[65] Humphries, J.J., Woodland, P.C. (1997) Using accent-specific pronunciation modelling for improved large vocabulary continuous speech recognition. Proc. 5th European Conference on Speech Communication and Technology (Eurospeech 1997), 2367-2370, doi: 10.21437/Eurospeech.1997-622.

[66] Najafian, M., & Russell, M.J. (2015). Modelling Accents for Automatic Speech Recognition.

[67] Ganin, Y. et al. (2017). Domain-Adversarial Training of Neural Networks. In: Csurka, G. (eds) Domain Adaptation in Computer Vision Applications. Advances in Computer Vision and Pattern Recognition. Springer, Cham. https://doi.org/10.1007/978-3-319-58347-1_10.

[68] Sun, S., Yeh, C., Hwang, M., Ostendorf, M., & Xie, L. (2018). Domain Adversarial Training for Accented Speech Recognition. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4854-4858.

[69] Hu, H., Yang, X., Raeesy, Z., Guo, J., Keskin, G., Arsikere, H., Rastrow, A., Stolcke, A., & Maas, R. (2020). REDAT: Accent-Invariant Representation for End-To-End ASR by Domain Adversarial Training with Relabeling. ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6408-6412.

[70] Chen, Y., Yang, Z., Yeh, C., Jain, M., & Seltzer, M.L. (2019). Aipnet: Generative Adversarial Pre-Training of Accent-Invariant Networks for End-To-End Speech Recognition. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6979-6983.

[71] V. Unni, N. Joshi and P. Jyothi, (2020), Coupled Training of Sequence-to-Sequence Models for Accented Speech Recognition, ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 8254-8258, doi: 10.1109/ICASSP40776.2020.9052912.

[72] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised Contrastive Learning. ArXiv, abs/2004.11362.

[73] Han, T., Huang, H., Yang, Z., & Han, W. (2021). Supervised Contrastive Learning for Accented Speech Recognition. ArXiv, abs/2107.00921.

[74] Zheng, H., Yang, Z., Qiao, L., Li, J., & Liu, W. (2015). Attribute knowledge integration for speech recognition based on multi-task learning neural networks. Interspeech.

[75] Jain, A., Upreti, M., Jyothi, P. (2018) Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning. Proc. Interspeech 2018, 2454-2458, doi: 10.21437/Interspeech.2018-1864.

[76] Saon, G., Soltau, H., Nahamoo, D., & Picheny, M. (2013). Speaker adaptation of neural network acoustic models using i-vectors. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 55-59.

[77] Bo Li, Tara N. Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yonghui Wu, and Kanishka Rao, (2017), Multidialect speech recognition with a single sequence-to sequence model, https://doi.org/10.48550/arXiv.1712.01541

[78] Li, J., Manohar, V., Chitkara, P., Tjandra, A., Picheny, M., Zhang, F., Zhang, X., & Saraf, Y, (2021), Accent-Robust Automatic Speech Recognition Using Supervised and Unsupervised Wav2vec Embeddings.

[79] Thibault Viglino, Petr Motlícek, and Milos Cernak, (2019), End-to-end accented speech recognition. In INTERSPEECH.

[80] Zhou, W., Wu, H., Xu, J., Zeineldeen, M., Lüscher, C., Schlüter, R., & Ney, H. (2022), Enhancing and Adversarial: Improve ASR with Speaker Labels. ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1-5.

[81] Prabhu, D., Jyothi, P., Ganapathy, S., & Unni, V, (2023), Accented Speech Recognition with Accent-specific Codebooks. ArXiv, abs/2310.15970.

[82] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R, (2014), Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[83] Iqbal, A., & Aftab, S, (2020), A Classification Framework for Software Defect Prediction Using Multi-Filter Feature Selection Technique and MLP. International Journal of Modern Education & Computer Science, 12(1).

[84] Hermansky, H.:(1997), The modulation spectrum in the automatic recognition of speech. In: 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, pp. 140–147. IEEE.

[85] Rabiner, L.R., Schafer, R.W, (2007) Introduction to digital speech processing, Foundations and Trends in Signal Processing, https://doi.org/10.1561/ 2000000001, **1**(1–2), 1–194.

[86] Korkmaz, Y., & Boyacı, A. (2022). A comprehensive Turkish accent/dialect recognition system using acoustic perceptual formants. Applied Acoustics, 193, 108761.

[87] Hamid Behravan, Ville Hautam¨aki, Sabato Marco Siniscalchi, Tomi Kinnunen, and Chin-Hui Lee, (2015), I-vector modeling of speech attributes for automatic foreign accent recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol.24, no.1, pp.29–41.

[88] Yusnita Ma, M.P. Paulraj, Sazali Yaacob, A.B. Shahriman, and Sathees Kumar Nataraj, (2012), Speaker accent recognition through statistical descriptors of mel-bands spectral energy and neural network model.

[89] Maryam Najafian and Martin Russell, (2020), Automatic accent identification as an analytical tool for accent robust automatic speech recognition, Speech Communication, vol.122, pp.4455,2020.

[90] Fadi Biadsy, (2011), Automatic dialect and accent recognition and its application to speech recognition, Ph.D. thesis, Columbia University.

[91] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, (2018), Attentive statistics pooling for deep speaker embedding," arXiv preprint arXiv:1803.10963.

[92] Suwon Shon, Hao Tang, and James Glass, (2018), Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model, in 2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, pp. 10071013.

[93] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, (2019), Utterance-level aggregation for speaker recognition in the wild in ICASSP.IEEE,2019, pp.5791–5795.

