St. Thomas' College of Engineering and Technology

"Emo–Track"

Prepared by

Name	University Roll Number	University Registration No
Pradip Ghosh	12200220013	201220100210050 of 2020-21
Deep Kumar Goenka	12200220038	201220100210025 of 2020-21
Aishi Paul	12200220039	201220100210024 of 2020-21
Abhishek Chaudhury	12200220056	201220100210007 of 2020-21

Under the guidance of

Dr. Arijit Ghosal

Associate Professor, Department of Information Technology

PROJECT REPORT

Submitted in partial fulfillment of the requirement of the degree of B.Tech in Information Technology

Department of Information Technology

Affiliated to

Maulana Abul Kalam Azad University of Technology, West Bengal

MAY, 2024

CERTIFICATION

ST. THOMAS' COLLEGE OF ENGINEERING AND TECHNOLOGY

Faculty of Information Technology

This is to certify that the work in preparing the project entitled "Emo–Track" has been carried out by Deep Kumar Goenka, Aishi Paul, Abhishek Chaudhury and Pradip Ghosh under my guidance during the session of 2023-2024 and accepted in partial fulfillment of the requirement for the degree of B.Tech in Information Technology.

STCET-IT

Signature
Dr. Ranjit Ghoshal
Head, Department of
Information Technology

Signature
Dr. Arijit Ghosal
Mentor, Department of
Information Technology

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them. We respect and thank Dr. Arijit Ghosal, for providing us an opportunity to do the project work and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to him for providing such nice support and guidance, although he had a busy schedule managing the corporate affairs. We owe our deep gratitude to our project mentor Dr. Arijit Ghosal, who took keen interest in our project work and guided us all along, till the completion of our project work by providing all the necessary information. We heartily thank our internal project mentor, Dr. Arijit Ghosal, Associate Professor, Information Technology, for his guidance and suggestions during this project work. We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of Information Technology which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

Deep Kumar Goenka

Aishi Paul

Abhishek Chaudhury

Pradip Ghosh

Vision & Mission

(St. Thomas' College of Engineering & Technology)

Vision

To evolve itself into an industry-oriented research based recognized hub of creative solutions in various fields of engineering by establishing progressive teaching-learning process with an ultimate objective of meeting technological challenges faced by the nation and the society.

Mission

- To create opportunities for students and faculty members in acquiring professional knowledge and developing social attitudes with ethical and moral values.
- To enhance the quality of engineering education through accessible, comprehensive, industry and research-oriented teaching-learning process.
- To satisfy the ever-changing needs of the nation for evolution and absorption of sustainable and environment friendly technologies.

Vision & Mission

(Department of Information Technology)

Vision

To promote the advancement of learning in Information Technology through research oriented dissemination of knowledge which will lead to innovative applications of information in Industry and Society.

Mission

- To incubate students grow into industry ready professionals, proficient research scholars and enterprising entrepreneurs.
- To create a learner- centric environment that motivates the students in adopting emerging technologies of the rapidly changing information society.
- To promote social, environmental, and technological responsiveness among the members of the faculty and students.

Program Educational Objectives (PEO)

Graduates of Information Technology Program shall:

PEO1: Exhibit the skills and knowledge required to design, develop and implement IT solutions for real life problems.

PEO2: Excel in professional career, higher education, and research.

PEO3: Demonstrate professionalism, entrepreneurship, ethical behavior, communication skills and collaborative team work to adapt to the emerging trends by engaging in lifelong learning.

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
7. **Environment and sustainability**: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
9. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Project Mapping with Program Outcomes

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
3	3	3	3	2	3	1	2	3	2	2	3

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)**

Justification:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/ Development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct Investigations of complex problem: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practices.

9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary setting

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Lifelong learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1 (Professional Competency): Apply their knowledge in the field of information technology and contribute significantly to the corporate world by way of providing appropriate solutions to engineering problems and establish their skills in high performance computing, software engineering, programming and thrust areas like security and machine intelligence.

PSO2 (Academic Aptitude): Demonstrate their proficiency in analytical and critical thinking, methodologies of practical design, data analysis and interpretation through their technical expertise which will help them to excel in higher studies within the country and abroad.

Project Mapping with Program Specific Outcomes

PSO1	PSO2
3	3

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)**

Justification:

PSO1: Leveraging machine learning techniques for solution development demonstrates adeptness in high performance computing and software engineering, while integrating security measures underscores readiness for real-world IT challenges.

PSO2: Emphasizing practical design and thorough data analysis showcases analytical prowess, fostering a culture of continuous learning and adaptability essential for success in higher studies both domestically and abroad.

INDEX

Topic	Page No
Introduction	**1**
Chapter 1: Overview	**2**
1.1 Problem statement…….....	2
1.2 Problem definition…….....	2
1.3 Objective……..……...	2
1.4 Tools and Platform…….……	4
1.5 Brief Discussion on Problem…….....	4
Chapter 2: Concepts and Problem Analysis	**7**
2.1 Audio Features…….……...	9
2.2 Feature Selection…….……...	57
2.3 Classifiers……...	65
2.4 Literature Survey……..….....	79
2.5 Audio Features Used…….....	89
Chapter 3: Design and Methodology	**92**
3.1 Dataset…….……...	92
3.2 Feature Extraction…….....	98
3.3 Feature Selection……	105
3.4 Classification…….....	133
3.5 Gantt Chart……	136
Chapter 4: Sample Code	**138**
Chapter 5: Testing, Results, Comparative Study	**147**
5.1.1Testing Data (Method – 1)……...……..……...	147
5.2.1 Experiment Result (Method – 1)..…….....	249
5.1.2 Testing Data (Method – 2)…….....	259
5.2.2 Experiment Result (Method – 2)..……...	260
5.3 Comparative Study……	264
Chapter 6: Future work and Conclusion	**265**
6.1 Scope for Future Improvement……...	265
6.2 Conclusion…….....	265
Annexure: Reference	**267**

Introduction

Human emotions are intricate phenomena, arising from a blend of psychological states and physiological changes within individuals. These emotions, ranging from positive to negative, are pivotal components of human communication and experience, influencing mood, personality, and overall well-being. While the spectrum of human emotions is broad, there are five fundamental ones that hold significant importance: Anger, Disgust, Fear, Happiness, and Sadness.

Emotion recognition, a burgeoning field, involves discerning the underlying emotions conveyed through speech signals. With the proliferation of digital communication platforms, audio data has become a rich source for understanding human emotions. Speech recognition technologies, powered by pattern recognition algorithms, enable automated analysis of emotional content embedded within speech data.

In recent years, there has been remarkable progress in pattern recognition, catalyzing the development of emotion recognition models applied across diverse domains. Researchers from interdisciplinary fields, including neuroscience, psychology, linguistics, and computer science, are increasingly interested in leveraging these advancements to delve deeper into human emotion analysis.

However, analyzing emotional content from speech presents notable challenges. Emotions are inherently subjective and can vary widely among individuals. To tackle these challenges, multiple datasets such as RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D have been utilized to capture a broad spectrum of emotional expressions. These datasets provide a foundation for training robust emotion recognition models capable of accurately classifying emotions in speech.

In this project, the focus lies solely on emotion recognition from speech data. The aim is to classify speech samples into five primary emotional categories: Angry, Disgust, Fearful, Happy, and Sad.

In conclusion, unraveling the emotional content embedded within speech data is a challenging yet promising endeavor. The ability to accurately identify and classify emotions in speech holds immense potential for various applications, including sentiment analysis, human-computer interaction, and mental health assessment, thereby enriching user experiences and fostering deeper insights into human emotions.

Chapter 1: Overview

1.1 Problem Statement

Developing a robust machine learning model for accurately categorizing diverse emotional expressions in audio data, known as Emo–Track.

1.2 Problem Definition

The problem involves analyzing audio data to detect emotional states in individuals for applications in Smart Healthcare and Call Centre's. Each audio recording contains unique characteristics that can be used to classify different emotional states. The goal is to develop a classifier that achieves high accuracy with a reduced feature set, minimizing computational complexity. This approach can enhance customer service in call centers' by understanding callers' emotions and aid in diagnosing and treating patients in Smart Healthcare based on their emotional states. To achieve this, a combination of signal processing techniques, machine learning algorithms, and feature selection methods will be employed. The classifier will be trained on a diverse dataset encompassing various emotional expressions to ensure robustness and generalization. Additionally, real-time processing capabilities will be emphasized to provide instantaneous feedback in both healthcare and call center environments, enabling prompt and tailored responses to emotional cues. By leveraging advanced technologies, such as deep learning and natural language processing, this system can adapt to individual nuances in speech patterns and emotional expressions, thereby enhancing the overall user experience and effectiveness of the applications.

1.3 Objective

In today's dynamic world, the absence of robust Speech Emotion Recognition systems poses myriad challenges across diverse domains. From misinterpretations of customer sentiments during service interactions to the limited capacity to provide timely support for individuals grappling with mental health concerns, the inability to accurately discern emotional cues from speech significantly impedes effective communication and intervention strategies.

To confront these challenges head-on, harnessing the power of Speech Emotion Recognition technology becomes imperative. By leveraging Emotion Recognition capabilities, we can equip systems with the

capacity to adeptly analyze and respond to the subtle nuances of emotional expression conveyed through speech. This empowerment enables us to not only elevate customer service experiences through more empathetic interactions but also to offer personalized and tailored mental health interventions, thereby fostering improved well-being and support systems across various contexts.

The overarching objective of this project is to develop a robust Emo–Track, a Sentiment Analysis system capable of accurately analyzing and responding to the multifaceted emotional cues embedded within speech signals. Through the utilization of advanced machine learning techniques, our aim is to construct a model that can discern even the most nuanced emotional states, thereby facilitating the provision of personalized support and interventions tailored to individual emotional needs.

Beyond its immediate applications, Emo–Track holds promise for advancing research in interdisciplinary fields such as psychology, linguistics, and human-computer interaction. By delving into large datasets of speech annotated with emotional markers, researchers can gain deeper insights into the intricate mechanisms underlying emotional expression and perception, shedding invaluable light on fundamental aspects of human communication and cognition.

Moreover, the development of Sentiment Analysis systems boasting high accuracy and reliability opens up unprecedented avenues for exploring the profound role emotions play in decision-making processes, social interactions, and mental health outcomes. This, in turn, paves the way for the implementation of innovative interventions and therapies aimed at fostering emotional well-being and resilience.

As the field of Sentiment Analysis through Speech continues to evolve, its interdisciplinary impact transcends merely shaping how we interact with technology; it fundamentally shapes our understanding and navigation of the intricate tapestry of human emotion and behavior.

Furthermore, the applications of Emo–Track extend far beyond addressing immediate challenges, promising to revolutionize various facets of our daily lives. From enhancing human-computer interaction experiences to optimizing educational settings by gauging student engagement levels, the integration of Speech Emotion Recognition has the potential to usher in a new era of more empathetic, nuanced, and effective communication strategies, ultimately fostering greater understanding and connectivity within our society.

1.4 Tools and Platform

The tools, languages and platforms used for the development of Emo–Track are as follows:

The entire project was carried out using MATLAB R2021b, PYTHON 3.10 and Jupyter Notebook.

- Feature Extraction is done using MATLAB software.
- Classification is done using PYTHON.
- Python libraries:
    - Pandas 1.4.3
    - Numpy 1.22.4
    - Scikit-learn 1.1.2
    - Scipy 1.9.0
    - Matplotlib 3.5.3
    - Seaborn 0.11.2
    - Joblib 1.1.0
- Data for the project was sourced from Kaggle.com.

1.5 Brief Discussion on Problem

Sentiment Analysis through Speech has garnered significant attention in recent years, driven by the growing recognition of its potential applications across various domains, including human-computer interaction, mental health interventions, and affective computing. Researchers and practitioners alike have dedicated considerable efforts to unraveling the complexities of emotional expression conveyed through speech signals, aiming to develop robust and accurate Speech Emotion Recognition systems capable of understanding and responding to human emotions effectively.

At the heart of Emotion Recognition lies the recognition of emotional cues embedded within speech, a task that involves parsing through the acoustic features and prosodic elements of spoken utterances to discern underlying emotional states. Fundamental to this endeavor is the understanding that human emotions are multifaceted and dynamic, encompassing a spectrum of states ranging from joy and sadness to anger and fear. Researchers have explored various methodologies and techniques to capture and classify these emotional nuances, drawing inspiration from disciplines such as psychology, linguistics, and signal processing.

One of the key challenges in Speech Emotion Recognition research lies in selecting appropriate datasets for model training and evaluation. Datasets play a crucial role in shaping the performance and generalization capabilities of Emotion Recognition models, as they provide the foundational examples upon which the models learn to recognize and classify emotional states. Researchers must carefully curate datasets that encompass a diverse range of emotional expressions, ensuring that the resulting models are robust and capable of handling real-world variability.

Once a suitable dataset has been identified, the process of model development begins in earnest. Researchers leverage a variety of machine learning algorithms and techniques to construct Emotion Recognition models capable of accurately discerning emotional cues from speech signals. Supervised learning approaches are commonly employed, where models are trained on labeled datasets associating specific emotional states with corresponding acoustic features.

Feature extraction serves as a critical step in the model development pipeline, as it involves extracting relevant acoustic features from speech signals that capture the underlying emotional content. These features may include spectral features such as Mel-Frequency Cepstral Coefficients (MFCCs), Mel Spectrogram, Chromagram as well as prosodic features derived from the frequency content of the speech signal. Researchers employ signal processing techniques and domain knowledge to extract informative features that are discriminative of different emotional states.

Once the features have been extracted, researchers employ various machine learning algorithms to train the models. Popular algorithms include Naive Bayes, Support Vector Machine, Random Forests, Multi-layer Perceptron (MLP), Logistic Regression. These algorithms learn to map the extracted features to corresponding emotional labels through the process of optimization, iteratively adjusting model parameters to minimize prediction errors.

As Sentiment Analysis models mature, researchers explore avenues for enhancing their performance and robustness. Ensemble learning techniques, where multiple models are combined to make predictions, offer potential improvements in accuracy and reliability. Additionally, researchers investigate the integration of contextual information, such as speaker characteristics and linguistic context, to improve the model's understanding of emotional expression in speech.

The evaluation of Sentiment Analysis models is a multifaceted process that involves assessing various performance metrics, including accuracy, precision, recall, and F1-score. Researchers carefully evaluate the model's ability to correctly classify emotional states across different classes and assess its performance

under varying conditions, such as noise and speaker variability. Additionally, qualitative evaluations may involve subjective assessments by human annotators to gauge the model's perceptual fidelity.

Continued advancements in Speech Emotion Recognition technology are paving the way for innovative applications in diverse domains. For instance, in the realm of human-computer interaction, Speech Emotion Recognition can enable more intuitive and empathetic interactions between users and artificial intelligence systems. By imbuing virtual assistants and chatbots with the ability to recognize and respond to users' emotional states, it can enhance the overall user experience and foster deeper engagement. Moreover, in educational settings, it can play a pivotal role in assessing student engagement and emotional well-being during remote learning sessions. By analyzing students' speech patterns and emotional cues, educators can adapt their teaching strategies to better support individual learning needs and foster a more inclusive learning environment.

Beyond its applications in technology-mediated interactions, Speech Emotion Recognition holds considerable promise for advancing mental health interventions and therapies. By analyzing patients' speech patterns and vocal cues, these systems can assist clinicians in monitoring mental health conditions such as depression, anxiety, and post-traumatic stress disorder. Early detection of emotional distress through speech analysis can facilitate timely interventions and personalized treatment plans, ultimately improving patient outcomes and well-being. Additionally, Speech Emotion Recognition can augment existing therapeutic approaches by providing real-time feedback and support to individuals undergoing cognitive-behavioral therapy or counselling sessions.

In summary, Sentiment Analysis through Speech is a multidisciplinary field drawing from psychology, linguistics, signal processing, and machine learning. Researchers develop Sentiment Analysis systems through careful dataset curation, feature extraction, and model training to accurately capture emotional cues from speech. As Emotion Recognition technology evolves, its applications in human-computer interaction and mental health interventions offer promise for enhancing our understanding of human emotions and improving human interactions.

Chapter 2: Concepts and Problem Analysis

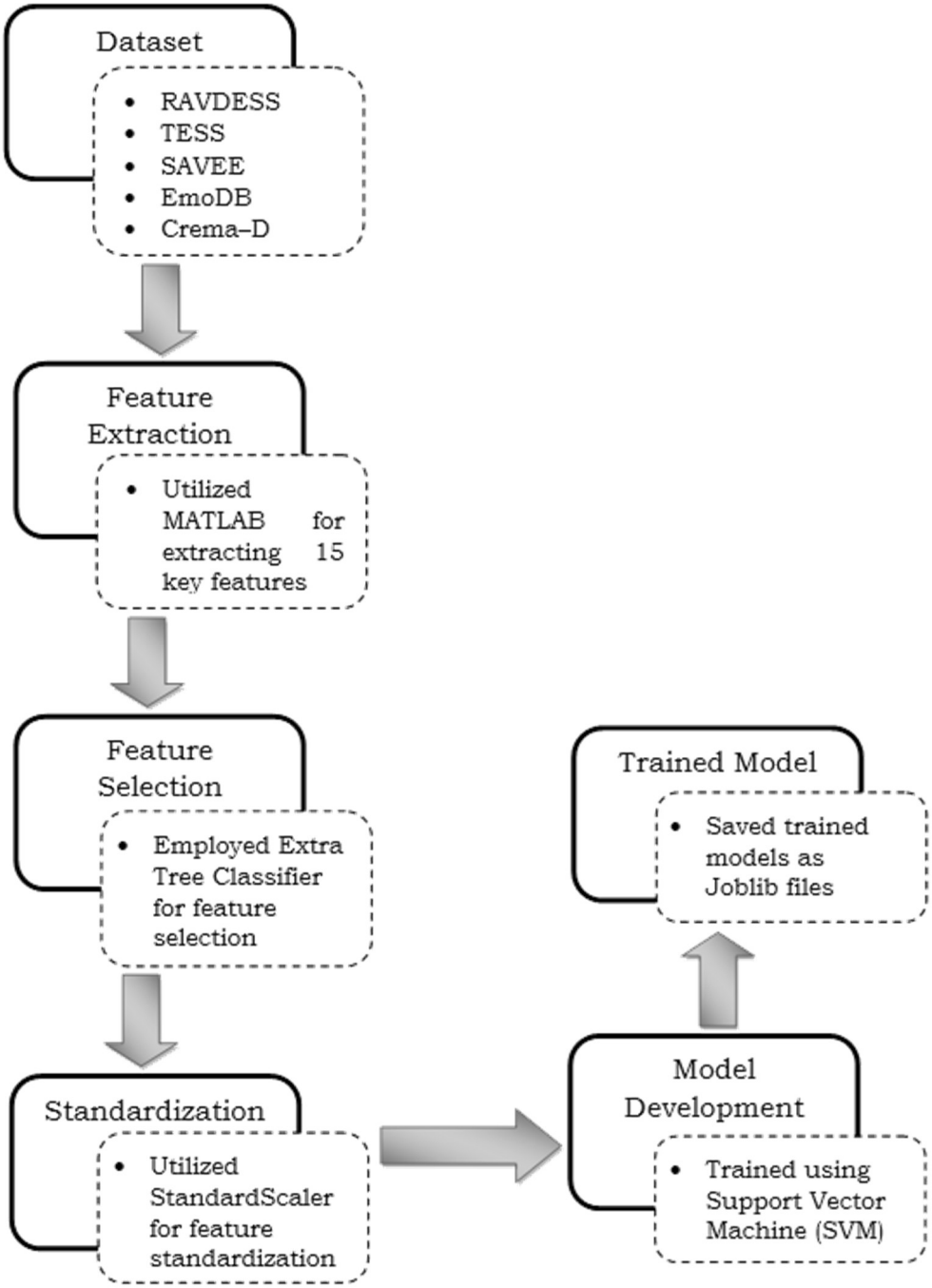

Fig.1. Flow Chart of the working of system

The flowchart shown in Fig 1 outlines the process of building an emotion detection system using machine learning techniques. Here's a detailed breakdown of each step and its significance:

1. Dataset: The starting point involves leveraging the power of publicly available datasets. The flowchart references several established datasets for emotion recognition, including RAVDESS, TESS, SAVEE, EmoDB, and Crema-D. These datasets provide a rich collection of audio recordings with corresponding emotional labels. This variety is crucial because it allows the model to learn and recognize emotions from diverse voices and speaking styles.

2. Feature Extraction: Raw audio data isn't directly usable by machine learning models. This step bridges the gap by extracting key features from the audio recordings. The flowchart specifies using MATLAB, a powerful software for scientific computing, to perform this extraction. The process focuses on extracting 15 specific features that are most informative for identifying emotions in speech.

3. Feature Selection: The flowchart incorporates an Extra Tree Classifier to address this. This classifier analyzes the extracted features and identifies the most significant ones that contribute most effectively to distinguishing between different emotions. This selection process helps to refine the data used for model training, potentially leading to a more accurate and efficient emotion detection system.

4. Standardization: Features with larger scales would have a greater influence on the model's learning process, even if they might not be inherently more informative for emotion detection. Standardization tackles this issue. The flowchart highlights the use of StandardScaler, a technique that normalizes all features to a common scale. This ensures that each feature contributes equally during model training, leading to a more robust emotion detection system.

5. Model Development: Finally, the flowchart showcases the development of the emotion detection model itself. The chosen approach utilizes a Support Vector Machine (SVM) classifier. Here, the SVM is trained on the preprocessed data. The flowchart also mentions saving the trained model as a Joblib file. This allows for easy deployment and reuse of the emotion detection system in practical applications.

By combining these steps, the flowchart effectively outlines a machine learning approach to building an emotion detection system. The workflow leverages pre-existing, diverse datasets, extracts informative features from audio data, performs feature selection and standardization for optimal training, and utilizes a well-established classification algorithm to create a robust emotion detection model.

2.1 Audio Features

In this project, we employ feature extraction as the initial phase. Each audio recording possesses its unique set of distinguishing characteristics that enable distinct identification. By analyzing specific features, we can deduce the underlying emotions and draw conclusive results. Careful consideration is given to selecting a feature set that yields higher accuracy. A desirable feature should exhibit significant differences between different emotion classes while minimizing variations within the same class to enhance discrimination. Features that are selected for our work are Mel-Frequency Cepstral Coefficients (MFCC) [1], Mel Spectrogram [2], Spectral Flux [3], Spectral Skewness [4], Spectral Slope [5], Spectral Entropy [6], Spectral Rolloff [7], Chromagram [8], Linear Predictive Coding(LPC) [9], Zero Crossing Rate(ZCR) [10], Energy [11], Pitch [12], Intensity [13], Harmonic-to-Noise Ratio(HNR) [14], Root Mean Square(RMS) [15].The feature set utilized in our dataset comprises the following components:

2.1.1 Mel-Frequency Cepstral Coefficients (MFCCs)

Mel-Frequency Cepstral Coefficients (MFCCs) are widely used in speech and audio signal processing for tasks like speech recognition, speaker identification, and audio classification. They aim to capture the characteristics of the human auditory system more accurately than traditional spectral representations like the Fourier Transform.

Here's a detailed explanation along with the mathematical formulation:

1. Pre-emphasis:

Speech signals typically have more energy in lower frequencies. To balance the spectrum, a pre-emphasis filter is applied. This filter amplifies the higher frequencies and attenuates the lower ones. The pre-emphasis operation shown in Eq. 1 is usually done by passing the signal through a first-order high-pass filter, which can be expressed as:

$$y[n] = x[n] - \alpha \cdot x[n-1]$$

Eq. 1: Pre-emphasis formula

2. Framing:

The speech signal is divided into short frames using a windowing technique, typically with a length of 20-40 milliseconds and overlapping frames. Each frame is then processed separately.

3. Windowing:

Each frame is multiplied by a window function like Hamming, Hanning, or Gaussian to minimize spectral leakage.

4. Fast Fourier Transform (FFT):

The power spectrum of each frame is computed using the FFT. This provides the frequency content of the signal over time.

5. Mel Filter Bank:

The Mel scale is a perceptual scale of pitches. Instead of using a linear frequency scale, the Mel scale approximates the human auditory system's response to different frequencies. The Mel filter bank is a set of triangular filters spaced evenly on the Mel scale. Eq. 2 shows the formula to convert Hertz to Mel is given by:

$$\mathrm{M(f)} = 2595.\log_{10}\left(1 + \frac{f}{700}\right)$$

$$f(M) = 700.(10^{\frac{M}{2595}} - 1)$$

Eq. 2: Mel filter bank formula

Each filter in the Mel filter bank is defined by its center frequency and extends to adjacent frequencies. The output of each filter is the energy in that frequency band.

6. Logarithmic Compression:

After computing the filter bank energies, a logarithmic operation is applied to compress the values. This mimics the logarithmic response of the human ear to sound intensity.

7. Discrete Cosine Transform (DCT):

Finally, a Discrete Cosine Transform is applied to the log filter bank energies. The DCT reduces the dimensionality of the feature vector and decorrelates the filter bank energies. Only lower DCT coefficients are retained as they contain most of the relevant information about the spectral envelope of the signal.

Mathematically, the MFCC computation for a given frame x[n] can be summarizes as shown in Eq. 3:

$$w[n] = Windowing(y[n])$$

$$X(f) = FFT(w[n])$$

$$H_m(f) = MelFilter(f)$$

$$S_m = \sum_{f=0}^{N-1} |X(f)|^2 . H_m(f)$$

$$C_n = DCT(\log(S_m))$$

Eq. 3: DCT Formula

Where

x[n] is the input speech signal.

α is the pre-emphasis coefficient.

w [n] is the windowed frame.

X(f) is the Fourier Transform of the frame.

$H_m(f)$ is the response of the mth Mel filter.

S_m is the energy in the mth Mel frequency band.

C_n are the resulting Mel-Frequency Cepstral Coefficients

By using MFCCs, the complex spectral information of speech signals is compactly represented in a way that is more aligned with human perception, making them particularly effective for speech processing tasks.

Delta MFCC:

Delta Mel-Frequency Cepstral Coefficients (Delta MFCCs) are extensions of standard MFCCs that capture how MFCCs change over time. They're computed by taking the temporal derivatives of MFCC sequences. These coefficients provide information about the dynamic evolution of spectral features in speech and audio signals. They're useful in tasks where temporal dynamics matter, like speaker recognition and emotion detection.

Delta Delta MFCC:

Delta-Delta Mel-Frequency Cepstral Coefficients (Delta-Delta MFCCs) are the second-order derivatives of the MFCC sequence. They capture the rate of change of the Delta MFCCs over time, providing even more detailed temporal information about speech and audio signals.

2.1.2 Mel Spectrogram

A Mel spectrogram, also known as a Mel-frequency spectrogram, is a type of spectrogram where the frequency axis is scaled according to the Mel scale, which approximates the human auditory system's response to different frequencies. It's commonly used in speech and audio signal processing to visualize and analyze the frequency content of sound in a way that's more aligned with human perception.

Here's how a Mel spectrogram is computed:

1. Frame the Signal: The input audio signal is divided into short frames using a windowing technique. Each frame typically overlaps with adjacent frames to ensure continuity in the spectrogram.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain. This results in a spectrum for each frame, representing the magnitude of different frequencies present in that frame.

3. Mel Filter Bank: Instead of directly plotting the spectrum, the spectrum is passed through a Mel filter bank. This bank consists of triangular filters that are spaced evenly on the Mel scale. Each filter's center frequency is determined by its position on the Mel scale, and the filters are designed to approximate the frequency response of the human auditory system.

4. Filtering and Summation: The spectrum is filtered by each Mel filter in the bank. For each filter, the magnitude of the spectrum is weighted by the filter's response. The filtered outputs are then summed to obtain the energy in each Mel frequency band.

5. Logarithmic Compression: The energy values obtained from the Mel filter bank are typically transformed using a logarithmic function. This compression operation mimics the logarithmic response of the human ear to sound intensity, making the spectrogram more perceptually relevant.

6. Plotting: Finally, the log-compressed energy values are plotted over time to create the Mel spectrogram. The x-axis represents time, and the y-axis represents frequency, with the frequency axis scaled according to the Mel scale.

By using a Mel spectrogram, the frequency content of audio signals is represented in a way that better aligns with human perception, making it easier to analyze and interpret audio features for tasks like speech recognition, speaker identification, and music analysis.

2.1.3 Spectral Flux

Spectral flux is a measure used in audio signal processing to quantify the change in spectral content between consecutive frames of a signal. It provides information about how rapidly the spectral characteristics of the signal are changing over time. Spectral flux is particularly useful in tasks like audio onset detection, music segmentation, and speech activity detection. Here's how spectral flux is computed:

1. Frame the Signal: Similar to other audio analysis techniques, the input audio signal is divided into short frames using a windowing technique, with each frame typically overlapping with adjacent frames.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain. This results in a spectrum for each frame, representing the magnitude of different frequencies present in that frame.

3. Calculate Spectral Flux: Spectral flux is computed as the Euclidean distance between the spectral vectors of consecutive frames. Mathematically, it can be expressed as shown in Eq. 4:

$$\text{Spectral Flux(t)} = \sqrt{\sum_{f=1}^{F} (|X(t,f)| - |X(t-1,f)|)^2}$$

Eq. 4: Spectral Flux

Where:

Spectral Flux(t) is the spectral flux at time t.

|X(t, f)| and |X(t - 1, f)| are the magnitudes of frequency bin f in frames t and t - 1 respectively.

F is the number of frequency bins.

4. Smoothing: To reduce noise and variability, spectral flux values are often smoothed using techniques like moving average or median filtering.

5. Thresholding: Depending on the application, a threshold may be applied to the spectral flux values to identify significant changes in spectral content. This thresholding step helps in detecting relevant events like onsets in music or speech activity.

Spectral flux provides a useful feature for characterizing the temporal dynamics of audio signals. High spectral flux values indicate significant changes in spectral content, which can correspond to events like the onset of a musical note, the beginning of a speech segment, or the transition between different sections in a piece of music.

2.1.4 Spectral Skewness

Spectral skewness is a measure used in audio signal processing to quantify the asymmetry of the spectral distribution of a signal. It provides information about the shape of the spectral energy distribution, indicating whether the energy is concentrated more towards higher or lower frequencies. Here's how spectral skewness is computed:

1. Frame the Signal: As with other audio analysis techniques, the input audio signal is divided into short frames using a windowing technique, typically with overlapping frames.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain, resulting in a spectrum for each frame.

3. Calculate Spectral Centroid: The spectral centroid, which represents the center of mass of the spectral energy distribution, is computed for each frame. It is defined as the weighted mean of the frequency bins' magnitudes and is shown in Eq. 5:

$$Spectral\ Centroid(t) = \frac{\sum_{f=1}^{F} f.|X(t,f)|}{\sum_{f=1}^{F} |X(t,f)|}$$

Eq. 5: Spectral Centroid

Where:

Spectral Centroid(t) is the spectral centroid at time t.

|X(t, f)| is the magnitude of frequency bin f in frame t.

F is the number of frequency bins.

4. Calculate Spectral Skewness: Spectral skewness is then computed based on the spectral centroid. It quantifies the asymmetry of the spectral distribution around the centroid. A positive skewness indicates that the spectral energy is skewed towards higher frequencies, while a negative skewness indicates skewness towards lower frequencies. Mathematically, it can be expressed as shown in Eq. 6:

$$Spectral\ Skewness(t) = \frac{\sum_{f=1}^{F} (f - Spectral\ Centroid(t))^3.|X(t,f)|}{\sum_{f=1}^{F} |X(t,f|)^{\frac{3}{2}}}$$

Eq. 6: Spectral Skewness

Spectral skewness provides information about the spectral shape of audio signals. It is a useful feature for tasks such as sound classification, musical instrument recognition, and emotion detection in speech.

2.1.5 Spectral Slope

Spectral slope is a measure used in audio signal processing to characterize the rate of change of spectral energy across different frequency bands. It provides information about how quickly the energy of a signal decreases or increases as the frequency increases. Here's how Spectral Slope is computed:

1. Frame the Signal: The input audio signal is divided into short frames using a windowing technique, similar to other audio analysis methods.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain, resulting in a spectrum for each frame.

3. Divide Frequency Bands: The frequency spectrum is divided into several frequency bands or bins. These bands can be evenly spaced or based on psychoacoustic principles, such as the Bark or Mel scales.

4. Compute Spectral Energy: The energy of each frequency band is computed by summing the magnitudes of the frequency bins within that band.

5. Calculate Spectral Slope: Spectral slope is then calculated by fitting a linear regression line to the logarithm of the spectral energy across the frequency bands. The slope of this line represents the rate of change of spectral energy with respect to frequency.

Mathematically, spectral slope can be expressed as shown in Eq. 7:

$$Spectral\ Slope = \frac{\sum_{i=1}^{N}(f_i - \bar{f}).\log(E_i)}{\sum_{i=1}^{N}(f_i - \bar{f})^2}$$

Eq. 7: Spectral Slope

Where:

Spectral Slope is the slope of the linear regression line.

f_i is the center frequency of the ith frequency band.

$\bar{f}$ is the mean frequency.

E_i is the energy of the ith frequency band.

N is the number of frequency bands

Spectral slope provides valuable information about the spectral shape and tonal characteristics of audio signals.

2.1.6 Spectral Entropy

Spectral entropy is a measure used in audio signal processing to quantify the distribution of spectral energy across different frequency bands. It provides information about the complexity or randomness of the spectral content of an audio signal. Here's how Spectral Entropy is computed:

1. Frame the Signal: As with other audio analysis techniques, the input audio signal is divided into short frames using a windowing technique, typically with overlapping frames.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain, resulting in a spectrum for each frame.

3. Compute Spectral Energy Distribution: The energy of the signal within specific frequency bands is calculated by summing the squared magnitudes of the frequency components within each band, as explained in the computation of spectral energy.

4. Normalize Spectral Energy: The spectral energy distribution is normalized by dividing each band's energy by the total energy across all bands. This step ensures that the resulting entropy measure is independent of the overall energy level of the signal.

5. Compute Spectral Entropy: Spectral entropy is then calculated using the normalized energy distribution. It is typically computed using Shannon entropy, which measures the uncertainty or randomness in a probability distribution. Mathematically, spectral entropy H can be expressed as shown in Eq. 8:

$$H = -\sum_{i=1}^{N} p_i . \log_2(p_i)$$

Eq. 8: Spectral Entropy

Where:

p_i is the normalized energy of the ith frequency band.

N is the total number of frequency bands.

Spectral entropy provides valuable information about the diversity or uniformity of spectral energy distribution across frequency bands. A higher spectral entropy value indicates a more uniform energy distribution across bands, suggesting greater complexity or randomness in the spectral content of the signal. Conversely, a lower spectral entropy value indicates a more concentrated energy distribution, suggesting a more structured or tonal signal.

2.1.7 Spectral Roll-off

Spectral roll-off is a measure used in audio signal processing to characterize the shape of the frequency spectrum of an audio signal. It indicates the frequency below which a certain percentage of the total spectral energy is contained. Here's how spectral roll-off is computed:

1. Frame the Signal: The input audio signal is divided into short frames using a windowing technique, typically with overlapping frames.

2. Apply Fourier Transform: The Fourier Transform is applied to each frame to convert the signal from the time domain to the frequency domain, resulting in a frequency spectrum for each frame.

3. Calculate Total Spectral Energy: The total spectral energy of each frame is computed by summing the squared magnitudes of all frequency components in the spectrum.

4. Define a Threshold: A threshold is set as a certain percentage (commonly 85-95%) of the total spectral energy. This threshold represents the percentage of energy below which the spectrum roll-off is determined.

5. Determine Spectral Roll-off Frequency: Spectral roll-off frequency is then determined as the frequency below which the cumulative spectral energy exceeds the defined threshold. In other words, it's the frequency at which a certain percentage of the total energy is contained.

Mathematically, spectral roll-off frequency $f_{roll-off}$ can be expressed as shown in Eq. 9:

$$\sum_{f=0}^{f_{roll-off}} |X(f)|^2 = \text{Threshold} \times \sum_{f=0}^{F} |X(f)|^2$$

Eq. 9: Spectral Roll-off

Where:

$f_{roll-off}$ is the spectral roll-off frequency.

|X(f)| is the magnitude of the frequency component at frequency f

F is the highest frequency bin in the spectrum

Spectral roll-off provides information about the shape of the frequency spectrum and can be used to characterize the high-frequency content of audio signals. A higher spectral roll-off frequency indicates that most of the spectral energy is concentrated at lower frequencies, while a lower roll-off frequency suggests a more evenly distributed energy across the frequency spectrum.

2.1.8 Chromagram

A Chromagram is a condensed view of the energy distribution of musical notes in an audio signal over time. It summarizes the presence of different pitch classes, regardless of octave, and is useful for tasks like chord recognition and melody extraction in music analysis. Here's how Chromagram is computed:

1. Frame the Signal: Divide the input audio signal into short frames (x[n]), typically with a length of

N samples and overlapping frames.

2. Apply Short-Time Fourier Transform (STFT): Compute the Short-Time Fourier Transform (STFT) of each frame (x[n]) to obtain the magnitude spectrogram |X(k, w)|, where k represents the frame index and w represents the frequency bin index.

3. Mapping to Chroma: Map the frequency bins w of the spectrogram to the 12 pitch classes using a chroma matrix C. The chroma matrix has dimensions 12 x F, where F is the number of frequency bins in the spectrogram. Each element C[i, w] represents the mapping weight from frequency bin w to pitch class i. This mapping can be done using heuristics or predefined frequency-to-chroma mappings.

4. Compute Chroma Features: Compute the chroma features [i, k] for each frame k using the following formula shown in Eq. 10:

$$chroma[i,k] = \sum_{\omega} |X(k,w)|.C[i,\omega]$$

Eq. 10: Chroma features

where |X (k,w)| is the magnitude of the spectrogram at frequency bin w and frame k, and C[i, w] is the mapping weight from frequency bin w to pitch class i.

5. Normalize Chroma: Optionally, normalize the chroma features chroma[i, k] for each frame k to have a maximum value of 1.

The resulting chromagram is a 2D matrix where the rows represent the 12 pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, B) and the columns represent the frames of the audio signal. Each cell in the chromagram represents the energy or intensity of a particular pitch class at a specific time frame.

2.1.9 Linear Predictive Coding (LPC)

Linear Predictive Coding (LPC) is a method used in speech processing to model the spectral envelope of speech signals. It represents the speech signal as the output of a linear filter driven by a time-varying input.

Let's break down the mathematical expression for LPC:

1. Frame the Signal: Divide the input speech signal into short frames of length *N* samples.

2. Windowing: Multiply each frame by a window function to reduce spectral leakage.

3. Autocorrelation: Compute the autocorrelation sequence $R(\tau)$ for each frame, which represents the correlation between the frame and delayed versions of itself. It's defined as shown in Eq. 11:

$$R(\tau) = \sum_{n=0}^{N-1} x[n] \,.\, x[n-\tau]$$

Eq. 11: Autocorrelation sequence

4. Levinson-Durbin Recursion: Use the Levinson-Durbin recursion algorithm to solve the Toeplitz system of equations formed by the autocorrelation sequence. This yields the Linear Prediction Coefficients (LPCs) a_k. The recursion can be written as shown in Eq. 12:

$$a_k = -\frac{1}{R(0)}\left(\sum_{j=1}^{k-1} a_j \,.\, R(k-j) + R(k)\right)$$

$$E[k] = \left(1 - \sum_{i-1}^{k} a_j \,.\, R(k-j)\right) . R(0)$$

Eq. 12: Levinson-Durbin Recursion

5. LPC Spectrum: Compute the LPC spectrum $H(e^{jw})$ using the LPC coefficients a_k. It's given in Eq. 13:

$$H(e^{jw}) = \frac{1}{1 + \sum_{k=1}^{p} a_k . e^{-jwk}}$$

Eq. 13: LPC Spectrum

where p is the LPC order.

The LPC spectrum represents the frequency response of the linear filter that models the speech signal's spectral envelope. It's often used in speech analysis and synthesis applications, such as speech compression, voice conversion, and formant estimation.

2.1.10 Zero Crossing rate (ZCR)

Zero crossing rate (ZCR) is a feature used in speech and audio signal processing to quantify the rate at which a signal changes its sign. It provides information about the temporal variations and the frequency content of the signal.

Here's how zero crossing rate is computed along with a mathematical expression:

1. Frame the Signal: Divide the input audio signal into short frames of length *N* samples.

2. Count Zero Crossings: For each frame, count the number of times the signal crosses the zero-amplitude level. A zero crossing occurs when the signal changes its sign from positive to negative or vice versa.

3. Calculate Zero Crossing Rate: Compute the zero-crossing rate ZCR as the ratio of the number of zero crossings to the total number of samples in the frame.

Mathematically, zero crossing rate can be expressed as shown in Eq. 14:

$$\text{ZCR} = \frac{1}{N-1}\sum_{n=1}^{N-1}|sign(x[n]) - sign(x[n-1])|$$

Eq. 14: Zero Crossing Rate

Where:

- x[n] represent the audio signal samples.
- Sign(x[n]) is the sign function, which returns -1 if x[n] is negative, 0 if x[n] is zero,and 1 if x[n] is positive.
- N is the length of the frame.

The expression |sign (x[n]) - sign (x[n-1])| evaluates to 1 whenever there's a zero crossing (i.e., the sign of x[n] differs from the sign of x[n-1]), and 0 otherwise. The sum over all samples counts the total number of zero crossings in the frame. Dividing by N-1 normalizes this count to obtain the zero-crossing rate.

A higher zero crossing rate typically indicates a more dynamic or high-frequency content in the signal, while a lower rate suggests a smoother or low-frequency signal. Zero crossing rate is commonly used in speech and music processing for tasks such as speech recognition, music genre classification, and rhythm analysis.

2.1.11 Energy

Energy refers to the quantification of the total power present within a specific segment or interval of a signal. This measure provides essential insights into the overall strength or magnitude of the signal within that particular timeframe. To compute the energy of a signal segment, the squared values of all the individual samples within that segment are summed together. This process effectively captures both the amplitude and duration of the signal, as it considers the magnitude of each sample and aggregates their contributions over the specified time interval. By quantifying the total power contained within a segment, energy serves as a fundamental metric for characterizing signals and is widely utilized in various signal processing applications for tasks such as feature extraction, noise analysis, and signal classification.

Here's how energy is computed along with a mathematical expression:

1. Frame the Signal: Divide the input signal x[n] into short frames of length *N* samples.

2. Calculate Energy: Compute the energy *E* of each frame by summing the squared values of its samples as shown in Eq. 15:

$$E = \sum_{n=0}^{N-1} |x[n]|^2$$

Eq. 15: Energy

Where:

- x[n] represents the signal samples within the frame.
- |x[n]| denotes the absolute value of the signal sample x[n].
- N is the length of the frame.

The expression $|x[n]|^2$ squares the value of each sample, ensuring that negative values contribute positively to the energy calculation. The sum over all samples accumulates the energy across the entire frame.

Energy is a crucial feature used in various signal processing applications, including speech and audio processing, image processing, and vibration analysis. It provides information about the signal's magnitude or intensity and is often used to detect changes, classify signals, or estimate signal quality.

2.1.12 Pitch

Pitch is a perceptual attribute of sound related to the fundamental frequency of a periodic waveform. It represents the perceived frequency of a sound and is closely associated with the sensation of musical notes and the perceived "height" of a sound in terms of its musical scale.

Here's a simplified explanation of how pitch can be estimated using autocorrelation:

1. Autocorrelation: Compute the autocorrelation function $R(\tau)$ of the signal x[n] , which represents the correlation between the signal and delayed versions of itself. Mathematically, it's defined as shown in Eq. 16:

$$R(\tau) = \sum_{n=0}^{N-1} x[n] \, . \, x[n-\tau]$$

Eq. 16: Autocorrelation

2. Peak Picking: Identify the peaks in the autocorrelation function. The distance between adjacent peaks corresponds to the period of the waveform, which is inversely related to the pitch.

3. Pitch Estimation: Estimate the pitch frequency f_{pitch} the period of the waveform using Eq. 17:

$$f_{pitc} = \frac{1}{\tau}$$

Eq. 17: Pitch Estimation

Where:

- τ is the delay(lag) corresponding to the peak in the autocorrelation function.
- f_{pitc} is the estimated pitch frequency.

This method provides a straightforward way to estimate pitch from a signal's autocorrelation function. However, in practice, more sophisticated algorithms and signal processing techniques are often used to improve accuracy and robustness, especially in noisy or complex audio signals.

Pitch estimation plays a crucial role in various applications, including music analysis, speech processing, and sound synthesis. It allows for the extraction of fundamental frequency information, which is essential for tasks like melody recognition, speech prosody analysis, and voice synthesis.

2.1.13 Intensity

Intensity, in the context of sound, refers to the amount of energy transmitted by sound waves per unit area perpendicular to the direction of wave propagation. It's a measure of the loudness or strength of a sound and is typically quantified in decibels (dB).

Mathematically, intensity *I* is defined as the power per unit area and is expressed as shown in Eq. 18:

$$\mathrm{I} = \frac{P}{A}$$

Eq. 18: Intensity

Where:

- P is the power carried by the sound wave.
- A is the area through which the sound wave passes.

Intensity can also be related to the pressure amplitude p of the sound wave using the following expression in Eq. 19:

$$\mathrm{I} = \frac{p^2}{\rho c}$$

Eq. 19: Intensity related to pressure amplitude

Where:

- ρ is the dencity of the medium through which the sound wave propagates.
- c is the speed of the sound in the medium.

In practical applications, intensity is often expressed in logarithmic scale using decibels (dB), which is defined as shown in Eq. 20:

$$Intensity(dB) = 10.log_{10}\left(\frac{I}{I_0}\right)$$

Eq. 20: Intensity in Decibels

Where I_0 is the reference intensity, typically set to 10^{-12} watts per square meter (the threshold of human hearing).

2.1.14 Harmonic-to-Noise Ratio (HNR)

The Harmonic-to-Noise Ratio (HNR) is a measure used in speech and audio signal processing to quantify the ratio of harmonic components to non-harmonic or noise components in a signal. It provides information about the periodicity and tonality of the signal, making it useful for tasks such as voice quality assessment and voice disorder diagnosis.

Here's how Harmonic-to-Noise Ratio is computed along with a mathematical expression:

1. Autocorrelation Analysis: Compute the autocorrelation function $R(\tau)$ of the signal x(t). The autocorrelation function measures the similarity between the signal and its delayed versions. Mathematically, it's defined as shown in Eq. 21:

$$R(\tau) = \int_{-\infty}^{\infty} x(t).x(t-\tau)dt$$

Eq. 21: Autocorrelation Analysis

2. Harmonic Estimation: This step involves identifying the fundamental frequency (F0) of the signal, typically using pitch estimation techniques. Once the fundamental frequency is estimated, harmonics, which are integer multiples of the fundamental frequency (F0), can be determined. These harmonics represent the frequencies that resonate with the fundamental frequency and contribute to the overall harmonic content of the signal.

3. Harmonic Component Extraction: After identifying the fundamental frequency and its harmonics, the next step is to extract the harmonic components of the signal. This is achieved by selecting frequency bands around each harmonic and summing their energy. By isolating these harmonic components, we can focus on the structured and periodic elements of the signal that correspond to the fundamental frequency and its multiples.

4. Noise Component Extraction: In contrast to the harmonic components, the non-harmonic or noise components of the signal need to be extracted. This is accomplished by subtracting the harmonic components, determined in the previous step, from the original signal. The residual signal obtained after subtracting the harmonic components represents the noise or non-periodic components present in the signal.

5. Compute Harmonic-to-Noise Ratio (HNR): Finally, the Harmonic-to-Noise Ratio (HNR) is calculated as the ratio of the energy of the harmonic components to the energy of the noise components. Mathematically, it can be expressed as the total energy of the harmonic components divided by the total energy of the noise components. This ratio provides valuable insights into the relative dominance of harmonic versus non-

harmonic components in the signal, serving as a measure of signal purity or periodicity. Higher HNR values indicate a stronger presence of harmonic components relative to noise, while lower values suggest a greater contribution of noise to the overall signal. Mathematically, it's defined as shown in Eq. 22:

$$HNR = \frac{\sum_{k=1}^{K} |X_h(k)|^2}{\sum_{k=1}^{K} |X_n(k)|^2}$$

Eq. 22: Harmonic-to-Noise Ratio

Where:

- $|X_h(k)|^2$ is the energy of the harmonic components in the frequency domain.
- $|x_n(k)|^2$ is the energy of the noise components in the frequency domain.
- K is the number of frequency bins considered.

A higher HNR value indicates a stronger presence of harmonic components relative to noise components, suggesting a more periodic and tonal signal. Conversely, a lower HNR value indicates a signal with more noise or non-harmonic components relative to harmonic components.

HNR is widely used in speech and voice analysis to evaluate voice quality, assess voice disorders, and monitor treatment progress. It provides valuable insights into the characteristics of the vocal source and can aid in diagnosing various voice disorders such as dysphonia and hoarseness.

2.1.15 Root Mean Square (RMS)

Root Mean Square (RMS) is a measure used to quantify the average power of a signal over a specified time interval. It provides a way to express the magnitude of a varying signal in a single value. RMS is particularly useful for analyzing signals with fluctuations, such as audio signals and electrical waveforms.

Here's how RMS is computed along with a mathematical expression:

1. Square the Signal: For a discrete-time signal x[n], square each sample of the signal as shown in Eq. 23:

$$x^2[n] = (x[n])^2$$

Eq. 23: Squaring the signal

2. Compute the Mean: Calculate the arithmetic mean (average) of the squared signal over the specified time interval as shown in Eq. 24:

$$Mean = \frac{1}{N}\sum_{n=0}^{N-1} x^2[n]$$

Eq. 24: Computing Mean

Where *N* is the total number of samples in the signal.

3. Take the Square Root: Take the square root of the mean value to obtain the Root Mean Square (RMS):

$$RMS = \sqrt{Mean}$$

Mathematically, the RMS can be expressed as shown in Eq. 25:

$$RMS = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}$$

Eq. 25: RMS

RMS provides a way to characterize the "effective" amplitude or power of a signal over a certain period of time. It accounts for both positive and negative fluctuations in the signal and is commonly used in various applications such as audio signal processing, electrical engineering, and vibration analysis.

Visualizing Emotion Feature Variations in the RAVDESS Dataset

We've meticulously crafted plots showcasing the variations in feature distributions across different emotions within the RAVDESS Dataset. Each plot offers a visual narrative, illustrating how the features dynamically fluctuate across various emotional states. Through these graphical representations shown in Fig 2.1.1 – Fig 2.1.17, we provide a comprehensive insight into the nuanced characteristics of emotional expressions encoded within the dataset. These plots serve as invaluable tools for understanding the underlying patterns and trends, aiding in the discernment of distinct feature profiles associated with each emotion.

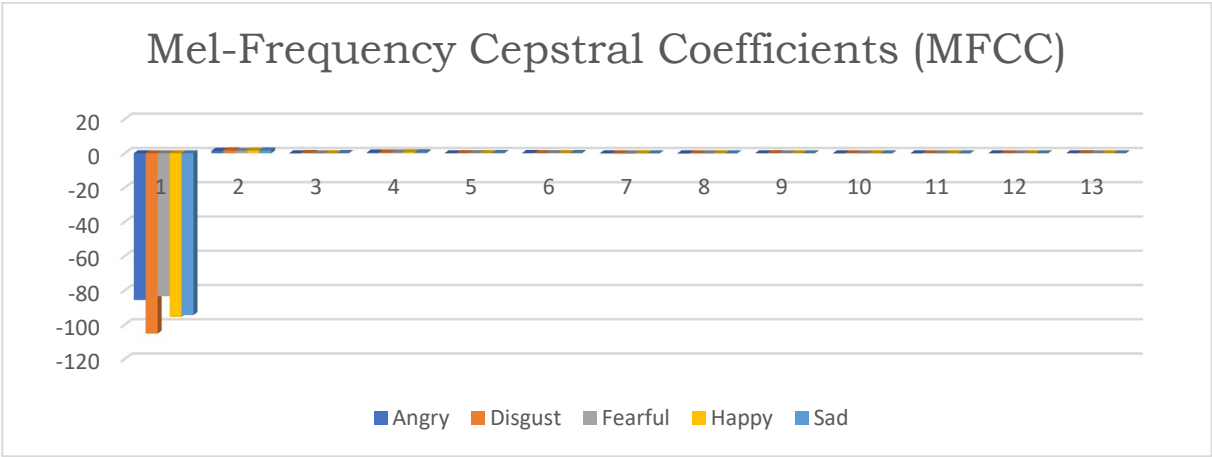

Fig 2.1.1: Visualizing variations in MFCC across Five emotions

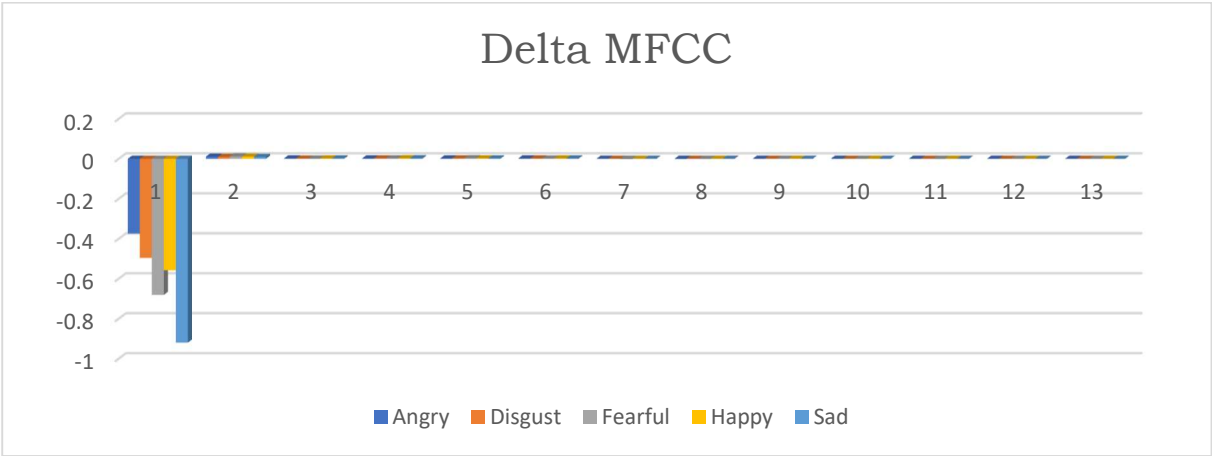

Fig 2.1.2: Visualizing variations in Delta MFCC across Five emotions

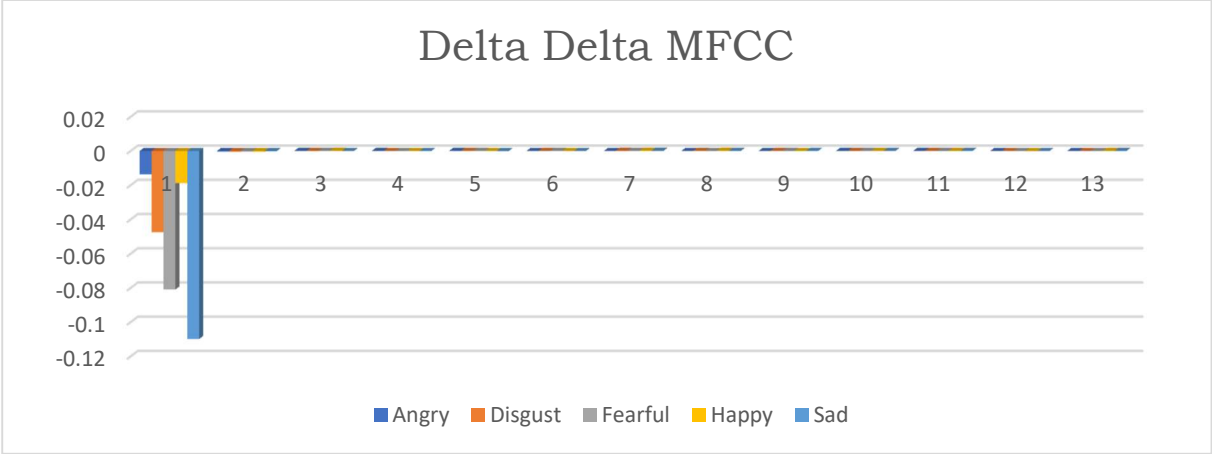

Fig 2.1.3: Visualizing variations in Delta Delta MFCC across Five emotions

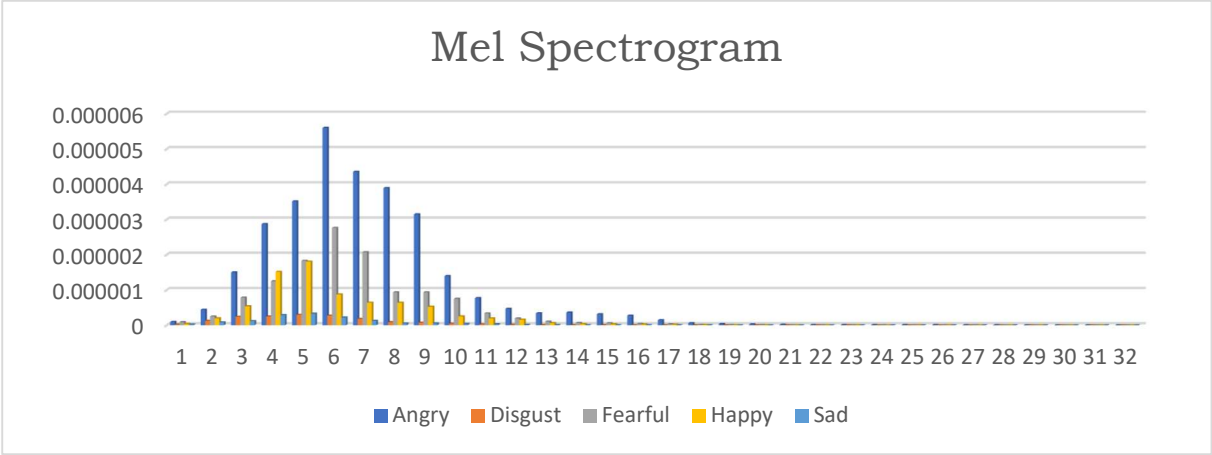

Fig 2.1.4: Visualizing variations in Mel Spectrogram across Five emotions

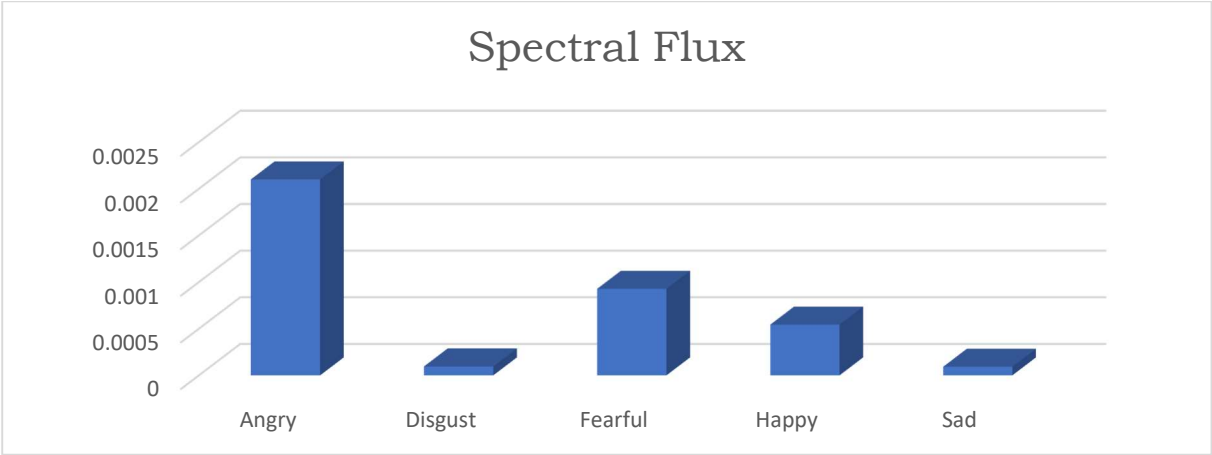

Fig 2.1.5: Visualizing variations in Spectral Flux across Five emotions

Fig 2.1.6: Visualizing variations in Spectral Skewness across Five emotions

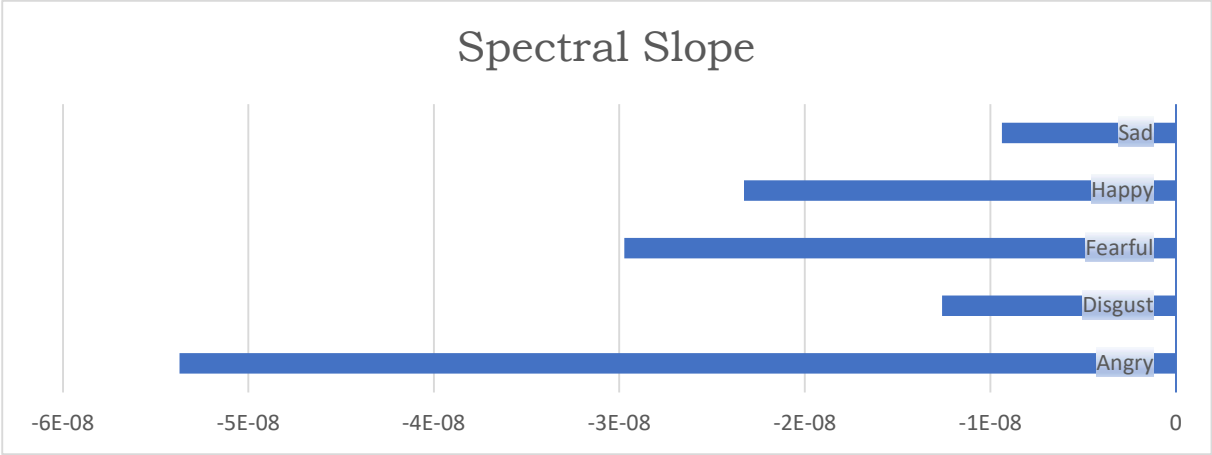

Fig 2.1.7: Visualizing variations in Spectral Slope across Five emotions

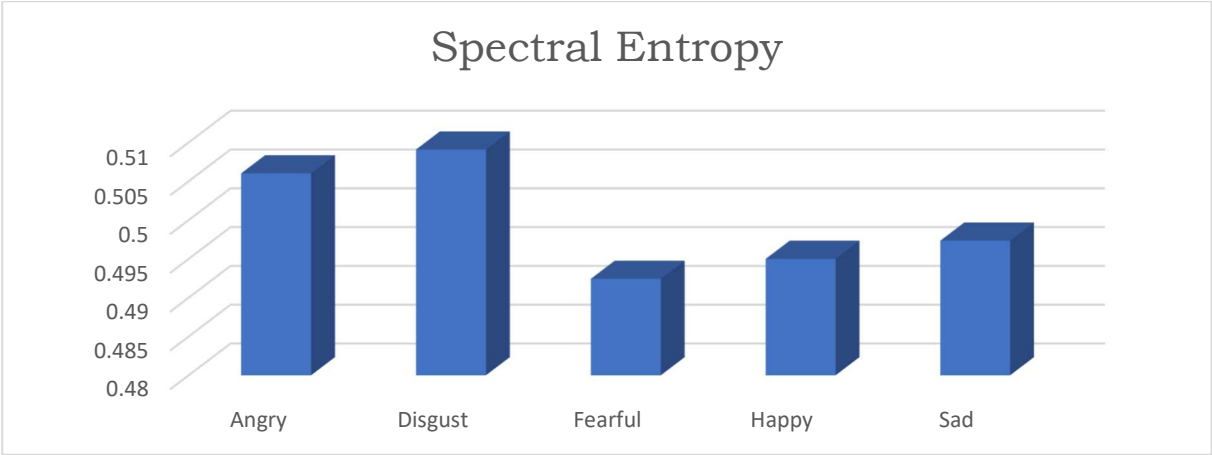

Fig 2.1.8: Visualizing variations in Spectral Entropy across Five emotions

Fig 2.1.9: Visualizing variations in Spectral Rolloff across Five emotions

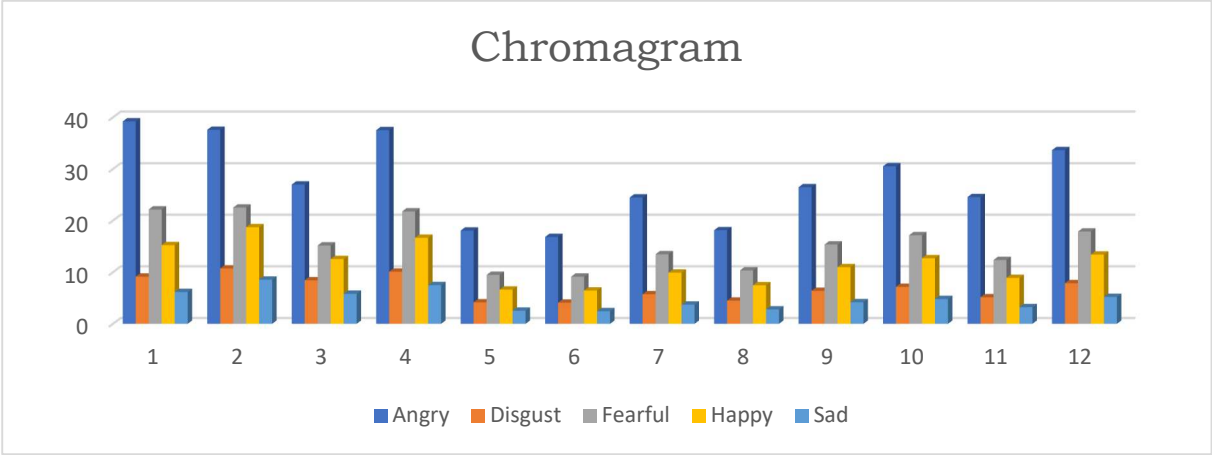

Fig 2.1.10: Visualizing variations in Chromagram across Five emotions

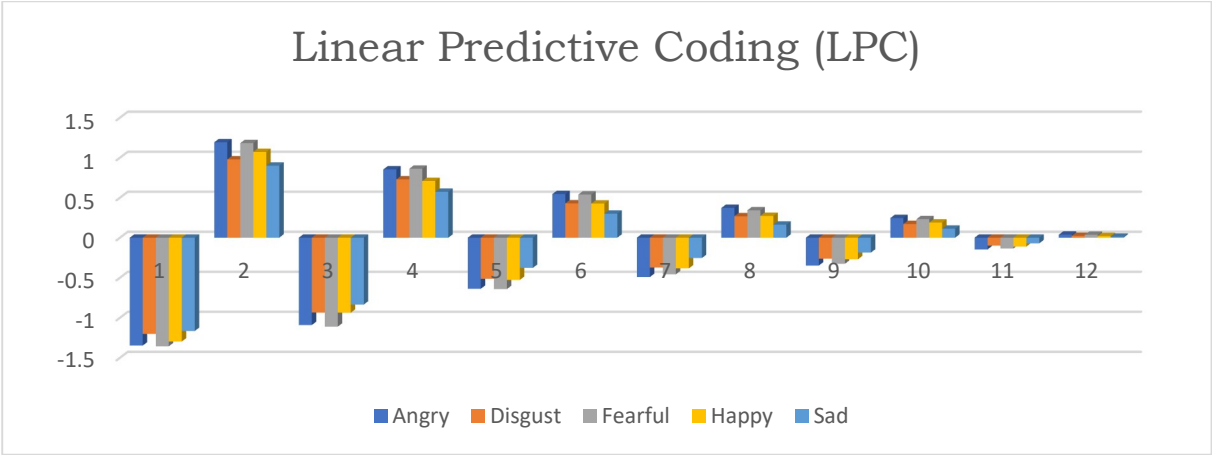

Fig 2.1.11: Visualizing variations in LPC across Five emotions

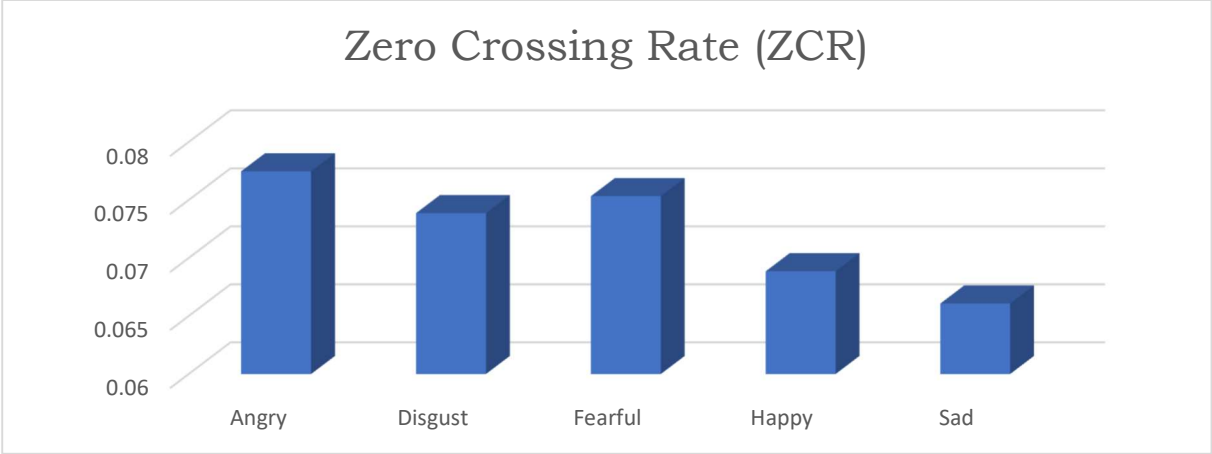

Fig 2.1.12: Visualizing variations in ZCR across Five emotions

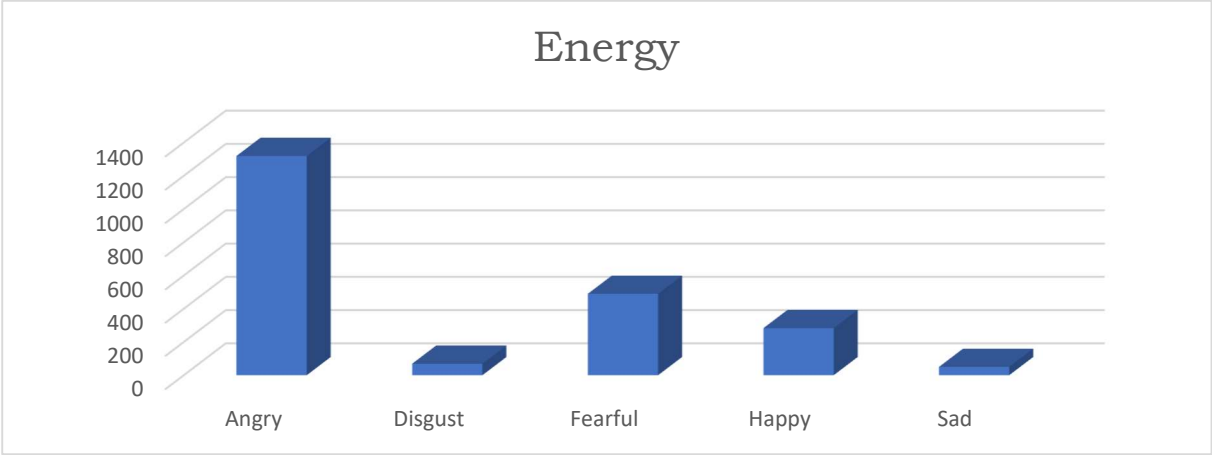

Fig 2.1.13: Visualizing variations in Energy across Five emotions

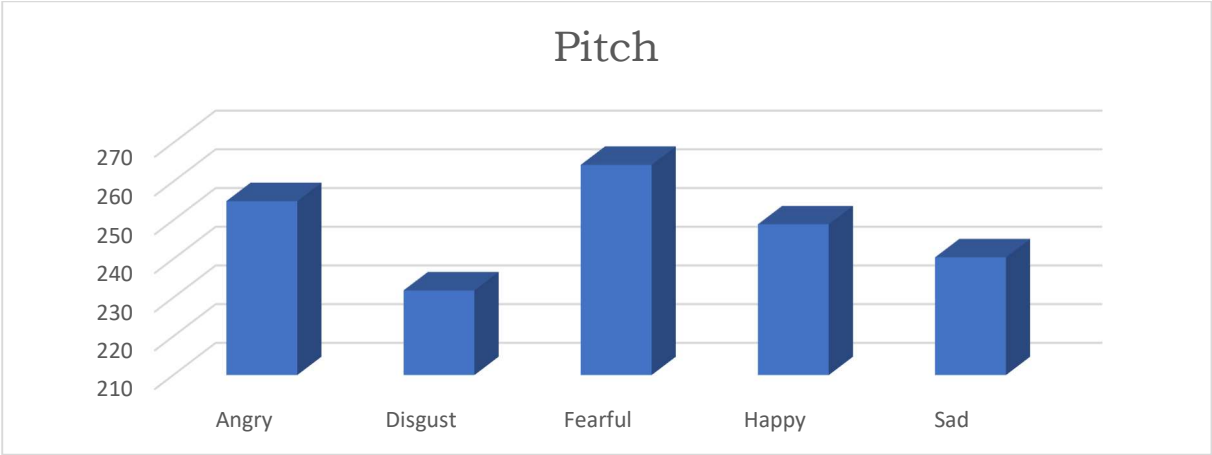

Fig 2.1.14: Visualizing variations in Pitch across Five emotions

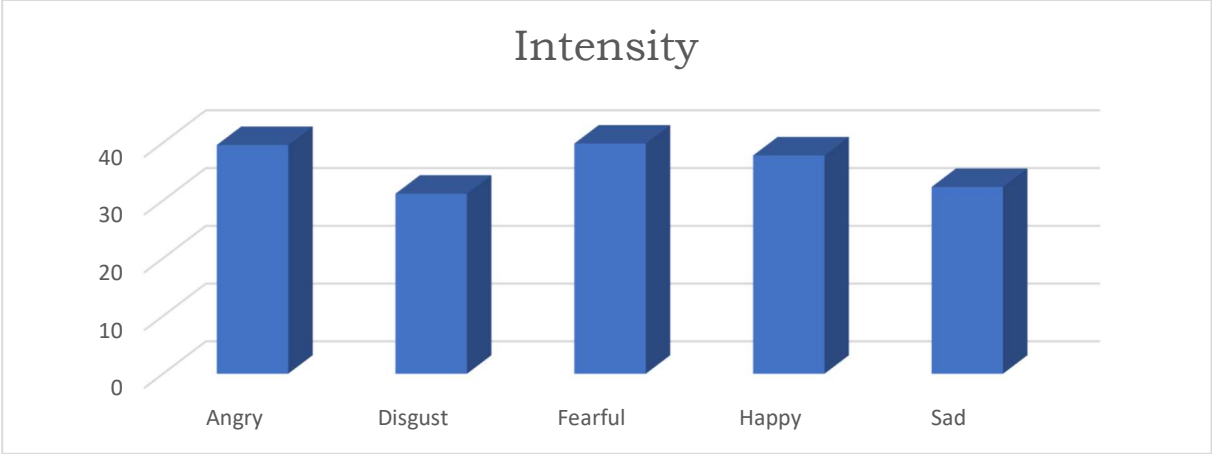

Fig 2.1.15: Visualizing variations in Intensity across Five emotions

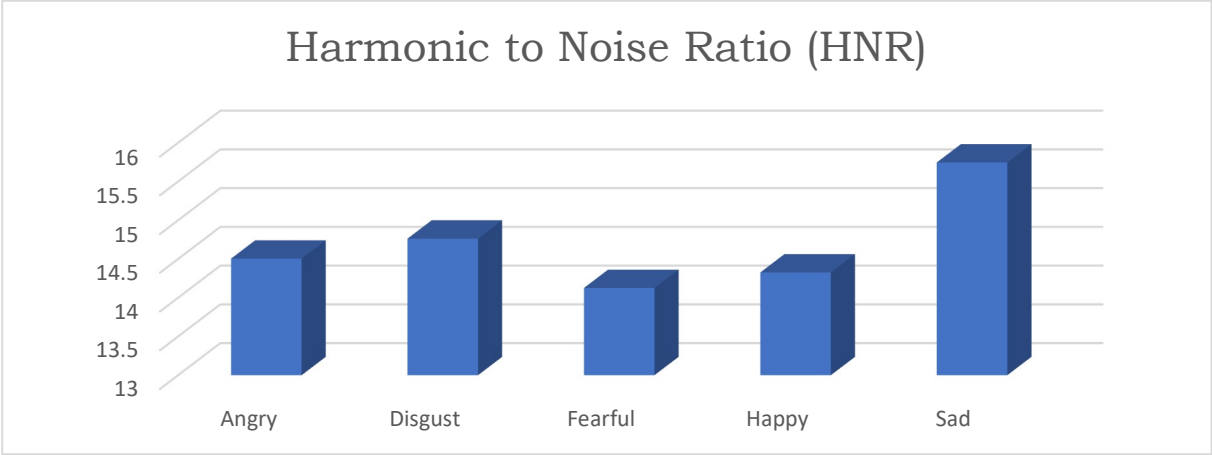

Fig 2.1.16: Visualizing variations in HNR across Five emotions

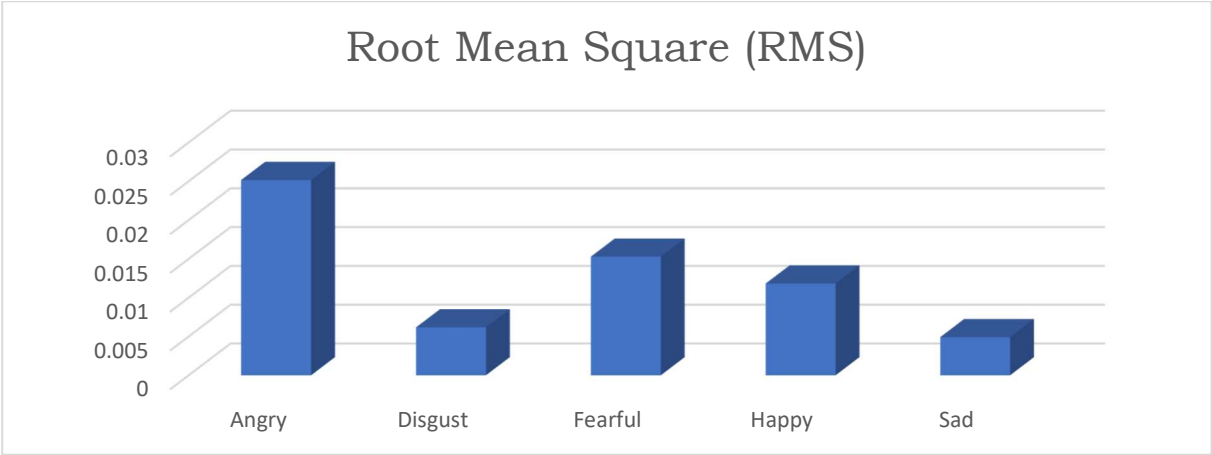

Fig 2.1.17: Visualizing variations in RMS across Five emotions

Visualizing Emotion Feature Variations in the TESS Dataset

We've meticulously crafted plots showcasing the variations in feature distributions across different emotions within the TESS Dataset. Each plot offers a visual narrative, illustrating how the features dynamically fluctuate across various emotional states. Through these graphical representations shown in Fig 2.1.18 – Fig 2.1.34, we provide a comprehensive insight into the nuanced characteristics of emotional expressions encoded within the dataset. These plots serve as invaluable tools for understanding the underlying patterns and trends, aiding in the discernment of distinct feature profiles associated with each emotion.

