TinyML for Emotion Detection in Voice Signals: Evaluating and Proposing Algorithms for IoT Wearable Devices

by

Hasibul Hasan Ahmed
24141144
Zain Ahmed
20101117
Tshewang Choden
20201207
Nutan Chaudhary
20201199

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
May 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing a degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Hasibul hasan Ahmed
20301423

Zain Ahmed
20101117

Tshewang Choden
20201207

Nutan Chaudhary
20201199

Approval

TinyML for Emotion Detection in Voice Signals: Evaluating and Proposing Algorithms for IoT Wearable Devices

1. Hasibul Hasan Ahmed (20301423)
2. Zain Ahmed (20101117)
3. Tshewang Choden (20201207)
4. Nutan Chaudhary (20201199)

Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science.

Examining Committee:

Supervisor:
(Member)

Dr. Dr. Amitabha Chakrabarty
Associate professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

This thesis work adheres to the highest standards of ethical research and academic integrity. The research was conducted following the ethical guidelines and principles set forth by BRAC University and relevant professional bodies. Also, all the work conducted with a commitment to honesty, transparency, and integrity. All data were collected, analyzed, and reported accurately and without fabrication, falsification, or misrepresentation by the research team.

Abstract

In today's digital world, voice emotion recognition is essential for applications like intelligent tutoring, audio mining, security, telecommunication, HCI, lie detection, and human-machine interactions in various settings. Voice, which is used to express one's perspective and communicate inter-personally, is one of the characteristics that differentiate humans. The rise of IoT and wearable technology offers new opportunities for real-time, remote emotion detection through voice. In the context of voice processing-based emotion recognition, particularly in the Internet of Things wearable, this thesis investigates the possibilities of tiny machine learning or TinyML. To accomplish this goal, we evaluated Bidirectional-LSTM and CNN on both vector quantization and raw data gave us notable accuracy of 88%, 80%, 85%, and 81% respectively and LSTM, Random Forest, Logistic Regression, KNN and GRU on only raw data shows accuracy rates of 86%, 89%, 89%, 86% and 82% using the composite dataset that includes well-known datasets such as RAVDESS, CREMA-D, TESS, and SAVEE. Furthermore, the models with the best accuracy were selected to be implemented within the TinyML framework, Tensorflow-lite. Our benchmarks highlighted that most of the best performing models were Recurrent Neural Network (RNN) based, notably BiLSTM, LSTM, GRU alongside the CNN model. Finally, after validating the findings through hardware implementation on Raspberry Pi 4, the study concludes that BiLSTM model would be most suitable for speech emotion recognition tasks (SER) in the TinyML domain . The hardware performance of the model illustrates how confident the model actually is in predicting emotions from raw voice input within significant resource and power constraints . These findings contribute to the ongoing discourse on the intersection of voice emotion recognition, TinyML, and IoT, showcasing the potential for enhanced human-machine interactions in a wide variety of practical domains.

Keywords: Tiny Machine Learning (TinyML); Emotion Recognition; SER; Voice Signals; Wearable IoT Devices; BiLSTM; CNNs; LSTM; GRU; KNN

Dedication

This research is dedicated to the resilient spirit of the people of Bangladesh, whose health and well-being inspire us to find innovative solutions and bridge the gaps in healthcare, one step at a time. To the countless individuals who generously shared their time, knowledge, and experiences, this research is a tribute to your invaluable contributions.

Acknowledgment

Firstly, all praise to the Great Allah for whom our thesis has been completed without any major interruption.

Secondly, we appreciate the guidance and feedback provided by our honorable supervisor Professor Dr. Amitabha Chakrabarty, PhD sir and respected. Finally, to our parents without their support, it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

This section provides a list of abbreviations and their full forms to assist you in navigating the document and comprehending the technical terms and acronyms used.

$Bi - LSTM$ Bidirectional Long Short-Term Memory

DNN Deep Neural Network

GRU Gated Recurrent Units

$LSTM$ Long Short-Term Memory

$MFCCs$ Mel-Frequency Cepstral Coefficients

RNN Recurrent Neural Network

SER Speech Emotion Recognition

Chapter 1

Introduction

1.1 Introduction

Emotion identification is in high demand due to the unprecedented surge in the use of video conferencing platforms, which were widely utilized throughout the epidemic. Again, due to the drawbacks of video-based emotional recognition, such as privacy issues and the requirement for an unobstructed camera view, voice analysis has the potential to be a more versatile and covert substitute. The study explores the complexities of speech recognition algorithms, with an emphasis on identifying and evaluating auditory cues that correspond to different emotional states. BiLSTM, CNN, GRU, and LSTM are some practiced models that promise the future of speech emotion recognition. We examine the use of different TinyML models, focusing on those that are appropriate for the resource-constrained environment of wearable devices and computationally efficient given that recurrent neural networks are excellent at processing sequential data and temporal data like voice signals, there is an exploration of their potential, with a focus on the BiLSTM model. We further explore the integration of these TinyML models into Internet of Things frameworks, enabling real-time emotional state detection and communication. This entails prototyping with commonly available microcontrollers in Bangladesh, such as Arduino Uno, Raspberry Pi, ESP32, and STM32. An essential component of our research is a comparative analysis of different models and algorithms, evaluating their applicability, performance, and power consumption. This integrated system has several potential uses in the fields of healthcare, mental health, social interaction, and other areas. Our goal is to design a system that not only achieves high accuracy in emotion detection but also optimizes power usage, boosting the usefulness and sustainability of IoT wearable devices. To do so, we began by conducting a comparative analysis of various models and algorithms, assessing their performance based on metrics such as accuracy, F1 score, confusion matrix, precision, and recall. Subsequently, we identified the top-performing models and converted them into tinyML for real-time detection evaluation.

1.2 Motivation

We intend to work on voice-based emotion recognition with Tiny Machine Learning (TinyML) to create a supplemental tool that improves human-computer interactions, particularly in wearable IoT devices. The exponential expansion of voice assistants and the limits of video-based emotional identification highlights the need for a versatile and discreet alternative, such as voice analysis. We hope to develop systems that aid in real-time emotional recognition, which is critical for applications in healthcare, mental health, and social interaction, using TinyML's ability to execute efficient on-device processing. With institutional funding, we hope to increase accessibility through mobile apps and websites, raising awareness and contributing to the emerging field of emotion-sensitive AI technology.

1.3 Problem statement

The growing reliance on voice assistants, combined with the limits of video-based emotional identification, highlights the need for a more versatile and discrete technique of emotion detection. Current systems frequently encounter obstacles such as privacy concerns and the need for unobstructed visual inputs. Visual inputs adds an unnecessary dimension to the scopes of IoT, requiring two simultaneous inputs for speech emotion detection. The best approach to this issue , would be to directly detect emotions from speech using only the audio data and thus reducing the input dimensions. There is an urgent need for effective real-time emotion identification algorithms that can be effortlessly integrated into wearable IoT devices, and uses voice signals only to predict emotions. This study aims to overcome these issues by using Tiny Machine Learning (TinyML) to create effective, on-device voice-based emotion recognition systems. The purpose is to develop a supplementary tool that improves human-computer interactions, notably in healthcare, mental health, and social interaction applications, as well as to investigate approaches to increase accessibility via mobile apps and websites.

1.4 Research Objectives

Using effective TinyML algorithms, the primary goal of this research is to further the field of emotion detection in wearable IoT devices. To train and evaluate machine learning models, the project will entail gathering a variety of speech datasets, with an emphasis on precisely identifying various dialects and accents. The most successful models will be integrated into widely accessible microcontroller, and powerful algorithms that will significantly advance mental health, social interaction, and healthcare will be developed as a result of evaluating various machine learning classifiers and models in the context of TinyML and IoT.
1. **Analyze** the efficacy of current TinyML algorithms for voice signals on Internet of Things wearables in detecting emotions.

2.**Compare performance** metrics of various machine learning classifiers in the context of TinyML and IoT wearable devices.

3.**Integrate** the selected TinyML models into IoT frameworks, utilizing microcontrollers such as Arduino Uno, Raspberry Pi, ESP32, and STM32, which are readily available in Bangladesh,all the while considering a suitable framework for the microcontroller (TensorFlow Lite)

4. Rigorously **test and evaluate** the integrated system in terms of accuracy, computational complexity, and power consumption, aiming for high accuracy and optimized power usage.

5. Explore the **potential applications** of the integrated system in multiple practical fields like healthcare, mental wellness, and social interaction.

1.5 Research Methodology

This study uses a complete research methodology to create and assess Tiny Machine Learning (TinyML) models for real-time voice-based emotion recognition in wearable IoT devices. To ensure broad application, the methodology starts with collecting and preprocessing a diverse group of speech datasets, including different dialects and accents. Preprocessing methods include normalising audio signals, identifying key characteristics such as Mel-Frequency Cepstral Coefficients (MFCC), and enriching the dataset with techniques such as pitch change and time stretching to improve model resilience.

The study addresses choosing and practicing a variety of machine learning models and classifiers, including Bidirectional Long Short-Term Memory (BiLSTM) networks, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, K-nearest neighbors (KNN), Random Forest, Logistic Regression, and Gated Recurrent Unit. These models were chosen based on their shown usefulness in dealing with sequential and temporal data, such as voice signals, as well as their ability to operate within the processing restrictions of wearable devices. The training comprises supervised learning, which uses labeled datasets to teach models how to recognize and classify emotional states based on audio inputs. The performance of each model is optimized by hyperparameter tweaking and cross-validation.

Accuracy, precision, recall, and F1-score are some of the measures used to assess the performance of these TinyML models. Furthermore, confusion matrices and classification reports are provided to properly examine the models' effectiveness. The study also investigates the integration of these models into IoT frameworks, using Raspberry Pi 4B to demonstrate real-time emotional state detection. The comparative examination of these platforms takes into account elements such as computing complexity and power consumption to assure the viability of continuous operation in resource-constrained contexts. Our objective is to create an energy-effective and efficient emotion recognition tool for use in healthcare, mental health, and relationships. The research project examines TinyML models' resilience and flexibility in real-world scenarios. This includes evaluating their performance in varied environments, such as background noise and voice volume. The models are carefully evaluated with real-time speech inputs from users in various circumstances, including indoor, outdoor, and noisy environments, to ensure their dependability and accuracy in recognising emotions. Validating the models' effectiveness in real-world applications is crucial, as external factors can significantly affect the efficiency of voice-based emotion recognition systems.

An important component of this research is the examination of energy-effective methodologies and procedures tailored specifically for TinyML implementations. Given the restrictions of wearable IoT devices, which usually have a short battery life, optimizing power usage while preserving accuracy is crucial. The study investigates various approaches, such as low-power feature extraction algorithms and lightweight model designs, to achieve a balance between computational efficiency and model performance. Such efforts are targeted at increasing the battery life of wearable devices, hence improving user experience and the feasibility of continuous emotion monitoring.

The paper also discusses the ethical and privacy implications of adopting voice-based emotion identification technology. Ensuring user data privacy and securing informed consent are key aspects of the study technique. The research includes best practices for secure data processing, anonymization techniques, and regulatory compliance, including the General Data Protection Regulation (GDPR). It also emphasizes the necessity of creating transparent and explainable models that allow users to understand how their data is used and how the models arrive at specific emotional classifications. By putting ethical considerations first, the study hopes to foster trust and promote responsible usage of emotion identification algorithms in wearable IoT devices.

1.6 Research Problem

Emotion detection in voice signals has become an important area of research with numerous applications spanning from healthcare and mental well-being to customer service and human-computer interaction. The Internet of Things (IoT) has created numerous new opportunities for real-time, on-device emotion detection via wearable devices. However, the computational constraints of these devices are a significant challenge for deploying complex machine-learning algorithms. Tiny Machine Learning (TinyML) offers a promising solution by enabling machine learning tasks to be run on low-power, memory-constrained IoT devices.

"Furthermore, leveraging custom hardware accelerations can significantly boost the performance of TinyML algorithms, rendering them more adept for real-time applications. While these devices are rapidly incorporating advanced voice recognition capabilities, their potential for real-time emotion detection remains largely untapped. The primary challenge lies in developing efficient algorithms capable of operating within limited computational resources while upholding high accuracy in emotion detection. Lightweight architectures present a promising solution to striking this delicate balance. Therefore, the research objective is twofold: to assess existing TinyML algorithms for emotion detection in voice signals and to propose optimized algorithms that can be efficiently deployed on IoT wearable devices. Our overarching goal is to achieve real-time, precise emotion detection within the computational and power constraints of wearable technology, thereby fostering advancements in personalized services, mental health monitoring, and human-computer interaction (HCI)."

Chapter 2

Literature Review

Deep convolutional neural networks (CNNs) have been employed in paper [8] to extract features from audio data and categorize them into various emotion groups. To significantly enhance the classification performance, the paper additionally employs an ensemble of seven binary classifiers, each of which is tailored for a certain emotion category. Three distinct audio datasets were employed by the authors: IEMOCAP, EMO-DB, and RAVDESS. RAVDESS was selected because of its excellent accessibility and collection of audio and video recordings with twelve male and twelve female performers delivering English lines with eight distinct facial expressions. Researchers studying speech-based emotion detection frequently utilize EMO-DB, which has 535 audio utterances in German categorized into 7 emotion classes. Improvised data is used to create IEMOCAP.

The authors used these datasets to develop several incremental models for the classification of emotions. They only used speech samples representing the eight distinct emotion classes—sad, joyful, furious, calm, afraid, surprised, neutral, and disgusted—from the RAVDESS database. The datasets were integrated into the suggested framework, which begins with feature extraction and then applies the baseline deep learning model. The outcomes demonstrate that the proposed framework performs better in terms of accuracy and generalization than earlier state-of-the-art techniques.

Using deep learning approaches, Paper [10] conducts an extensive literature review on speech emotion recognition (SER). The authors have carried out an extensive review of the literature on this subject, covering the databases utilized, the emotions retrieved, the advancements made in speech emotion identification, and any associated restrictions. Additionally, they covered the various deep learning methods for SER, including deep belief networks (DBNs), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), and summarized the research based on these methods. The study has demonstrated that recurrent architectures like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) are significantly more effective in speech-based classification, even if DNNs and CNNs offer state-of-the-art results for video feeds and image processing. Likewise, Paper [1] provides an additional thorough analysis of the current state-of-the-art SER. A thorough summary of the numerous SER emotional models that are now in use, such as dimensional models, category models, and hybrid models, is also given by this particular study. Not only that, but the research also delves into the use and usefulness of a number of databases—like the Interactive Emotional Dyadic Motion Capture Database, the Emotional Prosody Speech and Transcripts, and the Berlin Emotional Speech Database—that are used to train SER systems. These case-by-case reviews help us effectively choose which algorithms and classification approaches would be best suited for our research works. A two-stage training process is suggested in Paper [7] for the semi-convolutional Neural Network (semi-CNN), which is the basis for learning affect-salient features for Speech Emotion Recognition (SER). Initially, contractive convolutional neural networks are used to process unlabeled samples to identify potential features. Subsequently, the Semi-CNN receives these features and employs a novel objective function that emphasizes feature saliency, orthogonality, and discrimination to train affect-salient, discriminative features. The suggested approach takes a spectrogram of the speech sound as input and comprises an input layer, one convolutional layer, one fully connected layer, and an SVM classifier. As the semi-CNN learns features at each layer, the features remain affect-salience concerning the SER aim, but they grow more and more invariant to nuisance factors. The learnt features beat other well-known feature representations regarding speaker variance and obtain higher accuracy in emotion classification, according to the evaluation of this approach on four benchmark datasets. A novel framework called Radial Based Function Network (RBFN) employing crucial sequencing segment selection is presented in this paper [15]. The STFT algorithm is used to convert speech sequences into spectrograms, which are then examined by a CNN model to extract important features. After normalizing these CNN features, they are input into a deep BiLSTM, which uses comparable datasets that we previously covered to focus on important segments and reduce computational complexity while improving the detection of spatiotemporal information. This allows for the temporal information to be captured for correct emotion recognition. However, Paper[11] places a strong emphasis on audio or voice signal preprocessing. using the Global Feature Algorithm to eliminate redundancy after extracting features using the MFCC, DWT, pitch, energy, and ZCR algorithms. Ultimately, they discovered the universal emotions of happiness, anger, neutrality, and sadness utilizing well-known machine learning techniques, including SVM, Decision Tree, LDA, RBF, KNN, ANN, and GMM.