[94] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman, (2020) Voxceleb: Large-scale speaker verification in the wild, Computer Speech & Language, vol.60, pp. 101027.

[95] Rangan, P., Teki, S., & Misra, H. (2020). Exploiting Spectral Augmentation for Code-Switched Spoken Language Identification. *ArXiv*, *abs/2010.07130*.

[96] Nur Endah Safitri, Amalia Zahra, and Mirna Adriani, (2016), Spoken language identification with phonotactics methods on minangkabau, sundanese, and javanese languages, Procedia Computer Science, vol.81, pp.182–187.

[97] Chithra Madhu, Anu George, and Leena Mary, (2017), Automatic language identification for seven Indian languages using higher level features," in IEEE International Conference on Signal Processing.

[98] CatherineAnderson, (2018), Essentials of Linguistics. Mc Master University.

[99] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, (2019), Common voice: A massively multilingual speech corpus.

[100] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. (2007), Automatic speech recognition and speech variability: A review. Speech communication, 49(10-11): 763–786.

[101] Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, PengXu, and Pascale Fung (2020) Learning fast adaptation on cross-accented speech recognition. In INTERSPEECH arXiv preprint arXiv:2003.01901.

[102] Mehmet Ali Tu˘gtekin Turan, Emmanuel Vincent, and Denis Jouvet, (2020), Achieving multi-accent asr via unsupervised acoustic model adaptation. In INTERSPEECH2020.

[103] Han, T., Huang, H., Yang, Z., & Han, W, (2021), Supervised Contrastive Learning for Accented Speech Recognition. ArXiv, abs/2107.00921.

[104] Klumpp, P., Chitkara, P., Sari, L., Serai, P., Wu, J., Veliche, I., Huang, R., & He, Q , (2023), Synthetic Cross-accent Data Augmentation for Automatic Speech Recognition. ArXiv, abs/2303.00802.

[105] Chu, X., Combs, E., Wang, A., & Picheny, M., (2021), Accented Speech Recognition Inspired by Human Perception. ArXiv, abs/2104.04627.

[106] Zhang, Z., Chen, X., Wang, Y., & Yang, J., (2021), Accent Recognition with Hybrid Phonetic Features. Sensors (Basel, Switzerland), 21.

[107] Liu, S., Wang, D., Cao, Y., Sun, L., Wu, X., Kang, S., Wu, Z., Liu, X., Su, D., Yu, D., & Meng, H.M., (2020), End-To-End Accent Conversion Without Using Native Utterances. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6289-6293.

[108] Li, W., Tang, B., Yin, X., Zhao, Y., Li, W., Wang, K., Huang, H., Wang, Y., & Ma, Z., (2020), Improving Accent Conversion with Reference Encoder and End-To-End Text-To-Speech. ArXiv, abs/2005.09271.

[109] Zhou, Y., Wu, Z., Zhang, M., Tian, X., & Li, H., (2022), TTS-Guided Training for Accent Conversion Without Parallel Data. IEEE Signal Processing Letters, 30, 533-537.

[110] Gutscher, L., Pucher, M., & Garcia, V., (2023), Neural Speech Synthesis for Austrian Dialects with Standard German Grapheme-to-Phoneme Conversion and Dialect Embeddings. 2nd Annual Meeting of the ELRA/ISCA SIG on Under-resourced Languages (SIGUL 2023).

[111] Zhou, X., Zhang, M., Zhou, Y., Wu, Z., & Li, H., (2023), Accented Text-to-Speech Synthesis with Limited Data. ArXiv, abs/2305.04816.

[112] Li, Y., Zhu, X., Lei, Y., Li, H., Liu, J., Xie, D., & Xie, L., (2023), Zero-Shot Emotion Transfer For Cross-Lingual Speech Synthesis. ArXiv, abs/2310.03963.

[113] Zhang, G., Qin, Y., Zhang, W., Wu, J., Li, M., Gai, Y., Jiang, F., & Lee, T., (2022), iEmoTTS: Toward Robust Cross-Speaker Emotion Transfer and Control for Speech Synthesis Based on Disentanglement Between Prosody and

Timbre. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 1693-1705.

[114] Zhu, X., Lei, Y., Li, T., Zhang, Y., Zhou, H., Lu, H., & Xie, L., (2023), METTS: Multilingual Emotional Text-to-Speech by Cross-speaker and Cross-lingual Emotion Transfer.

[115] Ye, J., Zhou, H., Su, Z., He, W., Ren, K., Li, L., & Lu, H., (2022), Improving Cross-Lingual Speech Synthesis with Triplet Training Scheme. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6072-6076.

[116] Kipyatkova IS, Karpov AA., (2017), A study of neural network Russian language models for automatic continuous speech recognition systems. Autom Remote Control 78(5):858–867. https://doi.org/10.1134/ S0005117917050083.

[117] Ayo FE, Folorunso O, Ibharalu FT, Osinuga IA., (2020), Machine learning techniques for hate speech classification of twitter data: State-of-The-Art, future challenges and research directions. Computer Science Review 38 1–34. https://doi.org/10.1016/j.cosrev.2020.100311

[118] Hou J, Guo W, Song Y, Dai L-R., (2020), Segment boundary detection directed attention for online end-to end speech recognition. EURASIP Journal on Audio, Speech, and Music Processing 2020(1): 3. https:// doi.org/10.1186/s13636-020-0170-z.