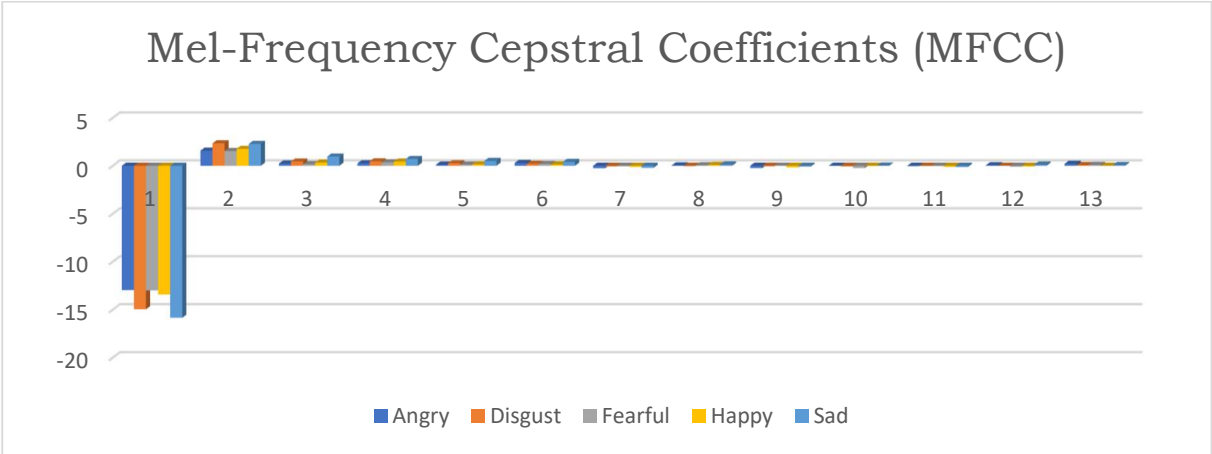

Fig 2.1.18: Visualizing variations in MFCC across Five emotions

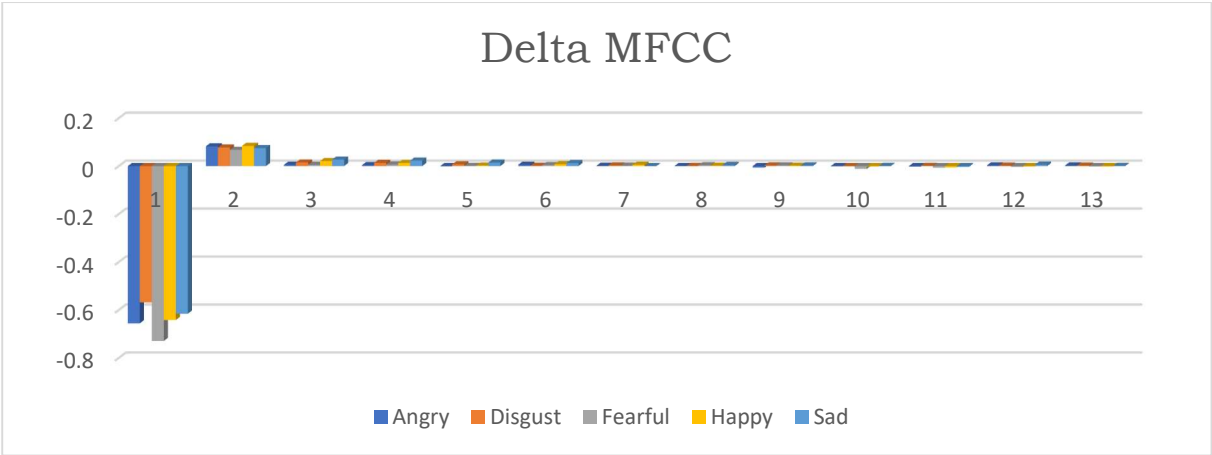

Fig 2.1.19: Visualizing variations in Delta MFCC across Five emotions

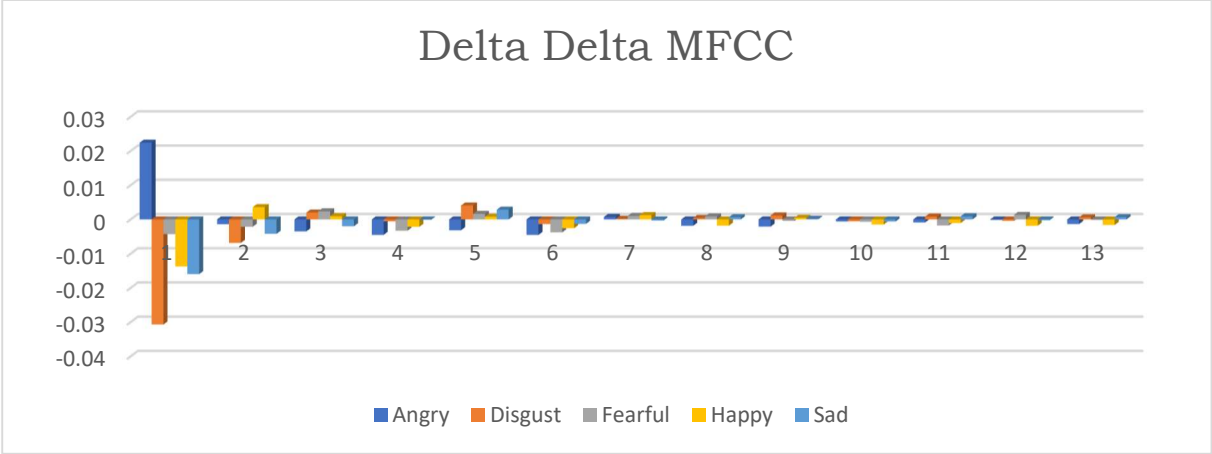

Fig 2.1.20: Visualizing variations in Delta Delta MFCC across Five emotions

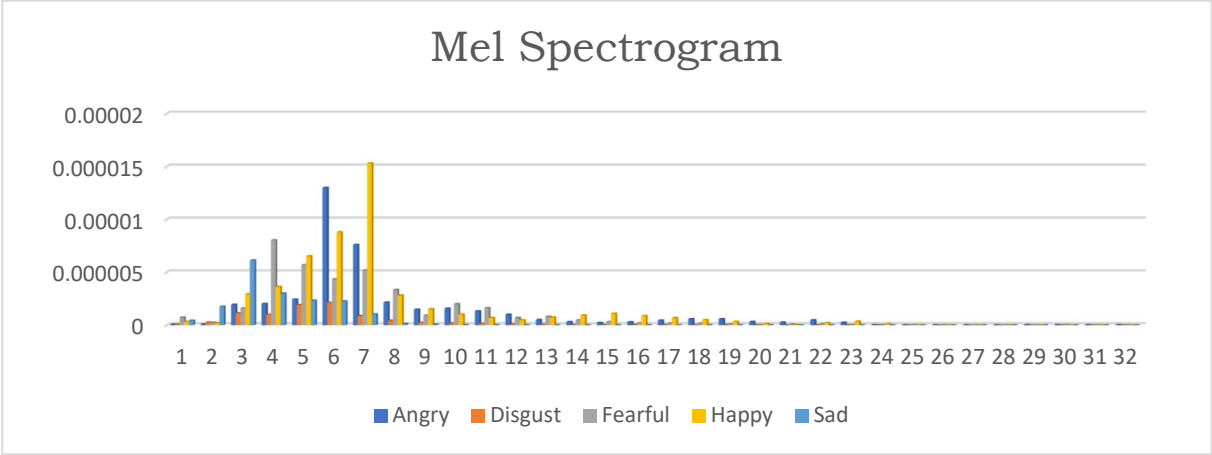

Fig 2.1.21: Visualizing variations in Mel Spectrogram across Five emotions

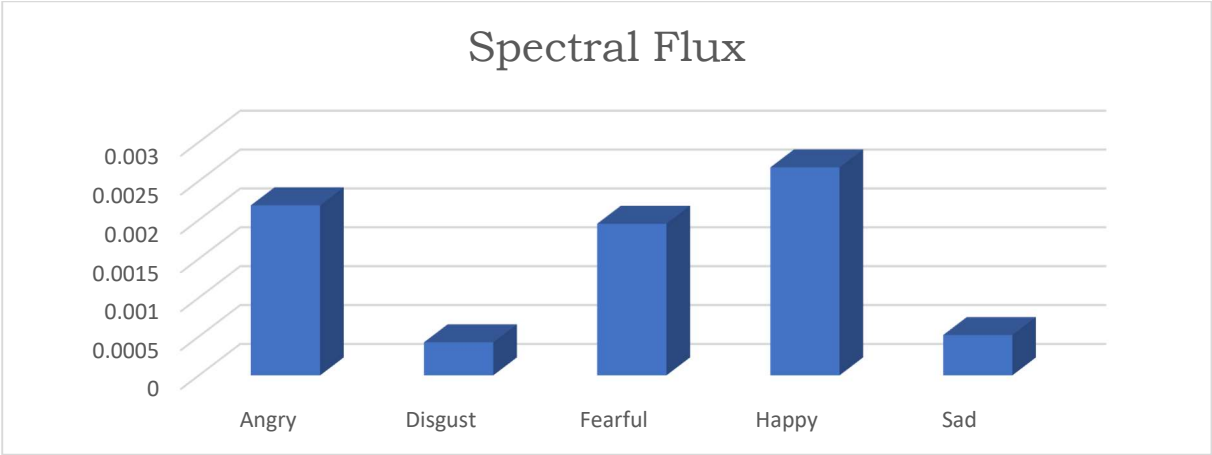

Fig 2.1.22: Visualizing variations in Spectral Flux across Five emotions

Fig 2.1.23: Visualizing variations in Spectral Skewness across Five emotions

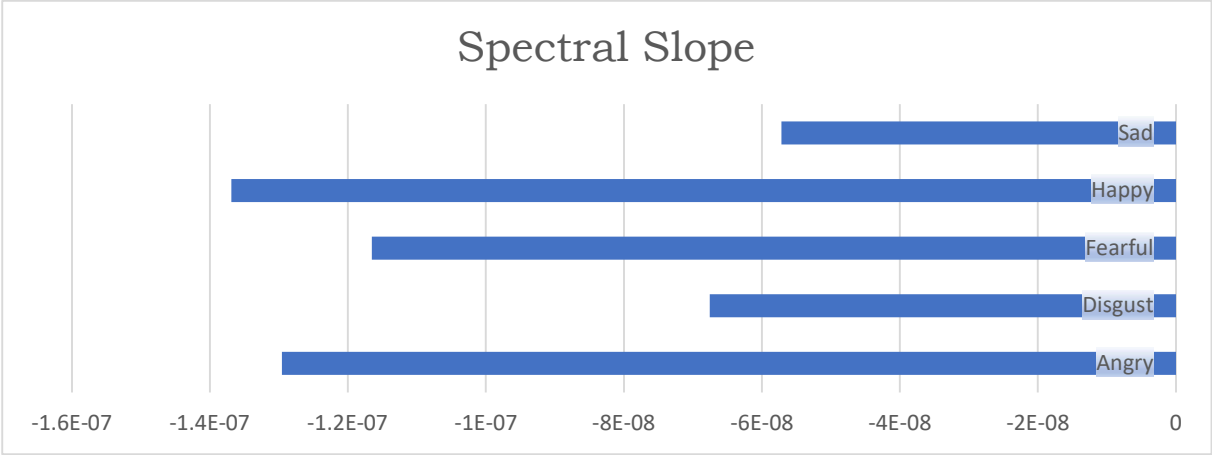

Fig 2.1.24: Visualizing variations in Spectral Slope across Five emotions

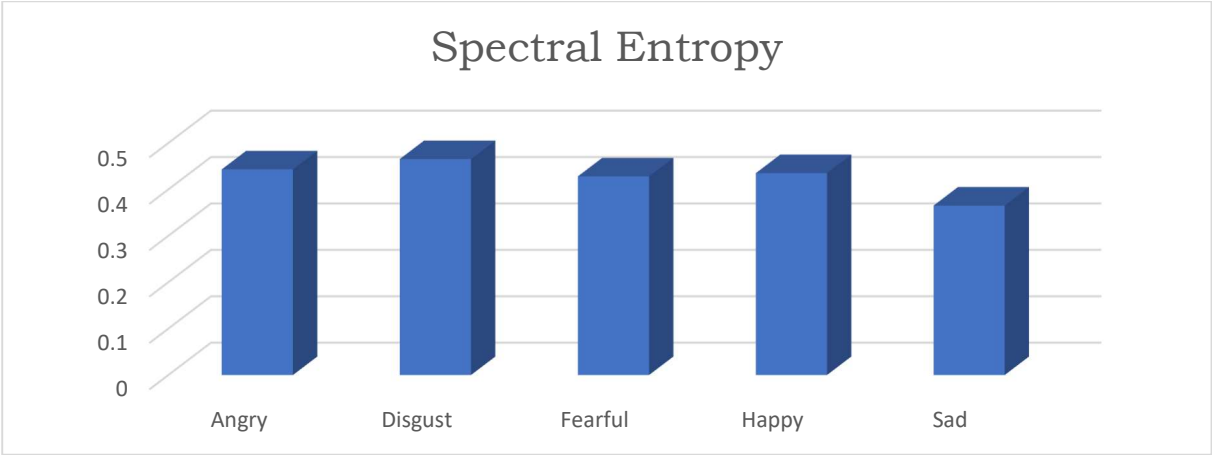

Fig 2.1.25: Visualizing variations in Spectral Entropy across Five emotions

Fig 2.1.26: Visualizing variations in Spectral Rolloff across Five emotions

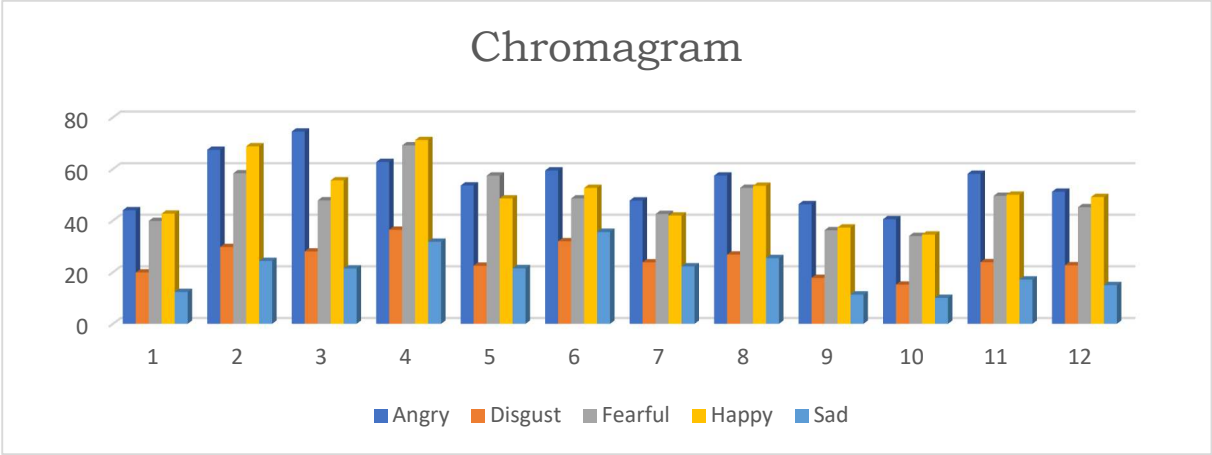

Fig 2.1.27: Visualizing variations in Chromagram across Five emotions

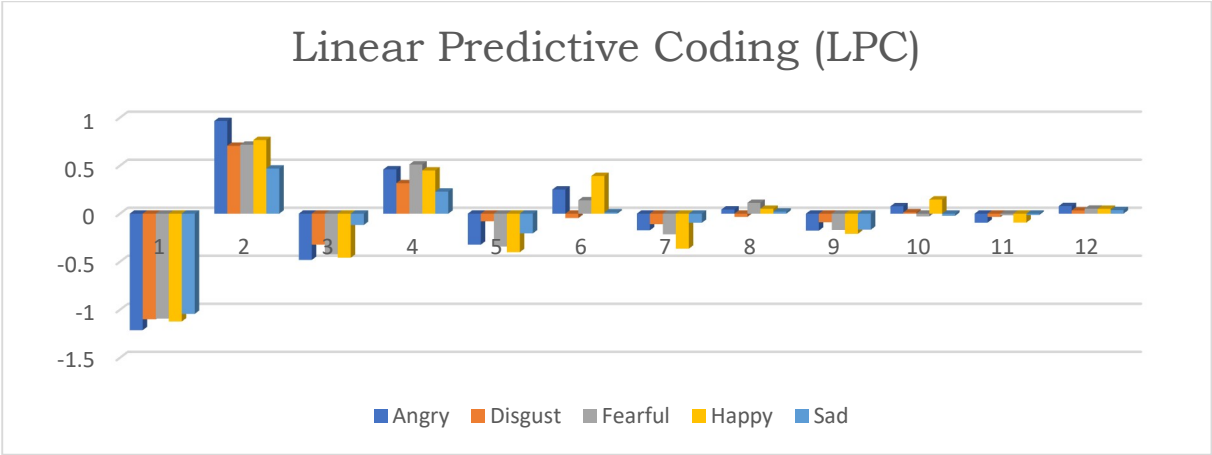

Fig 2.1.28: Visualizing variations in LPC across Five emotions

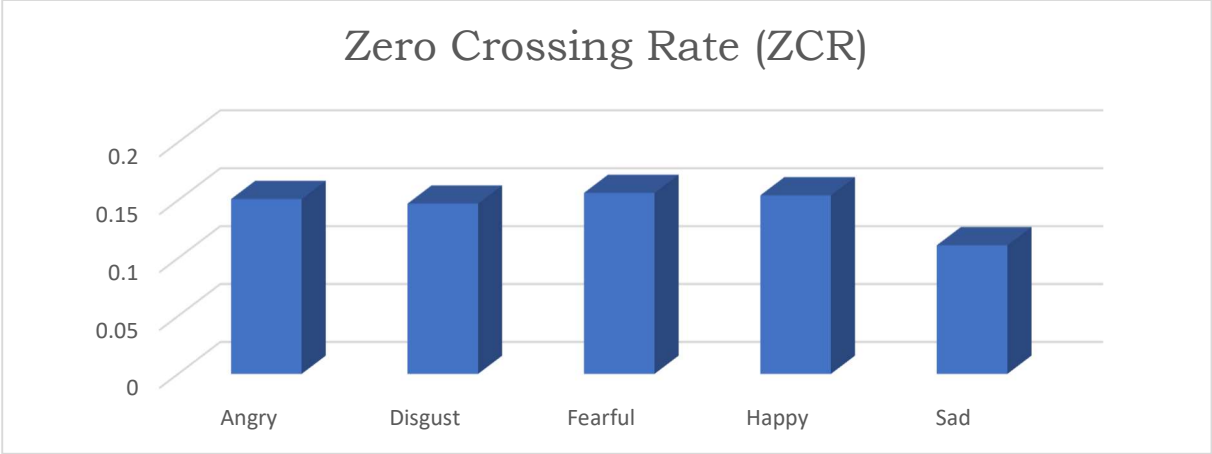

Fig 2.1.29: Visualizing variations in ZCR across Five emotions

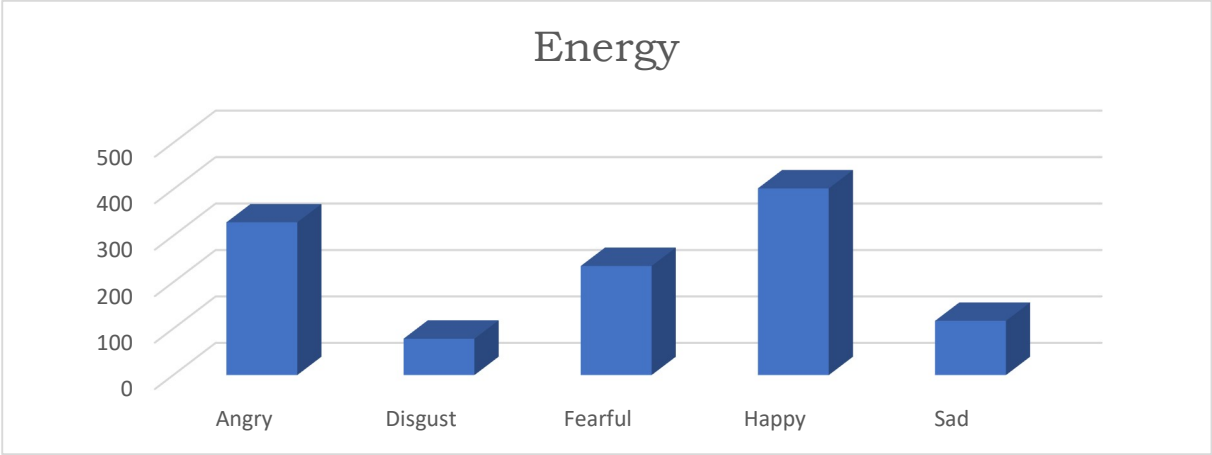

Fig 2.1.30: Visualizing variations in Energy across Five emotions

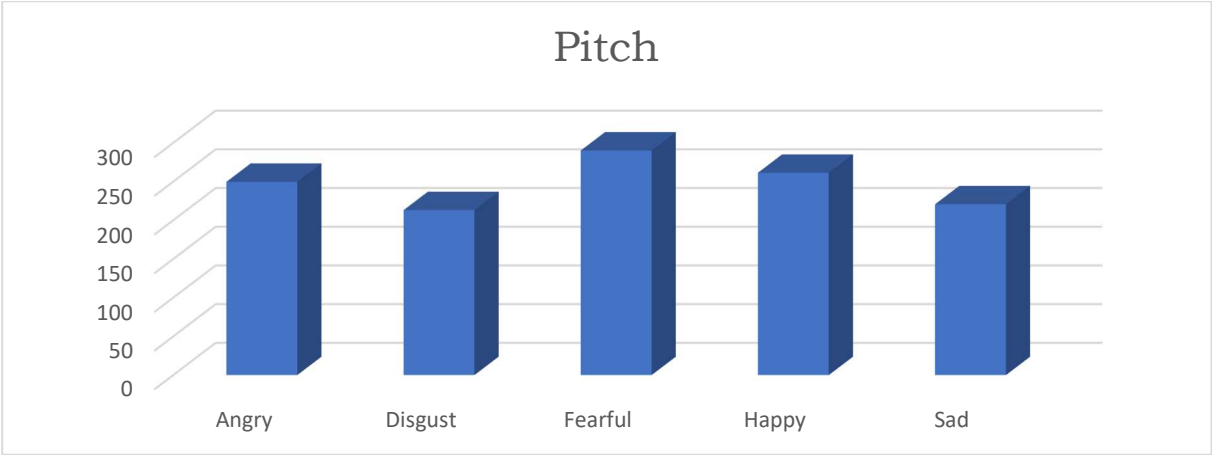

Fig 2.1.31: Visualizing variations in Pitch across Five emotions

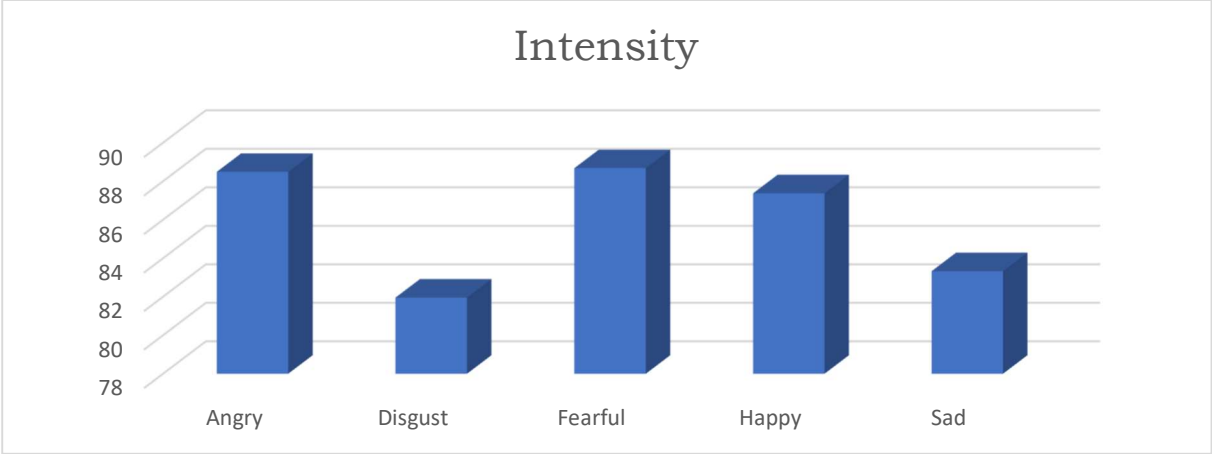

Fig 2.1.32: Visualizing variations in Intensity across Five emotions

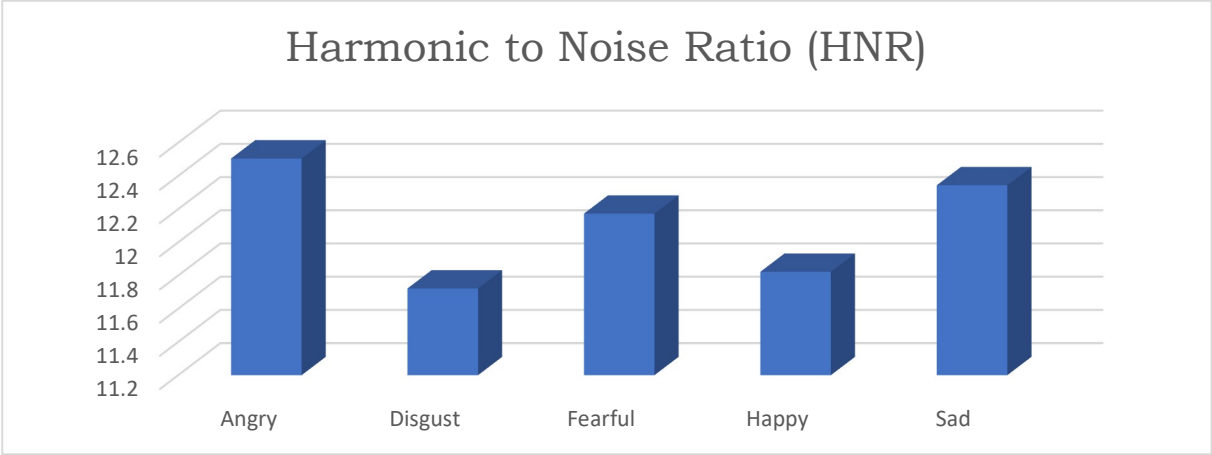

Fig 2.1.33: Visualizing variations in HNR across Five emotions

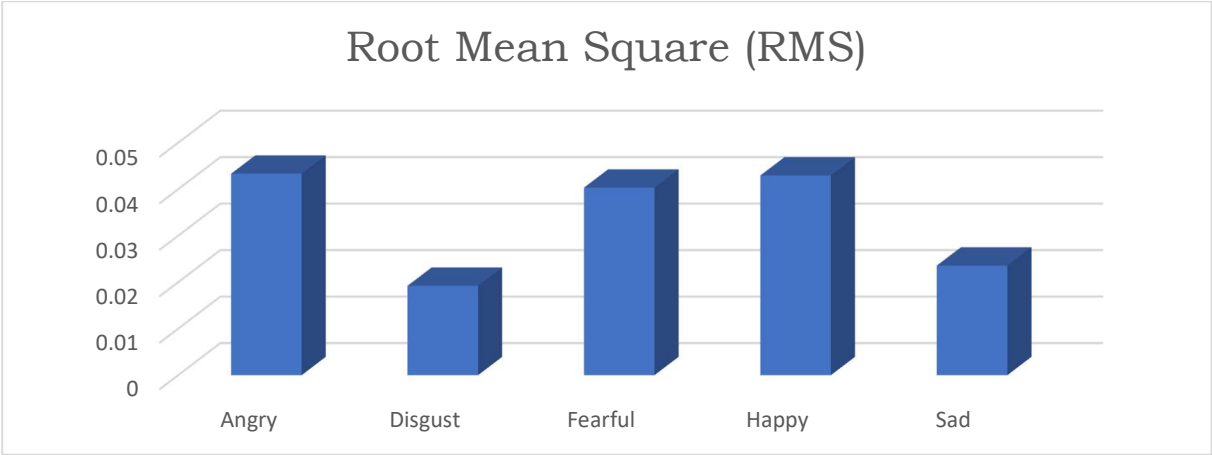

Fig 2.1.34: Visualizing variations in RMS across Five emotions

Visualizing Emotion Feature Variations in the SAVEE Dataset

We've meticulously crafted plots showcasing the variations in feature distributions across different emotions within the SAVEE Dataset. Each plot offers a visual narrative, illustrating how the features dynamically fluctuate across various emotional states. Through these graphical representations shown in Fig 2.1.35 – Fig 2.1.51, we provide a comprehensive insight into the nuanced characteristics of emotional expressions encoded within the dataset. These plots serve as invaluable tools for understanding the underlying patterns and trends, aiding in the discernment of distinct feature profiles associated with each emotion.

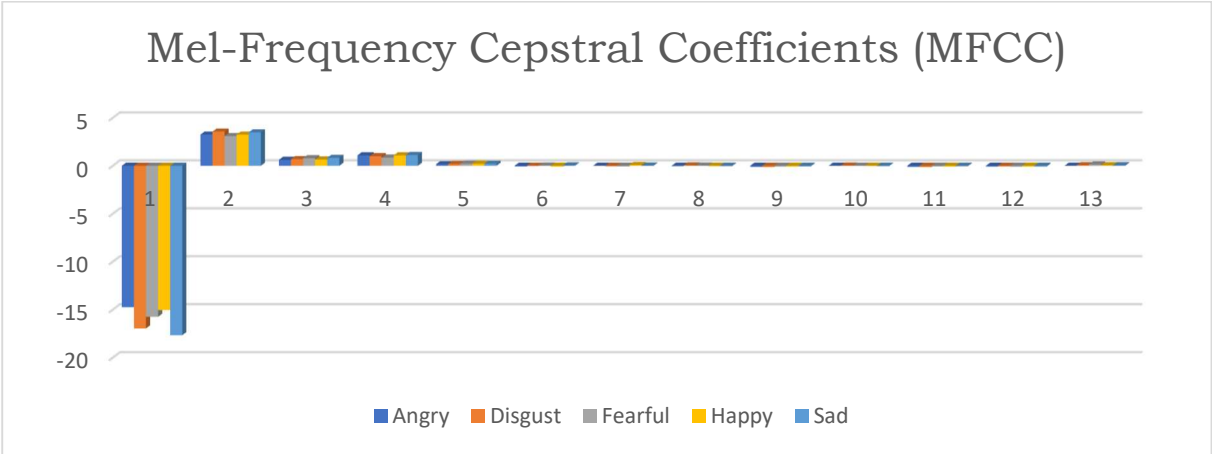

Fig 2.1.35: Visualizing variations in MFCC across Five emotions

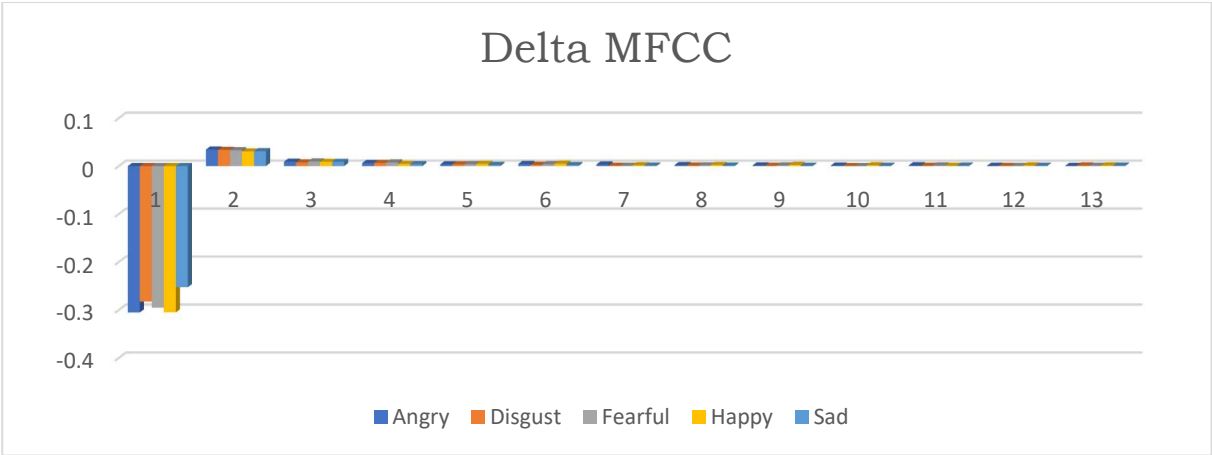

Fig 2.1.36: Visualizing variations in Delta MFCC across Five emotions

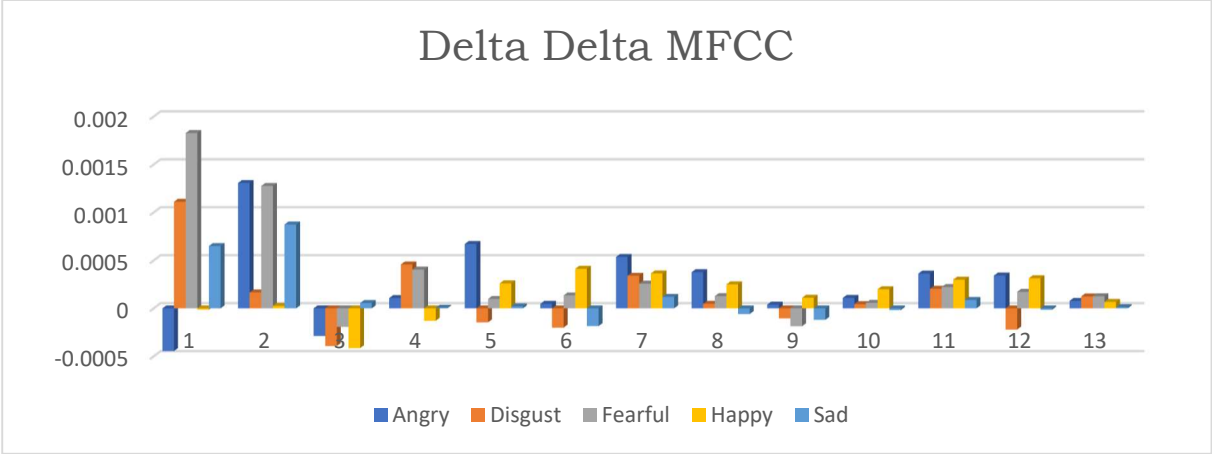

Fig 2.1.37: Visualizing variations in Delta Delta MFCC across Five emotions

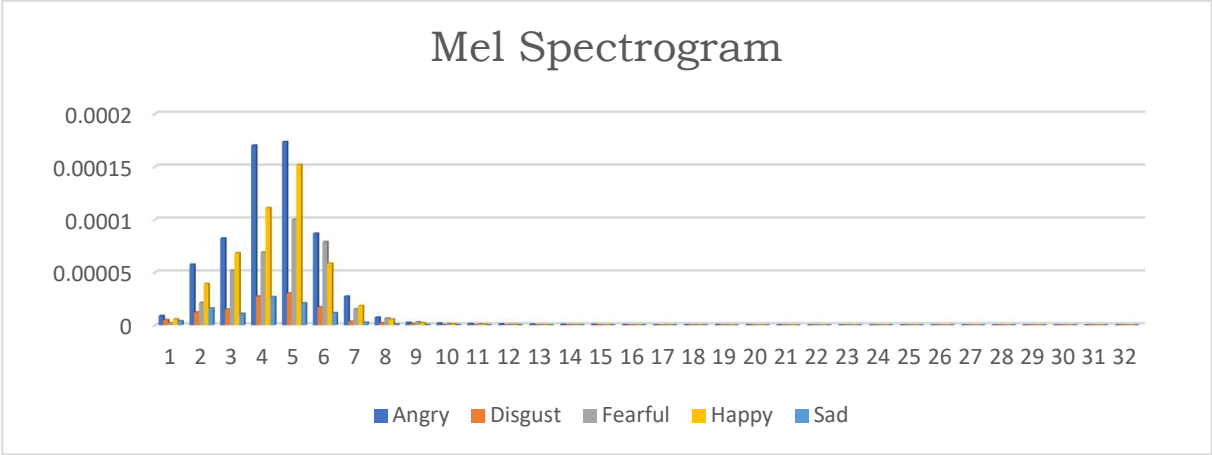

Fig 2.1.38: Visualizing variations in Mel Spectrogram across Five emotions

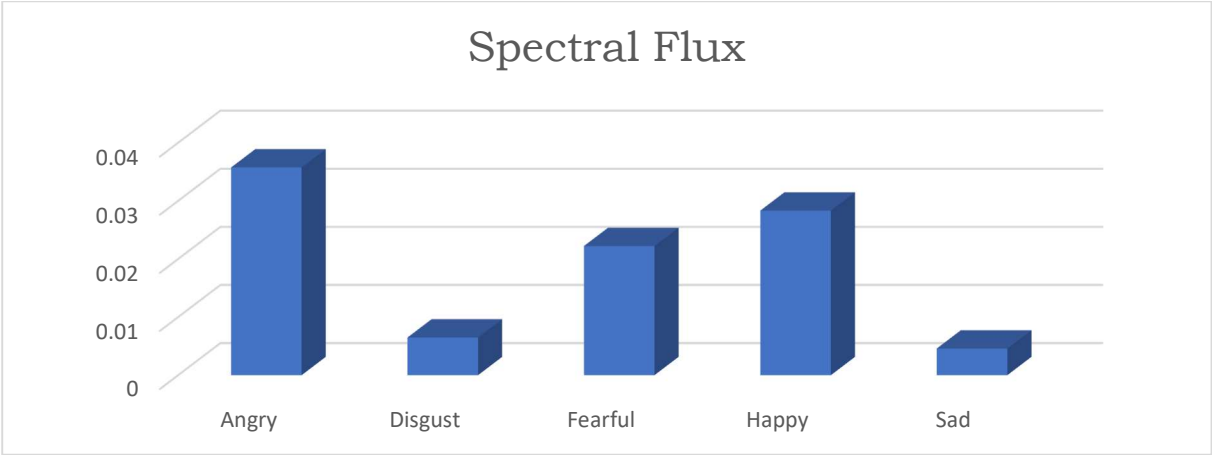

Fig 2.1.39: Visualizing variations in Spectral Flux across Five emotions

Fig 2.1.40: Visualizing variations in Spectral Skewness across Five emotions

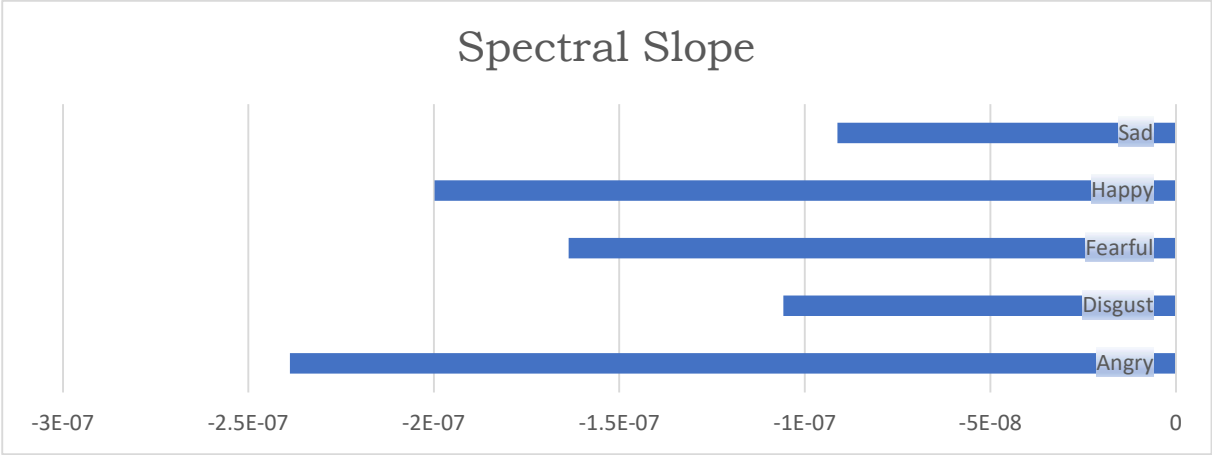

Fig 2.1.41: Visualizing variations in Spectral Slope across Five emotions

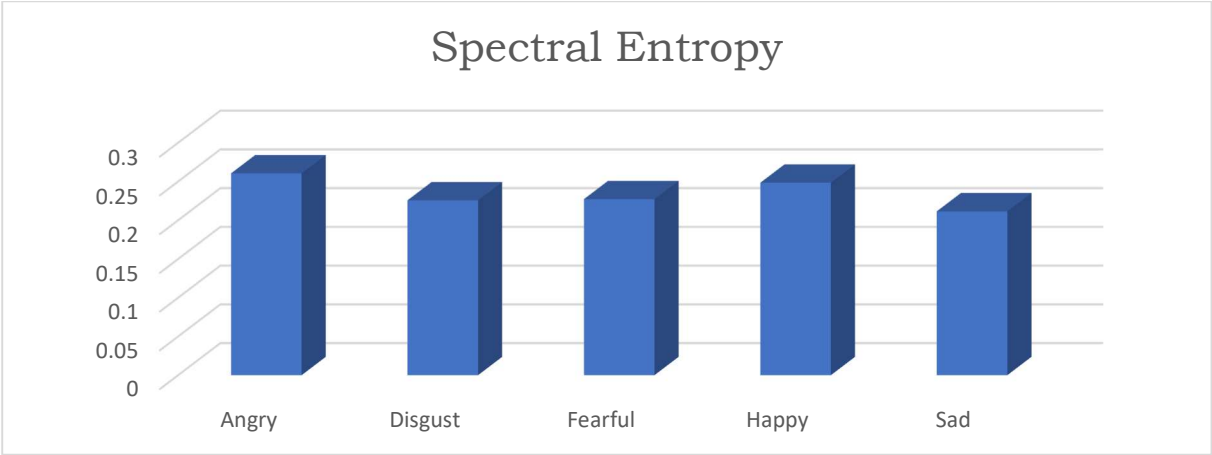

Fig 2.1.42: Visualizing variations in Spectral Entropy across Five emotions

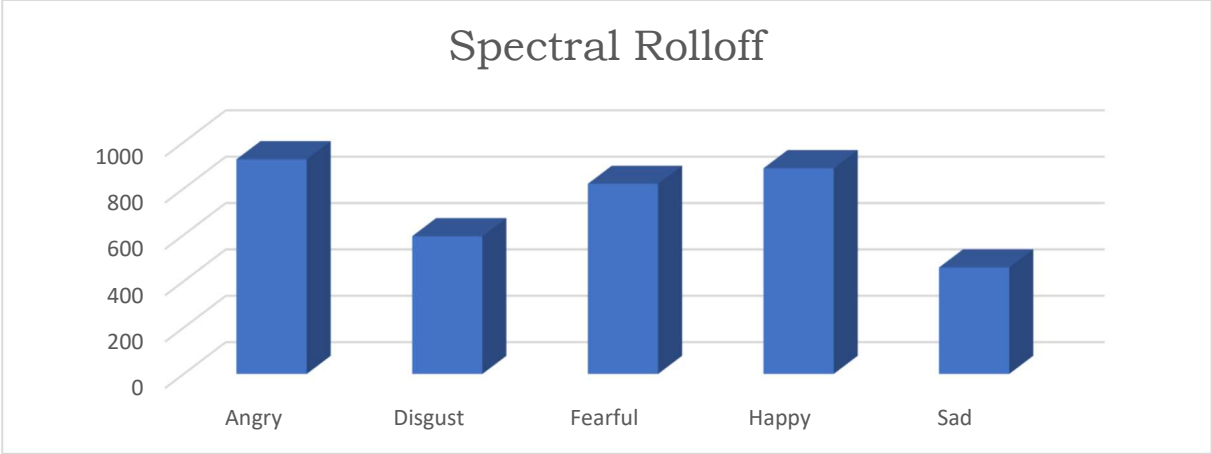

Fig 2.1.43: Visualizing variations in Spectral Rolloff across Five emotions

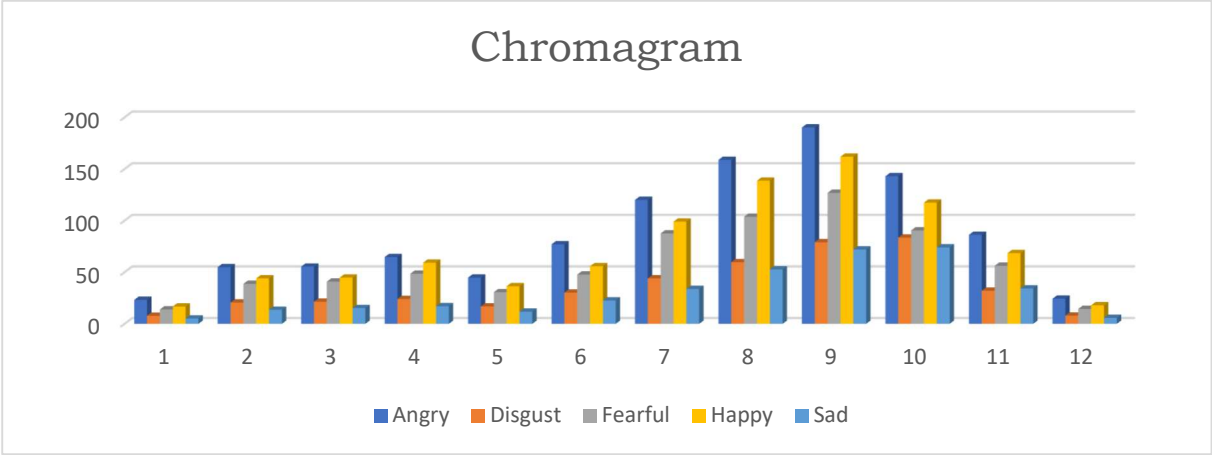

Fig 2.1.44: Visualizing variations in Chromagram across Five emotions

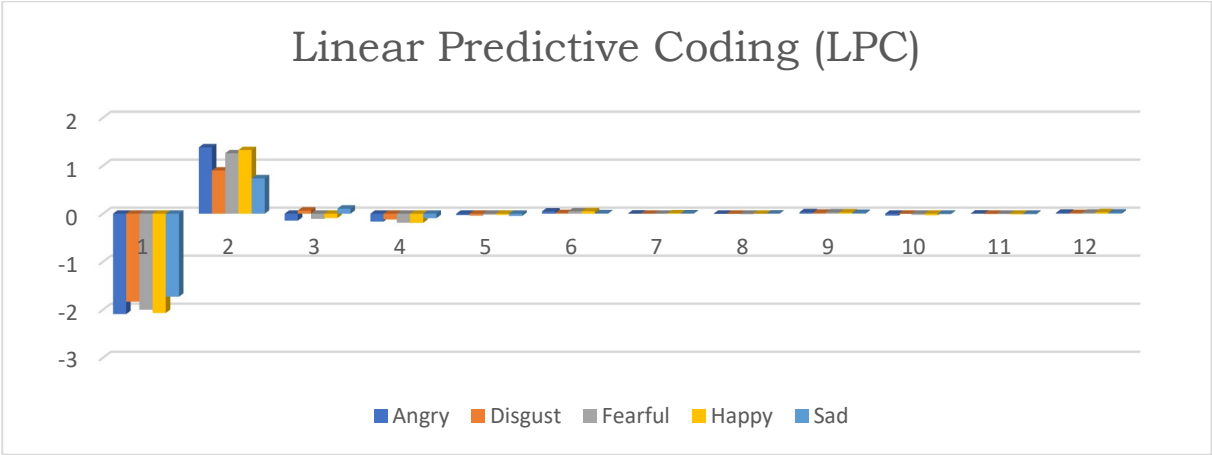

Fig 2.1.45: Visualizing variations in LPC across Five emotions

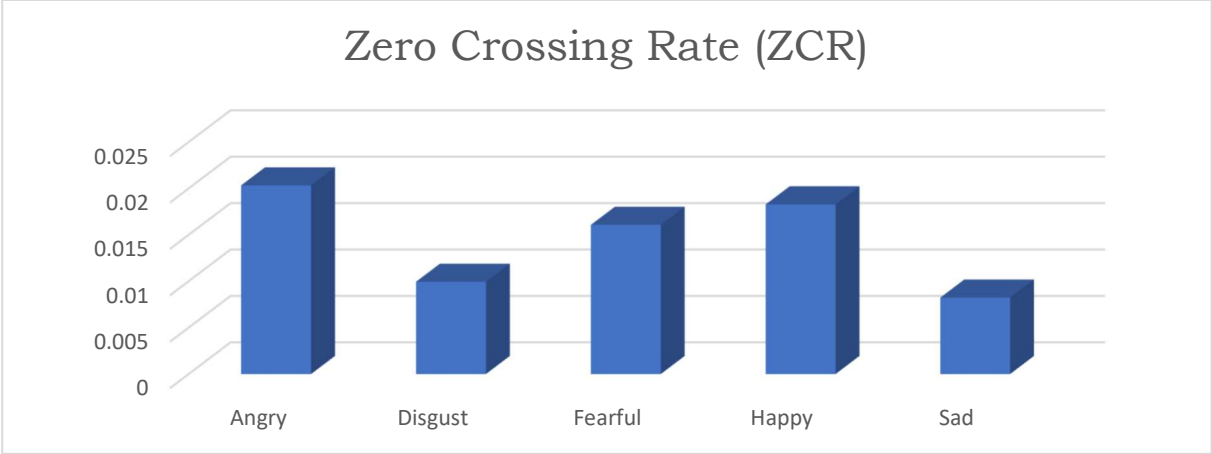

Fig 2.1.46: Visualizing variations in ZCR across Five emotions

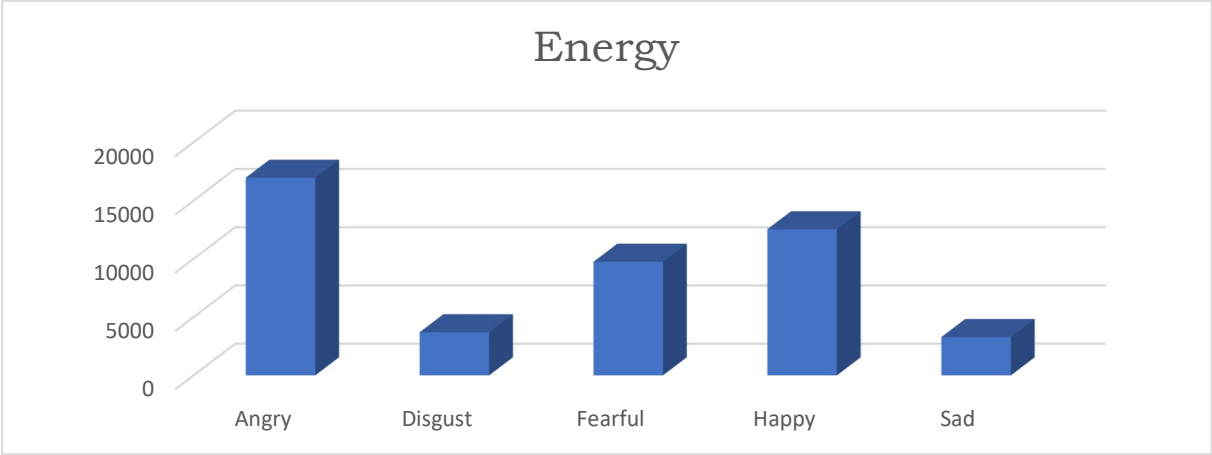

Fig 2.1.47: Visualizing variations in Energy across Five emotions

Fig 2.1.48: Visualizing variations in Pitch across Five emotions

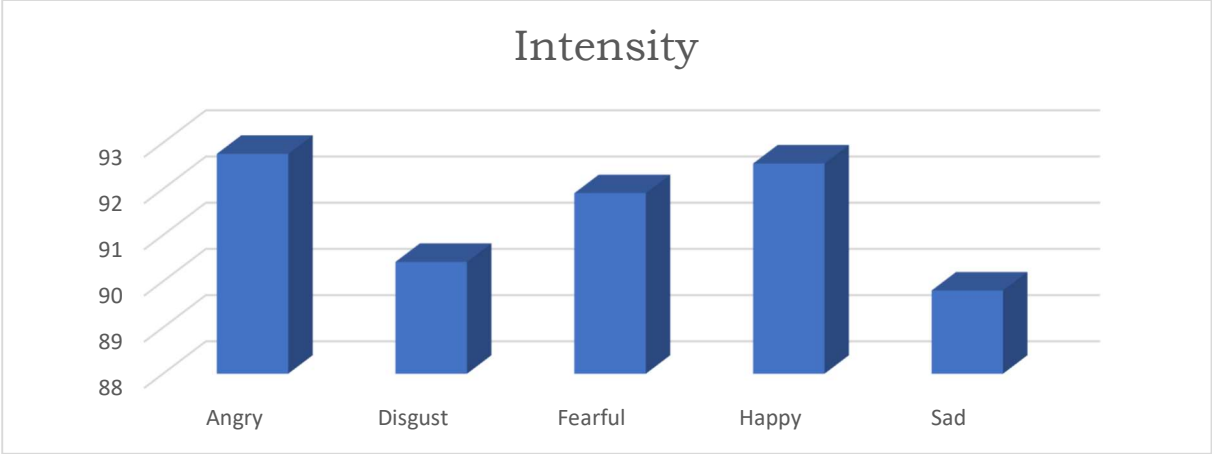

Fig 2.1.49: Visualizing variations in Intensity across Five emotions

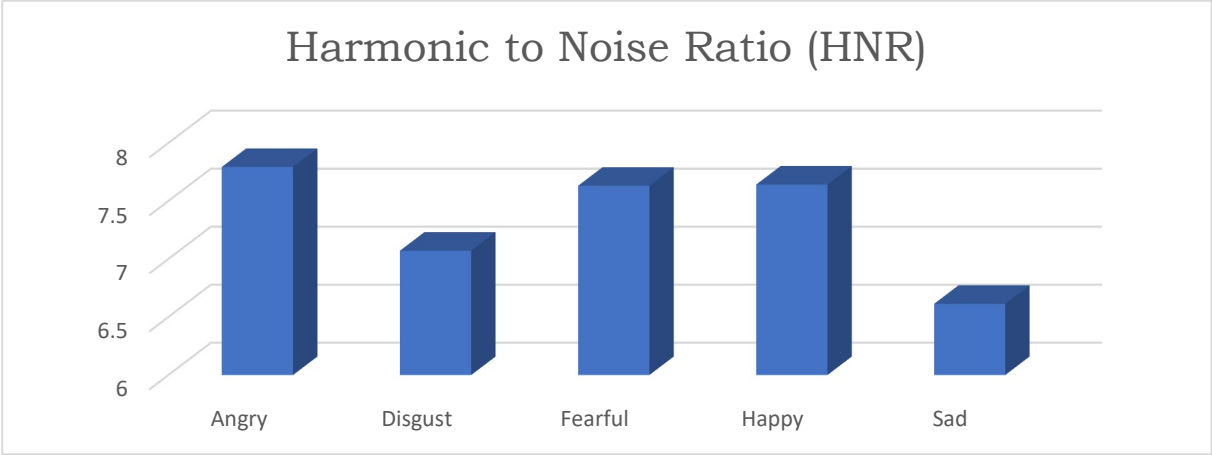

Fig 2.1.50: Visualizing variations in HNR across Five emotions

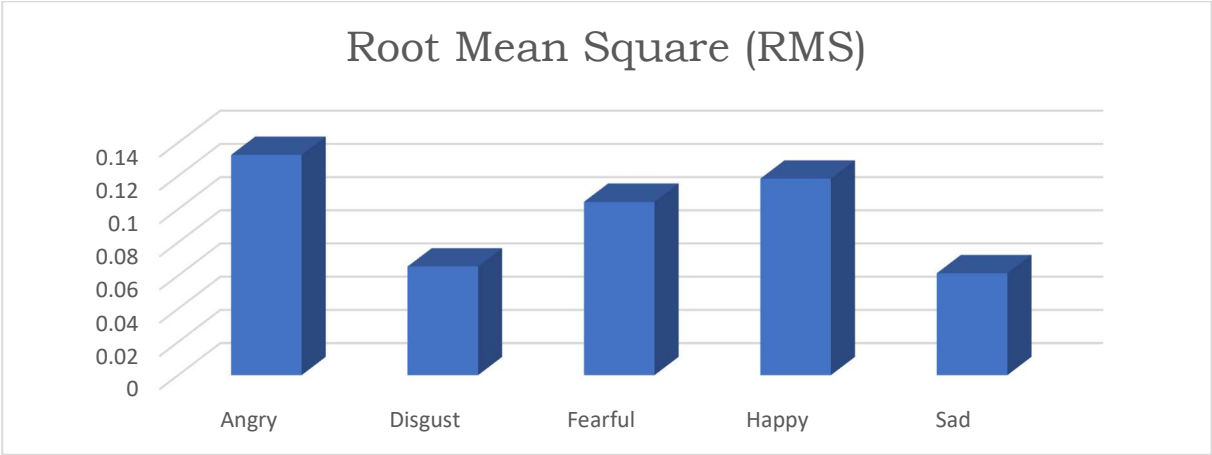

Fig 2.1.51: Visualizing variations in RMS across Five emotions

Visualizing Emotion Feature Variations in the EmoDB Dataset

We've meticulously crafted plots showcasing the variations in feature distributions across different emotions within the EmoDB Dataset. Each plot offers a visual narrative, illustrating how the features dynamically fluctuate across various emotional states. Through these graphical representations shown in Fig 2.1.52 – Fig 2.1.68, we provide a comprehensive insight into the nuanced characteristics of emotional expressions encoded within the dataset. These plots serve as invaluable tools for understanding the underlying patterns and trends, aiding in the discernment of distinct feature profiles associated with each emotion.

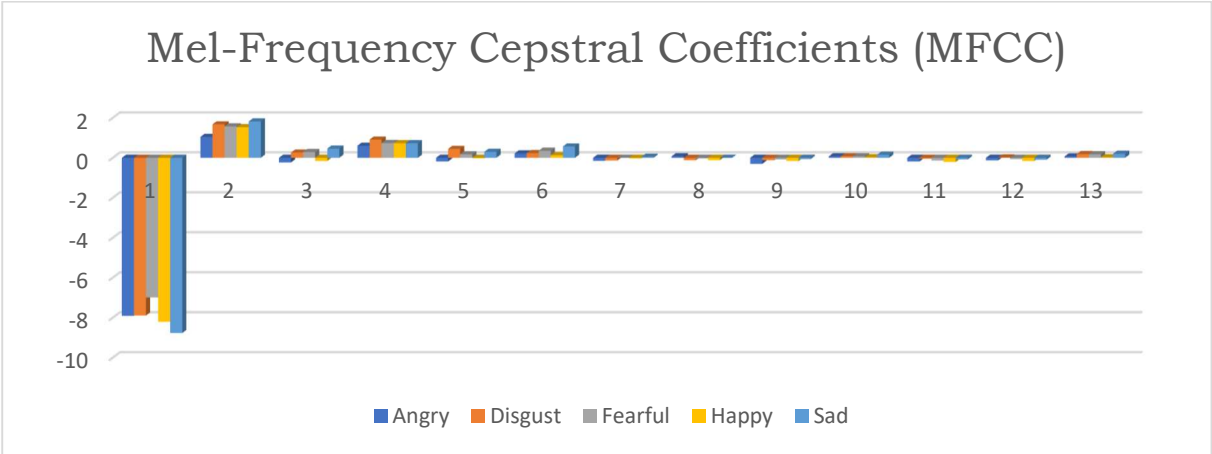

Fig 2.1.52: Visualizing variations in MFCC across Five emotions

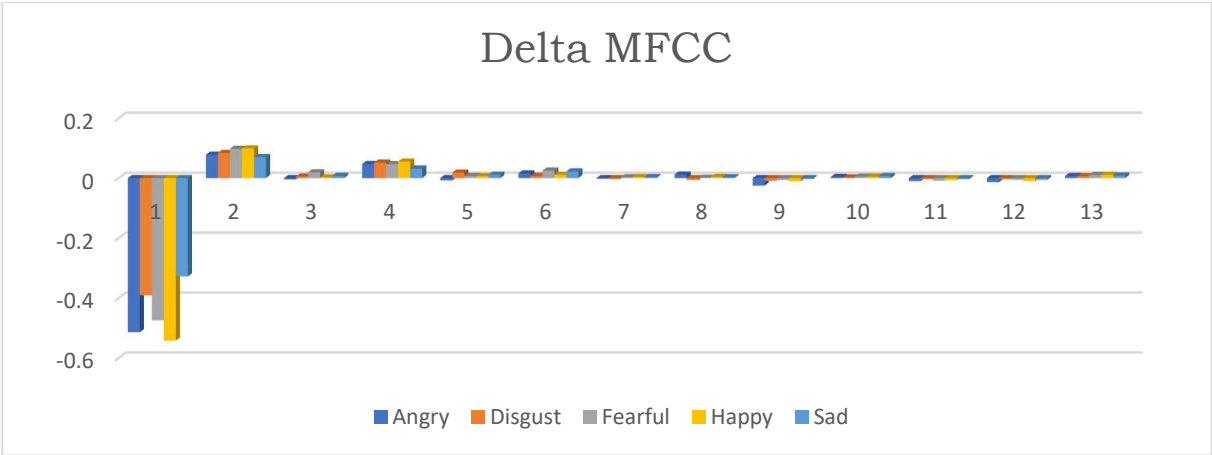

Fig 2.1.53: Visualizing variations in Delta MFCC across Five emotions

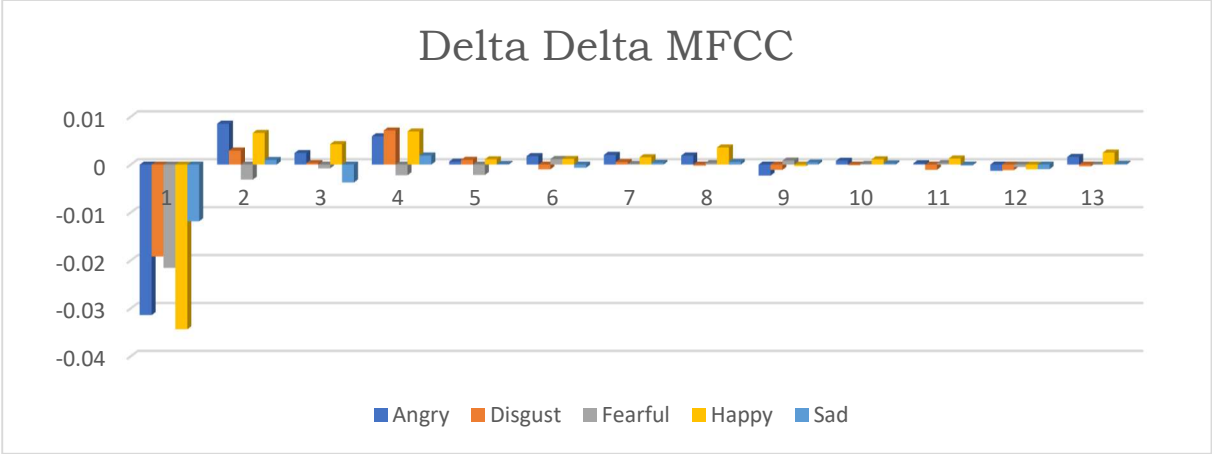

Fig 2.1.54: Visualizing variations in Delta Delta MFCC across Five emotions

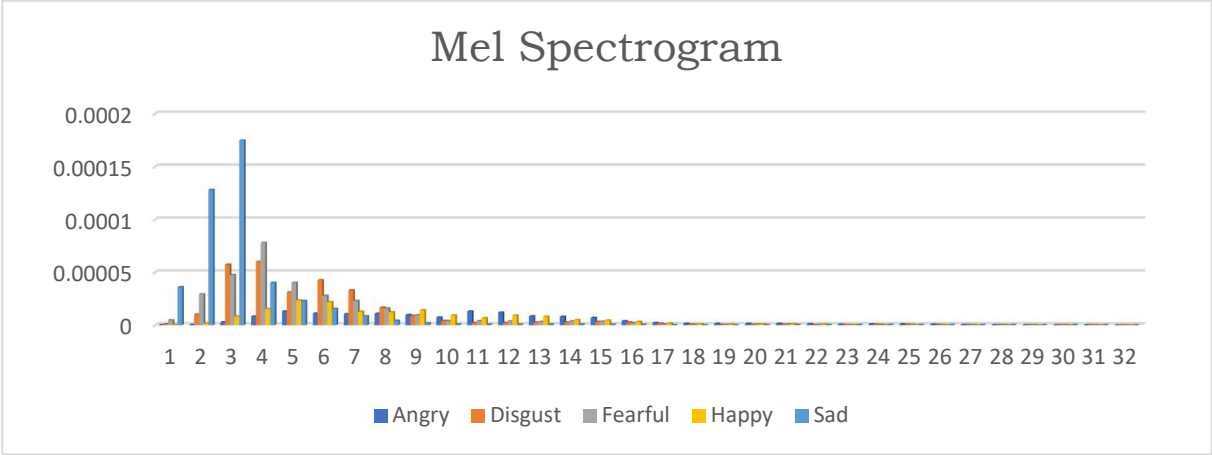

Fig 2.1.55: Visualizing variations in Mel Spectrogram across Five emotions

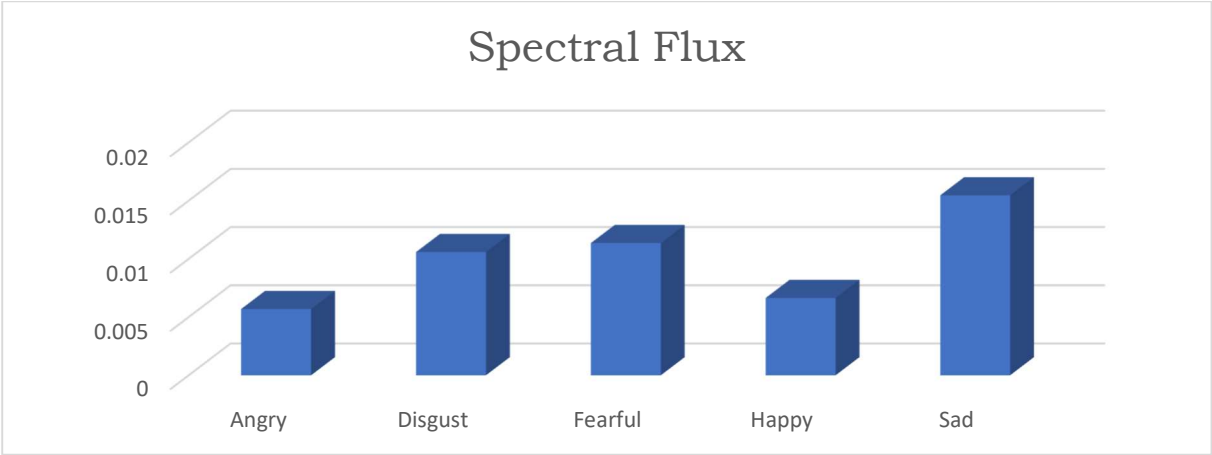

Fig 2.1.56: Visualizing variations in Spectral Flux across Five emotions

Fig 2.1.57: Visualizing variations in Spectral Skewness across Five emotions

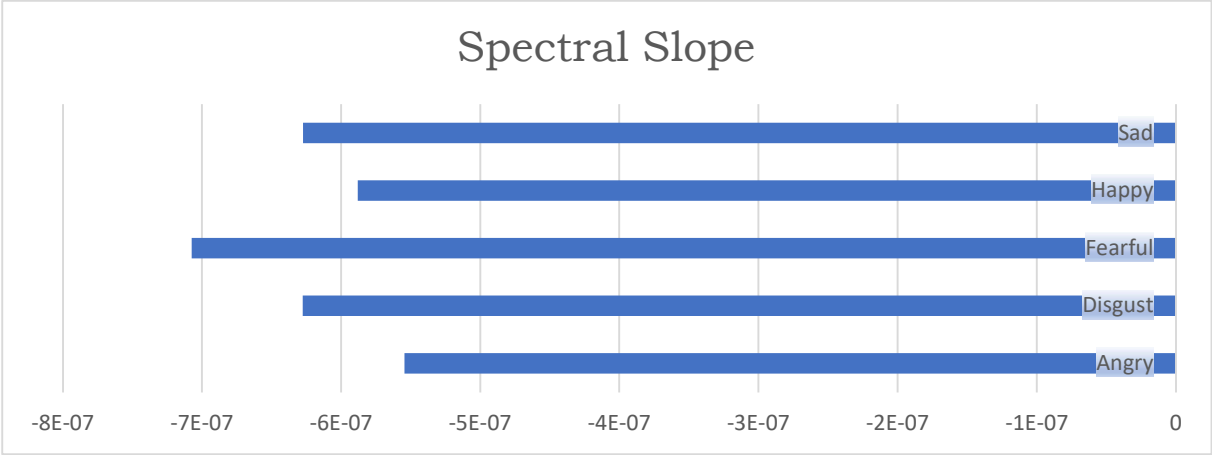

Fig 2.1.58: Visualizing variations in Spectral Slope across Five emotions

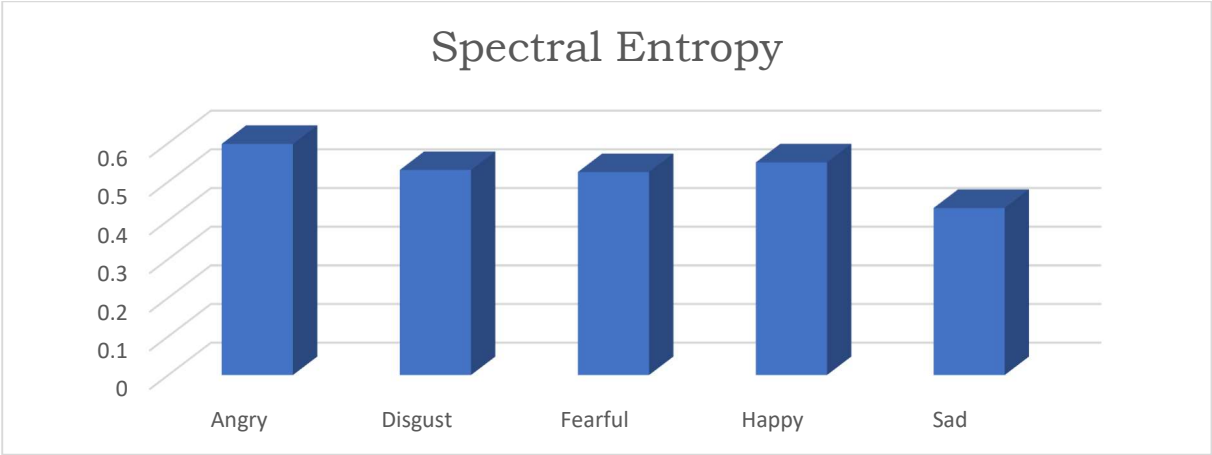

Fig 2.1.59: Visualizing variations in Spectral Entropy across Five emotions

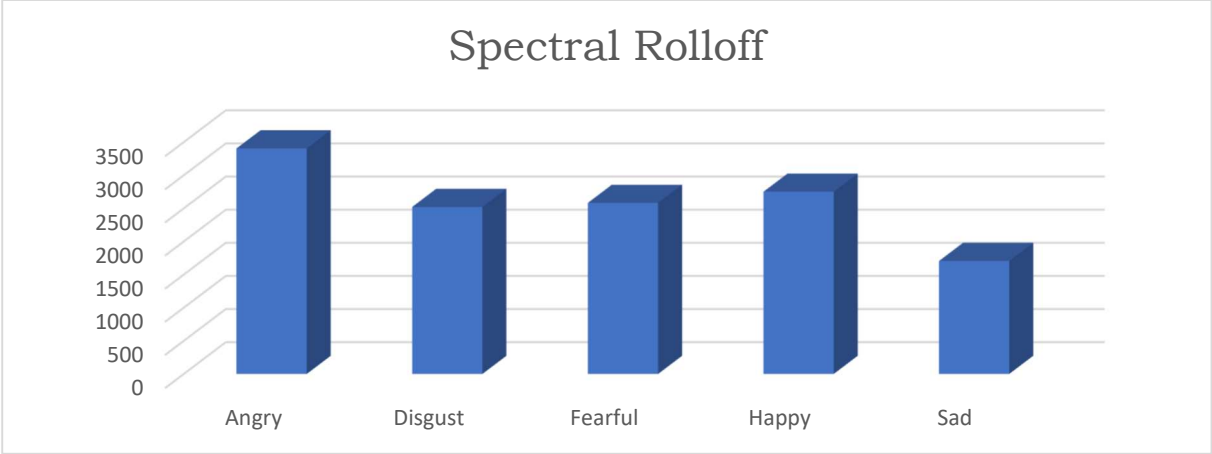

Fig 2.1.60: Visualizing variations in Spectral Rolloff across Five emotions

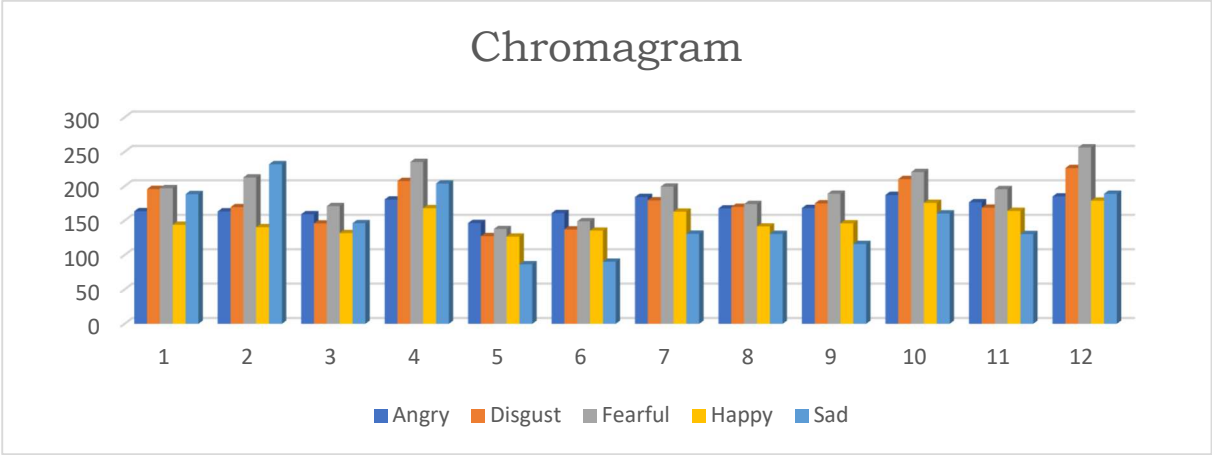

Fig 2.1.61: Visualizing variations in Chromagram across Five emotions

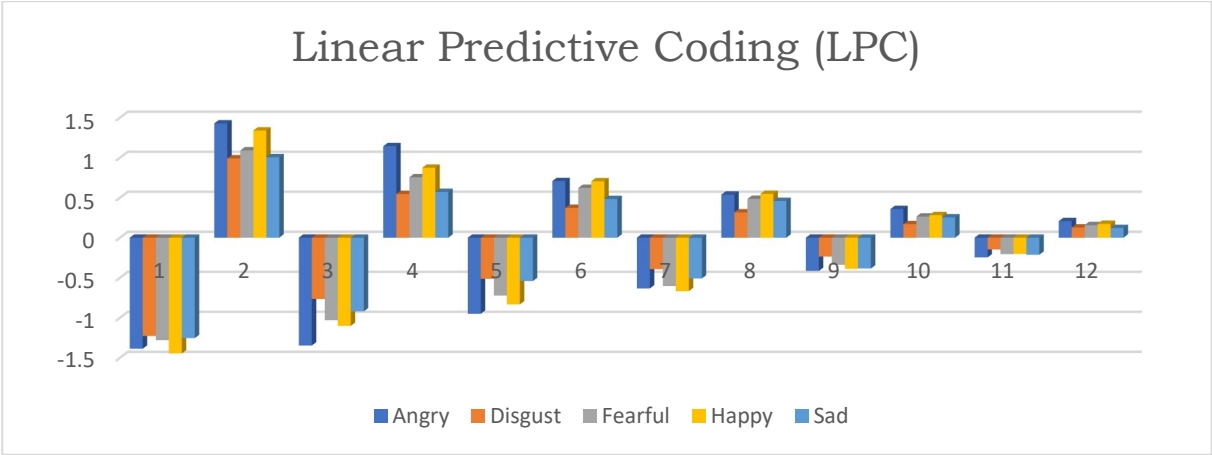

Fig 2.1.62: Visualizing variations in LPC across Five emotions

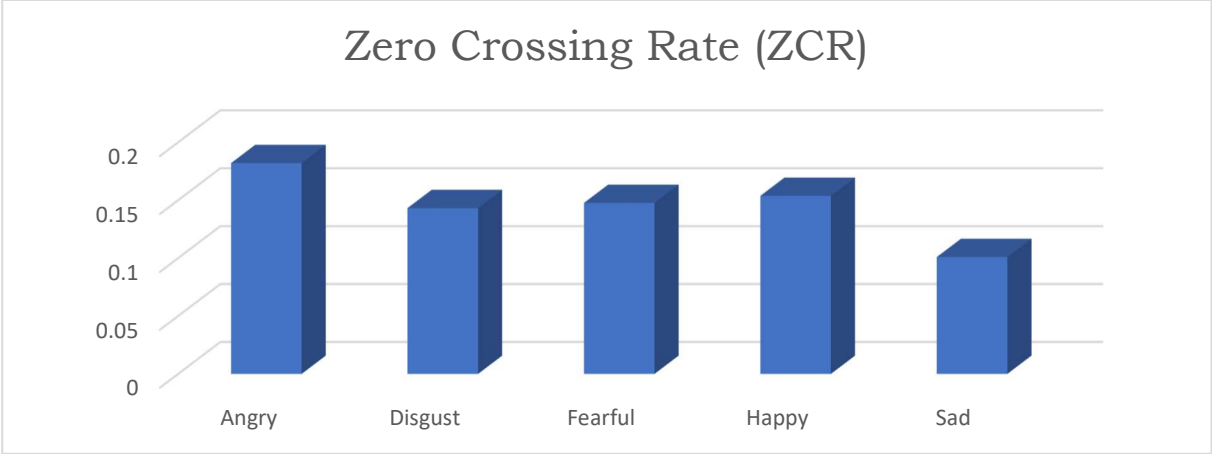

Fig 2.1.63: Visualizing variations in ZCR across Five emotions

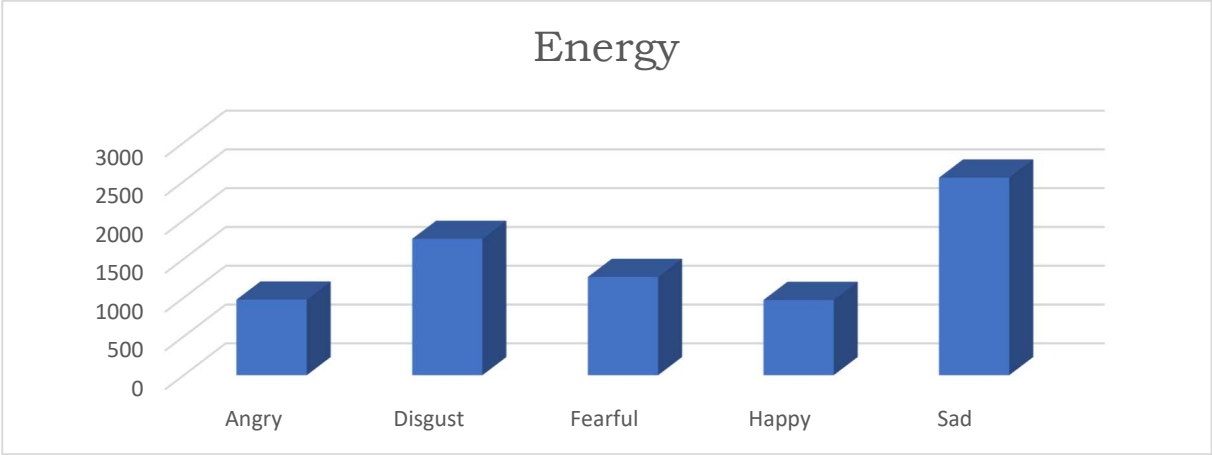

Fig 2.1.64: Visualizing variations in Energy across Five emotions

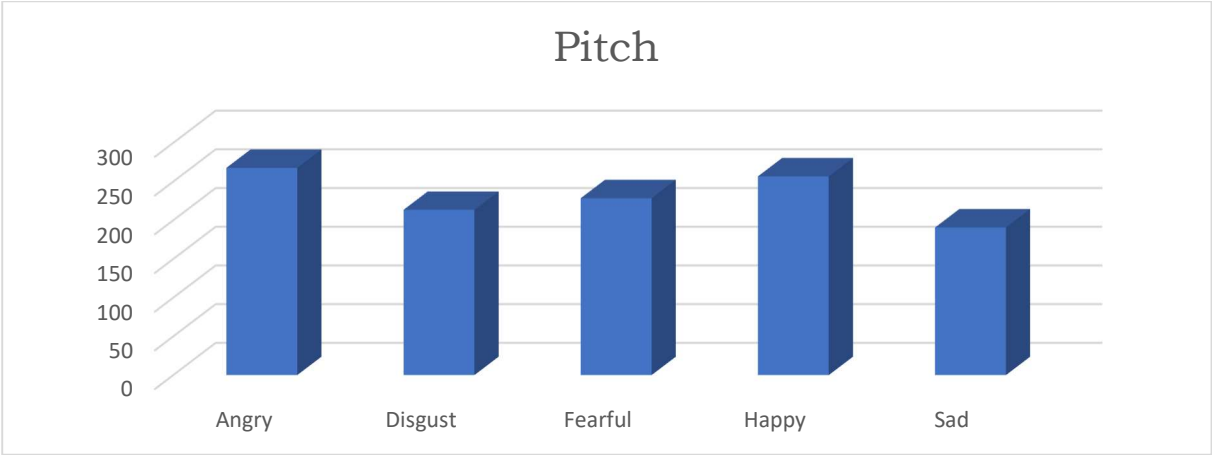

Fig 2.1.65: Visualizing variations in Pitch across Five emotions

Fig 2.1.66: Visualizing variations in Intensity across Five emotions

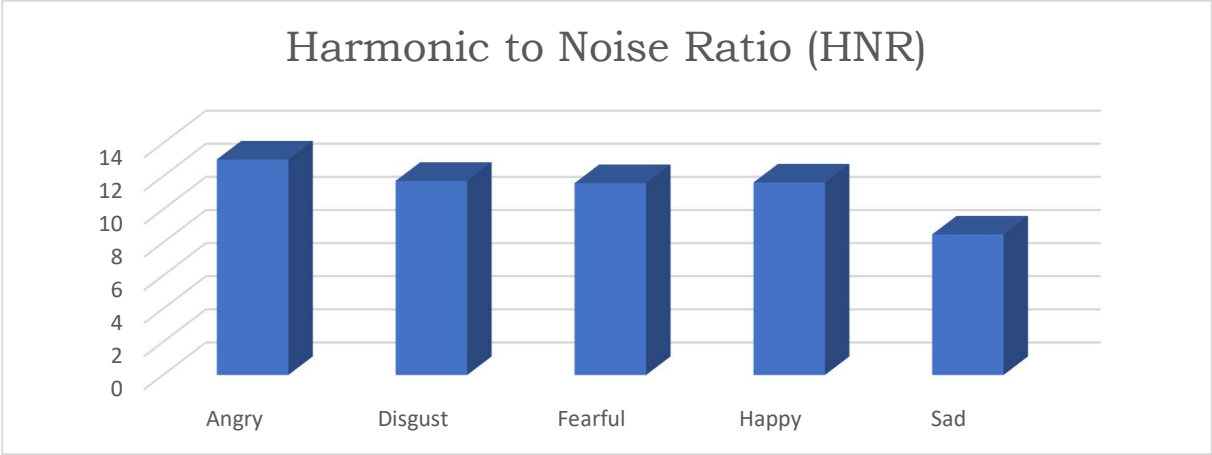

Fig 2.1.67: Visualizing variations in HNR across Five emotions

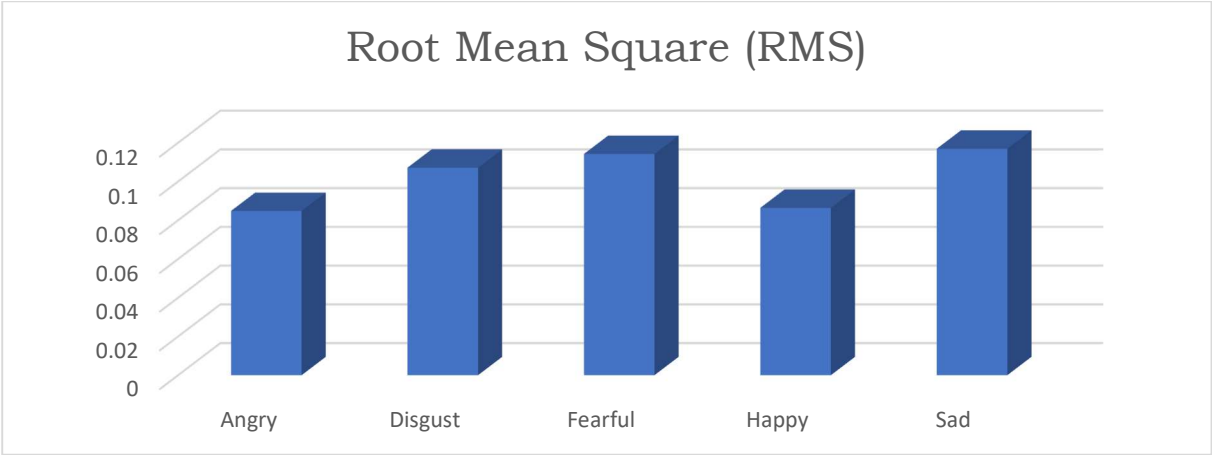

Fig 2.1.68: Visualizing variations in RMS across Five emotions

Visualizing Emotion Feature Variations in the CREMA-D Dataset

We've meticulously crafted plots showcasing the variations in feature distributions across different emotions within the CREMA-D Dataset. Each plot offers a visual narrative, illustrating how the features dynamically fluctuate across various emotional states. Through these graphical representations shown in Fig 2.1.69 – Fig 2.1.85, we provide a comprehensive insight into the nuanced characteristics of emotional expressions encoded within the dataset. These plots serve as invaluable tools for understanding the underlying patterns and trends, aiding in the discernment of distinct feature profiles associated with each emotion.

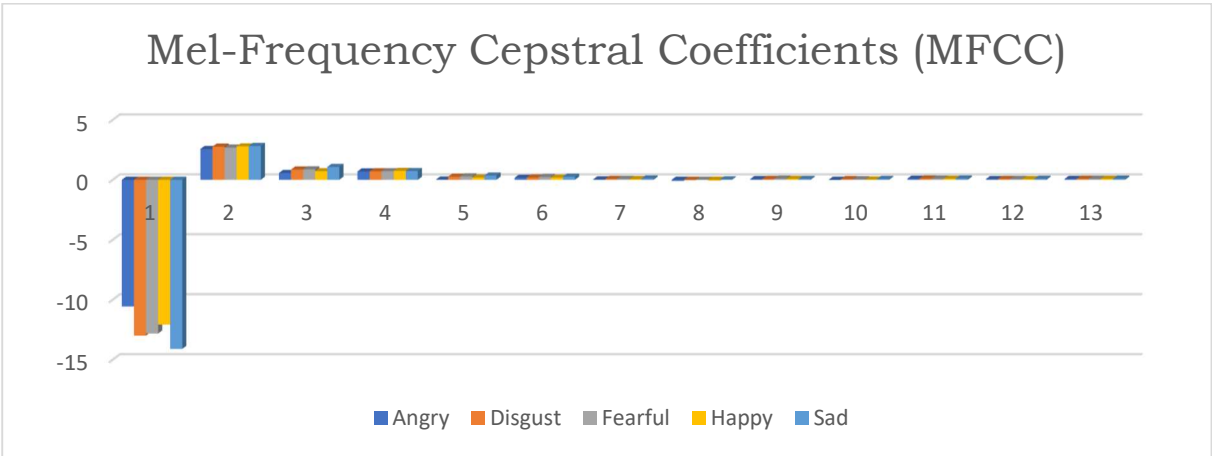

Fig 2.1.69: Visualizing variations in MFCC across Five emotions

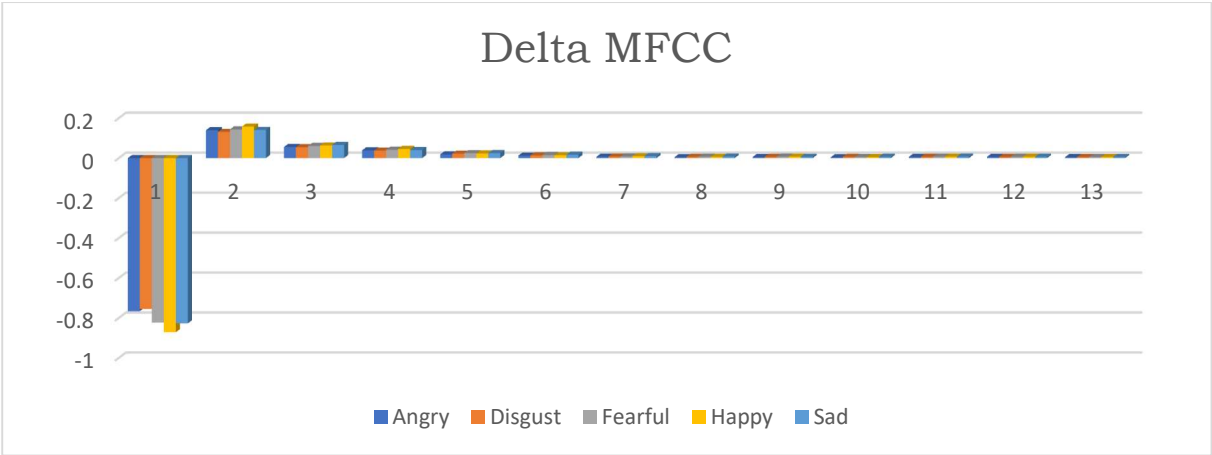

Fig 2.1.70: Visualizing variations in Delta MFCC across Five emotions

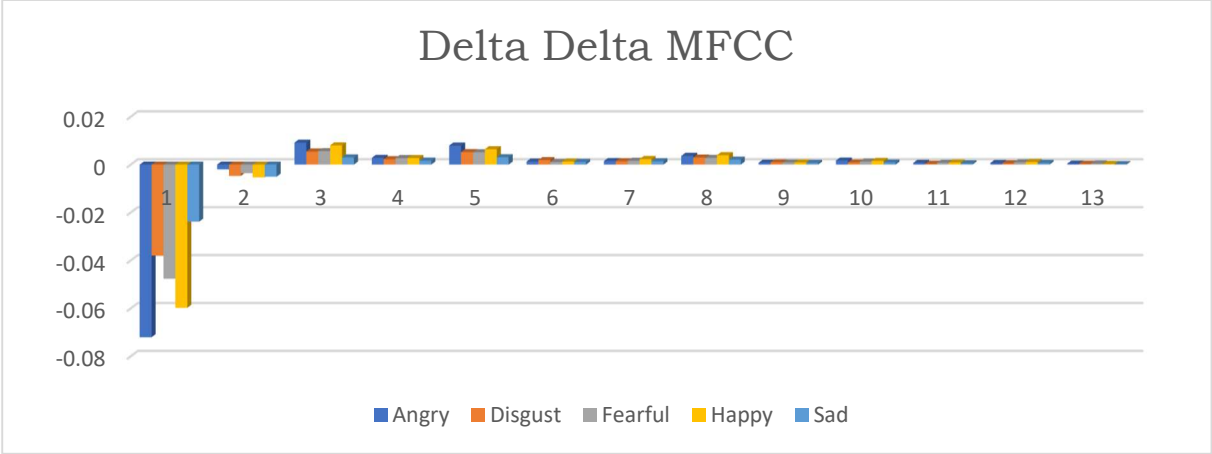

Fig 2.1.71: Visualizing variations in Delta Delta MFCC across Five emotions

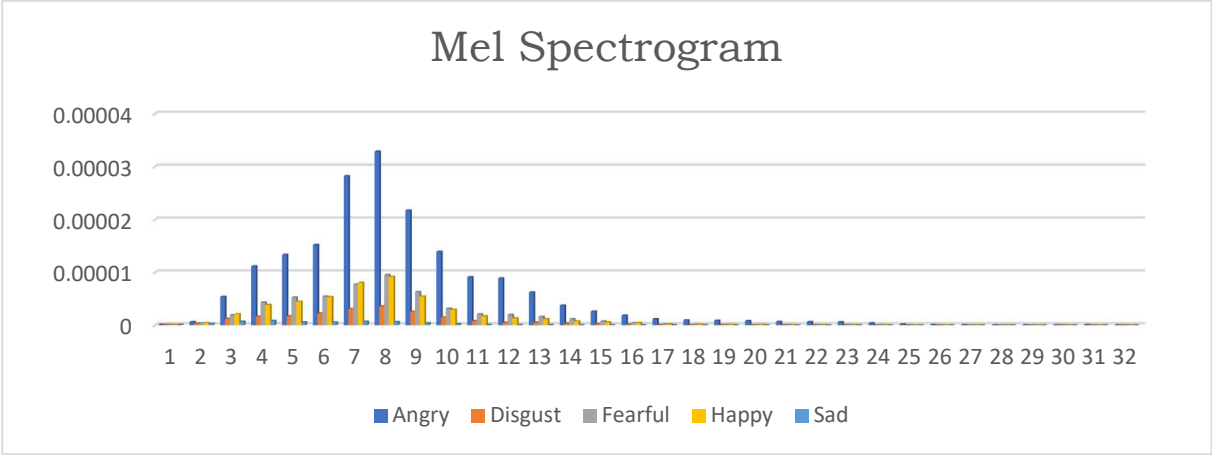

Fig 2.1.72: Visualizing variations in Mel Spectrogram across Five emotions

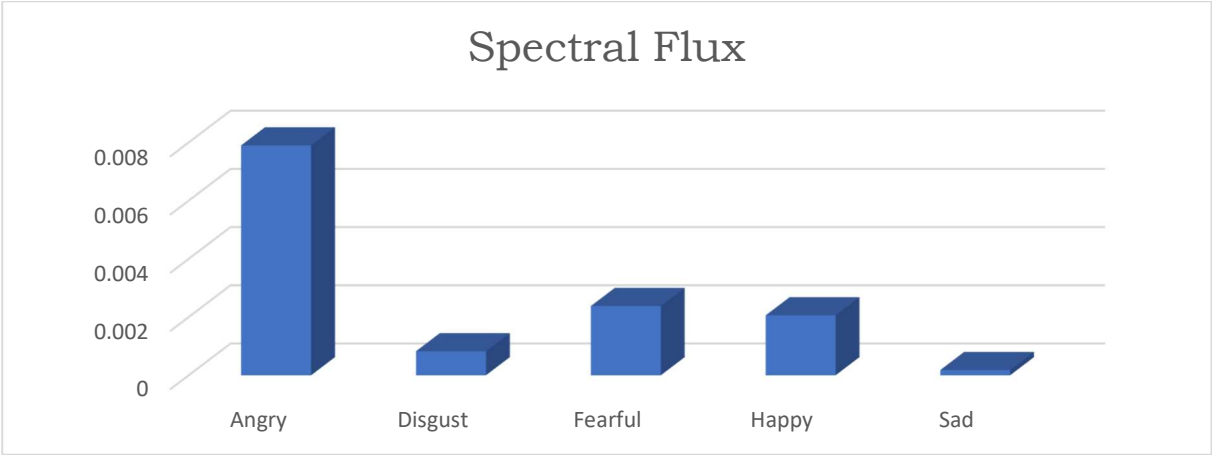

Fig 2.1.73: Visualizing variations in Spectral Flux across Five emotions

Fig 2.1.74: Visualizing variations in Spectral Skewness across Five emotions

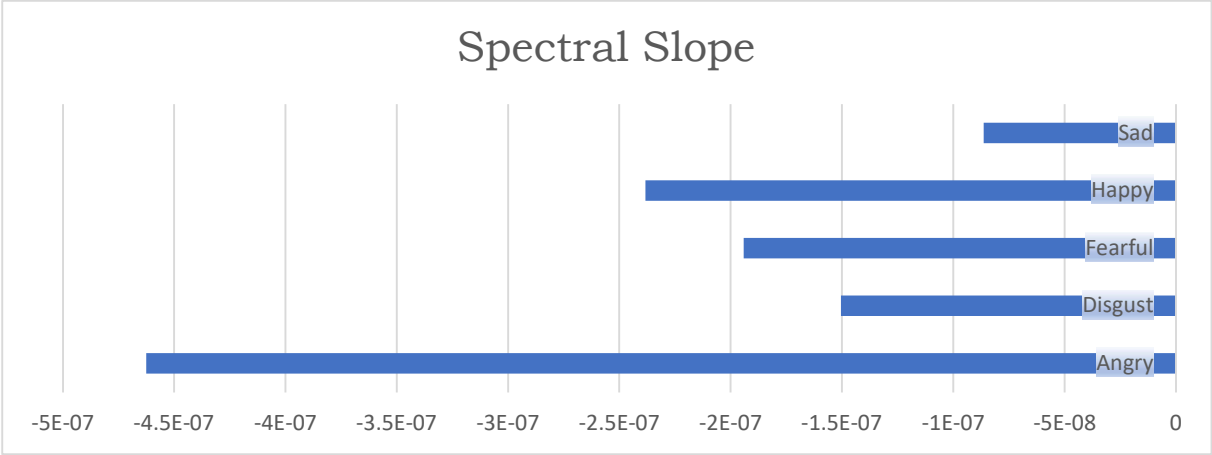

Fig 2.1.75: Visualizing variations in Spectral Slope across Five emotions

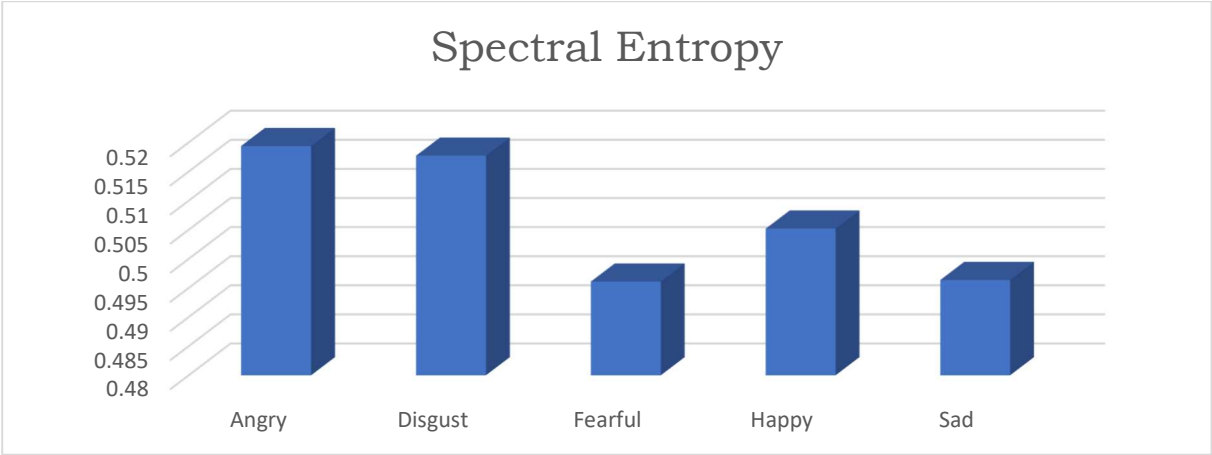

Fig 2.1.76: Visualizing variations in Spectral Entropy across Five emotions

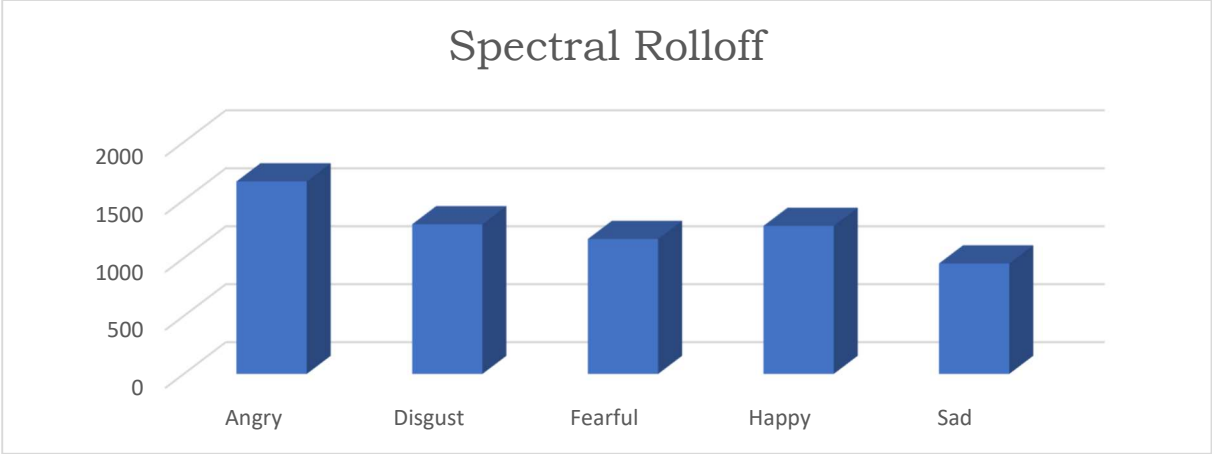

Fig 2.1.77: Visualizing variations in Spectral Rolloff across Five emotions

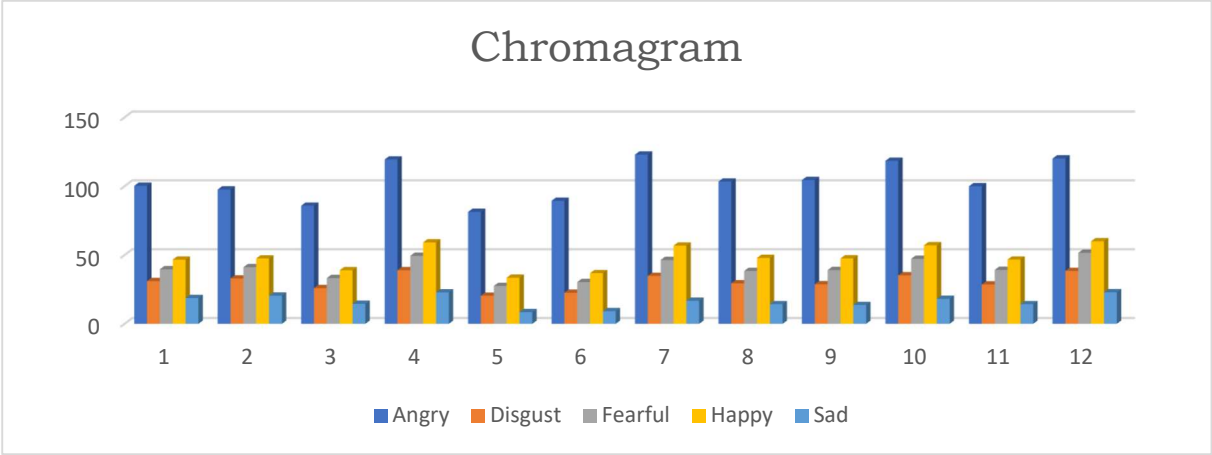

Fig 2.1.78: Visualizing variations in Chromagram across Five emotions

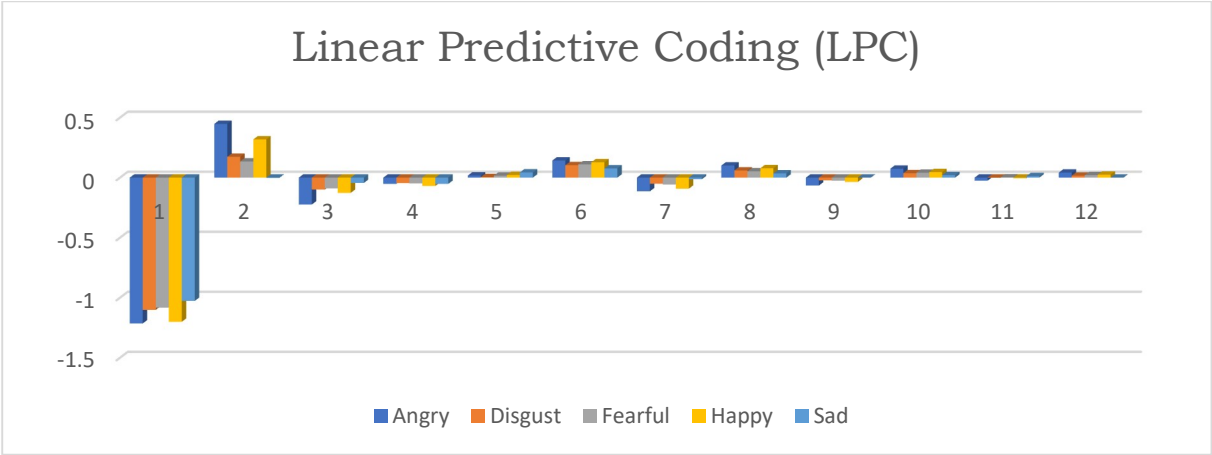

Fig 2.1.79: Visualizing variations in LPC across Five emotions

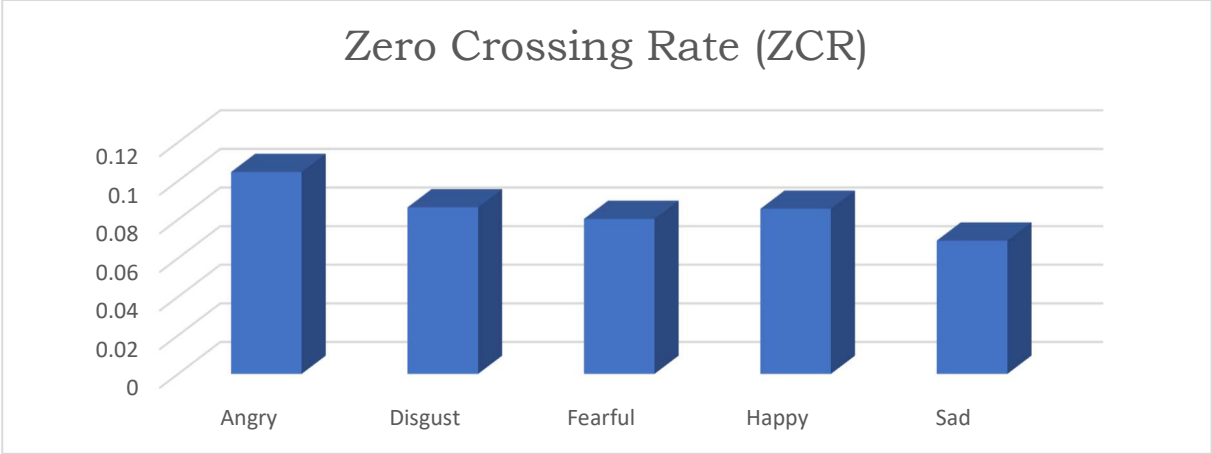

Fig 2.1.80: Visualizing variations in ZCR across Five emotions

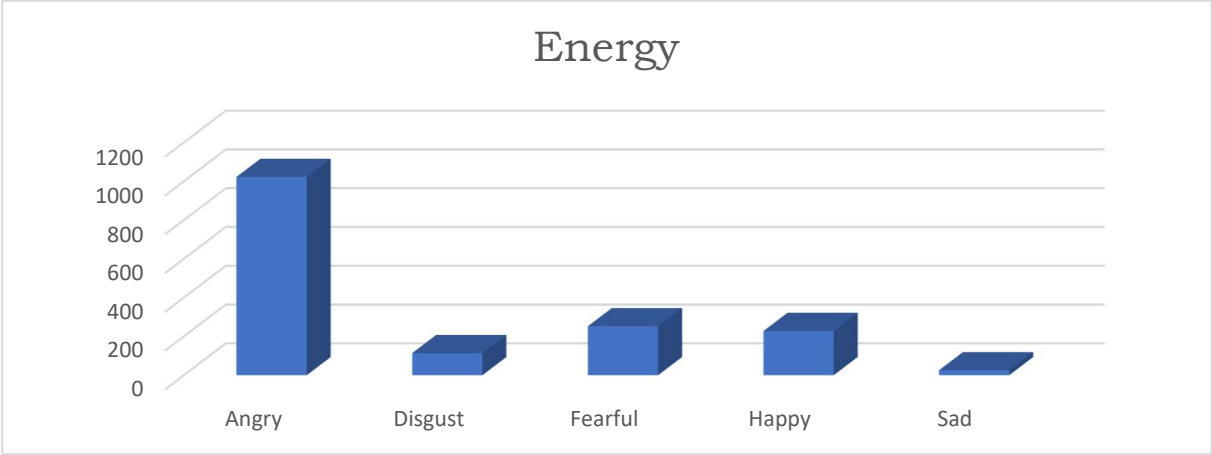

Fig 2.1.81: Visualizing variations in Energy across Five emotions

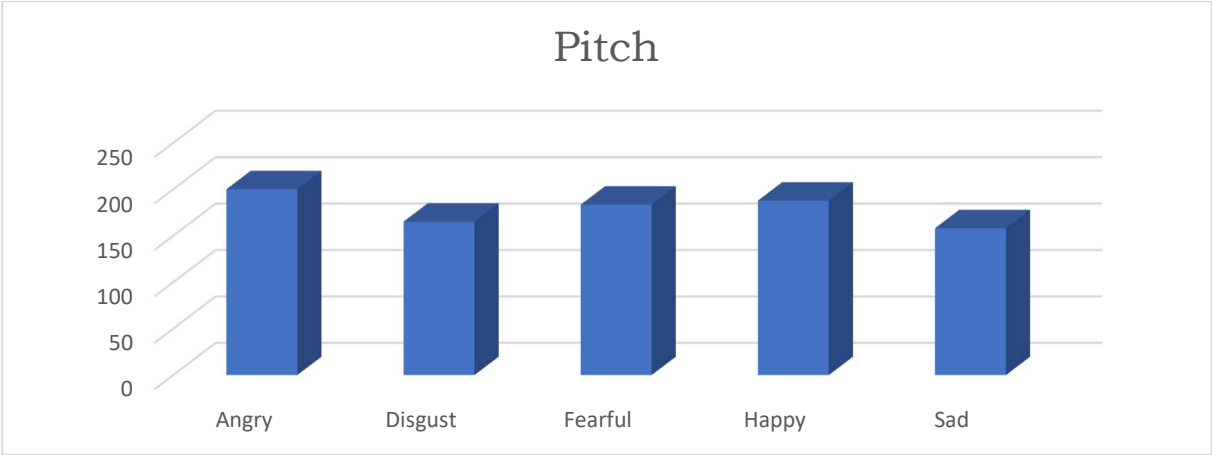

Fig 2.1.82: Visualizing variations in Pitch across Five emotions

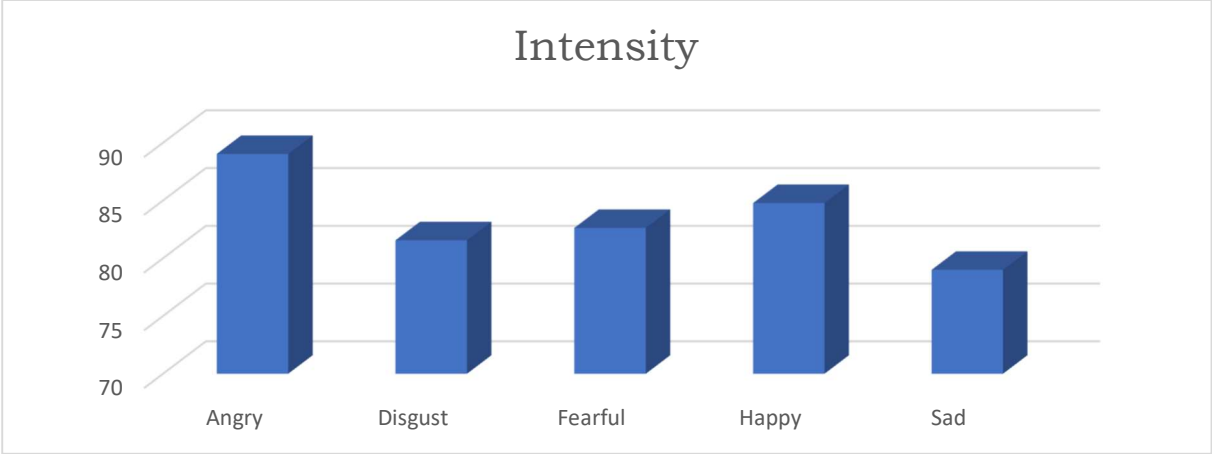

Fig 2.1.83: Visualizing variations in Intensity across Five emotions

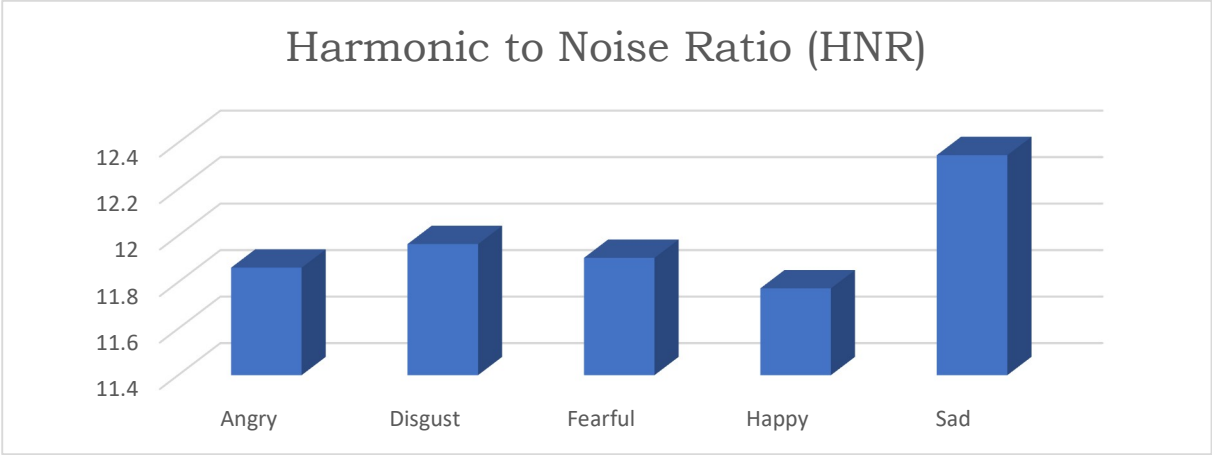

Fig 2.1.84: Visualizing variations in HNR across Five emotions

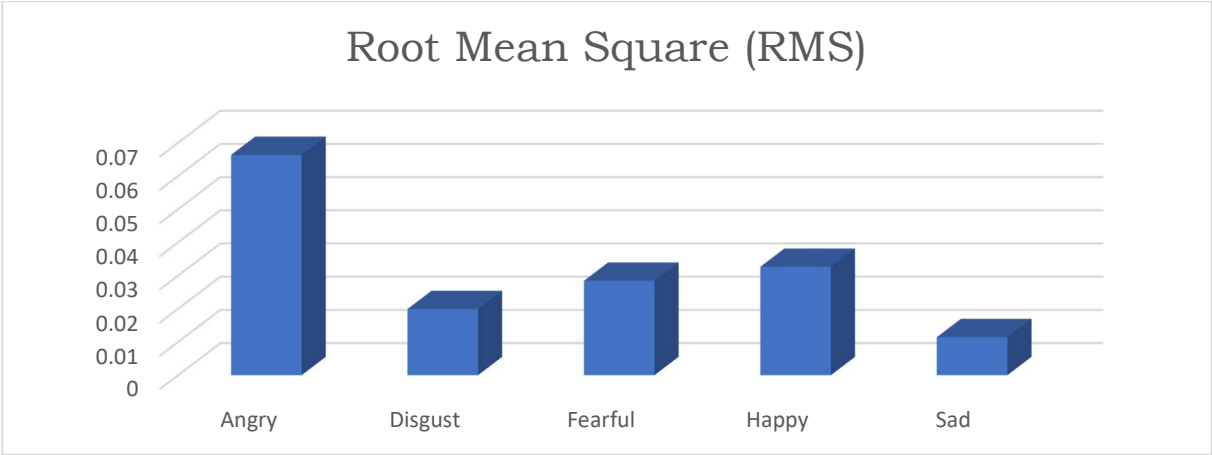

Fig 2.1.85: Visualizing variations in RMS across Five emotions

2.2 Feature Selection

2.2.1 Method – 1

Selected features in the context of feature selection must possess several key properties to be considered effective for subsequent tasks:

- **Relevance**: Selected features should be relevant to the task at hand. They should capture essential information or patterns in the data that are relevant for achieving the desired objective, such as classification or prediction. Irrelevant features can introduce noise and unnecessary complexity, potentially degrading the performance of the model.
- **Discriminative Power**: Selected features should exhibit discriminative power, meaning they should differentiate between different classes or categories in the dataset. Features with high discriminative power contribute significantly to distinguishing between classes, leading to improved classification accuracy and performance.
- **Robustness**: Selected features should be robust to variations or noise in the data. They should remain stable and informative even in the presence of irrelevant changes, such as noise, scale differences, or minor perturbations in the input data. Robust features ensure consistent and reliable performance across different conditions and environments.
- **Independence**: Selected features should be as independent of each other as possible. Redundant or highly correlated features can introduce multicollinearity issues, leading to instability in model estimation and difficulty in interpreting feature contributions. Independent features provide unique and complementary information, enhancing the overall representational capacity of the feature set.
- **Interpretability**: Selected features should be interpretable and understandable. They should offer insights into the underlying characteristics or mechanisms driving the observed patterns in the data. Interpretable features facilitate model understanding, validation, and decision-making, enabling stakeholders to trust and effectively utilize the model's outputs.