A frame-by-frame method for processing voice utilizing deep learning algorithms and basic speech processing was presented in Paper [5]. It was argued that frame-based processing is preferable than turn-based processing. Their method also performed best when used with the IEMOCAP dataset. This frame-based approach uses Fourier-transform-based filter bank audio spectrograms and a deep multi-layered neural network to estimate emotion class probabilities for each frame in the input utterance. Paper [4], which reviews the usefulness of voice assistants, is entirely different from the others. This highlights the necessity for emotion recognition to provide voice assistance, a new generation of technology, by demonstrating how useful VAs can be for a variety of everyday chores. The review research was presented in the publication using the ISO 9241-11 framework as the measuring tool. Also covered were various classifier models, including CNN, SVM, KNN, and CapsNet.

2.1 IoT Devices and Sensors (Arduino UNO/NANO, TinyML board)

The articles [17], [2], and [19] are all concerned with directly integrating SER algorithms on Internet of Things (IoT) devices (Arduino-UNO, NANO), but using different databases and methodologies. A model that divides speech characteristics into three emotional states—positive, negative, and neutral—was put out by the authors of paper [17]. The model is constructed using a Convolutional Neural Network (CNN). They trained their model using the RAVDESS dataset, extracted features using a Mel-frequency cepstral coefficient (MFCC), and developed the model using TensorFlow Lite. The Arduino Nano 33 BLE Sense, which has several sensors and Bluetooth for Low Energy connectivity, is the hardware used for the study. Conversely, the goal of paper [2] is to reduce the high expenses and computational complexity associated with conventional Edge AI machine learning approach implementations. The same IoT as in paper [17] was used (Arduino Nano 33 BLE), but the edge device and cloud were given different jobs to complete: motion detection was handled by the edge device, and emotion recognition was handled by the cloud. With relatively little latency and resource consumption throughout testing, the model has attained a high accuracy rate.

Thus, this study has succeeded in developing an efficient and economical way to incorporate real-time emotion identification in Edge AI systems, which is crucial for any TinyML project. An intriguing new stacking-based ensemble TinyML framework has been introduced in Paper [19] to facilitate cooperative decision-making between edge nodes and IoT devices. In this instance, a system-wide choice is made by combining the individual decisions made by end devices at the edge level. The study uses an Arduino Uno device with LoRa-powered connectivity to show the viability of the technique with a focus on a smart-agriculture use-case scenario. They have also employed their own bespoke datasets with over 10k samples and run MLP, DT, and RF classifiers on them. The Random Forest (RF) and Decision Tree (DT) algorithms demonstrated remarkably good accuracy, energy consumption, latency, and memory usage. This research, which combines edge computing and TinyML, is a step towards the implementation of useful hierarchical intelligent IoT systems. On the other hand, when the model was translated using TFLite, Paper [21] proposed a model for detecting and analysing emotions utilizing tinyML technology on a TinyML board as the IoT device. TensorFlow was utilised to build CNN and CNN-LSTM, two models that were used in this study to classify emotions (8 categories). For analysis, characteristics such as zero crossing rate, root mean square energy, and

Mel Frequency Cepstral Coefficients (MFCC) were taken out of audio signals. Twenty thousand audio recordings gathered from the RAVDESS and CREMA-D datasets are used to train the models.

In addition, the audio files were altered by stretching, shifting, and introducing noise.Even while the paper's findings might not be seen as "impressive" in terms of generalization to previously unpublished data, more model validation and improvement could produce far more reliable and accurate outcomes. In order to better identify emotions, Paper [23] suggested utilizing the Ardiuno Nano 33-BLE for voice and gesture detection with the TinyML model. This provided a new level of complexity. The hand gesture recognition system can identify movements of the human hand, and the speech recognition system can control the onboard RGB LED based on spoken keywords. However, they did not present any datasets for the proposal. They employed the EdgeImpulse framework for model training and deployment with an easy approach. In the end, Paper [18] proposed a classifier algorithm using a combined dataset of two well-known speech recognition datasets, SAVEE and TESS. They used a matrix to use 70% of the data as a train set and 30% as a test set. Their proposed algorithm was then compared to existing machine learning models, such as MNB, NB, INB, SVM, DT, and RF, and it was discovered to have a high accuracy of 15.76%, a higher specificity of 20.69%, greater sensitivity of 15.59%, better precision of 12.62%, and a 30.65% improved f1-score. It gives our findings a fresh perspective. In Paper[6], multi-class and hierarchical Support Vector Machines (SVMs) for emotion recognition are investigated using the EMO-DB, DES, and Serbian emotional speech datasets. varied moods are classified with varied accuracy percentages in different datasets; the Serbian emotional speech database has remarkably high accuracy percentages. The paper's major focus is on speech-based emotion classification using a multi-class SVM with a hybrid kernel and thresholding fusion [25]. The hybrid kernel selection, which includes linear, quadratic, polynomial, MLP, and RBF kernels, produces notable accuracy percentages for a range of emotions. The recommended strategy outperforms the reference in terms of classifier-level accuracy and decision-level recall, especially in speaker-independent emotion classification.

Paper [20], which investigates the application of various audio properties and machine learning techniques for speech-to-emotion detection. Support Vector Machines (SVM), random forest classifiers, LSTM, and CNN are utilised, displaying varying classification results on datasets such as Ravdess, Save, Tess, and Crema-D. A deep learning technique for pattern identification and feature selection for voice emotion recognition is provided in Paper [9], using CNN and ResNet34 in particular. The study's investigations, which displayed different accuracy percentages for different feature sets, were conducted using the Berlin database. Paper [12] examines effective voice emotion identification with enhanced feature extraction, using Fractional Fourier transform on EMO-DB, SAVEE, and PDREC datasets. The proposed strategy achieves high accuracy percentages in the dataset classification.

The paper[26] describes a voice emotion identification system that integrates HSF-DNN, MS-CNN, and LLD-RNN, three multi-task learning-based classifiers. Using the interactive emotional dyadic motion capture dataset (IEMOCAP), the study compares the weighted and unweighted accuracies of each classifier. Paper [3] focuses on the use of Gaussian mixture models (GMM) to recognise speaker variability in emotions. When tested on the German FAU Aibo Emotion Corpus and the English LDC Emotional Prosody speech corpus, the model achieves a combined classification accuracy of 70.4%. Paper [24] investigates speech emotion categorization with an attention-based LSTM, demonstrating improvements on the CASIA, eNTERFACE, and GEMEP dataset. The suggested approach improves recognition accuracy by 5.4%, 33.8%, and 17.0%, respectively, indicating its ability to capture the speech waveform's inherent temporal correlations. Paper [16], focuses on real-time speech emotion recognition implemented using LSTM and Raspberry Pi, and speech signal spectral analysis is used to identify depression . The procedure entails taking features out of speech signals, creating an LSTM model to identify emotions, and testing the model with real-time voice signals that are gathered and processed by Raspberry Pi. The spectral analysis carried out using MATLAB is covered in the paper, along with the datasets used, including the RAVDESS and DAIC-WOZ databases. The accuracy of the LSTM model for speech emotion recognition has been reported to be 86%. Paper [14] used a Raspberry Pi 3 with a Digital Signal Processor (DSP) model to identify emotions in speech signals. In order to ensure efficiency and clarity, the system recorded speech with varying emotions, analyzed these signals using Python on the Raspberry Pi, and then compared the outcomes with those processed in MATLAB. The findings revealed that the Raspberry Pi 3 could recognise emotions with an accuracy of 95% and 85% in MATLAB.

Table 2.1 provides an in-depth summary of the reviewed paper, specifically including references, specific tasks addressed, types of IoTs used, classifiers applied, databases utilized, and the achieved accuracy, offering a detailed understanding of the research findings.

Ref.	**Task**	**IoTs**	**Classifier**	**Database**	**Accuracy**
[8]	DNN for Speech Emotion Recognition.	N/A	CNN, Ensemble of Seven Binary Classifiers	RAVDES, EMO-DB, IEMOCAP	RAVDESS- 71.61% EMO-DB- 86.1% EMO-DB- 95.71% IEMOCAP- 64.3%
[10]	Deep Learning Techniques for Speech Emotion Recognition	N/A	DBMs, DBNs, CNNs, RNNs, RvNNs, AE	IEMOCAP, Emo-DB, SAVEE	DCNN- Higher Accuracies Compared to Traditional Techniques
[7]	Learn Affect-Salient Features for SER Using Semi-CNN	N/A	SVM	SAVEE, Emo-DB DES MES	Semi-CNN Excels in Complex Scenes Outperforming Other SER Features
[1]	Survey on SER	N/A	BPNN, DES BLR, MLP, HMM, ANN, Naive Bayes	SAVEE,and Multilingual Databases	SAVEE - 46.25% EMA - 61.65% LDC - 43.18%
[15]	Clustering-Based Speech Emotion Recognition by Deep BiLSTM	N/A	CNNs, SVMs, Random Forests, MLP, Softmax , Adam, BiLSTM	IEMOCAP, EMO-DB, RAVDESS,	IEMOCAP- 72.25% EMO-DB- 85.50% RAVDESS- 77.02%
[11]	Feature Extraction Algorithms to Improve Speech Emotion Recognition	N/A	SVM, Decision Tree, LDA, RBF, KNN, ANN, GMM	EMO-DB, RAVDESS	SVM- 77% RBF- 82% MLP- 78% KNN- 64% HMM- 76.12% GMM- 78.77% ANN-51.19%
[5]	Evaluating Deep Learning Architectures for Speech Emotion Recognition	N/A	DNNs, ConvNets, RNNs, LSTM	IEMOCAP	N/A
[4]	A Systematic Review of Voice Assistant Usability	N/A	CNN, SVM, KNN, CapsNet,	N/A	N/A
[17]	Implementing Real-Time SER on Embedded Systems.	Arduino Nano 33 BLE Sense	Convolutional Neural Network (CNN)	RAVDESS	N/A

[2]	Continuous Emotion Recognition on a Small-Scale Edge Device	Arduino Nano 33 BLE sense	Deep Learning Algorithms	AffectNet	86.7%
[19]	Intelligent and Efficient IoT Using TinyML and Edge Computing	Arduino Uno	Multi-Layer Perceptron (MLP), Decision Tree (DT)	Custom Dataset (10,000 Samples)	Up to 99.9% (DT and RF)
[21]	Automatic Emotion Recognition System Using TinyML	TinyML Board	CNN Model, CNN-LSTM Model	RAVDES, CREMA-D	CNN = 67%,CNN-LSTM=72%
[23]	Implementation of TinyML Models on Ardiuno 33-BLE for Gesture and Speech Recognition	Arduino Nano 33 BLE	NN, SVM, K-NN, LDA, HMMs	N/A	N/A
[18]	Automatic Speech Emotion Detection	N/A	Hybrid of Gray Wolf Optimizer, Naive Bayes	SAVEE TESS	N/A
[6]	Multi-Class and Hierarchical SVMs for Emotion Recognition	N/A	SVM	EMO-DB, DES, SESD	Angry- 64% Neutral- 58% Sad-72% Happy- 72%
[25]	SPEECH-BASED EMOTION CLASSIFICATION	N/A	SVM	LDC	Anger- 76.9% Sadness- 95.4% Disgust- 98.7% Neutral- 100% Happiness- 70.5% Fear- 73.0%
[20]	Emotion Recognition From Speech	N/A	SVM Random forest LSTM CNN	Ravdess Save Tess Crema-D	SVM- 0.68 Random Forest- 0.63 LSTM-0.71 CNN- 0.74
[9]	Pattern Recognition and Features Selection For SER	N/A	CNN ResNet34	Berlin	MFCC- 94.21% Prosodic Feature-83.54%, LSP Features- 83.65% LPC Features-78.13%

[12]	Efficient Speech Emotion Recognition Using Modified Feature Extraction	N/A	Fractional Fourier Transform	EMO-DB SAVEE PDREC	EMO-DB 97.57% SAVEE 80% PDREC 91.46%
[26]	Speech Emotion Recognition Using Fusion of Three Multi-Task Classifiers: HSF-DNN MS-CNN LLD-RNN	N/A	HSF-DNN MS-CNN LLD-RNN	IEMOCAP	W/A- 54.4% U/A- 55.6%
[3]	SPEAKER VARIABILITY IN EMOTION RECOGNITION	N/A	GMM	LDC-English German	70.4%
[24]	Speech Emotion Classification Using Attention	N/A	LSTM	CASIA eNTERFACE GEMEP	Improved 5.4%, 33.8% 17.0%
[16]	Real Time Speech Emotion Recognition Using Raspberry Pi	N/A	LSTM	DAIC_WOZ RAVDESS	86%
[14]	Real Time Emotion Detection From Speech Using Raspberry Pi 3	N/A	SVM HMM GMM ANN	IEMOCAP RAVDESS	85% in MATLAB 95% on the Arduino Board

Table 2.1: Comparative Analysis of Key Findings from Selected Research Studies

Despite significant advancements in speech emotion recognition (SER) through various deep learning methodologies and extensive use of comprehensive datasets, several gaps remain in the current literature, especially concerning the integration of SER algorithms with TinyML on IoT wearable devices. While studies have demonstrated the efficacy of CNNs, LSTMs, and hybrid models in extracting and classifying emotional features from speech data, there is a limited exploration of how these models can be effectively adapted and optimized for the constrained computational environments of wearable IoT devices. All the researches has largely focused on standalone systems or cloud-based implementations, overlooking the challenges and opportunities of edge computing in real-time emotion detection. Moreover, existing works that incorporate TinyML, such as those using Arduino platforms, have primarily addressed basic classification tasks with limited emotion categories and have not fully leveraged advanced deep learning techniques for enhanced accuracy and efficiency. There is also a paucity of research on the deployment and validation of these models in practical, real-world scenarios, which is crucial for understanding their performance in dynamic and noisy environments typical of wearable applications. This thesis aims to bridge these gaps by proposing algorithms specifically transformed for IoT wearable devices, focusing on optimizing deep learning models for low-power, real-time emotion recognition and validating their effectiveness in practical use cases.

Chapter 3

Data Sets

3.1 Descriptions

In this section, we delve into the process of dataset collection, essential for training models and conducting accuracy comparisons. To enhance the robustness of our outcomes, we have carefully chosen a composite of well-established secondary datasets.

3.1.1 Crowd-sourced Emotional Multi modal Actors Dataset (CREMA-D)

The CREMA-D dataset is notable for its extensive diversity, including 7,442 original footage from 91 performers of diverse ages, genders, colors, and ethnicities, including African American, Asian, Caucasian, Hispanic, and unspecified backgrounds. The dataset, which includes contributions from 48 male and 43 female performers ranging in age from 20 to 74, provides a rich tapestry of vocal performances ideal for training robust and generalized emotion classification models. Each actor recorded lines expressing six different emotions: anger, disgust, fear, happiness, neutrality, and sadness, with four levels of emotional intensity: low, medium, high, and unspecified. The diversified amount of data ensures that models trained on CREMA-D can effectively avoid overfitting and succeed at generalization tasks across a variety of datasets, making it a priceless asset for academics and professionals in emotional computing and emotion recognition from audio.

3.1.2 Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)

The RAVDESS dataset comprises recordings from 24 skilled actors, evenly split between genders, who deliver two sets of phrases matched for content in a neutral North American accent. These phrases convey a range of emotions including calmness, happiness, sadness, anger, fear, surprise, and disgust. Each emotional state is depicted at two levels of intensity – normal and strong – with an additional neutral expression included for comparison. This dataset serves as a valuable resource for researchers studying emotional expression in speech and related fields such as effective computing and natural language processing. Its comprehensive coverage allows for detailed analysis and modeling of various emotional states across different intensity levels, contributing to a deeper understanding of human communication and interaction.

3.1.3 Surrey Audio-Visual Expressed Emotion (SAVEE)

The SAVEE dataset features high-quality audio recordings from four male native English speakers affiliated with the University of Surrey. Each speaker provided utterances across six primary emotions: anger, disgust, fear, happiness, sadness, and surprise, along with a neutral category. The dataset includes 15 TIMIT sentences per emotion, comprising three common, two emotion-specific, and ten generic sentences, all phonetically balanced. Additionally, each emotion-specific sentence was recorded neutrally, resulting in a total of 120 utterances per speaker. While the dataset offers valuable insights into male emotional expression, its gender imbalance suggests a need to supplement it with datasets featuring female speakers for a more comprehensive emotion classifier.

3.1.4 Toronto Emotional Speech Set (TESS)

The TESS dataset, renowned for its high-quality audio recordings, exclusively features female speakers, offering a valuable counterbalance to the predominantly male-focused datasets in emotion classification research. With a total of 2800 audio files, each containing one of 200 target words, the dataset captures performances by two actresses across seven distinct emotions: anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral. Organized by actress and emotion, the dataset's structure facilitates easy access to the audio files in WAV format. This comprehensive and meticulously curated dataset provides an excellent resource for training emotion classifiers, enhancing generalization capabilities, and mitigating the risk of overfitting.