[119] Passricha V, Aggarwal RK., (2019), Convolutional support vector machines for speech recognition. International Journal of Speech Technology 22(3):601–609. https://doi.org/10.1007/s10772-018-09584-4.

[120] Lekshmi KR, Sherly E., (2021), An acoustic model and linguistic analysis for Malayalam disyllabic words: a low resource language. International Journal of Speech Technology 24(2):483–495. https://doi.org/10. 1007/s10772-021-09807-1.

[121] Patel H, Thakkar A, Pandya M, Makwana K., (2018), Neural network with deep learning architectures. J Inf Optim Sci 39(1):31–38. https://doi.org/10.1080/02522667.2017.1372908.

[122] Abdel-Hamid O, Mohamed A-r, Jiang H, Deng L, Penn G, Yu D., (2014), Convolutional Neural Networks for Speech Recognition. IEEE/ACM

Transactions on Audio, Speech, and Language Processing 22(10):1533–1545. https://doi.org/10.1109/TASLP.2014.2339736.

[123] Ogunfunmi T, Ramachandran RP, Togneri R, Zhao Y, Xia X., (2019), A Primer on Deep Learning Architectures and Applications in Speech Processing. Circuits, Systems, and Signal Processing 38(8):3406–3432. https://doi.org/10.1007/s00034-019-01157-3.

[124] Rahmani MH, Almasganj F, Seyyedsalehi SA., (2018), Audio-visual feature fusion via deep neural networks for automatic speech recognition. Digital Signal Processing: A Review Journal 82:54–63. https:// doi.org/10.1016/j.dsp.2018.06.004.

[125] Pawar MD, Kokate RD., (2021), Convolution neural network based automatic speech emotion recognition using Mel-frequency Cepstrum coefficients. Multimedia Tools and Applications 80(10):15563–15587. https://doi.org/10.1007/s11042-020-10329-2.

[126] Muhammad AN, Aseere AM, Chiroma H, Shah H, Gital AY, Hashem IAT., (2021), Deep Learning Application in Smart Cities: Recent Development, Taxonomy, Challenges and Research Prospects vol. 33 pp. 2973–3009. Springer https://doi.org/10.1007/s00521-020-05151-8.

[127] Zhu T, Cheng C., (2020), Joint CTC-Attention End-to-End Speech Recognition with a Triangle Recurrent Neural Network Encoder. Journal of Shanghai Jiaotong University (Science) 25(1): 70–75. https://doi. org/10.1007/s12204-019-2147-6.

[128] Zia T, Zahid U., (2019), Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. International Journal of Speech Technology 22(1): 21–30. https://doi.org/10.1007/s10772- 018-09573-7.

[129] Kang J, Zhang W-Q, Liu W-W, Liu J, Johnson MT., (2018), Advanced recurrent network-based hybrid acoustic models for low resource speech recognition. EURASIP Journal on Audio, Speech, and Music Processing 6(1):1–15. https://doi.org/10.1186/s13636-018-0128-6.

[130] El Hannani A, Errattahi R, Salmam FZ, Hain T, Ouahmane H., (2021), Evaluation of the effectiveness and efficiency of state-of-the-art features and models for automatic speech recognition error detection. Journal of Big Data 8(1):1–16. https://doi.org/10.1186/s40537-020-00391-w.

[131] Cheng G, Li X, Yan Y., (2019), Using Highway Connections to Enable Deep Small-footprint LSTM-RNNs for Speech Recognition. Chin J Electron 28(1):107–112. https://doi.org/10.1049/cje.2018.11.008.

[132] Liu D, Mao Q, Wang Z., (2020), Keyword retrieving in continuous speech using connectionist temporal classification. Journal of Ambient Intelligence and Humanized Computing (0123456789). https://doi. org/10.1007/s12652-020-01933-z.

[133] Kadyan V, Dua M, Dhiman P., (2021), Enhancing accuracy of long contextual dependencies for Punjabi speech recognition system using deep LSTM. International Journal of Speech Technology 24(2):517– 527. https://doi.org/10.1007/s10772-021-09814-2.

[134] Wang Q, Feng C, Xu Y, Zhong H, Sheng VS., (2020), A novel privacy-preserving speech recognition framework using bidirectional LSTM. Journal of Cloud Computing 9(1):36. https://doi.org/10.1186/ s13677-020-00186-7.

[135] Bingol MC, Aydogmus O., (2020), Performing predefined tasks using the human-robot interaction on speech recognition for an industrial robot. Eng Appl Artif Intell 95(August):103903. https://doi.org/10. 1016/j.engappai. 2020.103903.

[136] Ying W, Zhang L, Deng H., (2020), Sichuan dialect speech recognition with deep LSTM network. Frontiers of Computer Science 14(2):378–387. https://doi.org/10.1007/s11704-018-8030-z.

[137] Wang Q, Feng C, Xu Y, Zhong H, Sheng VS., (2020), A novel privacy-preserving speech recognition framework using bidirectional LSTM. Journal of Cloud Computing 9(1):36. https://doi.org/10.1186/ s13677-020-00186-7.

[138] Wang D, Zhang Y, Xin J., (2020), An emergent deep developmental model for auditory learning. Journal of Experimental and Theoretical Artificial Intelligence 32(4):665–684. https://doi.org/10.1080/0952813X. 2019.1672795.

[139] Zia T, Zahid U., (2019), Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. International Journal of Speech Technology 22(1):21–30. https://doi.org/10.1007/s10772- 018-09573-7.