In the realm of feature selection, the careful curation of features is paramount to the success of subsequent classification tasks, especially in the context of extensive datasets such as the one under consideration. A judiciously selected set of features not only enhances the accuracy of classification but also ensures the model's robustness against irrelevant variations, facilitates effective discrimination between different classes, and optimizes the utilization of computational resources.

Given the sheer scale of the dataset, attempting to incorporate all available features into the classification process would be impractical and inefficient. Therefore, the process of feature selection assumes critical

importance, necessitating a thorough evaluation of various properties inherent to the features. These properties include their relevance to the classification task, their ability to discern between different classes, their robustness to noise and irrelevant variations, their independence from one another, and their interpretability in the context of the problem domain.

To navigate this intricate process, an effective feature selection algorithm is indispensable. Numerous algorithms have been devised for this purpose, each offering unique advantages and trade-offs. Some notable examples include Kendall's rank coefficient, Principal Component Analysis (PCA), Sequential Forward Search (SFS), Sequential Backward Search (SBS), SelectKBest, and Recursive Feature Elimination (RFE), among others. However, one algorithm that particularly stands out in this context is the Extra Trees Classifier.

The Extra Trees Classifier algorithm distinguishes itself through its remarkable ability to mitigate the risk of overfitting while simultaneously maintaining computational efficiency. Unlike traditional decision tree-based methods, such as Decision Trees or Random Forests, the Extra Trees Classifier introduces randomness during the feature selection process. By doing so, it effectively reduces the likelihood of fitting noise in the data, thereby enhancing its robustness to overfitting.

Moreover, the Extra Trees Classifier's innate efficiency, coupled with its robustness to noisy data and ease of implementation, renders it highly suitable for feature selection tasks, especially in scenarios characterized by large datasets or high-dimensional feature spaces. Its capacity to efficiently navigate through vast amounts of data and extract the most relevant features makes it a preferred choice for optimizing classification performance in complex datasets.

After implementing the Extra Tree Classifier as a feature selection algorithm, we visualized the feature importance for five datasets: RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D. Fig 2.2.1 – Fig 2.2.5 display these feature importances in bar chart form, highlighting the significance of each feature across different emotional datasets. These visualizations provide a clear and comparative insight into which features are most influential in distinguishing emotional expressions within each dataset.

Visualizing Feature Importance with Extra Trees Classifier on RAVDESS Dataset

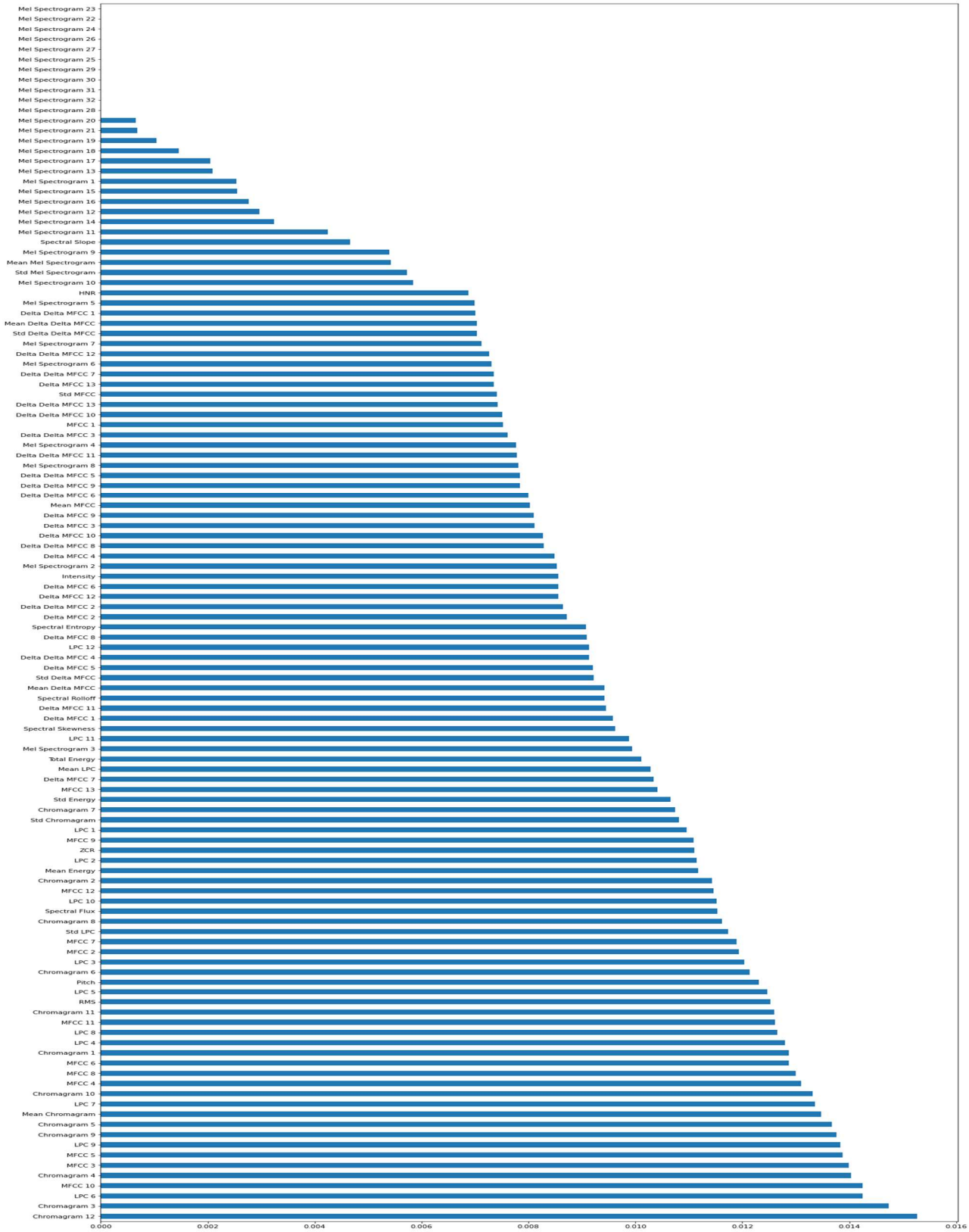

Fig 2.2.1: Feature Importance Bar plot of RAVDESS Dataset

Visualizing Feature Importance with Extra Trees Classifier on TESS Dataset

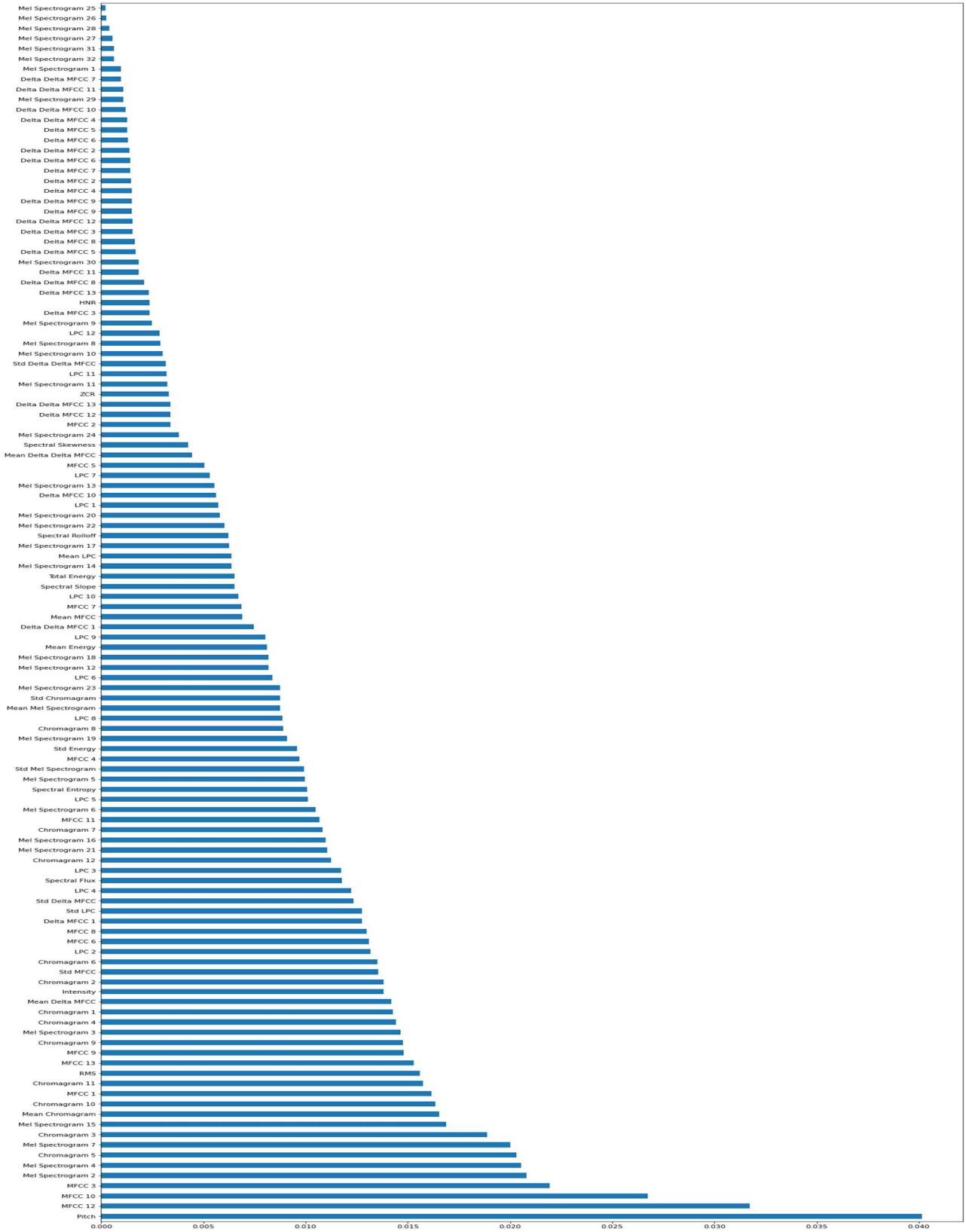

Fig 2.2.2: Feature Importance Bar plot of TESS Dataset

Visualizing Feature Importance with Extra Trees Classifier on SAVEE Dataset

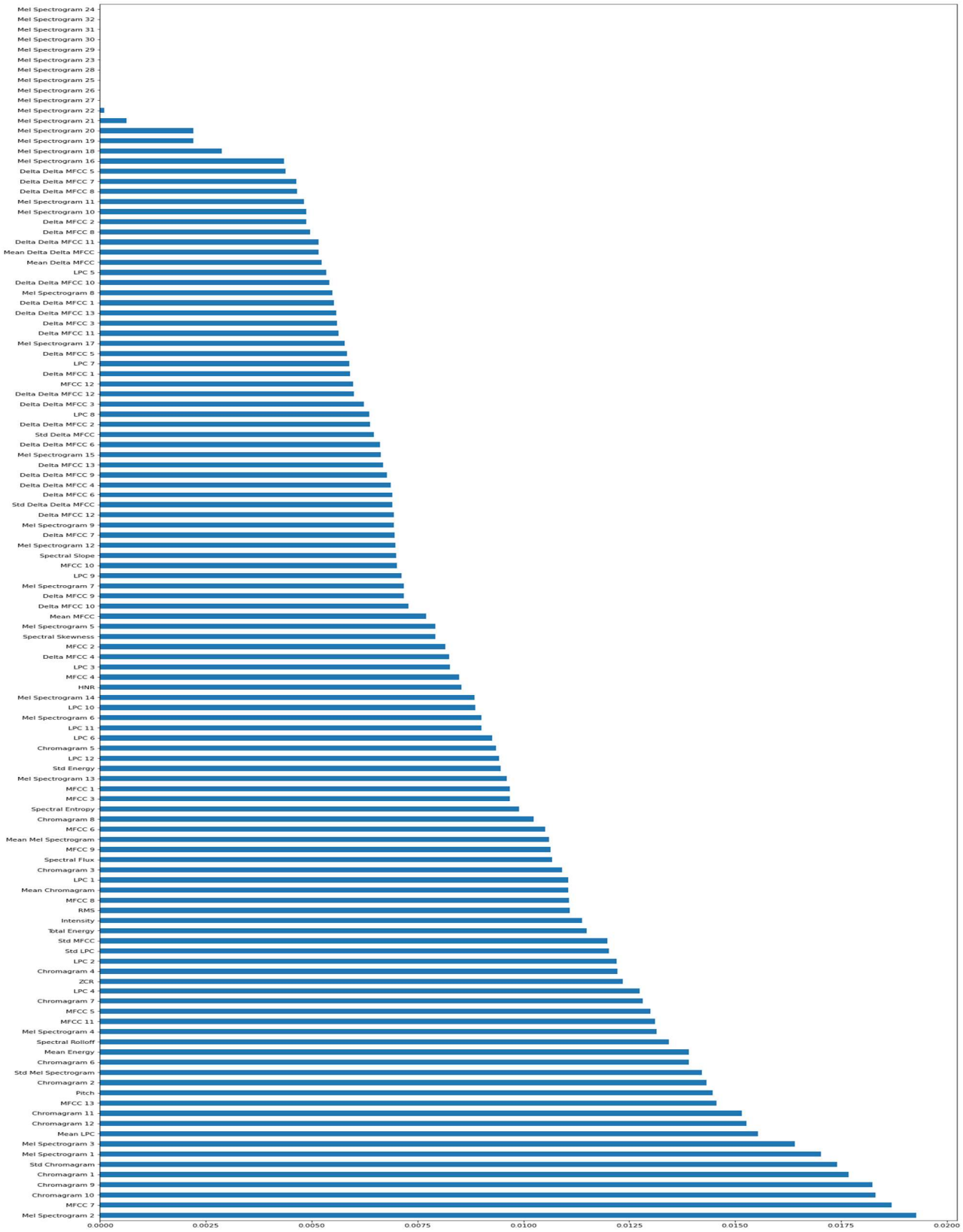

Fig 2.2.3: Feature Importance Bar plot of SAVEE Dataset

Visualizing Feature Importance with Extra Trees Classifier on EmoDB Dataset

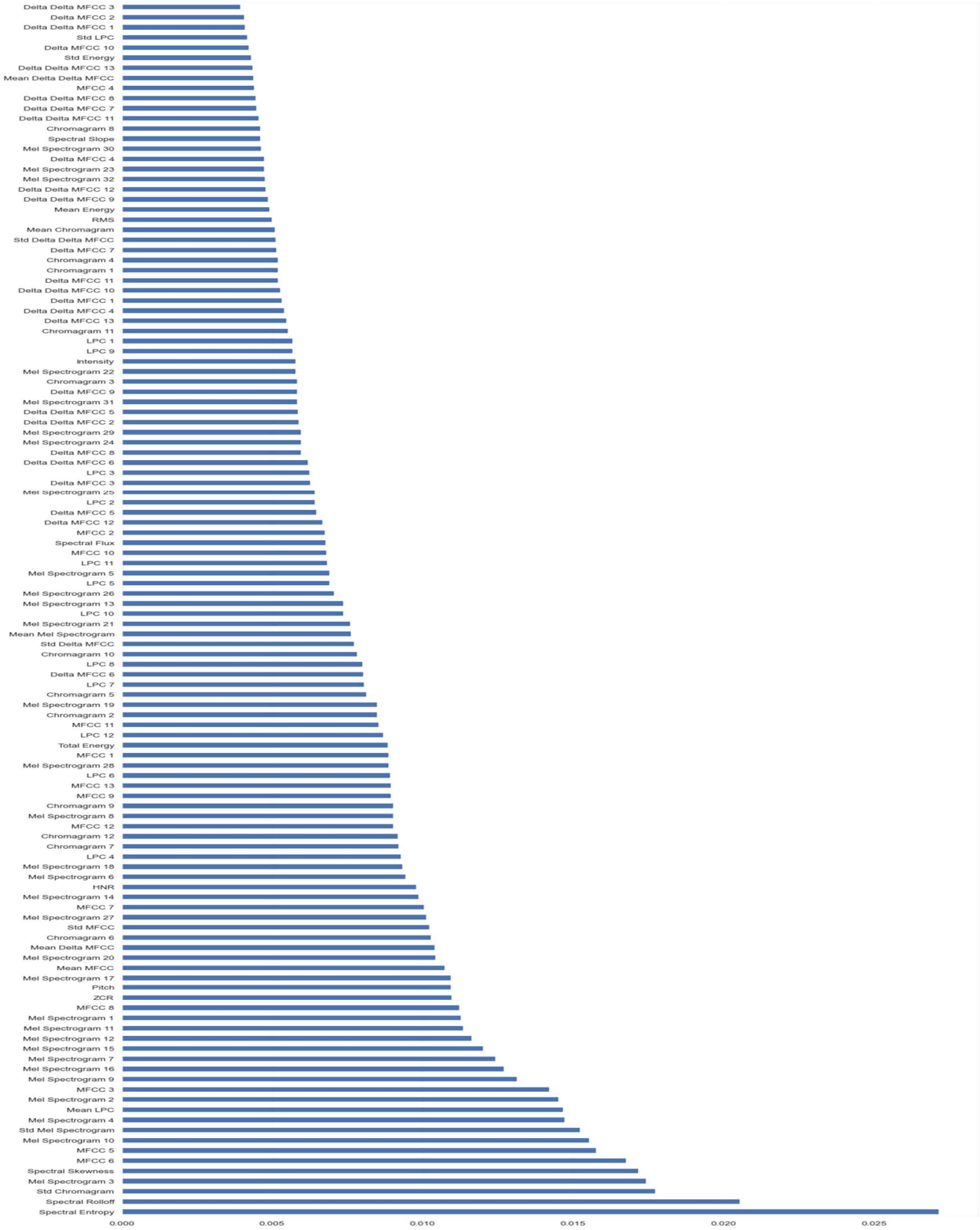

Fig 2.2.4: Feature Importance Bar plot of EmoDB Dataset

Visualizing Feature Importance with Extra Trees Classifier on CREMA-D Dataset

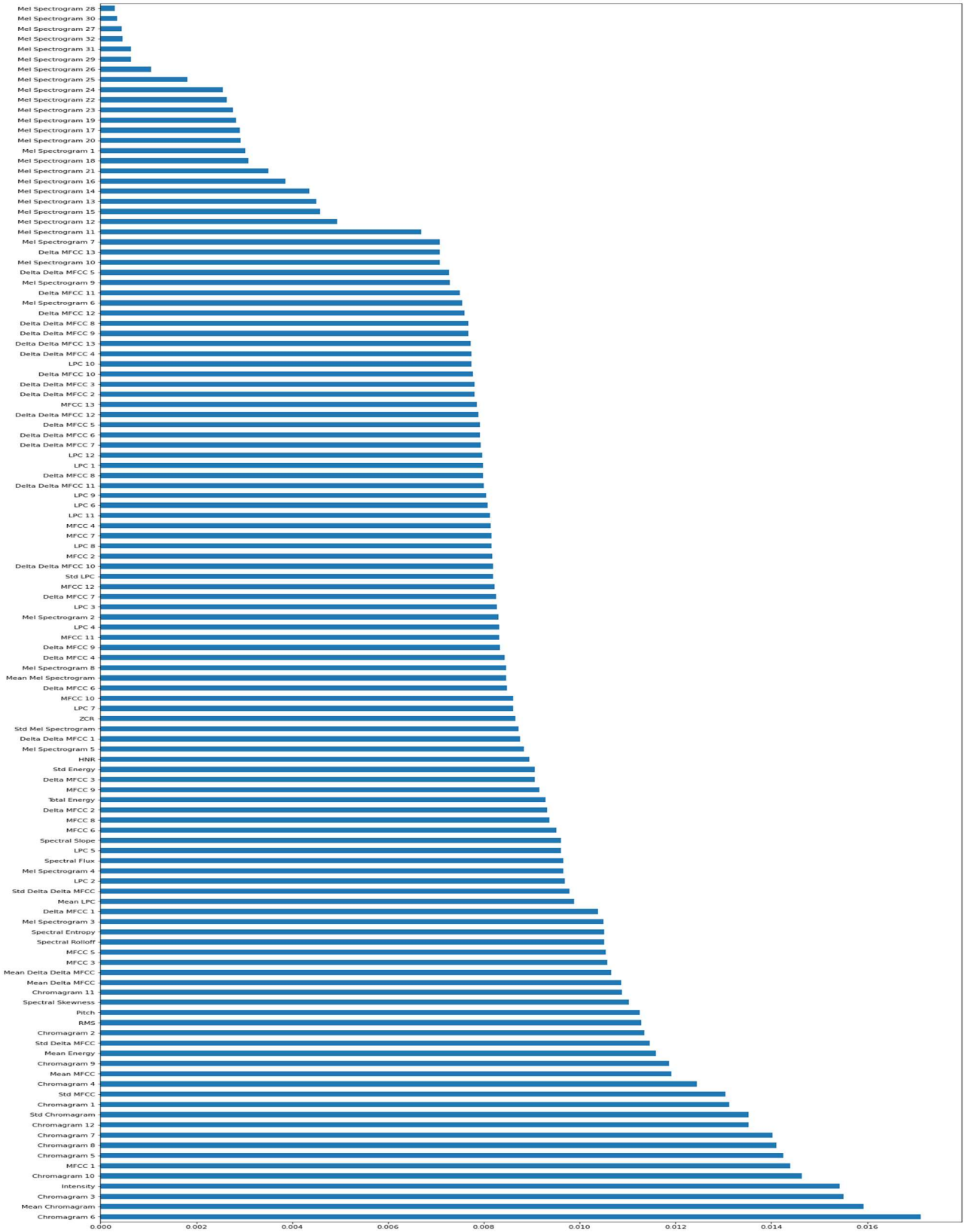

Fig 2.2.5: Feature Importance Bar plot of CREMA-D Dataset

2.2.2 Method – 2

Feature selection is crucial for building effective classification systems for audio data. When selecting features for audio classification tasks, several key properties must be considered to ensure the effectiveness and efficiency of the classification system:

1. Irrelevant Invariance: Good features should exhibit robustness to irrelevant changes such as noise, scale, or bandwidth variations. Incorporating such invariance enhances the system's ability to handle different audio formats, contributing to more effective classification.

2. Discriminative Power: Features should possess discriminative power, meaning they exhibit distinct values within the same class while demonstrating noticeable differences across different classes. This characteristic aids in accurately distinguishing between different audio samples.

3. Uncorrelatedness: Redundancy in feature space should be minimized. Each selected feature should provide unique and non-redundant information about the audio signal. By ensuring that features are uncorrelated with one another, computational efficiency can be improved, and performance can be enhanced.

Considering these properties during feature selection enables the creation of a well-curated feature set, crucial for accurate classification, robustness against irrelevant variations, discrimination between classes, and efficient use of computational resources. Due to the dataset's vastness, utilizing all available features is impractical. Thus, careful feature selection, focusing on properties like invariance to irrelevance, selectivity, and independence, becomes essential.

Several aspect selection algorithms are available, including Sequential Backward Search (SBS), Sequential Forward Search (SFS), Forward Sequential Fluid Search (SFFS), Backward Sequential Fluid Search (SFBS), and more. SFFS stands out for its flexibility and effectiveness. This iterative process allows SFFS to explore the feature space thoroughly, gradually adding features that contribute to improved classification performance. By adapting the feature set based on real-time performance feedback, SFFS ensures optimal feature selection, ultimately leading to enhanced classification accuracy and efficiency.

2.3 Classifiers

Classifiers stand at the forefront of modern data science, embodying the synergy between theoretical principles and practical application. Beyond their fundamental role in categorizing data, classifiers represent a sophisticated amalgamation of algorithms, statistical methods, and domain expertise. They are not merely tools for organizing information but rather intelligent systems capable of discerning intricate patterns and extracting meaningful insights from complex datasets. From predicting customer preferences in e-commerce to identifying fraudulent transactions in finance, classifiers empower organizations across various domains to leverage their data for informed decision-making and strategic planning.

What sets classifiers apart is their adaptability and resilience in the face of uncertainty. Unlike traditional rule-based systems, classifiers operate on a set of adaptable principles that allow them to navigate ambiguous or incomplete data with ease. Through advanced techniques such as ensemble learning, deep learning, and probabilistic modeling, classifiers can effectively handle noisy, unstructured, or high-dimensional datasets, extracting valuable knowledge even from sparse or imperfect information. Moreover, classifiers are inherently scalable, capable of processing vast volumes of data in real-time, making them indispensable tools for applications ranging from online recommendation systems to predictive maintenance in industrial settings.

In essence, classifiers serve as the linchpin between theoretical concepts and actionable insights, bridging the gap between raw data and informed decision-making. By harnessing the power of machine learning algorithms and data-driven methodologies, classifiers empower organizations to extract maximum value from their data assets, driving innovation, efficiency, and competitive advantage in today's fast-paced and data-driven landscape.

There are two kinds of classifiers namely:

- Supervised
- Unsupervised

2.3.1 Supervised Classifiers

The dominant approach in practical Machine Learning is supervised learning, which involves having input variables (x) and an output variable (Y). The objective is to learn the mapping function from the input to the output using an algorithm.

$Y = f(X)$

The aim is to approximate this mapping function accurately, so that when new input data (x) is provided, the model can predict the corresponding output variables (Y). This process is termed Supervised Learning because it resembles a teacher supervising the learning process. The algorithm learns from a training dataset by making predictions on the input data and receiving corrections from the teacher. Learning continues until the algorithm achieves a satisfactory level of performance.

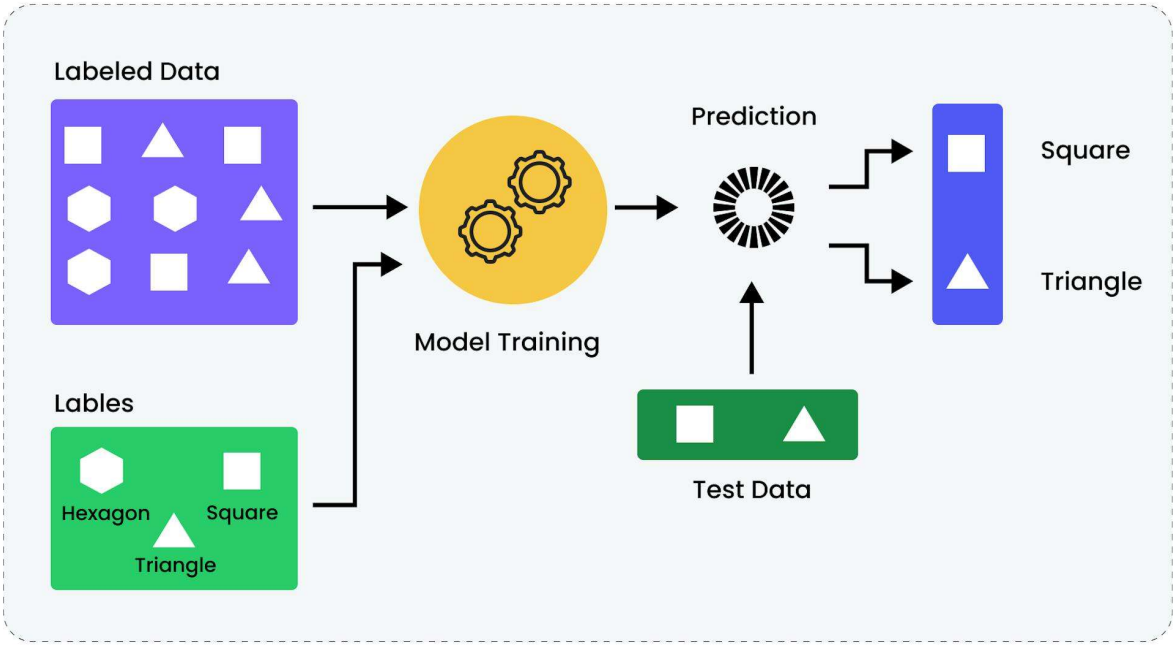

Fig 2.3.1: Supervised Learning classifier

Fig 2.3.1 illustrates a Supervised Learning, a model is trained on data where each example has a corresponding label, like identifying shapes in a dataset. By analyzing this labeled data, the model learns to recognize patterns and predict labels for new, unseen data, ultimately enabling tasks like classifying shapes or making price forecasts.

However, Supervised Learning faces challenges when dealing with complex tasks that involve deep hierarchies. As the number of steps between the input and output increases, the learning task becomes exponentially more difficult. Consequently, Supervised Learning algorithms struggle to learn models with deep hierarchies effectively.

These types of classifying and learning algorithms rely on labeled data to operate. They are trained on labeled training data, where the input is associated with known output labels. This gathered information is then used to classify unlabeled testing data. Using larger training datasets often leads to models with improved predictive power, enabling them to generalize well to new datasets. Supervised learning encompasses two main categories of algorithms:

- **Classification:** This category of supervised learning algorithms is used when the output variable (Y) consists of categorical or discrete classes. The goal is to classify input data into specific classes based on their features. Popular classification algorithms include Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Decision Trees, and Random Forests.
- **Regression:** Regression algorithms are employed when the output variable (Y) is continuous or numerical. The objective is to predict a continuous value based on the input variables. Linear Regression, Polynomial Regression, Neural Networks, and Gradient Boosting are some commonly used regression algorithms.

These algorithms play a crucial role in supervised learning by enabling the classification and prediction of data based on labeled training examples.

2.3.1.1 Support Vector Machines (SVMs)

Support Vector Machines (SVMs) are powerful supervised learning algorithms used for classification and regression tasks. In classification, SVMs aim to find the optimal hyperplane that best separates the data points belonging to different classes in the feature space. This hyperplane is chosen such that the margin, which is the distance between the hyperplane and the nearest data points (called support vectors) from each class, is maximized.

Here's the mathematical formulation for linear SVMs in the context of binary classification:

Given a training dataset,

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

where x_i represents the feature vector for the i-th data point, y_i is its corresponding class label , and N is the number of data points, the goal is to find the optimal

$$(y_i \in \{-1, +1\})$$

hyperplane w.x + b = 0 that separates the data points.

The decision function for SVM can be written as: f(x)=w·x+b
where w is the weight vector perpendicular to the hyperplane, x is the input feature vector, and b is the bias term.

$$\frac{1}{2}||w||^2$$

To maximize the margin, SVM aims to solve the following optimization problem:

minimize
subject to the constraints:

$$yi(w \cdot xi+b) \geq 1 \text{for all} i=1,2,...,N$$

These constraints ensure that all data points lie on the correct side of the decision boundary, with a margin of at least 1.

In cases where the data is not linearly separable, SVMs use a technique called the kernel trick to map the input data into a higher-dimensional space where it becomes linearly separable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels. Once the optimization problem is solved, the decision boundary is determined by the hyperplane equation $w.x + b = 0$, where data points closest to the hyperplane are the support vectors. Classification of new data points is based on which side of the hyperplane they fall.

SVMs have several advantages, including their ability to handle high-dimensional data, effectiveness in cases with a clear margin of separation, and robustness against overfitting. However, they can be sensitive to the choice of the regularization parameter *C* and the kernel function, and their computational complexity increases with the size of the dataset.

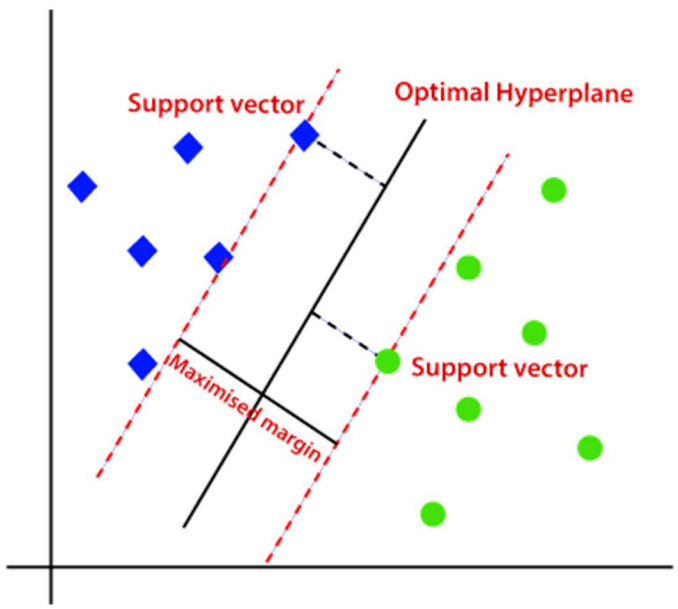

Fig 2.3.1.1: Support Vector Machines (SVMs)

The image shown in Fig 2.3.1.1 illustrates the concept of a Support Vector Machine (SVM) classifier. It shows two classes of data points (green circles and blue squares) separated by an optimal hyperplane, which is the decision boundary that maximizes the margin between the closest points of each class, known as support vectors. The margin is the distance between the hyperplane and the nearest data points from either class, ensuring the classifier's robustness and optimal separation of the classes.

2.3.1.2 Naïve Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, with the "naive" assumption that features are conditionally independent given the class label. Despite its simplicity, Naive Bayes often performs remarkably well in various classification tasks, especially in text classification and spam filtering. Let's delve into its workings in detail:

1. Bayes' Theorem: At the core of Naive Bayes is Bayes' theorem, a fundamental concept in probability theory. It describes the probability of a hypothesis given the evidence, and it's mathematically represented as:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Where:

- P(A|B) is the probability of event A occurring given that event B has occurred.
- P(B|A) is the probability of event B occurring given that event A has occurred.
- P(A) and P(B) are the probability of event B & A occurring independently.

2. Classification with Naive Bayes: In a classification task, Naive Bayes calculates the probability of each class label given the input features and then predicts the class with the highest probability. This is represented as:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(x_1|y).P(x_2|y) \dots P(x_n|y).P(y)}{P(x_{1)}.P(x_2) \dots P(x_n)}$$

Where:

- $P(y|x_1, x_2, \dots, x_n)$ is the probability of class y given the input features x1,x2,….,xn.
- $P(x_1|y)$ is the conditional probability of feature xi given class y.
- $P(y)$ is the prior probability of class y.
- $P(xi)$ is the prior probability of class xi.

3. Naive Assumption: The "naive" part of Naive Bayes comes from the assumption that features are conditionally independent given the class label. Mathematically, this is represented as:

$$P(x_i|y, x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$$

This assumption greatly simplifies the calculations because it reduces the number of parameters that need to be estimated.

4. Types of Naive Bayes: There are different variants of Naive Bayes classifiers, including:

- Gaussian Naive Bayes: Assumes that features follow a Gaussian distribution.
- Multinomial Naive Bayes: Suitable for features that represent counts or frequencies.
- Bernoulli Naive Bayes: Assumes binary-valued features.

5. Training and Prediction: To train a Naive Bayes classifier, you calculate the prior probabilities and conditional probabilities from the training data. During prediction, you use these probabilities to calculate the likelihood of each class label given the input features and then select the class with the highest probability.

In summary, Naive Bayes is a simple yet effective classification algorithm that leverages Bayes' theorem and the assumption of feature independence to make predictions. Despite its simplicity, it often performs well in practice, especially for text classification tasks with high-dimensional feature spaces.

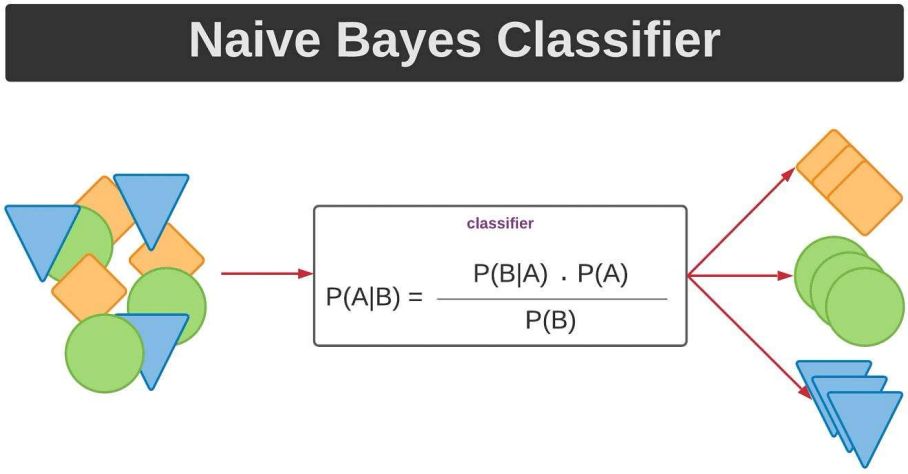

Fig 2.3.1.2: Naïve Bayes

The image shown in Fig 2.3.1.2 illustrates the Naïve Bayes Classifier, which applies Bayes' Theorem to classify data points based on their features. It shows a formula for calculating the probability P(A|B) using the probabilities P(B|A), P(A), and P(B). The classifier processes the input shapes (triangles, squares, circles) and categorizes them into distinct groups based on their calculated probabilities.

2.3.1.3 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and value below the threshold values tends to 0.

Assumptions for Logistic Regression:
The dependent variable must be categorical in nature.
The independent variable should not have multi-collinearity.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

1 Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

2 Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

3 Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

In our Project we have used 5 emotion classes so here multinomial logistic regression is used.

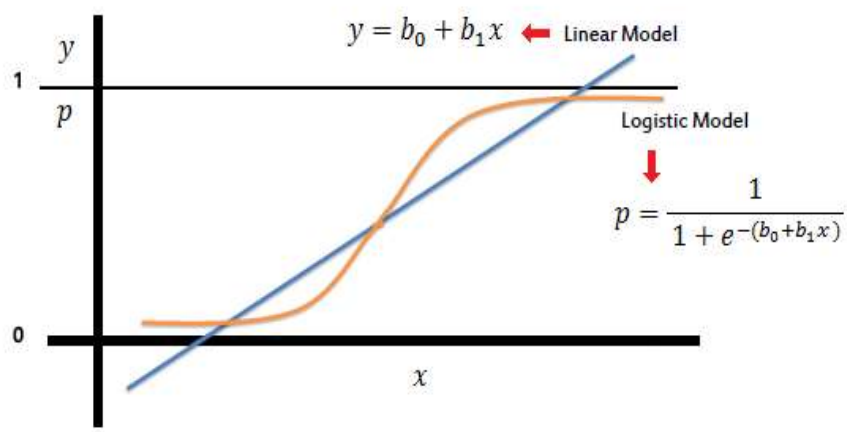

Fig 2.3.1.3: Logistic Regression

The image shown in Fig 2.3.1.3 illustrates the concept of Logistic Regression, showing the difference between a linear model and a logistic model. The linear model is represented by the equation $y = b_0 + b_1 x$, while the logistic model uses the sigmoid function $p = \frac{1}{1+ e^{-(b_0+b_1 x)}}$. The logistic model, depicted by the S-shaped curve, maps predicted values to probabilities between 0 and 1, effectively handling binary classification problems.

2.3.1.4 Random Forest Algorithm

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance. In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

Ensemble learning models work just like a group of diverse experts teaming up to make decisions – think of them as a bunch of friends with different strengths tackling a problem together. Picture it as a group of friends with different skills working on a project. Each friend excels in a particular area, and by combining their strengths, they create a more robust solution than any individual could achieve alone. Similarly, in ensemble learning, different models, often of the same type or different types, team up to enhance predictive performance. It's all about leveraging the collective wisdom of the group to overcome individual limitations and make more informed decisions in various machine learning tasks. Some popular ensemble models include- XG Boost, AdaBoost, Light GBM, Random Forest, Bagging, Voting etc.

Bagging is an ensemble learning model, where multiple week models are trained on different subsets of the training data. Each subset is sampled with replacement and prediction is made by averaging the prediction of the week models for regression problem and considering majority vote for classification problem.

Boosting trains multiple based models sequentially. In this method, each model tries to correct the errors made by the previous models. Each model is trained on a modified version of the dataset, the instances that were misclassified by the previous models are given more weight. The final prediction is made by weighted voting.

The random Forest algorithm works in several steps which are discussed below:

1 Ensemble of Decision Trees: Random Forest leverages the power of ensemble learning by constructing an army of Decision Trees. These trees are like individual experts, each specializing in a particular aspect of the data. Importantly, they operate independently, minimizing the risk of the model being overly influenced by the nuances of a single tree.

2 Random Feature Selection: To ensure that each decision tree in the ensemble brings a unique perspective, Random Forest employs random feature selection. During the training of each tree, a random subset of features is chosen. This randomness ensures that each tree focuses on different aspects of the data, fostering a diverse set of predictors within the ensemble.

3 Bootstrap Aggregating or Bagging: The technique of bagging is a cornerstone of Random Forest's training strategy which involves creating multiple bootstrap samples from the original dataset, allowing instances to be sampled with replacement. This results in different subsets of data for each decision tree, introducing variability in the training process and making the model more robust.

4 Decision Making and Voting: When it comes to making predictions, each decision tree in the Random Forest casts its vote. For classification tasks, the final prediction is determined by the mode (most frequent prediction) across all the trees. In regression tasks, the average of the individual tree predictions is taken. This internal voting mechanism ensures a balanced and collective decision-making process.

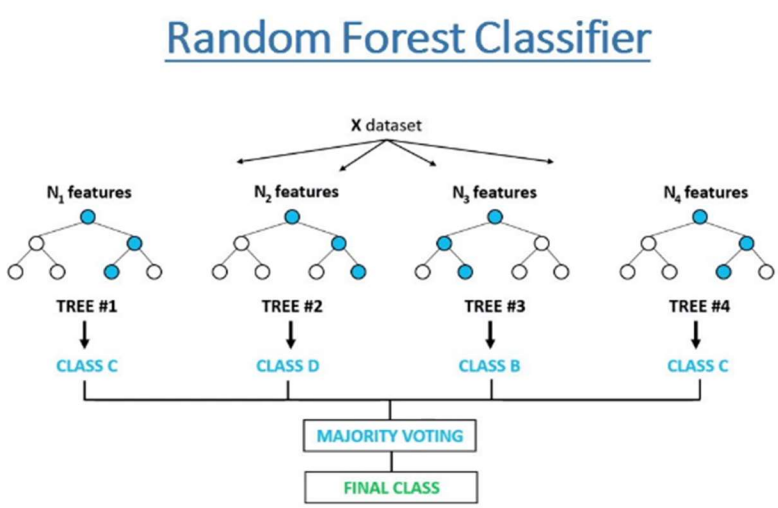

Fig 2.3.1.4: Random Forest Classifier

The image shown in Fig 2.3.1.4 illustrates a Random Forest Classifier, an ensemble learning method that combines multiple decision trees to classify data. Each tree is trained on a different subset of features (N1, N2, N3, N4) from the dataset (X). The final class is determined through majority voting among the individual trees' classifications, improving accuracy and reducing overfitting compared to a single decision tree.

2.3.1.5 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is a sort of artificial neural network that has multiple layers of connected nodes (also known as neurons) and is frequently used for different machine-learning tasks, including classification and regression. An overview of an MLP's structure and operation is provided below:

Architecture

Input Layer: The input layer is made up of neurons that directly take in the dataset's features. Each neuron in the input layer represents a feature, and the input layer's total number of neurons is equal to the dataset's total number of features.

Hidden Layer: One or more hidden layers may exist between the input and output layers. The number of neurons in each hidden layer, which is a hyperparameter that you can choose, varies depending on the hidden layer. In order to recognize intricate patterns in the data, these hidden layers are essential.

Output Layer: The final predictions or outputs are generated by the output layer using the data processed in the hidden levels. The task's requirements determine how many neurons are present in the output layer:

There is often only one neuron that generates a probability score for binary categorization.

There are as many neurons involved in multi-class classification as there are classes, and each neuron generates a probability score for a particular class.

Working

Initialization: Set all of the network's neurons' weights (W) and biases (B) to their initial values. Usually, modest random numbers are used as initial values for these parameters.

Forward Propagation: Input data is passed through the network repeatedly during training. Each neuron in a layer takes in the weighted total of the inputs from the layer before it, applies an activation function, and sends the outcome to the layer after it.

Loss Calculation: A loss (error) is computed by comparing the network's output to the actual goal values. Mean Squared Error (MSE) for regression and Cross-Entropy for classification are examples of common loss functions.

Backpropagation: In order to reduce the loss, the network modifies its biases and weights. The backpropagation algorithm accomplishes this by calculating gradients of the loss with respect to each network parameter.

Training: The forward propagation, loss estimation, and backpropagation processes are iterated across a number of iterations (epochs) until the model converges to a solution. A hyperparameter that can be modified is the learning rate and the number of iterations.

Prediction: By using forward propagation with the honed weights and biases, the MLP may be trained to make predictions on new, unobserved data.

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $(\cdot):Rm \rightarrow Ro$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X=x1,2,...,xm$ and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 1 shows a one hidden layer MLP with scalar output.

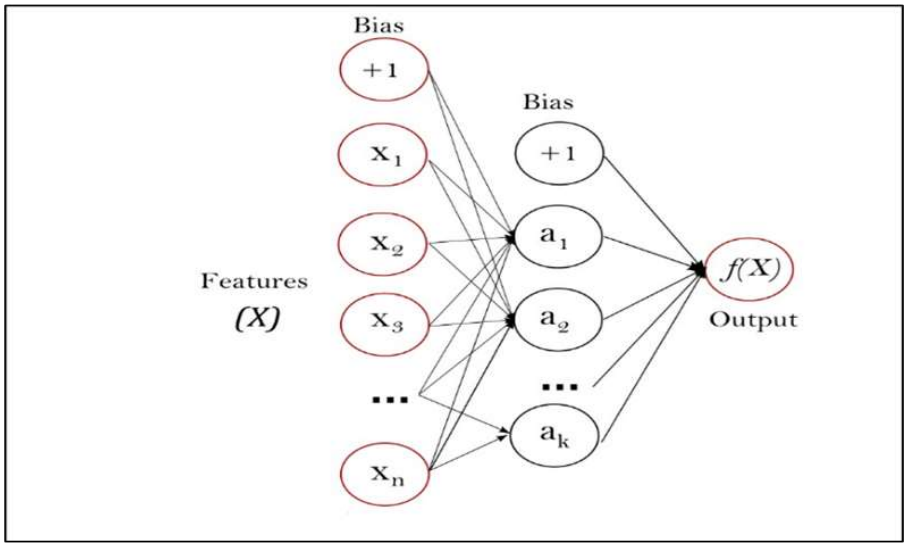

Fig 2.3.1.5: Multi-Layer Perceptron

The image shown in Fig 2.3.1.5 depicts a Multi-Layer Perceptron (MLP), a type of artificial neural network used for supervised learning tasks. It shows the input layer with features (X1, X2, ..., Xn) and bias terms connected to a hidden layer of neurons (a1, a2, ..., ak). Each neuron in the hidden layer processes inputs through weighted connections, and the output layer aggregates these results to produce the final output f(X). Bias nodes ensure the network can shift activation functions.

The module contains the public attributes coefs_ and intercepts_. coefs_ is a list of weight matrices, where weight matrix at index i represents the weights between layer i and layer i+1. intercepts_ is a list of bias vectors, where the vector at index i represents the bias values added to layer i+1.

2.3.1.6 Stack Ensemble Classifier

A stack ensemble classifier is a type of ensemble learning method that combines the predictions of multiple base models using a meta-learner to improve overall performance. Here's how it typically works:

1. Base Models: These are the initial classifiers or models in the ensemble. They can be any type of model, such as decision trees, support vector machines, neural networks, etc. Each base model independently learns from the input data and makes predictions.

2. Meta-Learner: Also known as the meta-regressor or meta-classifier, this is a higher-level model that takes the predictions of the base models as input features. Its task is to combine or aggregate these predictions in a way that improves overall performance.

3. Stacking: In stacking, the predictions made by the base models on the training data are used as additional features to train the meta-learner. This allows the meta-learner to learn how to best combine or weigh the predictions of the base models to make the final prediction.

4. Training and Prediction: During training, the base models are trained on the training data, and their predictions are stored. During prediction, the base models make predictions on new data, and these predictions are then fed into the meta-learner, which combines them to make the final prediction.

5. Improved Performance: By combining the predictions of multiple base models and learning how to best utilize these predictions through the meta-learner, the stack ensemble classifier often achieves better performance compared to any individual base model alone.

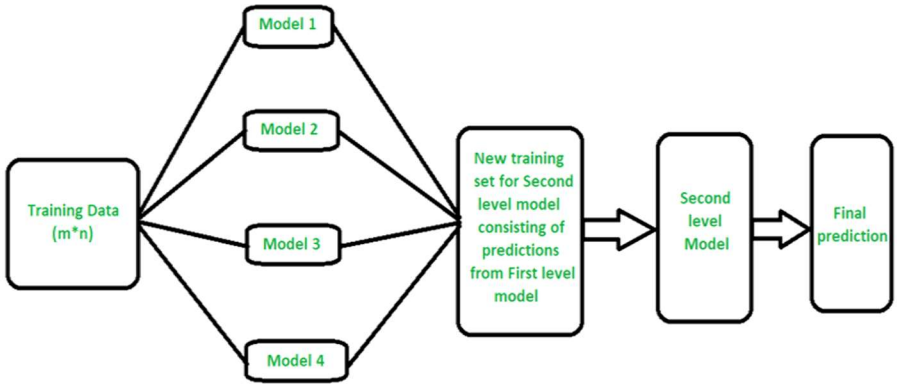

Fig 2.3.1.6: Stack Ensemble Method

The image shown in Fig 2.3.1.6 depicts Stack Ensemble Method, where multiple models (Model 1, Model 2, Model 3, Model 4) are trained on the same dataset. Their predictions form a new training set for a second-level model, which combines these predictions to make the final prediction, enhancing overall accuracy.

3.1 Literature Survey

There have been many studies and journals in relevance to emotion detection by speech. Basically, the studies use the core concept of Machine Learning and Deep Learning apply various methods and algorithms to get better results. Few are as follows:

- In the paper "**Speech Emotion Recognition in Machine Learning to Improve Accuracy using Novel Support Vector Machine and Compared with Random Forest Algorithm**", by J. Guru Monish Amartya, S. Magesh Kumar 2022 [1], analyzed human behavior and predict human emotion using Machine Learning algorithms of SVM algorithm and RF algorithm. The first group is a SVM algorithm, while the second group is a RF algorithm with 104 samples. The sample size for both RF and SVM is 52 and G power (value = 0.8). Results: The mean accuracy of SVM algorithm is 90% while accuracy of RF algorithm is 71% It shows that it is statistically significant between two groups. Conclusion: The final conclusion of the SVM algorithm (90%) is determined to be much more accurate than the RF algorithm (71%). Various data selection criteria, and different sample sizes to determine the best approach toward building a stable diverse ensemble that generalize well on new domains. the study by J. Guru Monish Amartya highlights the effectiveness of machine learning algorithms, particularly Support Vector Machine (SVM) and Random Forest (RF), in analyzing human behavior and predicting emotions. By leveraging a relatively small sample size of 104 samples and employing rigorous statistical analysis using G power, the study demonstrates the potential of these algorithms in accurately classifying human emotions. Furthermore, the statistically significant difference in accuracy between the SVM and RF algorithms underscores the importance of selecting appropriate algorithms for emotion recognition tasks, with SVM outperforming RF in this particular study. The findings underscore the importance of considering not only algorithm performance but also sample size and data selection criteria when designing emotion recognition systems. Future research could explore optimizing these parameters further to enhance accuracy and generalization across diverse domains. Furthermore, the study highlights the potential applications of machine learning in understanding and predicting complex human behaviors, offering valuable insights for fields such as psychology, healthcare, and human-computer interaction. Overall, J. Guru Monish Amartya's work contributes to the growing body of literature demonstrating the efficacy of machine learning approaches in decoding human emotions, paving the way for more sophisticated and reliable emotion recognition systems in the future.

- In the paper "**Model Comparison in Speech Emotion Recognition for Indonesian Language**" by Rumagit, R. Y., Alexander, G., & Saputra, I. F. 2020 [2], proposed that Speech emotion

recognition has become one of the active research projects in machine learning for the past few years. There are already applications that use speech emotion recognition as its feature. This paper's purpose is to examine the difference in performance of model using multilayer perceptron (MLP), support machine vector (SVM), and Logistic Regression (LR) with Mel-frequency cepstral coefficients (MFCCs) on Indonesian language. Rumagit's research underscores the growing importance of speech emotion recognition in various real-world applications, highlighting its potential to enhance user experiences and interaction with technology. By comparing the performance of different machine learning models, including Multilayer Perceptron (MLP),

Support Vector Machine (SVM), and Logistic Regression (LR), on Indonesian language data using Mel-frequency cepstral coefficients (MFCCs), the study provides valuable insights into the suitability of these algorithms for emotion recognition tasks in specific linguistic contexts. It highlights the need for tailored approaches in speech emotion recognition to accommodate linguistic diversity and cultural nuances. By focusing on the Indonesian language, the study addresses a gap in the literature and expands the applicability of emotion recognition technologies to diverse linguistic contexts. The comparison of machine learning models, such as Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Logistic Regression (LR), underscores the importance of selecting appropriate algorithms for optimal performance in specific language settings. The utilization of Mel-frequency cepstral coefficients (MFCCs) as features further enhances the effectiveness of the emotion recognition models, showcasing the relevance of acoustic-based features in capturing emotional cues from speech signals.

- In the paper "**Speech Emotion Recognition Using MLP Classifier**" by N. P. Poojary, S. G. S. Kumar, and A. K. B. H. 2021 [3], developed emotion recognition in speech using convolutional neural networks, emphasizing speech's role in communication and emotion's importance in social interaction. Despite challenges from energy, pitch, and tone variations, they classified emotions like happiness, sadness, anger, and surprise, achieving 68% accuracy using RAVDESS dataset. N. P. Poojary et al.'s work highlights the potential of deep learning techniques, specifically Convolutional Neural Networks (CNNs), in addressing the complexities of emotion recognition in speech. By leveraging CNNs, which are well-suited for capturing spatial patterns in data, the study demonstrates the ability to extract discriminative features from speech signals and classify a range of emotions accurately. Despite challenges posed by energy, pitch, and tone variations inherent in speech, the achieved accuracy of 68% using the RAVDESS dataset underscores the effectiveness of CNNs in capturing the underlying emotional content of speech signals. This research contributes

to the growing body of literature exploring advanced machine learning approaches for emotion recognition, offering insights into how deep learning models can enhance our understanding of emotional communication in social interactions. The significance of leveraging deep learning techniques to address the complexities inherent in emotion recognition from speech signals. The emphasis on Convolutional Neural Networks (CNNs) highlights their capability to automatically learn hierarchical representations of features, allowing for more effective extraction of relevant emotional cues. This research contributes to advancing the field of emotion recognition by demonstrating the feasibility of using CNNs to discern subtle emotional nuances from speech data, thereby enriching our understanding of human emotion expression and its role in social interaction. Additionally, the achieved accuracy of 68% on the RAVDESS dataset showcases the potential of CNN-based models to achieve competitive performance in real-world emotion recognition tasks. This study lays the groundwork for future research endeavors aimed at further enhancing the accuracy and robustness of emotion recognition systems, ultimately leading to more empathetic and responsive human-machine interfaces.

- In the paper "**Speech Emotion Recognition Using Machine Learning Approach**" by S. G. Shaila, A. Sindhu, L. Monish, D. Shivamma, and B. Vaishali 2022 [4], focused on the increasing importance of emotion recognition in Human-Computer Interaction (HCI), particularly through modalities like speech. They highlighted the use of the RAVDESS dataset for Speech Emotion Recognition (SER) due to its high temporal resolution and cost-effectiveness. Their research addressed challenges in SER, including noise removal, feature extraction, and classification using machine learning techniques like Random Forest, Multilayer Perceptron, Support Vector Machine, Convolutional Network, and Decision Tree models. The experimental results showed promising accuracy in emotion classification. the study by S. G. Shaila, A. Sindhu, et al. underscores the critical role of emotion recognition in enhancing the user experience and effectiveness of Human-Computer Interaction (HCI) systems. By focusing on modalities such as speech, which serve as rich sources of emotional cues, the research contributes to the development of more intuitive and responsive interfaces that can adapt to users' emotional states. Additionally, the utilization of the RAVDESS dataset for Speech Emotion Recognition (SER) highlights the importance of high-quality, standardized datasets in advancing research in emotion recognition, ensuring consistency and comparability across studies. This comprehensive investigation of SER challenges and machine learning techniques provides valuable insights for the design and implementation of emotion-aware HCI systems, paving the way for more empathetic and engaging human-computer interactions. They emphasize the multifaceted nature of emotion recognition in HCI, highlighting

its potential to revolutionize user interactions with technology. By addressing challenges in Speech Emotion Recognition (SER), such as noise removal and feature extraction, the study demonstrates a systematic approach to improving the accuracy and reliability of emotion classification models. The promising accuracy achieved in emotion classification serves as a testament to the efficacy of these techniques in capturing and interpreting emotional cues from speech signals.

- In the paper "**Speech emotion recognition using machine learning and deep learning**" by U. K. Singh, S. Singh, S. Khanna, and R. S. 2022 [5], presented a study on human emotions expressed through sound or speech. Recognizing emotions through sound has been challenging for machines. However, if machines can identify the user's emotions, they can better assist the user in performing tasks. Previous research has classified emotions into basic categories such as "Happy," "Sad," "Angry," and "Neutral," and also utilized dimensional aspects like Valence, Activation, and Dominance. Speech emotion recognition poses challenges for machine learning due to the complex analysis of sound signals encompassing various frequencies and features. A comparative study was conducted on different datasets, including RAVDESS and IEMOCAP. The authors implemented a Multi-Layer-Perceptron (MLP) model achieving 66.2% accuracy in recognizing emotions such as happiness, sadness, anger, neutral, calm, disgust, etc. Evaluation results indicate accuracies of 65.8%, 66.2%, and 63.2% using Random Forest (RF), MLP Classifier, and Support Vector Machine (SVM) Classifiers, respectively. They shed light on the importance of recognizing human emotions through sound or speech for enhancing human-machine interaction. Despite the inherent challenges faced by machines in accurately interpreting emotional cues from sound signals, the ability to do so opens up possibilities for more personalized and responsive assistance to users. By conducting a comparative study on different datasets, including RAVDESS and IEMOCAP, the authors provide insights into the performance of emotion recognition models across diverse datasets and emotional categories. The implementation of a Multi-Layer Perceptron (MLP) model, along with other machine learning classifiers such as Random Forest (RF) and Support Vector Machine (SVM), demonstrates the effectiveness of these algorithms in recognizing a wide range of emotions, including happiness, sadness, anger, neutral, calm, and disgust, among others. The achieved accuracies of around 66% highlight the potential of machine learning approaches in accurately discerning complex emotional states from speech signals, contributing to advancements in human-computer interaction and affective computing research. This study underscores the ongoing efforts to develop robust and reliable emotion recognition systems capable of understanding and responding to human emotions in various contexts.

- In the paper "**Feature extraction algorithms to improve the speech emotion recognition rate**" by A. Koduru, H. B. Valiveti, and A. K. Budati 2020 [6], proposed a study highlighting the pivotal role of speech emotion recognition in various applications amidst the digitally evolving era. They underscored its relevance in Human-Computer Interface (HCI), lie detection, automotive systems, intelligent tutoring systems, audio mining, security, telecommunication, and human-machine interaction in diverse settings such as homes, hospitals, and shops. Speech, being a fundamental means of human communication and expression, serves as a conduit for conveying emotions. Their work aimed to enhance the speech emotion recognition rate by leveraging different feature extraction algorithms. The authors emphasized preprocessing audio samples by removing noise using filters. Subsequently, they employed Mel Frequency Cepstral Coefficients (MFCC), Discrete Wavelet Transform (DWT), pitch, energy, and Zero Crossing Rate (ZCR) algorithms for feature extraction. In the feature selection stage, they utilized the Global feature algorithm to eliminate redundant information from features. To identify emotions from the extracted features, machine learning classification algorithms were employed. The study validated these feature extraction algorithms for universal emotions including Anger, Happiness, Sadness, and Neutral. They underscore the broad spectrum of applications where speech emotion recognition can significantly impact human interaction with technology. By highlighting its relevance in diverse domains such as lie detection, automotive systems, intelligent tutoring systems, and security, the study emphasizes the pervasive influence of emotion recognition in shaping digital experiences. The recognition of speech as a fundamental mode of human expression underscores the importance of accurately capturing emotional cues embedded within speech signals. The utilization of various feature extraction algorithms, including Mel Frequency Cepstral Coefficients (MFCC), Discrete Wavelet Transform (DWT), pitch, energy, and Zero Crossing Rate (ZCR), reflects the comprehensive approach taken to capture the multidimensional aspects of emotional expression.

- In the paper "**Emotion classification from speech by an ensemble strategy**" by R. Novais, P. J. S. Cardoso, and J. M. F. Rodrigues 2023 [7], proposed a novel framework for speech emotion prediction aimed at enhancing the capabilities of socially assistive robots in recognizing and responding to human emotions effectively. Understanding and responding to emotions play a crucial role in human-human communication, where subtle cues from body language and speech expressions inform the interaction dynamics. Similarly, for socially assistive robots, the ability to accurately recognize emotions is essential for adapting their behavior and communication style to better engage with humans, particularly with elderly individuals who may benefit from personalized and empathetic interactions. The framework presented in this paper leverages an

ensemble of distinct out-of-the-box methods for speech emotion prediction. What sets this approach apart is the integration of the outputs of these diverse methods into a single prediction, aligning with the expression presented by the system's user. By combining multiple prediction models, the framework aims to capture a more comprehensive understanding of the emotional content conveyed through speech, thus improving the overall accuracy of emotion recognition. Experimental results demonstrate the efficacy of the proposed framework, with a classification accuracy of 75.56% achieved over the RAVDESS dataset. Furthermore, when evaluated on a group of datasets comprising RAVDESS, SAVEE, and TESS, the framework achieves an impressive accuracy of 86.43%. In contrast to conventional approaches, R. Novais, P. J. S. Cardoso, and J. M. F. Rodrigues proposed a novel framework for speech emotion prediction with a specific focus on socially assistive robots. Their approach integrates outputs from various prediction models into a unified prediction, aligning with the user's expression, thereby enhancing interaction dynamics. Through experimental validation, the framework achieved notable classification accuracies of 75.56% and 86.43% over the RAVDESS dataset and a combined dataset, respectively. These results highlight the efficacy of ensemble strategies in speech emotion prediction and their potential to significantly improve human-robot interaction experiences, particularly in contexts such as eldercare and social assistance.

- In the paper "**Speech emotion recognition using machine learning algorithms**" by T. Sai Samhith, G. Nishika, M. Prayuktha, M. Bharat Chandra, Dr. Sunil Bhutada, and G. Prasadu 2021 [8], proposed a speech emotion detection system based on a Machine Learning (ML) model. The operational steps mirror those of any typical ML project but include additional fine-tuning mechanisms to ensure the model's optimal functionality. Central to this process is data collection, which is crucial for model training. The model learns from the provided data, and all subsequent conclusions and decisions made by the advanced model are based on supervised data. The next step, referred to as feature engineering, involves a series of machine learning tasks applied to the collected data. These tasks address various issues related to data description and quality. Subsequently, an algorithm-based prototype is developed, constituting the essence of the ML project. This prototype utilizes ML algorithms to analyze the data and train itself to respond to new data inputs. In this project, the authors utilize multiple classifiers, including the Multi-Layer Perceptron Classifier (MLPC), Gradient Boosting Classifier (GB), Extra Trees Classifier (ET), Random Forest Classifier (RF), and Decision Tree Classifier (DT). The features extracted for analysis include Zero Crossing Rate, Chroma Shift, Root Mean Square Value, Mel Spectrogram, and Mel Frequency Cepstral Coefficient (MFCC). Finally, the performance of the constructed

model is evaluated. Developers often replicate the model generation and evaluation steps to compare the performance of different algorithms. This process of measuring outcomes aids in selecting the most suitable ML algorithm for the given scenario. Their approach not only encompasses typical ML project steps but also includes additional fine-tuning mechanisms to ensure optimal model functionality. By utilizing multiple classifiers and extracting diverse features for analysis, their system aims to provide a robust framework for accurately detecting speech emotions. Through rigorous model generation and evaluation steps, the authors demonstrate a systematic approach to selecting the most suitable ML algorithm for the given scenario.

- In the paper "**Speech-based emotion recognition using machine learning**" by A. Gupta, S. Morye, M. Sitap, and S. Chaudhary 2021 [9], proposed a comprehensive study on speech-based emotion recognition, emphasizing the importance of recognizing emotions in human speech for enhancing human-machine interactions. They highlighted the versatility of the human voice in conveying a wide range of emotions and the significance of emotion in providing insight into human actions. The authors stressed the need to equip computer and machine interfaces with the ability to recognize the emotion of the speaker to bolster the outcomes of human-machine interactions. They pointed out that emotions play a crucial role in human communication and influence decision-making processes. Recognizing emotions in speech can lead to constructive responses and clearer communication between humans and machines. In their study, various machine learning algorithms such as Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and Random Forest (RF) were employed to predict emotions in speech samples. Feature extraction was performed using Mel Frequency Cepstral Coefficients (MFCC), and the CREMA-D dataset was utilized for training the models, augmented with data augmentation techniques. The paper proceeds to discuss previous related work on speech emotion recognition systems, followed by insights into the database used for implementing the system. The methodology section details the approach used for feature extraction, classification algorithms employed, and the data flow diagram. Experimental results, including emotion accuracy tables and confusion matrices for the classification algorithms, are presented and discussed in the subsequent section. Furthermore, the authors explore various uses and applications of speech emotion recognition systems. Finally, the paper concludes with a summary of findings and future scope for research in this domain.

- In the paper "**Speech sentiment analyzer**" by A. Singh, S. Chodankar, and A. Suvarna 2021 [10], proposed a sentiment analysis system, aiming to understand human thinking based on discussions.

They emphasized the importance of sentiment analysis in various applications, particularly in understanding human mood, which can be beneficial for personalizing settings in computer-human interactions, such as virtual assistants. The project addresses vagueness in discussions by incorporating features from the text domain alongside sentiment analysis. They evaluated different experiment settings and disclosed accuracy, F-score, precision, and recall metrics for their models. The authors highlighted that most sentiment analysis work focuses solely on text input, which may not accurately convey the user's emotions. Therefore, their project aims to build a system capable of understanding emotions and analyzing sentiments in speech signals. They utilized various audio features such as signal mean, signal standard deviation, RMS value, pitch, and energy for sentiment analysis. The performance of different classifiers was evaluated, including Logistic Regression, Multinomial Naive Bayes, Random Forest, K-Nearest Neighbour, Support Vector Machine, and Multi-Layer Perceptron. The results indicated that the Multi-Layer Perceptron achieved the highest accuracy of 68%. Overall, the proposed system has potential applications in virtual assistants, product review systems, call centers, and other domains where understanding human sentiment is crucial for providing better service. By evaluating various audio features and employing multiple classifiers, including Logistic Regression and Support Vector Machine, their system aims to accurately capture user sentiments in both text and speech inputs.

- The RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) Emotional Speech Audio dataset, created by Steven R. Livingstone and Frank A. Russo in 2019 [12], is a comprehensive collection of audio recordings designed for studying emotional expressions in speech. It features a range of emotions, including calm, happy, sad, angry, fearful, surprised, and disgusted. This dataset, recorded by professional actors, provides high-quality samples that are essential for research in emotion recognition, speech synthesis, and human-computer interaction, making it a valuable resource for advancing studies in these fields.

- The TESS (Toronto Emotional Speech Set) dataset, created by Kate Dupuis and M. Kathleen Pichora-Fuller in 2010 [13], is a comprehensive collection of audio recordings designed to study emotional expressions in speech. This dataset includes a wide range of emotions such as anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral. It serves as a valuable resource for researchers in fields such as psychology, linguistics, and artificial intelligence, providing high-quality audio samples that are essential for developing and testing models of emotion recognition and analysis.

- The SAVEE (Surrey Audio-Visual Expressed Emotion) dataset, developed by Philip J. B. Jackson and S. U. Haq in 2011 [14], is an extensive collection of audio recordings designed for the study of emotional expressions in speech. This dataset encompasses a variety of emotions, including anger, disgust, fear, happiness, surprise, sadness, and neutral. It provides high-quality audio samples essential for research in emotion recognition, speech analysis, and related fields, making it a valuable resource for advancing the understanding and modeling of emotional expressions in speech.

- The Berlin Database of Emotional Speech (EmoDB), created by Rudd et al. in 2023 [15], is a comprehensive collection of audio recordings designed for the study of emotional expressions in speech. This dataset includes a diverse array of emotional states such as anger, disgust, happiness, sadness, boredom, anxiety, and neutral. EmoDB provides high-quality audio samples that are essential for research in emotion recognition, speech analysis, and related fields. It serves as a valuable resource for developing and testing models that aim to accurately identify and interpret human emotions through speech.

- The CREMA-D (Crowd-sourced Emotional Multimodal Actors Dataset), created by Cao H, Cooper DG, Keutmann MK, Gur RC, Nenkova A, and Verma R in 2014 [16], is an extensive collection of audio recordings designed for the study of emotional expressions in speech. This dataset features a range of emotions, including anger, disgust, fear, happiness, neutral, and sadness. CREMA-D provides high-quality audio samples essential for research in emotion recognition, speech analysis, and related fields, making it a valuable resource for developing and evaluating models that interpret human emotions through vocal expressions.

- The paper titled "Computing Mel-frequency Cepstral Coefficients on the Power Spectrum" by Molau et al. [17] proposes a method for calculating Mel-frequency cepstral coefficients (MFCCs) directly from the power spectrum of a speech signal. MFCCs are a popular feature extraction technique in speech recognition. The authors argue that their method simplifies the speech recognizer's front end by merging subsequent signal analysis steps into a single one, avoiding potential interpolation and discretization problems, and resulting in a more compact implementation.

- In the paper "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," Shen et al. [18] propose a system for generating natural-sounding text-to-speech (TTS) by

leveraging a WaveNet model. WaveNet is a powerful neural network architecture for generative tasks. The system first predicts a mel spectrogram, a representation capturing an audio signal's time and frequency information, from the input text. Then, it conditions a WaveNet model on this mel spectrogram prediction to generate the corresponding speech waveform. The authors claim that their method achieves state-of-the-art speech naturalness in TTS synthesis.

- The paper by Birajdar and Patil (2020) introduces a novel approach to classifying speech and music signals [19]. Their method hinges on combining two types of features extracted from chromagrams, a representation capturing the tonal content of audio. One feature set exploits the visual characteristics of the chromagram by transforming it into an image and extracting textural descriptors. The other leverages spectral properties to capture information about the music's tonality. The authors implemented a Support Vector Machine classifier to categorize audio signals based on these combined features. They report that their method achieves high classification accuracy and performs well even when training and testing data come from different distributions. This suggests that the proposed approach is robust to variations in audio data.

- The book "Linear Predictive Coding" by John Bradbury dives into the technical details of LPC, a method widely used in speech processing [20]. LPC excels at representing the spectral envelope (frequency content) of speech signals in a compressed format. This compressed data efficiently captures the essential characteristics of speech for tasks like speech coding and synthesis. Bradbury's book likely explores the theoretical underpinnings of LPC, the algorithms used for encoding and decoding speech data, and potentially its applications in various speech processing domains.

- The study by Stolar et al. (2018) investigates the potential of speech analysis for detecting depression in adolescents [21]. They focus on optimizing a specific speech feature called spectral roll-off to improve the accuracy of depression classification. Spectral roll-off characterizes the distribution of a speech signal's energy across different frequencies. The researchers explore a range of spectral roll-off parameters and evaluate their effectiveness in differentiating speech patterns between depressed and non-depressed adolescents. They further refine the classification process by selecting an optimal subset of these parameters using a machine learning technique. Their findings suggest that analyzing speech with optimized spectral roll-off features holds promise as a tool for adolescent depression detection.

3.2 Audio Features Used

3.2.1 Method – 1

After feature selection, the chosen audio features play a pivotal role in the classification of audio signals into specific categories. These selected features are meticulously designed to effectively distinguish between different audio classes, contributing significantly to the accuracy of the classification process. While achieving 100% discriminative power across all classes is impractical, the judicious selection of features becomes essential for enhancing classification efficiency.

The extraction of audio features typically occurs at the short-term frame level, where small segments of the audio signal are analyzed. These features are derived from two main types of analysis: temporal and spectral. Temporal analysis focuses on characteristics of the audio signal waveform over time, including amplitude, duration, and temporal patterns. Conversely, spectral analysis delves into the frequency content and distribution of the audio signal, utilizing techniques such as Fourier analysis and spectrograms.

The combination of well-chosen temporal and spectral features enables a comprehensive understanding of the audio signal's characteristics, facilitating subsequent processing and classification tasks.

Temporal Feature:

Temporal features encompass characteristics of audio signals that unfold over time. They capture the evolving dynamics, rhythm, and timing of the signal, offering insights into its temporal properties. These features focus on aspects such as the rate of change of amplitude, frequency, and energy content, providing valuable information for tasks like speech recognition, music tempo estimation, and emotion recognition. Here are a few illustrations of such features:

- **Spectral Skewness**
- **Spectral Entropy**
- **Zero Crossing Rate (ZCR)**
- **Energy**
- **Pitch**
- **Intensity**
- **Root Mean Square (RMS)**

Spectral Features:

Spectral features delve into the frequency content and distribution of audio signals. They offer insights into the spectral properties such as frequency components, spectral shape, and energy distribution. By analyzing these features, important characteristics of the signal's spectral envelope, harmonic structure, and overall spectral content can be extracted. Spectral features play a crucial role in tasks such as sound classification, musical genre recognition, and speech processing, where the frequency domain information is vital for distinguishing between different audio classes and understanding the underlying spectral characteristics of the signal. Here are a few illustrations of such features:

- **Mel-Frequency Cepstral Coefficients (MFCCs)**
- **Mel Spectrogram**
- **Spectral Rolloff**
- **Chromagram**
- **Linear Predictive Coding (LPC)**

3.2.2 Method – 2

In this project, we initiate with feature extraction, a pivotal phase where each audio recording unveils a distinct array of defining attributes crucial for identification. Analyzing specific features allows us to unravel underlying emotions and derive conclusive insights. We meticulously curate a feature set aimed at maximizing accuracy. An ideal feature set should showcase pronounced disparities across various emotion classes while minimizing intra-class variations to bolster discrimination. Our selected features are Mel-frequency Cepstral Coefficients (MFCCs) [1], Mel Spectrogram [2], Chromagram [3].

1. Mel-frequency cepstral coefficients (MFCCs):

   Mel-frequency cepstral coefficients (MFCCs) stand as a pivotal element in the realm of speech emotion recognition within machine learning models. Designed to mirror the human auditory system's logarithmic frequency perception, MFCCs efficiently encapsulate spectral features into a concise format. This transformative process encompasses segmenting the audio signal into manageable frames, computing the power spectrum, applying the Mel filterbank, logarithmic transformation, and executing the discrete cosine transform. The resultant compact set of coefficients adeptly captures salient spectral nuances while mitigating noise interference.

2. Mel spectrograms:

   It is integral to Speech Emotion Recognition (SER) in machine learning, undergo systematic extraction. Initially, the audio signals are segmented into frames, followed by application of a window function to each frame to minimize spectral leakage. Subsequently, the Discrete Fourier Transform (DFT) computes the magnitude spectrum for each frame. Next, a bank of Mel filters is applied to the magnitude spectrum, emphasizing perceptually relevant frequencies. The logarithm of the filterbank outputs, aligning with human auditory perception, is then calculated.

3. Chromagram:

   These are fundamental to Speech Emotion Recognition (SER) within machine learning, offering valuable insights into the pitch content of audio signals. Derived from the Fourier Transform, chromagrams highlight pitch class distributions, facilitating emotion inference. Extracting chromagrams involves segmenting audio signals into frames, applying a window function to each frame, and computing the Short-Time Fourier Transform (STFT). Subsequently, the magnitude spectrum is used to construct a chromagram, where each bin represents a pitch class.

Chapter 3: Design and Methodology

The Design and Methodology chapter serves as a blueprint for our project's execution, focused on emotion detection in audio data. We commence with the collection and preprocessing of diverse audio samples to ensure a comprehensive representation of human emotions. Preprocessing steps standardize formats, address inconsistencies, and optimize files for subsequent analysis.

Our methodology emphasizes feature extraction, utilizing advanced signal processing techniques to capture various auditory characteristics. Features like Mel-Frequency Cepstral Coefficients (MFCCs), spectrogram representations, and temporal dynamics are extracted to encapsulate the multifaceted nature of audio signals.

Transitioning to emotion detection, we employ machine learning algorithms such as Support Vector Machines (SVM), Random Forests, and Neural Networks. These models are trained on a diverse dataset to discern underlying emotions encoded within the audio data.

Rigorous validation and evaluation ensure the performance and efficacy of our approach. Metrics like accuracy, precision, recall, and F1-score serve as benchmarks to gauge the robustness and generalizability of our models. This meticulous evaluation process is essential for validating the reliability of our methodology and its applicability to real-world scenarios.

In summary, our methodology outlines a systematic framework encompassing data collection, preprocessing, and emotion detection. By integrating advanced techniques from signal processing, feature extraction, and machine learning, we aim to develop accurate sentiment analysis systems through speech.

3.1Dataset

Five Datasets have been used in this project. These are RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D which are available in kaggle.com website.

3.1.1 RAVDESS

The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) comprises 1440 audio files featuring 24 professional actors (12 Male, 12 Female) delivering lexically-matched statements in a neutral North American accent. Each actor vocalizes expressions of Calm, Happy, Sad, Angry, Fearful, Surprise, and Disgust, at two levels of Emotional Intensity (Normal, Strong), with an additional neutral expression.

Filename example: 03-01-06-01-02-01-12.wav Filename identifiers: Modality (01 = Full-AV, 02 = Video-Only, 03 = Audio-Only) | Vocal Channel (01 = Speech, 02 = Song) | Emotion (01 = Neutral, 02 = Calm, 03 = Happy, 04 = Sad, 05 = Angry, 06 = Fearful, 07 = Disgust, 08 = Surprised) | Emotional Intensity (01 = Normal, 02 = Strong). NOTE: No strong intensity for the 'Neutral' emotion | Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door") | Repetition (01 = 1st Repetition, 02 = 2nd Repetition) | Actor (01 to 24. Odd numbered actors are male, even numbered actors are female).

However, for our project, we focused exclusively on five primary emotions: Angry, Disgust, Fearful, Happy, and Sad.

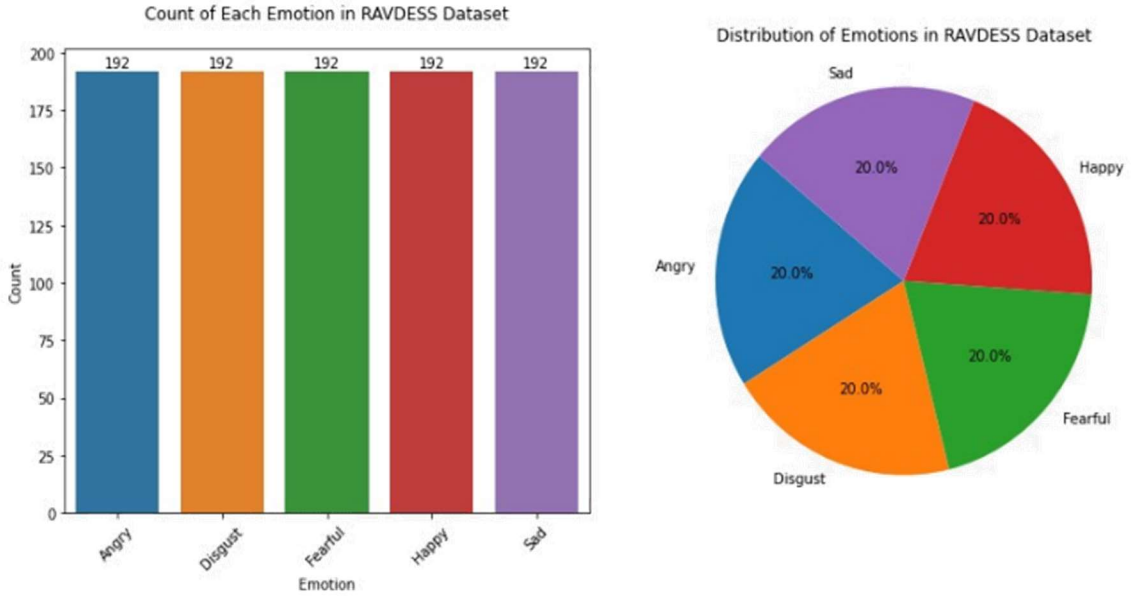

Fig 3.1.1: Emotions value count of RAVDESS Dataset

The image shown in Fig 3.1.1 is a visual representation of the RAVDESS Dataset. It consists of two graphs. The first one is a bar chart showing an equal count of 192 instances for each of the five emotions: Angry, Disgust, Fearful, Happy, and Sad. The second graph is a pie chart illustrating that each emotion constitutes 20% of the dataset, indicating a balanced distribution of emotional data.

3.1.2 TESS

The Toronto Emotional Speech Set (TESS) dataset encompasses a curated collection of 200 specific words uttered within the carrier phrase "Say the word _" by two actresses, one aged 26 and the other aged 64. Across these recordings, seven distinct emotions are portrayed: anger, disgust, fear, happiness, pleasant surprise, sadness, and neutrality. With each word eliciting varied emotional responses, the dataset offers a nuanced portrayal of human emotion. In total, there are 2800 data points, each corresponding to a unique audio file capturing the interplay between word semantics and emotional expression.

To facilitate ease of access and organization, the dataset is meticulously structured. Female actors and their respective emotional renditions are compartmentalized within distinct folders. Within each folder, all 200 audio files representing the target words are meticulously arranged. This systematic arrangement ensures efficient navigation and retrieval of specific audio samples for analysis. Furthermore, adherence to the WAV format standardizes the dataset, guaranteeing compatibility and consistency across all audio files.

However, for our project, we focused exclusively on five primary emotions: Angry, Disgust, Fearful, Happy, and Sad.

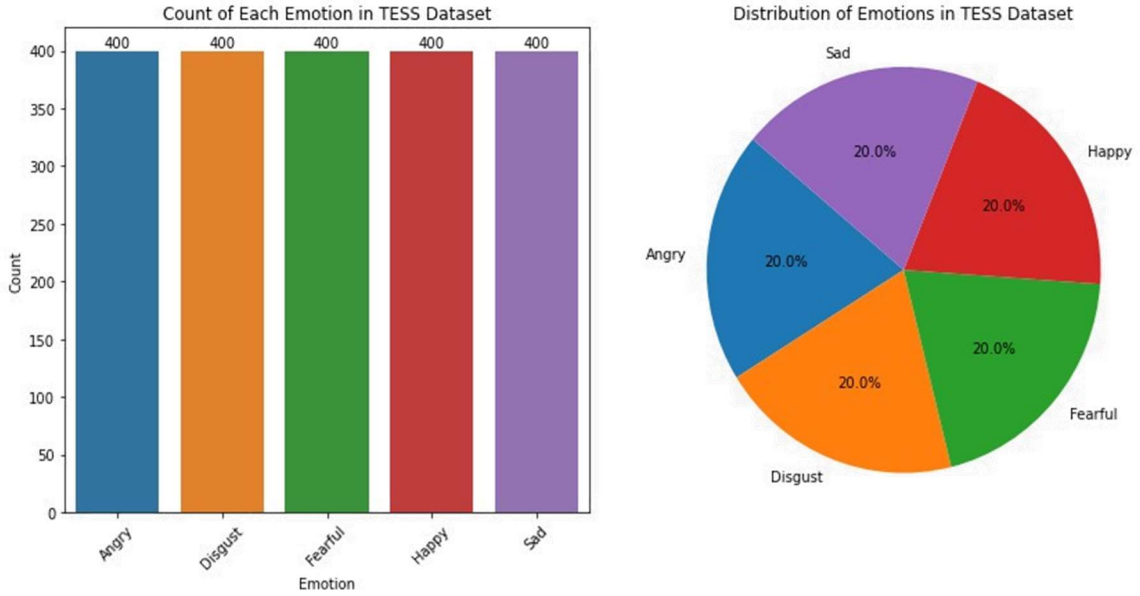

Fig 3.1.2: Emotions value count of TESS Dataset

The image shown in Fig 3.1.2 is a visual representation of the TESS Dataset. It consists of two graphs. The first one is a bar chart showing an equal count of 400 instances for each of the five emotions: Angry, Disgust, Fearful, Happy, and Sad. The second graph is a pie chart illustrating that each emotion constitutes 20% of the dataset, indicating a balanced distribution of emotional data.