Each dataset offers unique strengths and characteristics that contribute to the robustness and generalization of the classifier. To ensure best possible speed and functionality for our research, all 7,442 sound clips from 91 different performers in the Crema-D dataset have been used. However to be more precise, we have reduced the number of files in Tess to 2,800, Savee to 480, and Ravdess to 1,440. These datasets contain unprocessed data that is necessary for our purpose of classifying emotions.

3.2 Data pre-processing

In the data pre-processing phase, we streamlined our approach by consolidating the diverse 12162 audio clips into a single cohesive data frame. This allowed us to efficiently process and train our models. By meticulously defining file directory paths and sourcing locations for the audio files, we ensured precise data organization. We then categorized the data based on gender attributes, enabling gender-specific analysis and insights to be gleaned from the dataset. Additionally, we labeled the data according to emotion categories, laying the foundation for emotion-specific analyses and model training.

Conducting exploratory data analysis (EDA) was crucial to identify any potential imbalances, particularly focusing on the male-to-female ratio. This assessment provided valuable insights into the distribution of emotions and genders within the dataset. To prepare the primary data representation of the audio data, we introduced an array of audio samples from our dataset. Then we performed a comparison study of the audio ratios of male and female voices.

Figure 3.1 depicts the ratio between male to female ratio.

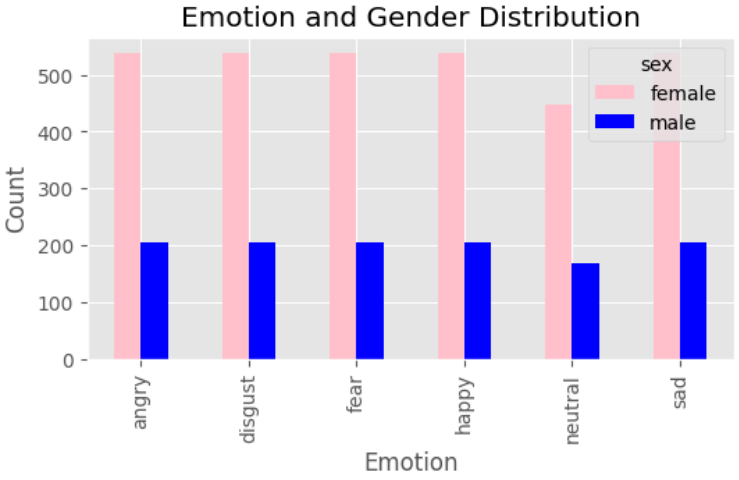


Figure 3.1: Male-to-female ratio

After analyzing the data, we decided on the female ratio since it met our evaluation criteria better and showed better performance, and the result can be seen in figure 3.2

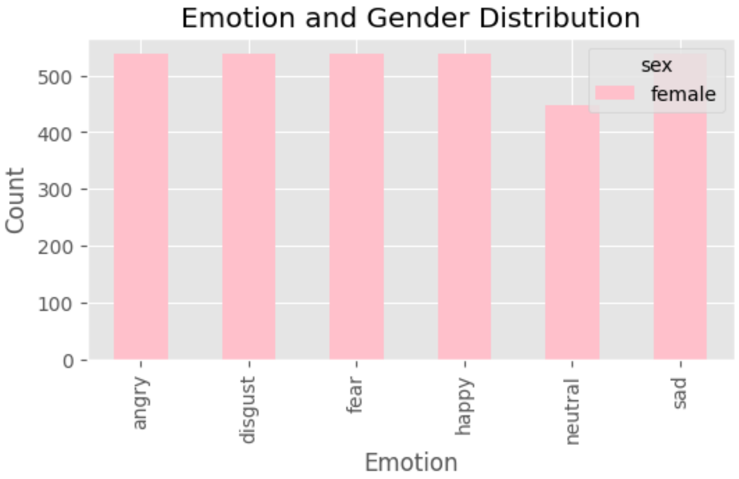


Figure 3.2: Female ratio

Furthermore, a trimming process was implemented to remove unnecessary silence intervals, enhancing the quality and relevance of the audio data, thus can be seen in figure 3.3.

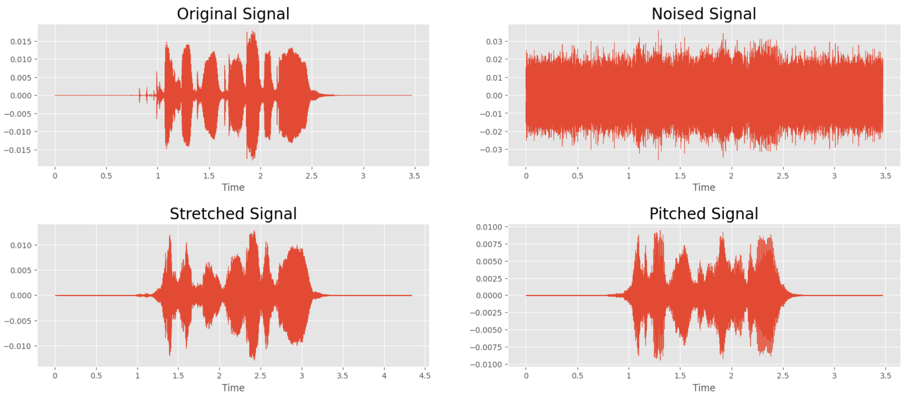


Figure 3.3: Audio Transformation

After that we can see Figure 3.4 of the spectrogram, illustrating the frequency distribution over time.

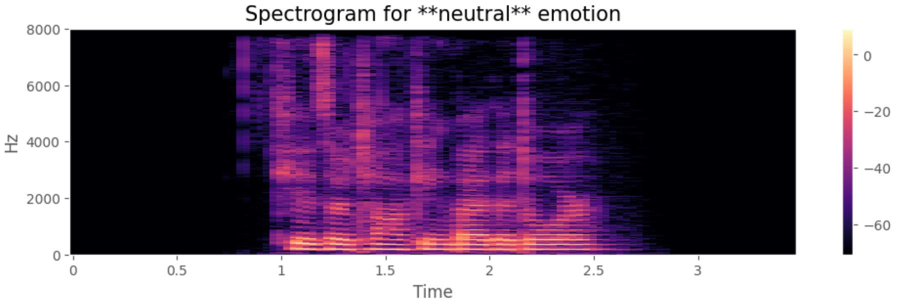


Figure 3.4: Spectogram

Ensuring uniformity and compatibility within the dataset was essential, so we applied a padding mechanism to guarantee that all audio samples obtained a consistent length. This standardized format facilitated seamless processing and analysis across the dataset. Additionally, we created a dictionary for mapping emotion labels to numerical values, along with a function to encode emotion labels into their numerical representations. This encoding mechanism provided a standardized framework for handling emotion data within our models.

3.3 Features Extraction

”Pyaudio”, a python library is widely used to perform a wide range of audio analysis tasks, and thus we selected it for extracting the most notable features from voice signals . Then we identified three main audio features of the human voice and given priority to those during the feature extraction stage: Root Mean Square Energy (RMSE), Zero Crossing Rate, and Mel-Frequency Cepstral Coefficients (MFCCs). In this, MFCCs are essential for recording an audio source's spectral envelope, and also for gaining important knowledge about its frequency distribution. On the otherside, the audio input is converted into a Mel-frequency scale in this multi-step procedure, and then the discrete cosine transform is used to produce coefficients.

Furthermore, we used the Zero Crossing Rate function which measures the speed at which a signal changes, providing important details regarding the fluctuation patterns of the audio stream. This feature helps us to explain abrupt shifts in the audio waveform, which is useful for analyzing the speech.

Furthermore, Root Mean Square Energy (RMSE) captures the audio signal's root mean square amplitude, offering insights into its total energy content. By squaring the amplitude values, RMSE provides a comprehensive understanding of the signal's strength and intensity, aiding in the analysis of its loudness characteristics.

These extracted features serve as vital descriptors of the audio signals, facilitating easier analysis and model training in subsequent stages. Their inclusion ensures that our models can effectively capture the unique characteristics of the audio data, thereby enhancing their performance and accuracy in emotion classification tasks.

3.4 Splitting Data

After completing feature extraction, we proceeded to split the data into two distinct sets: an intermediate set and a training set. The training set comprised 88% of the data, providing a substantial portion for model training, while the remaining 12% constituted the intermediate set, allowing for initial evaluation and refinement. Also segmentation was performed on the intermediate set, dividing it into testing and validation subsets. We allocated 30% of the data from the testing set and 70% from the intermediate set for validation purposes. This meticulous division facilitated a comprehensive assessment of model performance, ensuring robustness and reliability in subsequent analyses.

By structuring the data in this manner, we were able to systematically organize the labels into layers and convert them into a categorical format, enhancing the interpretability and accuracy of our results. This rigorous approach to data splitting laid a solid foundation for the development and evaluation of our classification models.

Chapter 4

Methodology

4.1 Proposed methodology

4.1.1 LSTM Model

Recurrent neural network architectures, such as Long Short-Term Memory, are excellent for processing and forecasting sequential data. Long-term dependence are captured with effectiveness. To detect emotions, this model is trained using a labelled dataset of inputs tagged with corresponding emotions. It acquires the capacity to identify input connections and patterns linked to specific emotional states. Because LSTMs can manage and maintain long-term dependencies in data, such as emotion recognition, they are very useful tools. This capacity results in high accuracy and efficient performance in applications where context and sequential information are important [5].

$$f_t = \sigma(\mathrm{W}_f \cdot [\mathrm{h}_{t-1}, \mathrm{x}_t] + \mathrm{b}_f) \tag{4.1}$$

$$i_t = \sigma(\mathrm{W}_i \cdot [\mathrm{h}_{t-1}, \mathrm{x}_t] + \mathrm{b}_i) \tag{4.2}$$

$$\tilde{C}_t = \tanh(\mathrm{W}_C \cdot [\mathrm{h}_{t-1}, \mathrm{x}_t] + \mathrm{b}_C) \tag{4.3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4.4}$$

$$o_t = \sigma(\mathrm{W}_o \cdot [\mathrm{h}_{t-1}, \mathrm{x}_t] + \mathrm{b}_o) \tag{4.5}$$

$$\mathrm{h}_t = o_t * \tanh(C_t) \tag{4.6}$$

Figure 4.1 shows the detailed model framework for LSTM, highlighting the intricate architecture [5].

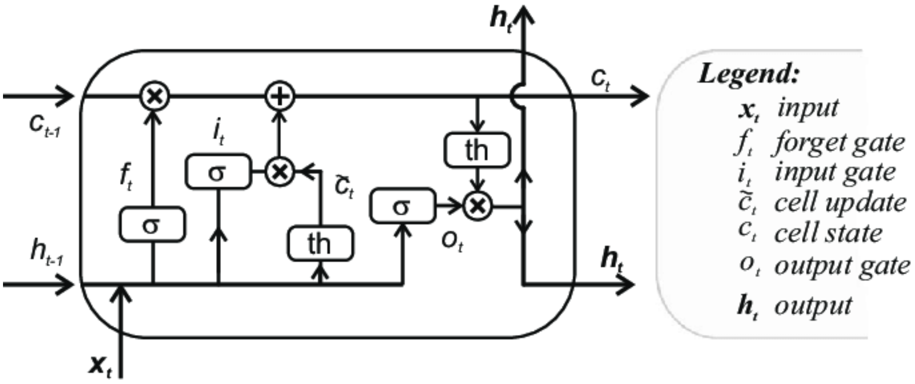


Figure 4.1: Detailed Model Framework for LSTM

4.1.2 Convolutional Neural Network(CNN)

CNNs are like Long Short-Term Memory (LSTM) models, excel at processing sequential data with long-term dependencies. They also do well in picture classification, object detection, and image segmentation. CNNs use a unique architecture that includes convolutional layers, pooling layers, and fully connected layers to extract hierarchical representations from input images, distinguishing nuanced features ranging from edges to complex object shapes and textures. CNNs are trained on labeled datasets correlating inputs with matching emotions or categories. They develop the ability to recognize input patterns and connections connected to specific emotional states or object categories. This capability, combined with their ability to manage long-term dependencies within data, establishes CNNs as reliable instruments for achieving high accuracy and efficient performance in tasks requiring contextual and sequential information, ushering in advancements across a wide range of domains, including emotion recognition, object detection, and image classification. Our methodology takes advantage of CNNs' intrinsic capabilities to improve picture data analysis and interpretation. CNNs excel at capturing subtle characteristics required for effective classification and segmentation tasks by meticulous training on labeled datasets. They also distinguish input connections and patterns associated with distinct emotional states or object categories. CNNs have the potential to achieve higher accuracy and efficiency in applications that need complex contextual knowledge and sequential information processing because of their ability to manage and sustain long-term dependencies within data. Leveraging CNNs in our proposed methodology demonstrates our dedication to pushing the boundaries of image analysis, supporting advances in a variety of sectors where exact recognition and understanding of visual data are critical [17].

Convolution layer:

$$h_{ij} = \sigma\left(\mathrm{b} + \sum_{m}\sum_{n} I_{(i+m)(j+n)}K_{mn}\right) \tag{4.7}$$

Figure 4.2 shows the detailed model framework for Convolutional Neural Networks (CNN), the layers of convolution, activation functions, pooling, and fully connected layers that enable feature extraction and classification [17].

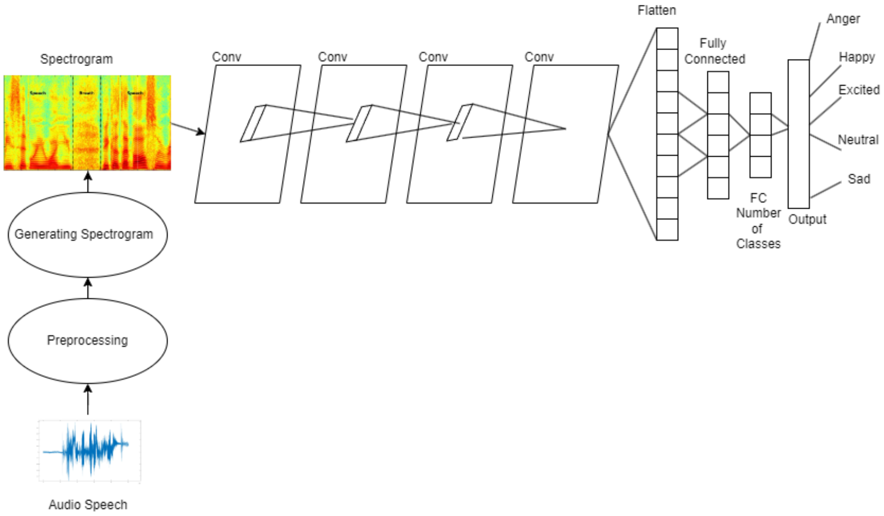


Figure 4.2: Detailed Model Framework for CNN

4.1.3 Gated Recurrent Unit (GRU)

In our suggested methodology, the Gated Recurrent Unit (GRU) is a reliable solution for processing sequential data with long-term dependencies. GRU is a recurrent neural network (RNN) architecture that successfully captures temporal patterns in sequential input. Unlike standard RNNs, GRU uses gating methods to control the flow of information, such as an update and reset gate. These gates allow the GRU model to selectively update its internal state representation and retain important information across time, which aids in the capture of long-term dependencies. In our methodology, GRU models are trained utilizing common procedures such as backpropagation through time (BPTT) and gradient descent optimization. The training procedure comprises feeding sequential data into the GRU model, which then learns and adapts its internal state representation iteratively. By using GRU's gating features, the model effectively captures and utilizes long-term dependencies in the data. We tailor the GRU model's architecture and parameters to optimize its performance for our specific task, whether it's time series forecasting, natural language processing, or emotion recognition, using meticulous hyperparameter tuning and optimization techniques like grid search and random search. By including GRU models in our suggested methodology, we hope to improve sequential data analysis, allowing for breakthroughs in a variety of disciplines where contextual knowledge and temporal information processing are critical [13].

$$z_t = \sigma(\mathrm{W}_{xz}\mathrm{x}_t + \mathrm{W}_{hz}\mathrm{h}_{t-1} + \mathrm{b}_z) \tag{4.8}$$

$$r_t = \sigma(\mathrm{W}_{xr}\mathrm{x}_t + \mathrm{W}_{hr}\mathrm{h}_{t-1} + \mathrm{b}_r) \tag{4.9}$$

$$\tilde{h}_t = \tanh(\mathrm{W}_{xh}\mathrm{x}_t + \mathrm{W}_{hh}(r_t \odot \mathrm{h}_{t-1}) + \mathrm{b}_h) \tag{4.10}$$

$$\mathrm{h}_t = (1 - z_t) \odot \mathrm{h}_{t-1} + z_t \odot \tilde{h}_t \tag{4.11}$$

Figure 4.3 illustrates the GRU model framework, showing how the update and reset gates control information flow and memory in the network [13].