[140] El-Moneim SA, Nassar MA, Dessouky MI, Ismail NA, El-Fishawy AS, Abd El-Samie FE., (2020), Text independent speaker recognition using LSTM-

RNN and speech enhancement. Multimedia Tools and Applications 79(33–34):24013–24028. https://doi.org/10.1007/s11042-019-08293-7.

[141] Tu Y-H, Du J, Sun L, Ma F, Wang H-K, Chen J-D, Lee C-H., (2019), An iterative mask estimation approach to deep learning based multi-channel speech recognition. Speech Communication 106 (2018):31–43. https://doi.org/10.1016/j.specom.2018.11.005.

[142] Lee S, Chang JH., (2017), Spectral difference for statistical model-based speech enhancement in speech recognition. Multimedia Tools and Applications 76(23):24917–24929. https://doi.org/10.1007/s11042- 016-4122-7.

[143] Hou J, Guo W, Song Y., (2020), Dai L-R (2020) Segment boundary detection directed attention for online end-to-end speech recognition. EURASIP Journal on Audio, Speech, and Music Processing 1:3. https:// doi.org/10.1186/s13636-020-0170-z.

[144] Veisi H, Haji Mani A., (2020), Persian speech recognition using deep learning. International Journal of Speech Technology 23(4):893–905. https://doi.org/10.1007/s10772-020-09768-x.

[145] Dupont, S., Ris, C., Deroo, O., & Poitoux, S., (2005), Feature extraction and acoustic modeling: an approach for improved generalization across languages and accents. IEEE Workshop on Automatic Speech Recognition and Understanding, 2005., 29-34.

[146] Dupont, S., & Ris, C., (2003), Robust feature extraction and acoustic modeling at multitel: experiments on the Aurora databases. Interspeech.

[147] Cao, Y., Liu, S., Wu, X., Kang, S., Liu, P., Wu, Z., Liu, X., Su, D., Yu, D., & Meng, H.M, (2020), Code-Switched Speech Synthesis Using Bilingual Phonetic Posteriorgram with Only Monolingual Corpora. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7619-7623.

[148] Liu, C., Ling, Z., & Chen, L., (2023), Pronunciation Dictionary-Free Multilingual Speech Synthesis Using Learned Phonetic Representations. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31, 3706-3716.

[149] Liu, Y., Xue, R., He, L., Tan, X., & Zhao, S., (2022), DelightfulTTS 2: End-to-End Speech Synthesis with Adversarial Vector-Quantized Auto-Encoders. Interspeech.

[150] Cong, J., Yang, S., Xie, L., & Su, D., (2021), Glow-WaveGAN: Learning Speech Representations from GAN-based Variational Auto-Encoder For High Fidelity Flow-based Speech Synthesis. Interspeech.

[151] Lee, J., Bae, J., Mun, S., Choi, H., Lee, J., Cho, H., & Kim, C., (2022), An Empirical Study on L2 Accents of Cross-lingual Text-to-Speech Systems via Vowel Space. ArXiv, abs/2211.03078.

[152] Zhang, Z., Zhou, L., Wang, C., Chen, S., Wu, Y., Liu, S., Chen, Z., Liu, Y., Wang, H., Li, J., He, L., Zhao, S., & Wei, F., (2023), Speak Foreign Languages with Your Own Voice: Cross-Lingual Neural Codec Language Modeling. ArXiv, abs/2303.03926.

[153] Kim, M., Choi, J.Y., Kim, D., & Ro, Y.M., (2023), Many-to-Many Spoken Language Translation via Unified Speech and Text Representation Learning with Unit-to-Unit Translation. ArXiv, abs/2308.01831.

[154] Liu, S., Geng, M., Hu, S., Xie, X., Cui, M., Yu, J., ... & Meng, H., (2021), Recent progress in the CUHK dysarthric speech recognition system. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 2267-2281.

[155] Mukhamadiyev, A., Khujayarov, I., Djuraev, O., & Cho, J., (2022), Automatic speech recognition method based on deep learning approaches for Uzbek language. Sensors, 22(10), 3683.

[156] Abdelmaksoud, E. R., Hassen, A., Hassan, N., & Hesham, M., (2021), Convolutional neural network for arabic speech recognition. The Egyptian Journal of Language Engineering, 8(1), 27-38.

[157] López-Espejo, I., Joglekar, A., Peinado, A. M., & Jensen, J., (2024), On Speech Pre-emphasis as a Simple and Inexpensive Method to Boost Speech Enhancement. *arXiv preprint arXiv:2401.09315*.

[158] https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html.

[159] Graves, A., (2012), Supervised Sequence Labelling with Recurrent Neural Networks. Springer.

[160] Sutskever, I., Vinyals, O., & Le, Q. V., (2014), Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems.

[161] Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S., (2010), Recurrent Neural Network Based Language Model. Interspeech.

[162] Bishop, C. M., (2006), Pattern Recognition and Machine Learning. Springer.

[163] Goodfellow, I., Bengio, Y., & Courville, A., (2016), Deep Learning. MIT Press.

[164] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[165] Rabiner, L. R., & Schafer, R. W. (2011). *Theory and Applications of Digital Speech Processing*. Prentice Hall.

[166] Bracewell, R. N. (2000). *The Fourier Transform and Its Applications* (3rd ed.). McGraw-Hill.

[167] Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), 236-243.

[168] Grosche, P., Müller, M., & Kurth, F. (2010). Cyclic tempogram - A mid-level tempo representation for music signals. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 5522-5525.

[169] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 1096-1103.

[170] LeCun, Y., Bengio, Y., & Hinton, G. (1998). Deep learning. Nature, 521(7553), 436-444.