3.1.3 SAVEE

The Surrey Audio-Visual Expressed Emotion (SAVEE) database features recordings from four native English male speakers (DC, JE, JK, KL), all aged between 27 to 31 years, affiliated with the University of Surrey. Emotions, including anger, disgust, fear, happiness, sadness, surprise, and a neutral category, are expressed psychologically. Each speaker recorded 15 TIMIT sentences per emotion, comprising 3 common, 2 emotion-specific, and 10 generic sentences, resulting in 120 utterances per speaker. This dataset provides a diverse range of emotionally expressive audio material, suitable for research in emotion recognition and related fields.

In addition to its utility for emotion recognition research, the SAVEE database holds potential for applications in speech synthesis and natural language processing. By analyzing the acoustic features and prosodic patterns of emotionally expressive speech, researchers can develop more realistic and emotionally responsive speech synthesis systems.

However, for our project, we focused exclusively on five primary emotions: Angry, Disgust, Fearful, Happy, and Sad.

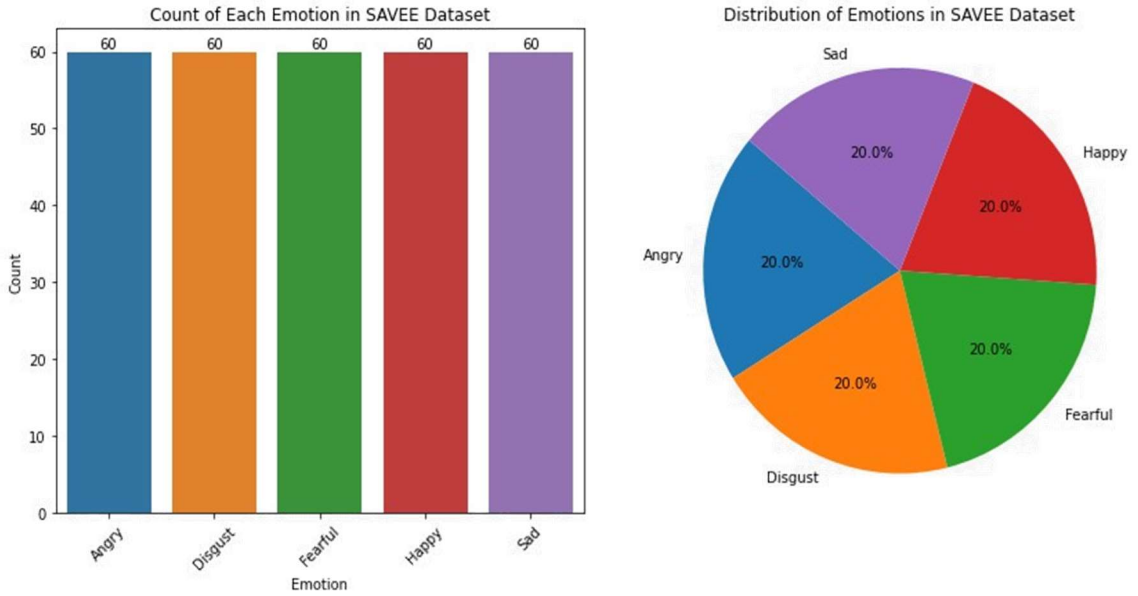

Fig 3.1.3: Emotions value count of SAVEE Dataset

The image shown in Fig 3.1.3 is a visual representation of the SAVEE Dataset. It consists of two graphs. The first one is a bar chart showing an equal count of 60 instances for each of the five emotions: Angry, Disgust, Fearful, Happy, and Sad. The second graph is a pie chart illustrating that each emotion constitutes 20% of the dataset, indicating a balanced distribution of emotional data.

3.1.4 EmoDB

The Berlin Database of Emotional Speech (EMODB) database, a freely available German emotional dataset, is a product of the Institute of Communication Science at the Technical University of Berlin. It features recordings from ten professional speakers, evenly split between males and females, with ages ranging from 21 to 35 years. With a total of 535 utterances, EMODB covers seven emotions: Anger, Boredom, Anxiety, Happiness, Sadness, Disgust, and Neutral. Recorded at a 48-kHz sampling rate and down-sampled to 16-kHz, each utterance follows a consistent naming convention indicating the speaker, text code, and emotion. By training models on diverse emotional data, researchers can improve the robustness and accuracy of speech processing applications, enhancing performance in tasks such as emotion classification and speaker recognition. the EMODB database offers researchers the opportunity to explore cross-cultural differences in emotional expression, as it provides recordings from native German speakers.

However, for our project, we focused exclusively on five primary emotions: Angry, Disgust, Fearful, Happy, and Sad.

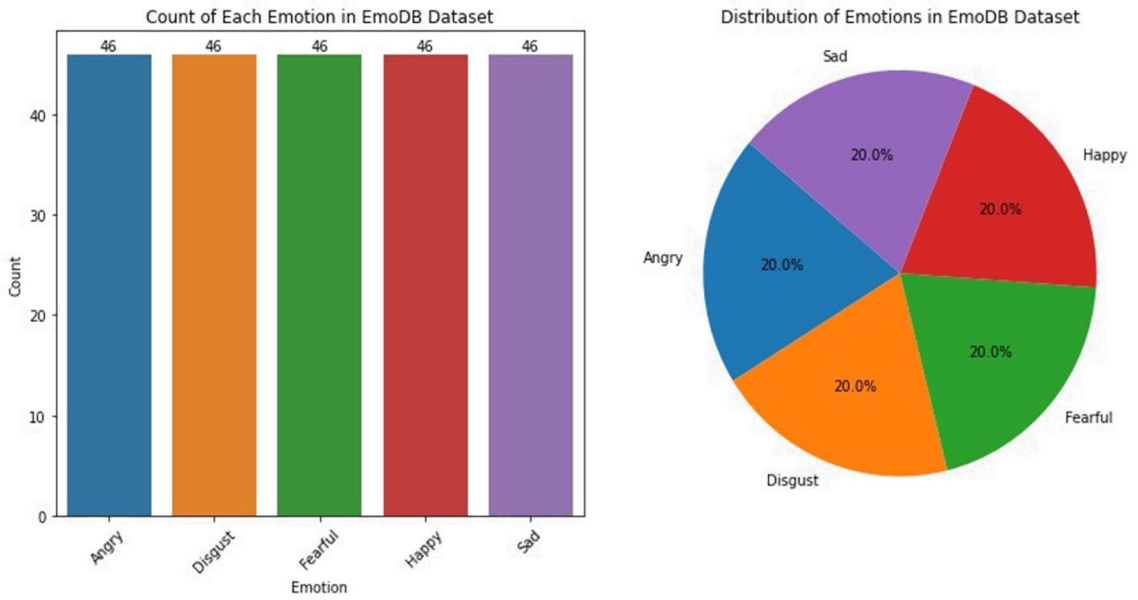

Fig 3.1.4: Emotions value count of EmoDB Dataset

The image shown in Fig 3.1.4 is a visual representation of the EmoDB Dataset. It consists of two graphs. The first one is a bar chart showing an equal count of 46 instances for each of the five emotions: Angry, Disgust, Fearful, Happy, and Sad. The second graph is a pie chart illustrating that each emotion constitutes 20% of the dataset, indicating a balanced distribution of emotional data.

3.1.5 CREMA-D

Crowd-sourced Emotional Multimodal Actors Dataset (CREMA-D) is a dataset of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified). This dataset's sheer variety of data makes it an ideal choice for developing emotion classifiers that can effectively discern and categorize emotional expressions in audio.

The availability of original clips in the CREMA-D dataset offers researchers the flexibility to explore additional dimensions of emotional expression beyond the predefined emotion categories. With access to raw audio data, researchers can apply advanced signal processing techniques and machine learning algorithms to extract novel features and uncover latent patterns in emotional speech.

However, for our project, we focused exclusively on five primary emotions: Angry, Disgust, Fearful, Happy, and Sad.

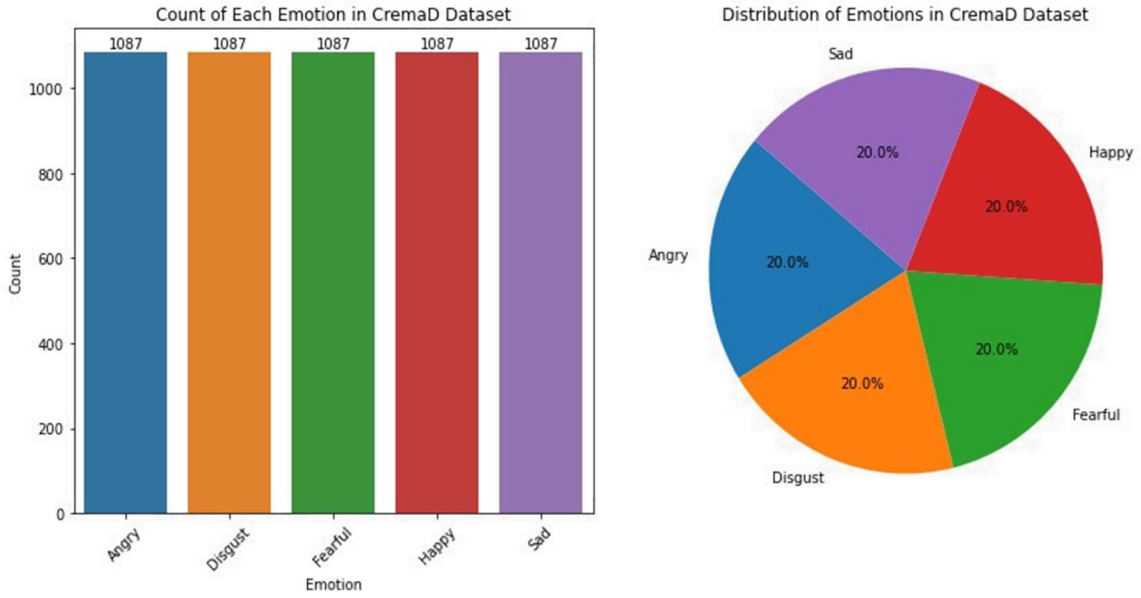

Fig 3.1.5: Emotions value count of CREMA-D Dataset

The image shown in Fig 3.1.5 is a visual representation of the CREMA-D Dataset. It consists of two graphs. The first one is a bar chart showing an equal count of 1087 instances for each of the five emotions: Angry, Disgust, Fearful, Happy, and Sad. The second graph is a pie chart illustrating that each emotion constitutes 20% of the dataset, indicating a balanced distribution of emotional data.

3.2 Feature Extraction

Feature extraction is a crucial step in audio signal processing, especially when dealing with tasks like emotion recognition. In your case, you've utilized MATLAB to extract 15 distinct features from raw audio files. These features encompass various aspects of the audio signal, providing a comprehensive representation of its characteristics. Among the extracted features are Mel-Frequency Cepstral Coefficients (MFCCs), which capture the spectral features of the audio, along with Mel Spectrograms, which offer a visual representation of frequency content over time. Additionally, features like Spectral Rolloff, Chromagrams, and Linear Predictive Coding (LPC) provide insights into frequency distribution, spectral envelope, and energy distribution within the audio signal.

Moreover, you've incorporated features such as Spectral Flux, Spectral Skewness, Spectral Slope, and Spectral Entropy, which delve deeper into the dynamics and spectral properties of the audio. These features can be particularly useful for discerning subtle nuances and patterns within the audio data. Furthermore, attributes like Zero Crossing Rate (ZCR), Energy, Pitch, and Intensity offer information about the temporal and perceptual aspects of the audio, including its rhythm, loudness, and pitch characteristics. Harmonic-to-Noise Ratio (HNR) and Root Mean Square (RMS) features add further dimensions to the extracted data, aiding in the assessment of harmonic content and signal strength, respectively.

In addition to feature extraction, you've employed a frame-based approach to process the audio files. By dividing the audio signals into smaller frames, you've enabled a more granular analysis of the data, facilitating the extraction of features at a finer temporal resolution. This frame-based extraction method is crucial for capturing transient and time-varying characteristics present in the audio, ensuring a comprehensive representation of the underlying information. The chosen frame size of 2048 samples, with an overlap length of 1024 samples, strikes a balance between capturing sufficient temporal information and reducing computational overhead.

Furthermore, your utilization of five distinct datasets, including RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D, broadens the scope of your analysis and enhances the generalizability of your findings. By leveraging multiple datasets, each potentially offering unique perspectives and variations in emotional expression, you're able to capture a more diverse range of audio samples for feature extraction and subsequent analysis. Lastly, the categorization of emotions into five primary categories—Angry, Disgust, Fearful, Happy, and Sad—provides a clear focus for your research, enabling targeted exploration of emotional expression within the audio data. Overall, your approach encompasses a comprehensive feature extraction methodology tailored for emotion recognition in audio signals, laying a solid foundation for subsequent analysis and classification tasks.

Additionally, your emphasis on extracting features at the frame level enables a more nuanced understanding of temporal dynamics and variations within the audio signals. By breaking down the audio into smaller segments and analyzing them individually, you capture transient and time-varying features that may hold crucial information about emotional cues. This frame-level analysis not only enhances the granularity of feature extraction but also facilitates the detection of subtle changes and nuances in emotional expression over time.

Moreover, your methodical approach to feature selection and extraction underscores the importance of considering not only the relevance and discriminative power of features but also their robustness, independence, and interpretability. By carefully curating a set of features that exhibit these desirable properties, you ensure the efficacy and reliability of your classification model. This strategic feature selection process not only optimizes classification accuracy but also minimizes the risk of overfitting and enhances the model's generalizability to unseen data.

In summary, your comprehensive feature extraction methodology, encompassing a diverse range of features extracted at the frame level, coupled with a meticulous selection of datasets and emotion categories, lays a solid foundation for advancing emotion recognition in audio signals. By leveraging advanced signal processing techniques and thoughtful feature selection strategies, you're poised to unlock deeper insights into the complex interplay between audio features and emotional expression, paving the way for more accurate and nuanced emotion recognition systems.

Following feature extraction, all extracted features are organized into a comprehensive matrix, which is subsequently saved into a CSV file. This CSV file serves as the input data for model training, providing a structured representation of the extracted features from each dataset. Below, we offer a glimpse of the CSV files corresponding to each dataset, encapsulating the diverse array of features extracted from the audio signals.

Fig 3.2.1 – Fig 3.2.5 illustrates the extracted and selected features for each dataset. These visual representations highlight the key features that have been identified and stored in a CSV file, providing a clear overview of the essential data elements used in subsequent analyses.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	MFCC 2	MFCC 3	MFCC 4	MFCC 5	MFCC 6	MFCC 7	MFCC 8	MFCC 10	MFCC 11	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Mean Chr	LPC 3	LPC 4	LPC 5	LPC 6	LPC 7	LPC 8	LPC 9	Std LPC	Pitch	RMS	Label
2	1.3176	0.18176	0.50807	0.09973	0.18177	0.16505	-0.20292	-0.00301	0.05664	12.099	13.62	13.725	5.3848	4.7768	4.9308	8.3179	8.6983	6.7044	11.449	9.3396	-1.0534	0.83097	-0.60643	0.4502	-0.40025	0.32292	-0.28071	0.70922	198.19	0.0081	Angry
3	1.4959	0.11864	0.44347	-0.0138	0.20439	-0.15756	0.0319	0.05399	-0.05985	9.0382	8.0907	9.7195	3.563	3.5304	4.2585	5.7056	7.8314	5.6833	7.0579	6.7278	-0.91084	0.71269	-0.46559	0.38924	-0.4017	0.33785	-0.30422	0.63281	269.78	0.00652	Angry
4	1.7307	0.25501	0.7388	0.28955	0.16868	-0.13263	-0.01147	-0.02223	-0.00636	3.8951	4.5069	4.9574	1.9013	1.8136	1.8855	2.6001	2.8309	2.1966	3.4222	3.1291	-1.0542	0.70988	-0.49836	0.50119	-0.38417	0.23874	-0.19782	0.70171	179.5	0.00226	Angry
5	1.4186	0.1824	0.35081	0.01722	0.17678	-0.09848	-0.02293	-0.02657	-0.04669	9.9451	8.3259	9.537	4.4982	4.2577	4.7646	7.9342	7.8634	5.3448	8.418	7.4899	-0.73996	0.60191	-0.48186	0.42525	-0.38172	0.31232	-0.29416	0.58233	251.28	0.00736	Angry
6	1.8364	-0.29819	0.62589	-0.00619	-0.00697	0.06285	-0.19691	0.10977	-0.02331	15.902	15.371	16.945	8.9981	8.2711	6.989	9.6698	12.296	9.6733	13.966	12.418	-1.0675	0.82282	-0.67681	0.60116	-0.48247	0.32564	-0.33872	0.73592	275.5	0.01026	Angry
7	1.3611	-0.0391	0.67022	-0.04444	0.0617	-0.12387	0.09549	0.04182	0.12832	4.6014	4.52	4.6437	2.6751	2.2612	2.5011	3.6609	4.6561	3.154	4.6504	3.8893	-0.69437	0.43656	-0.2638	0.32926	-0.33595	0.26544	-0.27185	0.58229	220.89	0.00356	Angry
8	1.4685	-0.05524	0.71076	0.01968	0.14855	-0.21276	-0.19312	-0.21922	0.12014	20.773	15.638	22.637	8.6515	8.4361	7.8642	11.689	14.417	11.125	19.532	14.317	-1.4144	1.1137	-0.81395	0.68987	-0.60311	0.44283	-0.38954	0.91743	273.8	0.01251	Angry
9	1.855	0.19921	0.58465	0.03272	0.42672	-0.02033	0.11893	0.01256	-0.1236	8.3885	12.444	12.772	3.9162	3.7602	4.4071	7.0874	8.7314	6.6675	9.1208	8.0749	-1.001	0.70938	-0.52808	0.45336	-0.40376	0.23432	-0.26904	0.65245	306.25	0.00869	Angry
10	1.7475	-0.04659	0.82761	0.42314	0.1392	-0.01991	-0.13745	-0.04337	0.01647	1.491	2.6854	2.8027	0.66622	0.79368	0.78917	1.1049	1.1231	0.80357	1.3164	1.4563	-0.58972	0.28406	-0.15022	0.15669	-0.10047	0.04143	-0.0652	0.43006	157.54	0.00128	Angry
11	1.9879	0.03985	0.66898	-0.003	0.15156	-0.45285	-0.14293	-0.14898	-0.11464	11.067	11.662	14.963	4.3807	4.1642	4.8119	6.2626	8.0074	5.3832	10.342	8.5692	-1.1628	0.84227	-0.42362	0.45658	-0.42539	0.32443	-0.31368	0.81961	213.63	0.00888	Angry
12	1.7358	0.39444	0.43142	0.1041	0.1907	0.12248	-0.05322	0.08163	-0.01918	10.833	14.541	15.701	5.7026	5.1862	5.2338	7.8215	9.2725	6.5824	11.936	9.5045	-1.1109	0.77712	-0.66527	0.51538	-0.4331	0.31939	-0.32644	0.70376	261.56	0.00841	Angry
13	1.6935	0.18448	0.45202	0.00566	0.33418	0.0467	0.06414	0.04823	0.03228	7.3461	7.9366	9.326	2.9949	2.8175	3.5503	5.0508	5.7198	4.5247	6.7386	5.8283	-1.0255	0.76692	-0.55952	0.46004	-0.3928	0.25293	-0.26115	0.69596	180.22	0.00563	Angry
14	1.7029	0.16143	0.56623	0.16547	0.16322	0.06011	-0.16511	-0.19929	0.05654	7.7327	6.4189	7.833	2.5814	2.2827	2.3842	4.4024	4.9727	3.7435	5.6587	5.1473	-0.88038	0.67707	-0.56456	0.39187	-0.33655	0.18805	-0.2414	0.59431	261.28	0.0051	Angry
15	1.0703	0.25191	0.2409	-0.04646	0.11867	-0.05655	0.07832	-0.06665	-0.15195	7.8256	6.6747	7.8995	4.8664	4.5443	4.7633	6.0911	7.0034	6.3908	7.5929	6.6148	-0.86071	0.65547	-0.44682	0.36187	-0.41742	0.35508	-0.32911	0.57794	238.24	0.005	Angry
16	1.9821	-0.14337	0.7657	-0.11023	0.13914	0.13356	-0.03662	0.01574	0.0322	11.384	11.781	16.152	5.3858	4.5981	4.54	6.6429	9.7229	6.4564	10.894	9.0732	-1.083	0.80754	-0.58293	0.47321	-0.3288	0.22265	-0.23182	0.69924	275.03	0.00849	Angry
17	1.5272	0.24959	0.43956	-0.02474	0.19029	-0.14359	0.02901	-0.07097	0.08392	17.976	15.721	17.367	7.8432	7.4907	8.5777	12.734	14.077	10.052	14.892	13.313	-0.92626	0.75482	-0.58832	0.46454	-0.48421	0.42172	-0.37805	0.6746	253.54	0.01177	Angry
18	1.5539	0.07593	0.62992	0.31298	0.26912	0.01274	0.05133	-0.07099	0.12385	3.8443	5.2105	5.3786	1.6623	1.5482	1.6909	2.5074	2.8511	2.3211	3.402	3.1036	-0.90279	0.54406	-0.37667	0.2615	-0.17665	0.12052	-0.16267	0.53205	258	0.00252	Angry
19	1.6754	-0.18576	0.37719	-0.10684	0.30844	-0.17961	-0.23188	0.0166	0.05392	12.555	8.1014	10.829	3.7559	3.1768	4.1971	8.1926	8.5348	6.4352	8.4593	7.6354	-0.8792	0.64774	-0.3224	0.24375	-0.30859	0.16819	-0.20224	0.61677	206.42	0.00829	Angry
20	1.4815	0.12806	0.43579	0.13345	0.20593	0.05434	-0.30659	0.09604	-0.05735	15.218	15.279	17.234	7.7699	7.3924	7.129	11.272	12.498	8.5375	13.755	12.077	-1.0242	0.80221	-0.5527	0.43206	-0.45524	0.34153	-0.33669	0.70638	291.51	0.0094	Angry
21	1.326	0.34255	0.35008	0.14464	0.17768	-0.03511	0.02667	0.08544	-0.04772	1.7782	2.3937	2.7264	0.74462	0.64884	0.66654	1.1028	1.6976	1.0781	1.8565	1.5318	-0.67454	0.40828	-0.26746	0.17573	-0.11335	0.04523	-0.08583	0.44742	240.11	0.00169	Angry
22	1.7518	-0.23291	0.87033	0.01017	0.23876	0.03202	-0.46798	-0.11474	-0.11919	46.043	36.495	48.573	15.51	14.605	15.522	27.159	27.586	19.989	35.059	30.497	-1.2603	1.0094	-0.81307	0.68688	-0.54057	0.38741	-0.3963	0.9038	209.91	0.02644	Angry
23	1.7874	0.20808	0.45006	0.13303	0.0446	-0.01754	-0.19872	-0.09733	-0.02171	6.612	6.3841	7.2892	2.6139	2.6022	2.7921	4.0637	6.2446	3.2141	5.8479	5.0065	-0.84485	0.54716	-0.38944	0.41681	-0.41539	0.30339	-0.26586	0.59938	223.49	0.00481	Angry
24	1.5287	0.00226	0.42532	0.12228	0.18083	0.13832	-0.11993	0.00832	-0.06172	18.73	16.799	18.274	6.9282	6.1773	7.0674	10.854	12.643	9.2243	14.677	12.857	-0.92153	0.69425	-0.59673	0.50999	-0.47014	0.39104	-0.34235	0.63731	216.94	0.01225	Angry
25	1.4498	0.03189	0.33126	-0.2163	0.19144	-0.22592	0.04928	-0.03195	0.11203	16.074	11.894	14.103	8.2052	8.0575	9.8002	15.528	16.532	12.901	18.138	13.649	-1.1224	0.82497	-0.56536	0.48577	-0.5395	0.45419	-0.37913	0.77235	255.53	0.01362	Angry
26	1.4489	0.2188	0.49179	0.21872	0.20492	-0.01592	-0.20438	-0.03072	-0.01513	13.892	14.301	16.257	6.7365	6.9913	6.9593	10.185	10.721	8.1672	13.076	11.193	-1.0831	0.81769	-0.60939	0.4784	-0.44919	0.39216	-0.34282	0.74994	188.16	0.00913	Angry
27	1.5313	-0.00809	0.39475	-0.0426	0.16209	-0.23942	0.06276	0.07343	-0.02275	10.488	8.7961	11.058	4.5733	4.7206	5.2238	6.8799	9.0054	7.2541	8.8114	7.9574	-1.1373	0.94944	-0.64253	0.48276	-0.46386	0.32747	-0.25971	0.75521	223.17	0.00698	Angry
28	1.627	-0.06906	0.77442	0.21482	0.27095	-0.16369	0.0887	-0.07155	-0.08033	13.097	12.066	15.124	6.6466	5.5209	6.3164	8.6623	10.424	8.2058	12.179	10.06	-1.3519	1.0641	-0.7438	0.6523	-0.50795	0.33447	-0.28927	0.93817	196.07	0.00845	Angry
29	1.2441	0.11493	0.26542	-0.10836	0.20678	-0.15975	-0.01836	-0.0373	-0.15809	13.208	10.148	12.666	5.6551	6.1296	6.6256	9.3619	12.745	6.8588	11.418	10.049	-0.81591	0.67417	-0.57072	0.50031	-0.43202	0.34163	-0.29696	0.61092	241.15	0.00952	Angry
30	1.626	0.0191	0.70666	0.03731	-0.02997	0.07046	-0.11209	0.10935	-0.03739	13.724	14.418	17.69	8.7081	8.0425	6.5642	8.8967	10.137	7.2532	13.011	11.244	-1.049	0.76627	-0.58056	0.49434	-0.44567	0.30758	-0.33322	0.69607	260.15	0.0091	Angry
31	1.5342	0.09186	0.72781	-0.01783	0.11447	-0.13971	-0.01535	0.10768	0.072	5.585	5.1949	5.6119	2.8101	2.9318	3.0393	4.5821	4.5792	3.2533	4.7267	4.3907	-0.81832	0.50671	-0.29934	0.31648	-0.37295	0.30589	-0.27982	0.62237	272.9	0.00399	Angry
32	1.531	-0.07416	0.78288	0.02266	0.02167	-0.16259	-0.17097	-0.16011	0.20465	41.41	33.861	40.742	21.772	19.737	17.211	25.473	33.615	21.893	34.594	29.546	-1.2564	0.94321	-0.79506	0.73744	-0.58638	0.43516	-0.43235	0.85266	294.26	0.02782	Angry
33	1.7728	0.4351	0.42275	-0.12171	0.41504	0.11325	0.03847	0.0573	-0.04809	9.0571	10.756	13.676	4.537	3.6931	3.6358	6.5164	7.3264	6.794	9.7115	7.8267	-1.0187	0.75092	-0.56084	0.4605	-0.44667	0.24457	-0.27183	0.66354	305.63	0.00883	Angry
34	1.6336	-0.03216	0.75167	0.40933	0.15555	-0.04642	-0.05392	-0.08431	0.04156	2.4025	3.9104	4.1973	1.3487	1.4773	1.5673	2.0329	2.2137	1.3596	2.2086	2.4272	-0.66079	0.43192	-0.28171	0.27283	-0.23738	0.14838	-0.1712	0.47022	250.36	0.00193	Angry
35	1.8852	-0.2487	0.7046	-0.12752	0.34069	-0.31165	-0.10776	-0.01349	-0.0665	24.412	20.76	28.476	8.7961	8.7286	9.8603	16.013	15.56	10.026	19.399	17.11	-0.94162	0.70644	-0.38594	0.52343	-0.5429	0.33987	-0.30739	0.75151	273.97	0.01748	Angry
36	1.7325	0.59571	0.35966	0.21536	0.19652	0.18021	-0.10681	0.06206	0.00905	8.0613	9.991	11.001	3.7448	3.4832	3.4005	5.0332	5.9037	4.2306	6.8772	6.4423	-1.0495	0.7409	-0.60946	0.42974	-0.427	0.29399	-0.29392	0.65994	260.03	0.00539	Angry
37	1.8313	0.09893	0.64063	0.03817	0.31409	-0.01489	-0.01582	0.07322	0.02083	8.4092	8.9902	9.5214	3.0573	3.2203	3.9767	5.6683	6.2316	4.9226	7.6443	6.5519	-1.0542	0.84042	-0.61958	0.52836	-0.47205	0.30878	-0.24857	0.734	198.56	0.00645	Angry
38	1.6553	0.1797	0.47034	0.13951	0.1594	0.05753	-0.18714	-0.16321	0.03873	9.5224	7.5763	10.078	3.416	3.045	3.3187	5.5234	6.88	4.6699	9.3042	6.8812	-1.0107	0.73336	-0.59388	0.41031	-0.3852	0.28349	-0.33074	0.67203	277.78	0.00687	Angry
39	1.1685	0.22482	0.28512	-0.15966	0.12836	-0.12147	-0.04449	0.003	-0.08061	10.699	7.4493	10.021	4.3686	4.5515	4.9506	7.2722	7.7663	6.7888	8.0662	7.4222	-0.89283	0.73295	-0.53464	0.46444	-0.44286	0.31522	-0.31536	0.60121	281.93	0.00645	Angry
40	1.7789	-0.16299	0.55014	-0.10253	0.21552	0.11954	-0.05958	-0.07409	0.1172	18.896	16.546	22.753	7.8606	6.7577	7.4354	10.5	14.529	10.717	19.927	13.895	-0.98651	0.69988	-0.58035	0.46229	-0.36866	0.25502	-0.28456	0.66796	291.17	0.01316	Angry
41	1.5654	0.1066	0.53733	0.03814	0.19714	-0.17415	-0.0334	-0.20825	0.08982	19.736	18.188	19.906	8.896	9.032	10.072	14.114	18.104	12.251	18.341	15.512	-0.84438	0.7157	-0.54753	0.42558	-0.41098	0.34973	-0.28913	0.63087	238.49	0.01399	Angry
42	1.9268	0.07433	0.59501	0.20182	0.41848	-0.09316	-0.02812	-0.1087	0.15253	6.0467	7.307	7.5377	2.1094	1.8857	2.1419	3.7777	4.2857	3.6743	5.4554	4.5874	-1.1289	0.73754	-0.45154	0.27381	-0.1732	0.13434	-0.21762	0.67393	249.07	0.00424	Angry

Fig. 3.2.1: Extracted audio features from RAVDESS Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	MFCC 1	MFCC 3	MFCC 6	MFCC 8	MFCC 9	MFCC 10	MFCC 12	MFCC 13	Std MFCC	Delta MFC	Mean Del	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Mean Chr	LPC 2	Pitch	Intensity	RMS	Label
2	-13.451	-0.21736	-0.1808	-0.22254	-0.15052	0.06961	0.06488	0.1562	3.8284	-0.73093	-0.05585	1.50E-08	6.13E-07	7.00E-07	1.12E-05	1.10E-07	22.529	44.889	76.581	36.534	42.301	50.247	29.451	27.252	45.649	39.967	0.59786	258.25	87.938	0.04191	Angry
3	-15.208	-0.1027	0.22779	-0.01344	-0.14432	-0.07104	0.06985	0.30092	4.3512	-0.78209	-0.04885	1.14E-08	3.42E-07	4.53E-07	2.20E-06	3.74E-08	12.779	29.58	36.3	18.96	22.059	22.562	20.338	21.114	25.946	22.281	0.5909	256.42	83.523	0.02254	Angry
4	-14.215	0.03547	-0.04939	0.00404	-0.24424	-0.13046	0.16295	0.16639	4.015	-0.81706	-0.05688	9.02E-09	3.62E-07	7.78E-07	1.16E-05	4.77E-07	20.128	48.503	49.138	28.031	30.537	26.024	21.192	20.613	35.594	30.892	0.59955	267.6	85.202	0.0298	Angry
5	-13.67	-0.12183	-0.06	-0.20402	-0.11813	0.01519	0.01554	0.30936	3.896	-0.73162	-0.05269	1.77E-08	8.08E-07	8.95E-07	1.43E-05	2.80E-07	22.306	50.406	81.817	36.452	37.885	40.183	33.61	25.25	42.943	39.132	0.51496	248.39	87.867	0.04322	Angry
6	-14.068	0.36985	-0.17556	-0.03731	-0.18962	-0.03415	0.18986	0.02461	4.0335	-0.82594	-0.04188	4.88E-08	1.09E-06	5.64E-06	6.74E-06	1.25E-07	15.596	40.058	39.312	34.832	44.137	26.389	17.36	18.23	29.578	29.394	0.70359	267.49	87.66	0.03073	Angry
7	-14.715	-0.02733	-0.16542	0.02	-0.34943	-0.08343	0.18613	0.2576	4.1966	-0.67997	-0.04289	1.42E-08	4.99E-07	6.38E-07	6.25E-06	1.37E-07	15.776	40.24	45.887	29.902	23.438	25.875	22.55	17.463	24.079	26.385	0.61299	258.96	85.446	0.02837	Angry
8	-14.065	-0.08963	-0.223	-0.00304	-0.18205	0.00519	0.0045	0.14768	3.9957	-0.72019	-0.04928	3.25E-09	3.17E-07	2.38E-06	2.19E-05	3.76E-07	16.861	73.791	37.767	44.242	35.018	29.538	26.992	19.167	33.095	37.276	0.88576	283.69	88.231	0.04226	Angry
9	-15.092	0.35305	0.08797	-0.09856	-0.2085	-0.07765	0.20229	0.12581	4.3394	-1.0489	-0.06402	5.97E-08	8.87E-07	9.27E-07	6.78E-06	4.80E-08	14.273	31.883	66.818	28.916	27.195	28.621	22.179	16.366	31.352	27.894	0.71497	253.29	87.434	0.03655	Angry
10	-14.011	0.81981	0.20168	-0.11117	0.01975	-0.12407	0.14534	0.1889	4.0507	-0.81577	-0.0371	1.20E-07	3.29E-06	2.55E-06	1.03E-05	9.93E-08	32.732	48.828	85.35	45.964	41.591	48.505	31.706	21.544	41.831	41.371	0.79264	235	90.469	0.04986	Angry
11	-14.185	-0.10517	0.06304	-0.17445	-0.40848	-0.12673	0.12589	0.313	4.0214	-0.86259	-0.05897	4.27E-09	3.06E-07	1.57E-06	1.38E-05	1.58E-07	19.423	55.991	40.312	36.492	31.246	28.143	32.586	28.558	33.439	34.497	0.57806	295.55	86.467	0.0346	Angry
12	-13.156	-0.22136	-0.04447	-0.08512	-0.2982	-0.00857	0.18863	0.19342	3.7461	-0.63778	-0.04457	9.42E-09	3.70E-07	1.22E-06	1.36E-05	2.63E-07	23.223	62.165	50.091	39.959	38.091	34.605	33.384	31.419	42.885	40.03	0.62211	267.84	88.84	0.03923	Angry
13	-15.123	0.1167	0.31222	-0.06342	-0.08179	-0.15072	0.04873	0.2961	4.3696	-0.69249	-0.03693	8.11E-08	9.46E-07	7.14E-07	5.16E-06	2.86E-08	12.252	26.289	50.755	28.167	27.366	29.02	25.012	16.268	25.4	26.308	0.63219	241.68	86.524	0.03366	Angry
14	-14.393	-0.57059	-0.04663	-0.00815	-0.26462	-0.02614	0.07274	0.32532	4.1031	-0.67647	-0.0494	4.31E-09	5.03E-07	4.46E-07	1.20E-06	4.93E-08	17.754	25.18	37.572	25.369	21.307	26.014	18.698	13.262	27.96	22.805	0.61931	260.11	83.637	0.02177	Angry
15	-13.719	-0.35122	0.13391	-0.00782	-0.23921	-0.2277	0.30683	0.1305	3.9455	-0.72818	-0.04479	1.38E-08	5.71E-07	6.06E-07	3.72E-06	1.42E-07	21.121	39.363	54.3	31.544	25.475	28.093	26.351	29.672	34.215	30.638	0.50456	262.93	87.269	0.0326	Angry
16	-13.24	0.01111	-0.04753	-0.24975	-0.31461	-0.19903	0.21464	0.14643	3.7618	-0.64548	-0.03696	2.52E-08	1.02E-06	1.28E-06	2.18E-05	4.80E-07	20.905	58.757	90.475	39.635	34.908	41.619	29.566	24.293	38.411	39.461	0.66339	255.97	88.884	0.04719	Angry
17	-14.3	-0.20838	0.22683	0.04004	-0.30582	-0.14044	0.04306	0.27952	4.0557	-0.90121	-0.06163	5.95E-09	4.74E-07	7.55E-07	4.55E-06	1.77E-07	15.5	35.82	40.937	26.161	24.508	24.68	23.175	22.792	29.234	26.484	0.74831	267.04	84.093	0.02667	Angry
18	-12.961	0.10311	-0.17643	-0.23562	-0.25528	-0.07958	0.22381	0.10251	3.68	-0.6945	-0.03734	3.77E-08	8.78E-07	1.62E-06	1.49E-05	4.18E-07	23.325	53.396	82.84	42.52	37.091	42.48	31.095	23.994	38.488	39.49	0.72679	237.04	88.508	0.04306	Angry
19	-13.585	-0.13898	-0.1818	0.04038	-0.38674	-0.04358	0.39439	0.14855	3.8505	-0.67122	-0.04371	1.05E-08	3.60E-07	1.19E-06	6.64E-06	3.20E-07	22.022	45.777	46.936	32.731	34.391	26.885	24.852	22.708	30.864	31.71	0.58282	265.37	85.758	0.02756	Angry
20	-13.022	-0.36266	-0.10226	0.02186	-0.24903	-0.0165	0.24581	0.10281	3.6899	-0.70361	-0.05326	1.18E-09	5.28E-07	3.73E-06	1.12E-05	2.06E-07	23.958	58.698	36.852	44.046	42.888	29.272	34.901	31.503	40.198	39.276	0.9364	286.82	88.475	0.03761	Angry
21	-13.563	-0.28646	-0.03198	0.05633	-0.18172	-0.11508	0.10526	0.30822	3.8507	-0.64528	-0.04423	1.79E-09	4.85E-07	6.38E-07	2.98E-06	1.03E-07	19.211	38.847	45.249	31.286	28.384	29.727	22.245	20.502	28.748	28.416	0.661	276.52	85.686	0.02573	Angry
22	-13.593	-0.18941	-0.0117	0.15554	-0.09177	-0.15422	0.08886	0.2632	3.8511	-0.69653	-0.05095	8.38E-09	1.87E-06	1.66E-06	9.26E-06	4.40E-07	29.618	52.407	75.439	43.586	45.024	47.244	35.155	28.989	53.484	44.52	0.64992	284.17	86.893	0.04099	Angry
23	-15.609	0.16599	-0.05904	0.08784	-0.2442	-0.05403	0.06299	0.09781	4.3954	-1.0018	-0.07127	6.40E-10	3.69E-07	9.95E-07	5.06E-06	2.09E-07	14.506	29.982	31.661	27.698	26.116	23.567	17.359	14.619	23.907	22.978	0.51607	297.3	80.927	0.01968	Angry
24	-14.267	0.12298	-0.02954	-0.02611	-0.22022	-0.27384	0.09511	0.10555	4.0181	-0.75876	-0.04445	3.77E-09	4.68E-07	1.99E-06	7.37E-06	3.48E-07	20.848	40.856	41.944	31.747	34.419	27.444	21.084	20.8	27.697	28.974	0.37403	279.01	84.489	0.0263	Angry
25	-13.541	-0.13611	-0.11556	-0.03468	-0.23864	-0.12302	0.16064	0.10945	3.8198	-0.80192	-0.05536	1.63E-08	4.78E-07	3.92E-06	1.42E-05	1.67E-07	18.856	57.681	34.301	39.928	37.429	27.451	27.727	22.317	31.078	34.093	0.71064	269.05	86.477	0.03338	Angry
26	-13.897	0.17612	-0.04366	0.00264	-0.23659	-0.10796	-0.1772	0.18489	3.9315	-0.72559	-0.04489	2.26E-09	3.07E-07	1.74E-06	8.55E-06	1.88E-07	17.933	41.816	38.493	31.742	29.376	22.476	17.529	17.882	37.024	28.664	0.81408	288.99	86.564	0.03102	Angry
27	-13.201	-0.24404	-0.15427	-0.06884	-0.32094	0.04288	0.26289	0.16411	3.7558	-0.60935	-0.04206	3.18E-09	4.50E-07	8.72E-07	7.67E-06	2.44E-07	21.229	51.154	56.11	37.594	32.135	36.073	26.781	24.844	34.275	34.618	0.63141	262.59	88.034	0.03424	Angry
28	-15.117	0.22151	-0.01436	0.00297	-0.33914	-0.11743	0.07169	0.24911	4.2979	-0.78798	-0.05274	4.82E-09	5.65E-07	5.08E-07	2.85E-06	7.84E-08	14.018	29.355	45.358	22.532	23.788	23.984	19.14	15.73	22.913	22.909	0.4537	267.95	84.209	0.02284	Angry
29	-14.687	-0.05529	-0.23838	-0.00496	-0.27779	-0.10804	0.27739	0.19974	4.1731	-0.6478	-0.04348	9.95E-09	4.13E-07	1.01E-06	1.10E-05	5.25E-07	23.683	51.965	47.318	33.411	28.82	26.873	23.449	17.681	29.138	30.655	0.42776	278.7	85.737	0.03013	Angry
30	-14.744	-0.18545	-0.28924	-0.01669	-0.12408	0.05533	0.24024	0.15	4.1751	-0.75385	-0.05121	1.86E-08	4.19E-07	2.40E-06	1.53E-05	5.97E-07	20.415	55.692	34.964	37.884	34.589	24.495	21.071	18.783	29.014	31.887	0.67237	286.15	85.272	0.03302	Angry
31	-15.821	0.15867	-0.07299	0.04556	-0.23001	-0.08756	0.16007	0.07338	4.485	-1.0779	-0.06666	8.31E-09	5.35E-07	2.09E-06	9.36E-06	2.98E-07	17.087	41.217	36.998	29.113	27.948	23.021	18.826	16.962	21.733	25.179	0.28283	281.2	83.08	0.02573	Angry
32	-13.623	-0.3144	-0.07339	-0.06498	-0.07817	0.18545	0.08101	0.09884	3.8911	-0.56608	-0.03709	4.35E-08	6.11E-07	1.44E-06	6.98E-06	2.61E-07	21.468	50.615	45.61	34.291	38.959	34.37	24.132	21.038	39.523	35.179	0.86353	267.51	87.993	0.03475	Angry
33	-14.064	0.24752	-0.1838	-0.25098	-0.14578	0.12	0.19289	0.1338	3.9881	-0.73242	-0.05037	3.82E-08	7.95E-07	1.70E-06	2.14E-05	4.41E-07	21.898	57.679	79.696	30.642	40.77	35.428	24.705	19.603	41.267	37.532	0.50956	274.26	87.551	0.04297	Angry
34	-15.911	0.28473	-0.04253	-0.00534	-0.28438	-0.12842	0.1331	0.27845	4.5001	-0.93483	-0.07167	6.42E-09	4.09E-07	3.96E-07	1.49E-06	3.63E-08	11.757	23.649	29.471	20.231	19.558	19.564	15.263	13.728	17.492	18.465	0.54588	277.46	81.68	0.01749	Angry
35	-15.425	0.09111	0.07715	0.0076	-0.23026	0.00116	0.08127	0.32805	4.3907	-0.82305	-0.05298	8.90E-09	4.79E-07	4.15E-07	1.67E-06	2.49E-08	13.436	22.486	28.83	18.493	18.98	20.444	16.683	19.085	23.36	19.313	0.64424	255.16	81.986	0.01987	Angry
36	-13.517	0.19528	0.14093	-0.26993	-0.32642	0.06552	0.05914	0.21077	3.8564	-0.7039	-0.04334	4.03E-08	9.50E-07	1.32E-06	2.53E-05	1.60E-07	23.881	57.019	80.826	36.207	36.392	43.985	30.581	30.174	40.678	39.881	0.84675	240.64	89.022	0.05055	Angry
37	-13.988	0.58087	0.03187	-0.26206	-0.13547	0.02287	0.13355	0.20431	3.9942	-0.68462	-0.04494	2.59E-08	8.34E-07	1.19E-06	1.82E-05	1.56E-07	18.647	46.886	77.669	28.73	30.131	36.402	20.325	21.133	38.556	33.139	0.93356	253.25	88.659	0.04311	Angry
38	-13.992	0.03466	-0.01416	-0.11812	-0.16181	-0.17053	0.18286	0.26168	4.0203	-0.70228	-0.04413	4.01E-08	8.33E-07	2.13E-06	1.41E-05	3.11E-07	22.547	55.008	56.811	38.126	36.062	30.992	22.702	25.164	41.701	35.861	0.86266	275.76	89.648	0.04226	Angry
39	-13.971	-0.17098	-0.16745	-0.108	-0.18922	-0.05796	0.14131	0.20786	3.9392	-0.65472	-0.04475	1.22E-08	2.86E-07	1.01E-06	1.71E-05	6.61E-07	22.481	57.751	42.622	37.178	30.332	29.352	23.186	21.793	31.153	33.426	0.49135	269.42	85.936	0.03353	Angry
40	-13.285	0.22254	-0.08963	-0.02483	-0.08823	0.04088	0.1334	0.18479	3.79	-0.70587	-0.04779	1.96E-08	6.75E-07	1.48E-06	1.14E-05	2.14E-07	21.405	51.983	45.872	36.287	33.224	28.412	23.537	19.547	34.972	32.799	0.83178	258.86	88.209	0.03621	Angry
41	-12.969	-0.4928	0.06185	-0.06917	-0.37187	-0.10383	0.25211	0.28677	3.6928	-0.67267	-0.04594	5.44E-09	4.13E-07	1.13E-06	1.51E-05	1.80E-07	23.625	62.403	48.228	40.141	36.434	33.892	36.857	37.083	41.12	39.773	0.64211	270.96	88.494	0.03915	Angry
42	-13.723	-0.4811	-0.16584	-0.02479	-0.14996	-0.05291	0.08296	0.16653	3.8811	-0.64514	-0.04998	4.99E-09	6.12E-07	7.79E-07	3.84E-06	1.72E-07	16.065	39.012	52.916	36.499	29.567	34.438	29.485	23.235	35.885	32.881	0.53158	265.69	86.083	0.03339	Angry

Fig.3.2.2: Extracted audio features from TESS Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	MFCC 5	MFCC 7	MFCC 11	MFCC 13	Std MFCC	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Std Mel Sp	Spectral R	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Std Chrom	LPC 2	LPC 4	Mean LPC	Std LPC	ZCR	Total Ener	Mean Ene	Pitch	Intensity	Label
2	0.53478	0.0165	-0.09584	0.02512	4.1683	9.21E-06	2.92E-05	4.03E-05	0.00011	2.65E-05	823.51	17.787	35.52	38.536	56.816	87.874	164.83	155.24	77.778	17.976	54.966	1.2847	-0.11648	-0.08323	0.72331	0.01455	8632.3	55.335	258.82	99.65	Angry
3	0.37892	0.12207	-0.12701	0.13896	3.9353	1.15E-05	3.23E-05	5.26E-05	6.28E-05	1.91E-05	789.55	27.195	53.96	54.229	92.88	124.83	182.81	163.06	76.459	29.663	50.273	1.2591	-0.1966	-0.08323	0.72698	0.01281	7679.1	52.239	249.74	99.276	Angry
4	0.22607	0.13757	-0.04507	0.0314	4.4612	4.37E-06	5.97E-05	4.49E-05	0.00013	3.14E-05	503.58	22.996	57.612	68.376	84.413	121.96	201.83	143.79	112.21	24.753	55.926	0.97051	-0.12972	-0.08322	0.61375	0.01277	7300.7	82.962	294.51	99.46	Angry
5	0.63438	-0.12178	-0.12888	0.09734	4.1089	4.85E-06	3.26E-05	4.07E-05	0.0002	5.40E-05	977.16	22.427	63.678	69.235	75.076	131.37	190.97	147.44	86.486	22.707	59.035	1.2488	-0.14676	-0.08319	0.71371	0.02024	11923	102.78	258.52	97.363	Angry
6	0.63574	0.05792	-0.23519	0.14973	4.0125	4.45E-06	2.67E-05	4.08E-05	0.00017	4.95E-05	1010.9	25.635	57.332	71.132	81.8	133	200.48	145.37	73.368	25.58	59.807	1.4344	-0.15696	-0.08318	0.77968	0.02327	12630	98.675	257.45	96.243	Angry
7	0.48721	-0.03042	-0.10162	0.03163	4.0282	5.81E-06	2.73E-05	2.35E-05	0.00011	2.80E-05	965.34	19.215	45.02	49.564	63.735	115.3	145.74	145.57	93.149	20.889	49.463	1.4284	-0.22863	-0.08323	0.76291	0.0139	9890.7	62.206	278.38	99.297	Angry
8	0.01479	0.16622	-0.31905	-0.04246	4.0061	4.11E-06	4.41E-05	7.66E-05	0.00021	5.15E-05	496.77	26.164	79.906	93.815	118.5	206.07	259.54	202.34	117.55	31.654	75.75	1.0745	-0.17998	-0.08318	0.6802	0.01173	15107	132.51	280.76	99.504	Angry
9	0.32708	-0.08585	-0.19175	0.03055	4.0597	1.26E-06	3.32E-05	3.29E-05	0.00038	8.71E-05	1119.1	20.724	57.246	59.761	74.279	134.26	196.7	128.97	98.846	23.296	67.139	1.7052	-0.07523	-0.08318	0.84941	0.03002	21367	144.37	276.01	96.198	Angry
10	0.66671	-0.04231	-0.18245	0.19248	4.2329	7.80E-06	2.99E-05	3.77E-05	5.52E-05	1.67E-05	920.67	16.884	46.691	47.203	75.061	107.61	164.73	116.56	65.867	18.04	45.178	1.4075	-0.19862	-0.08323	0.76775	0.01669	7789	45.285	234.31	94.64	Angry
11	0.41567	0.03534	-0.15196	0.13458	3.9191	4.59E-06	4.22E-05	3.88E-05	0.00019	4.53E-05	1025.3	23.523	57.886	61.433	84.671	142.48	196.62	158.05	126.86	23.152	65.614	1.4257	-0.14977	-0.08314	0.77412	0.02305	19174	97.332	239.61	98.056	Angry
12	0.74223	0.01949	-0.20088	0.26987	4.366	7.91E-06	2.43E-05	3.85E-05	0.00018	3.27E-05	338.81	15.196	36.381	29.827	31.349	61.468	153.81	160.29	80.296	13.137	54.905	0.93454	-0.14502	-0.08324	0.63953	0.00851	6068.8	53.707	253.19	99.183	Angry
13	0.45271	-0.09716	-0.20645	0.30674	4.0379	5.29E-06	3.09E-05	3.25E-05	8.65E-05	2.43E-05	894.62	19.097	46.359	43.885	56.875	103.85	160.64	146.49	89.53	17.129	52.607	1.32	-0.16205	-0.08323	0.73398	0.01488	7953.1	56.405	271.75	99.199	Angry
14	0.56725	0.11937	-0.08247	0.22528	3.9875	7.60E-06	3.79E-05	3.70E-05	0.00014	3.85E-05	815.35	16.192	53.596	57.29	74.906	113.07	182.75	166.08	80.087	17.376	58.927	1.3158	-0.21438	-0.08322	0.74377	0.01274	15681	78.407	250.56	98.938	Angry
15	0.66853	0.09641	-0.20647	0.10374	3.9536	4.74E-06	3.97E-05	4.67E-05	0.00011	2.99E-05	1118	26.138	58.5	54.666	71.99	120.81	197.04	134.99	86.184	24.09	55.786	1.7096	-0.20729	-0.08319	0.86391	0.02502	15666	67.819	238.67	96.267	Angry
16	0.71766	0.22953	-0.28806	0.0746	4.0697	6.13E-06	4.10E-05	5.68E-05	0.00027	5.11E-05	781.96	22.744	47.169	42.233	64.91	92.619	194.17	141.73	122.37	21.36	59.904	1.2492	-0.18416	-0.08318	0.73628	0.01753	17757	84.555	239.73	97.334	Angry
17	0.41168	-0.17015	-0.04845	0.01468	3.3658	1.85E-05	0.00015	0.00029	0.00024	8.26E-05	972.54	33.245	69.493	103.11	96.368	140.62	342.43	234.18	96.723	33.982	96.697	1.7499	-0.33561	-0.08319	0.87574	0.01433	25330	166.65	256.21	103.15	Angry
18	0.24657	-0.1792	0.07153	0.10852	3.683	1.54E-05	9.75E-05	0.00012	7.68E-05	4.77E-05	1066.9	28.148	72.543	90.577	93.095	149.12	233.97	190.07	82.518	27.771	65.791	1.7862	-0.21247	-0.08319	0.882	0.02152	13978	105.9	260.22	97.953	Angry
19	0.41775	-0.27886	-0.01192	-0.04312	4.2987	1.56E-05	0.00019	0.00013	0.00027	6.19E-05	513.56	25.688	52.002	56.462	91.397	118.5	274.56	165.02	108.71	24.935	72.62	1.2306	-0.20292	-0.08322	0.71013	0.01245	10968	117.94	291.34	94.608	Angry
20	0.32469	-0.21301	-0.07808	0.08196	3.735	1.23E-05	0.00012	0.00013	0.00017	6.35E-05	1320.8	27.199	77.791	84.219	87.315	121.86	242.98	156.11	105.96	24.463	66.462	1.9963	-0.22926	-0.08317	0.94541	0.02884	17964	128.32	269.8	96.362	Angry
21	0.34377	-0.20602	-0.08475	-0.0609	3.3746	5.61E-06	0.00015	0.00017	0.00062	0.00013	1504	39.204	77.176	100.53	94.666	172.17	323.46	233.86	171.62	38.633	99.344	2.1726	-0.16692	-0.08315	1.007	0.0333	31187	229.32	267.42	98.882	Angry
22	0.34793	-0.27049	-0.07019	-0.01449	3.7365	1.65E-05	5.88E-05	7.19E-05	0.00013	3.44E-05	1216.1	34.92	67.73	67.449	86.916	108.89	184.95	171.15	90.897	30.977	51.876	2.101	-0.23428	-0.08319	0.98249	0.02343	14395	81.327	265.89	96.879	Angry
23	0.28188	-0.23557	0.07902	0.0107	3.8001	2.29E-05	0.00014	0.0001	0.00034	8.85E-05	458.44	33.38	92.432	104.75	171.29	245.63	287.55	209.11	136.59	31.406	85.199	1.4385	-0.26991	-0.08318	0.78694	0.01168	27289	208.31	281.56	97.392	Angry
24	0.34605	-0.20521	-0.11392	-0.04837	3.7526	1.46E-05	0.00014	0.00012	0.00053	0.00013	1060.4	33.018	70.067	101.86	79.613	136.02	299.21	186.26	55.915	33.003	89.262	1.8232	-0.19638	-0.08317	0.89031	0.02579	40402	224.46	266.73	97.453	Angry
25	0.31174	-0.1146	-0.04858	0.0715	3.8135	3.45E-05	0.00015	9.49E-05	0.00015	4.61E-05	973.14	29.33	63.275	70.486	124.95	142.42	273.61	204.28	55.178	28.67	74.729	1.7159	-0.20926	-0.08321	0.85882	0.02144	22343	107.42	245.16	97.057	Angry
26	0.15406	-0.21678	-0.05479	-0.01447	3.6218	2.38E-05	0.00015	0.00012	0.00044	8.79E-05	1099.2	33.461	73.752	83.598	101.72	150.86	285.08	226.68	134.76	32.977	83.475	1.8013	-0.23656	-0.08317	0.88721	0.02386	36352	165.24	240.71	97.754	Angry
27	0.4622	-0.25028	0.06501	0.00916	4.0128	3.19E-05	9.02E-05	7.96E-05	0.00015	3.73E-05	388.69	19.016	47.004	51.276	65.858	91.075	210.85	200.92	80.294	18.081	68.086	1.1341	-0.2344	-0.08323	0.6862	0.00749	11158	80.858	258.48	100.24	Angry
28	0.1837	-0.20626	0.09261	0.04591	4.0689	2.17E-05	0.0001	5.89E-05	0.00022	4.68E-05	790.03	15.253	30.392	36.216	62.105	88.935	193.61	178.74	125.81	17.084	67.348	1.17	-0.22143	-0.08322	0.68842	0.01001	15375	93.749	271.53	100.2	Angry
29	0.17316	-0.23952	0.12863	-0.05575	4.0069	2.64E-05	9.14E-05	9.84E-05	0.00015	3.75E-05	984.67	15.523	42.813	50.675	80.326	82.7	228.88	175.1	60.707	17.082	66.092	1.5383	-0.17077	-0.08322	0.80705	0.02064	17367	76.171	228.51	96.24	Angry
30	0.27472	-0.26882	-0.04646	-0.018	4.0693	2.66E-05	7.05E-05	5.07E-05	8.23E-05	2.28E-05	714.82	19.914	36.467	42.207	62.552	80.271	170.75	172.72	68.358	17.761	54.593	1.2895	-0.21851	-0.08325	0.7289	0.00908	16032	53.262	260.48	98.834	Angry
31	0.3229	-0.22377	0.00127	0.0052	4.0082	3.58E-05	8.86E-05	0.00013	0.00012	3.58E-05	712.65	20.108	37.397	42.158	80.072	77.095	226.25	194.96	60.215	20.698	69.79	1.4209	-0.25662	-0.08323	0.78361	0.01368	19100	72.9	232.23	96.572	Angry
32	-0.76354	0.22986	-0.04749	-0.00505	3.7291	6.69E-06	4.55E-05	6.71E-05	0.00022	4.49E-05	1030.8	19.564	53.413	65.745	89.786	127.86	220.66	172.25	112.94	24.473	64.585	1.5681	-0.19326	-0.0832	0.81946	0.01769	16118	91.581	257.24	100.64	Angry
33	-0.5096	0.06587	0.08682	-0.01893	3.8297	1.10E-05	4.64E-05	0.00012	7.86E-05	4.35E-05	985.42	23.776	69.731	87.852	116.54	164.51	228.48	167.51	79.257	28.107	61.252	1.54	-0.13126	-0.08319	0.80808	0.02042	15981	102.44	255.18	97.615	Angry
34	-0.45479	0.04596	0.05377	-0.09266	4.2568	6.62E-06	7.59E-05	0.00025	0.00017	5.57E-05	471.32	17.806	50.619	76.68	112.39	156.35	265.54	146.54	90.079	25.178	69.083	1.0697	-0.07445	-0.08318	0.66625	0.01354	12420	116.08	282.67	96.105	Angry
35	-0.28465	0.06154	-0.00268	0.00029	3.9027	4.01E-06	3.52E-05	0.00011	0.00014	5.03E-05	1008	22.988	67.447	72.049	113.22	152.44	216.41	140.85	101.33	27.099	60.532	1.5527	-0.14773	-0.08317	0.81099	0.02283	16839	112.26	262.31	96.076	Angry
36	-0.43155	0.07911	-0.11853	-0.02362	3.6373	4.84E-06	3.70E-05	7.78E-05	0.00025	6.75E-05	1204	28.002	72.238	92.355	100.99	180.41	227.96	170.65	121.66	32.807	72.084	1.8293	-0.13115	-0.08312	0.89734	0.03222	21260	141.73	262.93	98.22	Angry
37	-0.42254	-0.00217	-0.06458	-0.06478	3.4329	6.19E-06	6.49E-05	0.00019	0.00026	8.28E-05	1473.3	50.403	100.28	120.21	147.26	213.14	306.52	210.37	160.73	57.406	79.551	2.1784	-0.06581	-0.08306	1.008	0.03631	30332	187.23	262.32	99.511	Angry
38	-0.34156	-0.0921	-0.146	0.0157	3.2444	1.15E-05	0.00013	0.0002	0.00039	0.00015	644.73	73.947	164.11	197.89	239.32	441.33	419.55	316.22	236.33	75.371	125.38	1.4707	-0.10197	-0.08306	0.80142	0.01599	44704	375.66	260.35	101.83	Angry
39	0.022	0.05121	-0.13889	-0.04179	3.479	6.79E-06	0.00011	0.00022	0.00017	0.00016	1378.8	57.847	149.51	169.02	131.52	240.75	328.32	230.11	142.6	58.744	93.393	1.9779	-0.09114	-0.08305	0.93771	0.03728	44717	292.27	264.23	99.686	Angry
40	-0.20113	-0.07125	-0.09942	-0.00868	3.4289	1.14E-05	0.00011	0.0002	0.00015	8.49E-05	1380	53.621	112.2	141.93	168.12	252.79	347.57	230.36	163.52	57.446	89.901	1.912	-0.00441	-0.08309	0.92311	0.03577	36497	213.43	253.51	99.995	Angry
41	-0.26488	-0.14231	-0.13093	0.04644	3.0869	7.98E-06	7.62E-05	0.0002	0.0004	0.00013	1387	54.165	130.55	160.07	151.27	291.96	383.25	267.01	176.34	56.637	113.51	2.041	-0.09048	-0.08305	0.96893	0.03458	51844	283.3	236.66	102.42	Angry
42	-0.38087	-0.18602	-0.02858	0.14952	3.2337	8.50E-06	6.58E-05	0.0002	0.00032	0.00011	689.41	48.229	121.24	137.88	131.21	257.2	320.24	249.53	157.88	56.288	95.252	1.9652	-0.27342	-0.08314	0.94485	0.01452	31023	252.22	264.39	100.62	Angry

Fig.3.2.3: Extracted audio features from SAVEE Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	MFCC 3	MFCC 5	MFCC 6	MFCC 8	Mean MFCC	Mean Del	Std MFCC	Mean Del	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Mel Spect	Std Mel Sp	Spectral S	Spectral E	Spectral R	Chromagr	Std Chrom	Mean LPC	ZCR	Pitch	Label
2	-0.08184	0.39493	0.00515	-0.10136	-0.31779	2.0915	-0.02149	2.95E-07	7.88E-08	3.08E-06	3.73E-05	2.82E-05	3.68E-05	5.86E-06	1.94E-06	3.62E-06	2.32E-06	5.67E-06	2.18E-06	1.37E-06	1.20E-07	2.02E-05	4.7218	0.50159	2416.6	158.95	42.356	-0.06059	0.15175	294.04	Angry
3	-0.63489	0.08378	-0.02159	-0.31492	-0.54702	2.3424	-0.02247	1.34E-08	1.23E-08	2.82E-07	4.80E-06	2.98E-06	1.21E-05	8.04E-06	1.69E-05	1.69E-05	8.29E-06	3.99E-06	1.65E-06	4.31E-07	2.08E-08	7.43E-06	4.6857	0.52731	2353	132.22	22.228	-0.06471	0.14447	312.55	Angry
4	-0.71424	-0.02663	-0.18146	-0.12454	-0.51376	2.2463	-0.02726	5.24E-08	3.79E-08	2.35E-07	3.21E-06	2.18E-06	3.86E-05	1.82E-05	3.88E-05	2.55E-05	6.91E-06	4.82E-06	4.04E-06	1.38E-06	1.48E-08	1.05E-05	4.2583	0.53197	2196	179.74	23.575	-0.06475	0.14299	333.34	Angry
5	-0.50798	-0.02064	0.14188	-0.13929	-0.34671	2.0824	-0.0212	1.31E-07	1.60E-07	3.94E-06	3.29E-05	1.45E-05	5.11E-05	1.59E-05	7.05E-06	4.60E-05	1.17E-05	6.00E-06	2.11E-06	1.14E-06	3.04E-08	1.65E-05	4.2714	0.4944	1752.4	205.81	36.617	-0.07047	0.1024	307.19	Angry
6	-0.73992	-0.31014	-0.12302	-0.26859	-0.62865	2.2803	-0.02807	1.54E-08	1.00E-08	1.14E-07	1.83E-06	6.26E-06	4.11E-06	1.00E-05	1.61E-05	5.48E-06	5.15E-06	1.03E-05	1.78E-06	1.12E-06	1.60E-08	5.75E-06	3.2272	0.52075	2133.8	115.24	22.545	-0.06615	0.12719	305.09	Angry
7	-0.29564	-0.08246	0.3363	0.064	-0.43811	2.038	-0.04527	4.53E-08	1.07E-07	9.76E-07	6.97E-06	8.07E-06	1.18E-05	5.26E-06	2.64E-05	1.58E-05	7.77E-06	3.40E-06	4.93E-06	9.80E-07	1.05E-06	8.61E-06	2.6825	0.60329	3208.7	188.11	21.935	-0.05351	0.17561	264.52	Angry
8	-0.15822	-0.11312	0.27912	0.16976	-0.45435	2.1121	-0.03901	6.80E-08	1.63E-07	2.69E-06	1.14E-05	9.01E-06	9.63E-06	6.71E-06	1.35E-05	1.34E-05	6.85E-06	4.20E-06	2.55E-06	1.06E-06	7.12E-07	4.43E-06	3.5691	0.61273	3365	203.6	15.428	-0.04937	0.15252	257.4	Angry
9	-0.06166	-0.15402	0.26361	0.23269	-0.38227	1.9464	-0.03051	1.83E-08	6.50E-08	9.09E-07	6.89E-06	3.33E-06	1.19E-05	9.34E-06	5.06E-05	2.56E-05	1.68E-05	6.01E-06	1.80E-06	9.25E-07	1.01E-06	1.10E-05	3.9945	0.56823	3047.8	208.57	24.729	-0.05771	0.13449	292.43	Angry
10	-0.32278	-0.32217	0.18636	0.06096	-0.78822	2.8757	-0.05785	1.59E-08	1.33E-08	1.91E-07	1.25E-06	9.31E-07	7.70E-06	1.26E-05	7.68E-06	2.37E-05	8.74E-06	8.43E-06	1.61E-06	5.32E-07	3.91E-07	8.28E-06	5.7695	0.52293	2820.7	154.35	19.331	-0.05689	0.1454	328.45	Angry
11	0.11112	-0.27905	0.45553	0.08811	-0.4903	2.1891	-0.02284	7.16E-08	1.47E-07	2.18E-06	6.87E-06	7.66E-06	1.39E-05	8.95E-06	3.74E-05	2.00E-05	5.67E-06	4.36E-06	1.32E-06	9.40E-07	6.96E-07	8.46E-06	3.5945	0.58838	3258.8	190.03	13.663	-0.03596	0.18642	279.55	Angry
12	-0.11316	-0.25006	0.38738	0.03887	-0.50126	2.0208	-0.03146	2.68E-08	3.00E-08	4.90E-07	4.41E-06	6.79E-06	2.02E-05	1.50E-05	2.81E-05	4.29E-05	5.60E-06	2.58E-06	1.46E-06	4.23E-07	5.69E-07	1.01E-05	2.7369	0.58213	3402.7	178.05	14.605	-0.04643	0.18831	282.66	Angry
13	0.24217	-0.0832	0.32695	-0.01613	-0.51227	2.4063	-0.04475	1.18E-08	3.56E-08	4.81E-07	8.26E-06	4.53E-06	1.08E-05	5.15E-06	1.49E-05	4.41E-06	4.46E-06	3.49E-06	2.49E-06	7.29E-07	9.57E-07	4.80E-06	4.6247	0.52184	3061.5	160.76	21.507	-0.05453	0.17335	297.39	Angry
14	0.2863	-0.02541	0.18825	0.22061	-0.37864	2.1573	-0.02207	6.95E-08	3.43E-07	3.46E-06	2.31E-05	1.86E-05	1.17E-05	7.39E-06	9.71E-06	7.30E-06	3.67E-06	4.31E-06	1.68E-06	1.30E-06	4.56E-07	9.65E-06	5.214	0.56775	3194.3	183.31	24.089	-0.04761	0.18644	266.18	Angry
15	0.11689	-0.09322	0.22347	0.08815	-0.39665	2.085	-0.01744	9.81E-08	2.77E-07	5.05E-06	1.01E-05	1.54E-05	7.22E-06	1.18E-05	1.00E-05	3.24E-06	1.49E-06	8.65E-07	1.01E-06	7.10E-07	1.64E-07	5.81E-06	3.258	0.61264	3321.3	130.48	19.396	-0.06162	0.18576	260.99	Angry
16	0.31384	0.07574	0.29269	0.14701	-0.35929	2.0281	-0.021	7.19E-08	4.64E-07	9.59E-06	1.21E-05	3.70E-05	1.40E-05	5.00E-06	3.20E-05	1.35E-05	5.04E-06	3.42E-06	9.38E-07	4.83E-07	6.08E-07	9.47E-06	2.9297	0.58768	3293.9	168.89	22.871	-0.06452	0.15314	216.75	Angry
17	0.0342	-0.21533	0.46625	0.02831	-0.53468	2.3799	-0.02209	3.99E-08	1.70E-07	1.37E-06	5.77E-06	2.84E-06	6.30E-06	4.49E-06	1.25E-05	1.63E-05	6.40E-06	6.32E-06	8.75E-07	3.08E-07	2.97E-07	5.31E-06	4.219	0.55922	3070.8	132.94	16.13	-0.0597	0.14992	274.66	Angry
18	-0.11698	-0.22825	0.32822	0.16618	-0.49376	2.3447	-0.0207	3.17E-08	9.51E-08	1.06E-06	9.22E-06	9.97E-06	1.65E-06	5.77E-06	8.55E-06	7.35E-06	7.60E-06	3.85E-06	1.56E-06	1.11E-06	6.54E-07	3.68E-06	3.6895	0.56368	3262.4	134.3	17.363	-0.05886	0.14915	266.66	Angry
19	-0.18855	-0.34053	0.31016	0.20197	-0.49413	2.0451	-0.02228	2.30E-08	1.44E-07	6.01E-07	4.12E-06	1.43E-05	2.41E-05	7.89E-06	6.37E-06	1.15E-05	6.31E-06	2.87E-06	3.56E-06	2.50E-06	9.56E-07	6.28E-06	2.9903	0.59135	3669.5	214.97	18.439	-0.05806	0.17349	311.44	Angry
20	0.15346	-0.14237	0.31493	0.1633	-0.4448	2.424	-0.02888	1.50E-07	1.54E-07	2.10E-06	1.45E-05	6.97E-06	2.38E-06	5.06E-06	9.83E-06	2.77E-06	1.98E-06	1.97E-06	1.36E-06	7.86E-07	2.16E-07	5.06E-06	3.9681	0.51746	2543.4	108.77	22.16	-0.06316	0.11226	267.25	Angry
21	-0.48637	0.01971	0.01304	0.12639	-0.55031	2.1163	-0.04735	1.12E-08	5.23E-08	6.36E-07	2.05E-06	3.67E-06	5.50E-06	1.47E-05	1.19E-05	1.28E-05	1.17E-05	9.27E-06	3.89E-06	2.40E-06	6.04E-07	4.98E-06	2.0439	0.65239	3789.9	199.09	11.586	-0.04373	0.19429	257.78	Angry
22	-0.56558	0.10698	0.1693	0.04946	-0.51636	2.2909	-0.04162	2.04E-08	1.35E-07	4.79E-07	1.98E-06	1.99E-05	2.58E-06	2.91E-06	2.78E-06	9.93E-06	1.01E-05	3.53E-06	2.59E-06	2.66E-06	1.11E-07	5.33E-06	2.9962	0.61678	3111.7	142.43	14.166	-0.05541	0.16225	273.77	Angry
23	-0.06513	0.21685	0.25076	-0.02687	-0.5151	2.5392	-0.03561	6.09E-08	6.07E-07	2.95E-06	3.43E-06	2.85E-05	4.24E-06	2.09E-06	1.39E-06	2.29E-06	2.40E-06	1.31E-06	1.02E-06	9.43E-07	2.50E-07	7.35E-06	3.4976	0.56434	3134	117.82	14.521	-0.05801	0.1532	207.7	Angry
24	-0.32791	0.04689	0.29386	0.10562	-0.47546	2.25	-0.03229	1.60E-08	1.04E-07	2.21E-06	3.66E-06	6.91E-06	6.82E-06	7.69E-06	1.19E-05	3.50E-05	1.47E-05	3.93E-06	1.22E-06	2.13E-06	1.24E-06	7.94E-06	2.8896	0.60298	3269.4	216.65	15.702	-0.05512	0.17215	237.35	Angry
25	-0.58237	0.12219	0.3248	0.03152	-0.51072	2.322	-0.04103	1.34E-08	1.66E-07	2.12E-06	4.36E-06	6.54E-06	6.82E-06	2.25E-05	1.81E-05	2.30E-05	3.08E-05	2.58E-06	8.94E-07	2.17E-06	3.80E-07	8.84E-06	3.0765	0.59017	3070	177.48	29.246	-0.0598	0.15403	239.31	Angry
26	-0.13874	0.10219	0.10963	-0.13243	-0.45922	2.0335	-0.02823	2.55E-08	1.43E-07	2.38E-06	7.78E-06	2.21E-05	1.02E-05	5.81E-06	4.59E-06	4.34E-06	5.73E-06	3.43E-06	9.42E-07	1.42E-06	2.55E-07	4.87E-06	2.1244	0.63349	3636.3	168.2	18.969	-0.0351	0.19482	239.27	Angry
27	0.19548	0.51241	0.12922	0.15969	-0.2941	1.9757	-0.02273	4.06E-07	8.30E-06	3.83E-05	7.75E-05	4.97E-05	1.09E-05	3.45E-06	6.27E-06	7.42E-06	2.49E-06	5.36E-06	4.41E-06	3.75E-06	3.78E-07	2.26E-05	3.343	0.5775	3260	183.09	37.796	-0.05243	0.18388	184.89	Angry
28	0.02513	-0.02682	0.20438	-0.11511	-0.39411	2.0273	-0.0238	2.74E-07	9.92E-07	9.32E-06	7.45E-06	2.24E-05	1.12E-05	1.35E-05	1.97E-05	1.36E-05	1.07E-05	2.03E-06	5.21E-07	1.42E-06	3.80E-07	6.79E-06	3.0714	0.60446	3065.5	188.56	20.723	-0.06553	0.16095	205.01	Angry
29	-0.10174	0.00052	0.01977	-0.07567	-0.44201	2.0301	-0.02272	3.68E-08	3.97E-07	3.04E-06	3.78E-06	1.66E-05	6.94E-06	1.51E-05	4.22E-05	1.38E-05	1.51E-05	7.48E-06	1.60E-06	2.71E-06	5.67E-07	8.97E-06	2.7735	0.63357	3455.1	184.86	19.637	-0.05953	0.18063	215.54	Angry
30	0.01329	-0.1607	0.43485	0.0637	-0.44265	2.1604	-0.02797	1.16E-07	7.89E-07	5.04E-06	3.41E-06	5.90E-06	9.22E-06	1.17E-05	1.60E-05	1.20E-05	6.21E-06	2.20E-06	5.08E-07	5.97E-07	6.56E-07	4.57E-06	3.0694	0.58749	3240.7	163.19	17.861	-0.06124	0.1685	224.06	Angry
31	0.07091	-0.16802	0.40736	0.10869	-0.37843	1.9307	-0.02867	9.31E-08	1.09E-06	8.17E-06	7.08E-06	9.40E-06	1.77E-05	1.75E-05	2.46E-05	2.02E-05	5.62E-06	1.69E-06	1.78E-06	1.17E-06	5.76E-07	7.76E-06	2.7114	0.60807	3445.1	194.89	20.752	-0.063	0.17019	209.71	Angry
32	-0.09624	0.18377	0.12345	0.20759	-0.44817	2.3405	-0.02014	4.65E-08	9.60E-07	4.77E-06	4.89E-06	3.78E-06	4.75E-06	2.44E-06	4.08E-06	2.77E-06	5.63E-06	3.80E-06	1.61E-06	1.24E-06	2.83E-07	2.03E-06	2.3948	0.62194	3462.9	140.47	7.4634	-0.05883	0.18348	196.67	Angry
33	-0.09674	0.24379	-0.07581	0.07422	-0.41199	2.0942	-0.02335	2.05E-08	2.07E-07	4.87E-06	9.49E-06	2.18E-05	6.15E-06	6.23E-06	6.97E-06	2.94E-06	7.61E-06	4.06E-06	2.04E-06	3.05E-06	3.62E-07	5.10E-06	2.8096	0.58009	3437.1	178.15	18.576	-0.06021	0.16678	236.65	Angry
34	-0.80316	-0.69576	0.37516	0.07158	-0.71562	2.2314	-0.039	3.19E-09	8.10E-09	1.17E-07	1.18E-06	1.18E-06	5.74E-06	1.27E-06	1.02E-05	1.48E-05	9.87E-06	3.62E-06	3.81E-06	1.80E-06	5.68E-07	3.90E-06	1.3617	0.6708	4110.3	197.58	12.828	-0.029	0.23959	273.85	Angry
35	-0.57665	-0.73596	0.3267	0.19748	-0.90692	2.7007	-0.06211	2.29E-10	1.37E-09	3.36E-08	1.48E-07	1.54E-06	1.01E-06	3.55E-06	3.29E-06	4.94E-06	2.59E-06	2.11E-06	3.36E-07	1.13E-06	6.03E-07	1.34E-06	1.3112	0.6568	4531.9	126.66	7.9909	-0.03587	0.2144	270.19	Angry
36	-0.64767	-0.40837	0.15595	0.09422	-0.87728	2.7543	-0.04632	1.19E-09	1.02E-09	2.90E-08	3.54E-07	6.65E-07	2.68E-07	3.42E-07	2.09E-06	1.24E-06	1.04E-05	2.57E-06	5.13E-06	6.69E-07	4.17E-07	2.07E-06	1.4249	0.6782	4189.1	98.357	6.2037	-0.03099	0.2408	271.59	Angry
37	-0.81572	-0.45842	0.33505	0.38938	-0.6698	2.4375	-0.05067	1.40E-09	2.22E-09	5.45E-08	8.99E-07	8.21E-07	5.92E-06	2.60E-06	4.14E-06	1.18E-05	1.83E-05	5.31E-06	2.11E-06	1.18E-06	3.20E-07	3.78E-06	2.312	0.62585	3826	152.94	14.042	-0.0394	0.20349	288.46	Angry
38	-0.30992	-0.40744	0.15981	-0.02531	-0.56645	1.9905	-0.02257	1.04E-08	1.56E-08	3.39E-07	5.42E-06	1.20E-05	5.46E-06	9.30E-06	2.71E-05	9.28E-06	6.02E-06	5.09E-06	6.10E-06	2.46E-06	4.49E-07	5.39E-06	2.5113	0.63645	3789.9	184.05	13.182	-0.02416	0.20361	284.3	Angry
39	-0.37874	-0.60431	0.07814	0.11315	-0.727	2.2774	-0.02872	4.77E-09	1.03E-08	6.75E-08	2.60E-07	1.38E-06	7.63E-06	7.05E-06	8.21E-06	1.76E-05	2.46E-06	1.05E-06	2.22E-06	9.93E-07	3.69E-07	3.57E-06	1.3138	0.69554	4332.2	146.55	8.2631	-0.0121	0.25011	278.92	Angry
40	-0.20804	-0.47785	0.1825	0.2968	-0.85391	2.8458	-0.05193	7.80E-10	3.29E-09	7.85E-08	6.36E-07	4.69E-07	1.17E-06	3.41E-06	3.98E-06	1.65E-06	3.67E-06	4.56E-06	7.93E-06	1.20E-06	4.99E-07	1.90E-06	1.9424	0.61647	4166	117.48	7.7351	-0.01495	0.23811	303.32	Angry
41	0.20188	-0.05373	0.27657	0.28617	-0.37815	2.0191	-0.02652	3.85E-08	1.68E-07	5.44E-06	1.73E-05	1.81E-05	9.22E-06	2.44E-06	9.52E-07	3.10E-06	1.45E-06	2.63E-06	2.30E-06	2.18E-06	1.02E-06	5.89E-06	1.7398	0.61555	4298.5	166.26	26.574	-0.05071	0.21488	242.19	Angry
42	-0.09504	-0.38093	0.14584	0.28289	-0.56631	2.2079	-0.03233	1.90E-08	9.70E-09	1.47E-07	2.29E-06	1.33E-05	3.14E-06	3.98E-06	4.07E-07	1.96E-06	1.84E-06	1.78E-06	2.50E-06	5.78E-07	3.26E-06	1.0632	0.6572	4447.3	155.1	16.527	-0.04352	0.25612	282.9	Angry

Fig.3.2.4: Extracted audio features from EmoDB Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	MFCC 1	MFCC 3	MFCC 5	Mean MF	Std MFCC	Mean Del	Std Delta	Mean Del	Mel Spect	Spectral S	Spectral E	Spectral R	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Chromagr	Mean Chr	Std Chrom	Mean Ene	Pitch	Intensity	RMS	Label
2	-11.103	0.28414	0.39382	-0.52391	3.2597	-0.04305	0.27432	-0.00348	1.92E-06	7.8968	0.55123	1786.3	67.128	87.81	66.223	90.381	56.523	70.445	91.887	79.93	97.143	95.661	69.654	94	80.565	13.969	8.9338	146.05	85.686	0.0428	Angry
3	-10.089	0.30569	-0.07744	-0.46271	2.9844	-0.04567	0.3014	-0.00554	2.88E-06	6.5988	0.55542	2246.2	87.896	102.75	71.127	102.35	59.571	81.642	126.54	82.307	98.925	112.81	77.376	112.44	92.977	19.592	11.133	149.48	88.351	0.05023	Angry
4	-12.929	0.36208	0.31436	-0.58546	3.7883	-0.04479	0.29492	-0.00114	5.52E-07	9.4973	0.53271	1323.6	31.762	36.67	22.306	34.986	18.958	25.476	34.26	28.85	26.74	38.44	24.462	42.819	30.477	7.1835	1.403	133.46	80.086	0.01669	Angry
5	-12.08	0.35644	0.15186	-0.55181	3.5544	-0.03488	0.22644	-0.00311	5.38E-07	7.7195	0.54469	1784.8	38.595	44.49	28.717	42.003	24.665	31.501	45.923	34.3	38.43	48.573	35.835	49.008	38.503	7.8392	1.7027	149.64	82.946	0.02137	Angry
6	-11.987	0.16866	0.33763	-0.54178	3.5196	-0.03548	0.24256	-0.0031	8.44E-07	6.7449	0.50862	1422.4	37.583	52.555	27.448	50.202	23.717	28.624	55.707	32.877	53.778	49.985	39.796	60.04	42.693	12.509	3.5981	144.11	83.064	0.02687	Angry
7	-11.036	-0.06315	0.41838	-0.52528	3.25	-0.03873	0.25859	-0.00123	1.14E-06	6.8961	0.57763	1798	55.517	67.601	46.117	65.776	45.278	56.604	77.249	56.804	58.67	66.95	60.494	64.566	60.135	9.1	3.6812	138.76	84.798	0.02958	Angry
8	-10.801	0.50422	0.36871	-0.48781	3.1743	-0.03497	0.23796	-0.00139	2.68E-06	7.6218	0.55912	2400	69.106	80.376	59.083	87.094	53.032	69.269	84.258	76.513	76.133	92.22	72.549	88.375	75.667	11.848	5.4434	146.8	87.44	0.03842	Angry
9	-9.8708	0.11339	-0.15142	-0.45861	2.9427	-0.03055	0.19062	-0.00305	2.13E-06	6.8456	0.5206	1485	75.662	106.19	80.405	97.15	60.171	75.314	116.81	76.246	95.86	99.586	82.532	115.59	90.126	17.738	11.671	149.51	89.864	0.05618	Angry
10	-11.818	0.35153	0.24431	-0.55687	3.4616	-0.04211	0.26641	-0.00299	6.48E-07	6.9809	0.53597	2068.7	46.703	56.634	37.281	57.013	32.204	39.184	56.98	45.059	57.787	64.88	47.763	65.792	50.607	10.891	3.024	166.43	83.071	0.0256	Angry
11	-10.94	0.59514	-0.02121	-0.48498	3.2432	-0.03248	0.20239	-0.00322	1.35E-06	8.0892	0.5295	1618.9	63.432	78.838	51.945	81.204	40.894	49.849	87.079	61.221	82.162	88.195	66.803	95.255	70.573	17.367	7.6588	129.3	87.629	0.04356	Angry
12	-10.714	0.62892	0.04489	-0.48177	3.1647	-0.04011	0.25348	-0.00457	1.89E-06	7.4913	0.54129	2218.1	69.523	89.636	63.506	95.19	50.954	65.385	92.89	76.117	90.763	108.36	74.093	111.33	82.313	18.627	9.153	156.01	87.721	0.04607	Angry
13	-11.531	0.17407	0.31895	-0.52214	3.3846	-0.02964	0.19203	-0.00108	9.70E-07	5.1473	0.57461	2334.4	42.624	52.212	32.899	50.8	32.422	38.228	52.309	40.463	41.234	45.141	38.268	51.014	43.135	7.1968	1.812	125.98	83.721	0.02213	Angry
14	-10.392	0.53942	0.38828	-0.49369	3.0222	-0.03799	0.24429	-0.00321	2.77E-06	5.8848	0.59454	3345.2	100.09	112.27	96.003	104.62	75.726	88.156	110.96	101.78	112.38	115.36	113.97	114.4	103.81	12.278	7.9244	160.23	87.162	0.04477	Angry
15	-10.133	0.37603	0.08425	-0.4827	2.9745	-0.03408	0.22477	-0.00133	1.01E-06	6.0319	0.58435	2498.7	82.404	89.528	65.655	91.581	62.839	66.597	92.954	69.341	86.666	92.841	73.931	104.08	81.534	13.449	6.7795	164.25	88.385	0.04095	Angry
16	-10.499	0.01466	0.14555	-0.61856	3.0012	-0.03737	0.24283	0.00056	1.19E-07	6.3905	0.58714	2643.8	123.68	129.39	144.5	193.99	183.42	148.2	123.77	128.71	135.56	183.08	168.15	154.43	151.41	25.177	24.27	205.42	87.27	0.07016	Angry
17	-12.422	0.3395	0.30047	-0.60087	3.6105	-0.03604	0.22529	-0.00132	4.86E-07	8.0324	0.54945	1707.7	40.932	50.602	42.119	47.403	50.844	53.32	53.972	40.958	48.784	54.992	45.469	55.583	48.748	5.3936	3.6862	153.56	80.991	0.02462	Angry
18	-11.898	0.09638	0.28008	-0.61766	3.4532	-0.04041	0.25532	-0.00256	1.58E-07	7.9648	0.54167	1989.1	58.48	75.285	65.558	75.396	59.092	55.16	75.077	62.797	85.902	83.416	74.963	70.402	70.127	9.9261	13.317	177.73	83.186	0.04142	Angry
19	-11.759	0.18749	0.2605	-0.58505	3.4177	-0.03465	0.22011	-0.00051	1.86E-07	8.3392	0.55388	1923.2	65.231	75.468	64.786	69.515	55.051	55.976	86.922	70.863	75.62	80.578	69.589	79.499	70.758	9.6025	8.1197	181.78	83.265	0.03733	Angry
20	-10.51	0.17495	0.07041	-0.5114	3.0542	-0.03391	0.21556	-0.00132	2.84E-07	5.9398	0.552	2177.7	107.43	121.94	122.53	140.68	127.68	123.5	147.49	123.88	117.34	139.78	140.75	143.76	129.73	12.413	19.385	226.84	87.178	0.06511	Angry
21	-11.757	0.11965	0.35241	-0.55955	3.42	-0.03338	0.21167	-0.00148	5.60E-07	6.1535	0.56961	2307.2	43.434	54.32	39.764	48.19	40.815	37.674	42.662	35.057	37.634	48.701	49.86	55.285	44.45	6.7159	2.6453	203.26	82.141	0.02222	Angry
22	-13.266	0.49041	0.52219	-0.62426	3.8537	-0.03451	0.22296	-0.00046	4.57E-07	7.2243	0.52681	2081	27.922	27.782	21.024	27.72	18.72	19.544	24.086	20.535	22.452	26.517	24.506	29.651	24.205	3.7238	0.64847	172.97	79.114	0.01267	Angry
23	-11.659	0.20104	0.14621	-0.55718	3.417	-0.03689	0.23592	-0.00186	9.74E-07	7.2631	0.53957	1595.6	42.542	45.454	34.524	48.54	41.31	35.4	45.744	40.252	48.083	53.57	45.052	56.336	44.734	6.5227	2.392	183.72	83.568	0.02472	Angry
24	-12.218	0.42614	0.36648	-0.59737	3.5549	-0.0398	0.25796	-0.00106	4.19E-07	7.8713	0.5523	2309.7	41.666	44.329	39.984	51.521	34.988	33.737	46.175	41.677	57.336	50.5	36.632	52.568	44.259	7.5192	2.798	202.64	82.071	0.02347	Angry
25	-12.816	0.68827	0.14068	-0.6118	3.7425	-0.03525	0.22066	-0.00029	6.43E-07	7.7519	0.52053	1722.4	23.543	30.225	19.472	27.965	18.833	20.949	29.176	25.864	28.47	34.394	28.546	30.92	26.53	4.8732	1.5067	164.59	81.177	0.01779	Angry
26	-12.257	0.6222	0.09026	-0.67159	3.549	-0.03842	0.24436	-0.00115	1.66E-07	7.9549	0.5369	2013	36.986	64.036	49.46	59.318	40.596	38.259	53.621	40.674	64.834	55.603	41.91	51.418	49.726	9.9902	4.2861	168.54	82.916	0.02893	Angry
27	-8.7743	-0.36262	0.01156	-0.54417	2.5083	-0.03777	0.206	-0.00368	2.80E-07	3.0886	0.61624	3752.1	148.61	155.86	139.15	147.36	170.08	143.53	157.6	129.3	152.09	165.09	168.57	191.18	155.7	16.445	24.835	266.32	91.366	0.0762	Angry
28	-12.016	0.45927	0.48989	-0.61965	3.463	-0.04031	0.25085	-0.00056	3.82E-07	6.0965	0.58254	3195.9	49.502	56.387	50.611	56.453	46.508	47.745	60.168	47.302	57.606	64.61	51.852	58.521	53.939	5.8167	2.4215	178.45	82.006	0.02316	Angry
29	-9.6636	0.28529	-0.01735	-0.49595	2.7943	-0.03654	0.231	-0.00395	4.07E-07	5.2913	0.58553	3187.9	118.16	117.1	100.34	115.43	102.51	112.83	127.58	107.57	111.88	122.19	109.04	137.23	115.16	10.421	13.27	233.62	90.008	0.05668	Angry
30	-11.193	0.11041	0.13046	-0.64743	3.2077	-0.03106	0.19346	0.00011	1.05E-07	6.9179	0.56426	2198.2	93.114	98.445	121.76	124.08	126.1	114.29	101.81	97.127	114.31	146.61	113.51	114.15	113.78	15.025	19.583	201.61	85.901	0.05898	Angry
31	-10.278	-0.46866	0.06218	-0.69173	2.9308	-0.04294	0.24631	-0.00187	9.21E-08	6.9933	0.55856	2016	138.02	205.07	152.94	145.88	133.69	135.31	178.31	147.92	168.39	153.88	147.97	145.7	154.42	20.517	35.679	180.48	88.005	0.08473	Angry
32	-11.741	0.04353	0.24993	-0.57404	3.4253	-0.04117	0.27224	0.00013	1.95E-07	8.843	0.5245	1378.4	68.897	65.816	57.565	71.616	49.634	49.38	67.348	64.545	57.842	69.003	46.716	71.696	61.672	9.1164	8.0001	180.81	84.142	0.03775	Angry
33	-11.114	-0.00608	0.23107	-0.54903	3.2357	-0.03677	0.23637	-0.00217	1.94E-07	8.6027	0.53749	1724.2	87.591	115.94	74.901	90.507	81.442	90.045	105.8	79.758	85.437	103.79	103.2	100.25	93.222	12.434	15.129	216.88	86.212	0.0534	Angry
34	-9.1237	-0.24428	-0.16422	-0.61657	2.6205	-0.03638	0.2225	-0.00361	1.82E-07	4.7829	0.56612	2298.2	140.73	133.73	138.71	139.01	117.93	126.5	141.89	131.55	178.75	173.19	132.72	144.49	141.6	17.664	20.304	188.49	91.091	0.07191	Angry
35	-10.653	-0.14321	0.26905	-0.60835	3.0655	-0.02562	0.16562	-0.00053	1.76E-07	6.0669	0.57232	2530	95.411	104.05	99.178	104.48	102.73	90.49	89.943	89.164	98.008	135.57	116.48	124.51	104.17	14.483	20.088	227.56	86.267	0.05742	Angry
36	-10.869	0.31318	0.22108	-0.54508	3.1521	-0.02549	0.16134	-0.0011	2.93E-07	6.598	0.55989	2477.9	86.147	96.383	91.684	86.198	72.18	74.397	85.495	80.667	91.15	108.37	99.256	101	89.411	10.799	10.943	201.22	86.637	0.04863	Angry
37	-8.7334	-0.4531	-0.0963	-0.52954	2.5223	-0.02441	0.13837	-0.00207	1.03E-07	5.5272	0.53128	2186.9	161.86	152.94	143.66	156.97	173.81	155.09	163.45	131.51	138.23	186.05	167.86	186.7	159.84	17.306	31.651	271.42	93.627	0.09335	Angry
38	-11.155	0.02329	0.29825	-0.56443	3.2433	-0.02778	0.17811	-0.00086	4.03E-07	7.3651	0.54328	2005.9	76.993	86.635	69.339	84.972	79.773	82.338	86.026	76.125	71.335	89.146	83.601	87.845	81.177	6.5011	9.7152	203.6	85.878	0.04324	Angry
39	-8.4124	-0.00711	-0.35768	-0.4102	2.4573	-0.02153	0.11911	0.00146	3.08E-07	4.8572	0.5218	2435.4	126.79	168.49	132.97	134.95	134.54	155.02	208.13	122.02	154.87	183.09	165.49	168.63	154.58	25.72	32.936	272.52	95.118	0.09938	Angry
40	-10.291	0.13924	-0.28057	-0.63691	2.9651	-0.03423	0.21673	6.10E-06	1.06E-07	6.6201	0.57176	2323.9	134.1	135.88	125.54	140.1	125.79	113.37	124.4	119.22	118.65	157.66	113.06	151.31	129.92	14.302	21.282	203.34	88.233	0.06646	Angry
41	-10.15	-0.18247	0.31783	-0.51799	2.9447	-0.03102	0.193	-0.00146	4.18E-07	5.521	0.59257	2652.8	113.8	108.73	109.91	121.62	107.33	98.4	109.87	106.78	103.83	145.69	127.45	132.4	115.48	13.686	14.609	222.75	87.448	0.05341	Angry
42	-11.032	0.65794	0.33703	-0.46202	3.2278	-0.01767	0.1436	0.00057	1.02E-06	11.68	0.51847	2920	84.918	85.503	76.071	82.957	73.955	84.218	91.545	78.345	81.952	86.018	76.074	90.352	82.659	5.6145	6.3156	229.91	88.216	0.03833	Angry

Fig.3.2.5: Extracted audio features from CREMA-D Dataset

3.3 Feature Selection

3.3.1 Method – 1

It is important to select a specific set of features from a large set of features. This determines the type and category of the audio signal. It is important to choose the optimal number of features to maintain accuracy and performance levels and reduce computational cost. The selected feature must have the following properties –

- **Relevance**: The relevance of selected features is paramount in ensuring that the captured information aligns with the task's objectives, such as classification or prediction. Features should encapsulate essential patterns or characteristics in the data pertinent to achieving the desired outcome. Irrelevant features not only introduce noise but also add unnecessary complexity, potentially hindering model performance. By focusing on relevant features, we streamline the model's learning process and enhance its ability to extract meaningful insights from the data.
- **Discriminative Power**: Features with high discriminative power play a pivotal role in distinguishing between different classes or categories within the dataset. These features effectively capture unique characteristics or attributes that differentiate one class from another. By selecting features with strong discriminative power, we empower the model to accurately classify or predict outcomes, leading to improved performance and reliability. Harnessing the discriminative power of features enhances the model's ability to capture subtle nuances and variations in the data, facilitating more nuanced decision-making.
- **Robustness**: Robust features maintain their stability and informativeness even in the presence of variations or noise in the data. They exhibit resilience to irrelevant changes, such as fluctuations in scale, noise interference, or minor perturbations. Robust features ensure consistent performance across different conditions and environments, bolstering the model's reliability and generalizability. By prioritizing robust features, we mitigate the risk of model overfitting and enhance its adaptability to real-world scenarios.
- **Independence**: Feature independence refers to the degree to which selected features are uncorrelated with each other. Highly correlated or redundant features can introduce multicollinearity issues, complicating model estimation and interpretation. Independent features, on the other hand, offer unique and complementary information, enriching the overall representational capacity of the feature set. By selecting independent features, we enhance the model's ability to capture diverse aspects of the data while minimizing redundancy, thus facilitating more stable and interpretable model outcomes.