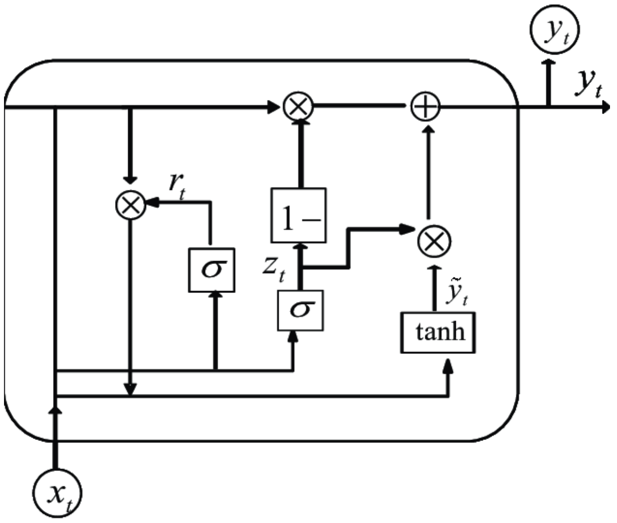


Figure 4.3: Detailed Model Framework for GRU

4.1.4 Bidirectional LSTM (BiLSTM)

To enhance our model's capacity in terms of LSTM for identifying correlations and connections, we utilized a Bidirectional Long Short-Term Memory (BiLSTM) network. This approach allowed us to extract contextual information from both the preceding and following states in the sequence data. The BiLSTM architecture is formed with two LSTM layers: one processes the sequence from beginning to end (forward direction), while the other processes the sequence from end to beginning (reverse direction). The model is able to analyse the preceding and subsequent context at every point in the sequence. In this case, the input data was preprocessed, including feature normalisation, missing value management, and noise removal, to ensure quality and consistency. The optimizer was utilised to effectively update the model parameters after the BiLSTM model was trained using a loss function appropriate for our goals, such as Mean Squared Error (MSE) for regression or Cross-Entropy Loss for classification. Metrics that we worked with including accuracy, precision, recall, and F1-score for classification tasks and Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE) for regression were employed to evaluate the performance of the BiLSTM model. Hyperparameter tweaking was used to adjust the model's performance, changing variables including the number of LSTM units, learning rate, batch size, and epoch count. We employed both grid search and random search to find the best hyperparameter configuration. When it came to collecting long-term dependencies and taking advantage of context in both directions, the BiLSTM model fared better than baseline techniques like unidirectional LSTM and other RNN variants [15].

The detailed BiLSTM model framework is shown in Figure 4.4, showing how the bidirectional structure processes information from past and future sequences to enhance context understanding [15].

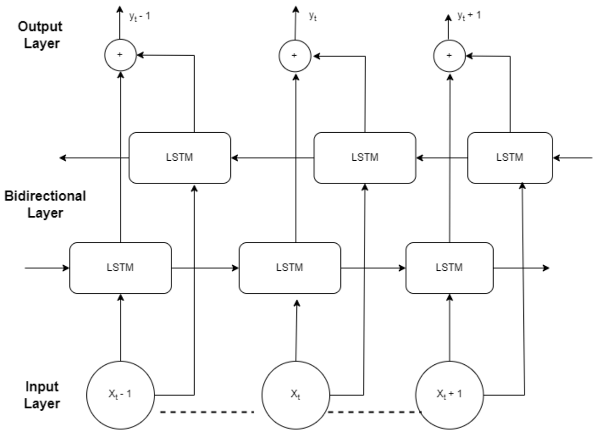


Figure 4.4: Detailed Model Framework for Bidirectional LSTM

4.1.5 Random Forest

Our research paper introduces Random Forest as a flexible ensemble learning algorithm for regression and classification applications in the methodology section. A type of supervised learning method called Random Forest creates a lot of decision trees during training and outputs the mean prediction (regression) or the mode of the classes (classification) for each tree. The Random Forest enhances the diversity and resilience of the ensemble model by building each decision tree using a random subset of attributes and data points. This collective approach finds complex patterns in the data and minimizes overfitting by averaging the predictions of several trees. Moreover, Random Forest offers information on the importance of features, which makes it possible to identify important predictors that have an impact on the result variable. In our study process, Random Forest is a useful categorization approach, particularly when there is a complex or nonlinear relationship between inputs and outcomes. By utilizing the combined knowledge of numerous decision trees, Random Forest generates forecasts that are trustworthy and strong, effectively capturing intricate patterns and correlations within the data. The random forest model is also flexible enough to handle categorical variables and missing data, which makes it appropriate for a variety of real-world situations. We seek to make use of Random Forest's capacity for achieving high accuracy and generalization performance through extensive testing and assessment, providing essential insights into the predictive elements that impact our research objectives[15].

Figure 4.5 explores the detailed model framework for Random Forest, a versatile and robust ensemble learning method. This framework illustrates how Random Forest combines multiple decision trees to make predictions, offering a powerful tool for classification and regression tasks across various domains [15].

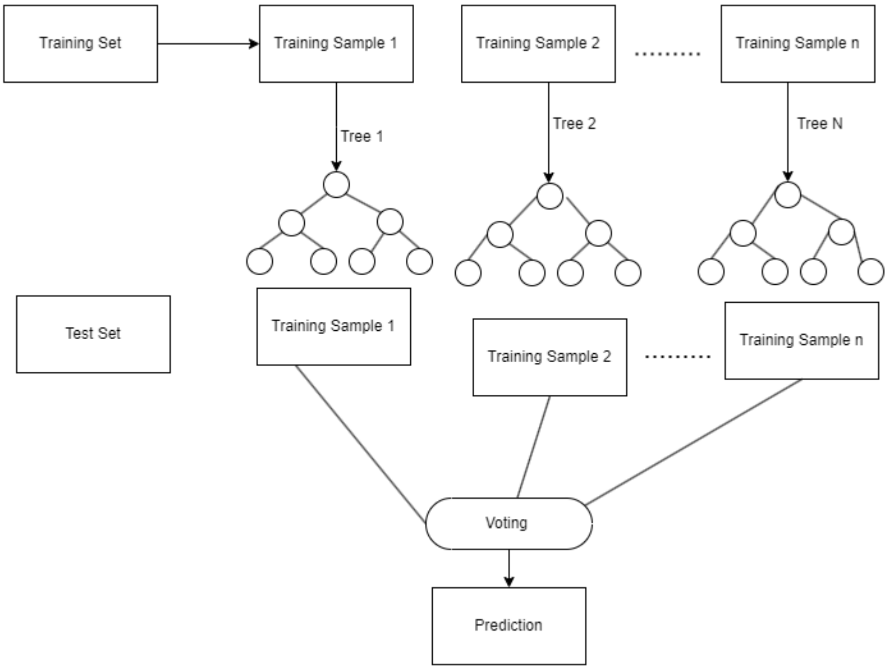


Figure 4.5: Detailed Model Framework for Random Forest

4.1.6 Logistic Regression

Logistic regression is a basic statistical method used for binary classification tasks, where we want to predict one of two possible outcomes. Unlike linear regression, which predicts continuous values, logistic regression estimates the probability that an observation belongs to one of two groups based on one or more predictor variables. This makes it especially useful when the outcome is categorical and the relationships between predictors and the outcome are not linear. In our study, we use logistic regression to understand the relationship between predictor variables and binary outcomes[22].

By calculating the coefficients for each predictor, logistic regression helps us see how these variables influence the probability of the outcome, making it easier to interpret and predict results in various situations.

Given features $(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_n)$

weights $(\mathrm{W}_1, \mathrm{W}_2, \ldots, \mathrm{W}_n)$

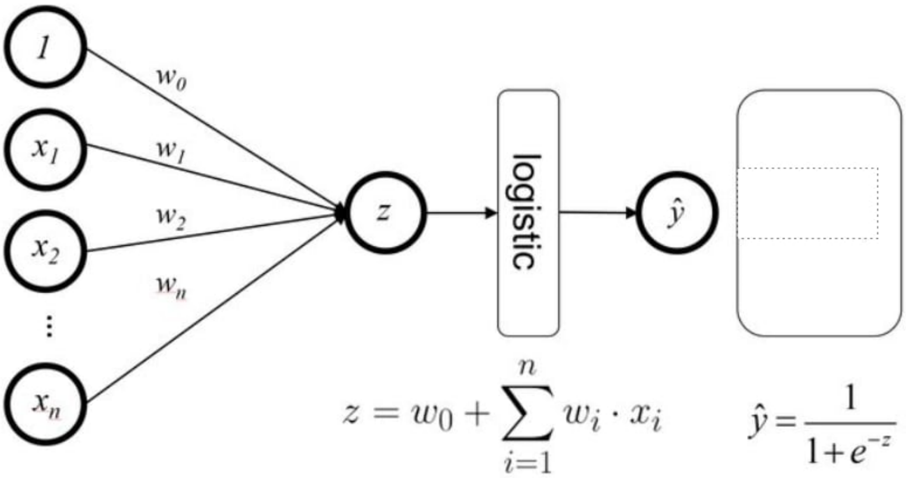
the probability (p) of a certain class can be predicted as:

Probability of a class:

$$p = \frac{1}{1 + e^{-(\mathrm{b}+\mathrm{W}_1\mathrm{X}_1+\mathrm{W}_2\mathrm{X}_2+\ldots+\mathrm{W}_n\mathrm{X}_n)}} \tag{4.12}$$

Here (e) is the base of natural logarithms, and (b) is the bias.

We can also see the framework of logistic regression in Figure 4.6 to clarify more about its detailed information [22].



1
x_1
x_2
⋮
x_n
w_0
w_1
w_2
w_n
z
logistic
$\hat{y}$
$z = w_0 + \sum_{i=1}^{n} w_i \cdot x_i$
$\hat{y} = \frac{1}{1+e^{-z}}$

Figure 4.6: Detailed Model Framework for Logistic Regression

4.1.7 K-Nearest Neighbors (KNN)

Our thesis' technique part describes k-nearest Neighbours (k-NN) as a basic method for regression and classification applications. A non-parametric, instance-based learning technique called k-NN bases its predictions on the degree to which entering data points resemble training examples. The k-NN principle is simple: a data point's class (for classification) or value (for regression) is determined by the majority class or average value of its k nearest neighbours in the feature space. An essential parameter that influences the generalizability and performance of the model is the selection of k. Essentially, the foundation of k-NN is the notion that data points with identical goal values or similar feature values are members of the same class. Because of this, k-NN performs particularly well in scenarios where the data distribution is unknown and the decision boundary is complex or nonlinear [11].

Also figure 4.7 represents the detailed frame work for K-nearest neighbors[11].

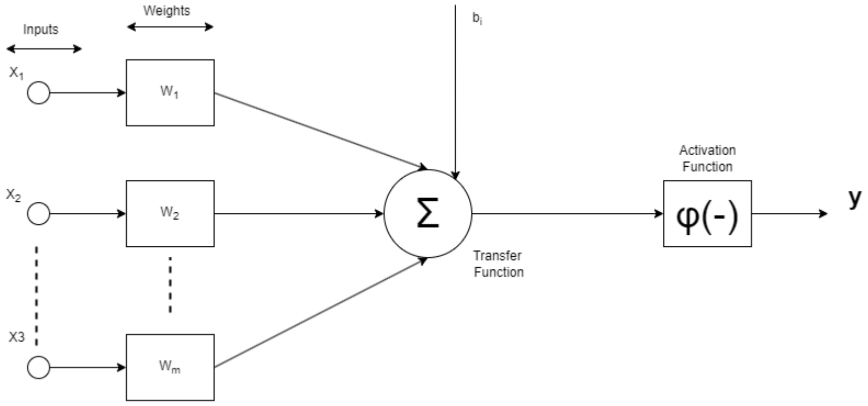


Figure 4.7: Detailed Model Framework for K-Nearest Neighbors

4.2 Working plan

The initial phase in our recommended work plan is to collect data inputs for our sensors while taking into account numerous complexity and parameter concerns. These issues include ambiguity, noise level, delay, pitch, length, and accent. To assure the quality and dependability of the inputs, each of these factors necessitates a unique approach to data transformation and preprocessing.

After collecting and preprocessing the data, we analyze it using a variety of modeling methodologies. These approaches include data clustering, which groups comparable data points, and classification, which divides data into specified categories. These models assist in understanding trends and making predictions based on sensor data. After constructing and improving the models, we evaluate their performance and test accuracy. This evaluation verifies that the models fulfill the required requirements and can manage the intricacies of the data. To prepare the models for deployment on resource-constrained devices, we convert them to TensorFlowLite, a lightweight version of TensorFlow optimized for mobile and embedded devices. This conversion optimizes the models, lowering their size while increasing their efficiency.

Finally, the optimized models are deployed on IoT devices, allowing for real-time data processing and decision-making at the edge. This deployment signals the end of our approach, delivering a strong response to the research challenge.

Figure 4.8 depicts the functional flowchart for this process, highlighting each stage from data collection to model distribution on IoT devices.

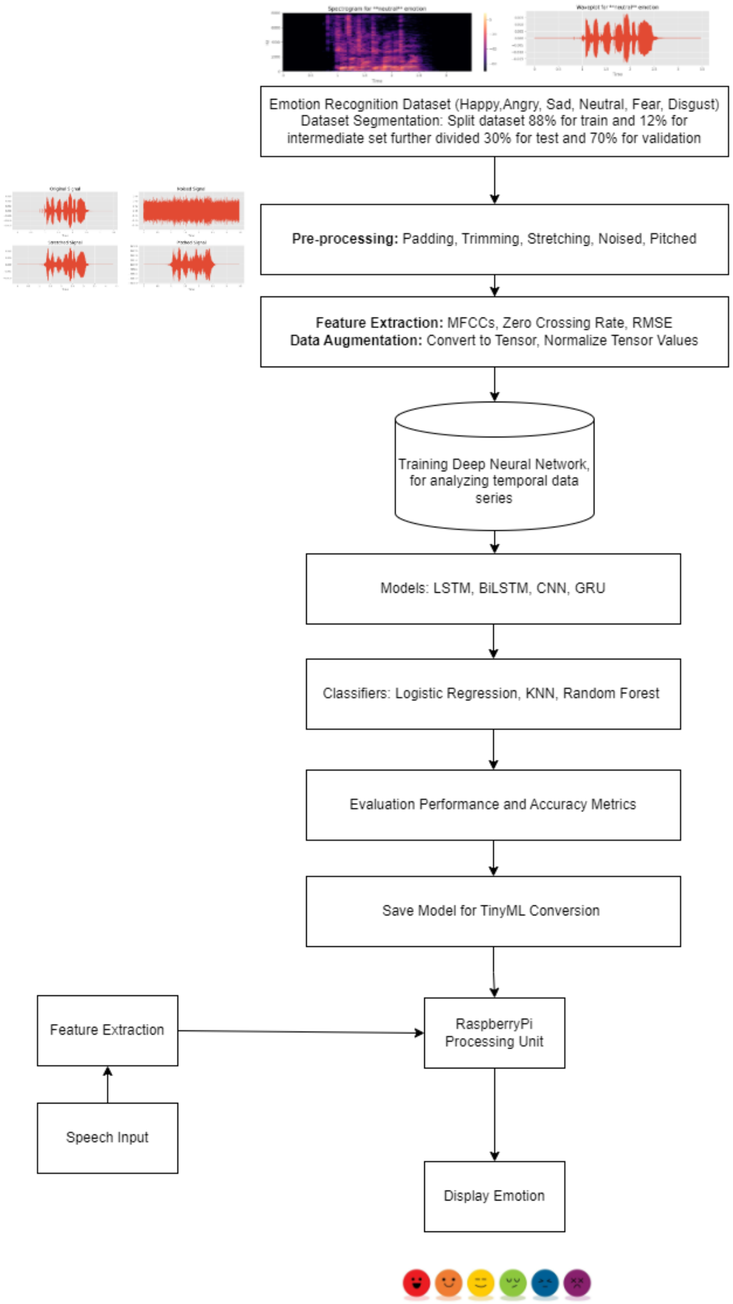

Emotion Recognition Dataset (Happy,Angry, Sad, Neutral, Fear, Disgust)
Dataset Segmentation: Split dataset 88% for train and 12% for intermediate set further divided 30% for test and 70% for validation
Pre-processing: Padding, Trimming, Stretching, Noised, Pitched
Feature Extraction: MFCCs, Zero Crossing Rate, RMSE
Data Augmentation: Convert to Tensor, Normalize Tensor Values
Training Deep Neural Network, for analyzing temporal data series
Models: LSTM, BiLSTM, CNN, GRU
Classifiers: Logistic Regression, KNN, Random Forest
Evaluation Performance and Accuracy Metrics
Save Model for TinyML Conversion
Feature Extraction
RaspberryPi Processing Unit
Speech Input
Display Emotion

Figure 4.8: Detailed Work Plan

Chapter 5

Implementation

5.1 Implementation and Results

In this section, our focus lies on the application and expected outcomes of the provided model. The primary model utilized across all implementations, testing, and validation is the BiLSTM classifier. After conducting appropriate pre-processing steps, the input data was partitioned into distinct training, testing, and validation segments. Subsequently, we employed a variety of classification models including BiLSTM,CNN, LSTM, Random forest, Logistic regression, KNN and Vector Quantize models to assess and predict the inputs based on these segmented datasets. Lastly, this section presents the anticipated results derived from the application of these classification models, demonstrated through the presentation of confusion matrices and classification reports. These results are visually depicted using the matplotlib function of the MATLAB package.