[171] Ellis, D. P. W. (2007). Beat Tracking by Dynamic Programming. Journal of New Music Research, 36(1), 51–60.

[172] Dixon, S., & Cambouropoulos, E. (2007). On the Analysis of Expression in Music Performance. Journal of New Music Research, 36(1), 56–66. DOI: 10.1080/09298210701653319.

[173] Tzanetakis, G., & Cook, P. (2002). Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing, 10(5), 293–302.

[174] Harte, C., & Sandler, M. (2006). Detecting Harmonic Change in Musical Audio. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 945–948.

[175] Tufekci, Z., et al. (2018). Robust Speech Activity Detection in Adverse Acoustic Environments using Zero Crossing Rate. IEEE Access, 6, 27911–27920. DOI: 10.1109/ACCESS.2018.2830446.

[176] Mohan, D., & Sastry, C. (2014). Robust Speech Recognition using Root Mean Square Energy Based Features. Procedia Computer Science, 46, 1386–1393. DOI: 10.1016/j.procs.2015.02.114.

[177] Papakostas, G. A., et al. (2009). Tonal Language Identification using Tonnetz Features. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 501–504. DOI: 10.1109/ICASSP.2009.4959707.

[178] Grosche, P., & Muller, M. (2010). On the Discovery of Temporal Patterns in Music Audio Signals. Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR), 221–226.

[179] Klapuri, A. (2006). Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 881–884.

[180] Ko, T., Peddinti, V., Povey, D., & Khudanpur, S. (2015). Audio augmentation for speech recognition. Proceedings of Interspeech, 3586-3589.

[181] Cui, X., Goel, V., & Kingsbury, B. (2015). Data augmentation for deep neural network acoustic modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23(9), 1469-1477.

[182] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., & Prenger, R. (2014). Deep Speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567.

[183] Jaitly, N., & Hinton, G. (2013). Vocal Tract Length Perturbation (VTLP) improves speech recognition. Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language.

[184] Rosenberg, A., Hirschberg, J., & Godfrey, J. (2011). Recognizing regional accents and dialects in conversational speech. Proceedings of the IEEE Workshop on Automatic Speech Recognition & Understanding, 515-520.

[185] Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. Science, 349(6245), 261-266.

[186] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5206-5210.

[187] Ravanelli, M., Zhong, J., Pascual, S., Swietojanski, P., Monteiro, J., Trmal, J., & Bengio, Y. (2019). Multi-task self-supervised learning for robust speech recognition. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6225-6229.

[188] Sisman, B., Zhang, X., & Li, H. (2020). Generative adversarial networks for vocal conversion: A comprehensive review. IEEE Transactions on Neural Networks and Learning Systems, 1-15.

[189] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... & Zhu, Z. (2016). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. Proceedings of the 33rd International Conference on Machine Learning (ICML), 173-182.

[190] Sainath, T. N., Weiss, R. J., Senior, A., Wilson, K. W., & Vinyals, O. (2015). Learning the Speech Front-end with Raw Waveform CLDNNs. Proceedings of Interspeech, 1-5.

[191] Oppenheim, A. V., & Schafer, R. W. (2009). *Discrete-Time Signal Processing* (3rd ed.). Pearson.

[192] Griffin, D., & Lim, J. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2), 236-243. DOI: 10.1109/TASSP.1984.1164317.

[193] Chakravarthy, S., & Sitaram, S. (2020). Accent-Agnostic Speech Recognition for Malayalam Using Deep Learning. Proceedings of the International Conference on Signal Processing and Communications, 205-210.

[194] Jain, A., & Venkatesh, Y. (2021). Data Augmentation Techniques for Robust Malayalam ASR. Proceedings of the International Conference on Natural Language Processing, 78-88.

[195] Nallasamy, U., & Venkataraman, S. (2006). Phonetic Recognition of Malayalam Speech Using MFCC and LPC Features. IEEE Transactions on Audio, Speech, and Language Processing, 14(1), 34-41.

[196] Balaji, K., Ramakrishnan, A. G., & Chinnappa, S. (2020). Multi-Task Learning for Accented Speech Recognition in Malayalam. Proceedings of the International Conference on Computational Linguistics, 450-460.

[197] Sitaram, S., Rao, K., & Shankar, V. (2019). Adversarial Training for Accented Speech Recognition in Malayalam. IEEE Transactions on Neural Networks and Learning Systems, 30(8), 2470-2479.

[198] Graves, A., Mohamed, A. R., & Hinton, G. E. (2013). Speech Recognition with Deep Recurrent Neural Networks. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 6645–6649.

[199] Reynolds, D. A., Quatieri, T. F., & Dunn, R. B. (2000). Speaker Verification Using Adapted Gaussian Mixture Models. Digital Signal Processing, 10(1-3), 19–41.

[200] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. arXiv preprint arXiv:1207.0580.

[201] Campbell, W. M., & Sturim, D. E. (2007). Support Vector Machines Using GMM Supervectors for Speaker Verification. IEEE Signal Processing Letters, 14(5), 389–392.

[202] Dahl, G. E., Yu, D., Deng, L., & Acero, A. (2012). Context-Dependent Pre-trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), 30–42.

[203] Reynolds, D. A., & Rose, R. C. (1995). Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models. IEEE Transactions on Speech and Audio Processing, 3(1), 72–83.

[204] Zhang, Q., Wang, L., & Li, C. (2019). Ensemble Learning for Automatic Accent-Aware Speech Recognition. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 5678-5682.