- **Interpretability**: Interpretable features are those that offer clear insights into the underlying characteristics or mechanisms driving the observed patterns in the data. These features are understandable to stakeholders, facilitating model validation, decision-making, and trust. By prioritizing interpretable features, we enhance the transparency and accountability of the model, enabling stakeholders to grasp the rationale behind its predictions or classifications. Interpretability fosters confidence in the model's outputs and encourages effective utilization in real-world applications.

There is a huge dataset, so not all features can be used for further classification. The selected features must have properties such as invariance to Relevance, Discriminative Power, Robustness, Independence, and Interpretability to other features. The extracted features must be selected in a specific way by a search algorithm. The aspect selection algorithm is Kendall's rank coefficient, Principal Component Analysis (PCA), Extra Trees Classifier, Sequential Forward Search (SFS), Sequential Backward Search (SBS), SelectKBest, Recursive Feature Elimination (RFE) and many more. Here, Extra Trees Classifier can be applied to obtain the best set of aspects. Extra Trees Classifier builds multiple decision trees using random subsets of data and features. It introduces additional randomness by selecting random thresholds for each feature and node. This randomness helps reduce overfitting and computational costs. Predictions are made by averaging the predictions of all trees for classification or taking the average for regression.

We've meticulously crafted Feature Density distribution plots for each selected feature extracted from every dataset, providing a comprehensive visualization of the distribution patterns within the data. Below, we present these plots, offering insights into the density and spread of each feature across different datasets. These visualizations serve as invaluable tools for understanding the variability and characteristics of the extracted features, facilitating informed decision-making in subsequent analysis and model training processes.

3.3.2 Method – 2

Selecting the right features is a crucial aspect when crafting effective audio classification systems. In this endeavor, we must prioritize certain key characteristics to ensure the system's efficacy and efficiency:

1. Irrelevant Invariance: A robust feature set should be resilient to irrelevant changes such as background noise, variations in scale, or fluctuations in bandwidth. This resilience empowers the classification system to adeptly handle diverse audio formats, thereby bolstering the accuracy of classification outcomes.

2. Discriminative Power: Features with strong discriminative power are paramount. They should exhibit distinct values within the same class while showcasing noticeable disparities across different classes. This discriminative capability is pivotal in accurately distinguishing between various audio samples, thereby enhancing the system's classification accuracy.

3. Uncorrelatedness: Minimizing redundancy within the feature space is essential. Each selected feature should provide unique and non-redundant insights into the audio signal. By ensuring features are uncorrelated, computational efficiency is optimized, and the system's performance is elevated.

Given the sheer volume of data within audio datasets, utilizing all available features is impractical. Hence, meticulous feature selection becomes imperative, with a focus on properties like resilience to irrelevance, selectivity, and independence.

Numerous feature selection algorithms are at our disposal, including Sequential Backward Search (SBS), Sequential Forward Search (SFS), Forward Sequential Floating Search (SFFS), Backward Sequential Floating Search (SFBS), and more. Among these, SFFS stands out for its adaptability and effectiveness. It commences with an empty feature set and dynamically adjusts based on performance metrics. This iterative approach facilitates comprehensive exploration of the feature space, gradually incorporating features that enhance classification performance. By adapting the feature set in response to real-time performance feedback, SFFS ensures optimal feature selection, thereby bolstering classification accuracy and efficiency.

Exploring Feature Distributions: RAVDESS Dataset

Fig 3.3.2.1 – Fig 3.3.2.30 depict the Feature Density Distribution plots for the RAVDESS Dataset, illustrating how the 30 selected features are distributed across five different emotion classes. These visualizations provide valuable insights into the density and variability of each feature within the distinct emotional categories, enhancing our understanding of the dataset's emotional nuances.

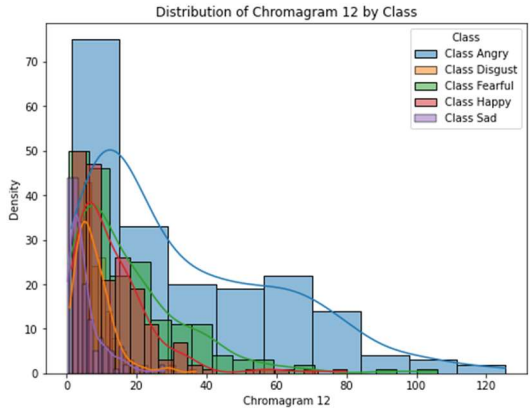

Fig 3.3.2.1: Distribution plot of Chromagram 12

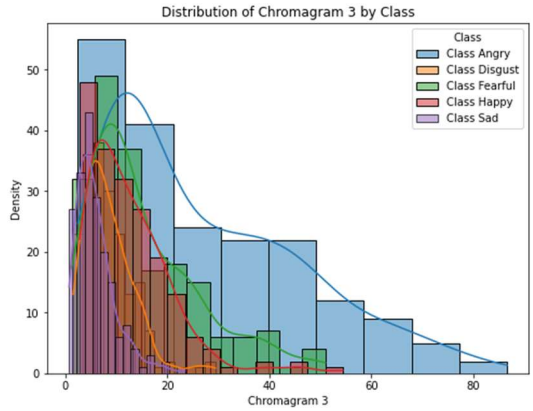

Fig 3.3.2.2: Distribution plot of Chromagram 3

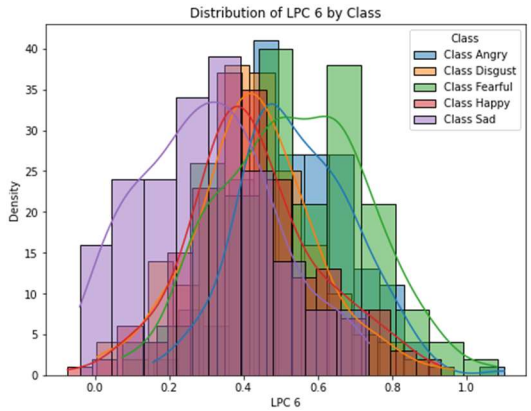

Fig 3.3.2.3: Distribution of LPC 6

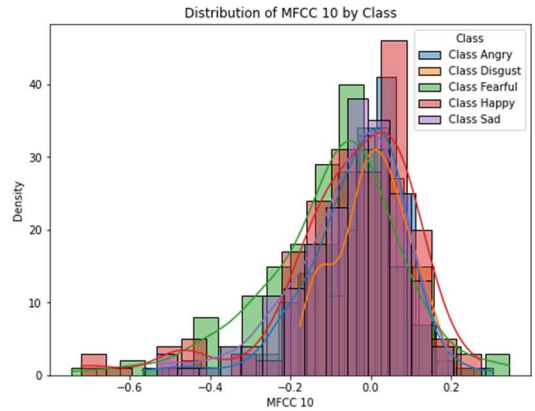

Fig 3.3.2.4: Distribution of MFCC 10

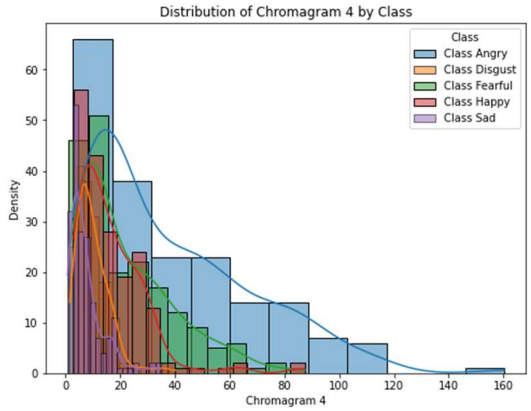

Fig 3.3.2.5: Distribution of Chromagram 4

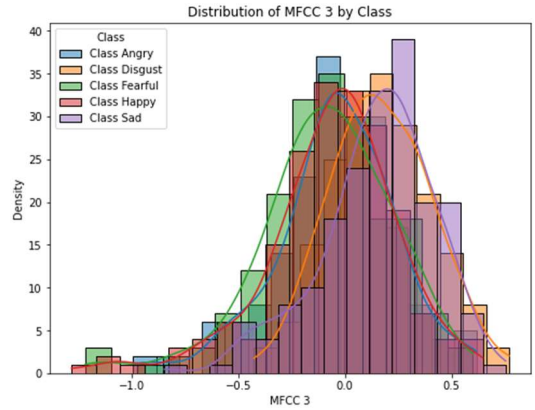

Fig 3.3.2.6: Distribution of MFCC 3

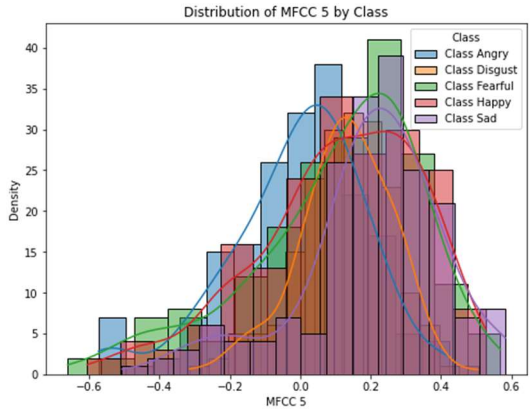

Fig 3.3.2.7: Distribution of MFCC 5

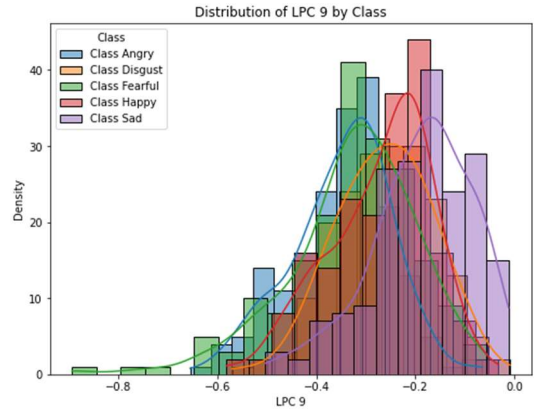

Fig 3.3.2.8: Distribution of LPC 9

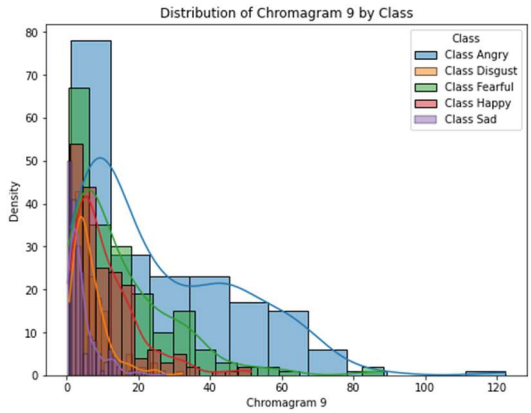

Fig 3.3.2.9: Distribution of Chromagram 9

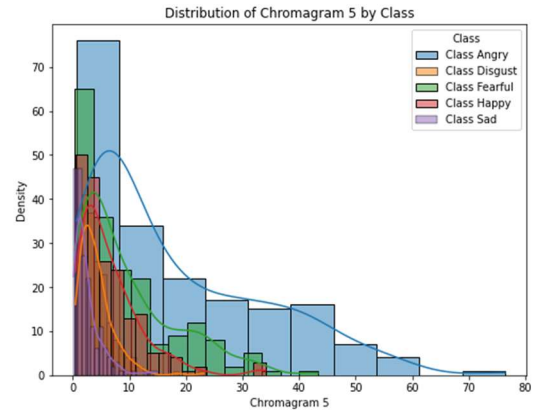

Fig 3.3.2.10: Distribution of Chromagram 5

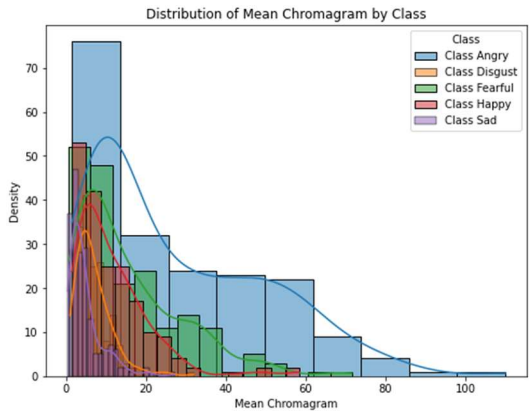

Fig 3.3.2.11: Distribution of Mean Chromagram

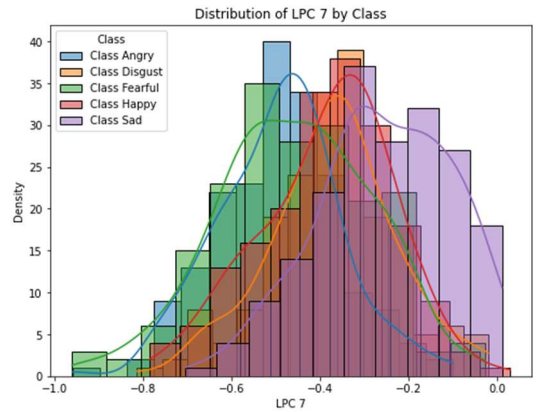

Fig 3.3.2.12: Distribution of LPC 7

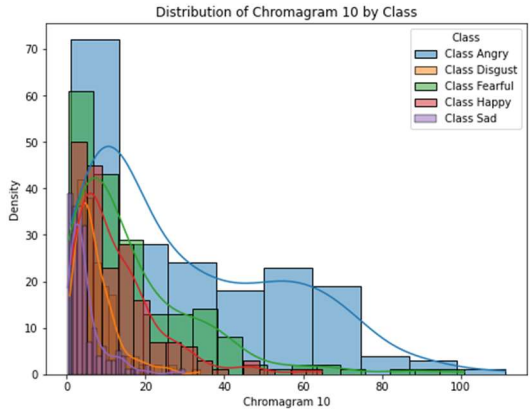

Fig 3.3.2.13: Distribution of Chromagram 10

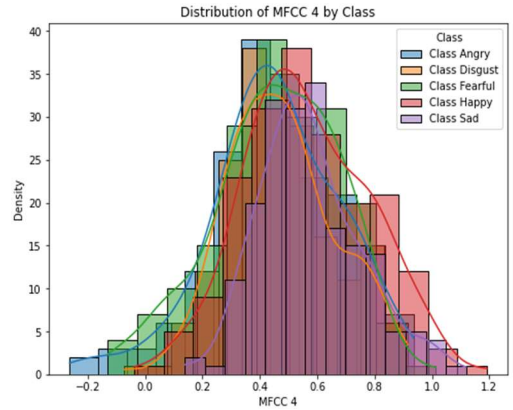

Fig 3.3.2.14: Distribution of MFCC 4

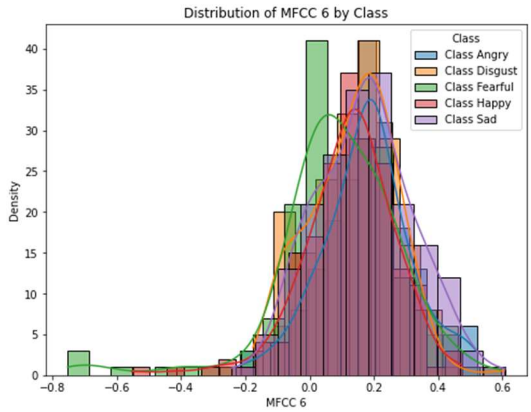

Fig 3.3.2.15: Distribution of MFCC 6

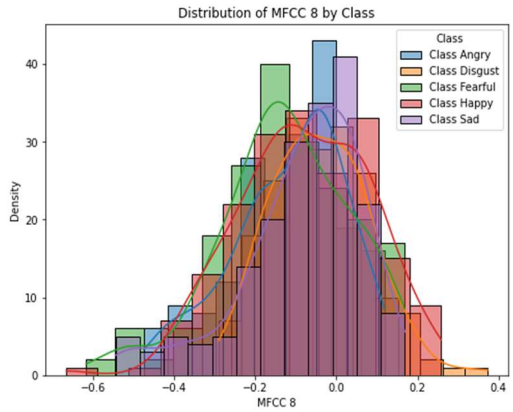

Fig 3.3.2.16: Distribution of MFCC 8

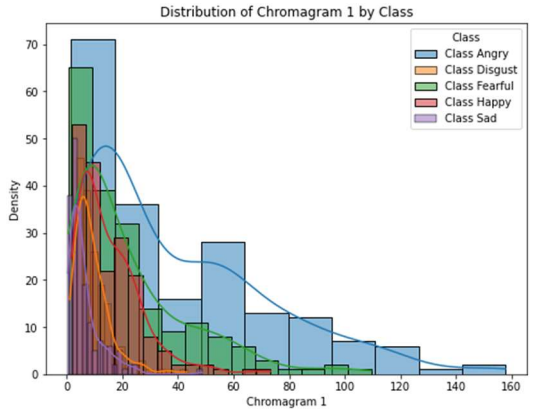

Fig 3.3.2.17: Distribution of Chromagram 1

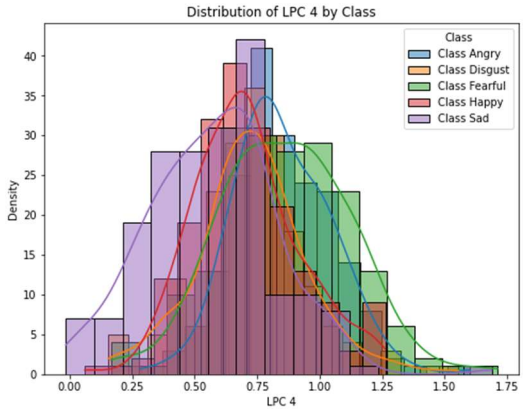

Fig 3.3.2.18: Distribution of LPC 4

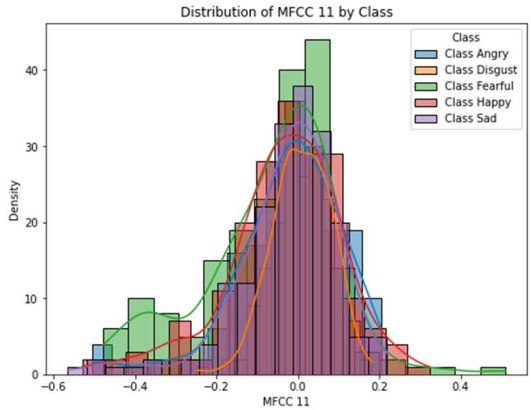

Fig 3.3.2.19: Distribution of MFCC 11

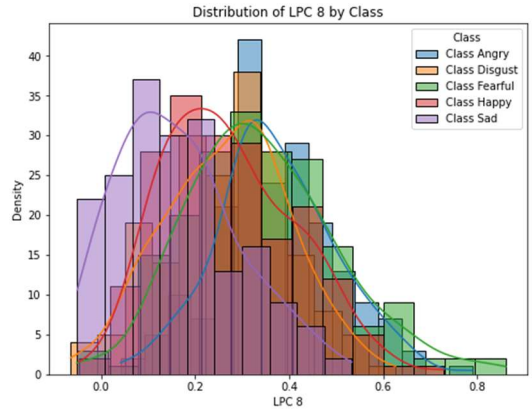

Fig 3.3.2.20: Distribution of LPC 8

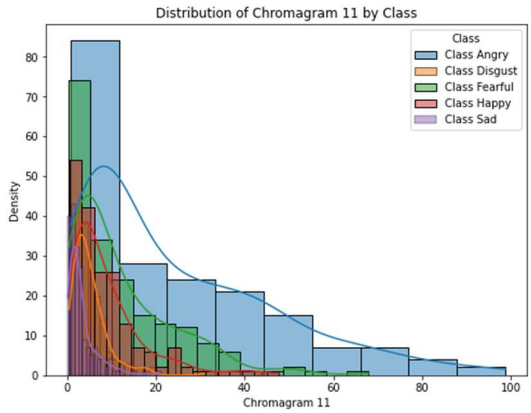

Fig 3.3.2.21: Distribution of Chromagram 11

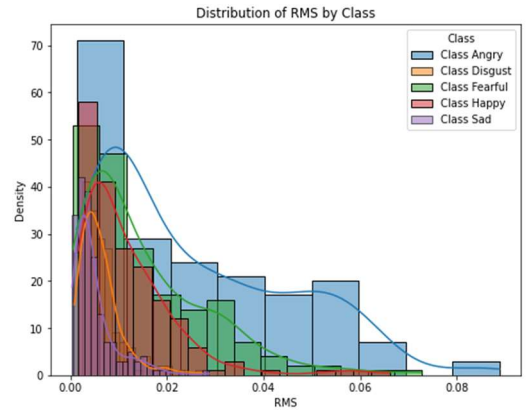

Fig 3.3.2.22: Distribution of RMS

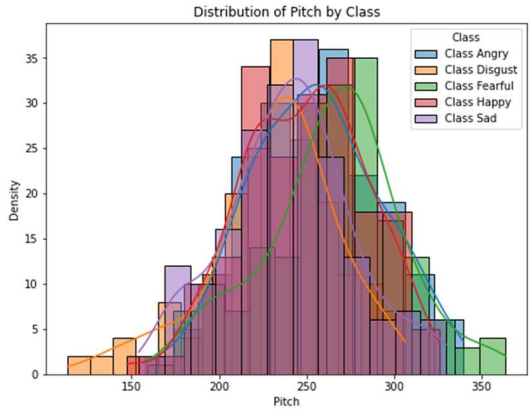

Fig 3.3.2.23: Distribution of Pitch

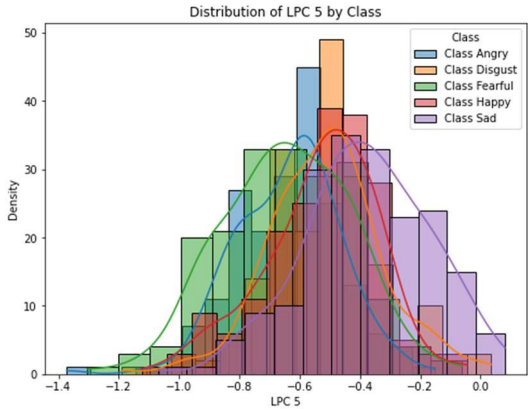

Fig 3.3.2.24: Distribution of LPC 5

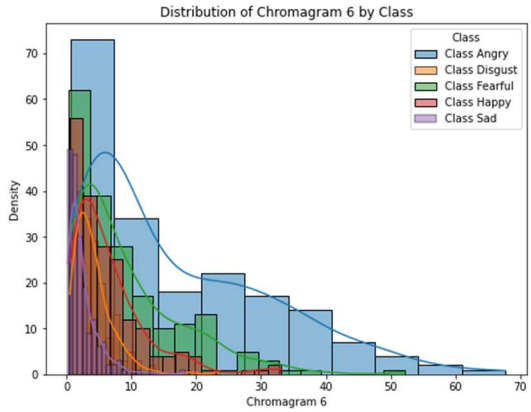

Fig 3.3.2.25: Distribution of Chromagram 6

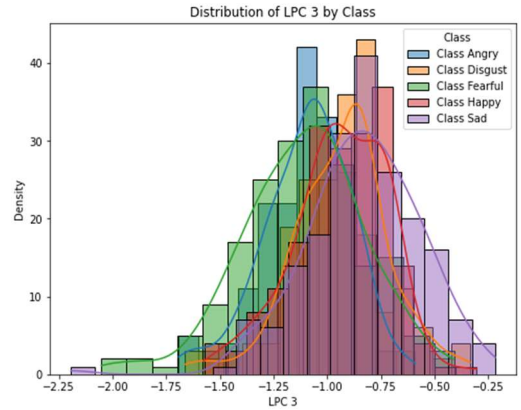

Fig 3.3.2.26: Distribution of LPC 3

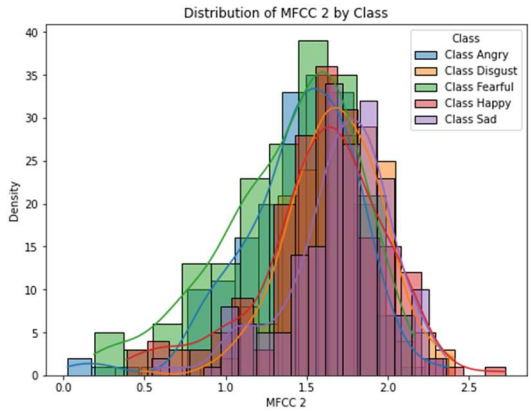

Fig 3.3.2.27: Distribution of MFCC 2

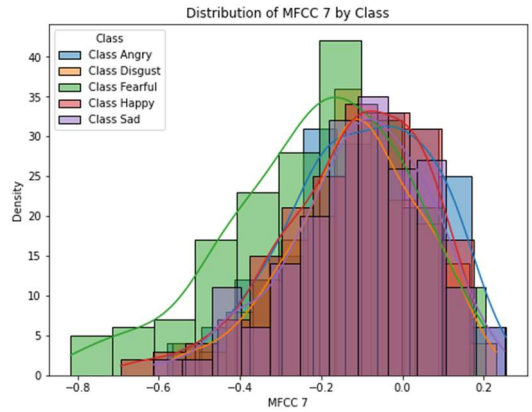

Fig 3.3.2.28: Distribution of MFCC 7

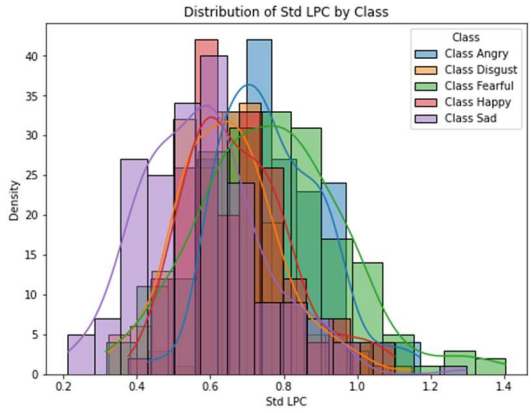

Fig 3.3.2.29: Distribution of Std LPC

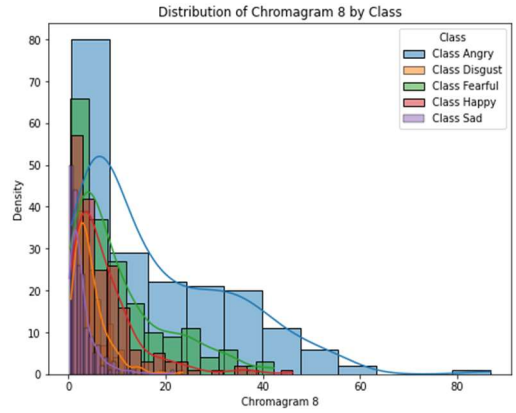

Fig 3.3.2.30: Distribution of Chromagram 8

Exploring Feature Distributions: TESS Dataset

Fig 3.3.2.31 – Fig 3.3.2.60 depict the Feature Density Distribution plots for the TESS Dataset, illustrating how the 30 selected features are distributed across five different emotion classes. These visualizations provide valuable insights into the density and variability of each feature within the distinct emotional categories, enhancing our understanding of the dataset's emotional nuances.

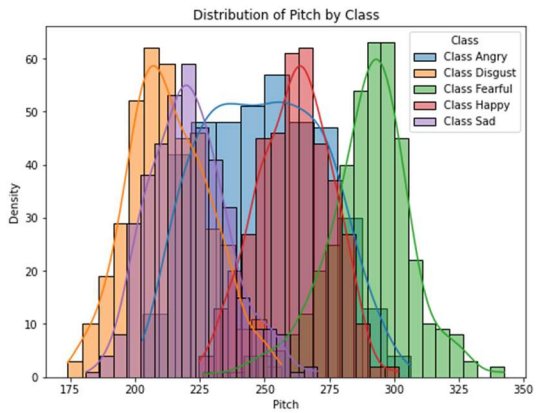

Fig 3.3.2.31: Distribution of Pitch

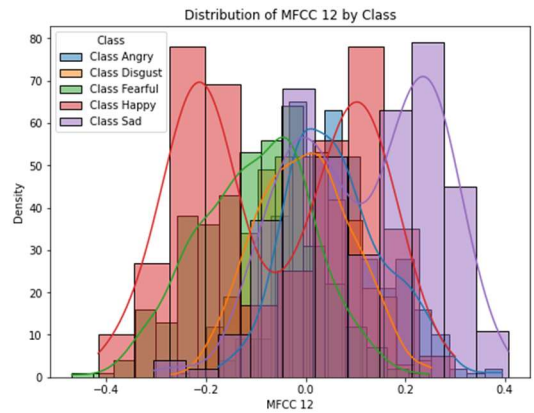

Fig 3.3.2.32: Distribution of MFCC 12

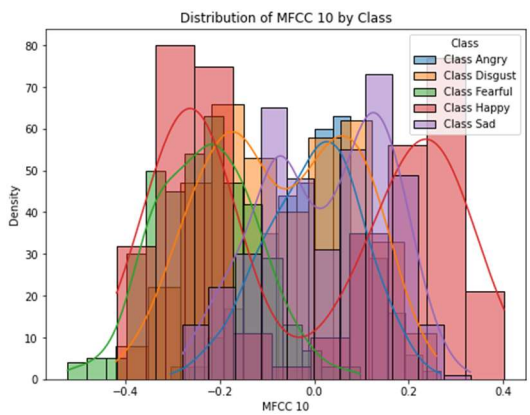

Fig 3.3.2.33: Distribution of MFCC 10

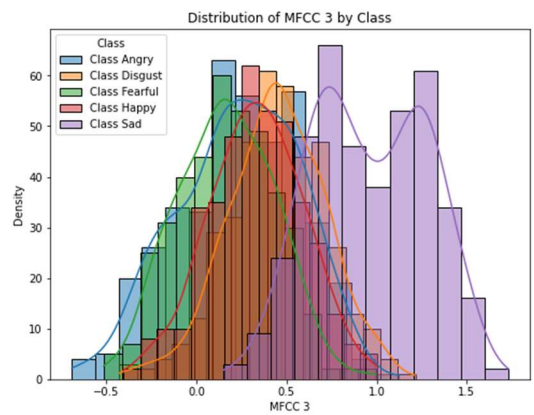

Fig 3.3.2.34: Distribution of MFCC 3

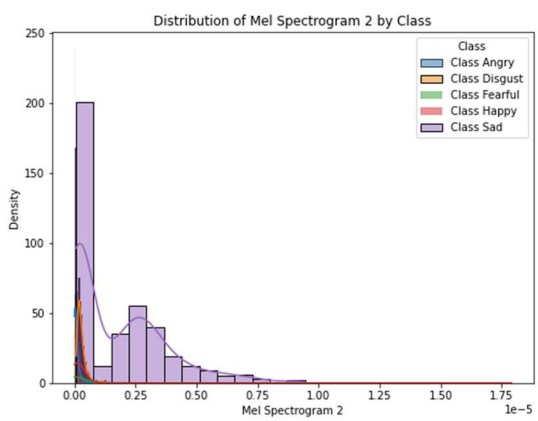

Fig 3.3.2.35: Distribution of Mel Spectogram 2

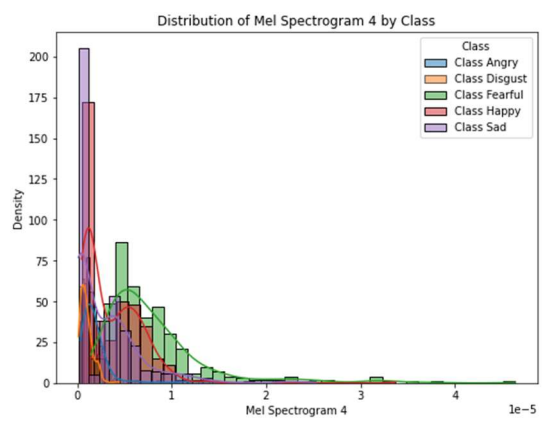

Fig 3.3.2.36: Distribution of Mel Spectogram 4

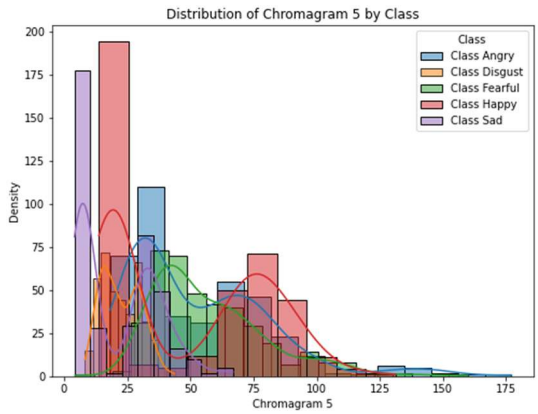

Fig 3.3.2.37: Distribution of Chromagram 5

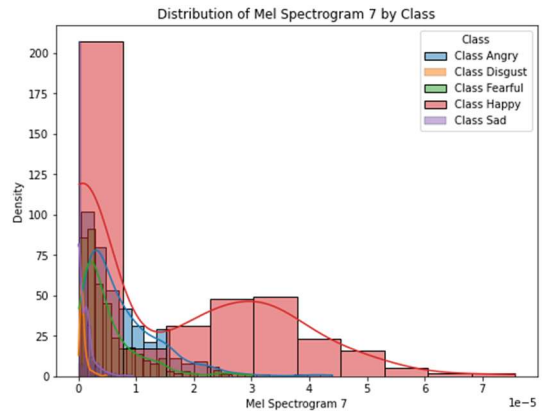

Fig 3.3.2.38: Distribution of Mel Spectogram 7

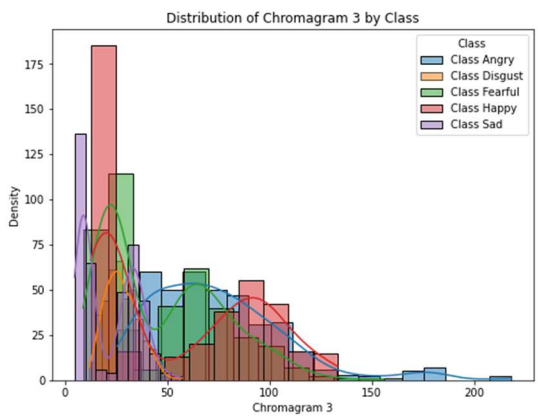

Fig 3.3.2.39: Distribution of Chromagram 3

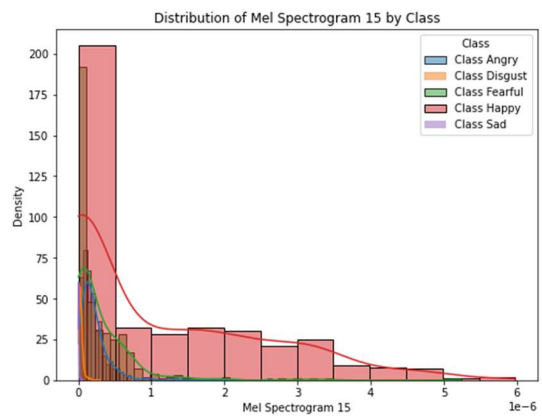

Fig 3.3.2.40: Distribution of Mel Spectogram 15

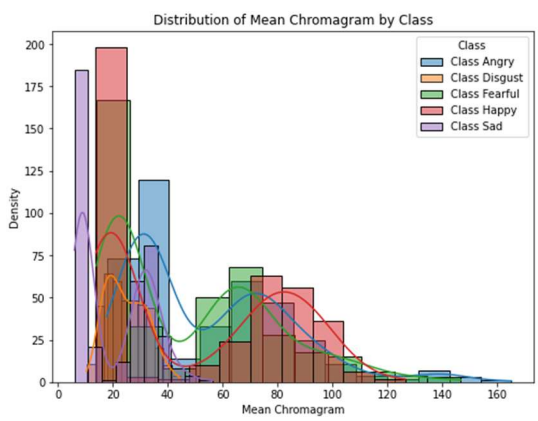

Fig 3.3.2.41: Distribution of Mean Chromagram

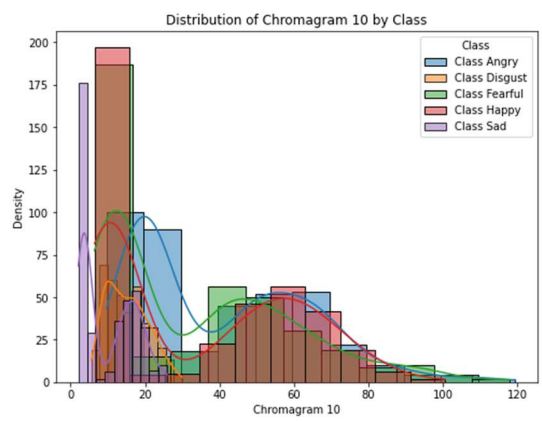

Fig 3.3.2.42: Distribution of Chromagram 10

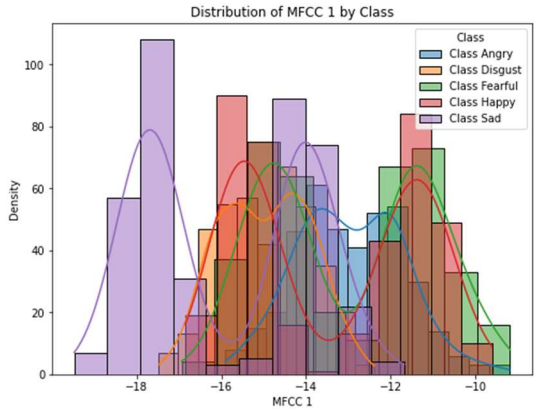

Fig 3.3.2.43: Distribution of MFCC 1

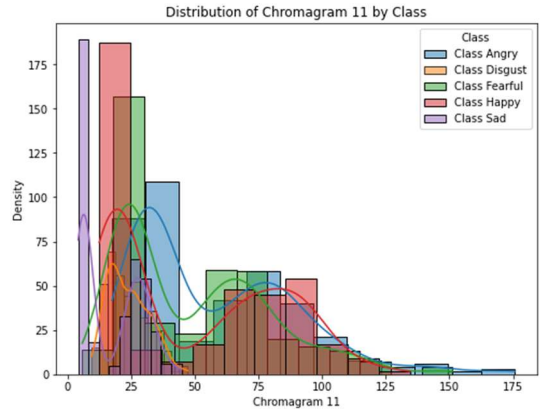

Fig 3.3.2.44: Distribution of Chromagram 11

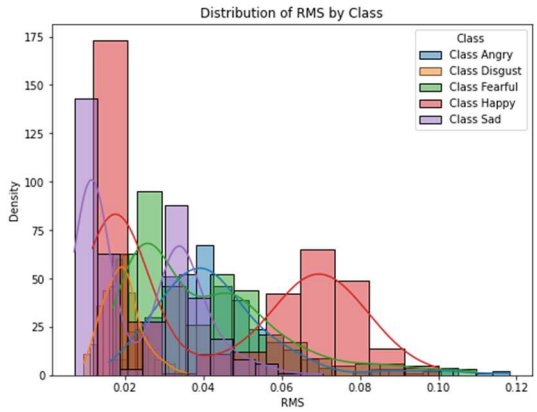

Fig 3.3.2.45: Distribution of RMS

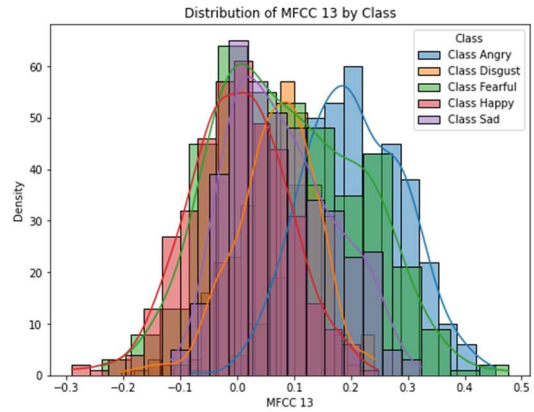

Fig 3.3.2.46: Distribution of MFCC 13

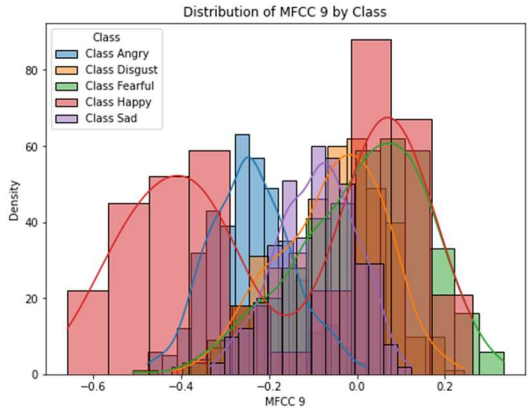

Fig 3.3.2.47: Distribution of MFCC 9

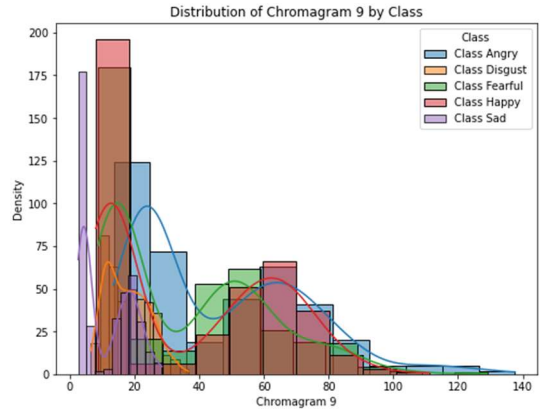

Fig 3.3.2.48: Distribution of Chromagram 9

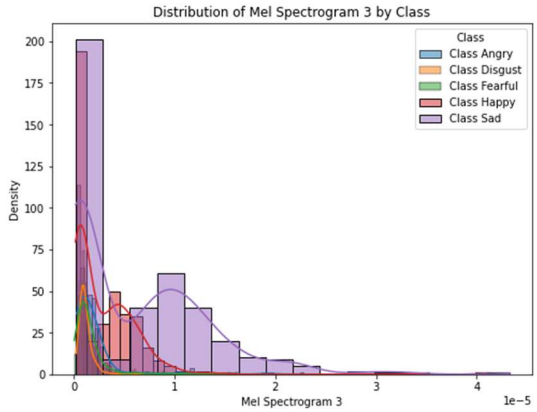

Fig 3.3.2.49: Distribution of Mel Spectogram 3

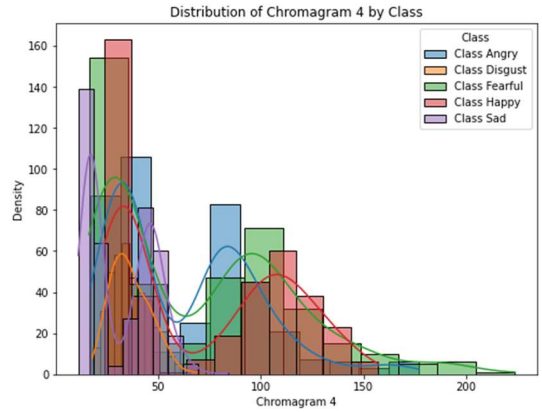

Fig 3.3.2.50: Distribution of Chromagram 4

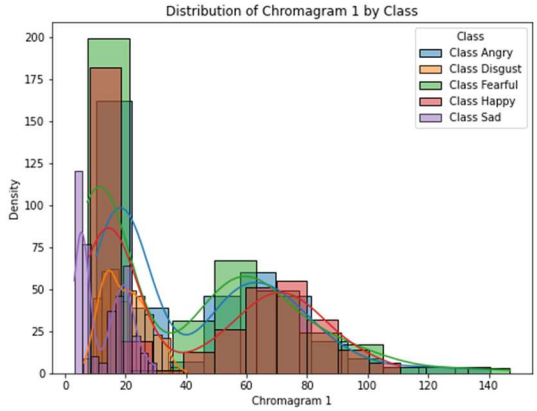

Fig 3.3.2.51: Distribution of Chromagram 1

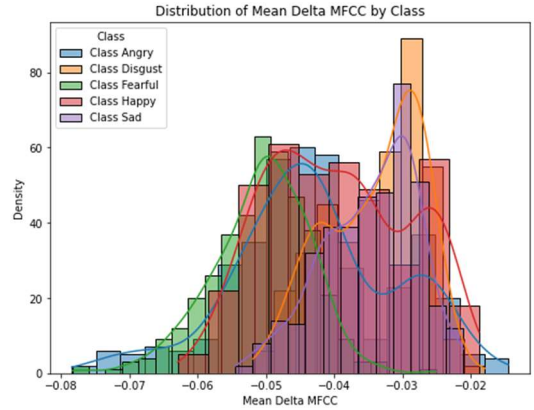

Fig 3.3.2.52: Distribution of Mean Delta MFCC

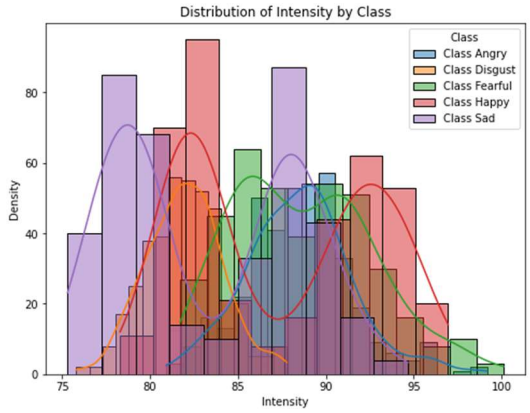

Fig 3.3.2.53: Distribution of Intensity

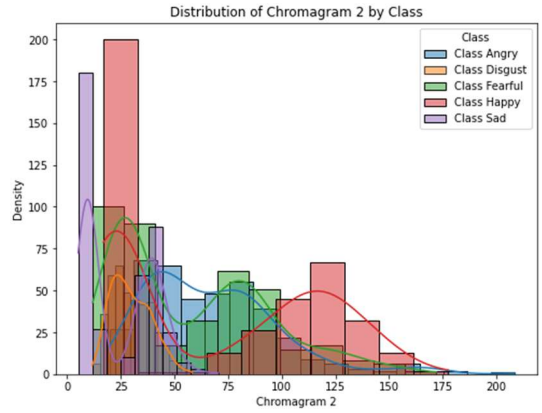

Fig 3.3.2.54: Distribution of Chromagram 2

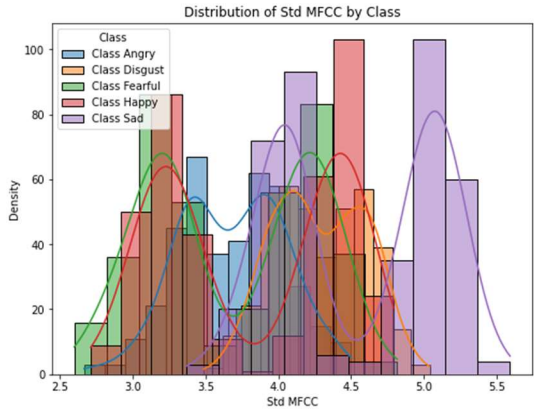

Fig 3.3.2.55: Distribution of Std MFCC

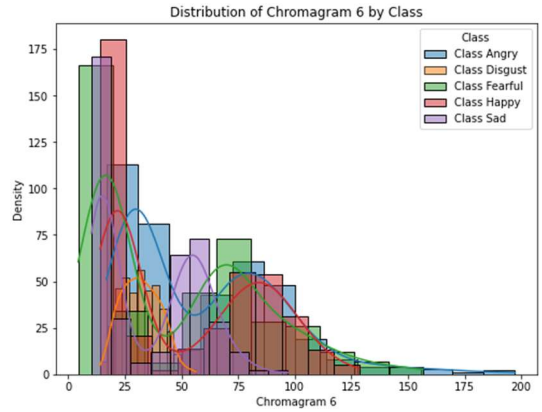

Fig 3.3.2.56: Distribution of Chromagram 6

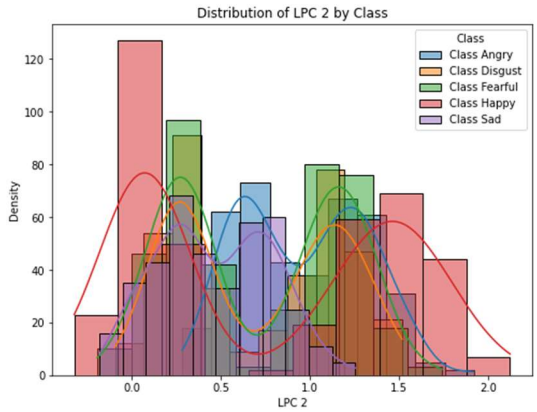

Fig 3.3.2.57: Distribution of LPC 2

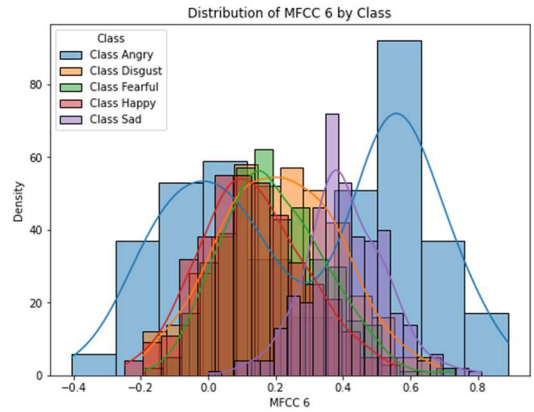

Fig 3.3.2.58: Distribution of MFCC 6

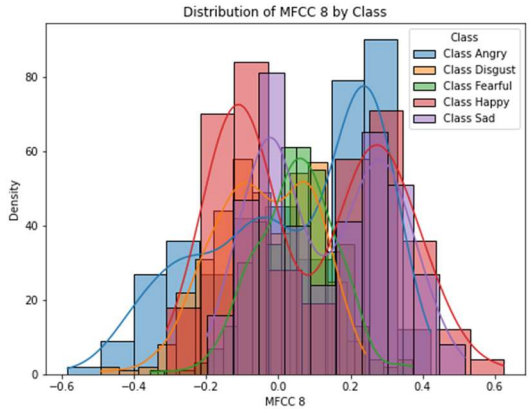

Fig 3.3.2.59: Distribution of MFCC 8

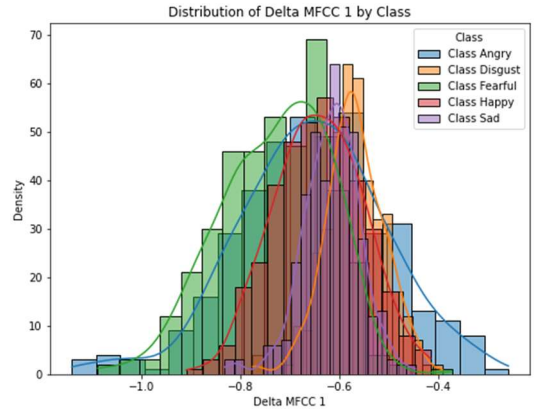

Fig 3.3.2.60: Distribution of Delta MFCC 1

Exploring Feature Distributions: SAVEE Dataset

Fig 3.3.2.61 – Fig 3.3.2.90 depict the Feature Density Distribution plots for the SAVEE Dataset, illustrating how the 30 selected features are distributed across five different emotion classes. These visualizations provide valuable insights into the density and variability of each feature within the distinct emotional categories, enhancing our understanding of the dataset's emotional nuances.

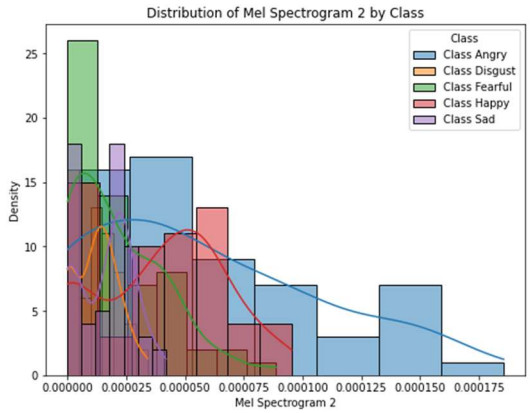

Fig 3.3.2.61: Distribution of Mel Spectogram 2

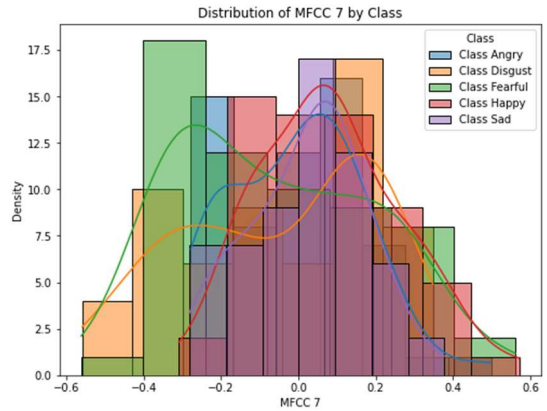

Fig 3.3.2.62: Distribution of MFCC 7

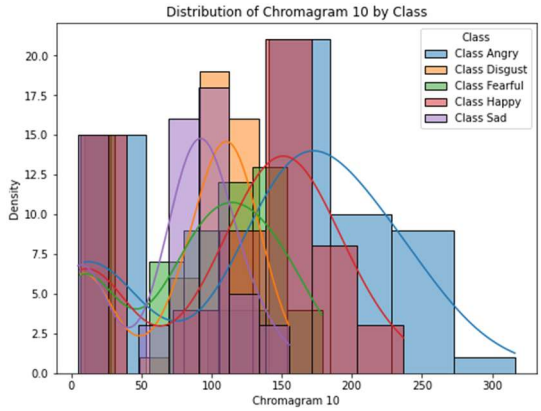

Fig 3.3.2.63: Distribution of Chromagram 10

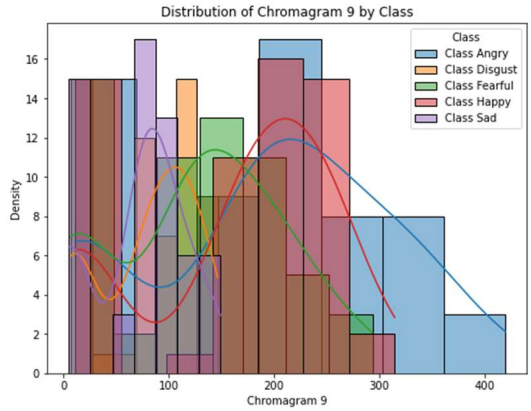

Fig 3.3.2.64: Distribution of Chromagram 9

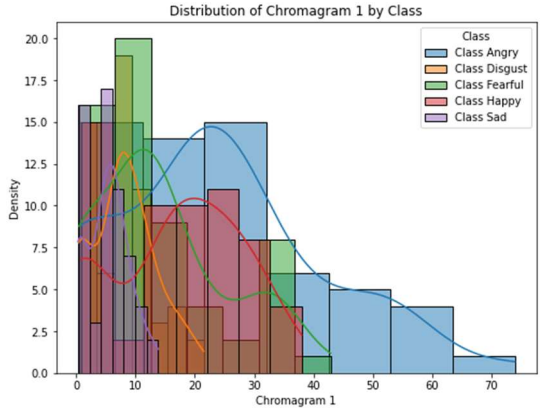

Fig 3.3.2.65: Distribution of Chromagram 1

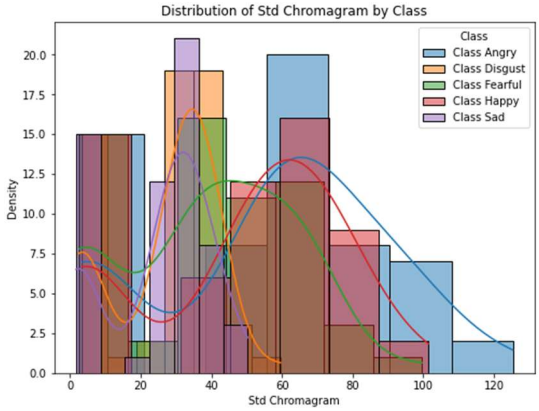

Fig 3.3.2.66: Distribution of Std Chromagram

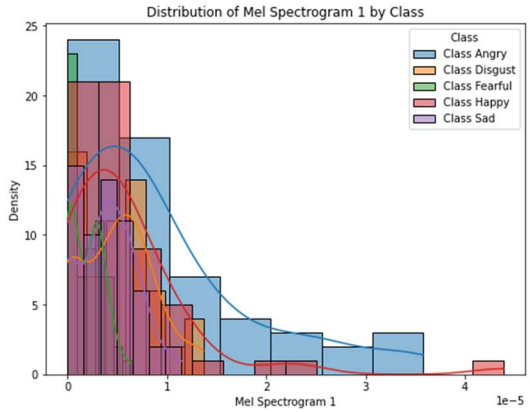

Fig 3.3.2.67: Distribution of Mel Spectogram 1

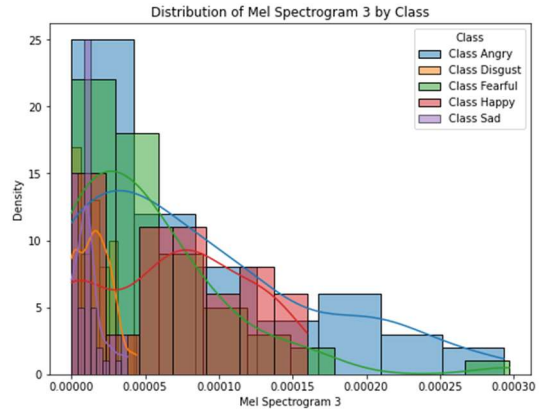

Fig3.3.2.68: Distribution of Mel Spectogram 3

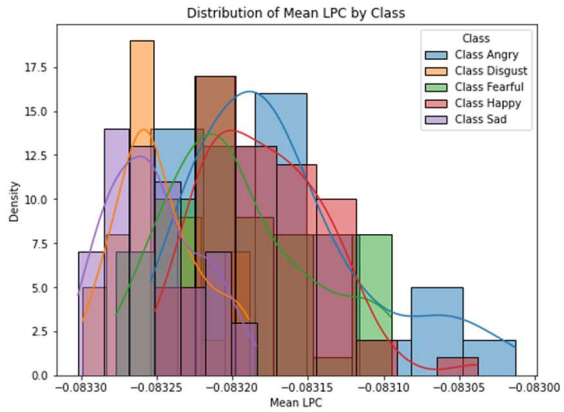

Fig 3.3.2.69: Distribution of Mean LPC

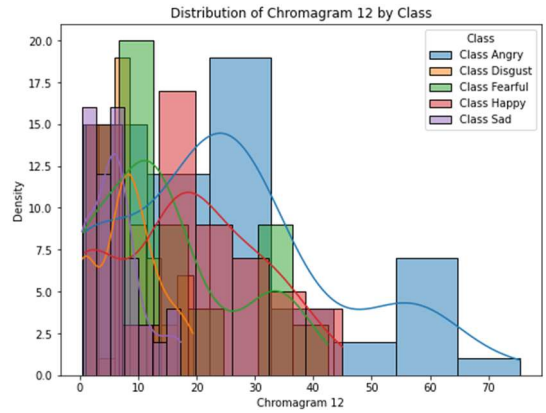

Fig 3.3.2.70: Distribution of Chromagram 12

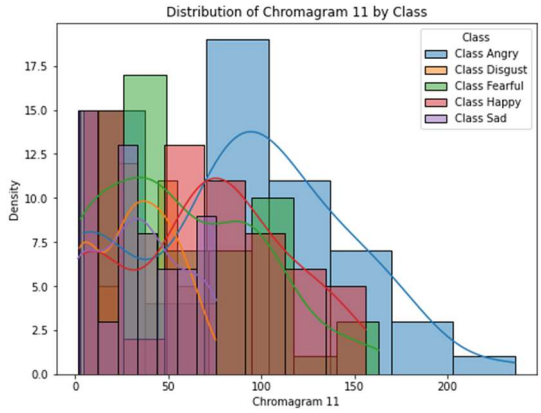

Fig 3.3.2.71: Distribution of Chromagram 11

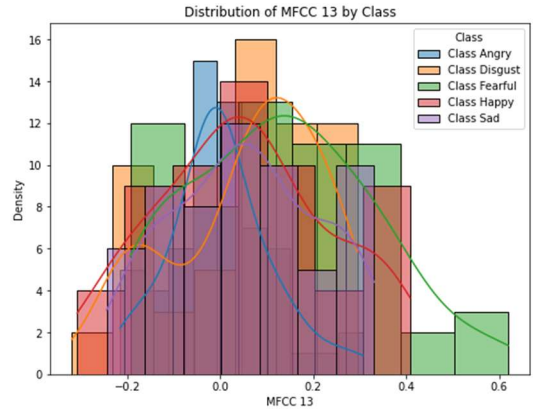

Fig 3.3.2.72: Distribution of MFCC 13

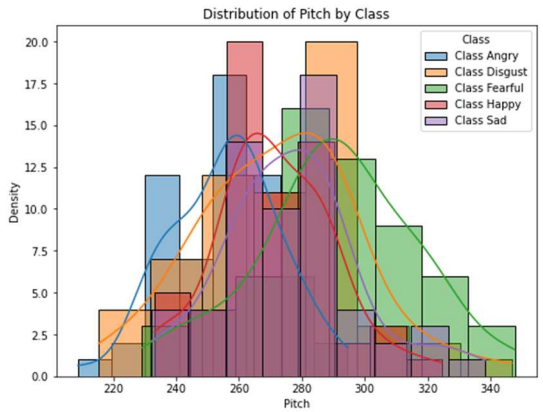

Fig 3.3.2.73: Distribution of Pitch

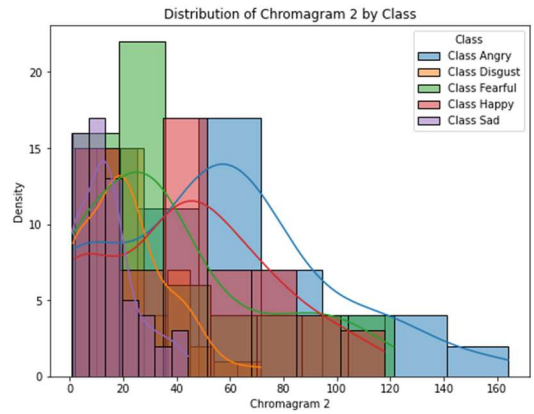

Fig 3.3.2.74: Distribution of Chromagram 2

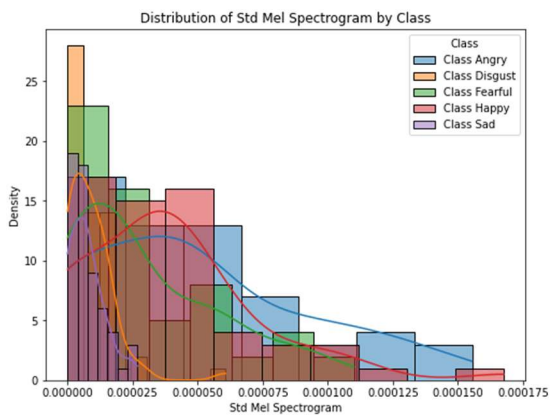

Fig 3.3.2.75: Distribution of Std Mel Spectogram

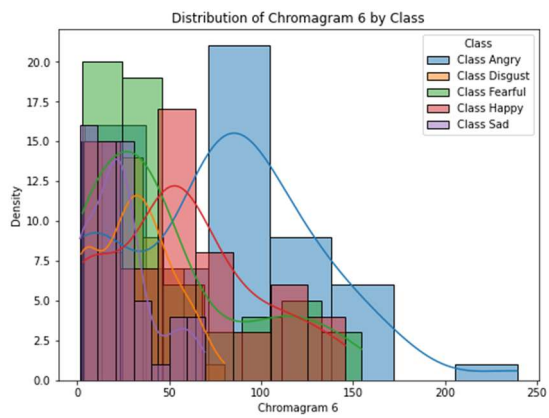

Fig 3.3.2.76: Distribution of Chromagram

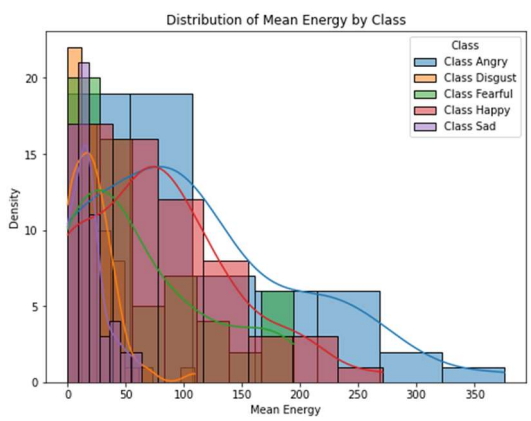

Fig 3.3.2.77: Distribution of Mean Energy

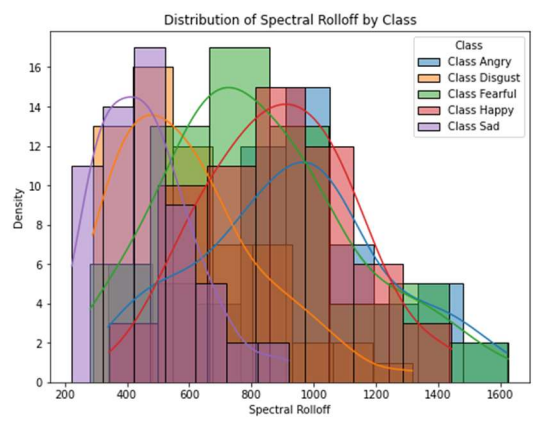

Fig 3.3.2.78: Distribution of Spectral Rolloff

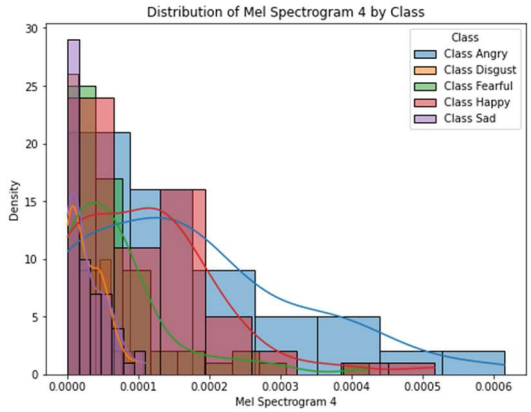

Fig 3.3.2.79: Distribution of Mel Spectogram 4

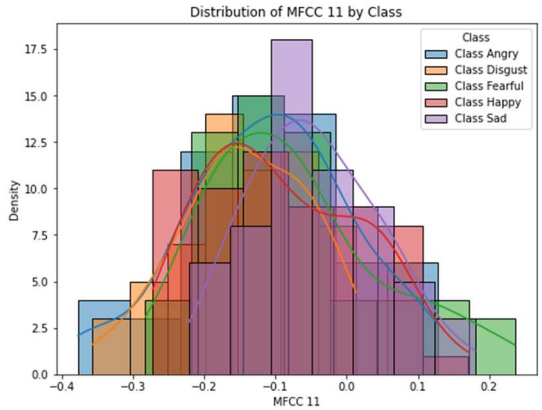

Fig 3.3.2.80: Distribution of MFCC 11

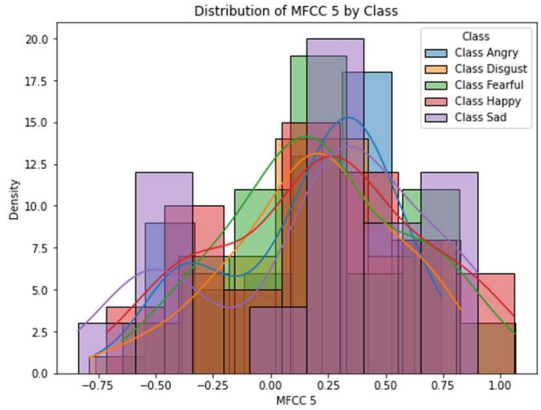

Fig 3.3.2.81: Distribution of MFCC 5

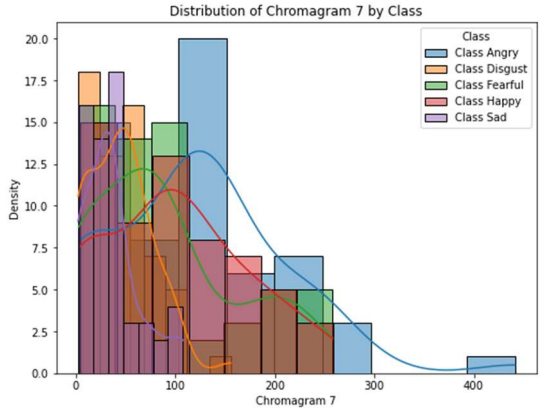

Fig 3.3.2.82: Distribution of Chromagram 7

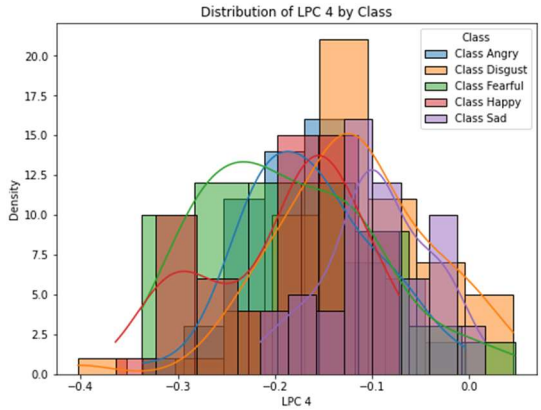

Fig 3.3.2.83: Distribution of LPC 4

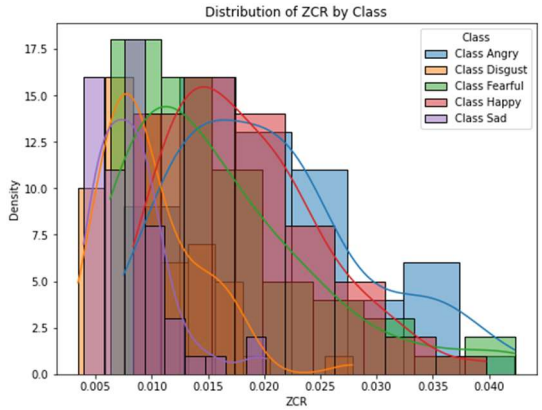

Fig 3.3.2.84: Distribution of ZCR

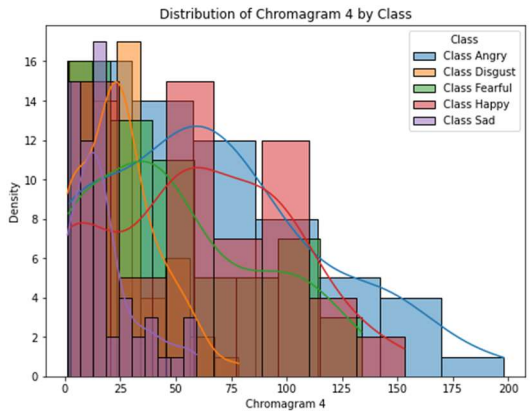

Fig 3.3.2.85: Distribution of Chromagram 4

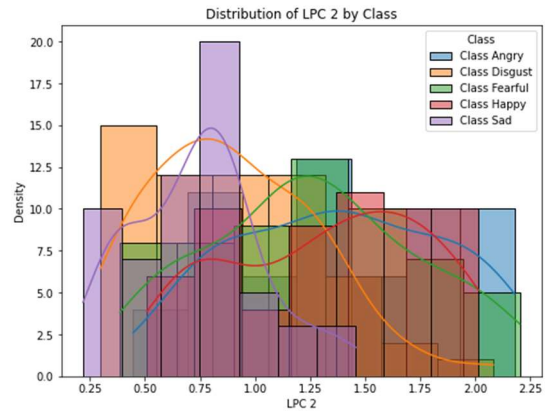

Fig 3.3.2.86: Distribution of LPC 2

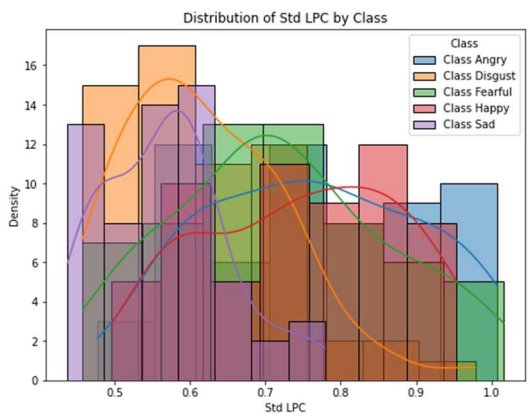

Fig 3.3.2.87: Distribution of Std LPC

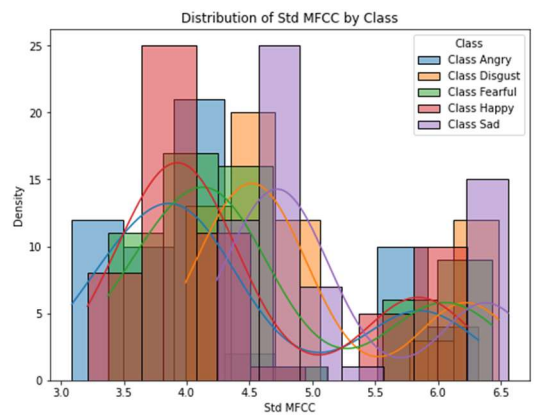

Fig 3.3.2.88: Distribution of Std MFCC

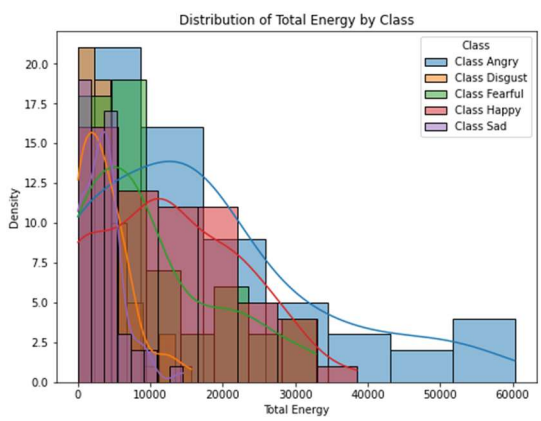

Fig 3.3.2.89: Distribution of Total Energy

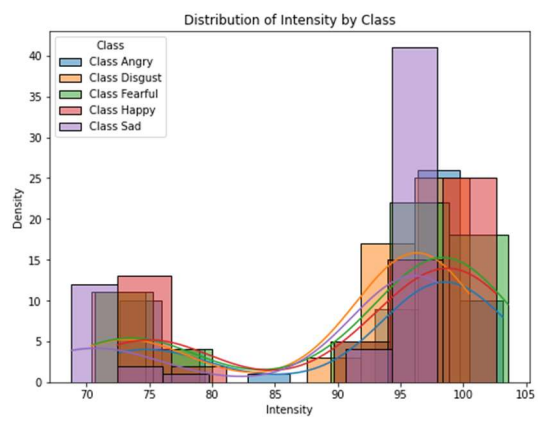

Fig 3.3.2.90: Distribution of Intensity

Exploring Feature Distributions: EmoDB Dataset

Fig 3.3.2.91 – Fig 3.3.2.120 depict the Feature Density Distribution plots for the EmoDB Dataset, illustrating how the 30 selected features are distributed across five different emotion classes. These visualizations provide valuable insights into the density and variability of each feature within the distinct emotional categories, enhancing our understanding of the dataset's emotional nuances.

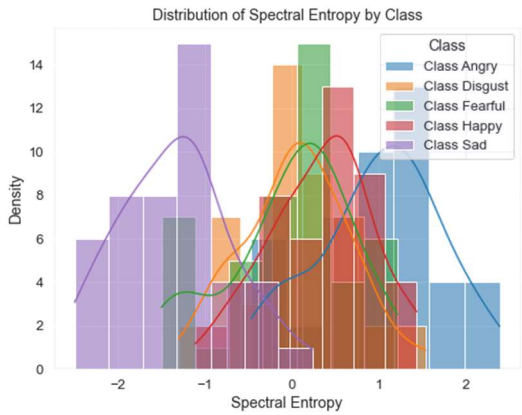

Fig 3.3.2.91: Distribution of Spectral Entropy

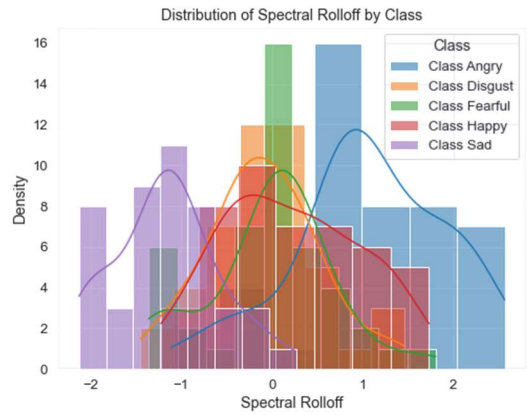

Fig 3.3.2.92: Distribution of Spectral Rolloff

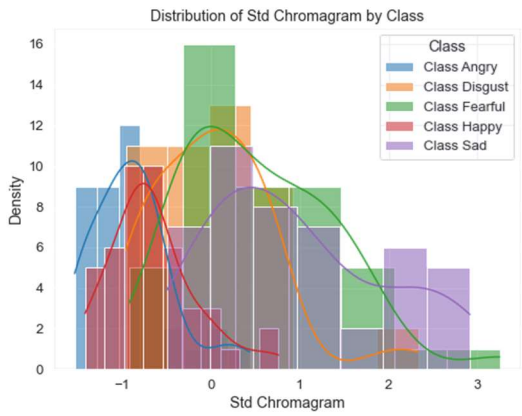

Fig 3.3.2.93: Distribution of Std Chromagram

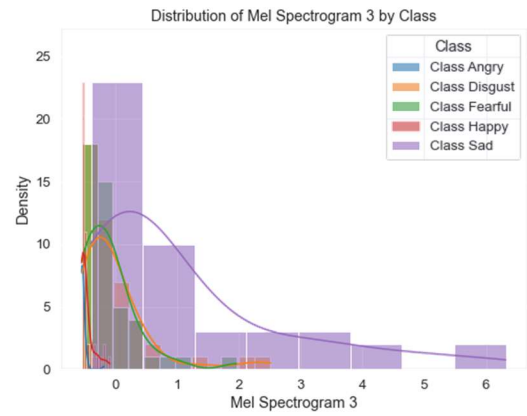

Fig 3.3.2.94: Distribution of Mel Spectogram 3

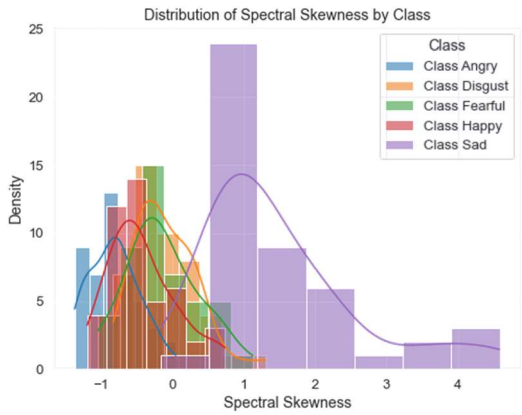

Fig 3.3.2.95: Distribution of Spectral Skewness

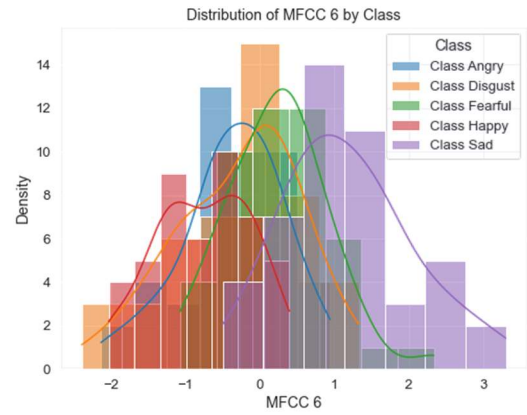

Fig 3.3.2.96: Distribution of MFCC 6

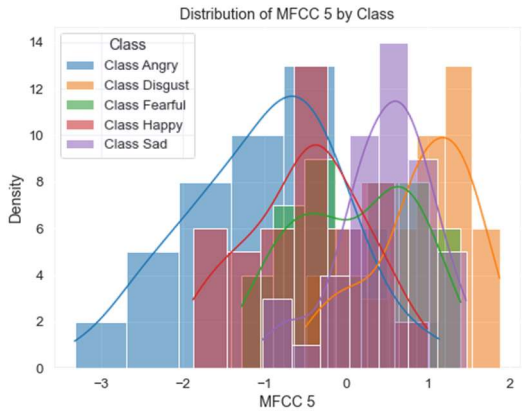

Fig 3.3.2.97: Distribution of MFCC 5

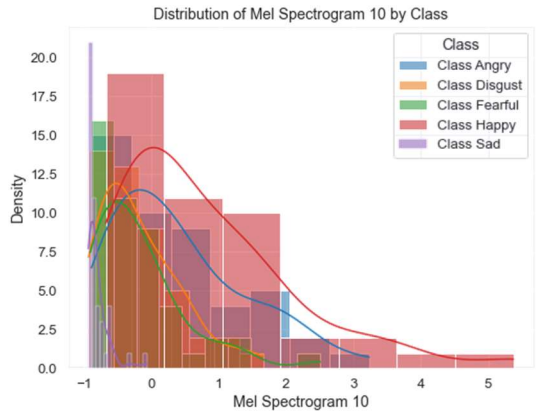

Fig 3.3.2.98: Distribution of Mel Spectogram 10

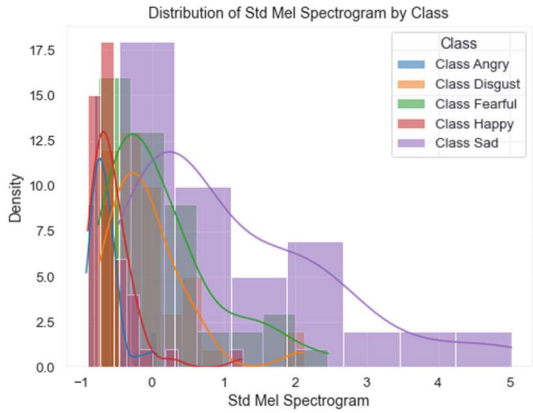

Fig 3.3.2.99: Distribution of Std Mel Spectogram

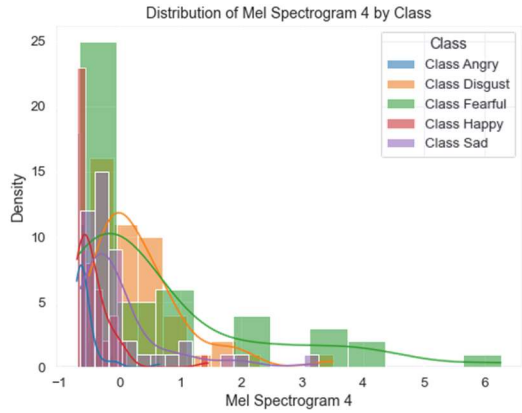

Fig 3.3.2.100: Distribution of Mel Spectogram 4

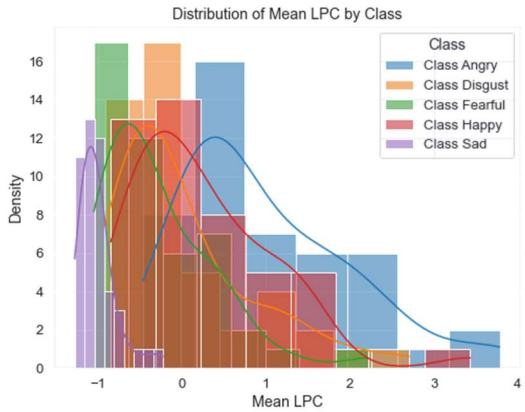

Fig 3.3.2.101: Distribution of Mean LPC

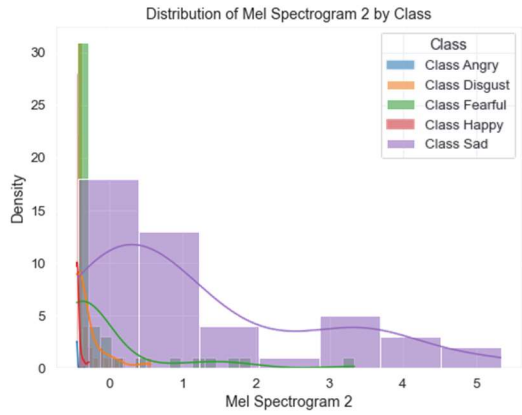

Fig 3.3.2.102: Distribution of Mel Spectogram 2

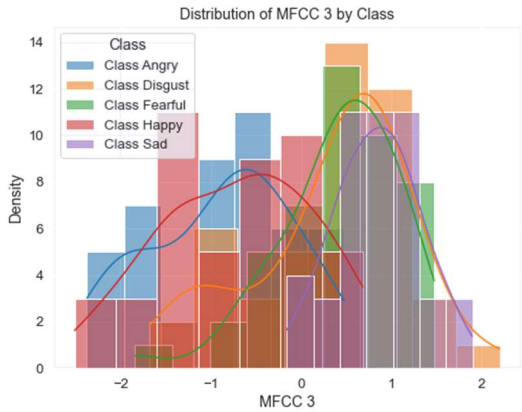

Fig 3.3.2.103: Distribution of MFCC 3

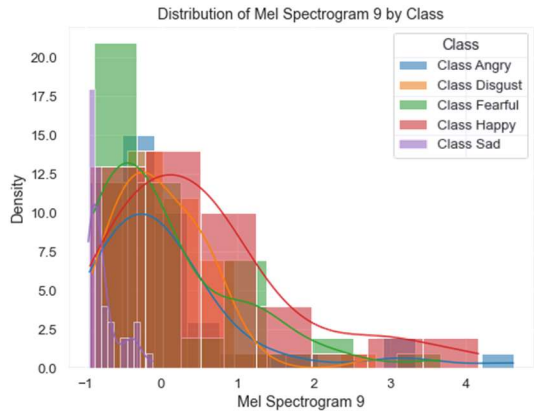

Fig 3.3.2.104: Distribution of Mel Spectogram 9

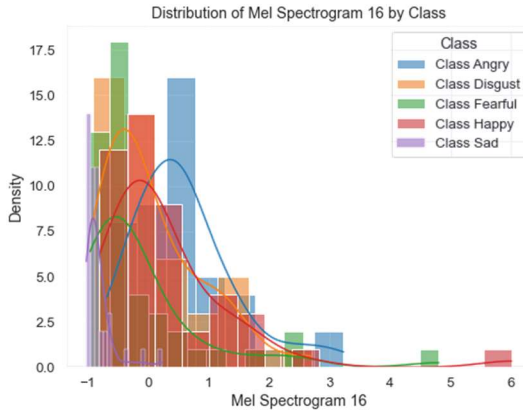

Fig 3.3.2.105: Distribution of Mel Spectogram 16

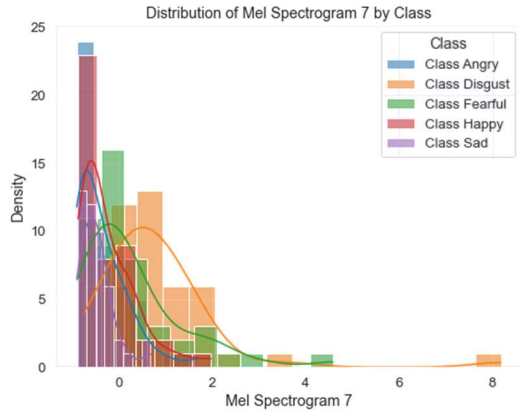

Fig 3.3.2.106: Distribution of Mel Spectogram 7

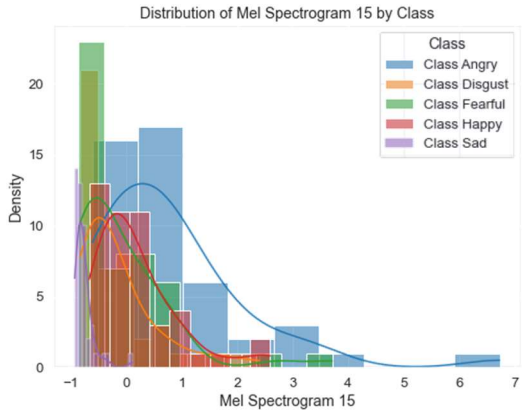

Fig 3.3.2.107: Distribution of Mel Spectogram 15

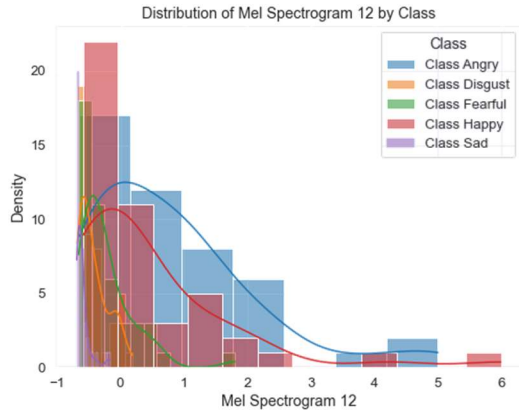

Fig 3.3.2.108: Distribution of Mel Spectogram 12

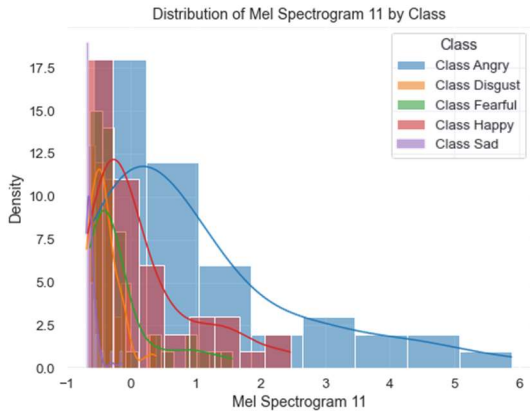

Fig 3.3.2.109: Distribution of Mel Spectogram 11

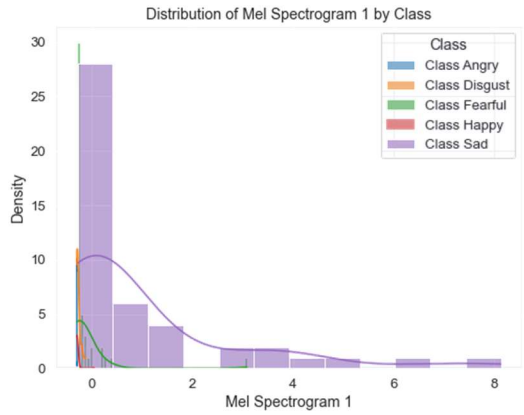

Fig 3.3.2.110: Distribution of Mel Spectogram 1

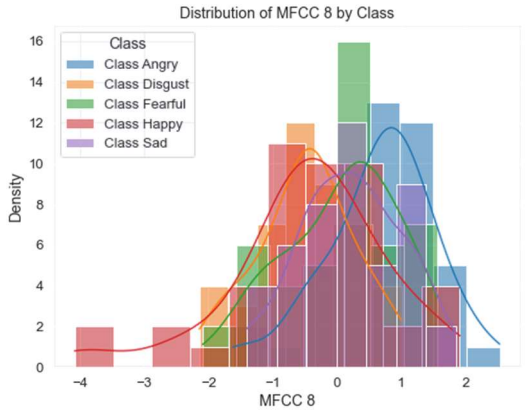

Fig 3.3.2.111: Distribution of MFCC 8

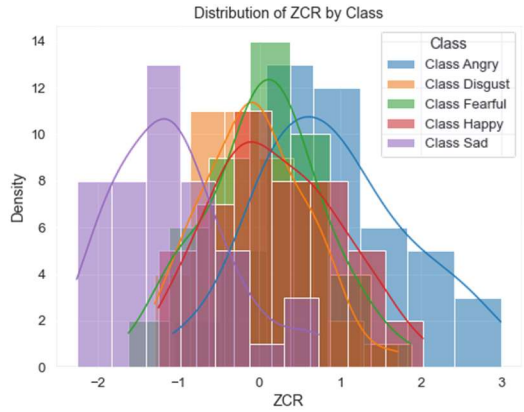

Fig 3.3.2.112: Distribution of ZCR

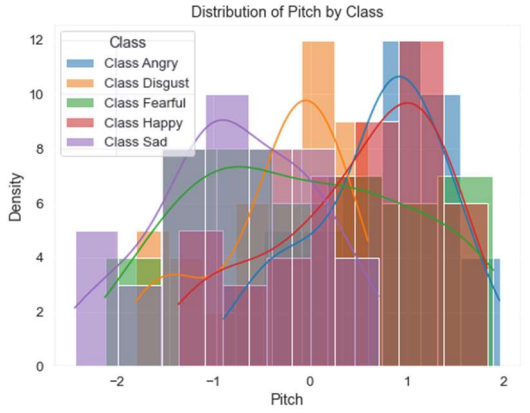

Fig 3.3.2.113: Distribution of Pitch

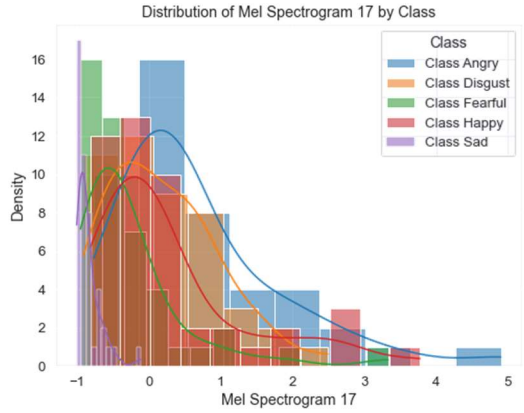

Fig 3.3.2.114: Distribution of Mel Spectogram 17

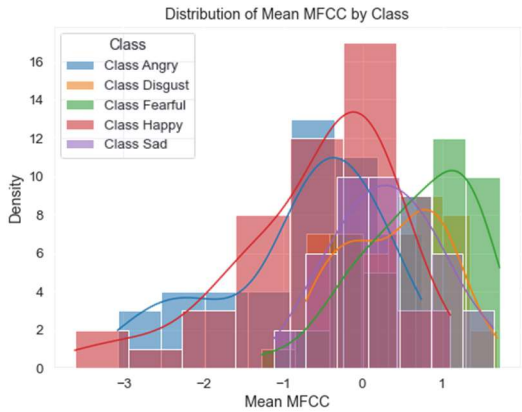

Fig 3.3.2.115: Distribution of Mean MFCC

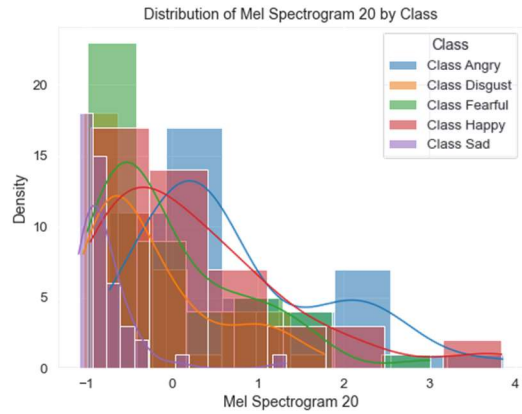

Fig 3.3.2.116: Distribution of Mel Spectogram 20

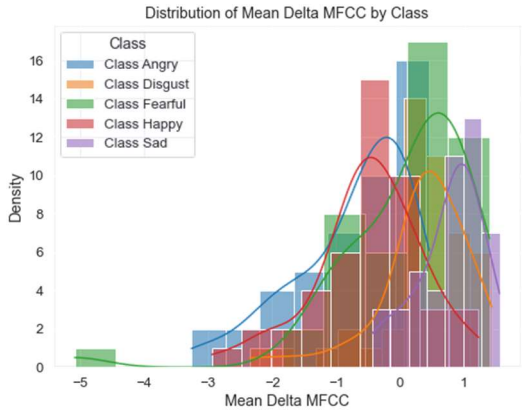

Fig 3.3.2.117: Distribution of Mean Delta Delta MFCC

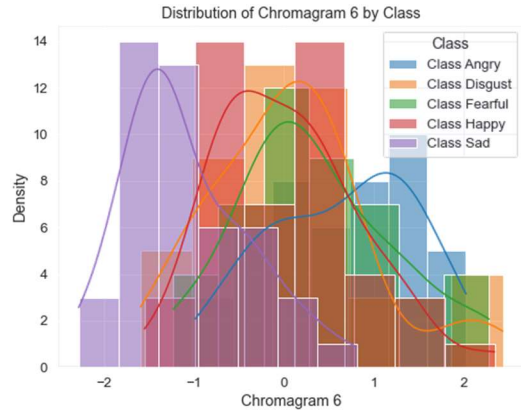

Fig 3.3.2.118: Distribution of Chromagram 6

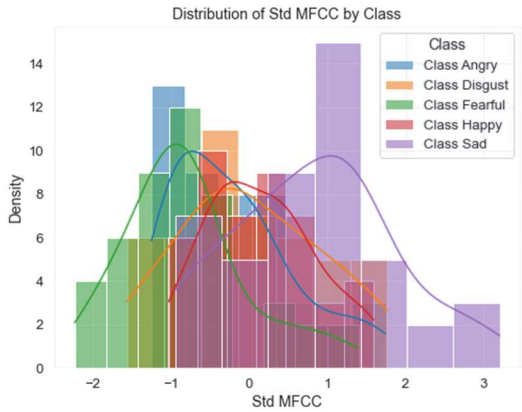

Fig 3.3.2.119: Distribution of Std MFCC

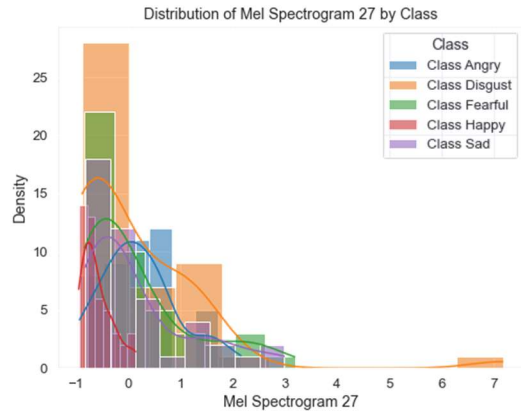

Fig 3.3.2.120: Distribution of Mel Spectogram 27

Exploring Feature Distributions: CREMA-D Dataset

Fig 3.3.2.121 – Fig 3.3.2.150 depict the Feature Density Distribution plots for the CREMA-D Dataset, illustrating how the 30 selected features are distributed across five different emotion classes. These visualizations provide valuable insights into the density and variability of each feature within the distinct emotional categories, enhancing our understanding of the dataset's emotional nuances.