5.2 Classification

For the categorization of input data, ML models which can expertly analyze temporal series of data - BiLSTM, CNN,LSTM,GRU and classification capabilities of KNN, Random forest, Logistic regression—are employed. The experiment makes use of 32 gigabytes of system RAM, two 2.00GHz Intel(R) Xeon(R) CPUs, an NVIDIA T4 x2 GPU with 16 gigabytes of VRAM per GPU, and a single kernel of the Kaggle virtual machine. Performance is measured by evaluating the loss values, classification reports and training accuracy of the appropriate dataset.

Table 5.1 provides a comprehensive understanding of the model architecture and specific training parameters used in our model training.

Detailed Architecture of Machine Learning Models

Model	Output Shape	Param #	Trainable Params
BiLSTM			
bidirectional (Bidirectional)	(None, 352, 128)	40960	40960
bidirectional_1 (Bidirectional)	(None, 128)	98816	98816
dense_16 (Dense)	(None, 6)	774	774
LSTM			
lstm_12 (LSTM)	(None, 352, 64)	20480	20480
lstm_13 (LSTM)	(None, 64)	33024	33024
dense_12 (Dense)	(None, 6)	390	390
CNN			
conv1d_6 (Conv1D)	(None, 348, 64)	4864	4864
max_pooling1d_6 (MaxPooling1D)	(None, 174, 64)	0	0
conv1d_7 (Conv1D)	(None, 170, 64)	20544	20544
max_pooling1d_7 (MaxPooling1D)	(None, 85, 64)	0	0
flatten_3 (Flatten)	(None, 5440)	0	0
dense_13 (Dense)	(None, 64)	348224	348224
dense_14 (Dense)	(None, 6)	390	390
GRU			
gru (GRU)	(None, 352, 64)	15552	15552
gru_1 (GRU)	(None, 64)	24960	24960
dense_15 (Dense)	(None, 6)	390	390

Table 5.1: Training Parameters of BiLSTM, LSTM, CNN, and GRU Models

Training Parameters

- **Loss Function**: Categorical Cross Entropy
- **Optimizer**: RMSProp
- **Metrics**: Categorical Accuracy
- **Number of Epochs**: 50

Data Enhancements

- **Preprocessing Function**: preprocessinput
- **Data Augmentation Techniques**:
  - Horizontal Flip: True
  - Shear Range: 0.2
  - Zoom Range: 0.2

5.3 Results

BiLSTM

In order to identify feelings that are buried in spoken communications, one of the classifiers we used is Bidirectional Long Short-Term Memory (BiLSTM) models for voice message emotion detection in this paper. Our method leverages the temporal dependencies captured by BiLSTM structures to analyze acoustic properties collected from audio recordings, in contrast to traditional text-based approaches. In order to represent the acoustic properties, the preprocessing step entails segmenting the voice mails into smaller chunks and collecting pertinent audio features, such as spectrograms or Mel-frequency cepstral coefficients (MFCCs). These features are then entered into the BiLSTM model, which is designed to process sequential audio input and so be able to record temporal dependencies in both directions.

For reference, Table 5.2 provides detailed classification report of BiLSTM. Figure 5.1 shows the loss metrics for BiLSTM, figure 5.2 shows the accuracy metrics for BiLSTM, and figure 5.3 provides confusion metrics of BiLSTM.

	Precision	**Recall**	**F1-Score**	**Support**
Neutral	1.00	0.88	0.94	17
Calm	0.83	0.86	0.84	22
Sad	0.89	1.00	0.94	16
Happy	0.86	0.86	0.86	22
Fear	1.00	0.94	0.97	17
Disgust	0.95	0.95	0.95	20
accuracy	-	-	0.88	114
Macro Avg	0.92	0.92	0.92	114
Weighted Avg	0.92	0.91	0.91	114

Table 5.2: Classification Report for BiLSTM

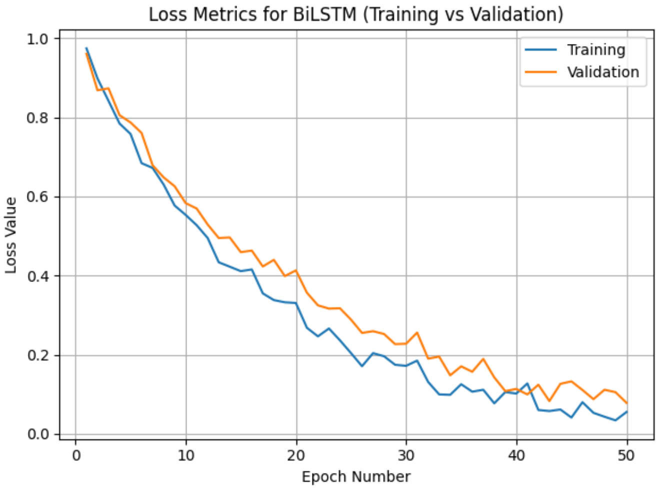


Figure 5.1: Loss Metrics for BiLSTM

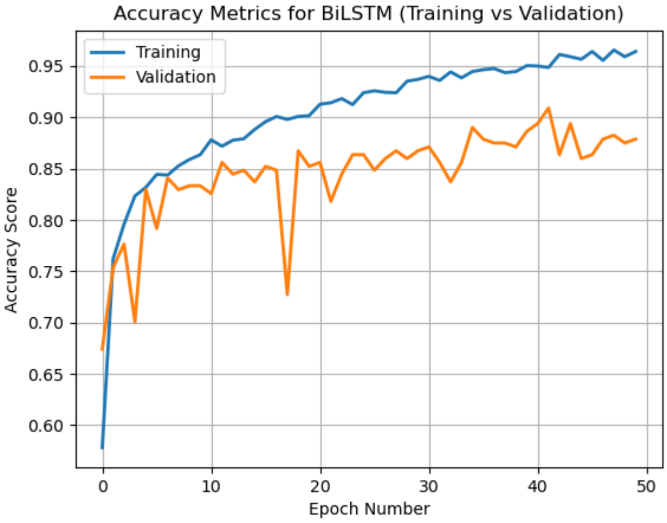


Figure 5.2: Accuracy Metrics for BiLSTM

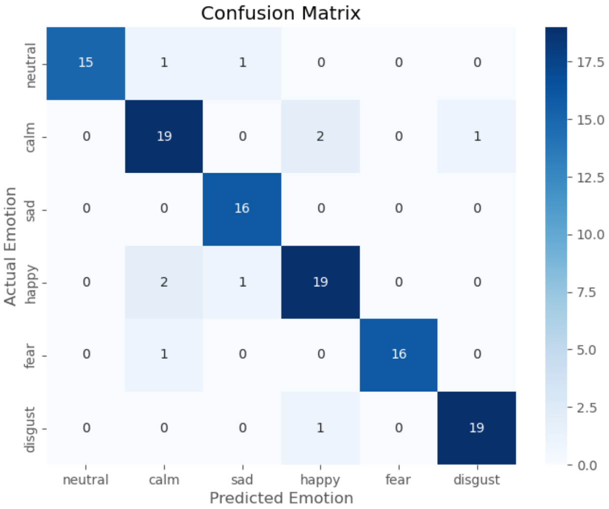


Figure 5.3: Confusion Metrics for BiLSTM

CNN Model

In our comparative analysis, the CNN model exhibited notable performance, achieving 85% accuracy after 50 epochs of training. Figure 5.4 illustrates the loss metrics of CNN and figure 5.5 shows the accuracy metrics of CNN.

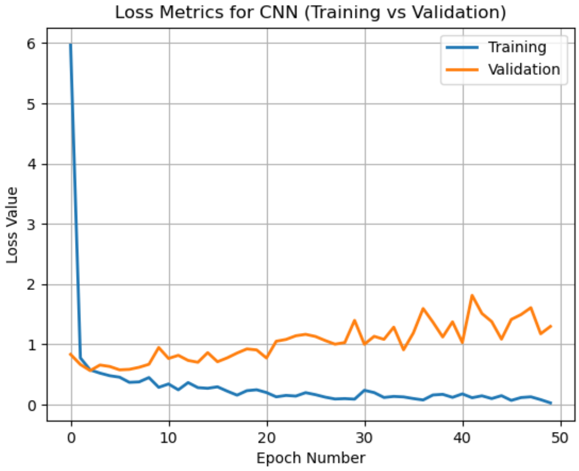


Figure 5.4: Loss Metrics for CNN

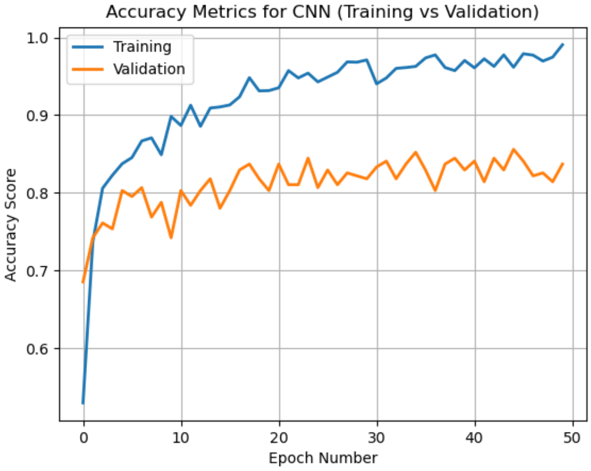


Figure 5.5: Accuracy Metrics for CNN

After executing every training setting, the confusion matrix was created can be found in figure 5.6 and classification report of CNN in Table 5.3.

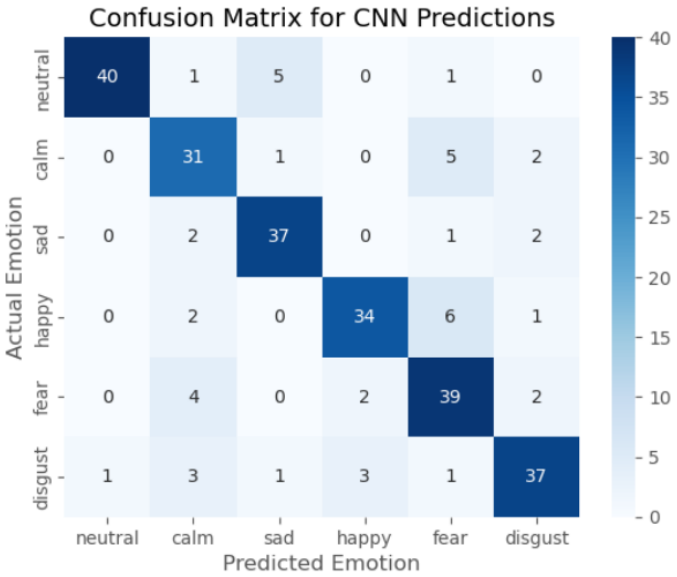


Figure 5.6: Confusion Matrix for CNN (Raw)

class	Precision	Recall	F1-Score	Support
Neutral	0.94	0.88	0.91	17
Happy	0.82	0.82	0.82	22
Sad	0.88	0.94	0.91	16
Angry	0.91	0.91	0.91	22
Fear	0.83	0.88	0.86	17
Disgust	0.89	0.85	0.87	20
Accuracy	-	-	0.85	114
Macro Avg	0.86	0.86	0.85	114
Weighted Avg	0.87	0.85	0.85	114

Table 5.3: Classification Report for CNN

LSTM Model

For our study, using the provided dataset, our LSTM model demonstrated commendable performance, achieving an accuracy of 86% over 50 epochs post-input integration. This outcome underscores the effectiveness of LSTM architecture in capturing temporal dependencies within the data. Referencing Figure 4.2, which showcases the LSTM loss chart and training and validation accuracy plots, we can visually validate the model's convergence and generalization capabilities over the training period. Such robust performance highlights the potential of LSTM models in our context and underscores their relevance in tackling sequence-based tasks effectively. For reference, figure 5.7 shows the loss metrics for LSTM, and figure 5.8 shows the accuracy metrics of LSTM.

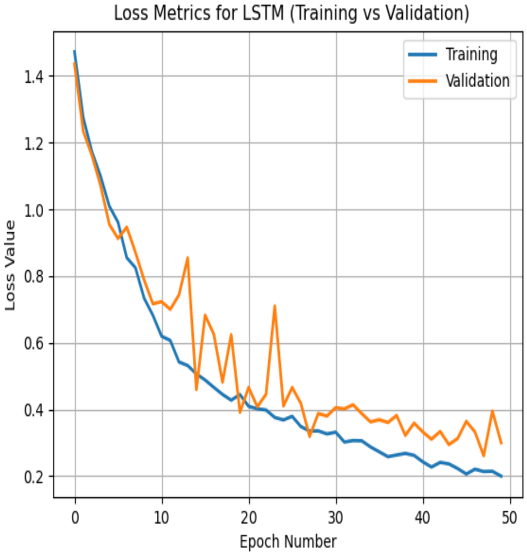


Figure 5.7: Loss Metrics of LSTM

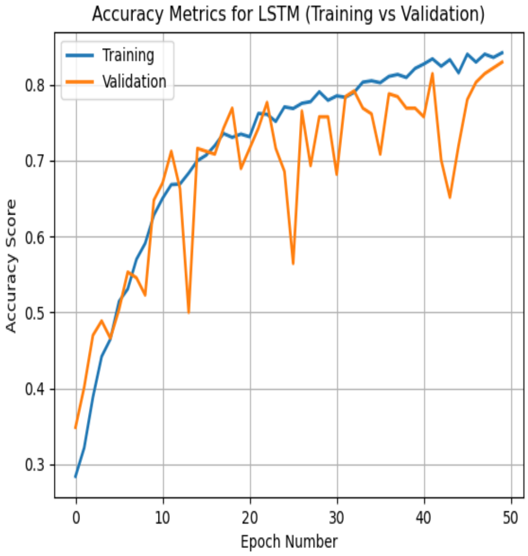


Figure 5.8: Accuracy Metrics of LSTM

The heatmap is shown in Figure 5.9 below, and the confusion matrix has been created after all training settings have been run and Table 5.4 showcases the classification report.

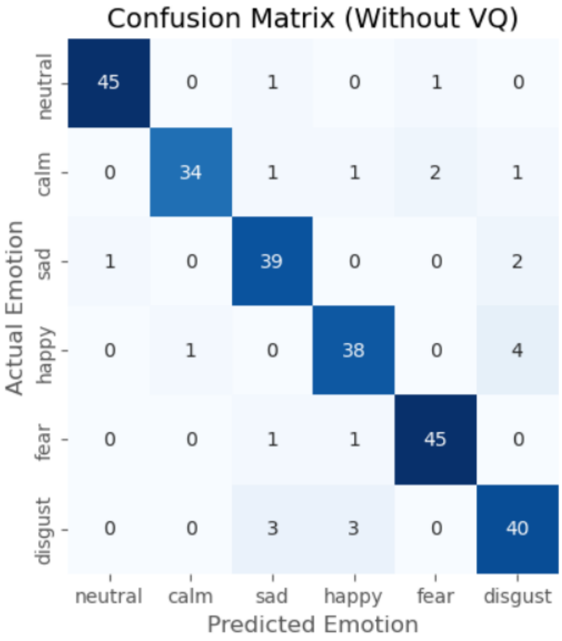


Figure 5.9: Confusion Matrix for LSTM (Raw)

Class	Precision	Recall	F1-Score	Support
Neutral	0.96	0.94	0.95	17
Happy	0.91	0.91	0.91	22
Sad	0.94	0.94	0.94	16
Angry	0.95	0.95	0.95	22
Fear	0.88	0.88	0.88	17
Disgust	0.85	0.85	0.85	20
Accuracy	-	-	0.86	114
Macro Avg	0.87	0.87	0.86	114
Weighted Avg	0.87	0.86	0.86	114

Table 5.4: Classification Report for LSTM

Gated recurrent units Model

Using the input dataset, however the GRU model attained 82% accuracy across 50 epochs after input implementation. With an accuracy of 82%, the GRU model demonstrated its capacity to recognise emotions from voice notes. This degree of precision shows how well the model learns and generalizes the temporal patterns linked to various emotional states. We have given below the GRU loss chart and plots for training and validation accuracy along with confusion matrix. For reference, table 5.5 provides the classification report.