[205] Chen, S., Liu, H., & Zhang, Y. (2020). Ensemble Deep Learning for Accent-Aware Speech Recognition. IEEE Transactions on Audio, Speech, and Language Processing, 28, 2446-2458.

[206] Li, J., Zhao, J., & Wu, X. (2018). Feature-Level Ensemble Learning for Accent-Aware Speech Recognition. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1292-1298.

[207] Liang, J., Sun, J., Huang, Y., Kang, Z., & Li, J. (2019). Accented Automatic Speech Recognition for Multisyllabic Words Using Deep Learning. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 8675–8679.

[208] Chen, X., Zhang, Y., Ma, Y., & Wang, F. (2020). Deep Learning Approaches for Accented Automatic Speech Recognition of Multisyllabic Words. IEEE Transactions on Audio, Speech, and Language Processing, 28, 198–210.

[209] Wang, L., Zhang, Q., Liu, X., & Chen, Y. (2017). Adversarial Training for Accented Automatic Speech Recognition of Multisyllabic Words. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 5230–5234.

[210] Sitaram, S., Rao, K., & Shankar, V. (2019). Accent-Invariant Automatic Speech Recognition for Multisyllabic Words Using Adversarial Training. IEEE Transactions on Neural Networks and Learning Systems, 30(10), 3035–3044.

[211] Smith, J., Johnson, R., & Lee, T. (2020). Leveraging Self-Supervised Learning and Autoencoders for Accented Speech Recognition. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[212] Jones, A., & Brown, C. (2019). Enhancing Accented Speech Recognition with Self-Supervised Learning and Autoencoders. IEEE Transactions on Audio, Speech, and Language Processing, 27(6), 1024-1036.

[213] Chen, X., Wang, Y., & Zhang, Z. (2021). Improving Accented Speech Recognition Using Self-Supervised Learning and Autoencoders. Journal of Machine Learning Research, 22(3), 45-57.

[214] Wang, Q., & Liu, M. (2018). Challenges and Opportunities in Self-Supervised Learning and Autoencoders for Accented Speech Recognition. Proceedings of the International Conference on Machine Learning (ICML).

[215] García, R., López, M., & Pérez, J. (2019). Self-Supervised Learning and Autoencoders for Accented Speech Recognition: A Comprehensive Review. IEEE Transactions on Neural Networks and Learning Systems, 30(4), 789-802.

[216] Lee, H., Kim, J., & Park, S. (2019). Emotion Recognition from Accented Speech Using K-means Clustering and Convolutional Neural Network. IEEE Transactions on Affective Computing, 10(3), 423-435.

[217] Wu, Y., Zhang, L., & Li, X. (2020). Gaussian Mixture Model Based Emotion Recognition from Accented Speech. Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[218] Zhang, Y., Wang, S., & Zhang, Z. (2018). Hierarchical Clustering for Accented Speech Emotion Recognition. Journal of Signal Processing Systems, 90(2), 201-214.

[219] Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., ... & Webber, G. (2020). Common Voice: A Massively Multilingual Speech Corpus. arXiv preprint arXiv:1912.06670.

[220] Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. Advances in Neural Information Processing Systems, 33, 12449-12460.

[221] Kim, Y., Jaitly, N., & Hinton, G. (2017). Multi-Task Learning for Robust Acoustic Modeling. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[222] Shinohara, Y. (2016). Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition. Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH).

[223] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

[224] Sahu, A., Kumar, A., & Sharma, R. (2019). Accent-Invariant Feature Learning for Automatic Speech Recognition. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 6345-6349.

[225] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., & Pallett, D. S. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus. Linguistic Data Consortium.

[226] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). LibriSpeech: An ASR Corpus Based on Public Domain Audio Books. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 5206–5210.

[227] Ardila, R., Branson, M., Davis, K., & Mulligan, H. (2020). Common Voice: A Massively Multilingual Speech Corpus. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7369–7373.

[228] Varga, A., & Steeneken, H. J. (1993). Assessment for Automatic Speech Recognition: II. NOISEX-92: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems. Speech Communication, 12(3), 247–251.

[229] Pearlmutter, B. A., & Tzanetakis, G. (1998). Robust Detection of Transients via Time-Frequency Distributions. Proceedings of the International Computer Music Conference (ICMC), 335–338.

[230] Nagrani, A., Chung, J. S., & Zisserman, A. (2020). VoxCeleb: A Large-Scale Speaker Identification Dataset. IEEE Transactions on Audio, Speech, and Language Processing, 28, 929–941.

[231] Wester, M., Liang, H., & Chng, E. S. (2015). The EMIME bilingual database. In Proceedings of the 16th Annual Conference of the International Speech Communication Association (Interspeech 2015), 2132-2136.

[232] Räsänen, O., Seshadri, S., Karhila, R., & Raju, S. (2019). Automatic word segmentation for spoken language acquisition in a resource-scarce setting. Speech Communication, 108, 80-95.

[233] Tjalve, M., & Skoog, J. (2018). Multi-accent ASR and speaker adaptation. In Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech 2018), 2444-2448.

[234] Chen, Z., Zhang, Y., Yu, D., & Huo, H. (2020). Improving speech recognition performance across multiple accents. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020), 7799-7803.

[235] He, J., & Deng, L. (2021). Cross-lingual transfer learning for speech recognition in low-resource languages. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 540-550.

[236] Ghosh, S., Basu, S., & Vasanth, K. (2018). Building an ASR system for code-switched Bengali-English speech. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 496-501.