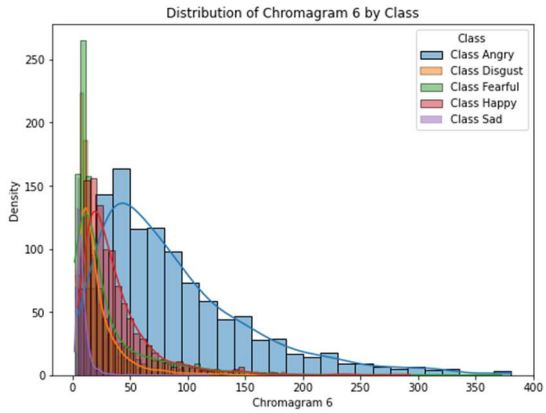

Fig 3.3.2.121: Distribution of Chromagram 6

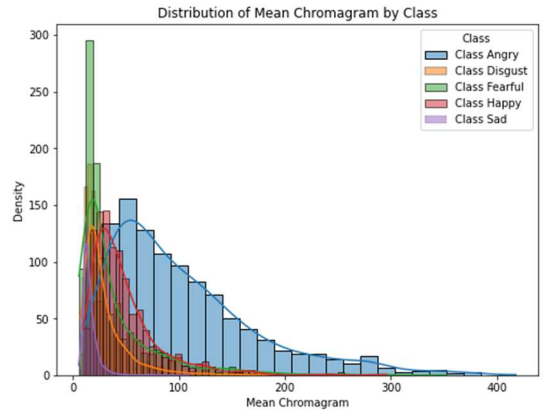

Fig 3.3.2.122: Distribution of Mean Chromagram

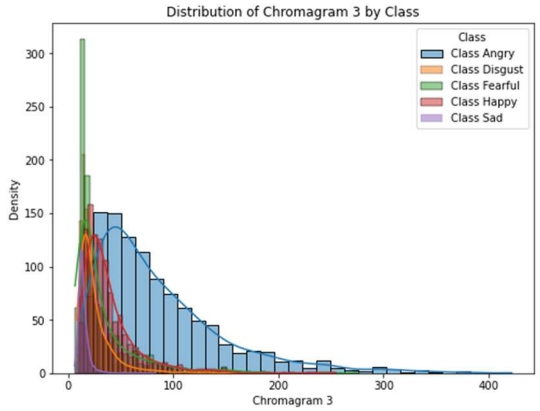

Fig 3.3.2.123: Distribution of Chromagram 3

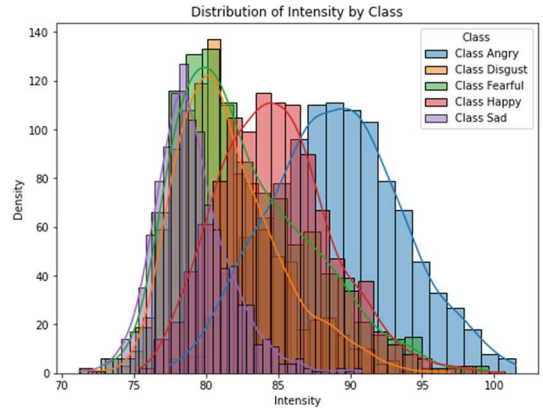

Fig 3.3.2.124: Distribution of Intensity

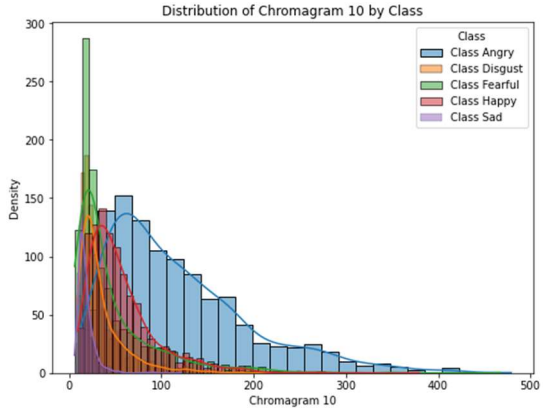

Fig 3.3.2.125: Distribution of Chromagram 10

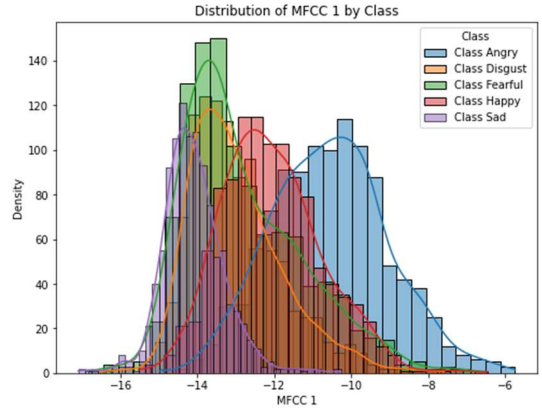

Fig 3.3.2.126: Distribution of MFCC 1

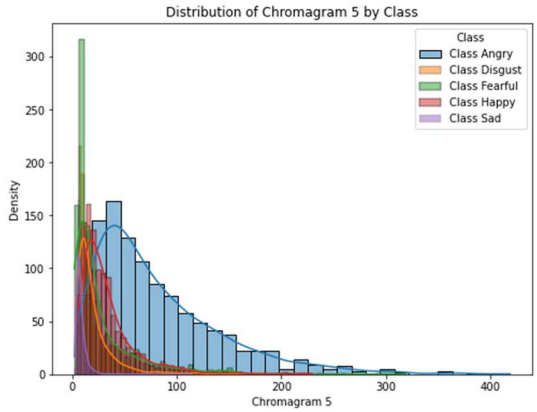

Fig 3.3.2.127: Distribution of Chromagram 5

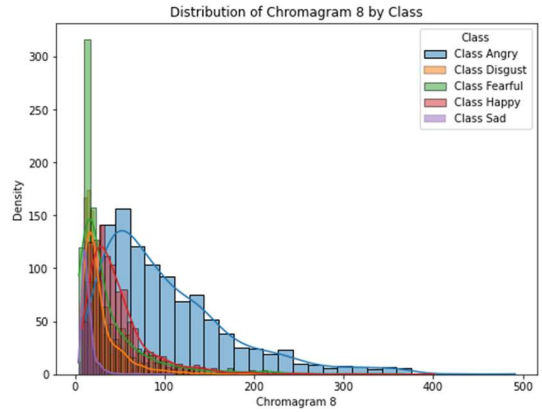

Fig 3.3.2.128: Distribution of Chromagram 8

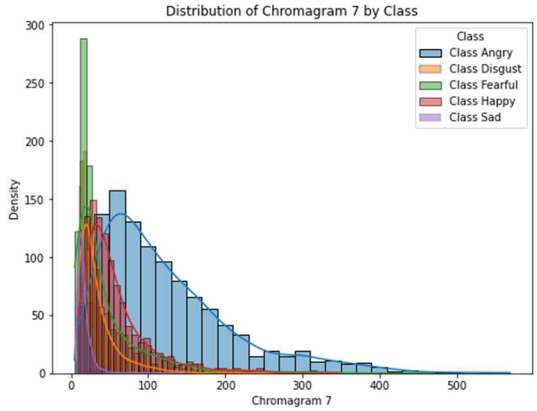

Fig 3.3.2.129: Distribution of Chromagram 7

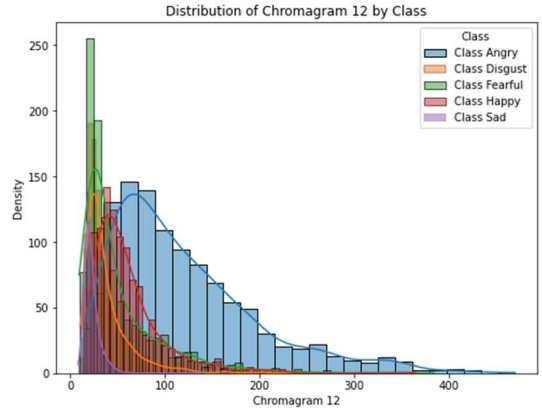

Fig 3.3.2.130: Distribution of Chromagram 12

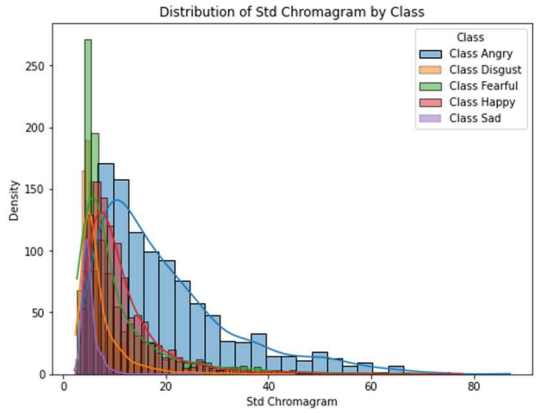

Fig 3.3.2.131: Distribution of Std Chromagram

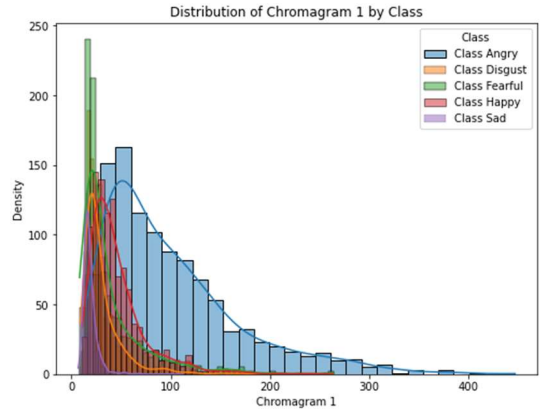

Fig 3.3.2.132: Distribution of Chromagram 1

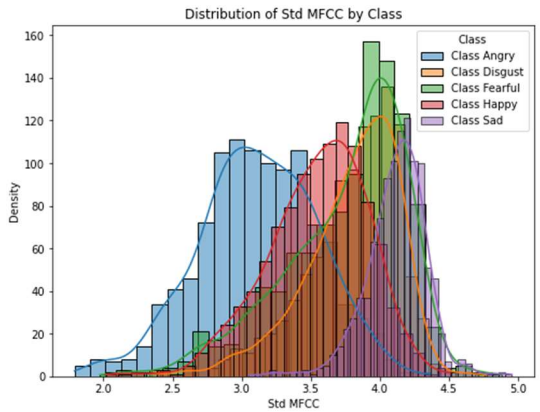

Fig 3.3.2.133: Distribution of Std MFCC

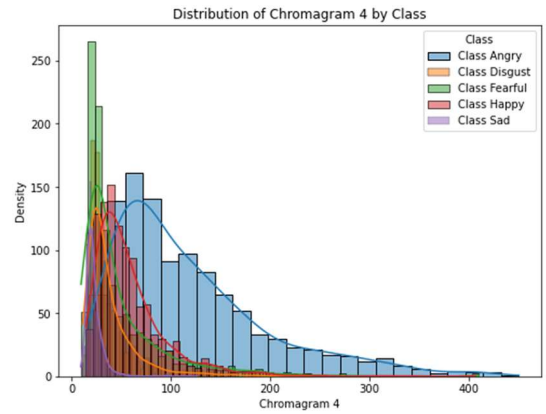

Fig 3.3.2.134: Distribution of Chromagram 4

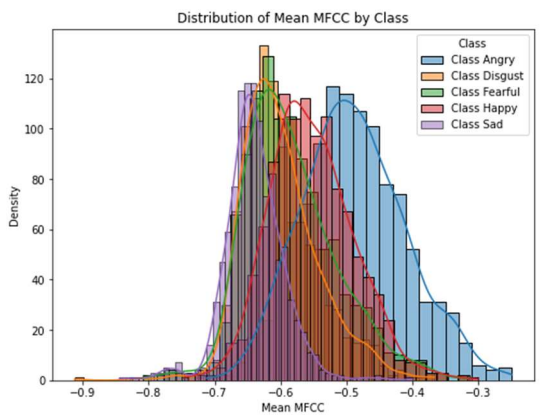

Fig 3.3.2.135: Distribution of Mean MFCC

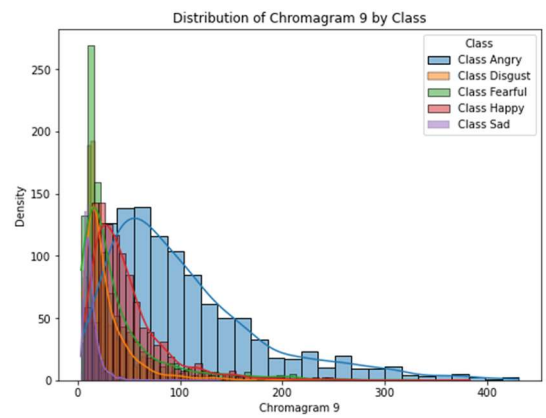

Fig 3.3.2.136: Distribution of Chromagram 9

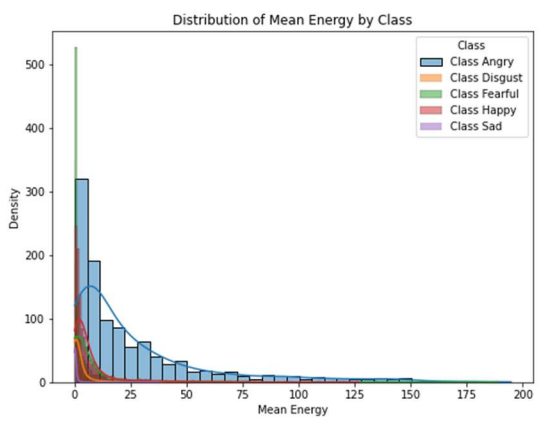

Fig 3.3.2.137: Distribution of Mean Energy

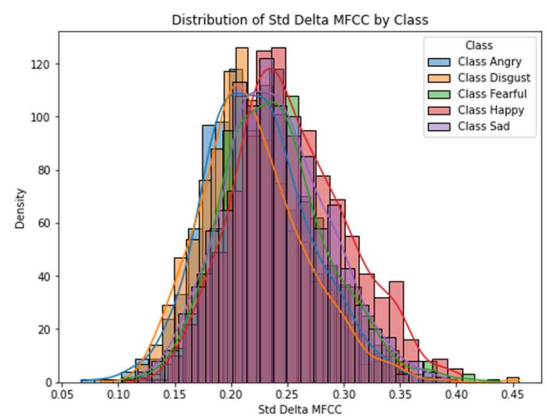

Fig 3.3.2.138: Distribution of Std Delta MFCC

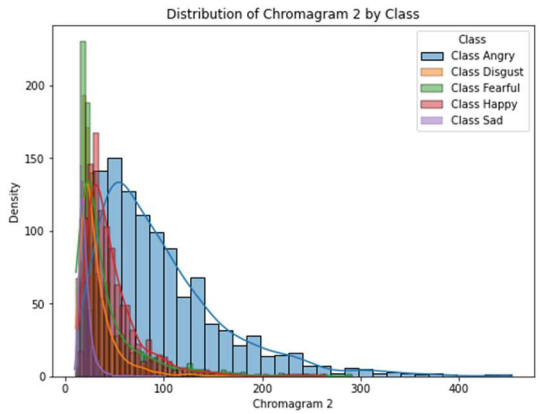

Fig 3.3.2.139: Distribution of Chromagram 2

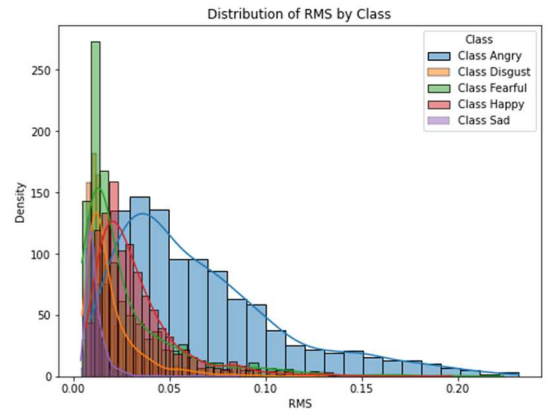

Fig 3.3.2.140: Distribution of RMS

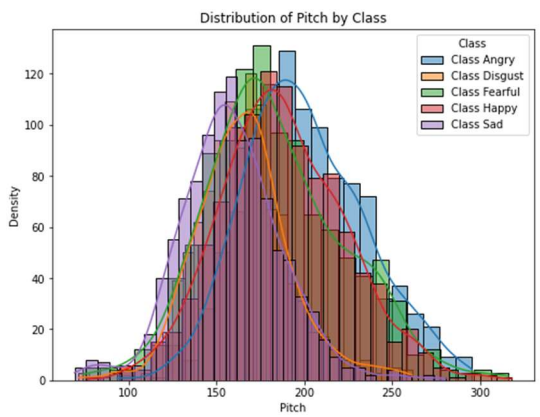

Fig 3.3.2.141: Distribution of Pitch

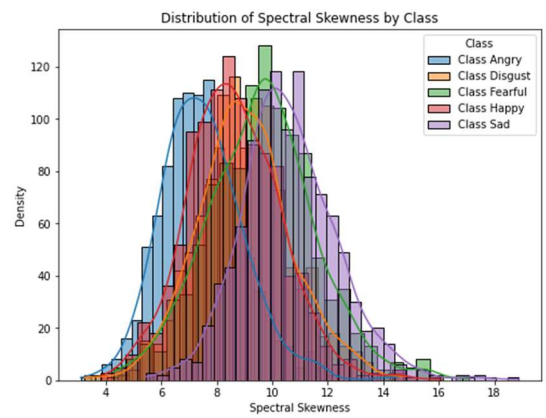

Fig 3.3.2.142: Distribution of Spectral Skewness

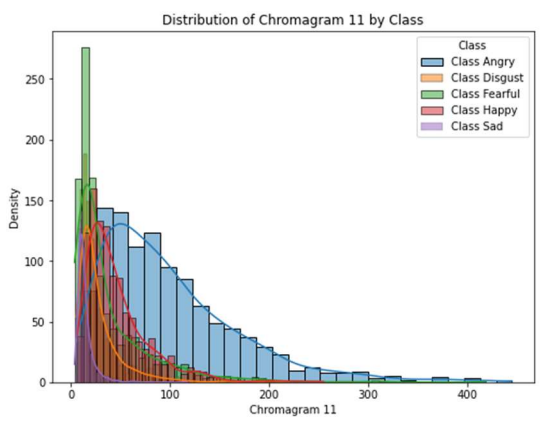

Fig 3.3.2.143: Distribution of Chromagram 11

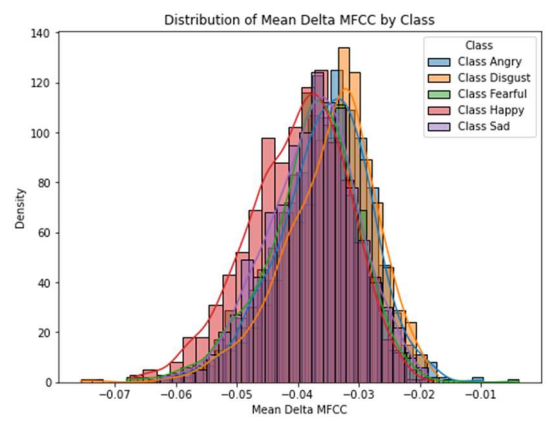

Fig 3.3.2.144: Distribution of Mean Delta MFCC

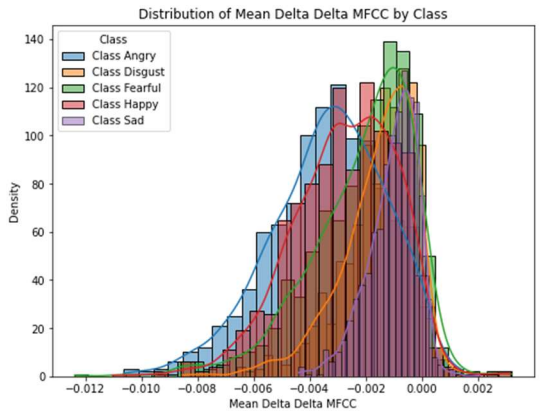

Fig 3.3.2.145: Distribution of Mean Delta Delta MFCC

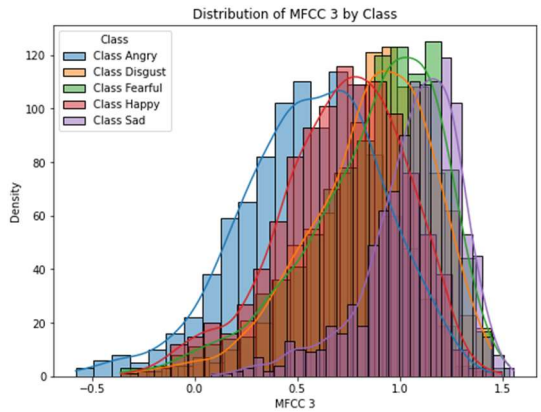

Fig 3.3.2.146: Distribution of MFCC 3

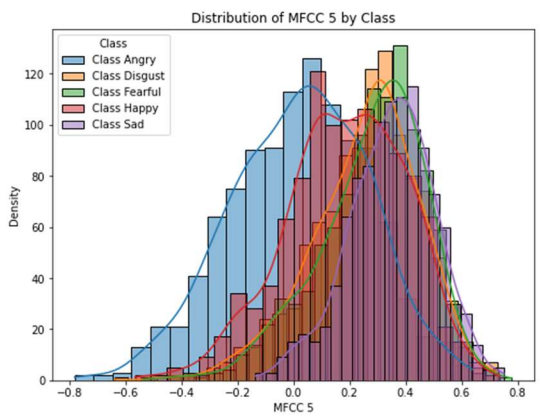

Fig 3.3.2.147: Distribution of MFCC 5

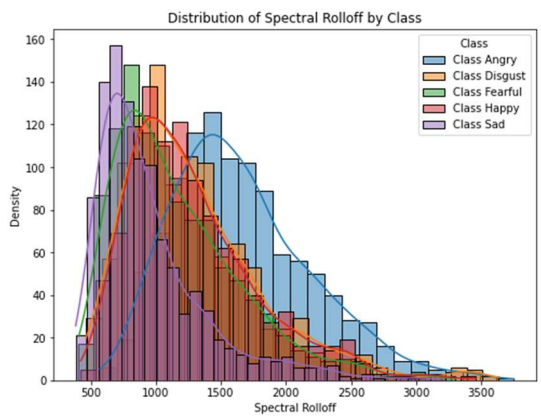

Fig 3.3.2.148: Distribution of Spectral Rolloff

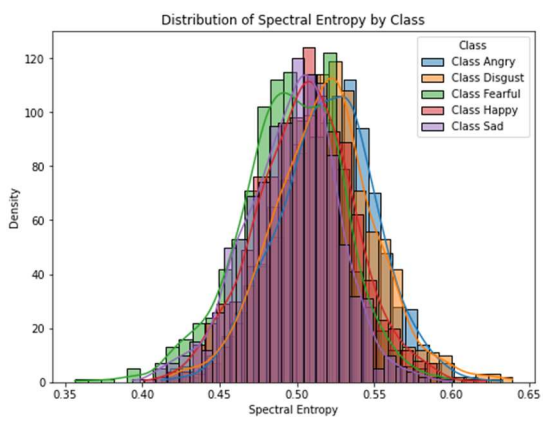

Fig 3.3.2.149: Distribution of Spectral Entropy

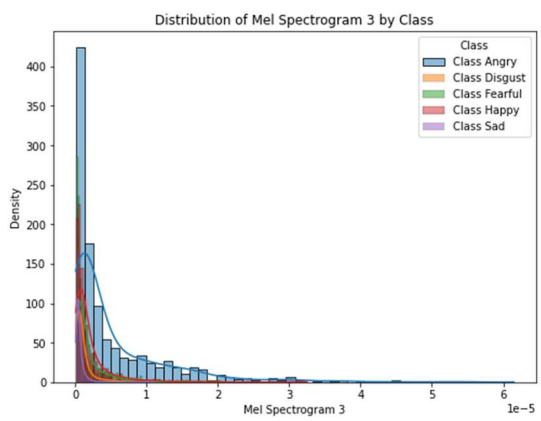

Fig 3.3.2.150: Distribution of Mel Spectogram 3

3.4Classification

3.4.1 Method – 1

In our classification phase, we harnessed the robust capabilities of the Support Vector Machines (SVMs) classifier from the Scikit-Learn library, renowned for its prowess in tackling intricate classification tasks. Armed with a feature vector composed of 30 meticulously chosen columns, extracted from diverse datasets spanning RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D, our model attained a comprehensive understanding of vocal expressions, encompassing an extensive spectrum of emotions and vocal nuances.

A cornerstone of our approach lay in the meticulous evaluation of our model's performance on unseen data, ensuring its adaptability and reliability in real-world scenarios. Through an iterative process of experimentation and parameter fine-tuning, we meticulously optimized our SVM classifier to achieve peak performance, aiming to maximize accuracy while minimizing misclassifications across the five targeted emotion classes: Angry, Disgust, Fearful, Happy, and Sad. This exhaustive refinement process not only bolstered the classifier's predictive prowess but also enhanced its resilience and versatility across diverse recording conditions and emotional states.

Furthermore, our strategy of individually training the model on each dataset facilitated a deeper understanding of the unique characteristics inherent in vocal expressions from various sources. By tailoring the classifier to accommodate these nuances, we ensured its adaptability and efficacy in accurately categorizing emotions across a wide range of vocal contexts.

Moreover, we conducted extensive experimentation with alternative models such as RandomForest, Naive Bayes, and Logistic Regression. While these models exhibited promising performance, our rigorous evaluation process ultimately led us to select the SVM classifier for its superior performance in handling the complexity of our classification task. In essence, our classification framework represents a meticulous fusion of advanced machine learning techniques and domain-specific expertise, poised to deliver precise and actionable insights into the rich tapestry of human emotions encapsulated within vocal expressions.

3.4.2 Method – 2

The classification phase involved using the KERAS library of Python 3.8 and WEKA software. We have done the classification by training our data using built in MLP, logistic regression, and random forest functions of KERAS Library. The dataset consisted of 1440 data files, considering both male and female voices. The feature vector having 57 columns as Feature header was used as input for the classification techniques. Supervised learning was employed, and hyperparameter tuning were used to improve the results. Each classifier had specific parameters that were optimized through experimentation and tweaking.

Classification using a Random Forest classifier on a dataset containing features and labels. It splits the data into training and testing sets, standardizes the features, trains the classifier with adjusted hyperparameters, evaluates its performance on both training and testing sets, and prints the accuracies achieved. Performs binary or multiclass classification using logistic regression on a dataset with features and labels. It splits the data into training and testing sets, standardizes the features, trains the logistic regression classifier, evaluates its performance on both training and testing sets, and prints the accuracies achieved.

A Multilayer Perceptron (MLP) classifier, a type of artificial neural network, for classification on a dataset containing features and labels. It splits the data into training and testing sets, standardizes the features, defines the MLP classifier with specified hyperparameters, trains the classifier, evaluates its performance on both training and testing sets, and prints the achieved accuracies. A stacking ensemble, combining predictions from multiple base classifiers (Random Forest and MLP) using a meta-classifier (another MLP). It trains the stacking ensemble on the training data, evaluates its performance on both training and testing sets, and prints the achieved accuracies.

In Speech Emotion Recognition project using the Ravdess Dataset, Stack Ensemble Classifier has been used. Here's how it works:

1. Base Models (Random Forest and Logistic Regression): These are my initial classifiers. They independently learn from the input features, likely acoustic features from speech recordings, and predict the emotions present in the samples.

2. Meta Regressor (MLP): This is my top-level model, also known as a meta learner. Instead of directly predicting emotions, it learns from the predictions of the base models. It takes the output of the base models (the probabilities or scores assigned to each emotion class) as input features and then learns to combine or weigh these predictions to make the final decision.

3. Stacking: In stacking, the predictions of the base models are used as features for training the meta regressor. This means each base model's predictions serve as input features for the meta regressor. The meta regressor then learns how to best combine or interpret these predictions to make the final prediction.

4. Training and Prediction: During training, the base models are trained on the training data, and their predictions are stored. Then, the meta regressor is trained on these predictions along with the actual labels. During prediction, the base models make predictions on new data, and these predictions are used as input for the meta regressor, which then makes the final prediction.

5. Improved Performance: By combining the strengths of multiple base models, the stack ensemble classifier often performs better than any individual base model alone. This is because it can capture different aspects of the data and learn from the strengths of each base model.

Overall, the stack ensemble classifier enhances the performance of my Speech Emotion Recognition system by leveraging the diversity of multiple base models and learning to effectively combine their predictions using the meta regressor.

3.4.3 Confusion Matrix

A Confusion Matrix serves as a fundamental tool in assessing the performance of classification models within our project. It provides a detailed breakdown of the model's predictions, categorizing them into correct and incorrect classifications across different classes. By offering this granular insight, the Confusion Matrix aids in comprehensively evaluating the accuracy and effectiveness of our classification approach.

In our project, the Confusion Matrix plays a crucial role in error analysis. By examining the matrix, we can identify patterns of misclassifications and discern which classes are more prone to errors. This knowledge enables us to pinpoint the weaknesses of our model and devise targeted strategies for improvement. Whether through adjusting features, fine-tuning parameters, or exploring alternative algorithms, the insights gleaned from the Confusion Matrix guide our efforts towards optimizing model performance.

3.4.4 Classification Report

The Classification Report is a crucial evaluation tool in machine learning, providing metrics like precision, recall, F1-score, and support for each class within the dataset. It assesses the model's accuracy, its ability to avoid false positives, and identify all relevant instances of a class. This comprehensive report aids in understanding the model's performance across different classes, guiding improvements and optimizations.

3.5GANTT Chart

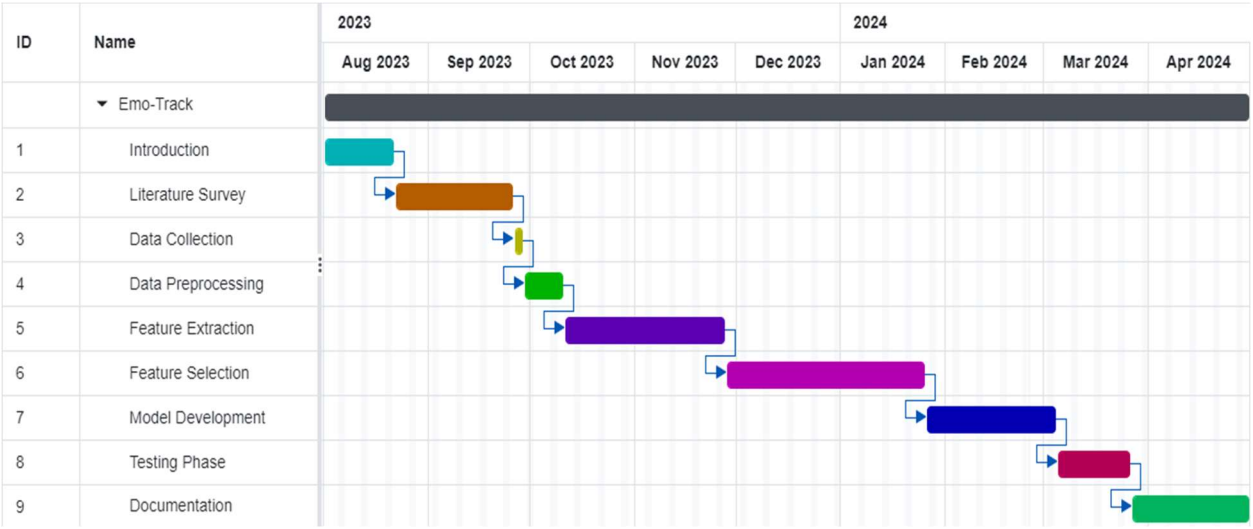

Fig.3.5: GANTT Chart of the project

Fig 3.5 explains the GANTT chart for our project.

1. **Introduction**

   During the Introduction phase, the fundamental project requirements were identified and outlined. This pivotal phase was successfully concluded by mid-August 2023, as per the project's GANTT chart schedule.

2. **Literature Survey**

   During the Literature Survey phase, a thorough examination of scholarly literature was conducted, encompassing 10 papers from international journals and conference proceedings. This comprehensive review provided valuable insights and foundational knowledge essential for informing the project's direction and methodology.

3. **Data Collection**

   In the Data Collection phase, a comprehensive set of audio datasets was gathered from Kaggle, comprising the RAVDESS, TESS, SAVEE, EmoDB, and CREMA-D databases.

4. **Data Preprocessing**

   During the Data Preprocessing phase, thorough checks were conducted on the audio datasets to ensure uniformity in file formats. Any files found to be in stereo format were converted to mono format to maintain consistency. Additionally, in this step, a strategic decision was made to focus on five specific emotions: anger, disgust, fear, happiness, and sadness, streamlining the dataset for further analysis and modeling.

5. **Feature Extraction**

   During the Feature Extraction phase, MATLAB was employed to extract pertinent features from the raw audio files. These extracted features were then meticulously stored in a structured CSV file format, facilitating seamless integration into subsequent analysis and modeling processes.

6. **Feature Selection**

   In Feature Selection, the Extra Tree Classifier algorithm was utilized to identify and retain the most relevant features, enhancing the system's efficiency and effectiveness.

7. **Model Development**

   The Model Development phase focused on implementing the code and training the machine learning models using the selected features and the dataset.

8. **Testing Phase**

   In the Testing Phase, different classifiers, including Support Vector Machine, Random Forest, Logistic Regression, and Multi-Layer Perceptron, were used to evaluate the accuracy. Additionally, hyperparameter tuning techniques were applied to optimize the model's performance and increase its accuracy. This comprehensive approach ensured that the model was fine-tuned to achieve the highest levels of precision and effectiveness in emotion recognition.

9. **Documentation**

   During the Documentation phase, concise project reports were created, summarizing the design and methodology used.

Chapter 4: Sample Codes

```
WindowSize = 2048;
Overlap = 1024;


% Computing Mel-Frequency Cepstral Coefficients (MFCCs)
win = hann(WindowSize, 'periodic');
S = stft(audioIn, 'Window', win, 'OverlapLength', Overlap, 'Centered', false);
[coeffs,delta,deltaDelta] = mfcc(S, fs, 'LogEnergy', 'Ignore');


% Computing Mel Spectrogram
S = melSpectrogram(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength', Overlap).';


% Computing Spectral Flux
specflux = spectralFlux(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength', Overlap);


% Computing Spectral Skewness
specskewness = spectralSkewness(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength',
Overlap);


% Computing Spectral Slope
specslope = spectralSlope(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength', Overlap);
```

```
% Computing Spectral Entropy
specentropy = spectralEntropy(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength', Overlap);


% Computing Spectral Rolloff
specrolloff = spectralRolloffPoint(audioIn, fs, 'Window', hann(WindowSize, 'periodic'), 'OverlapLength', Overlap);
% Computing Chromagram
[S, F, T] = stft(audioIn, fs, 'Window', hamming(WindowSize), 'OverlapLength', Overlap, 'Centered', false);
chroma_mapping = mod(round(12 * log2(F / 440) * 12), 12) + 1;
chroma_features = zeros(size(S,2), 12);
for j = 1:12
    bin_indices = chroma_mapping == j;
    chroma_features(:, j) = sum(abs(S(bin_indices, :)), 1);
end


% Computing Linear Predictive Coding (LPC)
order = 12;
[a, ~] = lpc(windowedAudio, order);


% Computing Zero Crossing Rate (ZCR)
zcr_batch = zerocrossrate(audioIn, 'WindowLength', WindowSize, 'OverlapLength', Overlap);


% Computing Energy
frame_energy(frame) = sum(frame_data.^2);
```

```
% Computing Pitch
f0 = pitch(audioIn, fs, 'WindowLength', WindowSize, 'OverlapLength', Overlap);


% Computing Intensity
I0 = 1e-12;
intensity_value = mean(frame_data.^2);
intensity_dB = 10 * log10(intensity_value / I0);


% Computing Harmonic to Noise Ratio (HNR)
hnr = zeros(noOfFrames, 1);
xFrame = audioIn(startIdx:endIdx);
[r, lag] = xcorr(xFrame);
r = r(lag >= 0);
r = r / max(r);
[pks, locs] = findpeaks(r);
if isempty(locs)
    hnr(j) = 0;
else
    hnr(j) = 10 * log10(sum(pks(2:end).^2) / pks(1)^2);
end


% Computing Root Mean Square (RMS)
rms(j:j+WindowSize-1) = sqrt(sum(window.^2) / WindowSize);
```

```
% Save all the features to the matrix
all_statistics(i, :) = [coeffs, mean_coeffs, std_coeffs, delta, mean_delta, std_delta, deltaDelta, …
                        mean_deltaDelta, std_deltaDelta, melspec, mean_melspec, std_melspec, …
                        specflux, specSkewness, slope, entropy, rolloff, chromagram, …
                        mean_chromagram, std_chromagram, lpc_coeff, mean_lpc, …
                        std_lpc, zcr, total_energy, mean_energy, std_energy, f0, intensity, …
                        mean_hnr, rms];


% Saving the matrix to a CSV file
csvwrite('Features.csv', all_statistics);
```

This MATLAB code snippet demonstrates the process of feature extraction from audio signals. Let's break down each step:

Setting Parameters

- The code begins by defining parameters such as `WindowSize` and `Overlap`, which determine the size of the analysis window and the amount of overlap between consecutive windows.

Feature Extraction

- **Computing MFCCs:** The Mel-Frequency Cepstral Coefficients (MFCCs) are computed using the `mfcc` function. The audio signal is first transformed into the frequency domain using Short-Time Fourier Transform (STFT), and then MFCCs, along with their delta and delta-delta coefficients, are calculated.
- **Computing Mel Spectrogram:** The Mel Spectrogram is generated using the `melSpectrogram` function, providing a visual representation of the frequency content of the audio signal over time.
- **Computing Spectral Flux, Skewness, Slope, and Entropy:** Various spectral features such as spectral flux, skewness, slope, and entropy are computed using dedicated MATLAB functions. These features offer insights into the dynamics and spectral properties of the audio signal.
- **Computing Spectral Rolloff:** The spectral rolloff point is calculated using the `spectralRolloffPoint` function, indicating the frequency below which a certain percentage of the total spectral energy lies.

- **Computing Chromagram:** Chromagrams, which represent the energy distribution across different pitch classes, are computed using a series of operations involving STFT and frequency mapping.
- **Computing Linear Predictive Coding (LPC):** LPC coefficients, which capture the spectral envelope of the audio signal, are calculated using the `lpc` function.
- **Computing Zero Crossing Rate (ZCR):** The ZCR, indicating the rate of signal crossings through zero, is calculated using the `zerocrossrate` function.
- **Computing Energy:** The energy of each frame in the audio signal is computed by summing the squared values of the signal samples.
- **Computing Pitch:** Pitch estimation is performed using the `pitch` function, which estimates the fundamental frequency (pitch) of the audio signal.
- **Computing Intensity:** The intensity of the audio signal is calculated using a predefined reference intensity level.
- **Computing Harmonic-to-Noise Ratio (HNR):** HNR, indicating the ratio of harmonic components to noise in the signal, is computed based on the autocorrelation function of each frame.
- **Computing Root Mean Square (RMS):** The RMS value of each frame in the audio signal is computed as the square root of the mean of the squared signal values.

Saving Features to Matrix

- All computed features are concatenated into a matrix called `all_statistics`.

Saving Matrix to CSV File

- Finally, the matrix containing all extracted features is saved to a CSV file named "Features.csv" for further analysis or model training.

Overall, this code snippet demonstrates the comprehensive process of extracting various features from audio signals using MATLAB, laying the groundwork for subsequent analysis or machine learning tasks.

4.1 Method - 1

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2)

# Initialize and train the SVM classifier
svm = SVC(kernel='rbf', C=C, gamma=gamma)
svm.fit(X_train, y_train)

# Predict on the testing data
y_pred = svm.predict(X_test)

# Calculate accuracy
acc = accuracy_score(y_test, y_pred)
```

This Python code snippet demonstrates the process of training a Support Vector Machine (SVM) classifier on extracted features. Let's break down each step:

- **Splitting the Data:** The dataset is split into training and testing sets using the `train_test_split` function from the `sklearn.model_selection` module. The `stratify=y` parameter ensures that the class distribution is preserved in both the training and testing sets, which is crucial for maintaining representative samples for each class. The `test_size=0.2` parameter specifies that 20% of the data will be reserved for testing, while the remaining 80% will be used for training.
- **Initializing the SVM Classifier:** An SVM classifier is initialized with specific parameters such as the kernel type ('rbf'), regularization parameter (C), and gamma value. In this case, the 'rbf' kernel is chosen for its flexibility in capturing complex decision boundaries. The regularization parameter (C) controls the trade-off between maximizing the margin and minimizing classification errors, while the gamma parameter influences the shape of the decision boundary.
- **Training the SVM Classifier:** The initialized SVM classifier is trained on the training data (X_train, y_train) using the `fit` method. During training, the classifier learns to delineate decision boundaries that separate different classes in the feature space.
- **Predicting on Testing Data:** Once trained, the SVM classifier is used to predict the class labels of the testing data (X_test). The `predict` method applies the learned decision boundary to the feature

vectors in the testing set, assigning class labels based on their position relative to the decision boundary.

- **Calculating Accuracy:** The predicted class labels (y_pred) are compared against the true class labels (y_test) from the testing set using the `accuracy_score` function from the `sklearn.metrics` module. This function calculates the accuracy of the classifier by measuring the proportion of correctly classified instances in the testing set.

Overall, this code snippet demonstrates the process of training an SVM classifier on extracted features and evaluating its performance using a held-out testing set. By leveraging machine learning techniques, such as SVM classification, researchers can build robust models for tasks such as emotion recognition, leveraging the features extracted from audio signals.

4.2 Method - 2

Stack Ensemble Classifier:

Model Architecture:

```
rf_classifier = RandomForestClassifier(n_estimators=150, max_depth=12, random_state=42)  # Adjust n_estimators and max_depth

lr_classifier = LogisticRegression(multi_class="multinomial")

# Define the meta-classifier (MLP)

meta_classifier = MLPClassifier(hidden_layer_sizes=(100,), alpha=0.01, learning_rate_init=0.001, max_iter=200)  # Adjust hyperparameters


Create a stacking ensemble with base classifiers and meta-classifier

stacking_classifier = StackingClassifier(

    estimators=[('random_forest', rf_classifier), ('Logistic_Regression', lr_classifier)],

    final_estimator=meta_classifier

)
```

Train the stacking ensemble

```
stacking_classifier.fit(X_train_scaled, y_train)
```

Make predictions on the test set