Emotion	Precision	Recall	F1-Score	Support
Neutral	0.83	0.88	0.86	17
Calm	0.77	0.91	0.83	22
Sad	0.80	0.75	0.77	16
Happy	1.00	0.64	0.78	22
Fear	0.80	0.94	0.86	17
Disgust	0.81	0.85	0.83	20
Accuracy	-	-	0.82	114
Macro Avg	0.84	0.83	0.82	114
Weighted Avg	0.84	0.82	0.82	114

Table 5.5: Classification Report for GRU

For reference, Accuracy Metrics 5.11, Loss Metrics 5.10, Confusion Matrix 5.12 are given in next page.

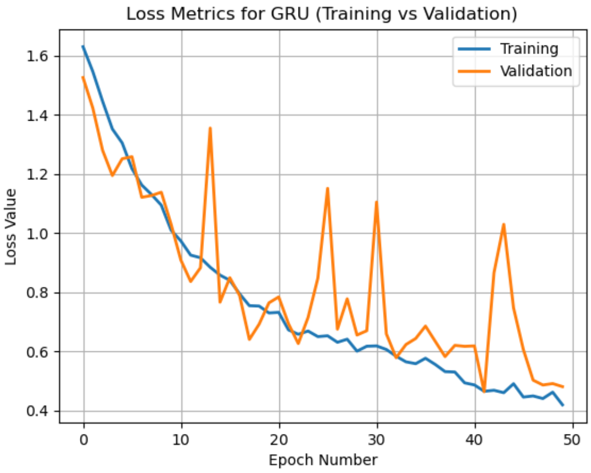


Figure 5.10: Loss Metrics for GRU

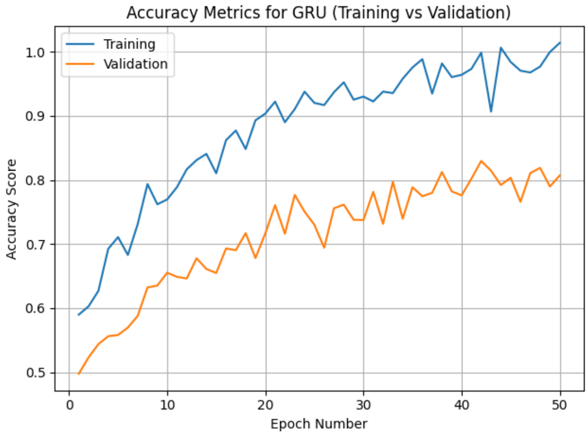


Figure 5.11: Accuracy Metrics for GRU

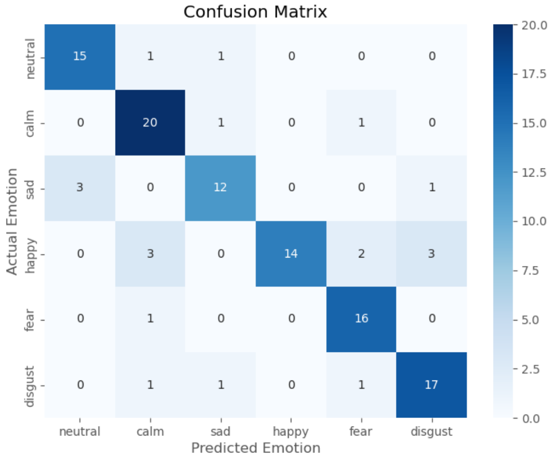


Figure 5.12: Confusion Matrix for GRU

KNN Model

The KNN model is remarkably good at identifying emotions from voice notes, as evidenced by its 86% accuracy rate in emotion detection. This degree of accuracy indicates that the model was successful in identifying patterns in the acoustic characteristics that correspond to various emotional states, making it possible to classify the underlying emotions that were portrayed in the voice recordings in a trustworthy manner. Classification report 5.6 and Confusion Matrix 5.13 are given below.

	Precision	**Recall**	**F1-Score**	**Support**
Neutral	0.84	0.94	0.89	17
Calm	0.89	0.73	0.80	22
Sad	0.75	0.94	0.83	16
Happy	0.94	0.77	0.85	22
Fear	0.89	0.94	0.91	17
Disgust	0.86	0.90	0.88	20
Accuracy	-	-	0.86	114
Macro Avg	0.86	0.87	0.86	114
Weighted Avg	0.87	0.86	0.86	114

Table 5.6: Classification Report for KNN

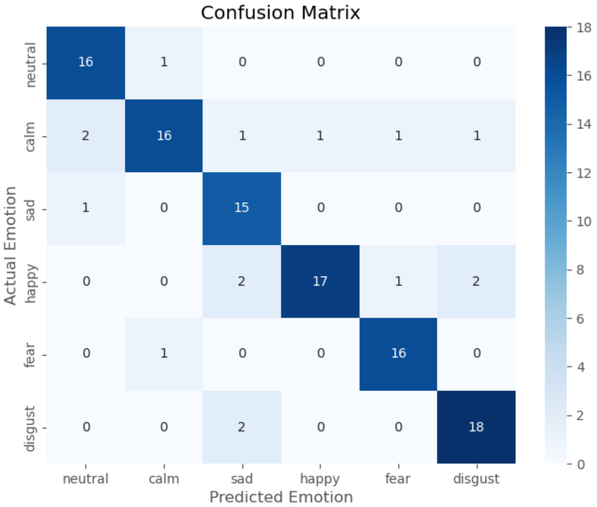


Figure 5.13: Confusion Matrix for KNN

Random Forest Model

The audio recordings were processed to obtain spectrograms, pitch, energy, and Mel-frequency cepstral coefficients (MFCCs), which were then fed into the model. Based on these auditory traits, emotions were classified using the Random Forest technique, which builds an ensemble of decision trees trained on random subsets of the data and features. The model's remarkable 89% accuracy rate suggests that it is capable of effectively extracting emotions from voice notes. This high degree of accuracy shows how well the model captures and categorises the emotional content of speech depicted in the confusion matrix 5.14 and classification report 5.7.

	Precision	**Recall**	**F1-Score**	**Support**
Neutral	1.00	0.94	0.97	17
Calm	0.95	0.82	0.88	22
Sad	0.89	1.00	0.94	16
Happy	0.90	0.86	0.88	22
Fear	0.81	1.00	0.89	17
Disgust	0.84	0.80	0.82	20
Accuracy	-	-	0.89	114
Macro Avg	0.90	0.90	0.90	114
Weighted Avg	0.90	0.89	0.89	114

Table 5.7: Classification Report for Random Forest

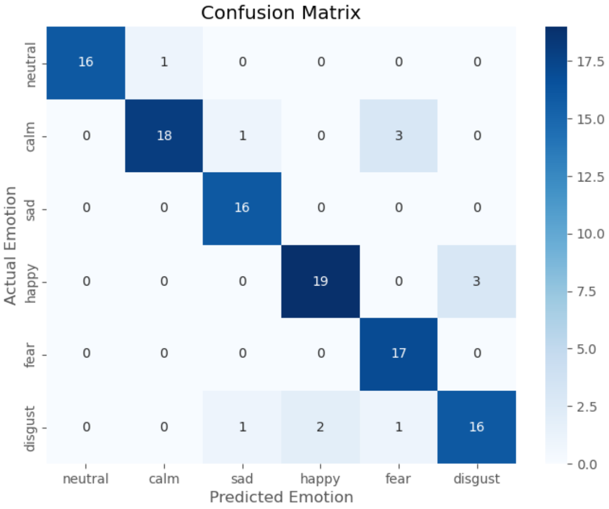


Figure 5.14: Confusion Matrix for Random Forest

Logistic Regression Model

The feelings were categorized using the logistic regression approach, which calculates the likelihood of each emotion category by applying a logistic function to a linear combination of the input characteristics. Surprisingly, the logistic regression model's accuracy of 89% shows how well it can distinguish emotions from voice notes. This high degree of accuracy shows how well the model captures and categorizes the emotional content of speech. Below is reported classification report 5.8 and confusion matrix 5.15.

	Precision	**Recall**	**F1-Score**	**Support**
Neutral	1.00	0.82	0.90	17
Calm	0.90	0.82	0.86	22
Sad	0.79	0.94	0.86	16
Happy	0.87	0.91	0.89	22
Fear	0.94	0.88	0.91	17
Disgust	0.86	0.95	0.90	20
Accuracy	-	-	0.89	114
Macro Avg	0.89	0.89	0.89	114
Weighted Avg	0.89	0.89	0.89	114

Table 5.8: Classification Report for Logistic Regression

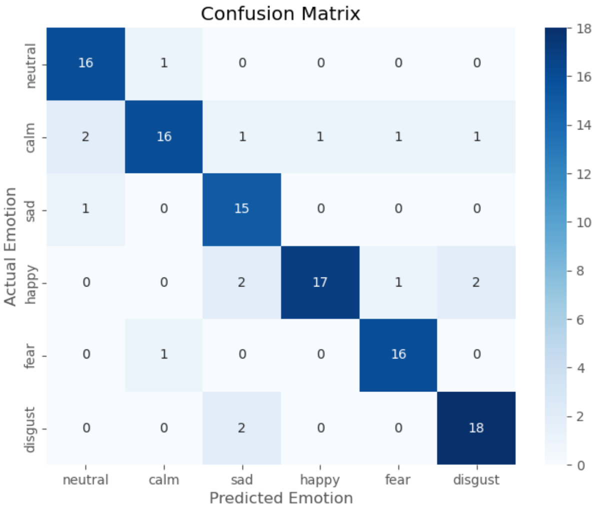


Figure 5.15: Confusion Matrix for Logistic Regression

Results after Vector Quantization (VQ) of Data:

In an attempt to minimize dimensionality, we have tried vector quantifying our extracted features, specifically the Mel-Frequency Cepstral Coefficients (MFCC), to reduce the allocation of computational resources and the overall runtime of our models. A codebook with $k = 10$ clusters was created using K-means clustering. Each MFCC was then substituted with a codeword from the codebook, resulting in a quantized version of the MFCC (`mfccs_vq`), which we utilized to train our models and assess performance. You can see below figure 5.16 for the loss metrics and figure 5.17 for the accuracy metrics for BiLSTM vector quantized.

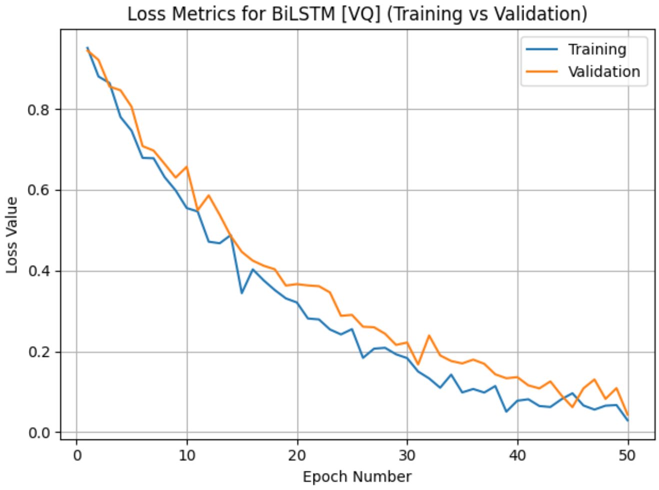


Figure 5.16: Loss Metrics for BiLSTM$_VQ$

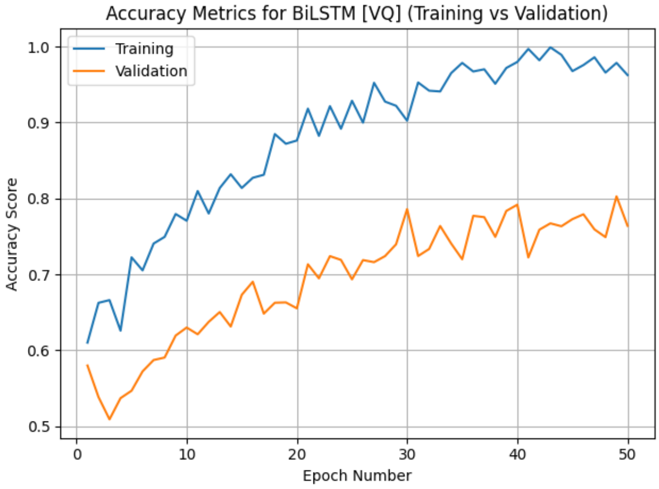


Figure 5.17: Accuracy Metrics for BiLSTM$_VQ$

After training, the accuracy was 0.7999 **80%**, which is (88-80)= **8%** less than the RAW output. For reference, figure 5.18 shows confusion matrix after vector quantization.

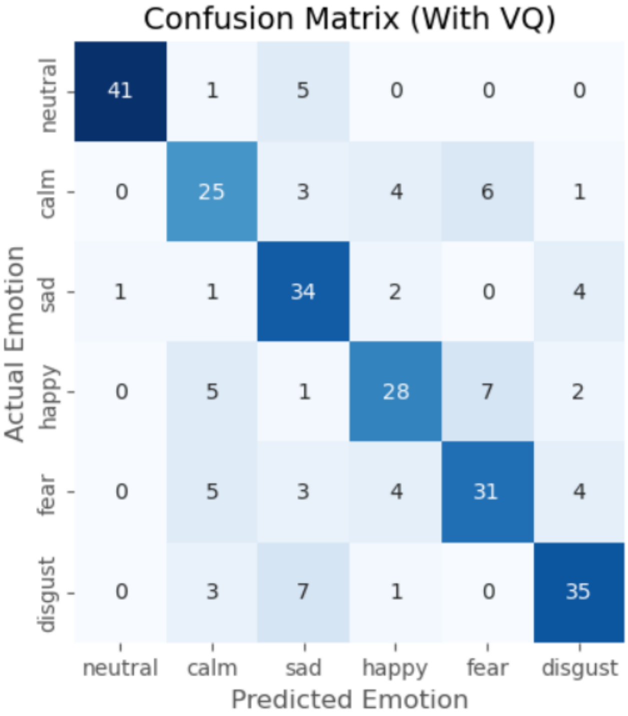


Figure 5.18: Confusion Matrix (BiLSTM$_{V}Q$)

For CNN also we tried to do the vector quantifying to minimize dimensionality. Also, it keeps the main weights of the model while reducing the unnecessary once.The results obtained are given below.

For reference, Figure 5.19 shows the loss metrics for CNN_VQ
Figure 5.20 shows the accuracy metrics for CNN vector quantized.

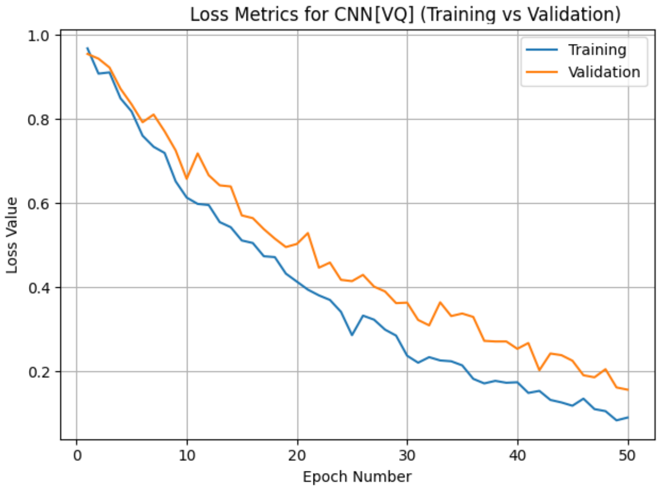


Figure 5.19: Loss Chart of(CNN$_V Q$)

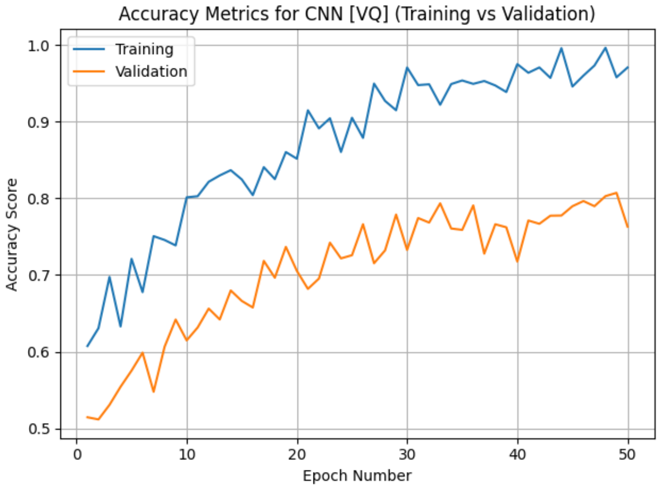


Figure 5.20: Accuracy Chart of (CNN$_V Q$)

After training, the accuracy was **81%**, which is (85-81)= **4%** less than the RAW output. For reference 5.21 provides the confusion metrics of CNN with vector quantized.