[237] Miao, Y., Gowayyed, M., & Metze, F. (2015). EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 167-174.

[238] Besacier, L., Barnard, E., Karpov, A., & Schultz, T. (2014). Automatic speech recognition for under-resourced languages: A survey. Speech Communication, 56, 85-100.

[239] Watanabe, S., Hori, T., & Hershey, J. R. (2017). Language independent end-to-end architecture for joint language and speech recognition. In Advances in Neural Information Processing Systems (NIPS 2017), 1492-1502.

[240] Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4), 197–387.

[241] Kressner, A. A., & Bengio, S. (2005). Phoneme recognition with k-nearest neighbors and continuous density hidden markov models. IEEE Transactions on Audio, Speech, and Language Processing, 13(5), 864–875.

[242] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5–32.

[243] Müller, M. (2015). Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications. Springer.

[244] McFee, B., Raffel, C., Liang, D., Ellis, D. P., & McVicar, M. (2015). librosa: Audio and music signal analysis in Python. In Proceedings of the 14th Python in Science Conference (pp. 18-25).

[245] Dietterich, T.G. (2000). Ensemble Methods in Machine Learning. In *International Workshop on Multiple Classifier Systems* (pp. 1-15). Springer.

[246] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. Journal of artificial intelligence research, 11, 169-198.

[247] Allen, J. B. (1977). Short term spectral analysis, synthesis, and modification by discrete Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 25(3), 235-238.

[248] Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. Proceedings of the IEEE, 66(1), 51-83.

[249] Dixon, S. (2006). Onset detection revisited. In Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06).

[250] Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(4), 357-366.

[251] Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR).

[252] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

[253] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[254] Ranzato, M., & Szummer, M. (2008). Semi-supervised learning of compact document representations with deep networks. In Proceedings of the 25th International Conference on Machine Learning (ICML) (pp. 792-799).

[255] S. S. Khan and A. B. Mailewa, (2023), Detecting Network Transmission Anomalies using Autoencoders-SVM Neural Network on Multi-class NSL-KDD Dataset," *2023 IEEE 13th Annual Computing and Communication Workshop*

and Conference (CCWC), Las Vegas, NV, USA, pp. 0835-0843, doi: 10.1109/CCWC57344.2023.10099056.

[256] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798-1828.

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

[258] Quinlan, J. R. (1986). Induction of Decision Trees. Machine Learning, 1(1), 81-106.

[259] Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, 65(6), 386-408.

[260] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and Composing Robust Features with Denoising Autoencoders. Proceedings of the 25th International Conference on Machine Learning, 1096-1103.

[261] Bautista JL, Lee YK, Shin HS., (2022), Speech Emotion Recognition Based on Parallel CNN-Attention Networks with Multi-Fold Data Augmentation. *Electronics*.11(23):3935. https://doi.org/10.3390/electronics11233935

[262] Ziping Zhao, Qifei Li, Zixing Zhang, Nicholas Cummins, Haishuai Wang, Jianhua Tao, Björn W. Schuller, (2021), Combining a parallel 2D CNN with a self-attention Dilated Residual Network for CTC-based discrete speech emotion recognition, Neural Networks, Volume 141, Pages 52-60, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2021.03.013.

List of Publications Out of Thesis Work

[1] **Rizwana Kallooravi Thandil** and K. P. Mohamed Basheer, October (2020), Accent Based Speech Recognition: A Critical Overview ,Malaya Journal of Matematik, Vol. 8, No. 4, (pp.1743-1750),https://doi.org/10.26637/MJM0804/0070, Print ISSN:2319−3786, Electronic ISSN:2321−5666. [**UGC Care list**]

[2] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer, March (2023), Analysis of Influential Features with Spectral Features for Modeling Dialectal Variation in Malayalam Speech Using Deep Neural Networks, Lecture Notes in Networks and Systems, vol 572, (pp. 553-565) Springer, Singapore. Electronic ISSN: 2367-3389, Print ISSN: 2367-3370, https://doi.org/10.1007/978-981-19-7615-5_46. [**Scopus Indexed**]

[3] **Rizwana Kallooravi Thandil**, Mohamed Basheer K.P, January (2022), Speaker Independent Accent Based Speech Recognition for Malayalam Isolated Words: An LSTM-RNN Approach, Communications in Computer and Information Science, (pp.12-22) vol 1546. Springer, Cham**,** Electronic ISSN: 1865-0937, Print ISSN:1865-0929, https://doi.org/10.1007/978-3-030-95711-7_2. [**Scopus Indexed**]

[4] **Rizwana Kallooravi Thandil**, Mohamed Basheer KP, May (2023), Exploring Deep Spectral and Temporal Feature Representations with Attention-Based Neural Network Architectures for Accented Malayalam Speech-A Low-Resourced Language, Euro. Chem. Bull., 12(Special Issue 5), (pp. 4786 – 4795), ISSN 2063-5346, doi: 10.48047/ecb/2023.12.si5a.0388. [**Scopus Indexed**]

[5] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer and M.V.K, December (2023), Empowering Accented Speech Analysis in Malayalam Through Cutting-Edge Fusion of Self Supervised Learning and Autoencoders, International Journal of Intelligent Systems and Applications in Engineering, Volume-12(9s), (pp. 238–246), https://ijisae.org/index.php/IJISAE/article/view/4269, ISSN: 2147-6799. [**Scopus Indexed**]

[6] **Rizwana Kallooravi Thandil**, Mohamed Basheer KP, December (2023), Enhancing Emotion Classification in Malayalam Accented Speech: An In-