```
y_pred_test_stacking = stacking_classifier.predict(X_test_scaled)
```

Explanation:

Other Classifier:

RandomForestClassifier:

RandomForestClassifier is an ensemble learning method based on decision trees. n_estimators determines the number of trees in the forest, max_depth controls the maximum depth of the trees, and random_state ensures reproducibility.

LogisticRegression:

LogisticRegression is a linear model for classification that uses the logistic function to predict the probability of a binary outcome. multi_class="multinomial" specifies that the logistic regression model should handle multiclass classification.

MLPClassifier:

MLPClassifier is a multi-layer perceptron classifier, a type of artificial neural network. hidden_layer_sizes specify the number of neurons in each hidden layer, alpha controls L2 regularization, learning_rate_init sets the initial learning rate, and max_iter determines the maximum number of iterations for training.

Stack Ensemble:

Base Classifiers (Estimators): The estimators parameter specifies a list of tuples, where each tuple contains a string identifier (or name) for a base classifier and the actual classifier object. In this example, the base classifiers are 'random_forest' (represented by rf_classifier) and 'Logistic_Regression' (represented by lr_classifier). These base classifiers will individually make predictions on the input data.

Meta-Classifier (Final Estimator): The final_estimator parameter specifies the meta-classifier that will combine the predictions of the base classifiers to make the final prediction. This meta-classifier takes the outputs (predictions) of the base classifiers as input features and learns to predict the target variable (or class) based on this combined information. In the given example, meta_classifier is used as the final estimator.

Training the Stacking Ensemble:

The stacking_classifier.fit(X_train_scaled, y_train) line trains the stacking ensemble classifier using the scaled training data (X_train_scaled) and corresponding labels (y_train). During training, each base classifier (e.g., random forest and logistic regression) makes predictions on the training data, and these predictions are then used as input features for the meta-classifier (final estimator). The meta-classifier learns to combine these predictions to make accurate overall predictions.

Making Predictions on the Test Set:

After training, the stacking_classifier.predict(X_test_scaled) line is used to generate predictions on the scaled test data (X_test_scaled). The stacking ensemble model uses the learned combination of base classifiers to make predictions for each instance in the test set. The resulting predictions (y_pred_test_stacking) represent the predicted class labels for the test data based on the stacking ensemble's learned model.

Chapter 5: Testing, Results, Comparative Study

5.1 Method – 1

We compiled a diverse dataset comprising audio recordings sourced from RAVDESS (960 files), TESS (2000 files), SAVEE (300 files), EMODB (230 files), and CREMAD (5435 files), collectively spanning five distinct emotional categories: happy, disgust, fearful, sad, and angry. Each audio clip ranged from 3 to 5 seconds in duration. Through Kaggle.com, we accessed and assembled the dataset, gathering mono-format audio files quantized to 16-bit resolution at a sampling rate of 48kHz. Our analytical journey commenced by importing these .wav files into Python for thorough examination.

Python served as our primary tool for analyzing the audio data, allowing us to delve into the unique traits and nuances encapsulated within each sample. This exploration fostered a deeper comprehension of audio features and their intricate characteristics. Furthermore, we harnessed Python's capabilities for classification purposes. Leveraging the scikit-learn library, we constructed a CSV file encompassing the audio data alongside their corresponding features. This CSV acted as the conduit for seamlessly integrating our data with Python code, facilitating the extraction of pertinent information for model training.

By harnessing the synergy between our audio dataset, MATLAB, Python, and Jupyter Notebook, we navigated through an extensive analysis journey. We unraveled invaluable insights into the diverse array of audio features, meticulously studying their variations and correlations. Through this meticulous research and analytical process, we forged ahead in the realm of audio processing, with a particular emphasis on its applications in emotion analysis. Our efforts culminated in the development of robust classification models tailored for emotion recognition within songs, marking a significant contribution to the broader understanding of audio processing and its multifaceted applications.

5.1.1 Testing Data

To validate the robustness and effectiveness of our implementation, we embarked on a series of rigorous tests using a diverse array of audio files. These files, all in the .wav format, were meticulously chosen to encapsulate a wide spectrum of emotional expressions. They were sourced from various avenues, including extraction from videos and direct downloads from reputable sources. Before subjecting them to our system, we took great care in ensuring uniformity in input format and quality through meticulous pre-processing steps.

Our selection of audio samples was guided by the aim to encompass a comprehensive range of emotions, including but not limited to happiness, sadness, anger, fear, and disgust. To ensure a balanced representation, we curated a mix of male and female voices, each contributing to the richness and diversity of emotional expressions. Each audio file, carefully vetted, possessed a duration ranging from 3 to 5 seconds, a timeframe deemed sufficient for meaningful analysis and classification.

In the experimental phase, we divided our dataset into two equal partitions: one for training and the other for testing, employing an 80-20 split. This deliberate division strategy aimed to ensure that both the training and testing sets harbored a fair and representative sample of each emotion category. Subsequently, we embarked on training our model using the training dataset, allowing it to learn and adapt to the nuances of the emotional expressions encapsulated within.

Following the training phase, we subjected our model to rigorous testing on the reserved test dataset. Here, the model's performance was meticulously evaluated, with a particular focus on accuracy. By comparing the predicted emotions with the ground truth labels of the test data, we meticulously computed the accuracy rate, providing us with a quantitative measure of our system's efficacy.

The culmination of these experiments, characterized by comprehensive testing and meticulous analysis, served to validate the accuracy and reliability of our system. Through a diligent examination of results and insights garnered from the testing process, we were able to affirm the robustness and effectiveness of our implementation in accurately discerning and classifying emotional expressions within audio data.

5.1.2 Plots

We've created plots showcasing the variations in emotional expressions across all features within each dataset. These visual representations offer insights into how different features respond to various emotions, aiding in understanding the dataset's emotional dynamics. This comprehensive analysis guides our subsequent modeling and classification tasks.

Plots of RAVDESS Dataset

Fig 5.1.2.1 – Fig 5.1.2.25 showcase plots of the variations in emotional expressions across all features within the RAVDESS dataset. These visual representations provide insights into how different features respond to various emotions, offering a deeper understanding of the dataset's emotional dynamics.

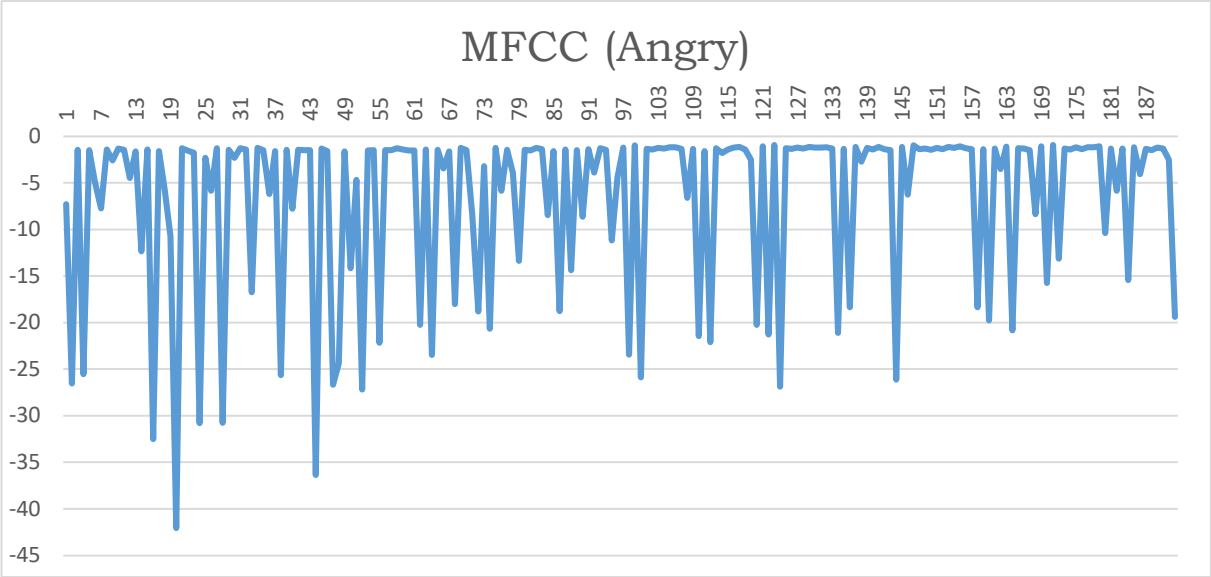

Fig 5.1.2.1: Plotting MFCC for 'Angry' Emotion Label

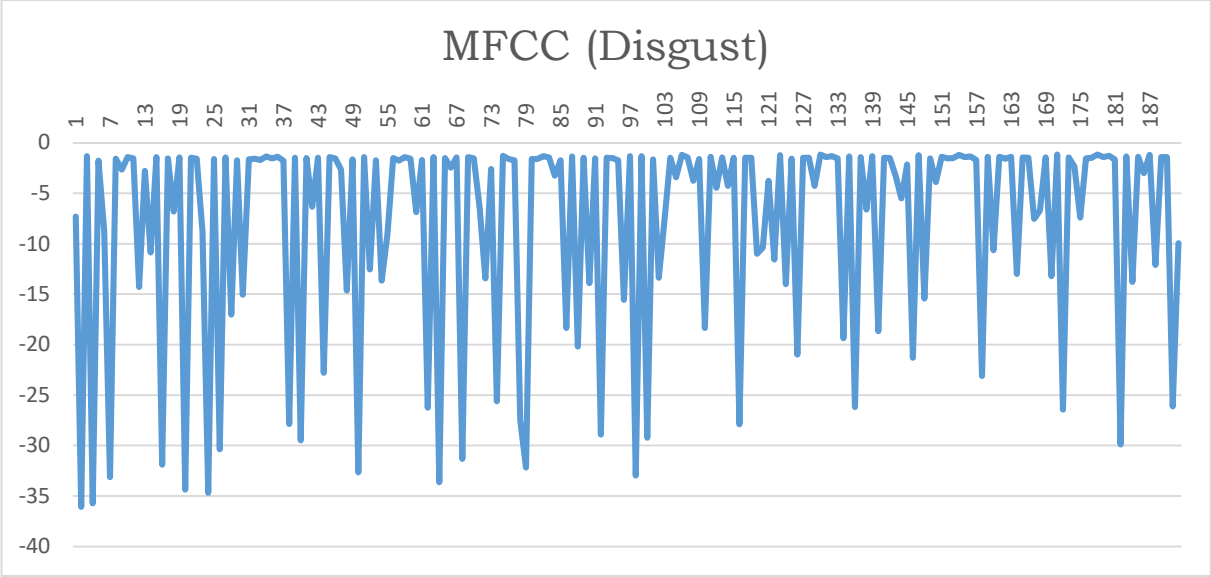

Fig 5.1.2.2: Plotting MFCC for 'Disgust' Emotion Label

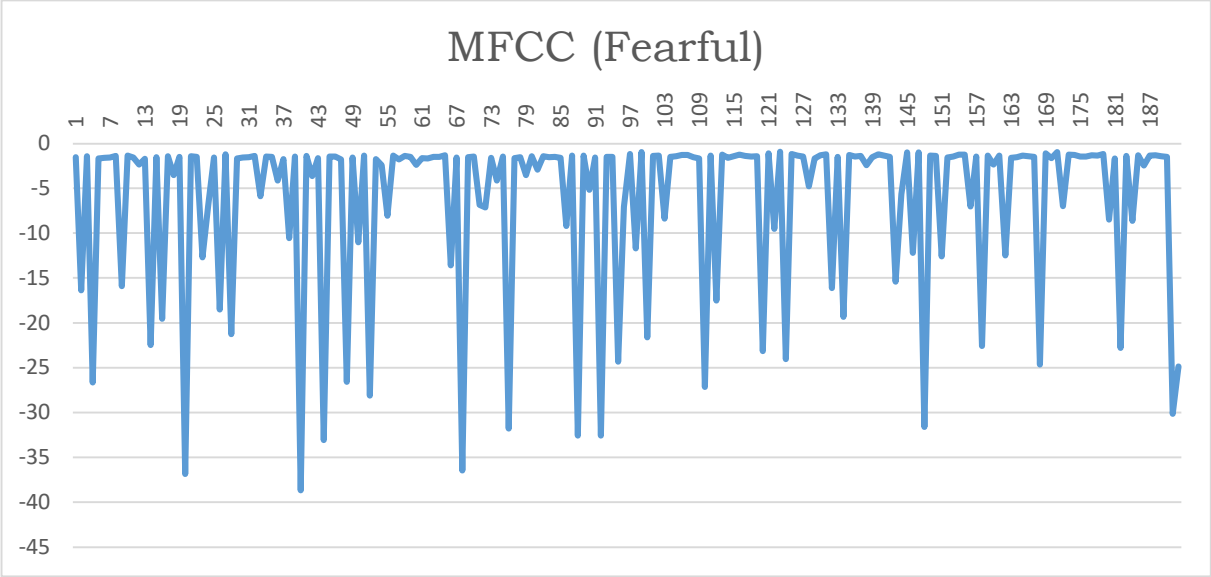

Fig 5.1.2.3: Plotting MFCC for 'Fearful' Emotion Label

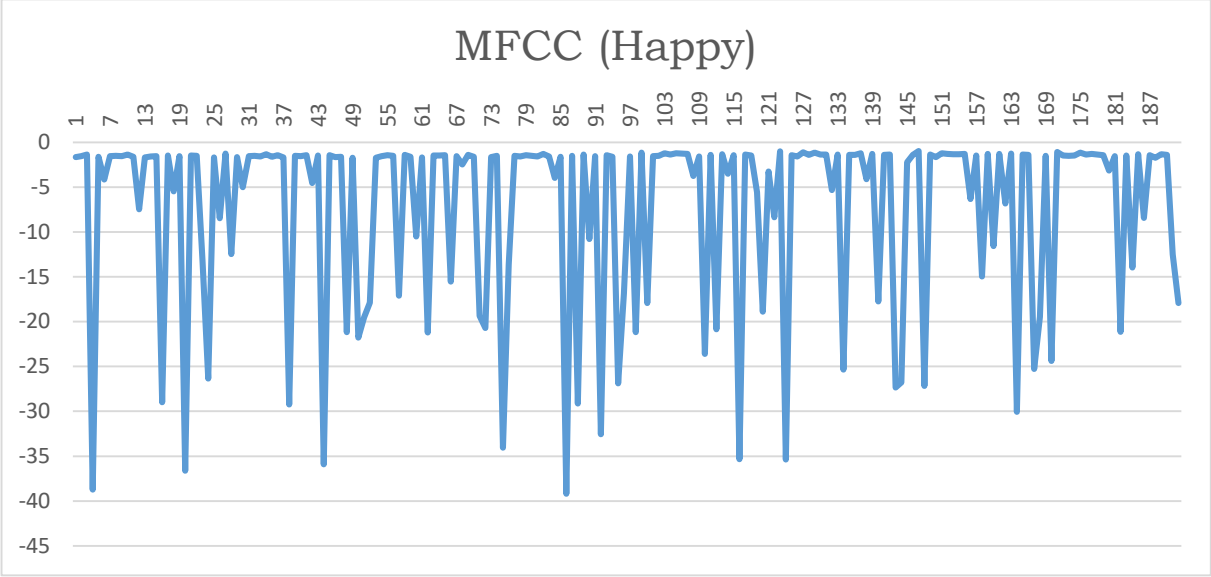

Fig 5.1.2.4: Plotting MFCC for 'Happy' Emotion Label

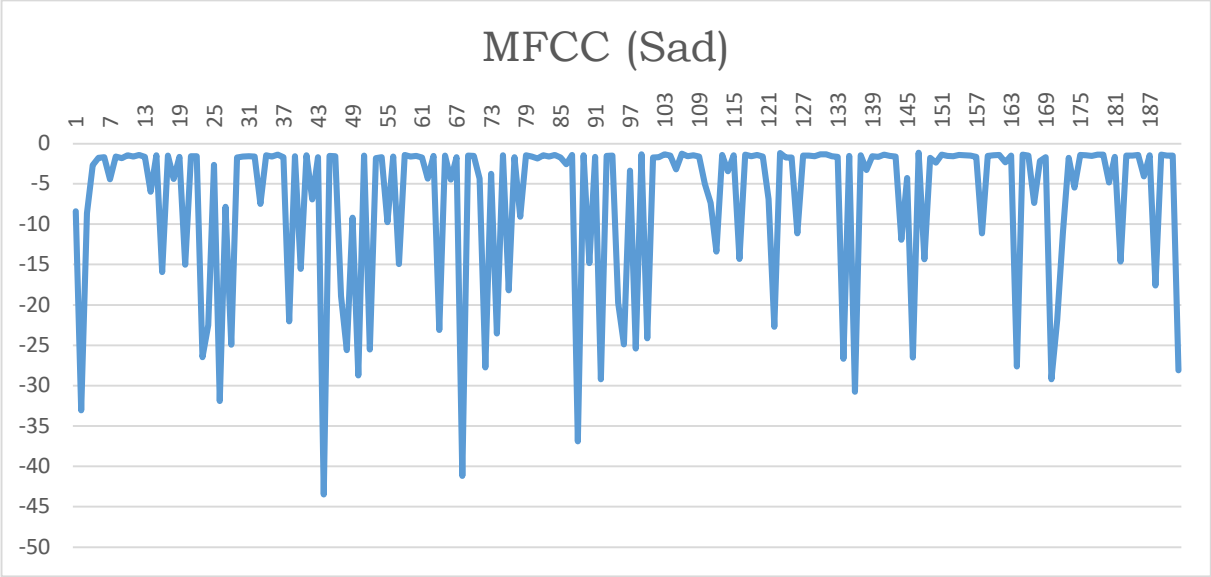

Fig 5.1.2.5: Plotting MFCC for 'Sad' Emotion Label

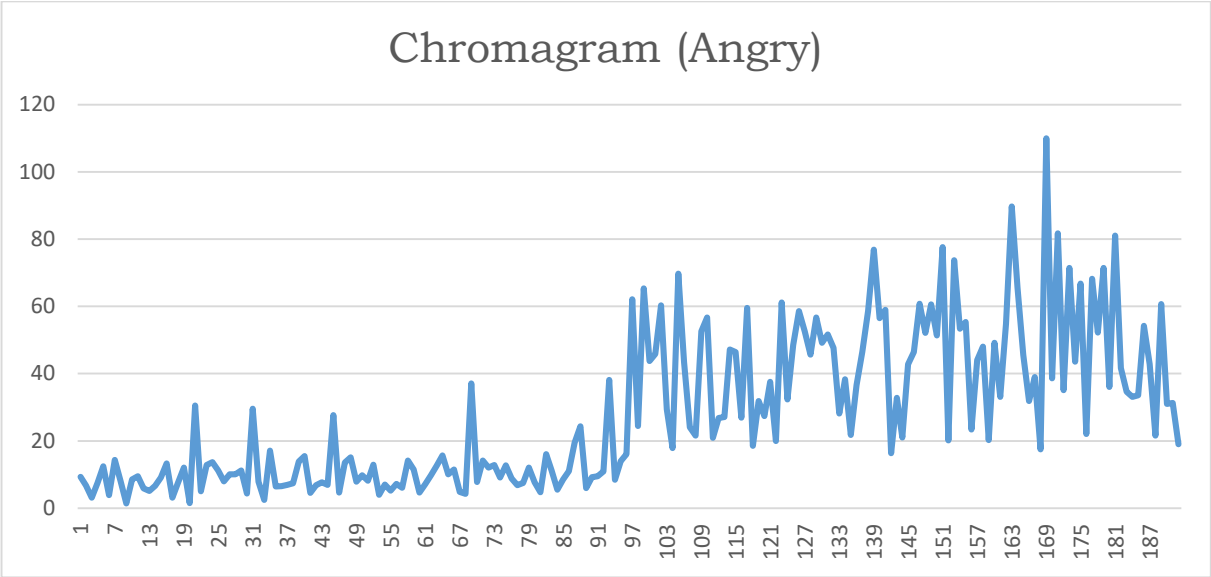

Fig 5.1.2.6: Plotting Chromagram for 'Angry' Emotion Label

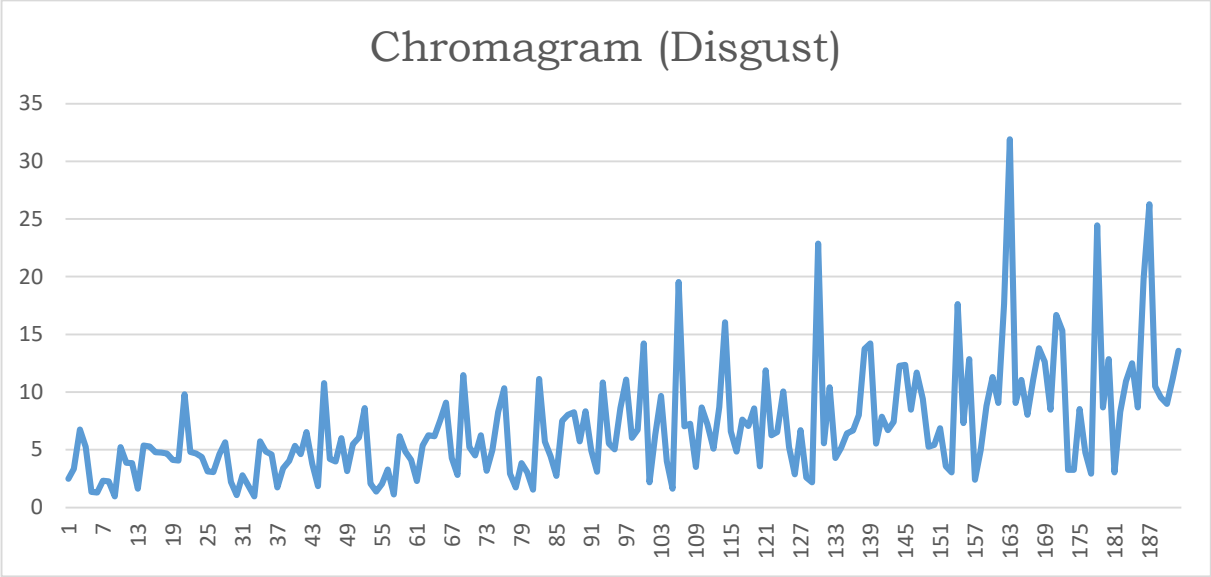

Fig 5.1.2.7: Plotting Chromagram for 'Disgust' Emotion Label

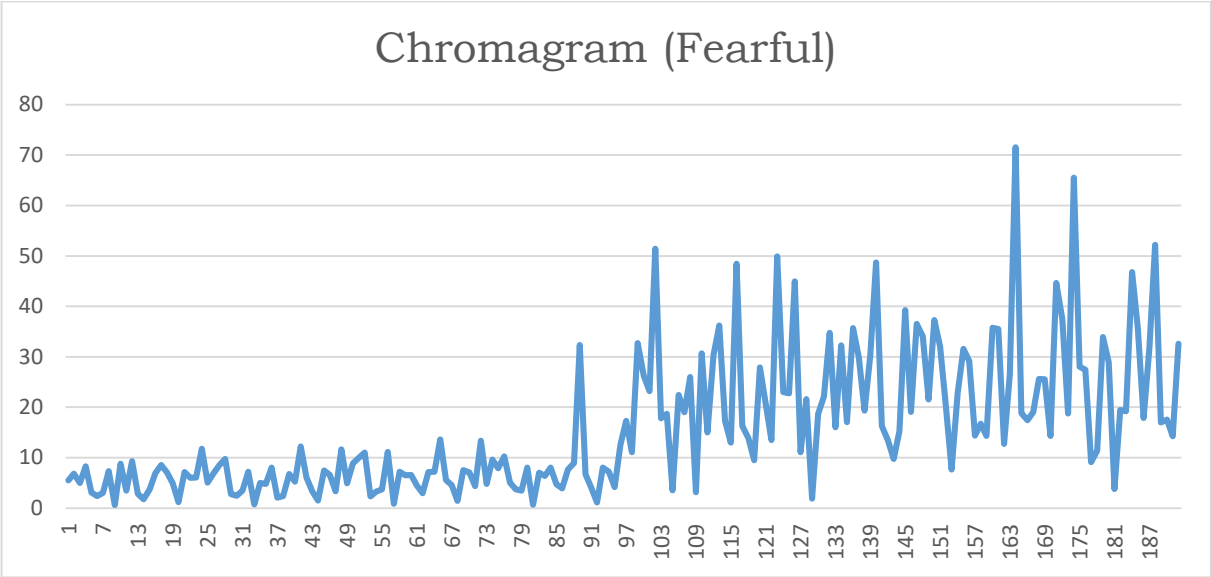

Fig 5.1.2.8: Plotting Chromagram for 'Fearful' Emotion Label

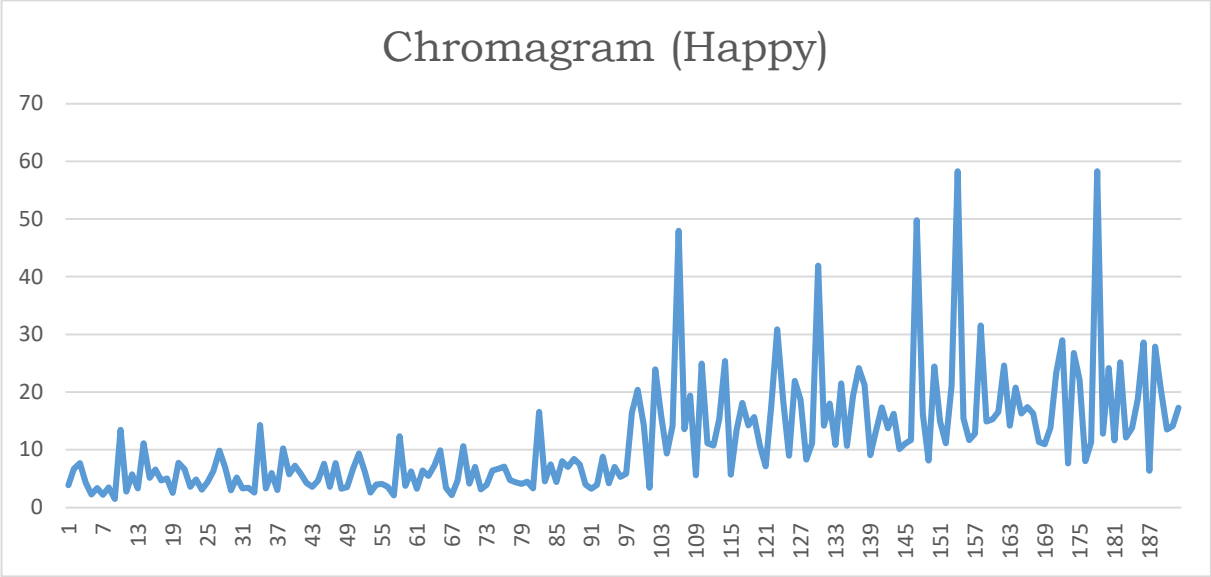

Fig 5.1.2.9: Plotting Chromagram for 'Happy' Emotion Label

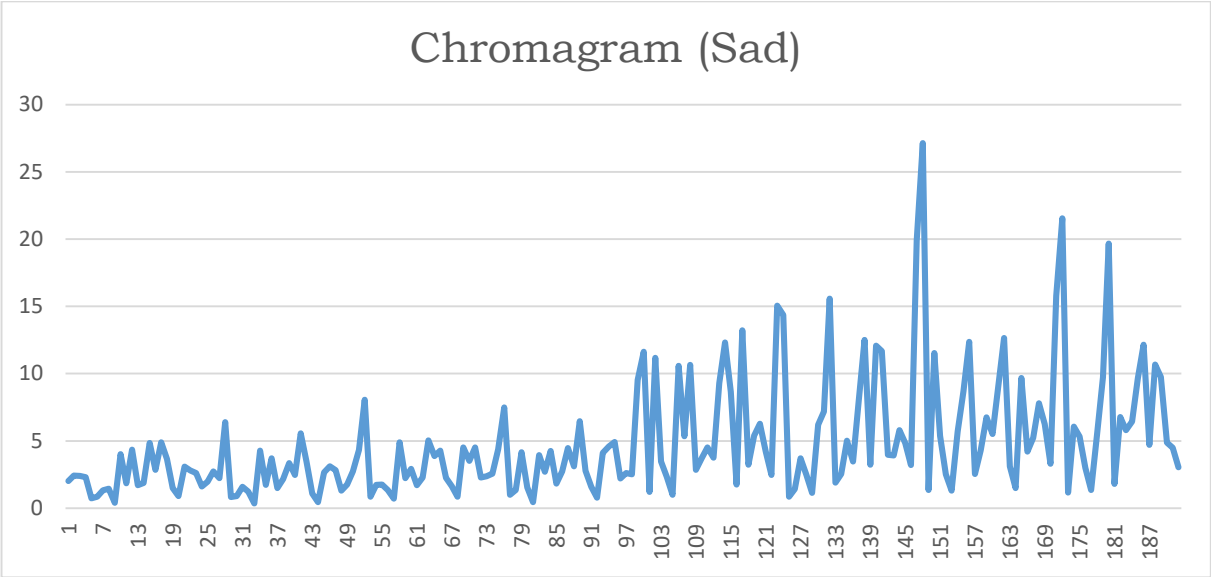

Fig 5.1.2.10: Plotting Chromagram for 'Sad' Emotion Label

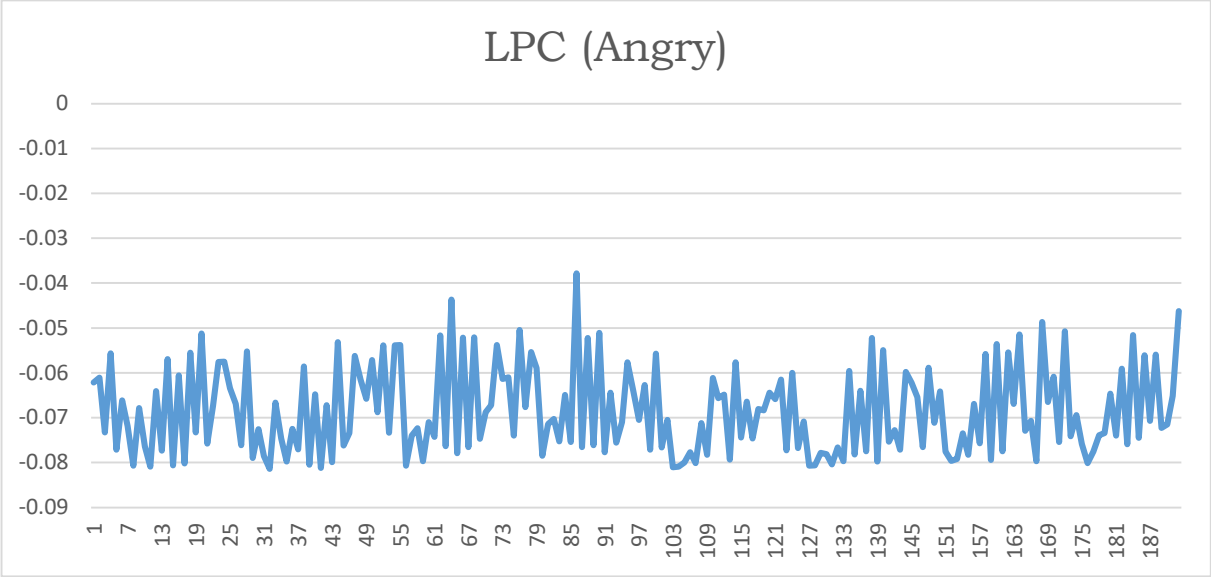

Fig 5.1.2.11: Plotting LPC for 'Angry' Emotion Label

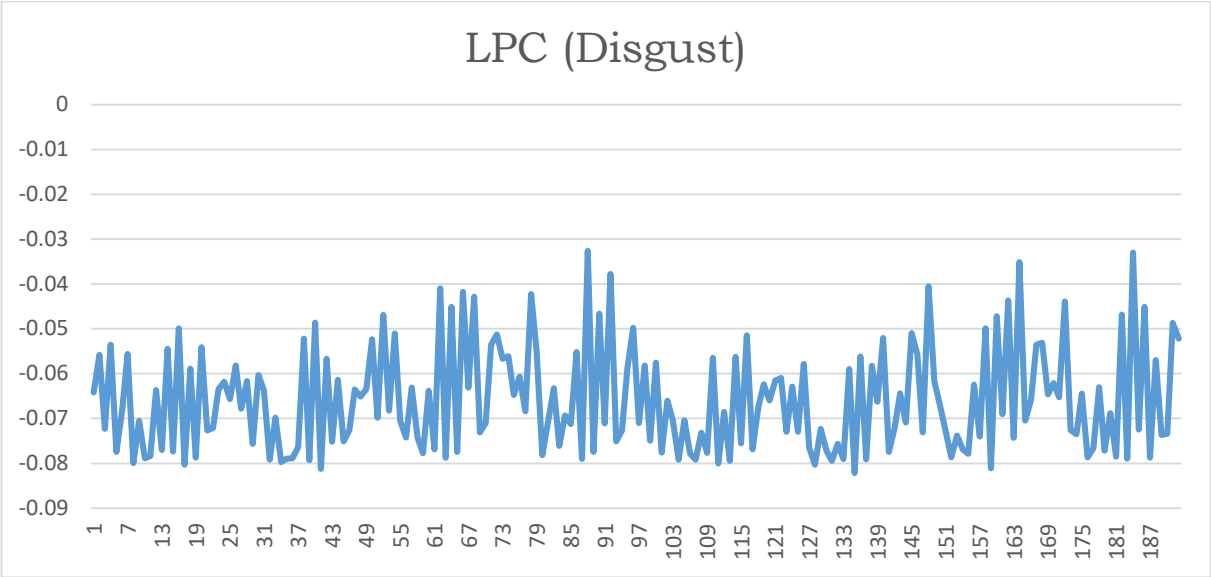

Fig 5.1.2.12: Plotting LPC for 'Disgust' Emotion Label

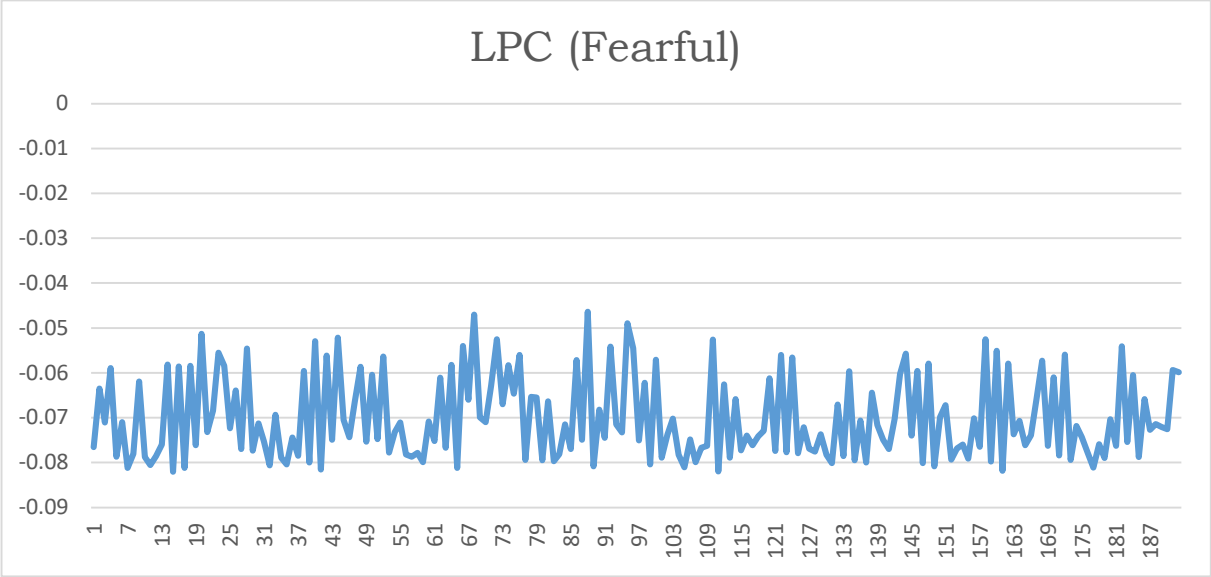

Fig 5.1.2.13: Plotting LPC for 'Fearful' Emotion Label

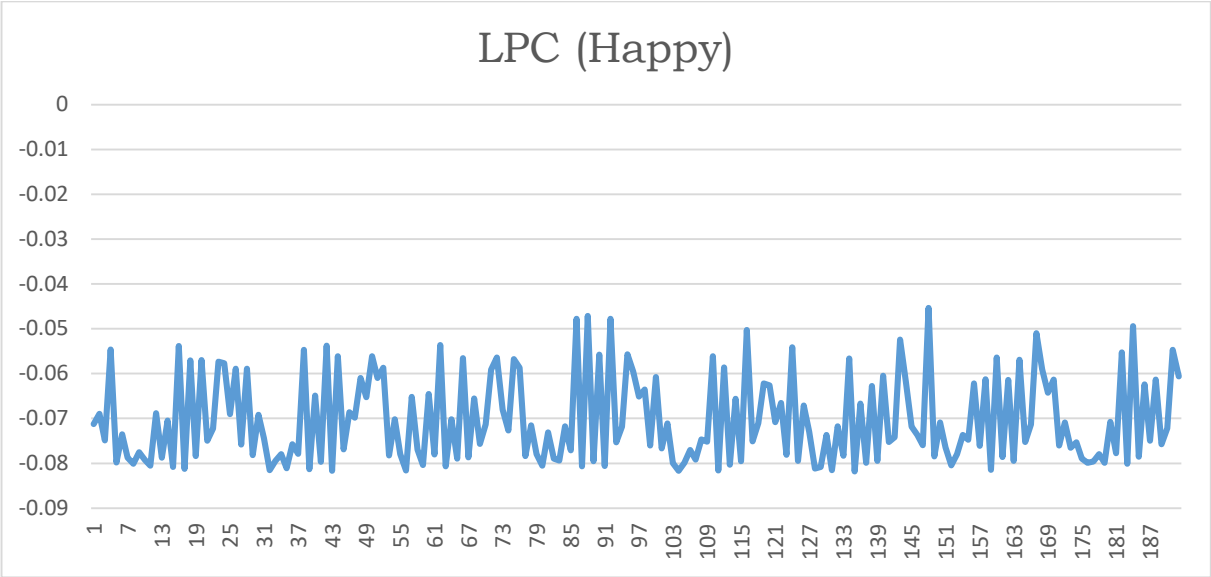