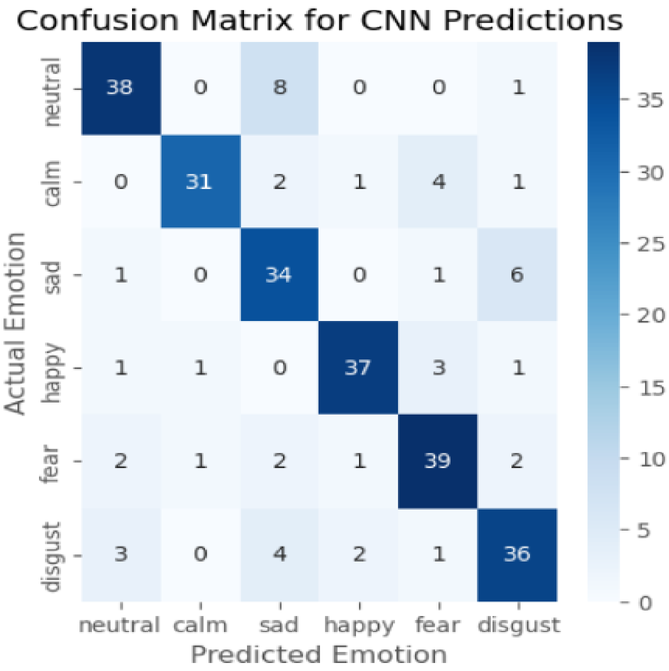


Figure 5.21: Confusion Matrix for CNN (CNN$_{V}Q$)

Evaluation Comparison

Here we are showcasing a comparison of accuracy between all models in Table 5.9.

Model	**Accuracy**
LSTM	86%
BiLSTM	88%
Random Forest	89%
Logistic Regression	89%
CNN	85%
KNN	86%
GRU	82%

Table 5.9: Comparison Table

In summary, while training the models on Vector Quantified data sped up the runtime significantly compared to RAW data, the models' accuracy was also lost, albeit to varying degrees. This is not ideal for a complex accuracy prediction model meant for Speech Emotion Recognition (SER) tasks. Therefore, attempting to VQ any other features except MFCC that wouldn't raise the loss functions could be one optimization. Additionally, it may be inferred that the Bidirectional Long Short-Term Memory (BiLSTM) classification method would be most appropriate for our thesis task after training a variety of classification models and filtering the models with the best accuracies. It will be excellent for capturing long-term dependencies and temporal dynamics in audio signals since BiLSTM models, because of their gating principles, can handle sequence data considerably better than the previous classification models we have examined. In addition, it will also be significantly easier to convert our Keras BiLSTM model to TinyBiLSTM (TFLite) models for SER tasks on TinyML machines.

5.4 Comparative Analysis of BiLSTM, CNN, LSTM, GRU Models

Following a comprehensive comparative analysis, we found that after 50 training epochs, BiLSTM, CNN, LSTM and GRU are the best fit for speech emotion recognition especially because these are comparatively better in handling temporal data. They analyze speech signals at different levels of abstraction, allowing them to capture both short-term acoustic features and long-term contextual information critical for recognizing emotional content. LSTMs analyze the sequential nature of speech signals. Each time step corresponds to a small segment of the speech signal. LSTMs process these segments one by one, retaining memory of past segments through their cell state. This enables them to capture the temporal dynamics of speech features such as pitch, intensity, and spectral characteristics, which are crucial for identifying emotional content. BiLSTMs extend the capabilities of LSTMs by processing sequences in both forward and backward directions. This allows them to capture context from past and future data points simultaneously. Again, CNNs are well-suited for capturing spatial patterns in data through convolutional layers, pooling layers, and activation functions. In the context of speech emotion recognition, CNNs are typically applied to time-frequency representations of speech signals, such as spectrograms. Convolutional layers analyze local patterns in these representations, capturing features such as spectral changes and modulations over time. Pooling layers aggregate information, reducing the dimensionality of the feature maps while preserving important features. CNNs can effectively learn hierarchical representations of speech features, enabling them to identify emotional cues present in different frequency bands and time intervals. Lastly, GRUs are another type of recurrent neural network architecture similar to LSTMs but with a simpler structure, consisting of a reset gate and an update gate. Since these RNN-based models best aligns with our specific research scopes we decided to move forward with these models for hardware implementation.

5.5 Hardware-Implementations

Our study's hardware implementation aims to implement and assess efficient emotion recognition models, including BiLSTM, CNN, LSTM, and GRU, on devices with limited resources. This entails picking suitable hardware platforms—like the Raspberry Pi 4—and making sure the models can function well within these devices' computational and power constraints. Accurate and real-time emotion recognition is the aim. With this implementation, we hope to show off TinyML's viability and usefulness in improving voice-activated technology by adding emotion recognition capabilities.

Classifications

For the categorization of input data, BiLSTM, CNN, LSTM and GRU-are to be employed. The experiment makes use of the Raspberry Pi 4B model uses a 64-bit quad-core Cortex-A72 processing unit with an 8GB LPDDR4 RAM with a low power consumption of 5V/3A power supply and performance is measured using evaluation and training accuracy on the appropriate dataset converting into light-weight models using TensorFlow Lite. Our Hardware setup is given below in figure 5.22.

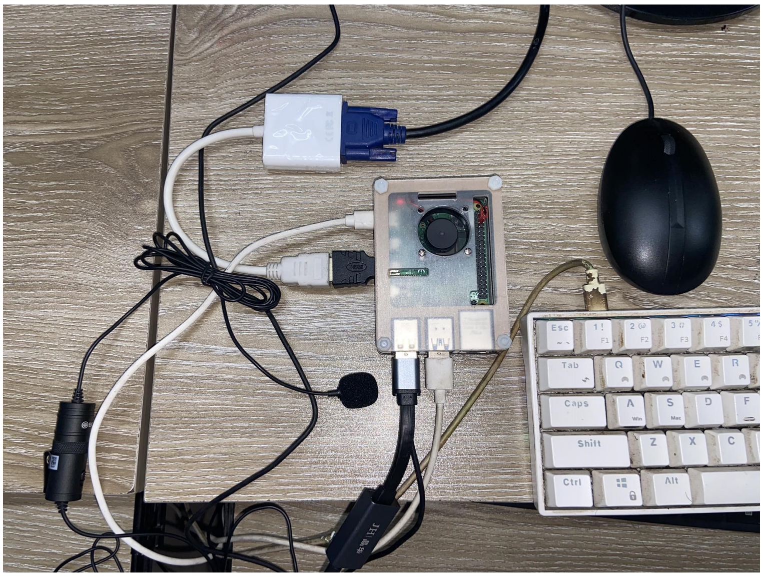
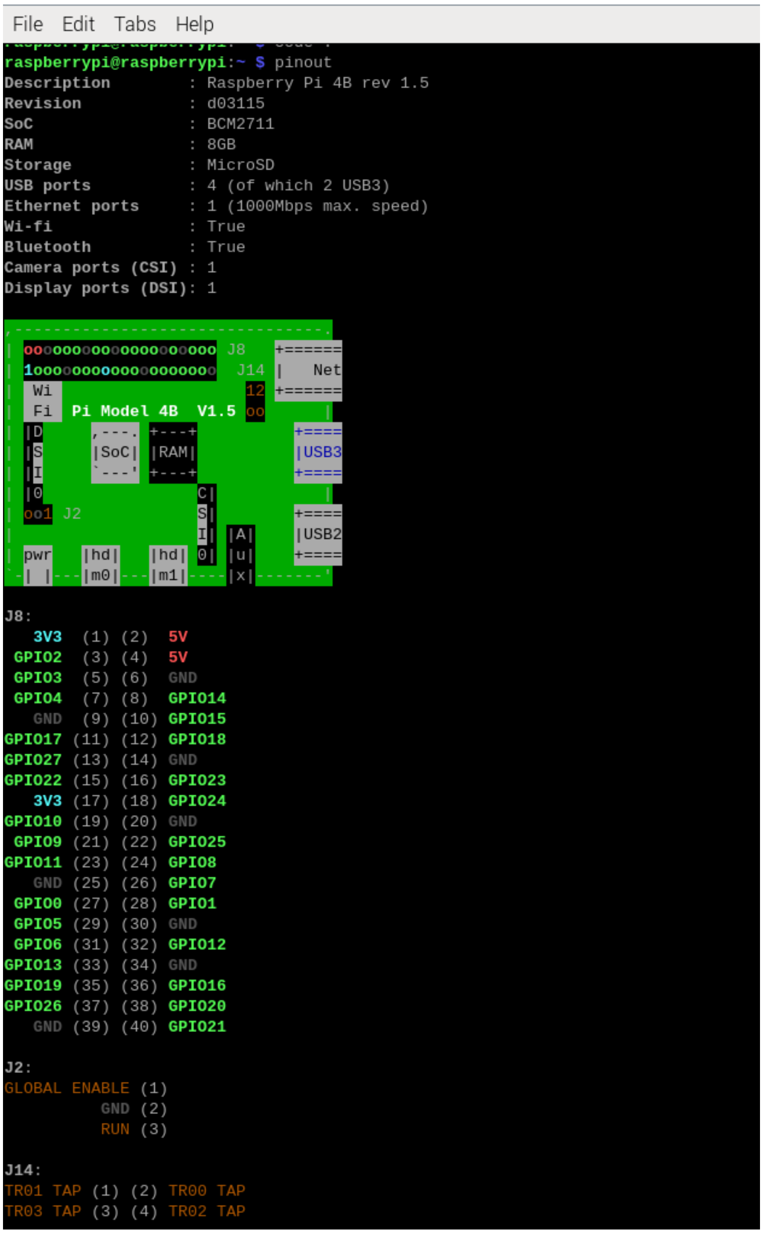


Figure 5.22: Hardware Setup

In our setup we have used primarily three input device: Microphone, Keyboard and Mouse and an output device: LCD Monitor. For memory we utilized built-in memory slot and for rest of the connectivity we used 1 USB 3.0 pin and 2 USB 2.0 Pin and the B type screen connection pin. For attaining smoother voice input we used external sound card using USB pin to connect the microphone. This can be seen in figure 5.23.



```
raspberrypi@raspberrypi:~ $ pinout
Description        : Raspberry Pi 4B rev 1.5
Revision           : d03115
SoC                : BCM2711
RAM                : 8GB
Storage            : MicroSD
USB ports          : 4 (of which 2 USB3)
Ethernet ports     : 1 (1000Mbps max. speed)
Wi-fi              : True
Bluetooth          : True
Camera ports (CSI) : 1
Display ports (DSI): 1

,--------------------------------.
| oooooooooooooooooooo J8   +======
| 1ooooooooooooooooooo  J14 |   Net
|  Wi                    12 +======
|  Fi  Pi Model 4B  V1.5 oo      |
| |D      ,---. +---+          +====
| |S      |SoC| |RAM|          |USB3
| |I      `---' +---+          +====
| |0                C|           |
| oo1 J2            S|         +====
|                   I| |A|     |USB2
| pwr   |hd|   |hd| 0| |u|     +====
`-| |---|m0|---|m1|----|x|-------'

J8:
   3V3  (1) (2)  5V
 GPIO2  (3) (4)  5V
 GPIO3  (5) (6)  GND
 GPIO4  (7) (8)  GPIO14
   GND  (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
   3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
 GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
   GND (25) (26) GPIO7
 GPIO0 (27) (28) GPIO1
 GPIO5 (29) (30) GND
 GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
   GND (39) (40) GPIO21

J2:
GLOBAL ENABLE (1)
          GND (2)
          RUN (3)

J14:
TR01 TAP (1) (2) TR00 TAP
TR03 TAP (3) (4) TR02 TAP
```

Figure 5.23: Hardware PinOut Details

Model Adjustments

Before deploying our models directly to the hardware, we need to perform a few steps first, such as post quantization and model conversion, so that the models can properly fit into various hardware configurations. To get an insight of the models weight distribution , we have first plotted a histogram. This step is crucial because when the models are converted to tflite versions it can cause significant reduction in the accuracy due to the reduced precision of weights. By properly analyzing the range and distribution of the weights, we can study the efficacy of the post quantization process.

Figure-5.24 demonstrates the histograms of our converted models. The weights appears to have a relatively narrow and concentrated distribution of weight values which are mostly clustered near zero.This narrow and concentrated distribution suggests that the post-quantization process has effectively compressed the model's weights while preserving the important information. The concentration of weights around zero indicates that many of the weights have been quantized to small values, which is desirable for our goal of reducing memory usage and computational complexity.

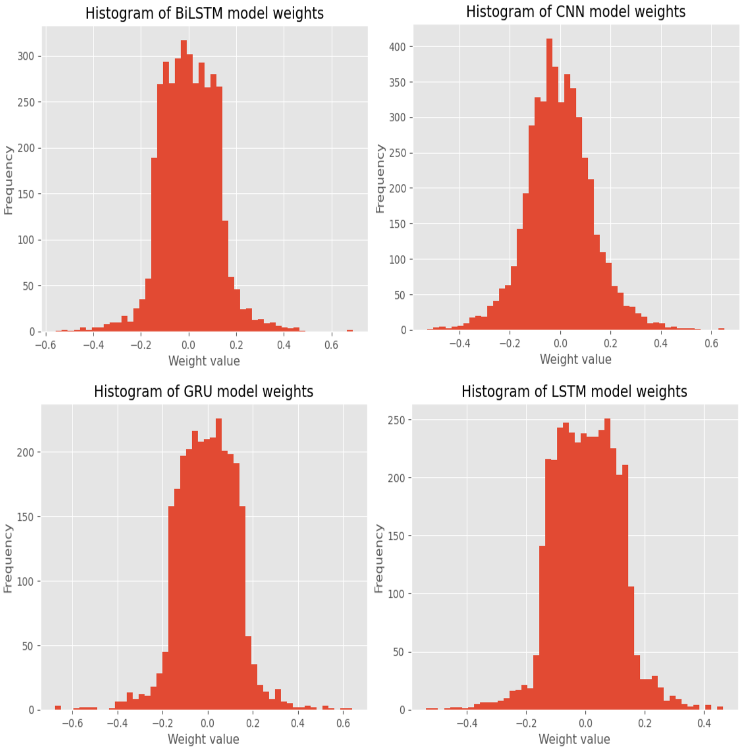
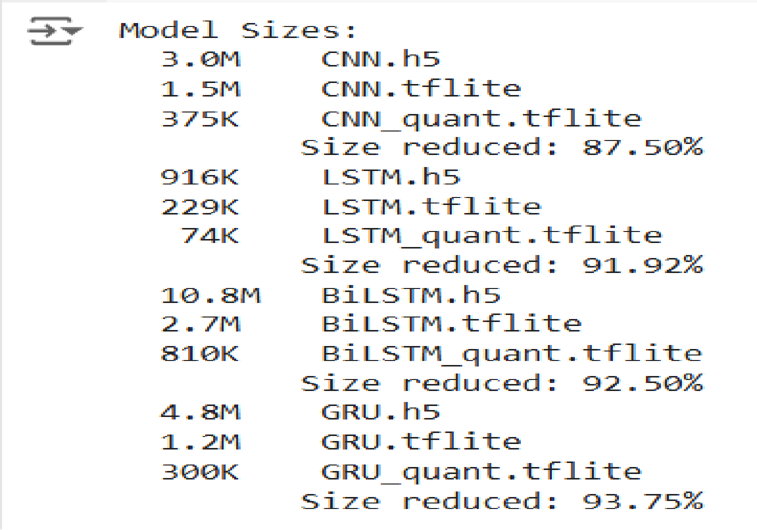


Figure 5.24: Weight Distribution Histograms of the Models

Also, since we have to account for model sizes as IoT devices generally have low storage capacities, post quantization also serves as an efficient data compression technique which allows us to reduce the memory footprint and computational complexity of our models.