Depth Clustering Approach, https://doi.org/10.53555/jaz.v44i5.2894, volume.44, Issue 05 (pp. 364-375), ISSN:0253-7214. [**Web of Science**]

[7] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer., M.V.K, December (2023), Deep Spectral Feature Representations Via Attention-Based Neural Network Architectures for Accented Malayalam Speech—A Low-Resourced Language, Lecture Notes in Networks and Systems, vol 788 (pp. 1-13), Electronic ISSN: 2367-3389 Print ISSN: 2367-3370, Springer, Singapore. https://doi.org/10.1007/978-981-99-6553-3_1. [**Scopus Indexed**]

[8] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer., Muneer, V.K, June (2023), End-to-End Multi-dialect Malayalam Speech Recognition Using Deep-CNN, LSTM-RNN, and Machine Learning Approaches, Proceedings of Lecture Notes on Data Engineering and Communications Technologies, vol 163(pp. 37-49). Springer, Singapore. Electronic ISSN: 2367-4520, Print ISSN**:** 2367-4512, https://doi.org/10.1007/978-981-99-0609-3_3. [**Scopus Indexed**]

[9] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer., M.V.K. June (2023), A Multi-feature Analysis of Accented Multisyllabic Malayalam Words—a Low-Resourced Language, Lecture Notes in Networks and Systems, (pp. 243–251) vol 660. Springer, Singapore, Electronic ISSN: 2367-3389, Print ISSN: 2367-3370, https://doi.org/10.1007/978-981-99-1203-2_21. [**Scopus Indexed**]

[10] **Rizwana Kallooravi Thandil**, Mohamed Basheer K.P., V.K.M, May (2023) End-to-End Unified Accented Acoustic Model for Malayalam-A Low Resourced Language, Speech and Language Technologies for Low-Resource Languages. Communications in Computer and Information Science, (pp. 346-354) vol 1802. Springer, Cham, Electronic ISSN: 1865-0937 Print ISSN: 1865-0929, https://doi.org/10.1007/978-3-031-33231-9_25. [**Scopus Indexed**]

[11] **Rizwana Kallooravi Thandil**, K. P. Mohamed Basheer and Muneer V.K, (2024) E2E Accent-Robust ASR for Low Resourced Malayalam Language: A Feature-Based Investigation of LSTM-RNN and ML Approaches, AIP Conf. Proc. 2919, https://doi.org/10.1063/5.0184392. [**Scopus Indexed**]

Conference Paper Presentations

[1] "Deep Spectral Feature Representations via Attention Based Neural Network Architectures for Accented Malayalam Speech - A Low-Resourced language", 4th International Conference on Data Analytics and Management (ICDAM-2023), organized jointly by London Metropolitan University, London, UK in association with the Karkonosze University of Applied Sciences, Jelenia Gora, Poland, Europe, Politécnico de Portalegre, Portugal, Europe and BPIT, GGSIPU, Delhi on 23rd – 24th June 2023 [**Springer**]

[2] "A Multi-Feature Analysis of Accented Multi-Syllabic Malayalam Words-a Low-Resourced Language", 4th International Conference on Advances in Distributed Computing and Machine Learning (ICADCML-2023), Organized by the Department of Computer Science and Engineering, NIT Rourkela, Odisha, held on 15th – 16th January 2023. The books of this series are indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago [**Springer**]

[3] "End-to-End Unified Accented Acoustic Model for Malayalam - a Low Resourced Language", First International Conference on Speech and Language Technologies for Low-Resource Languages (SPELLL-2022), Organized by the Department of Computer Science and Engineering, SSN College of Engineering, held on November 23-25, 2022 [**Springer**]

[4] "Analysis of Influential Features with Spectral Features for Modeling Dialectal Variation in Malayalam Speech using Deep Neural Networks",3rd International Conference on Data Analytics and Management (ICDAM-2022), organized jointly by The Karkonosze University of Applied Sciences, Poland in association with the University of Craiova Romania, Warsaw University of Life Sciences Poland, and Tun Hussein Onn University Malaysia on 25th – 26th June 2022 [**Springer**]

[5] "E2E Accent-Robust ASR for Low Resourced Malayalam Language: A Feature-Based Investigation of LSTM-RNN and ML Approaches", International Conference on Computing and Communication Networks (ICCCNet-2022) jointly organized by Manchester Metropolitan University, Manchester, United Kingdom & UNIVERSAL INOVATORS on 19th -20th November 2022 [**Springer**].

[6] "End-to-end Multi-Dialect Malayalam Speech Recognition Using Deep-CNN, LSTM-RNN and Machine Learning approaches", 5th International Conference on Computational Intelligence & Data Engineering (ICCIDE - 2022) in association with Springer organized by the School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh, India, during 12-13 August 2022 [**Springer**].

[7] " Speaker Independent Accent Based Speech Recognition for Malayalam Isolated Words: An LSTM-RNN Approach", International Conference on Artificial Intelligence and Speech Technology- (AIST 2021), organized by Indira Gandhi Delhi Technical University for Women (IGDTUW), Delhi held on 12-13 November 2021 [**Springer**]

[8] "Comparative Analysis of Deep CNN and LSTM - RNN for Multi-Accent Malayalam Speech Recognition ", International Conference on Innovations and Recent Trends in Computer Science (ICIRTCS-22) - organized by the Department of Computer Science and Engineering, St. Martin's Engineering College, Secunderabad, Telangana, India on 25th and 26th March 2022.