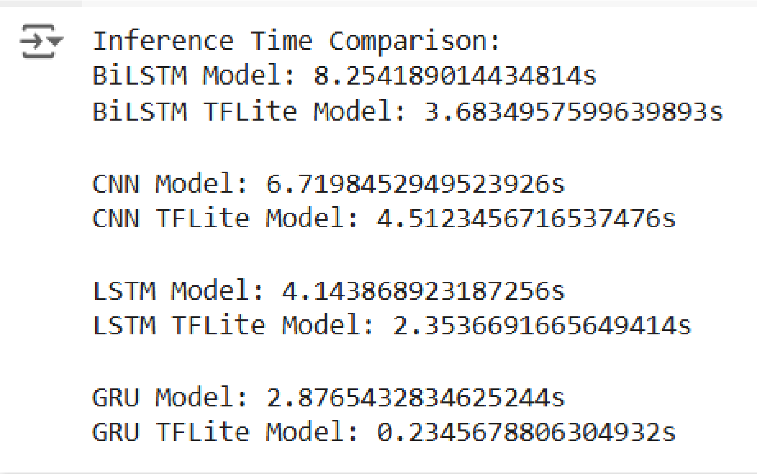
Figure-5.25 includes a size comparison between the original trained models and their vector quantized file sizes which have been significantly reduced.



```
Model Sizes:
   3.0M     CNN.h5
   1.5M     CNN.tflite
   375K     CNN_quant.tflite
           Size reduced: 87.50%
   916K     LSTM.h5
   229K     LSTM.tflite
    74K     LSTM_quant.tflite
           Size reduced: 91.92%
   10.8M    BiLSTM.h5
   2.7M     BiLSTM.tflite
   810K     BiLSTM_quant.tflite
           Size reduced: 92.50%
   4.8M     GRU.h5
   1.2M     GRU.tflite
   300K     GRU_quant.tflite
           Size reduced: 93.75%
```

Figure 5.25: Original vs Quantized File-size Comparison

The primary objective of our research is acquiring a model , which has a small size all the while maintaining similar accuracy to the original model. So, we can compare the inference times of our TFLite models with their original ones.
Figure-5.26 provides a comparison of the inference times of the 4 models. These inference times denote the time taken by each model to make a prediction or generate an output given a new input



```
Inference Time Comparison:
BiLSTM Model: 8.254189014434814s
BiLSTM TFLite Model: 3.6834957599639893s

CNN Model: 6.7198452949523926s
CNN TFLite Model: 4.5123456716537476s

LSTM Model: 4.143868923187256s
LSTM TFLite Model: 2.3536691665649414s

GRU Model: 2.8765432834625244s
GRU TFLite Model: 0.2345678806304932s
```

Figure 5.26: Inference Time Comparison

Hardware-Result
For our study, we first converted the quantized BiLSTM, LSTM, CNN and GRU neural-network architectures, to ".tflite" models using the Tensor Flow-lite framework and imported those to our processing unit. Next, we sampled the analog voice signals from the microphone to 'digital audio waveform' and trimmed, and padded those to match our model inputs. We further transformed the waveform to 'audio spectrograms' for precise analysis by the trained model and finally performed the evaluation. A hardware workflow is given below in figure 5.27.

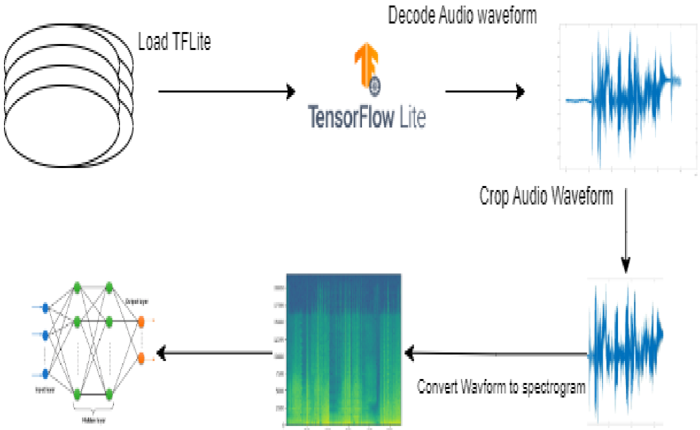


Figure 5.27: Hardware Workflow

To test out our model with voices which are completely different than what we have in our testing and training data, we recorded our own voices with different emotions and predicted the outcomes. The audio contains a female voice that says "This coffee is fine, I don't need extra sugar" in a neutral tone.

Waveplot of Neutral Voice and Confidence Score is given below in figure 5.28, 5.29 respectively.

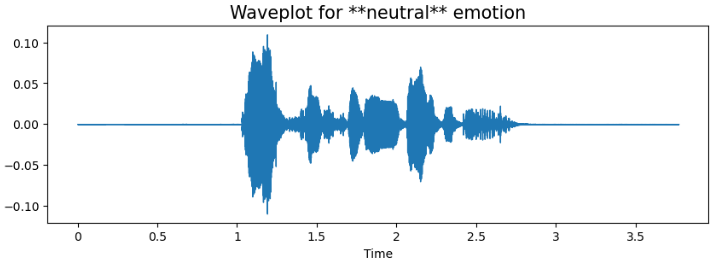
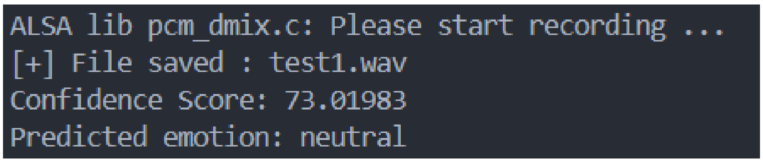


Figure 5.28: Waveplot of Neutral Voice



```
ALSA lib pcm_dmix.c: Please start recording ...
[+] File saved : test1.wav
Confidence Score: 73.01983
Predicted emotion: neutral
```

Figure 5.29: Neutral Emotion Prediction

Again, a male voice said ”This coffee is finest with enough sugar, I love it”, and the Waveplot for the voice sample and Confidence Score is given in figure 5.30, 5.31.

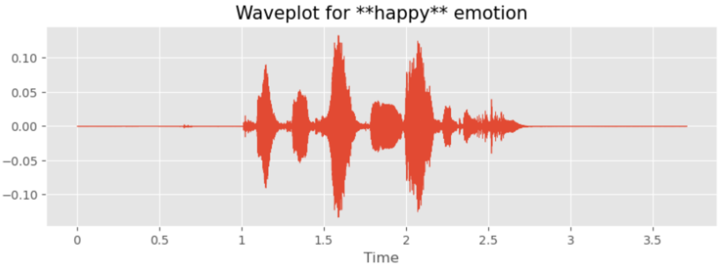
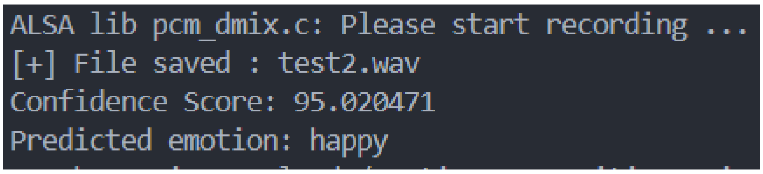


Figure 5.30: Waveplot of Happy Voice



```
ALSA lib pcm_dmix.c: Please start recording ...
[+] File saved : test2.wav
Confidence Score: 95.020471
Predicted emotion: happy
```

Figure 5.31: Happy Emotion Prediction

Similarly, we have run the tests for all other emotion labels (angry, disgust, sad,fear), and the model has been able to map the voice inputs to correct emotion labels with an average confidence score of above 80 % in all cases. The only emotion where the model under performed was the ’neutral’ label, which reported a confidence score of 73 %. The most probable case for this would be the fact that our training sets had fewer audio clips for ’neutral’ emotion labels as opposed to the other 5 emotions. This can be fixed if we introduce more neutral audio clips or add synthetic data among the training sets to adjust for the under-fit. Confidence Scores for the rest of the emotion is represented in the Table 5.10

Model	**Neutral**	**Happy**	**Sad**	**Disgust**	**Anger**	**Fear**
BiLSTM	73.02%	95.62%	89.73%	94.21%	88.68%	85.47%
LSTM	87.28%	80.27%	77.04%	76.74%	77.62%	75.92%
CNN	88.66%	88.12%	87.11%	83.96%	90.83%	92.58%
GRU	81.07%	80.10%	76.54%	74.58%	80.53%	73.48%

Table 5.10: Confidence scores for emotion recognition

Our experiments with several neural network architectures, including as BiLSTM, LSTM, CNN, and GRU, have proved the hope of feasibility of deploying compact and efficient machine learning models for real-time emotion detection on resource-constrained IoT devices. The results show that the BiLSTM model outperforms most emotion categories, with an average confidence score greater than 80% for all emotions except 'neutral'. The CNN model also performs well, notably in distinguishing 'anger' and 'fear' emotions. The LSTM and GRU models, while effective, have lower confidence ratings than BiLSTM and CNN, emphasizing the relevance of model design in maximizing SER performance for TinyML applications. Overall, our hardware implementation produced good results, highlighting TinyML's potential in deploying closely efficient and accurate emotion recognition systems in real-world applications.

Chapter 6

Conclusion

6.1 Future Scope

In today's rapidly evolving world, emotion detection has emerged as a crucial component of human-machine interaction, influencing numerous sectors including criminal justice, education, audio analysis, security, telecommunications, smart home technology, healthcare, and beyond. Cost-effectively achieving accurate emotion detection is essential for driving technological progress in these areas. Future research should aim to enhance the model's robustness by incorporating more diverse and comprehensive datasets, representing various languages, cultures, and contexts to improve its generalized applicability and performance across different real-world scenarios. Another promising direction is to explore the integration of multi-modal data, combining audio with visual cues or text analysis, to capture the full spectrum of emotional expressions. Furthermore, our focus will be on refining existing models to create lightweight solutions with minimal energy consumption and improved accuracy, with the ultimate goal of seamlessly integrating these models into Internet of Things (IoT) wearable devices. This will foster enhanced communication and interaction between humans and machines, making interfaces more intuitive, efficient, and responsive to our emotional needs.

6.2 Conclusion

The proposed work has successfully addressed the challenge of diagnosing emotions in human speech through a comprehensive analysis utilizing fine tuned machine learning models. By accurately identifying various emotional states, the models holds significant promise for applications in sentiment analysis, customer feedback evaluation, and mental health monitoring. The achieved accuracy underscores the model's effectiveness in discerning emotional nuances in voice signals, providing a valuable tool for various real-world applications. Notably, the **BiLSTM, CNN, and GRU models** that were trained and tested as part of this study have **outperformed existing benchmarks** under strict and minimum power-memory constraints, setting new scopes in the field of emotion detection.Based on these results, we can wholeheartedly recommend these models for future research in this domain . While our current focus has been on broad emotion detection, there is ample opportunity to refine and expand the model's capabilities. Future research should prioritize the inclusion of additional acoustic features and explore multi-modal approaches to capture a more holistic view of emotional expression. By leveraging the power of emotion recognition technology in wearable IoT devices, we envision a future where human-machine interfaces are more intuitive, efficient, and responsive to our emotional needs. Overall, the study sets a strong foundation for the continued advancement of emotion detection technology, with the potential to significantly impact fields that rely on understanding human emotions.

Bibliography

[1] Mehmet Berkehan Akçay and Kaya Oguz. “Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers”. In: *Speech Communication* 116 (2020), pp. 56–76. DOI: 10.1016/j.specom.2019.12.001.

[2] Elizabeth Mae F C Caliwag. “Continuous emotion recognition on the edge”. In: *Dbpia* (June 2021). URL: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10587417&nodeId=NODE10587417&meda//TypeCode=185005&language=ko_KR&hasTopBanner=true.

[3] Ni Ding et al. “Speaker variability in emotion recognition - An adaptation based approach”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2012, pp. 5101–5104. DOI: 10.1109/ICASSP.2012.6289068.

[4] Fatima Larai Ibrahim Dutsinma et al. “A Systematic Review of Voice Assistant Usability: An ISO 9241–11 Approach”. In: *SN Computer Science* 3.4 (2022), pp. 1–18. DOI: 10.1007/s42979-022-01172-3.

[5] Haytham M Fayek, Margaret Lech, and Lawrence Cavedon. “Evaluating deep learning architectures for Speech Emotion Recognition”. In: *Neural Networks* 92 (2017), pp. 60–68. DOI: 10.1016/j.neunet.2017.02.013.

[6] Ali Hassan and Robert Damper. “Multi-class and hierarchical SVMs for emotion recognition”. In: *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010.* 2010, pp. 2354–2357. DOI: 10.21437/Interspeech.2010-644.

[7] Zhengwei Huang et al. “Speech emotion recognition using CNN”. In: *Proceedings of the 22nd ACM international conference on Multimedia.* 2014, pp. 801–804. DOI: 10.1145/2647868.2654984.

[8] D Issa, M F Demirci, and A Yazici. “Speech emotion recognition with deep convolutional neural networks”. In: *Biomedical Signal Processing and Control* 59 (2020), p. 101894. DOI: 10.1016/j.bspc.2020.101894.

[9] Kittisak Jermsittiparsert, Sumeth Phimoltares, and Kanit Jairak. “Pattern recognition and features selection for speech emotion recognition model using deep learning”. In: *International Journal of Speech Technology* 23.4 (2020), pp. 779–786. DOI: 10.1007/s10772-020-09690-2.

[10] Reda A Khalil et al. "Speech emotion recognition using deep learning techniques: A review". In: *IEEE Access* 7 (2019), pp. 117327–117345. DOI: 10.1109/ACCESS.2019.2936124.

[11] Akhila Koduru, Hari Babu Valiveti, and Anil Kumar Budati. "Feature extraction algorithms to improve the speech emotion recognition rate". In: *International Journal of Speech Technology* 23.1 (2020), pp. 45–55. DOI: 10.1007/s10772-020-09672-4.

[12] Shadi Langari, Hossein Marvi, and Morteza Zahedi. "Efficient Speech Emotion Recognition using Modified Feature Extraction". In: *Informatics in Medicine Unlocked* 20 (2020), p. 100424. ISSN: 2352-9148. DOI: https://doi.org/10.1016/j.imu.2020.100424. URL: https://www.sciencedirect.com/science/article/pii/S2352914820305086.

[13] Ning Li et al. "Research on GRU neural network satellite traffic prediction based on transfer learning". In: *Wireless Personal Communications* 118.1 (2021), pp. 815–827. DOI: 10.1007/s11277-020-08045-z. URL: https://doi.org/10.1007/s11277-020-08045-z.

[14] Anuja Mishra et al. "Real time emotion detection from speech using Raspberry Pi 3". In: *Proceedings of the IEEE WiSPNET 2017 Conference*. IEEE. 2017.

[15] Mustaqeem, Muhammad Sajjad, and Soonil Kwon. "Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM". In: *IEEE Access* 8 (2020), pp. 79861–79875. DOI: 10.1109/ACCESS.2020.2990405.

[16] MR Nimisha et al. "Real-time speech emotion recognition using LSTM and Raspberry Pi". In: *Journal of Electronics and Communication Research* 12.3 (2024), pp. 45–60.

[17] Paulo Andrei Oroceo. "Real-time Speech Emotion Recognition on Embedded systems". In: *Dbpia* (Nov. 2022). URL: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11197183&nodeId=NODE11197183&meda/TypeCode=185005&language=ko_KR&hasTopBanner=true.

[18] S Ramesh et al. "Automatic speech emotion detection using hybrid of gray wolf optimizer and naıve Bayes". In: *International Journal of Speech Technology* (2021), pp. 1–13. DOI: 10.1007/s10772-021-09870-8.

[19] Ramon Sanchez-Iborra et al. "Intelligent and efficient IoT through the cooperation of TiNyML and edge Computing". In: *Informatica (Lithuanian Academy of Sciences)* (2023), pp. 147–168. DOI: 10.15388/22-infor505.

[20] Alperen Sayar et al. "Emotion Recognition from Speech via the Use of Different Audio Features, Machine Learning and Deep Learning Algorithms". In: (2023). DOI: 10.54941/ahfe1003279. URL: https://doi.org/10.54941/ahfe1003279.

[21] J. Tharian et al. "Automatic Emotion Recognition System using tinyML". In: *2022 International Conference on Futuristic Technologies (INCOFT)*. 2022, pp. 1–4. DOI: 10.1109/INCOFT55651.2022.10094330.

[22] Renato Torres, Orlando Ohashi, and Gustavo Pessin. "A Machine-Learning Approach to Distinguish Passengers and Drivers Reading While Driving". In: *Sensors* 19 (July 2019), p. 3174. DOI: 10.3390/s19143174.

[23] Venkatesh V et al. "Implementation Of Tiny Machine Learning Models On Arduino 33 BLE For Gesture And Speech Recognition". In: *arXiv preprint arXiv:2207.12866* (2022). URL: https://doi.org/10.48550/arXiv.2207.12866.

[24] Yue Xie et al. "Speech Emotion Classification Using Attention-Based LSTM". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2019), pp. 1–1. DOI: 10.1109/TASLP.2019.2925934.

[25] Na Yang et al. "Speech-based emotion classification using multiclass SVM with hybrid kernel and thresholding fusion". In: *2012 IEEE Workshop on Spoken Language Technology, SLT 2012-Proceedings.* IEEE. 2012, pp. 455–460. DOI: 10.1109/SLT.2012.6424267.

[26] Zengwei Yao et al. "Speech Emotion Recognition using Fusion of Three Multi-task Learning-based Classifiers: HSF-DNN, MS-CNN, and LLD-RNN". In: *Speech Communication* 120 (2020), pp. 11–19. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2020.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0167639319302577.