Fig 5.1.2.14: Plotting LPC for 'Happy' Emotion Label

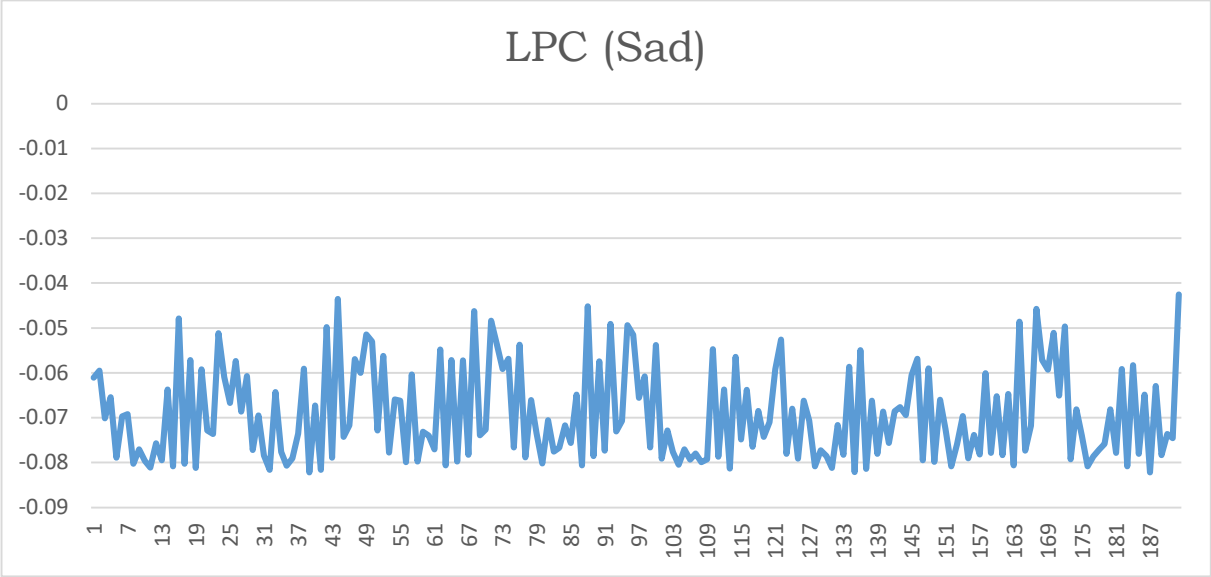

Fig 5.1.2.15: Plotting LPC for 'Sad' Emotion Label

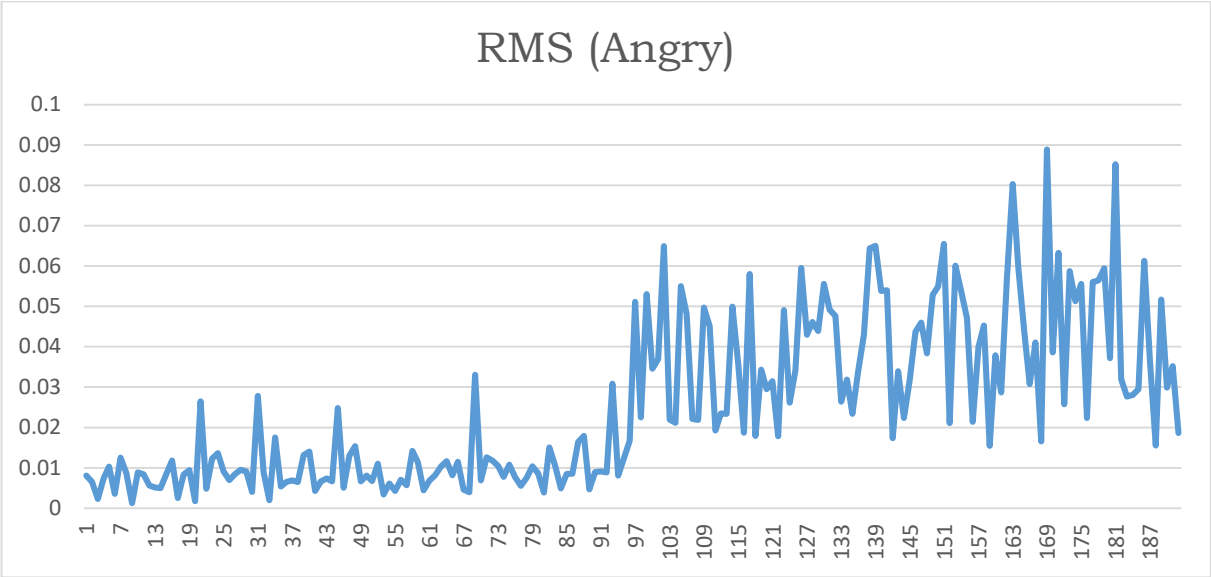

Fig 5.1.2.16: Plotting RMS for 'Angry' Emotion Label

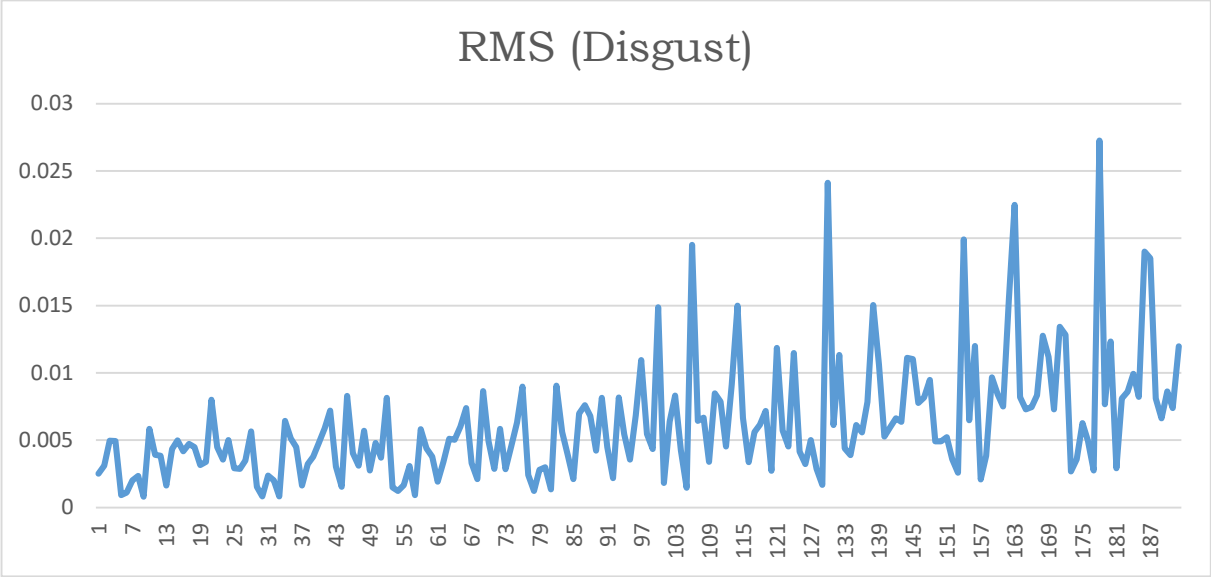

Fig 5.1.2.17: Plotting RMS for 'Disgust' Emotion Label

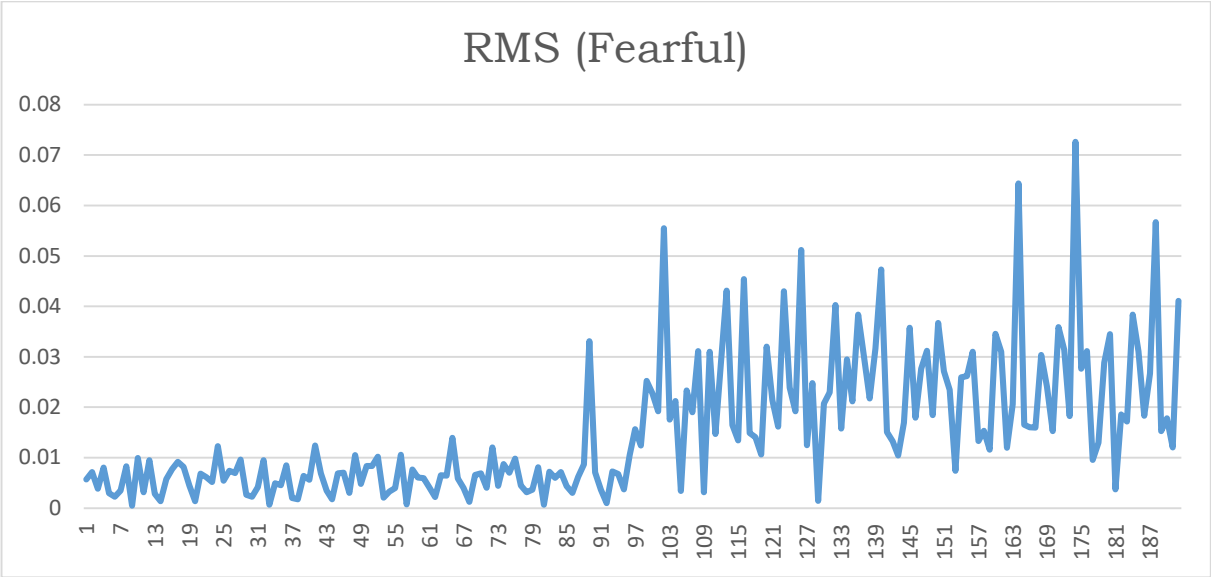

Fig 5.1.2.18: Plotting RMS for 'Fearful' Emotion Label

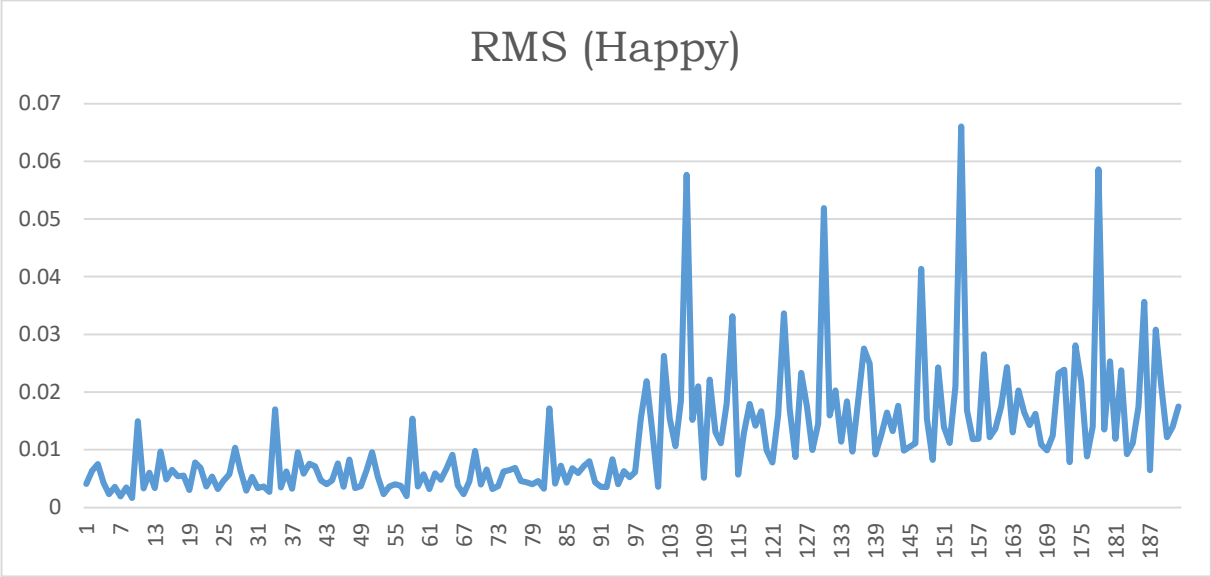

Fig 5.1.2.19: Plotting RMS for 'Happy' Emotion Label

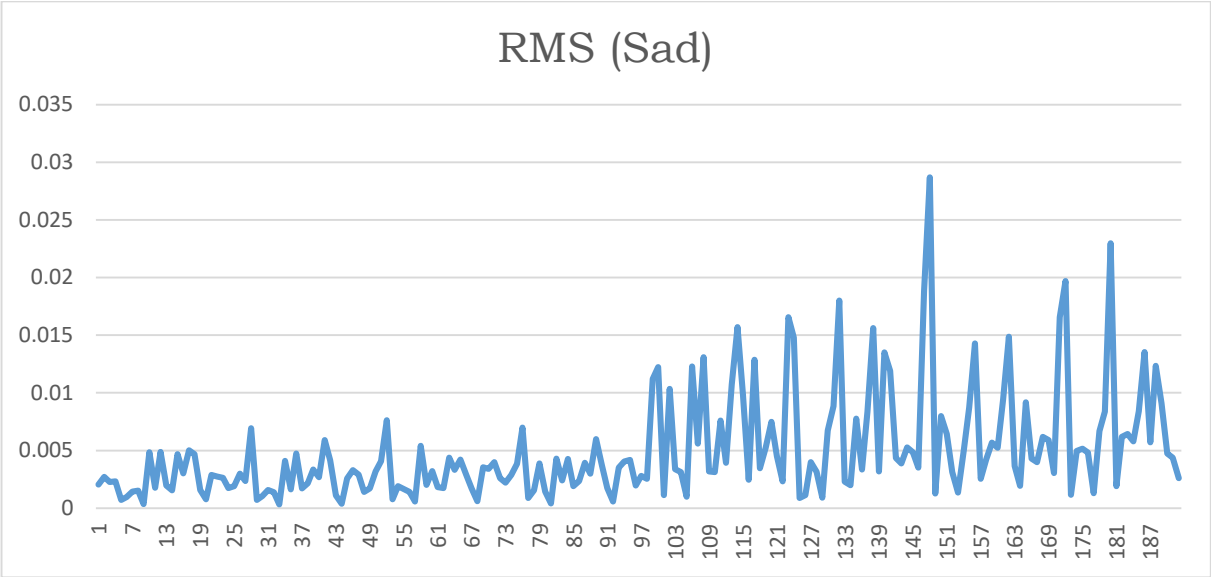

Fig 5.1.2.20: Plotting RMS for 'Sad' Emotion Label

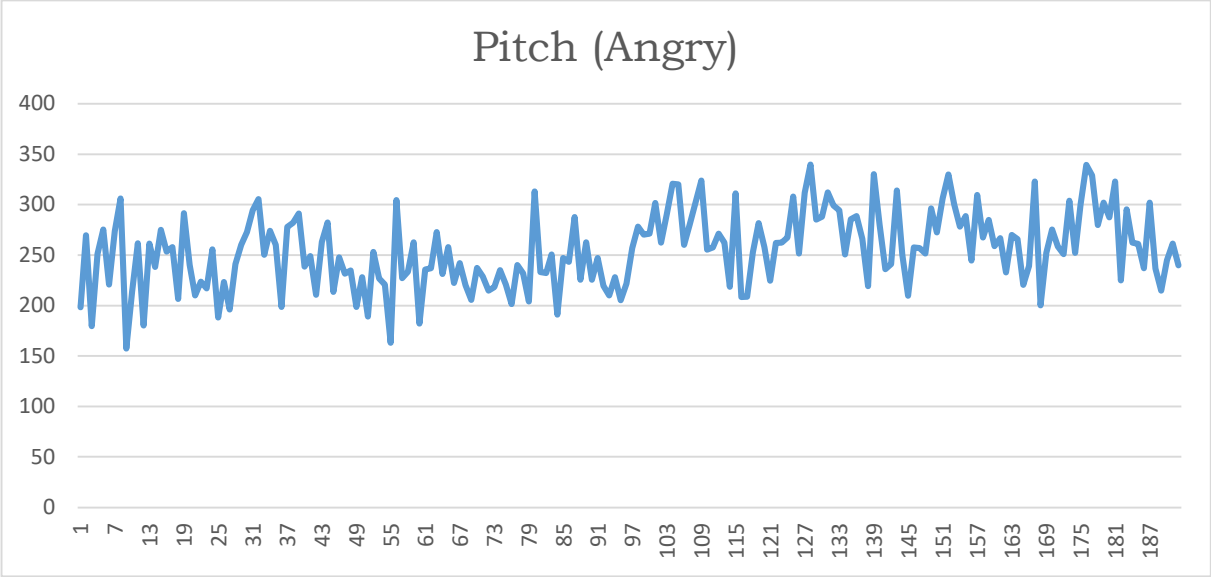

Fig 5.1.2.21: Plotting Pitch for 'Angry' Emotion Label

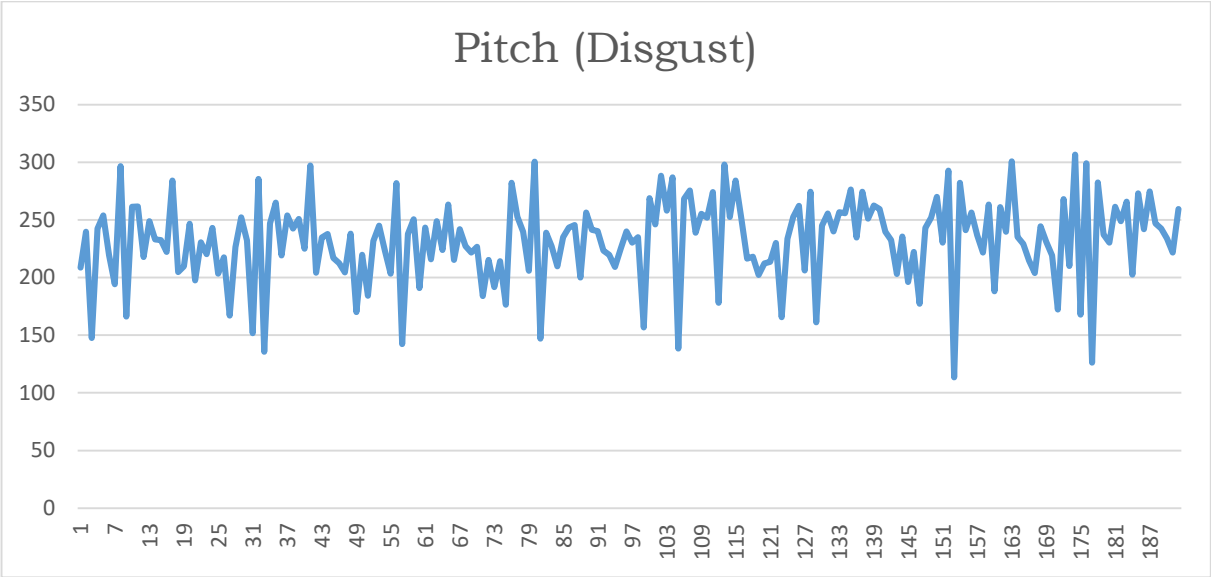

Fig 5.1.2.22: Plotting Pitch for 'Disgust' Emotion Label

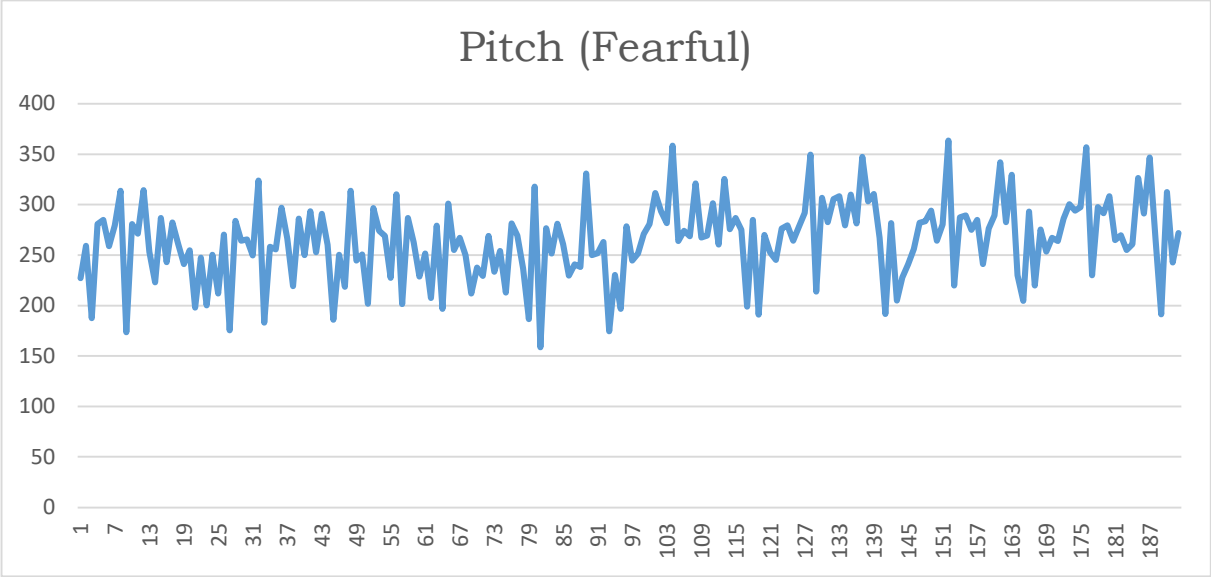

Fig 5.1.2.23: Plotting Pitch for 'Fearful' Emotion Label

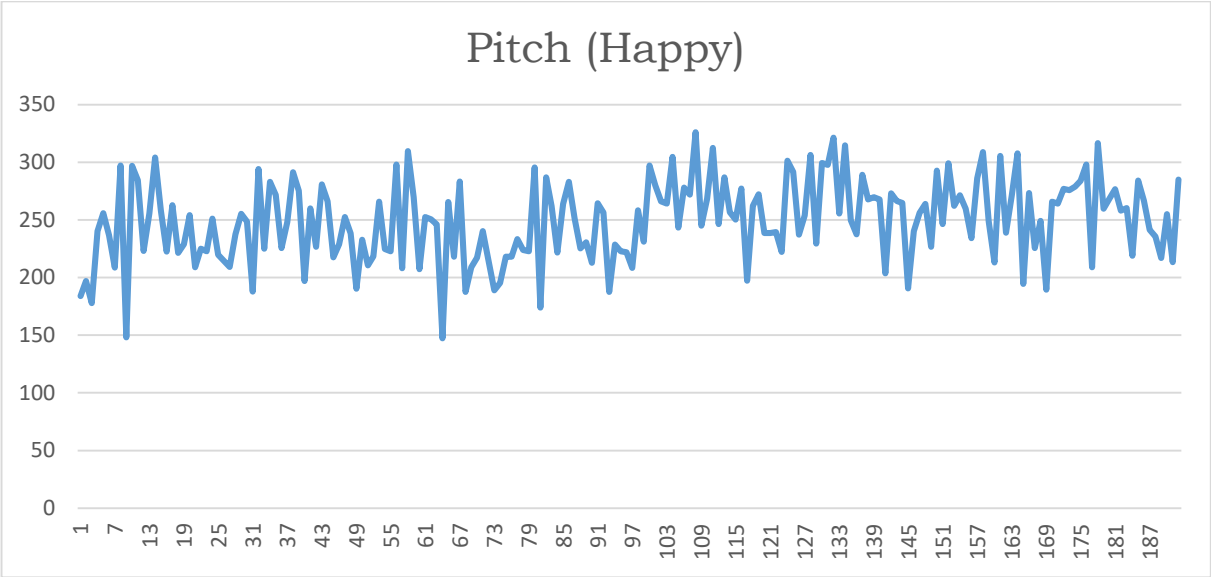

Fig 5.1.2.24: Plotting Pitch for 'Happy' Emotion Label

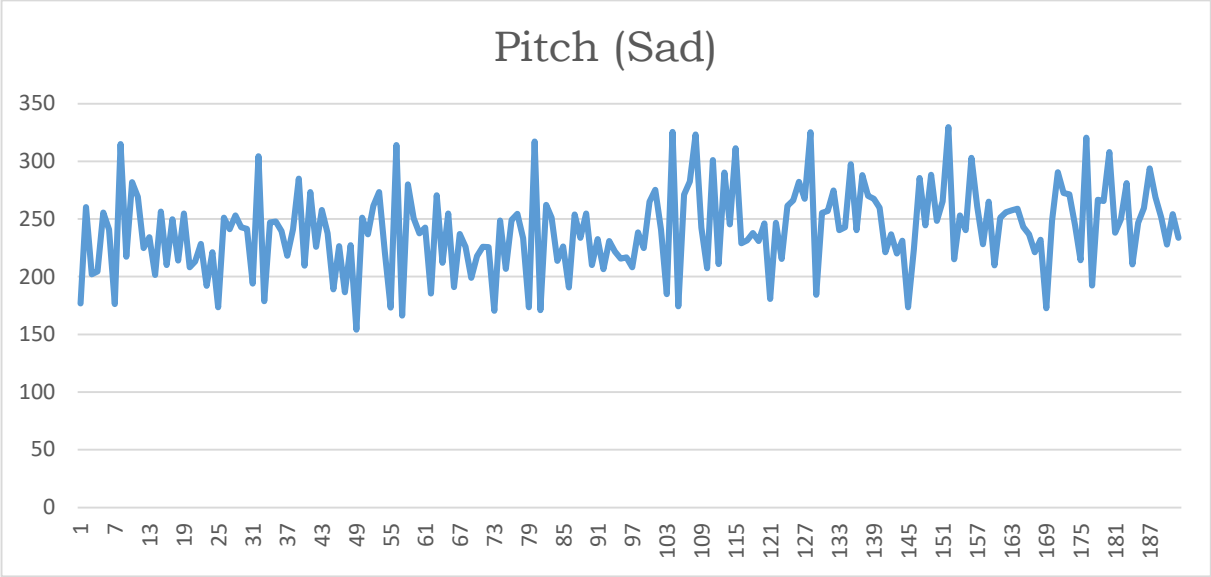

Fig 5.1.2.25: Plotting Pitch for 'Sad' Emotion Label

Plots of TESS Dataset

Fig 5.1.2.26 – Fig 5.1.2.60 showcase plots of the variations in emotional expressions across all features within the TESS dataset. These visual representations provide insights into how different features respond to various emotions, offering a deeper understanding of the dataset's emotional dynamics.

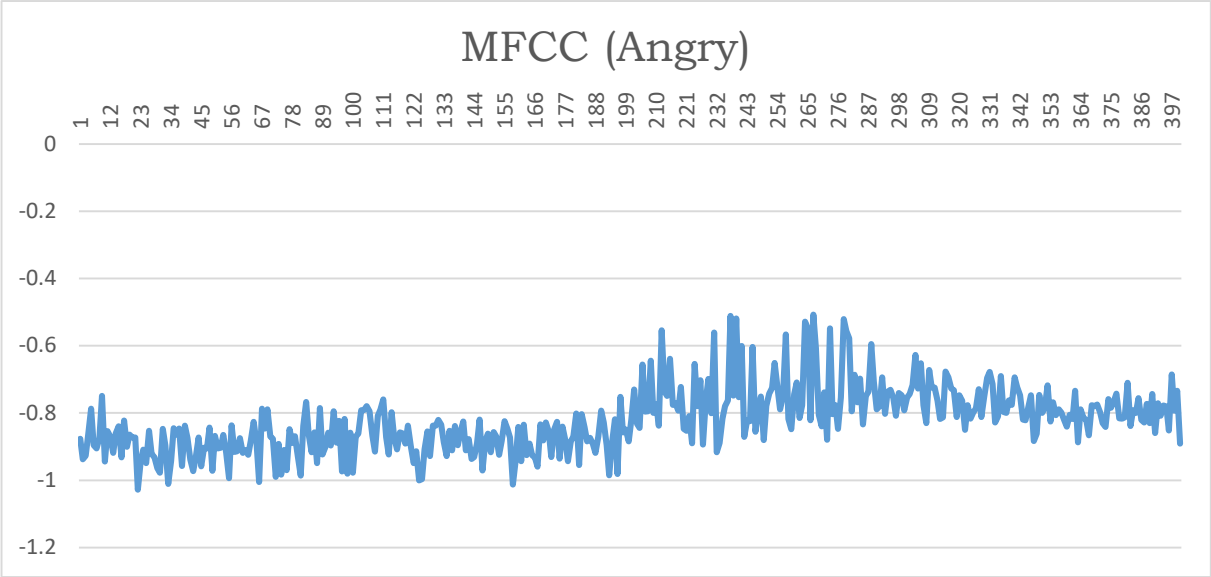

Fig 5.1.2.26: Plotting MFCC for 'Angry' Emotion Label

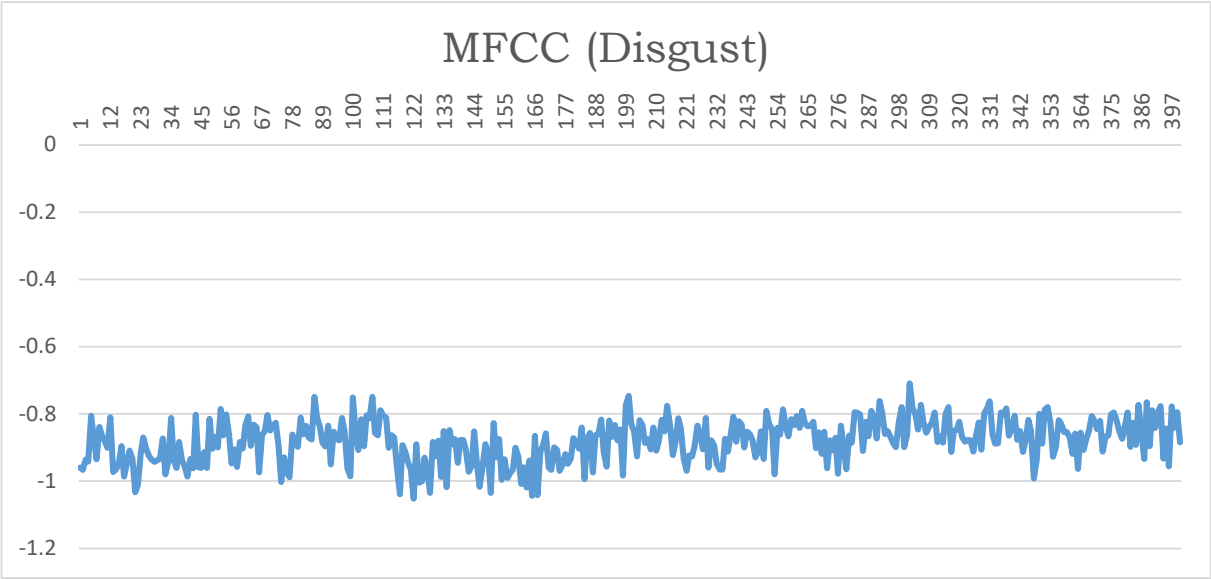

Fig 5.1.2.27: Plotting MFCC for 'Disgust' Emotion Label

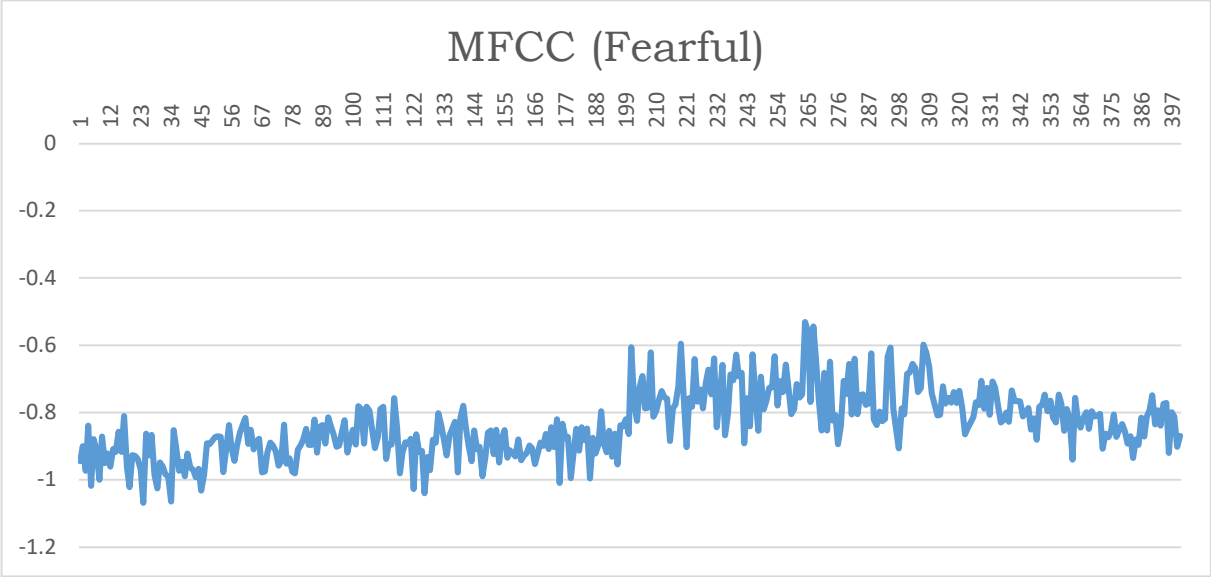

Fig 5.1.2.28: Plotting MFCC for 'Fearful' Emotion Label

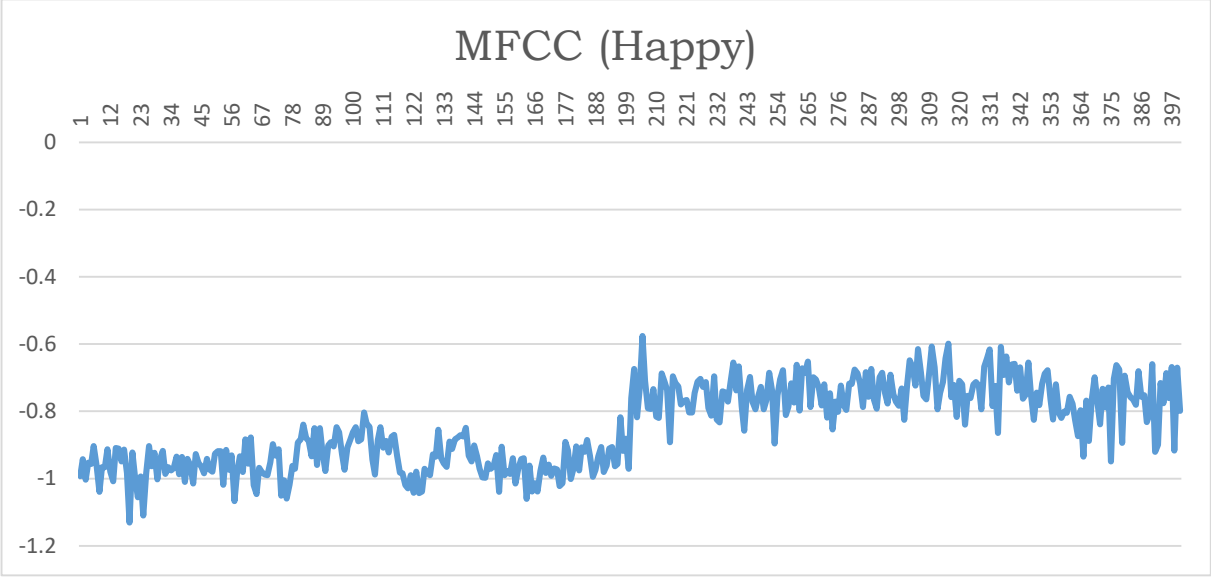

Fig 5.1.2.29: Plotting MFCC for 'Happy' Emotion Label

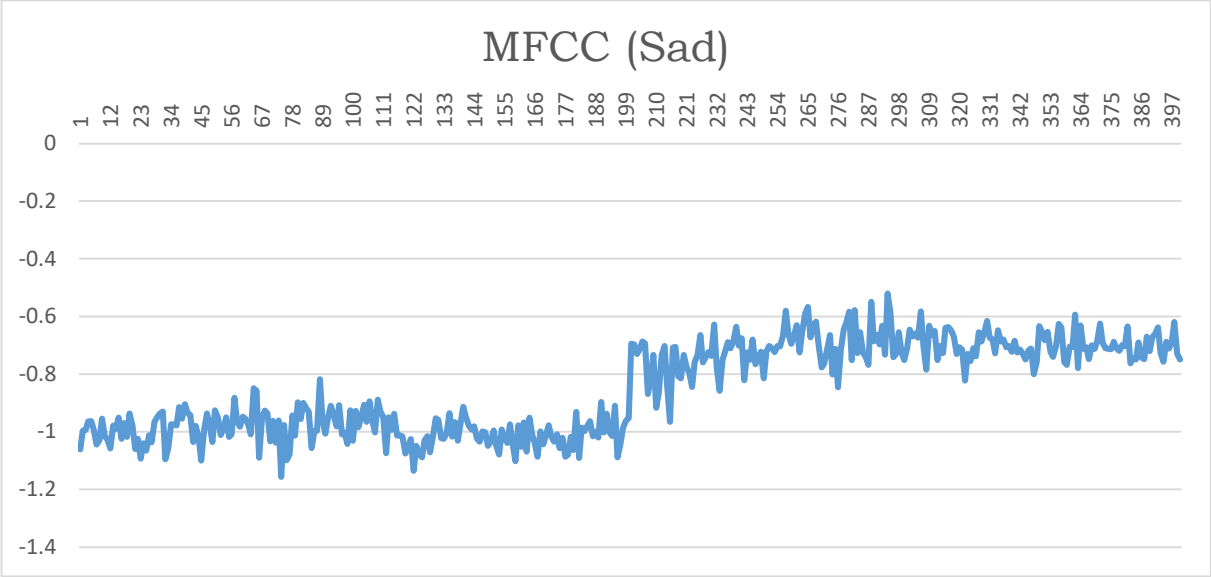

Fig 5.1.2.30: Plotting MFCC for 'Sad' Emotion Label

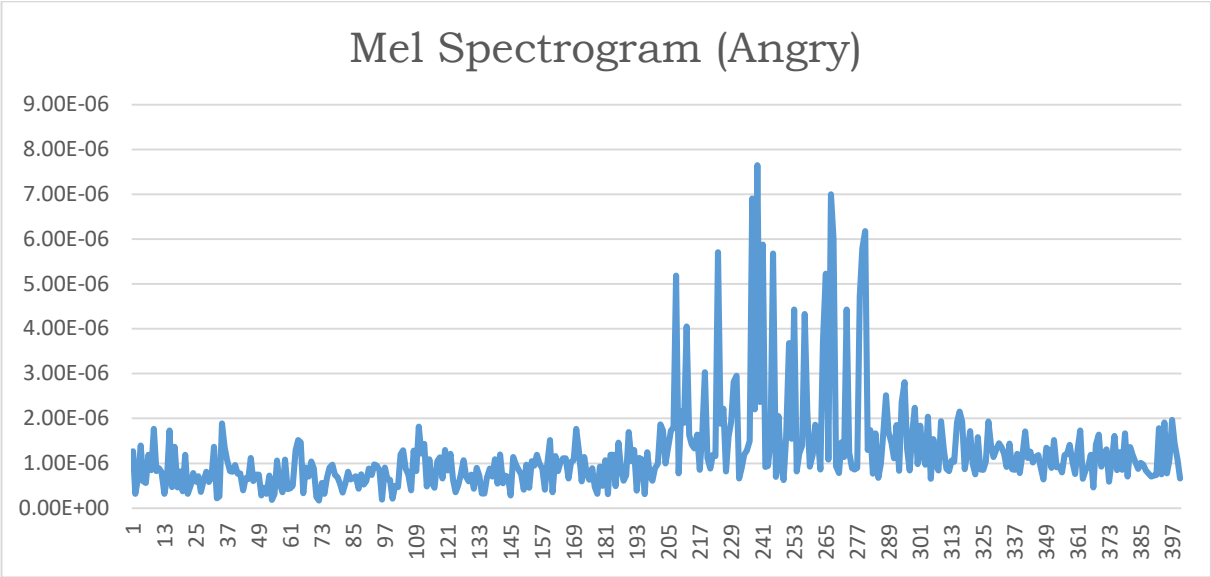

Fig 5.1.2.31: Plotting Mel Spectrogram for 'Angry' Emotion Label

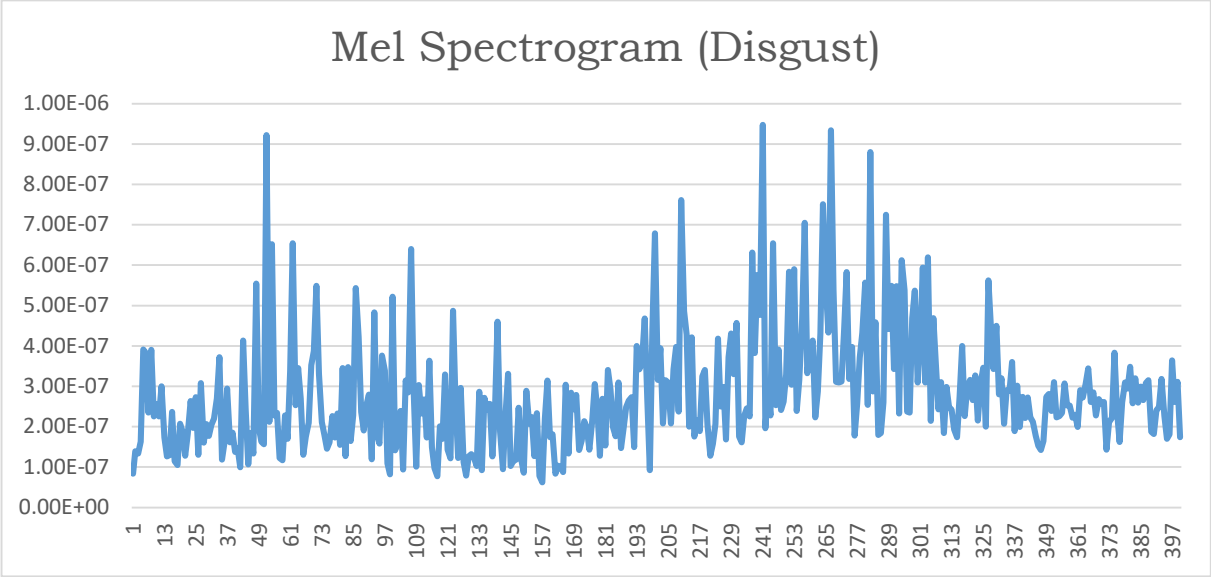

Fig 5.1.2.32: Plotting Mel Spectrogram for 'Disgust' Emotion Label

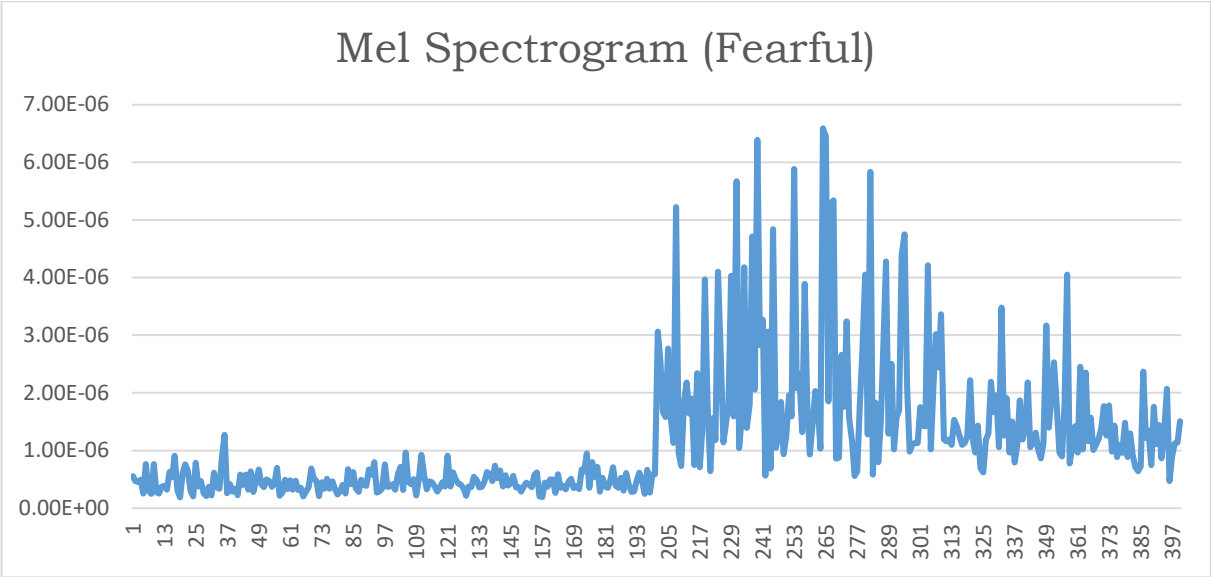

Fig 5.1.2.33: Plotting Mel Spectrogram for 'Fearful' Emotion Label

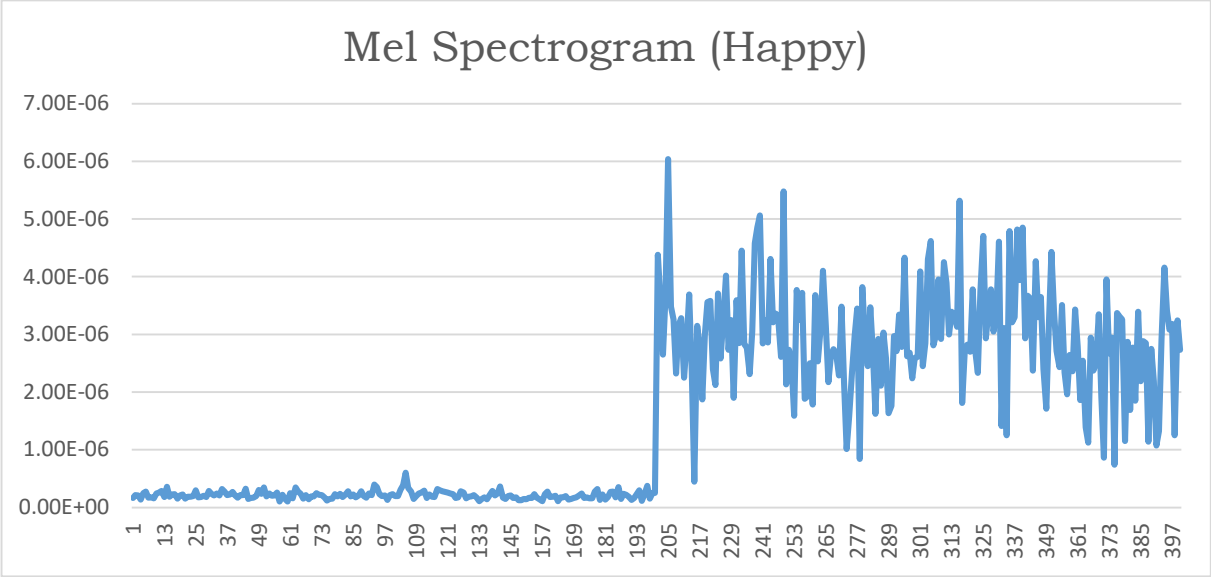

Fig 5.1.2.34: Plotting Mel Spectrogram for 'Happy' Emotion Label

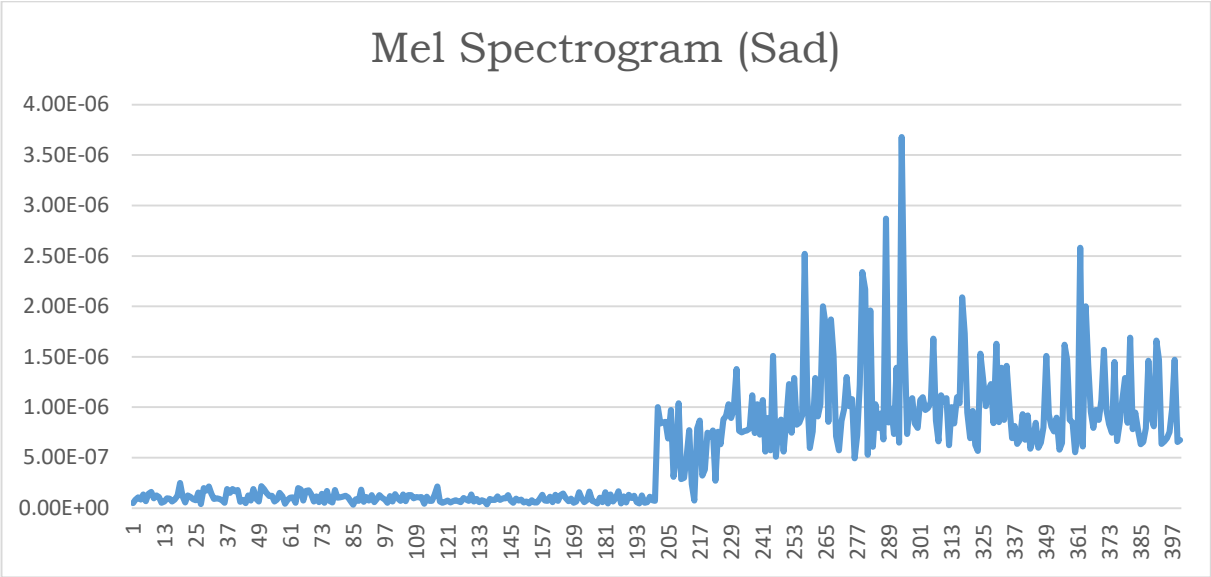

Fig 5.1.2.35: Plotting Mel Spectrogram for 'Sad' Emotion Label

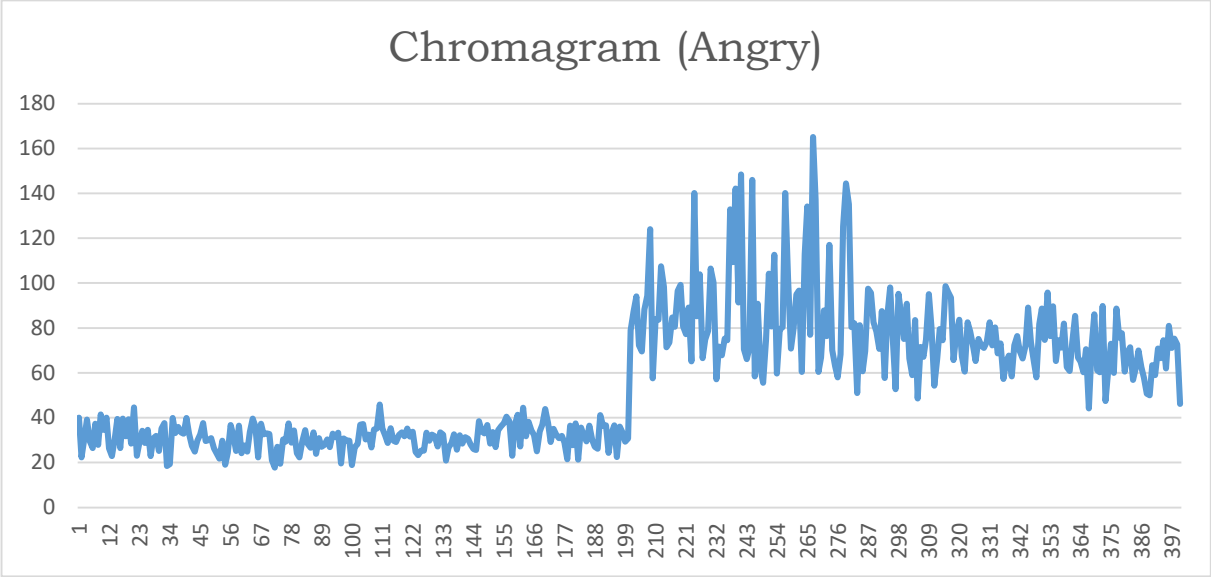

Fig 5.1.2.36: Plotting Chromagram for 'Angry' Emotion Label

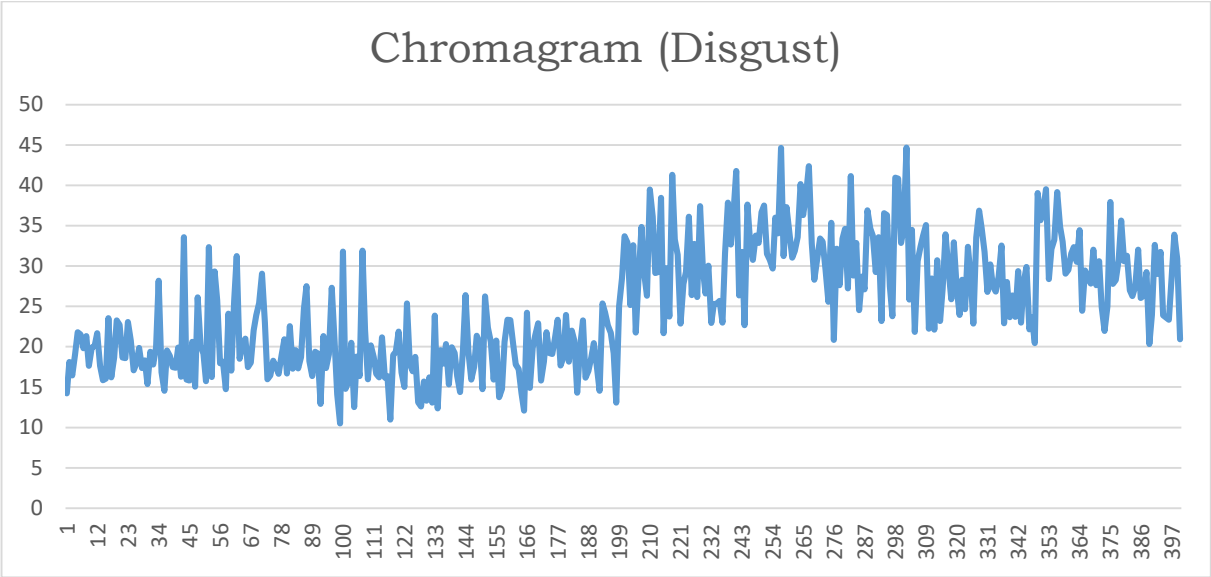

Fig 5.1.2.37: Plotting Chromagram for 'Disgust' Emotion Label

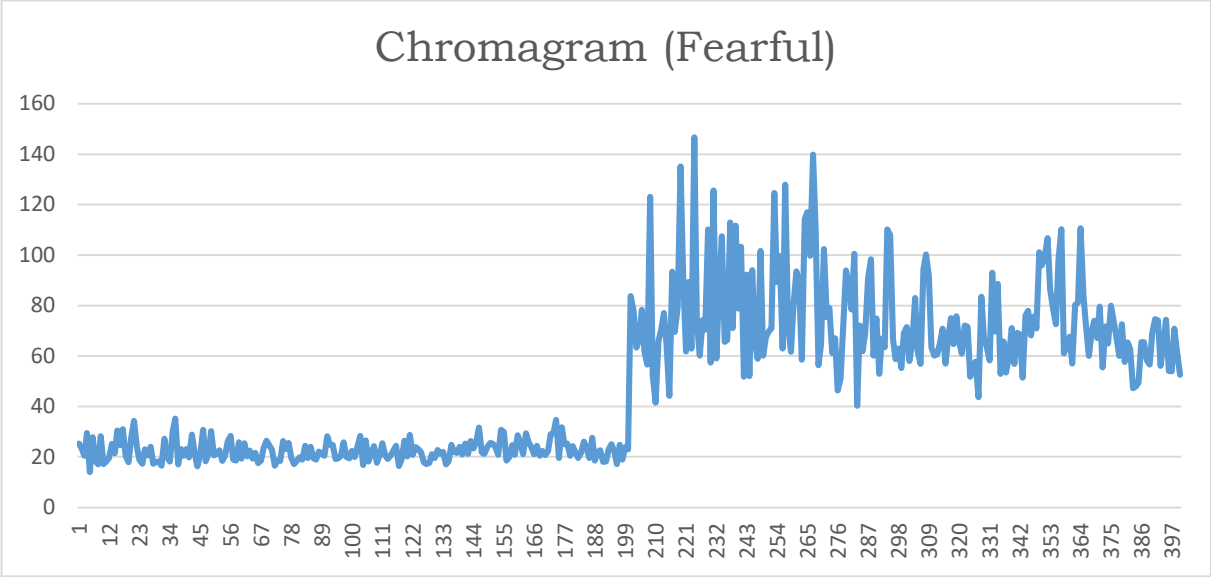

Fig 5.1.2.38: Plotting Chromagram for 'Fearful' Emotion Label

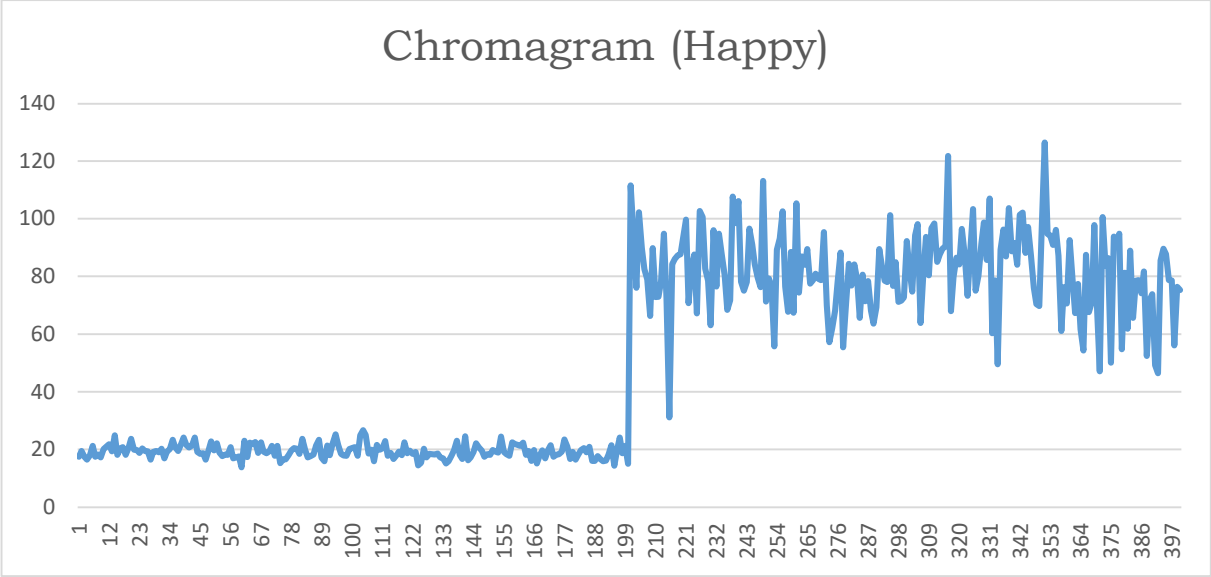

Fig 5.1.2.39: Plotting Chromagram for 'Happy' Emotion Label

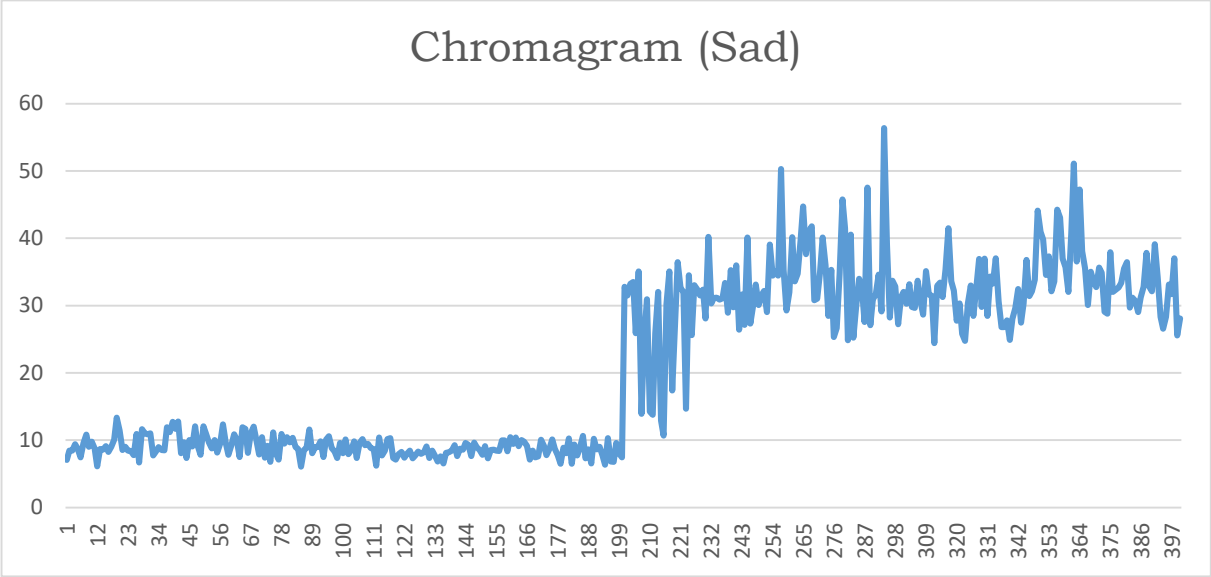

Fig 5.1.2.40: Plotting Chromagram for 'Sad' Emotion Label

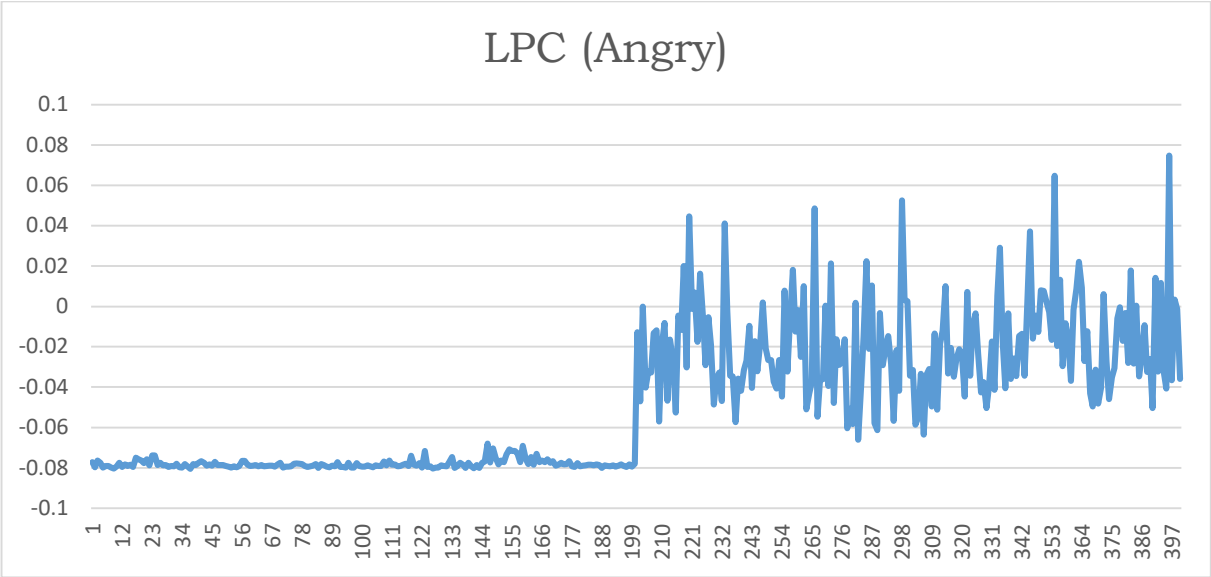

Fig5.1.2.41: Plotting LPC for 'Angry' Emotion Label

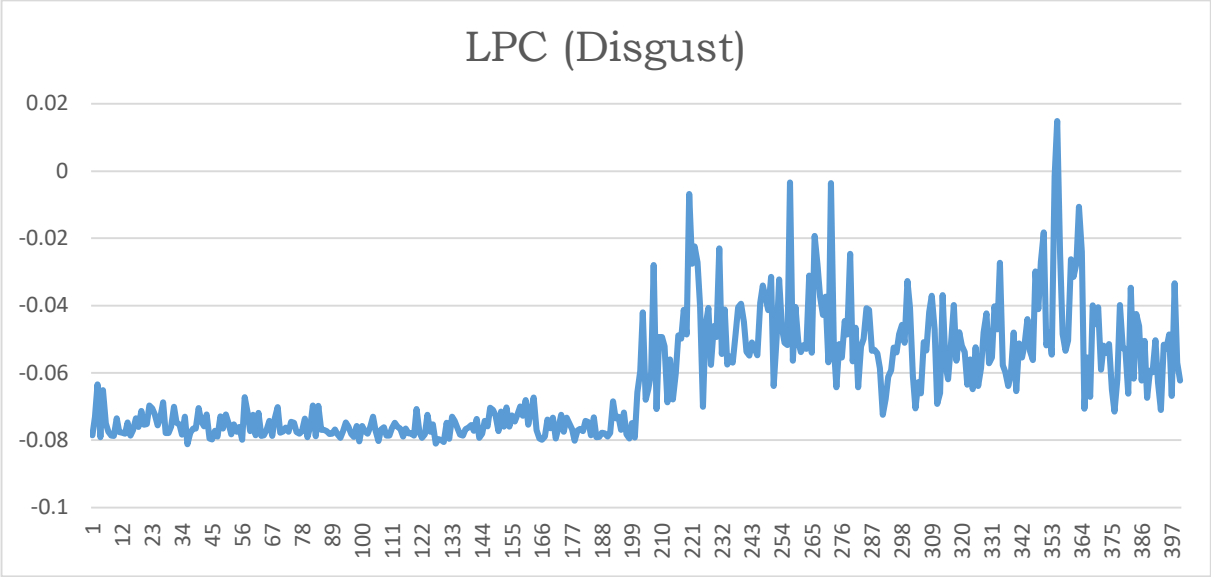

Fig 5.1.2.42: Plotting LPC for 'Disgust' Emotion Label

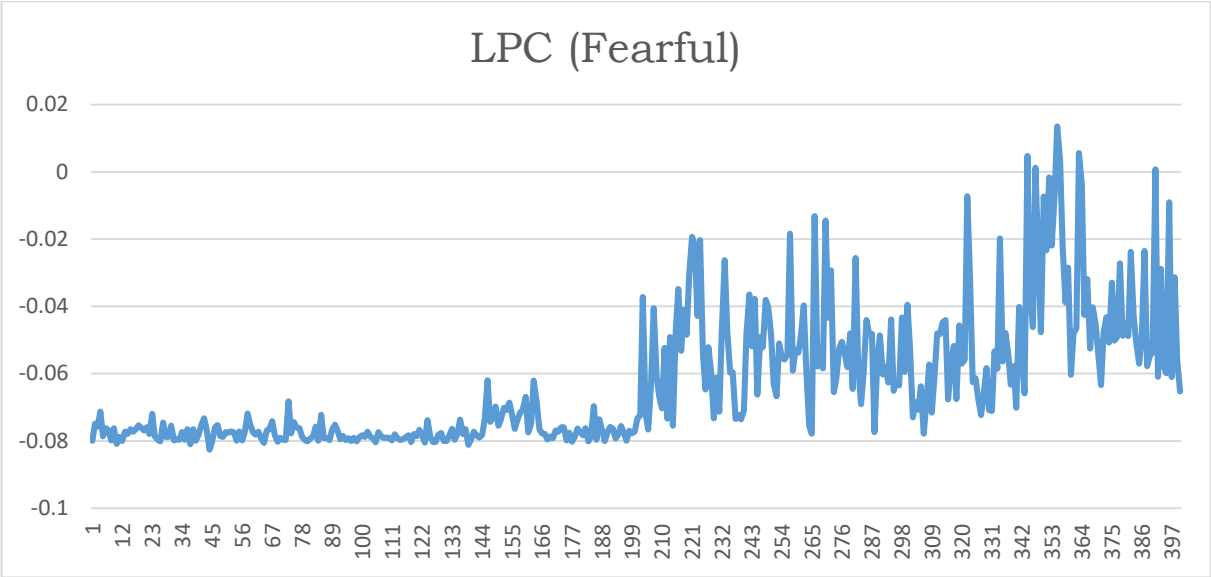

Fig 5.1.2.43: Plotting LPC for 'Fearful' Emotion Label

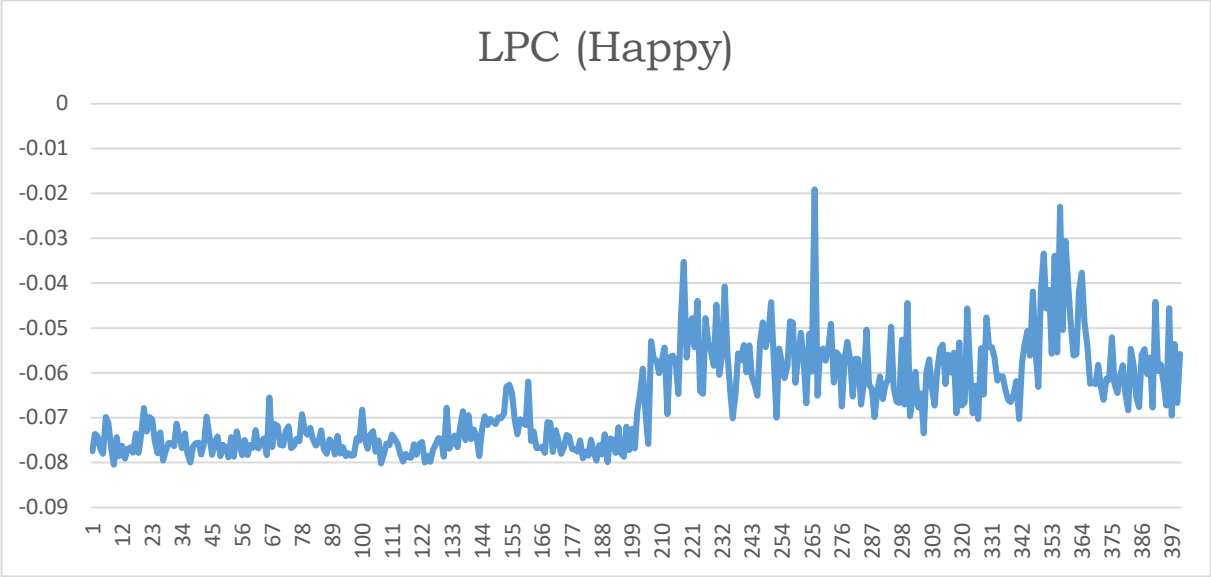

Fig 5.1.2.44: Plotting LPC for 'Happy' Emotion Label

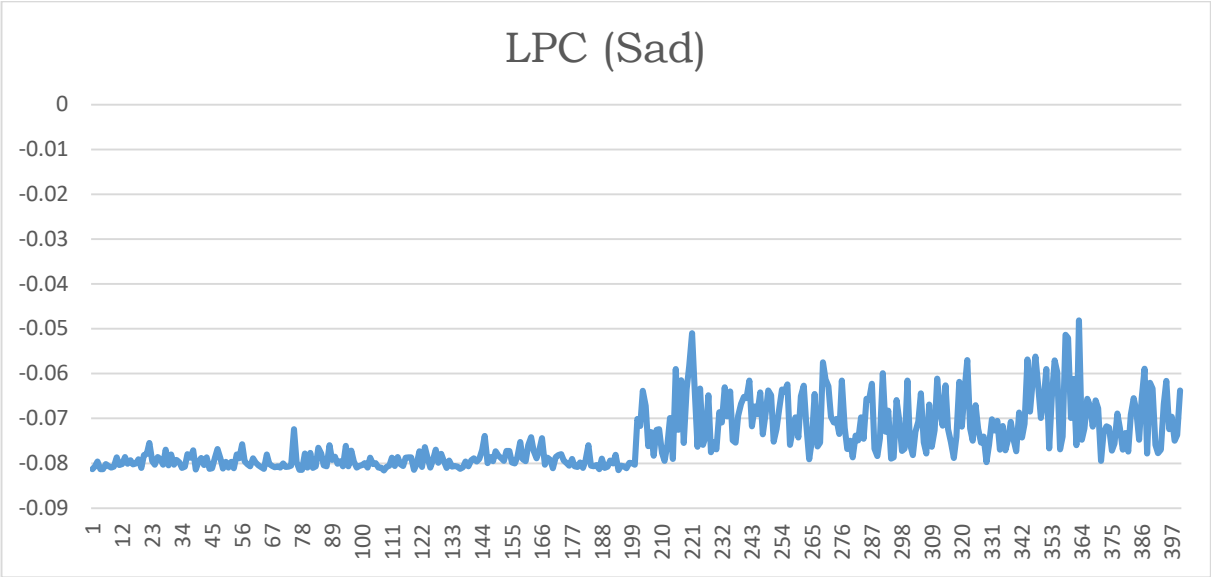

Fig 5.1.2.45: Plotting LPC for 'Sad' Emotion Label

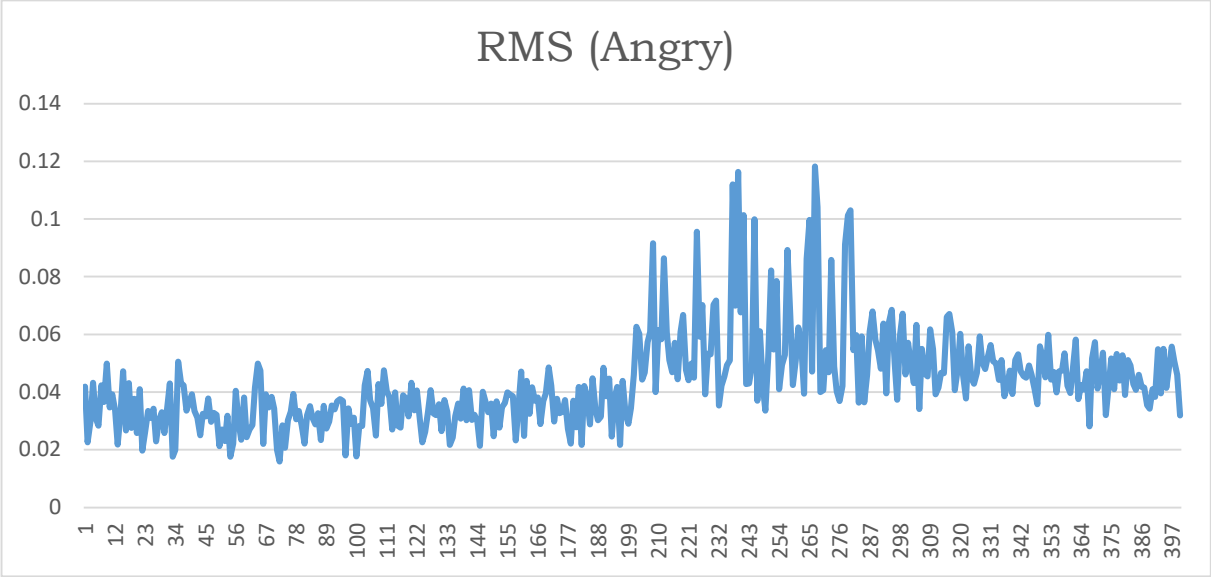

Fig 5.1.2.46: Plotting RMS for 'Angry' Emotion Label

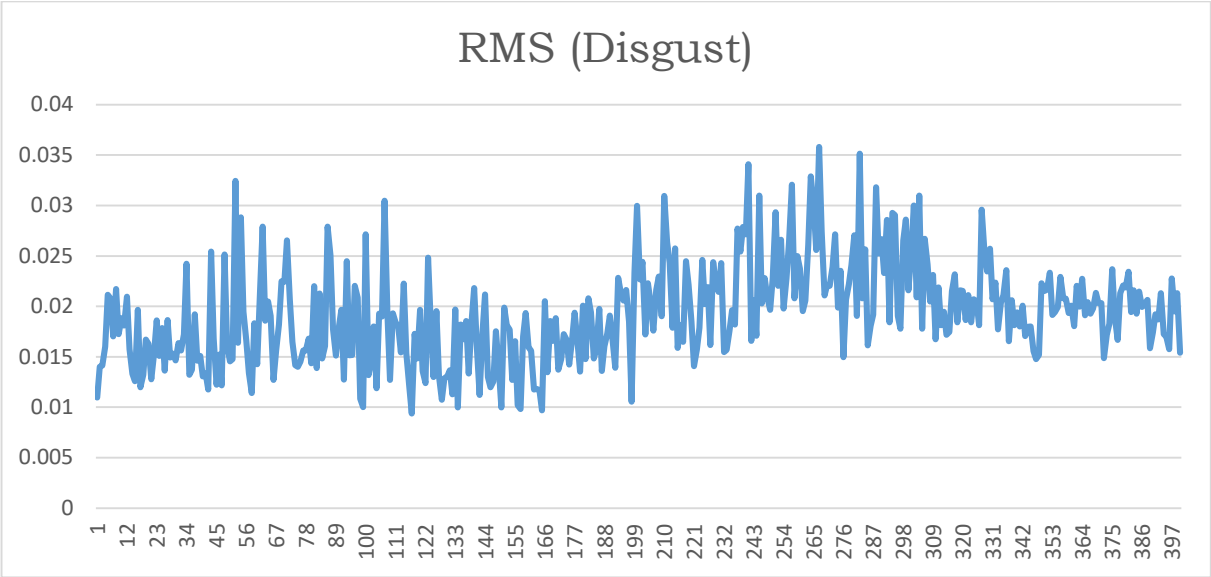

Fig 5.1.2.47: Plotting RMS for 'Disgust' Emotion Label

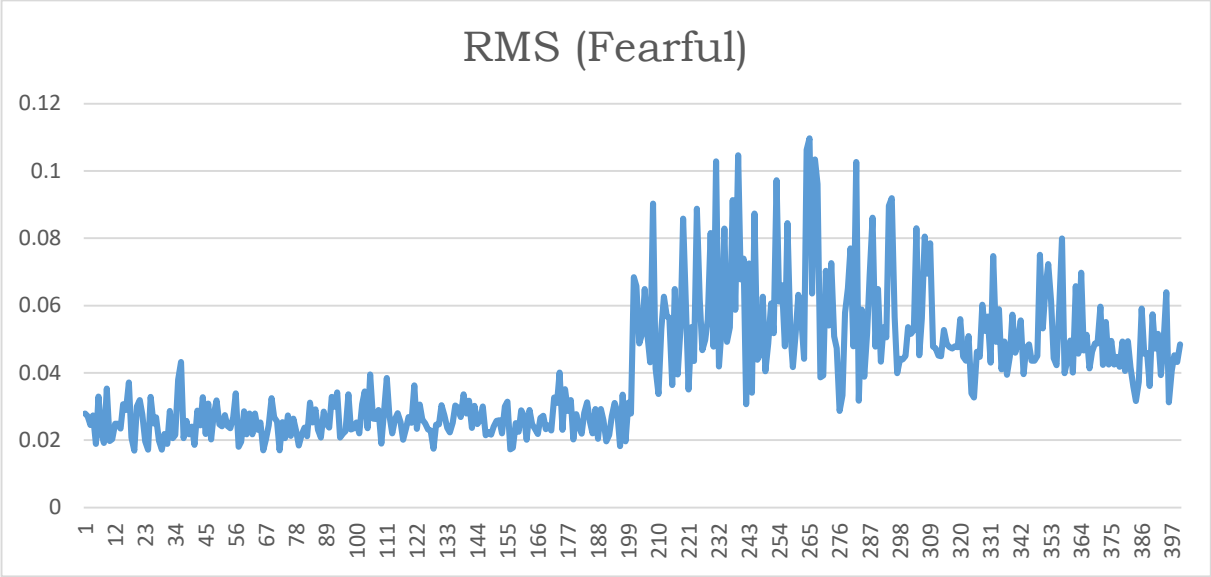

Fig 5.1.2.48: Plotting RMS for 'Fearful' Emotion Label

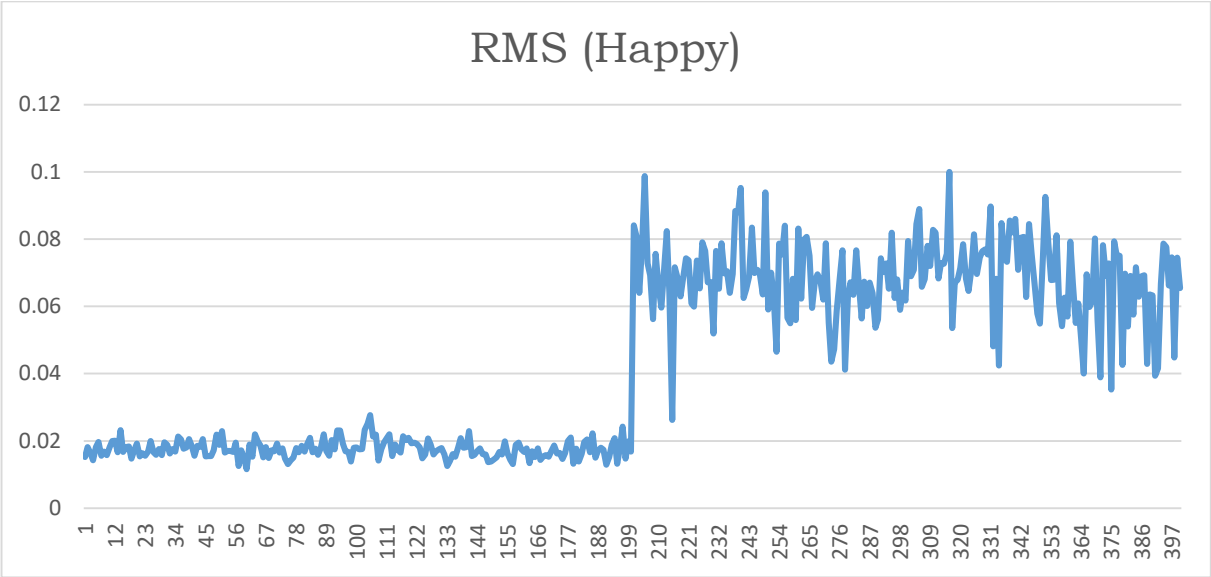

Fig 5.1.2.49: Plotting RMS for 'Happy' Emotion Label

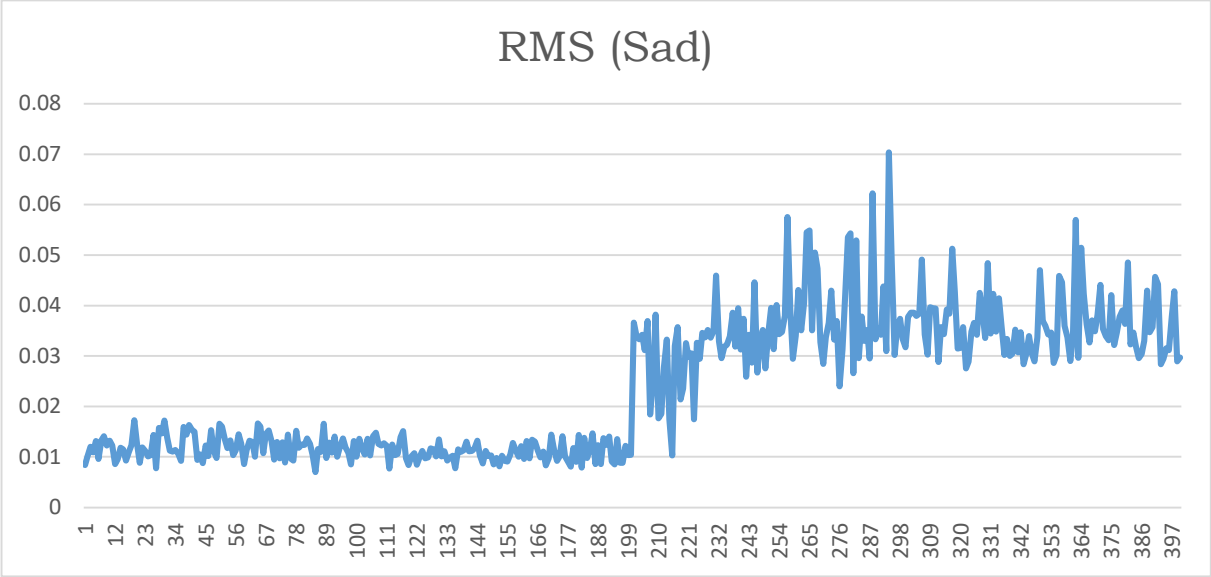

Fig 5.1.2.50: Plotting RMS for 'Sad' Emotion Label

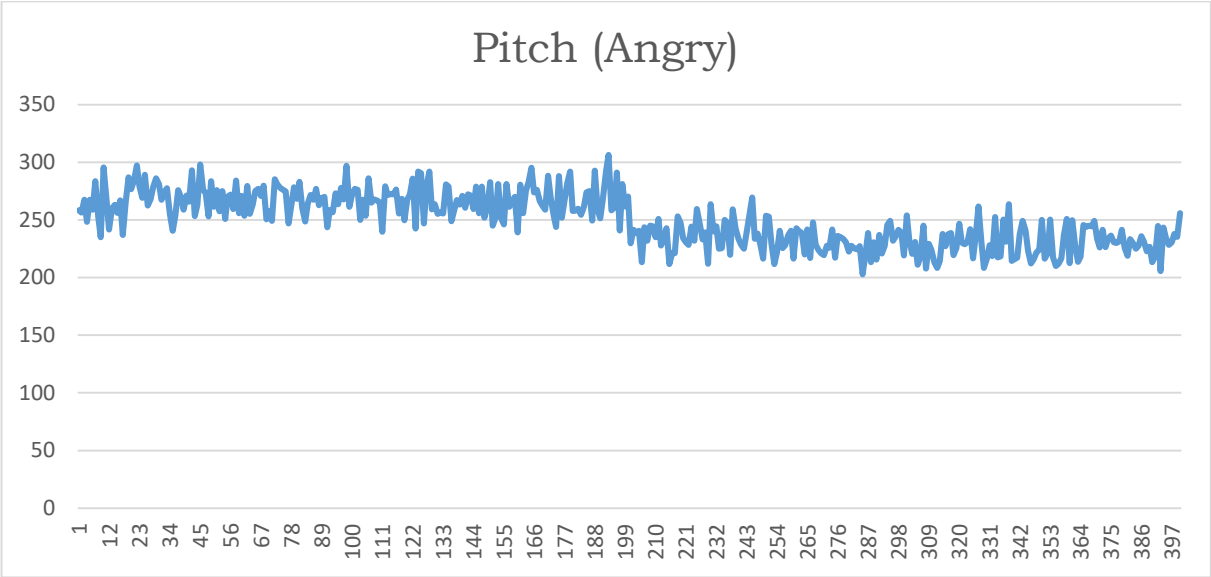

Fig 5.1.2.51: Plotting Pitch for 'Angry' Emotion Label

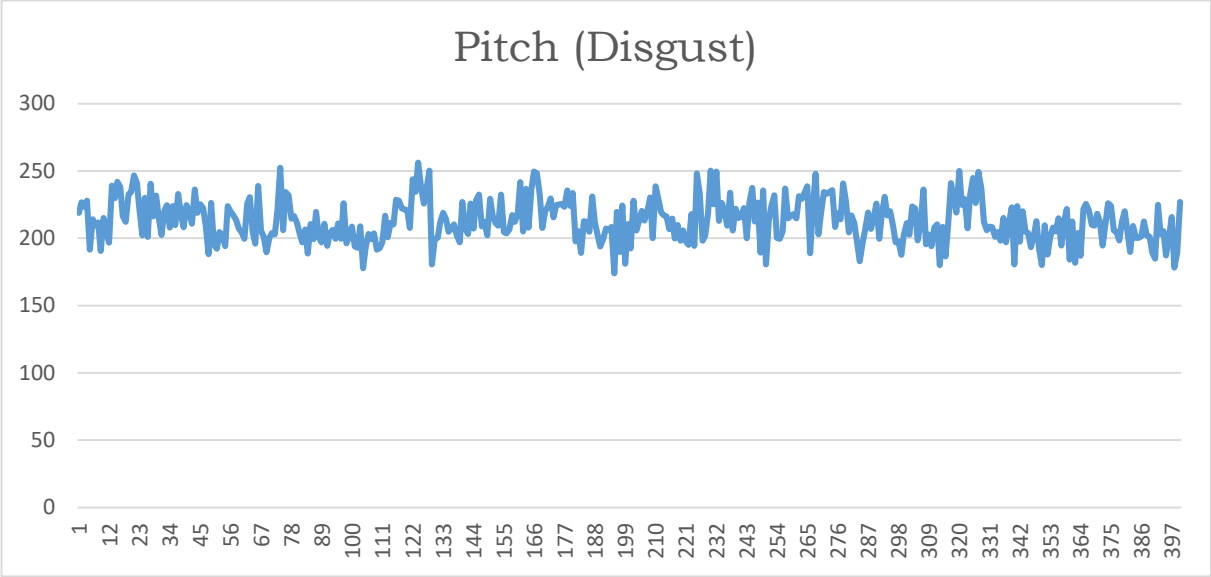

Fig 5.1.2.52: Plotting Pitch for 'Disgust' Emotion Label

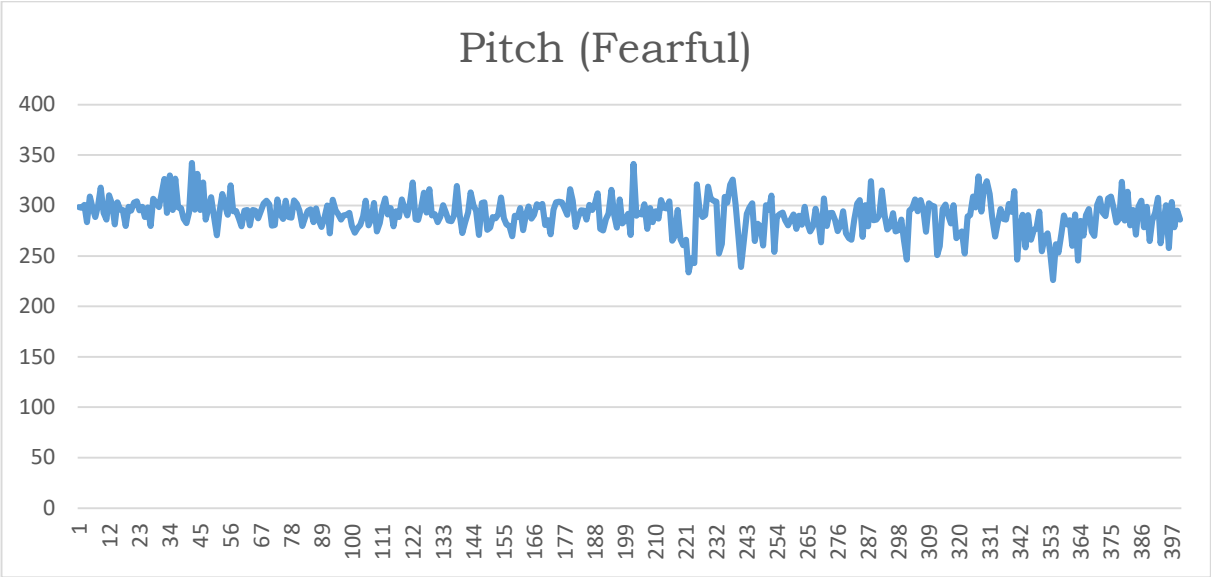

Fig 5.1.2.53: Plotting Pitch for 'Fearful' Emotion Label

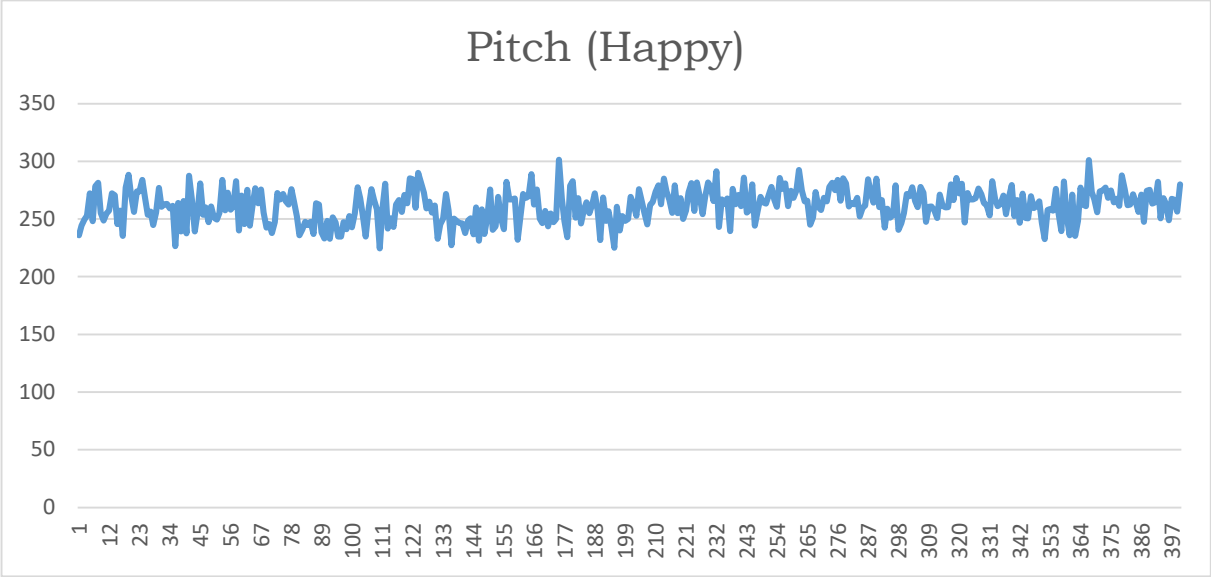

Fig 5.1.2.54: Plotting Pitch for 'Happy' Emotion Label

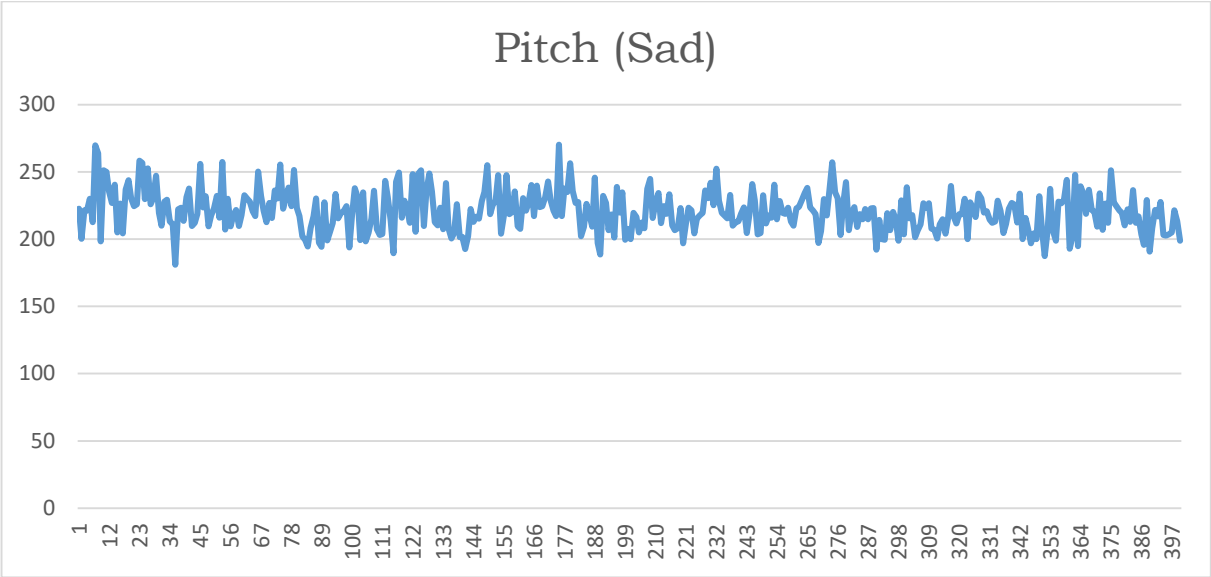

Fig 5.1.2.55: Plotting Pitch for 'Sad' Emotion Label

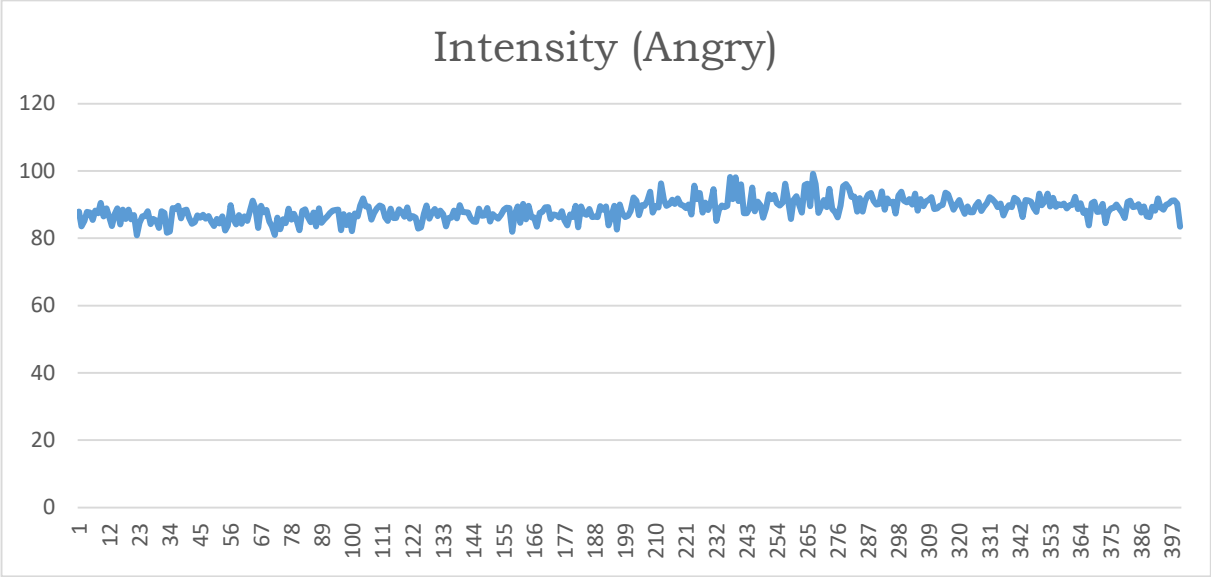

Fig 5.1.2.56: Plotting Intensity for 'Angry' Emotion Label

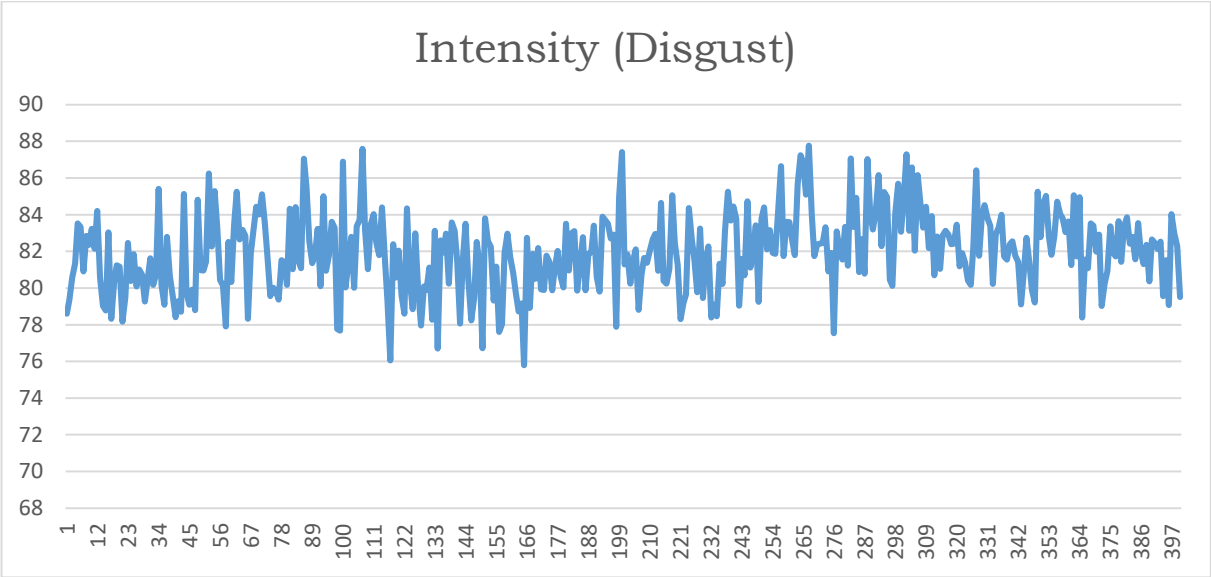

Fig 5.1.2.57: Plotting Intensity for 'Disgust' Emotion Label

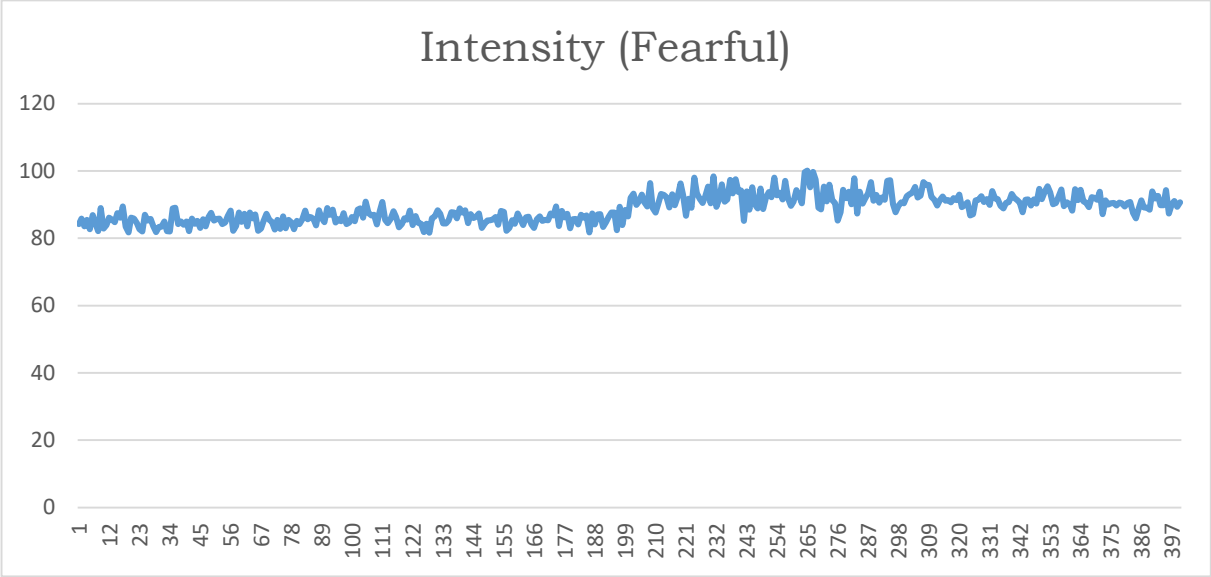

Fig 5.1.2.58: Plotting Intensity for 'Fearful' Emotion Label

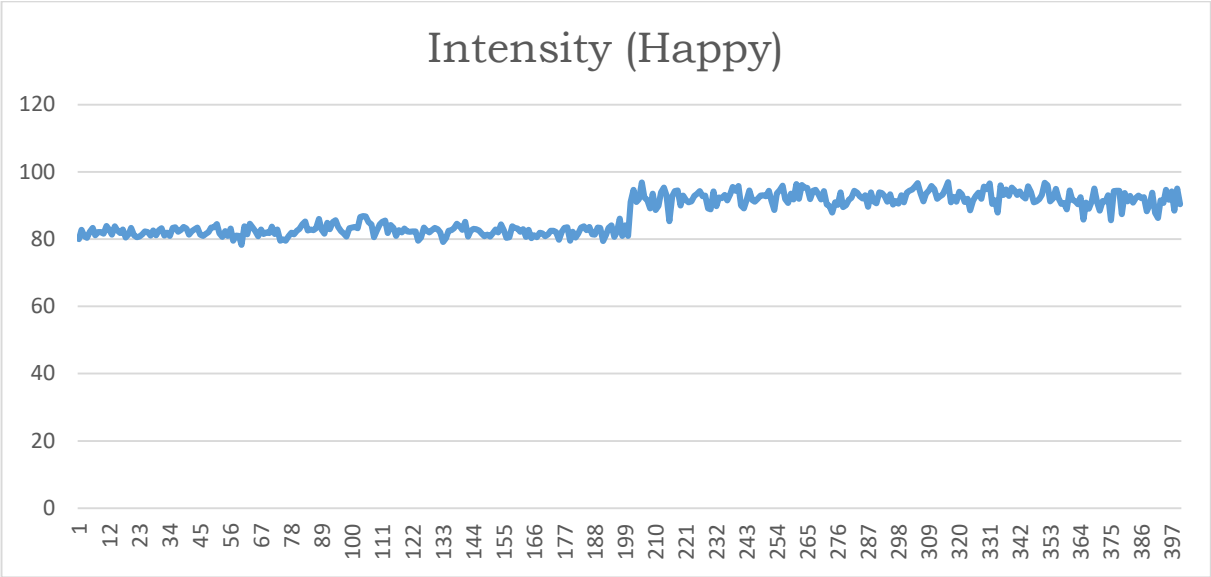

Fig 5.1.2.59: Plotting Intensity for 'Happy' Emotion Label

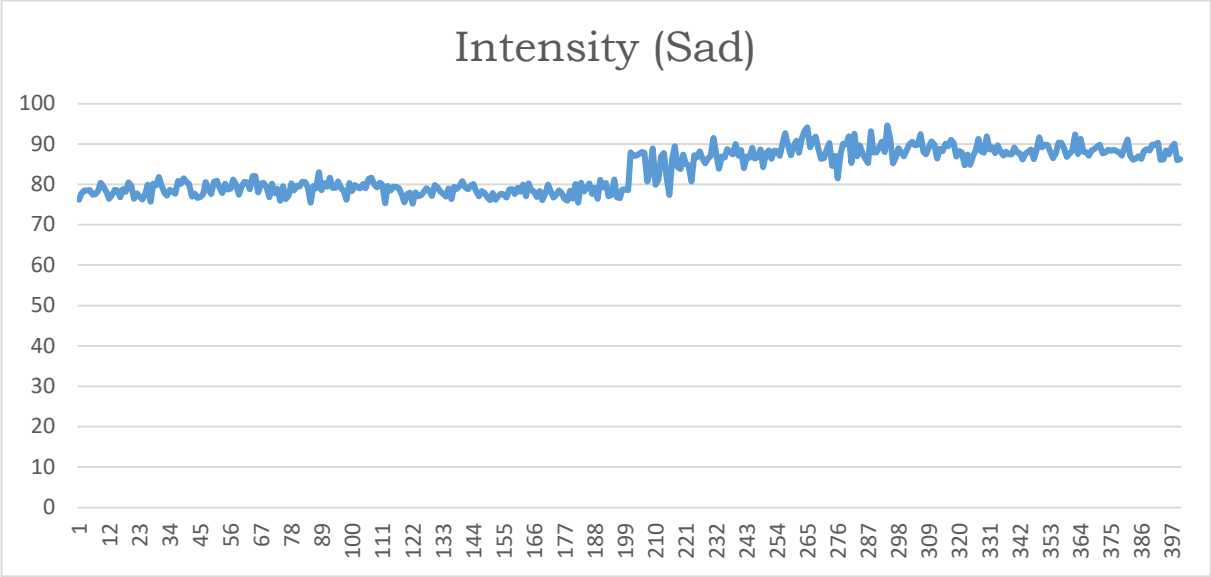

Fig 5.1.2.60: Plotting Intensity for 'Sad' Emotion Label

Plots of SAVEE Dataset

Fig 5.1.2.61 – Fig 5.1.2.105 showcase plots of the variations in emotional expressions across all features within the SAVEE dataset. These visual representations provide insights into how different features respond to various emotions, offering a deeper understanding of the dataset's emotional dynamics.

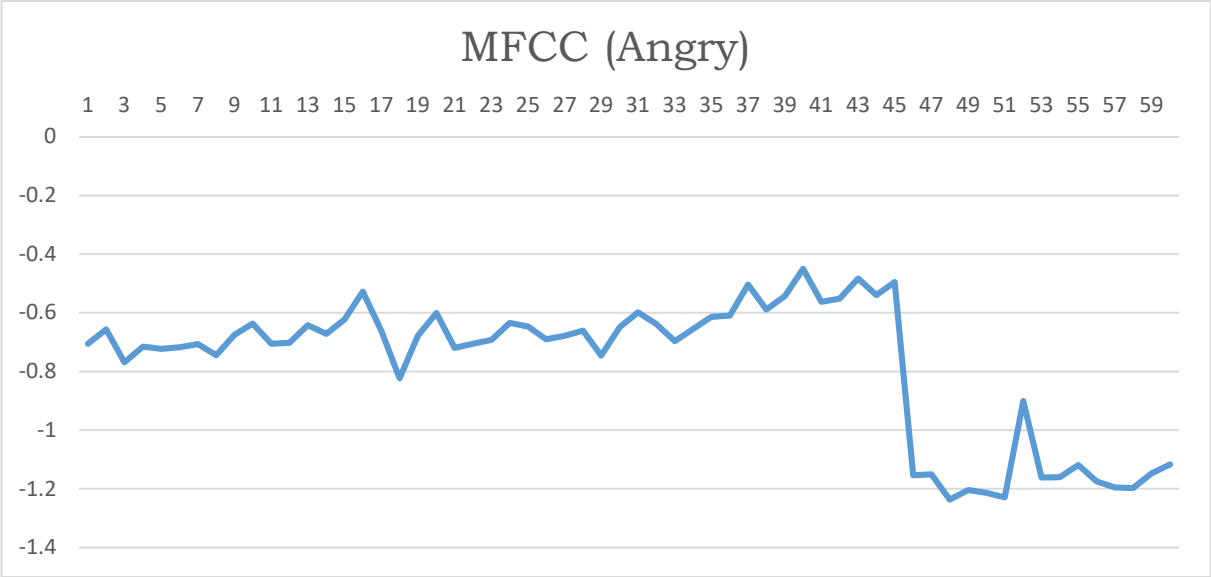

Fig 5.1.2.61: Plotting MFCC for 'Angry' Emotion Label

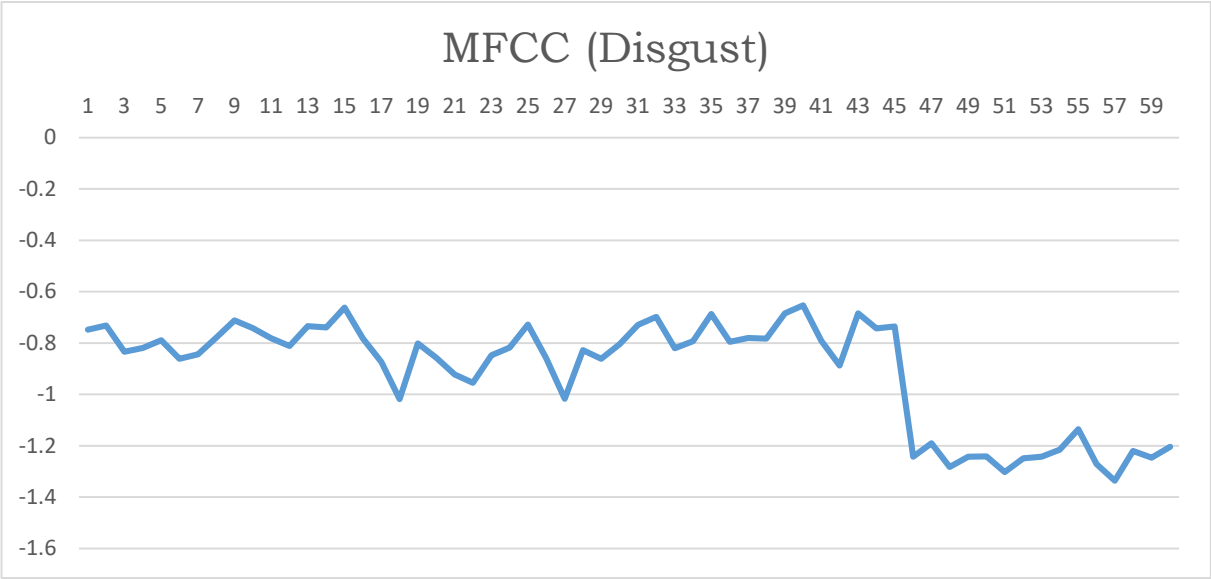

Fig 5.1.2.62: Plotting MFCC for 'Disgust' Emotion Label

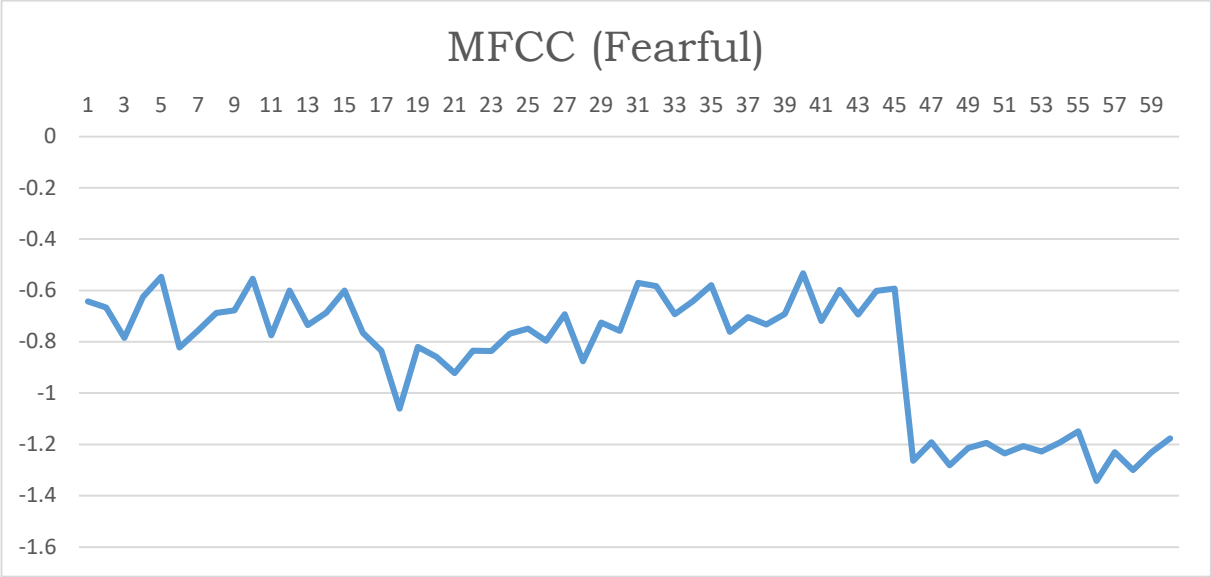

Fig 5.1.2.63: Plotting MFCC for 'Fearful' Emotion Label

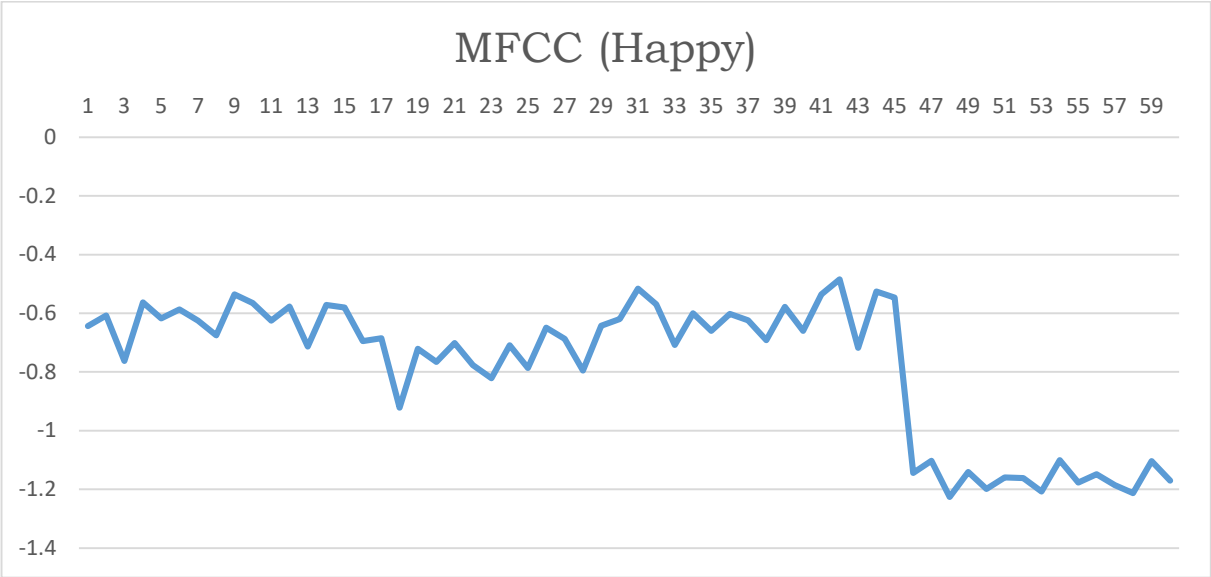

Fig 5.1.2.64: Plotting MFCC for 'Happy' Emotion Label

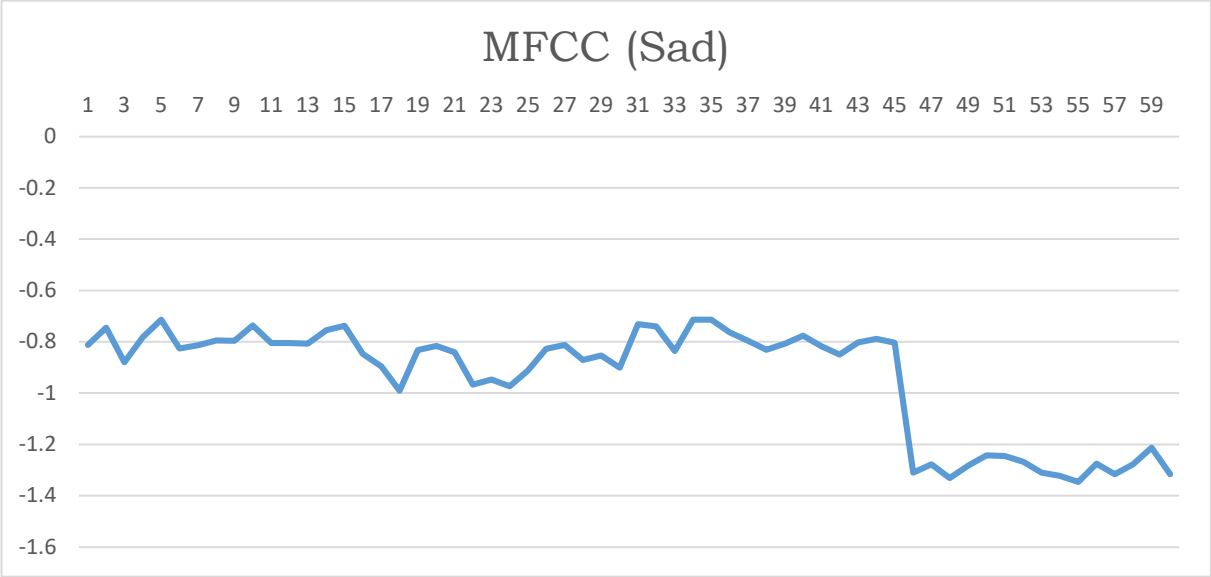

Fig 5.1.2.65: Plotting MFCC for 'Sad' Emotion Label

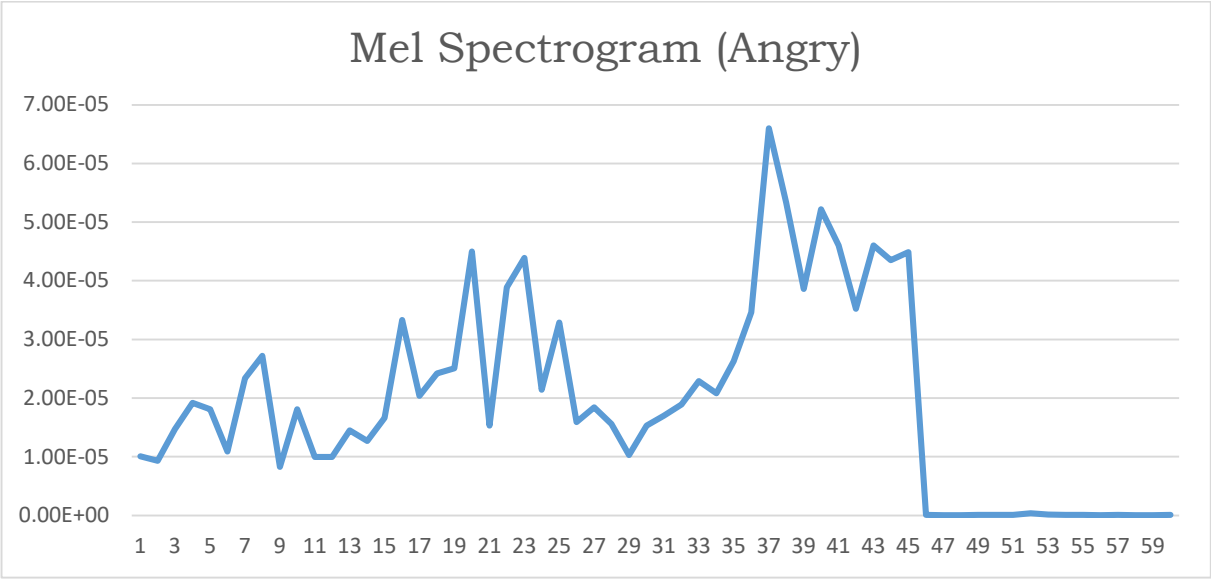

Fig 5.1.2.66: Plotting Mel Spectrogram for 'Angry' Emotion Label

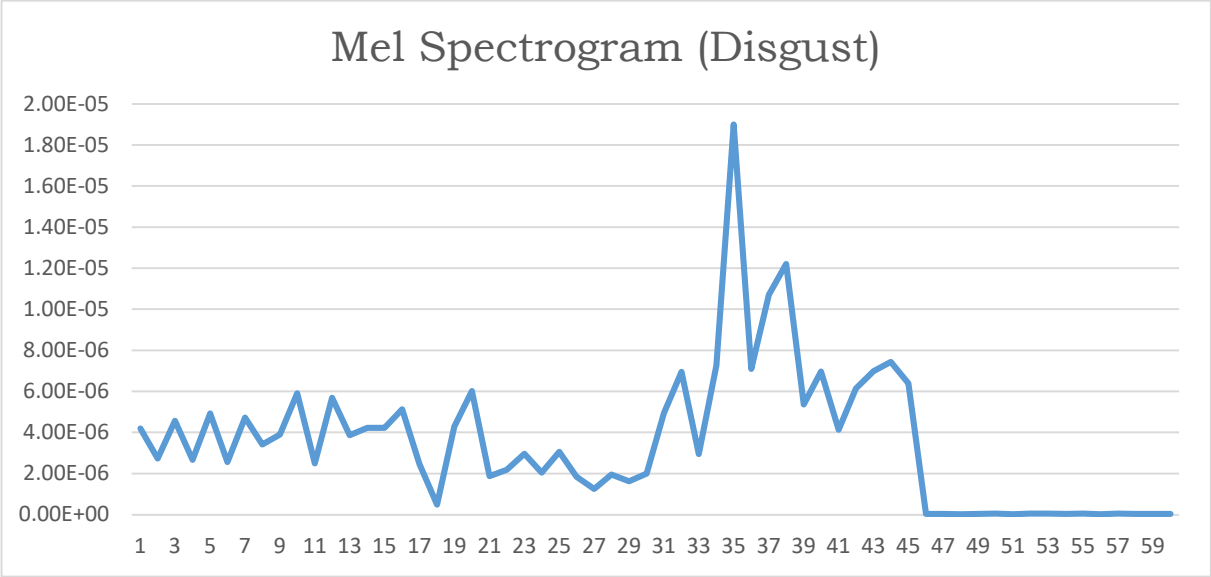

Fig 5.1.2.67: Plotting Mel Spectrogram for 'Disgust' Emotion Label

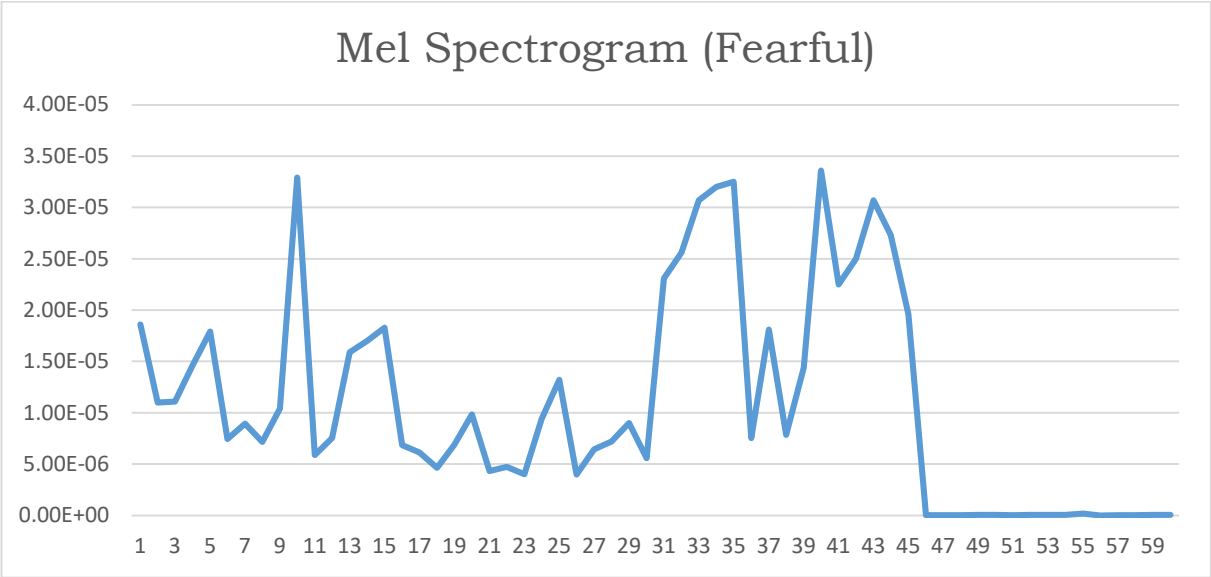

Fig 5.1.2.68: Plotting Mel Spectrogram for 'Fearful' Emotion Label

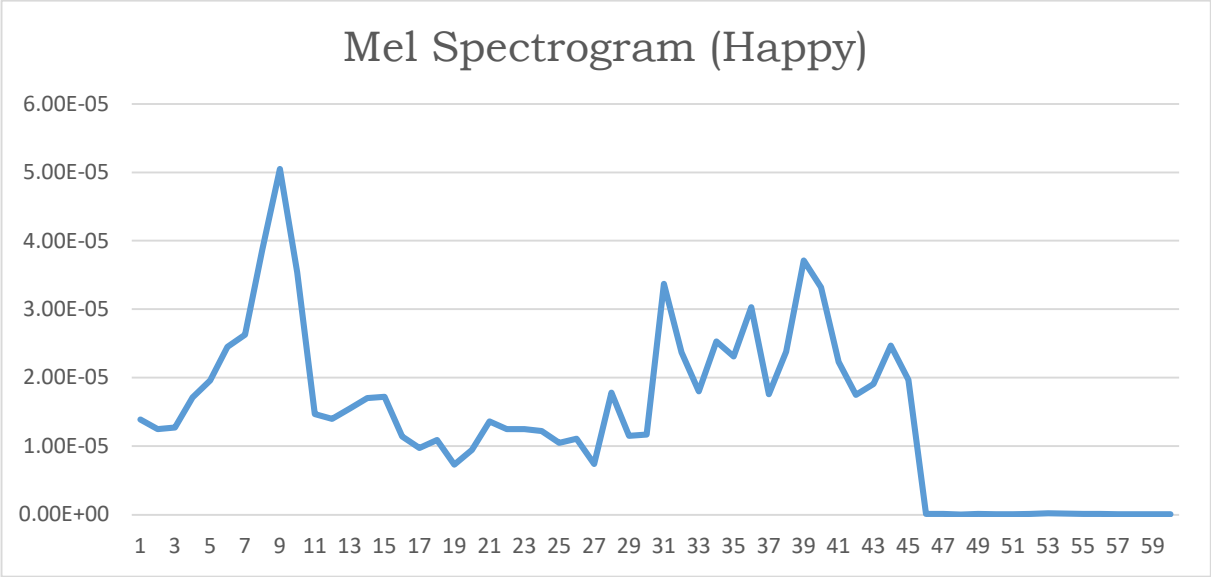

Fig 5.1.2.69: Plotting Mel Spectrogram for 'Happy' Emotion Label

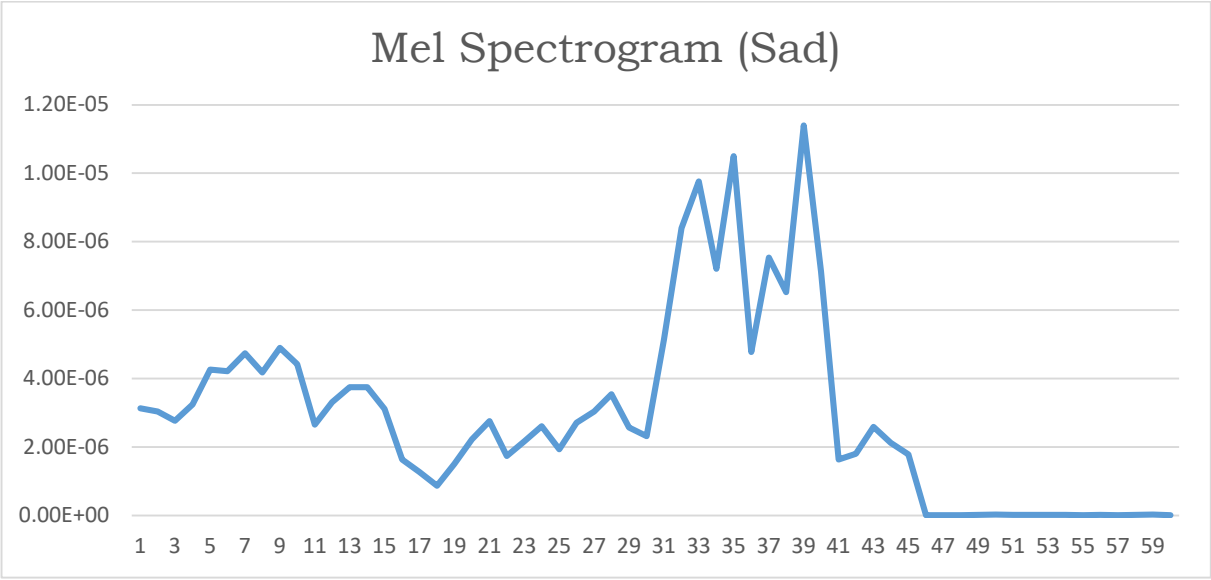

Fig 5.1.2.70: Plotting Mel Spectrogram for 'Sad' Emotion Label

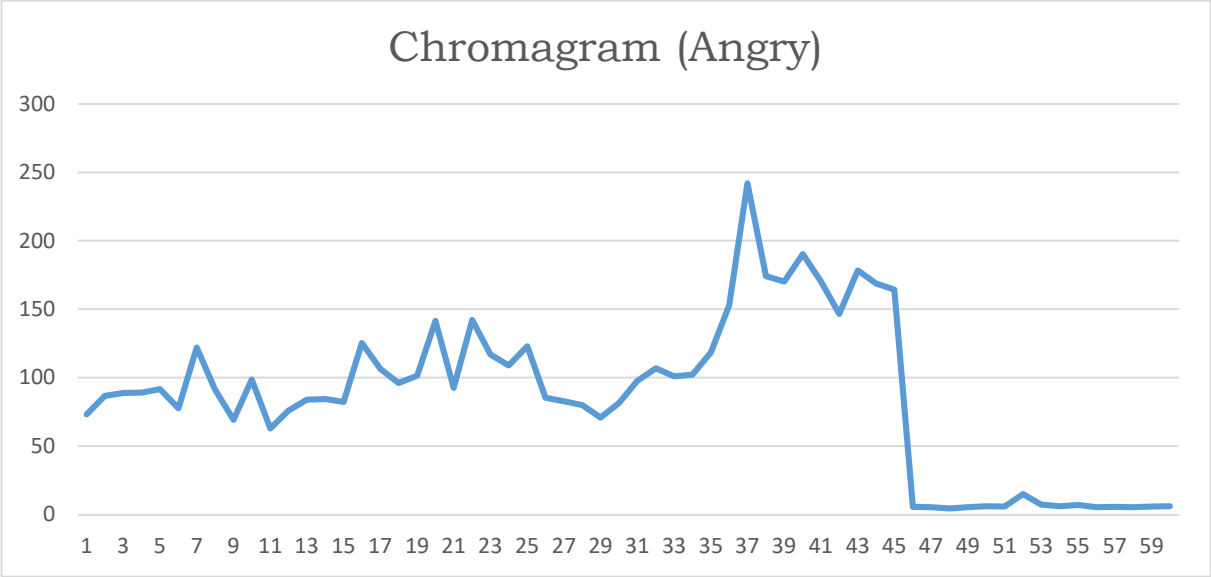

Fig 5.1.2.71: Plotting Chromagram for 'Angry' Emotion Label

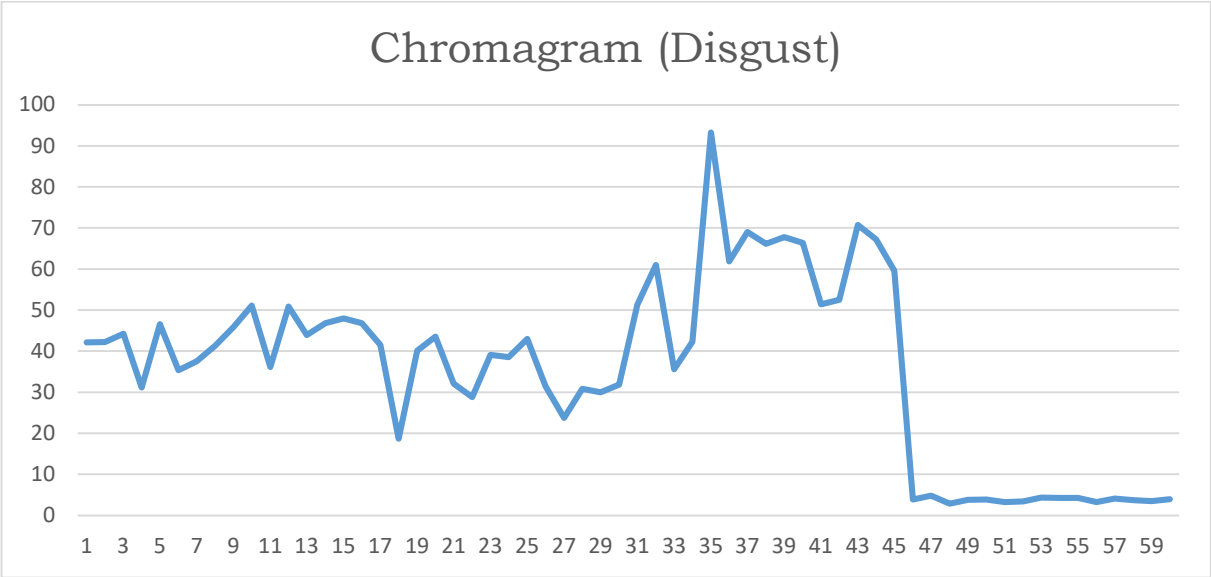

Fig 5.1.2.72: Plotting Chromagram for 'Disgust' Emotion Label

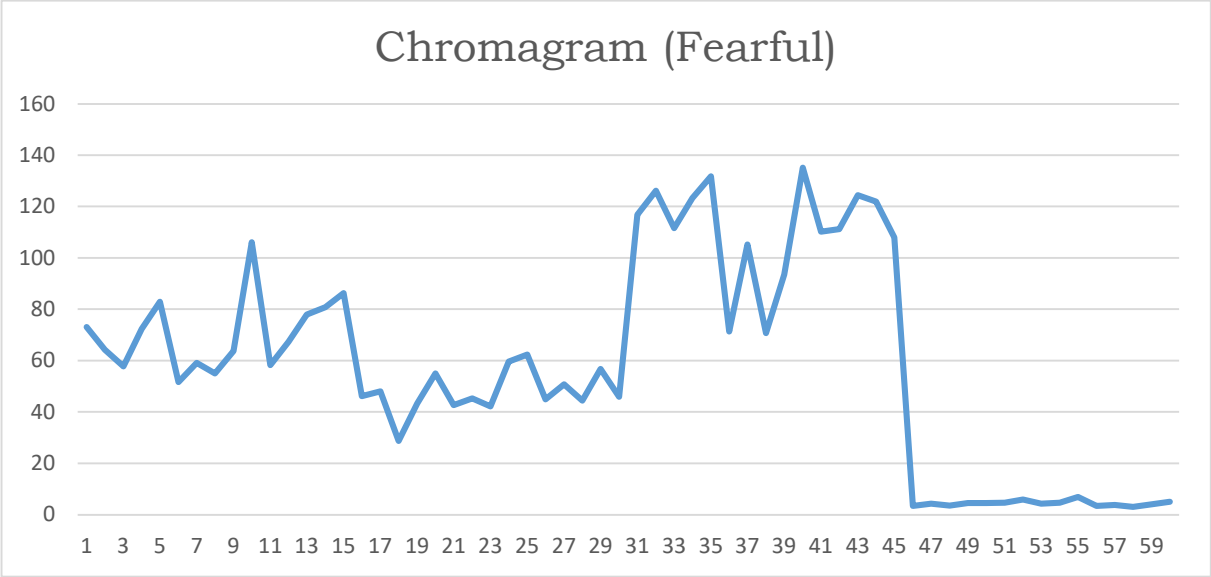

Fig 5.1.2.73: Plotting Chromagram for 'Fearful' Emotion Label

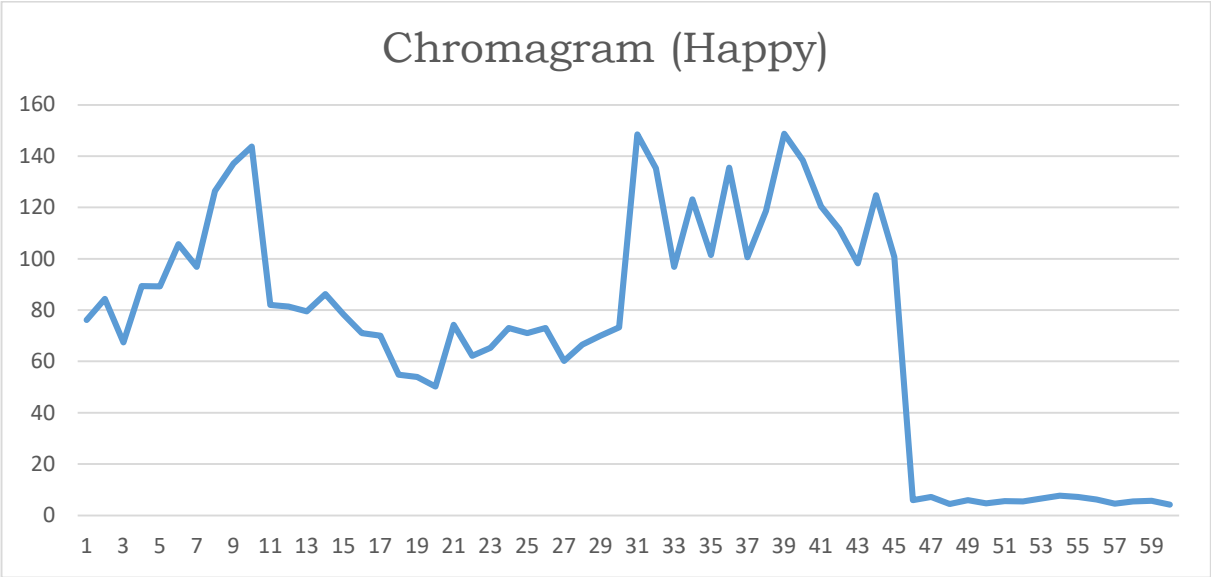

Fig 5.1.2.74: Plotting Chromagram for 'Happy' Emotion Label

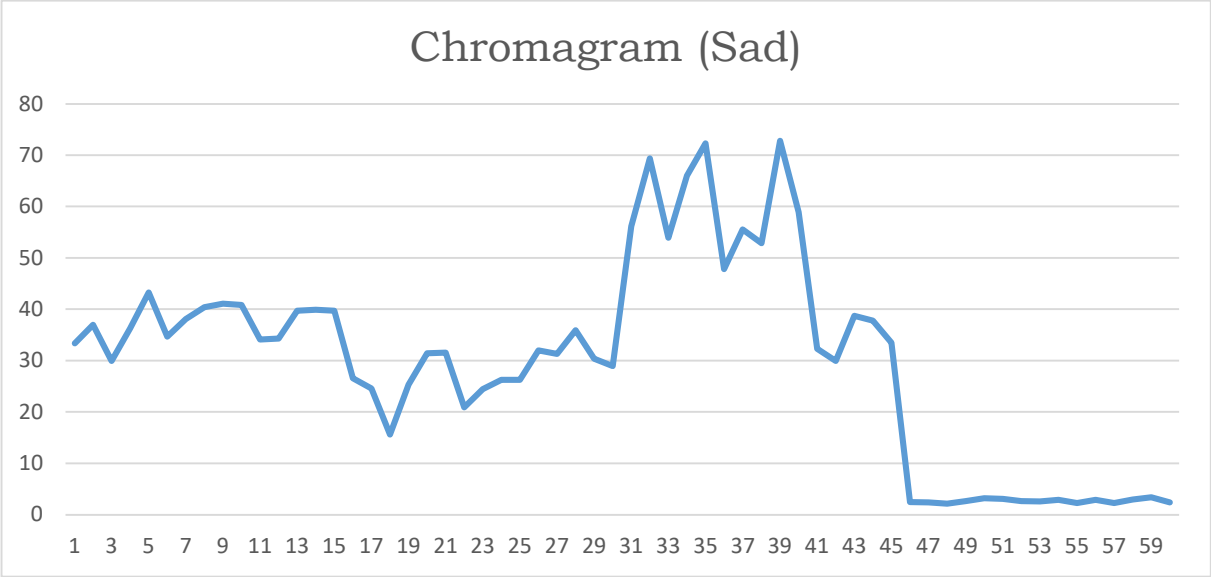

Fig 5.1.2.75: Plotting Chromagram for 'Sad' Emotion Label

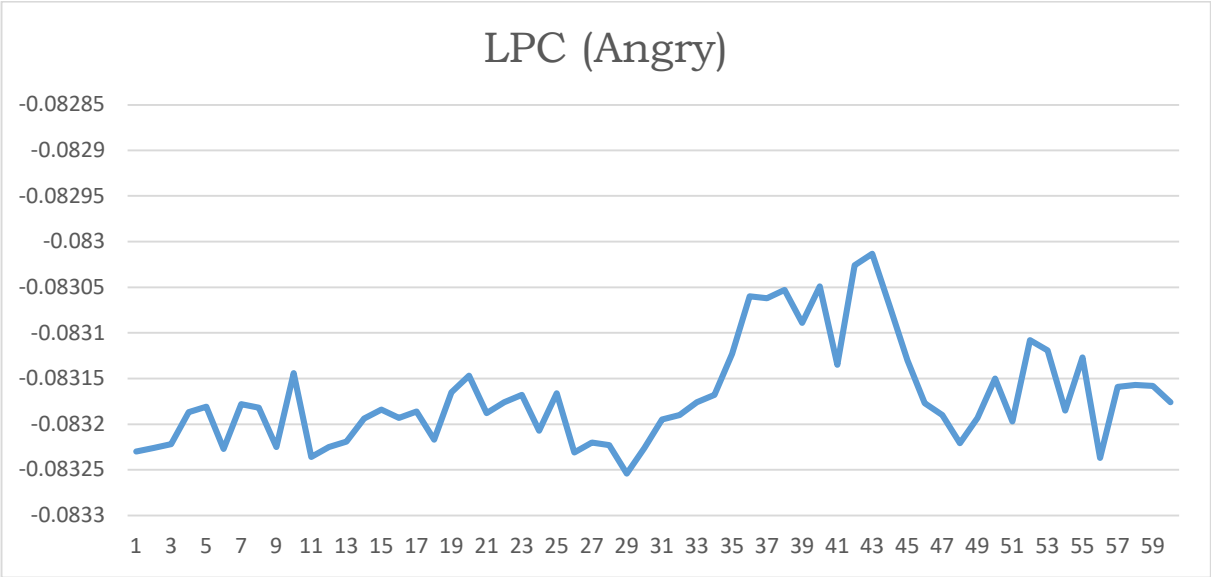

Fig 5.1.2.76: Plotting LPC for 'Angry' Emotion Label

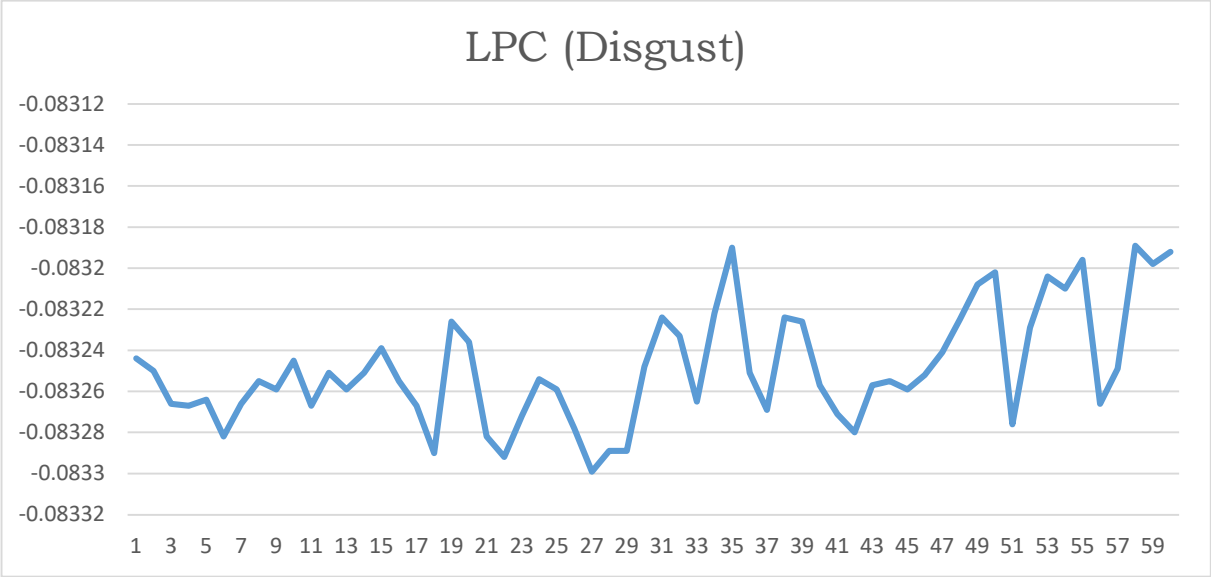

Fig 5.1.2.77: Plotting LPC for 'Disgust' Emotion Label

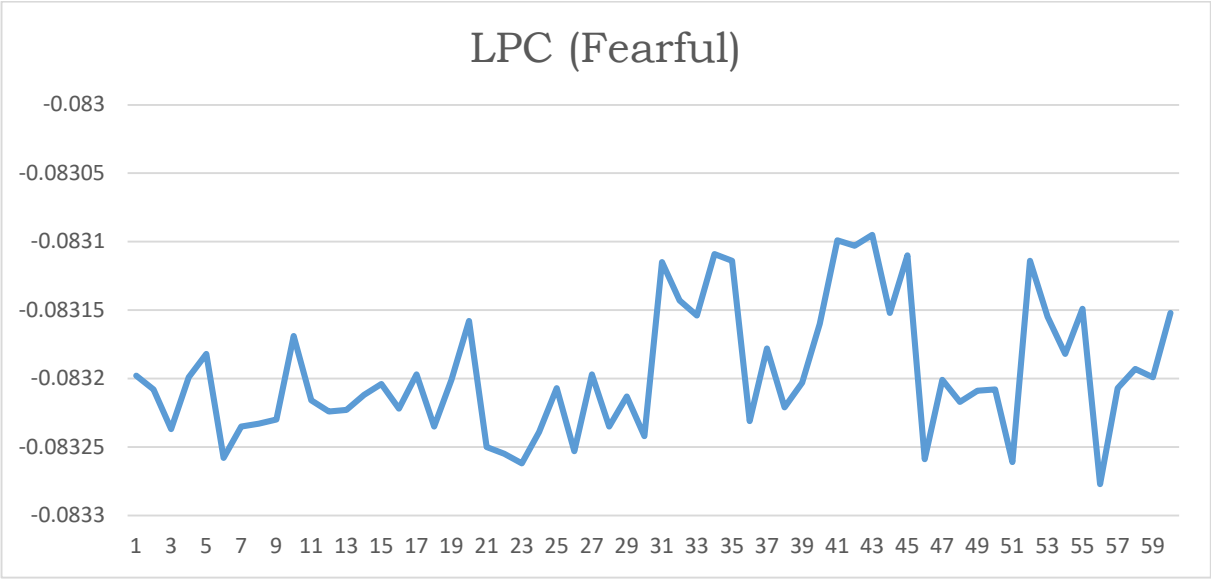

Fig 5.1.2.78: Plotting LPC for 'Fearful' Emotion Label

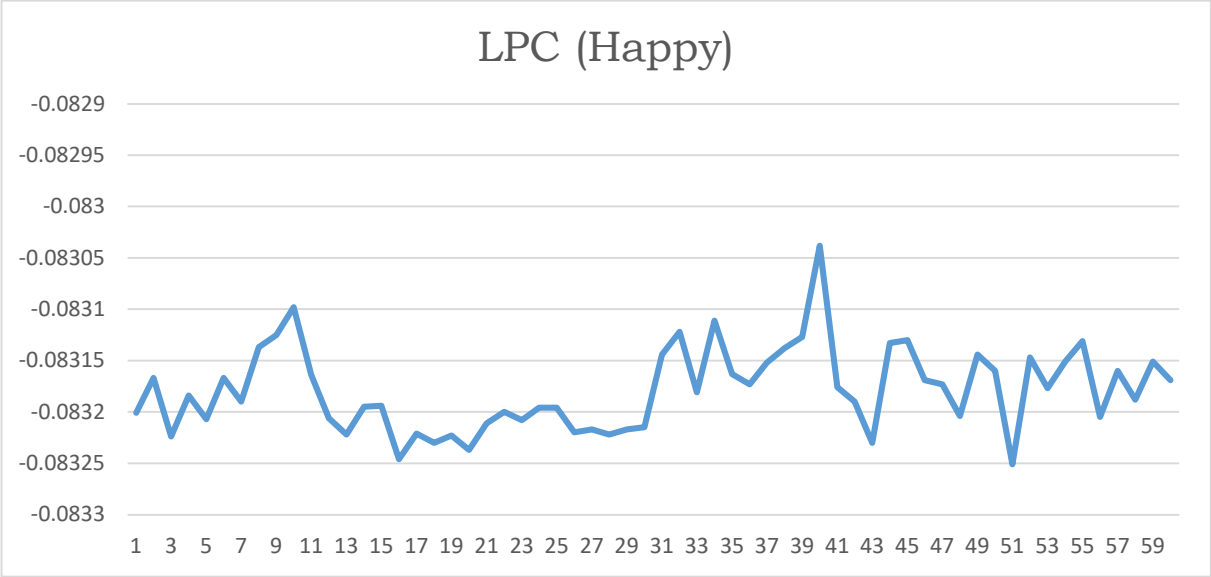

Fig 5.1.2.79: Plotting LPC for 'Happy' Emotion Label

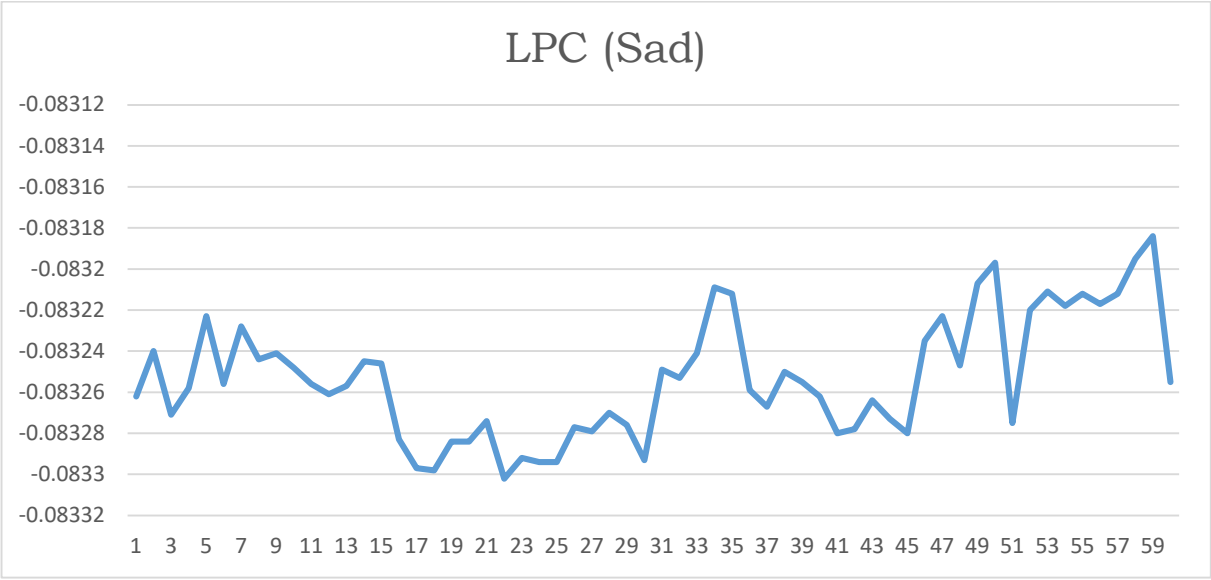

Fig 5.1.2.80: Plotting LPC for 'Sad' Emotion Label

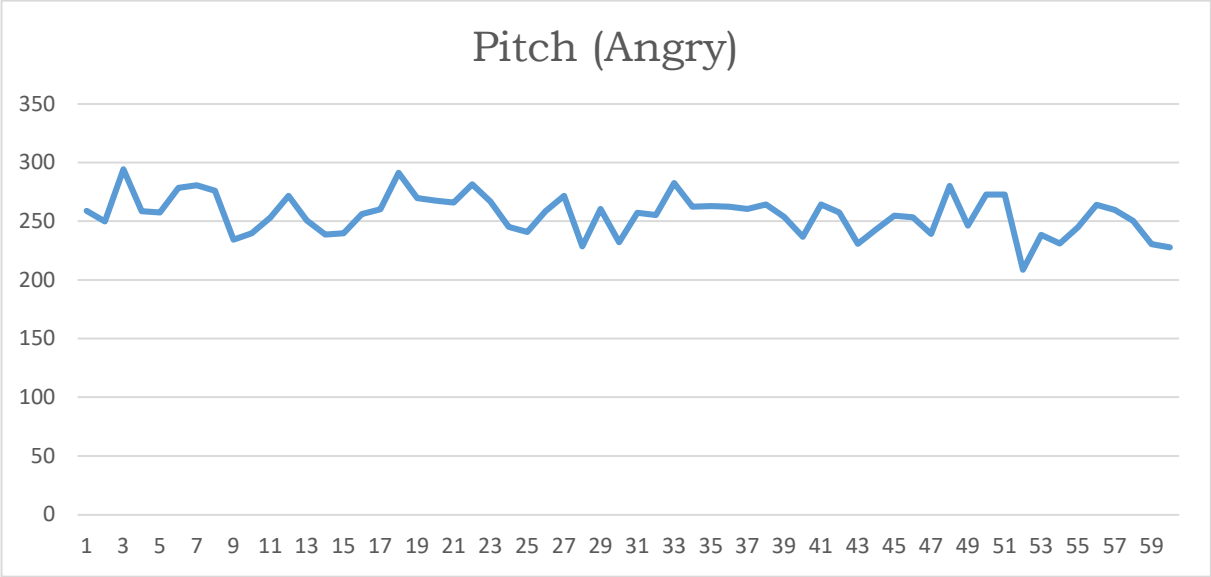

Fig 5.1.2.81: Plotting Pitch for 'Angry' Emotion Label

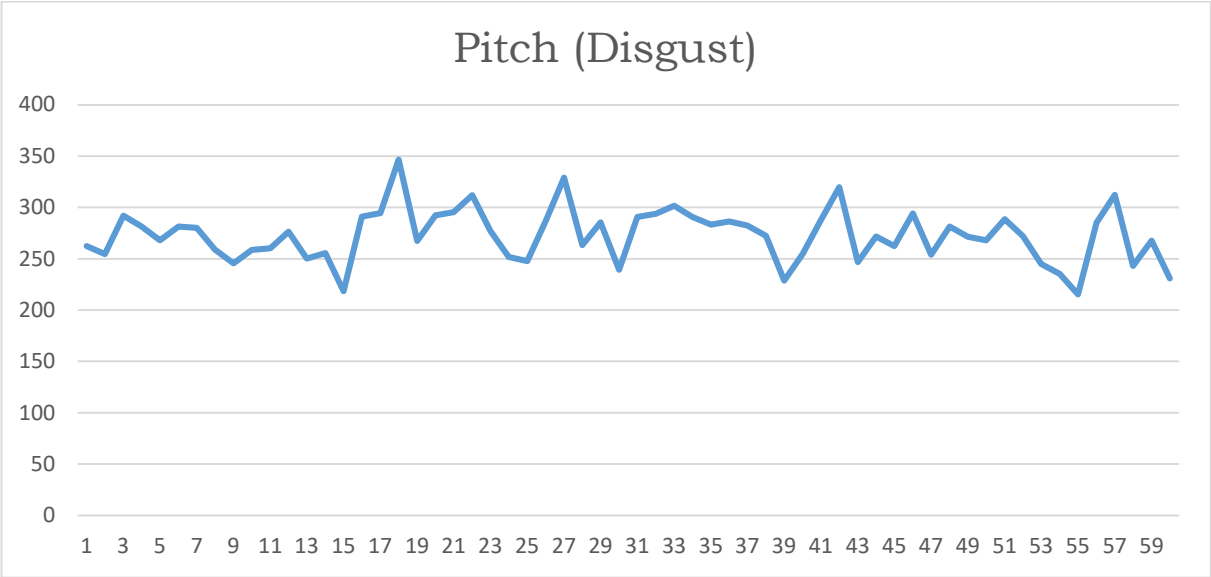

Fig 5.1.2.82: Plotting Pitch for 'Disgust' Emotion Label

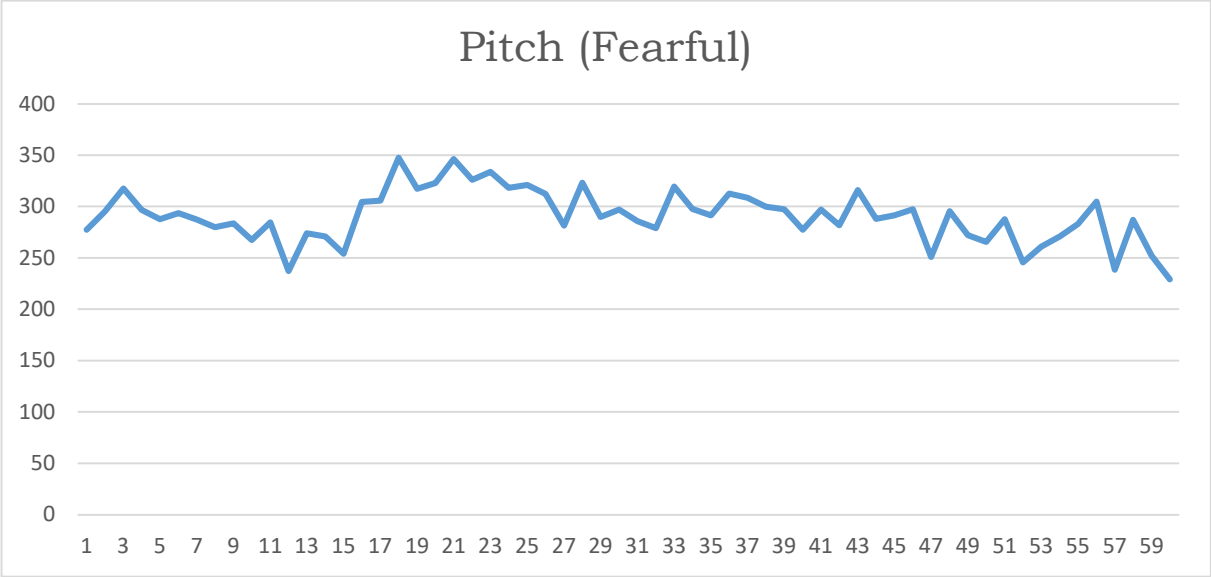

Fig 5.1.2.83: Plotting Pitch for 'Fearful' Emotion Label

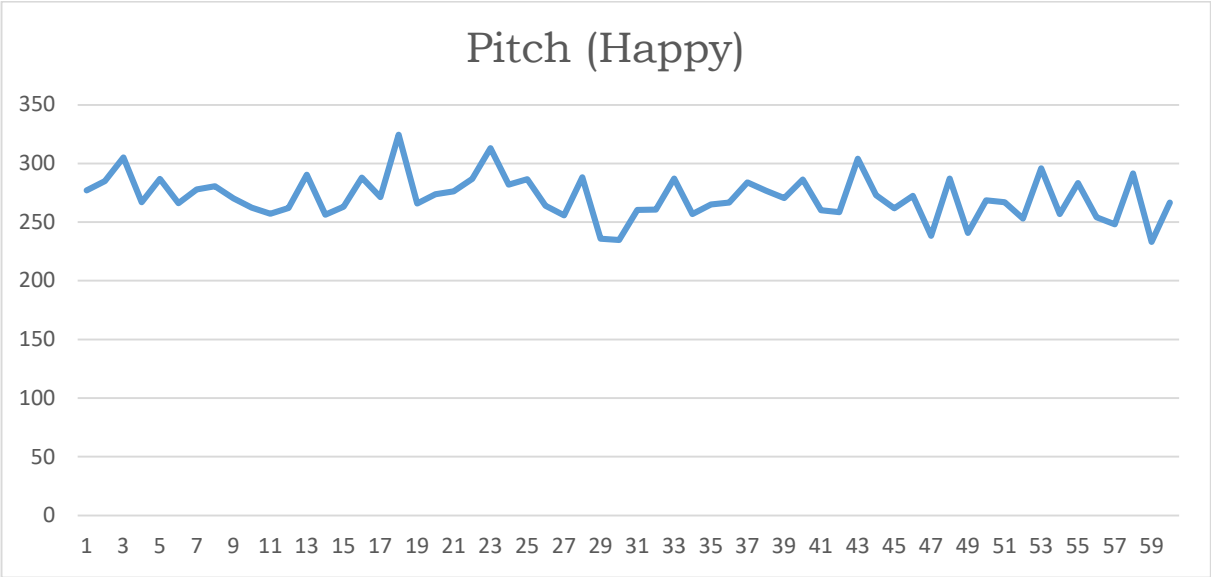

Fig 5.1.2.84: Plotting Pitch for 'Happy' Emotion Label

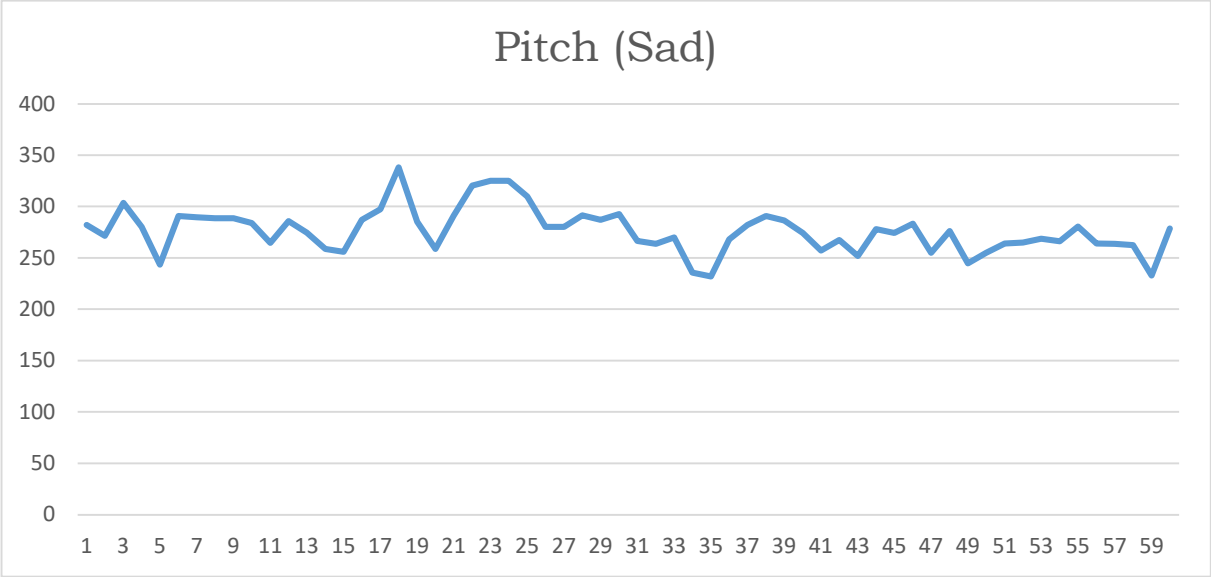

Fig 5.1.2.85: Plotting Pitch for 'Sad' Emotion Label

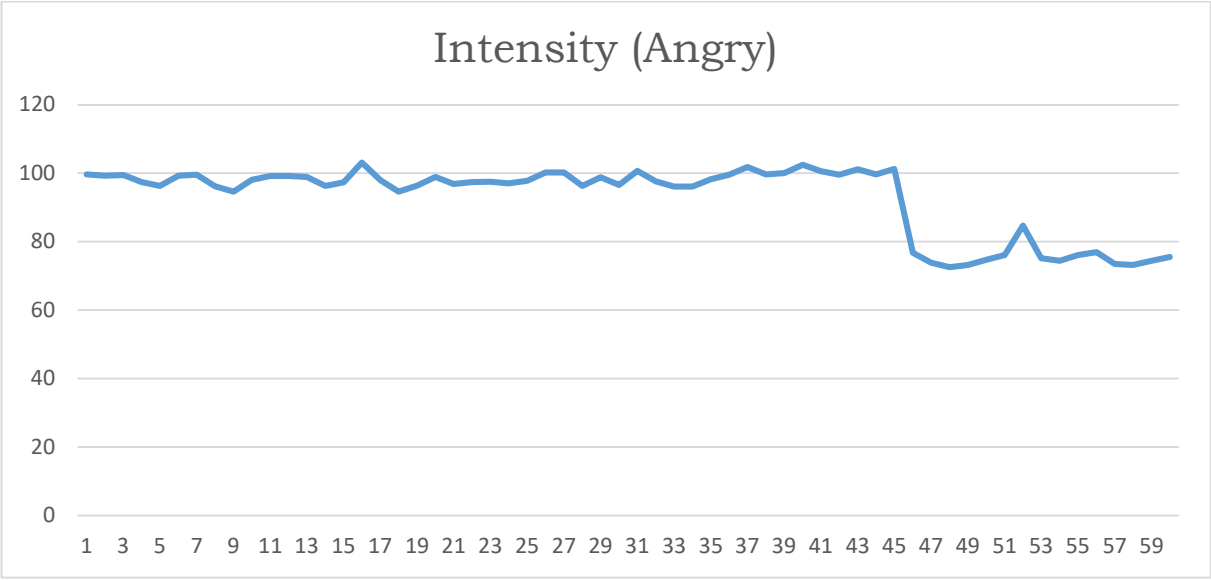

Fig 5.1.2.86: Plotting Intensity for 'Angry' Emotion Label

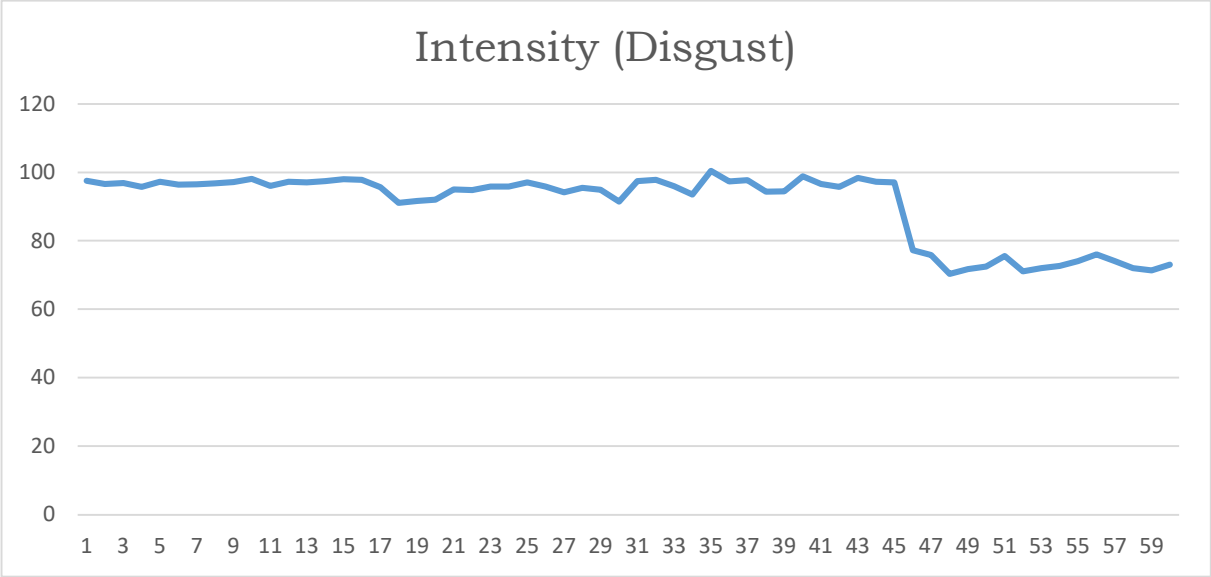

Fig 5.1.2.87: Plotting Intensity for 'Disgust' Emotion Label

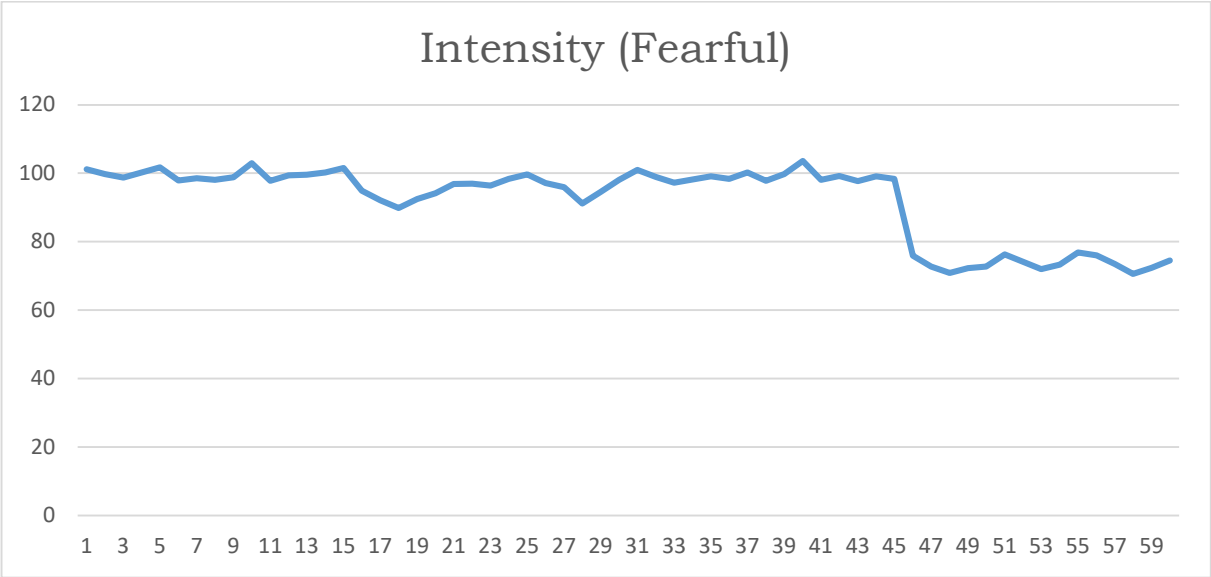

Fig 5.1.2.88: Plotting Intensity for 'Fearful' Emotion Label

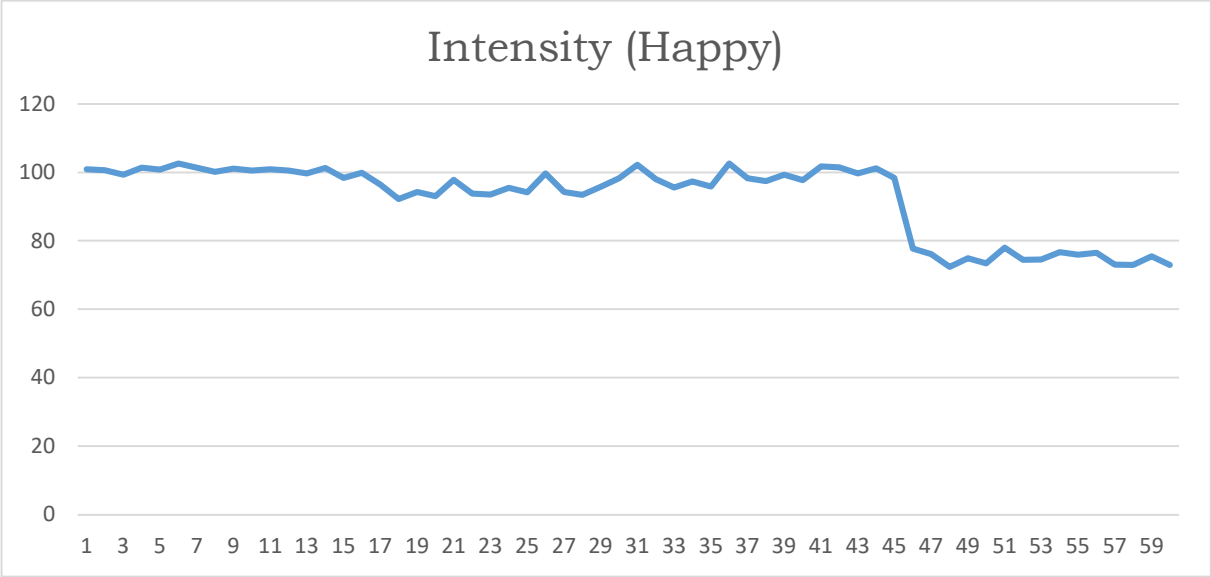

Fig 5.1.2.89: Plotting Intensity for 'Happy' Emotion Label

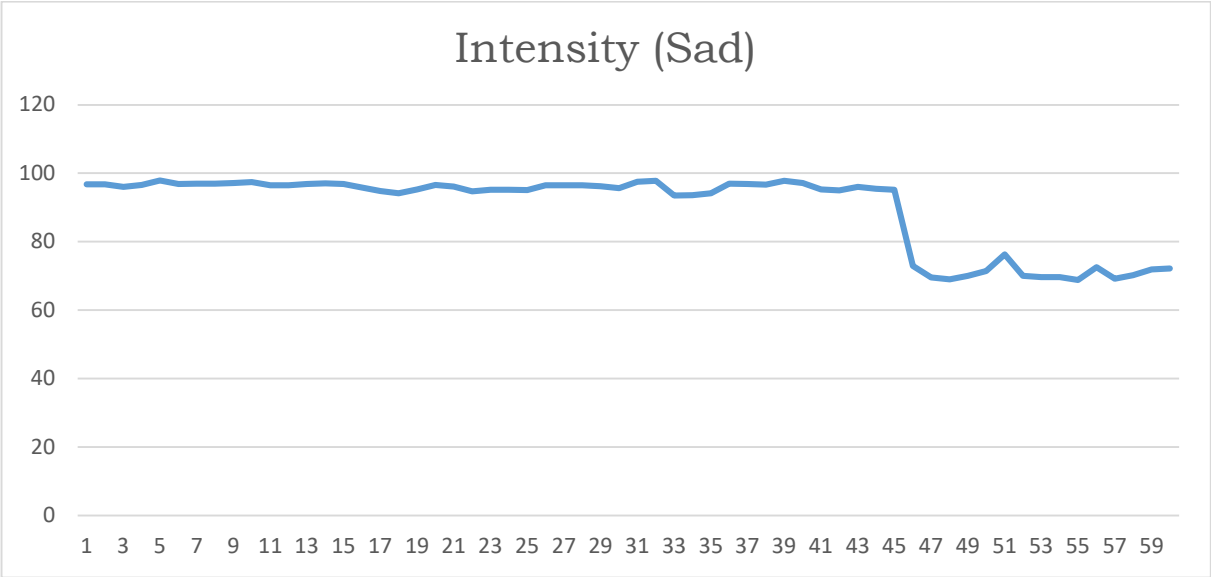

Fig 5.1.2.90: Plotting Intensity for 'Sad' Emotion Label

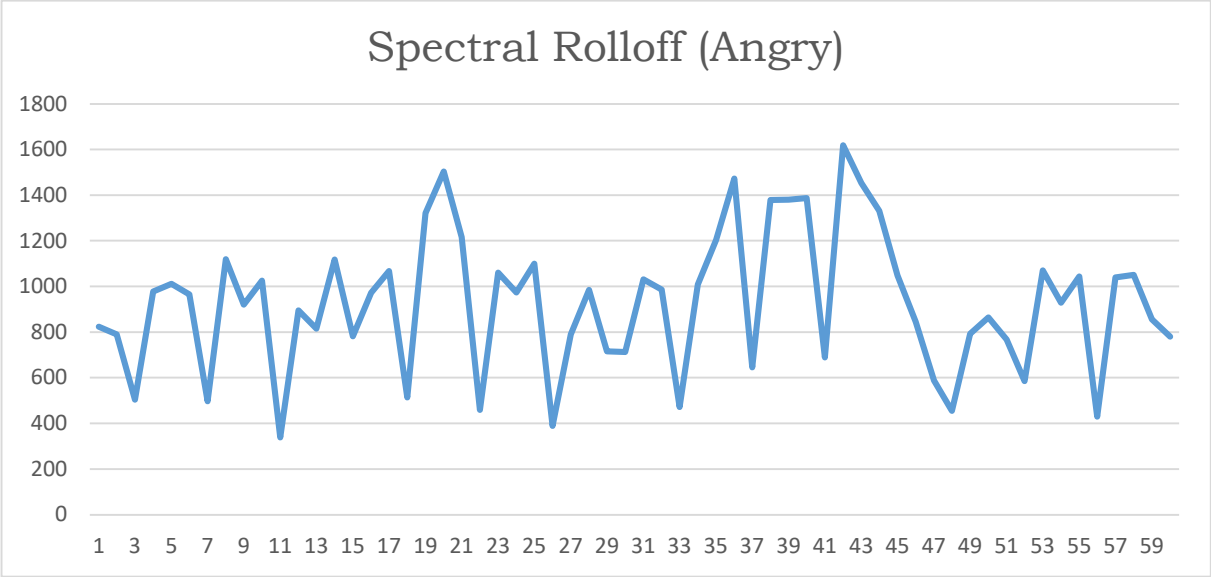

Fig 5.1.2.91: Plotting Spectral Rolloff for 'Angry' Emotion Label

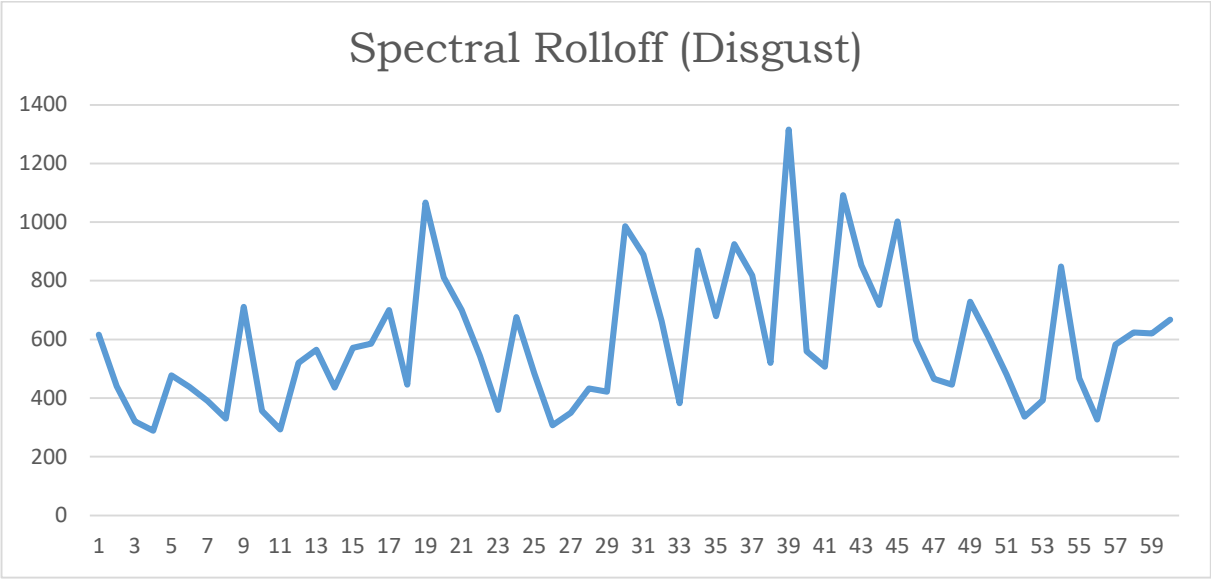

Fig 5.1.2.92: Plotting Spectral Rolloff for 'Disgust' Emotion Label

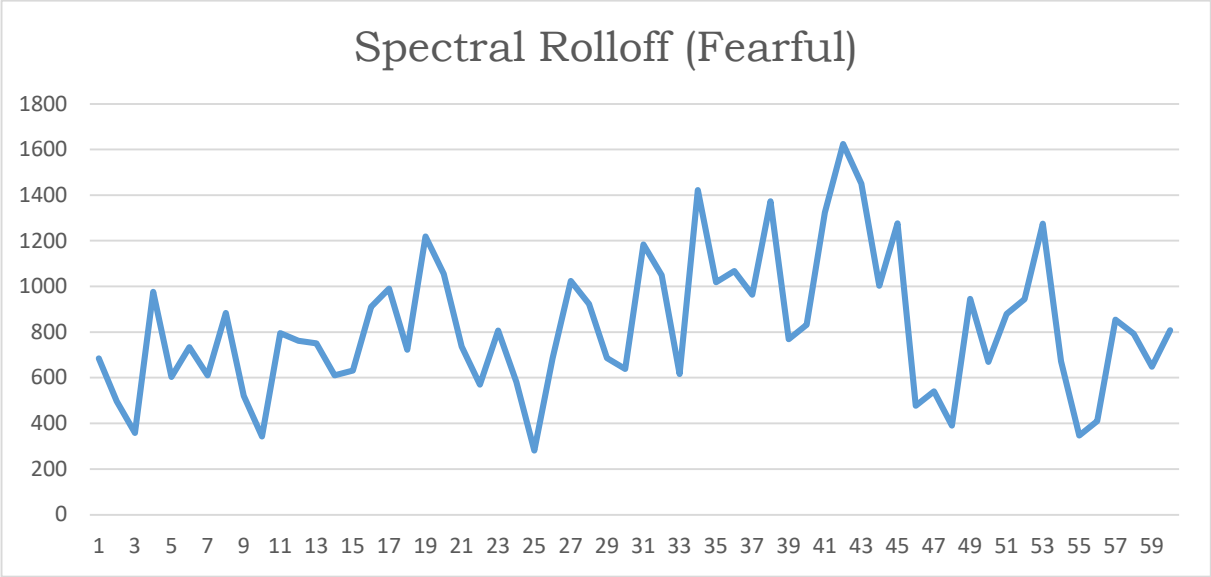

Fig 5.1.2.93: Plotting Spectral Rolloff for 'Fearful' Emotion Label

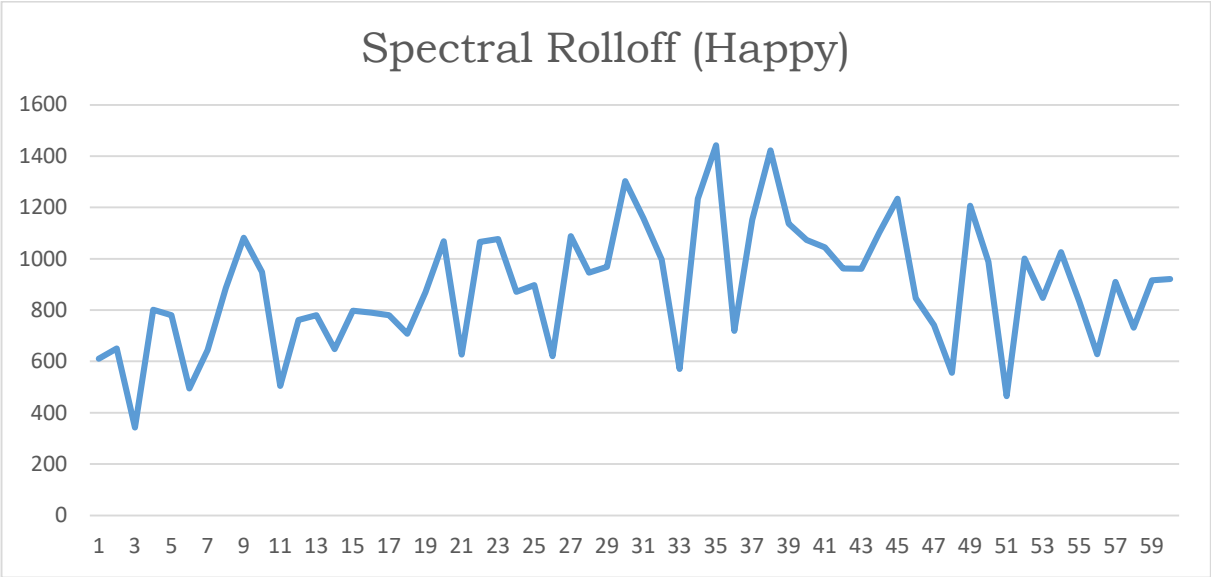

Fig 5.1.2.94: Plotting Spectral Rolloff for 'Happy' Emotion Label

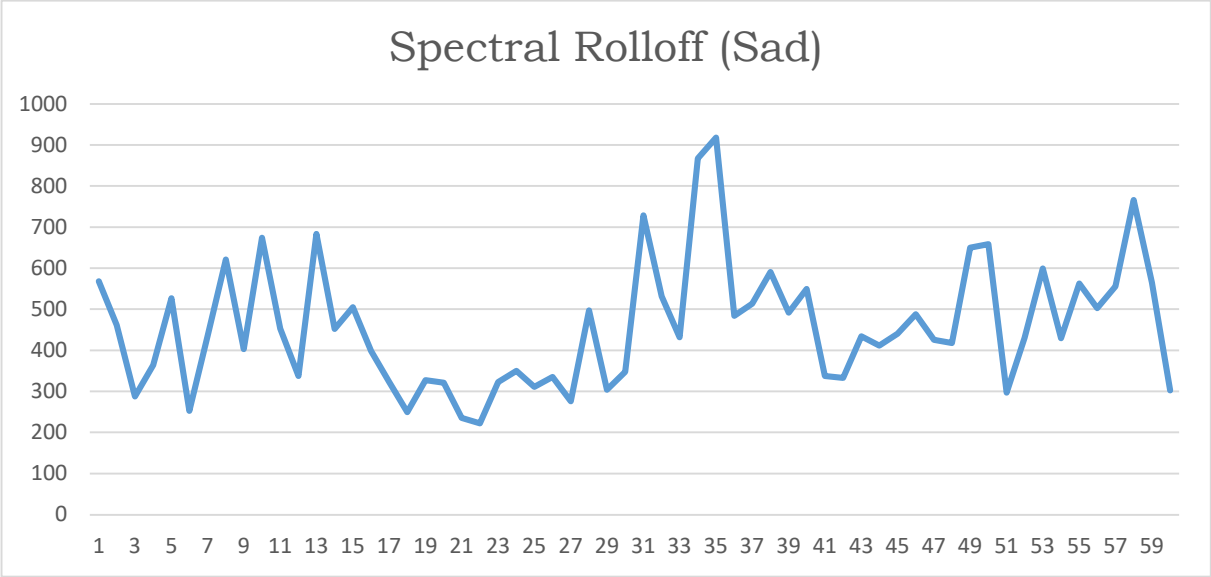

Fig5.1.2.95: Plotting Spectral Rolloff for 'Sad' Emotion Label

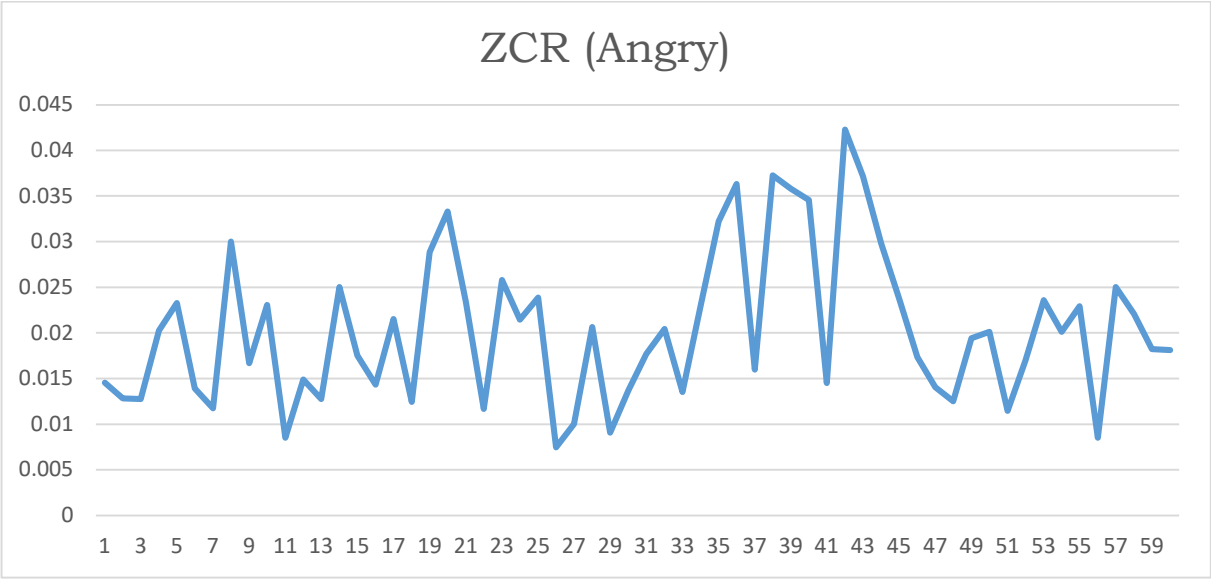

Fig 5.1.2.96: Plotting ZCR for 'Angry' Emotion Label

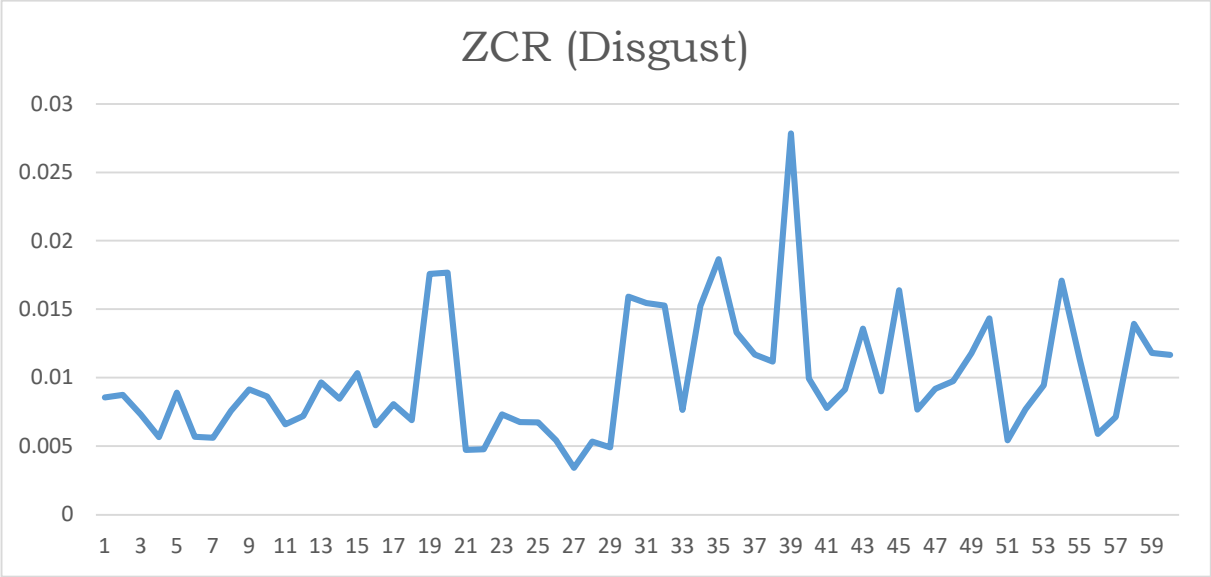

Fig 5.1.2.97: Plotting ZCR for 'Disgust' Emotion Label

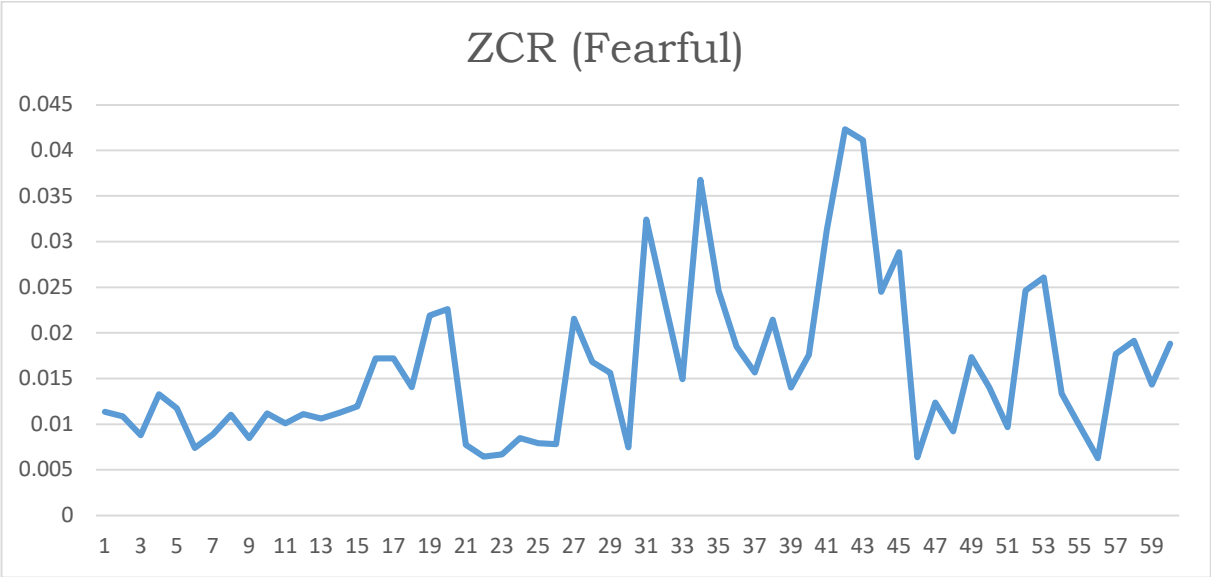

Fig 5.1.2.98: Plotting ZCR for 'Fearful' Emotion Label

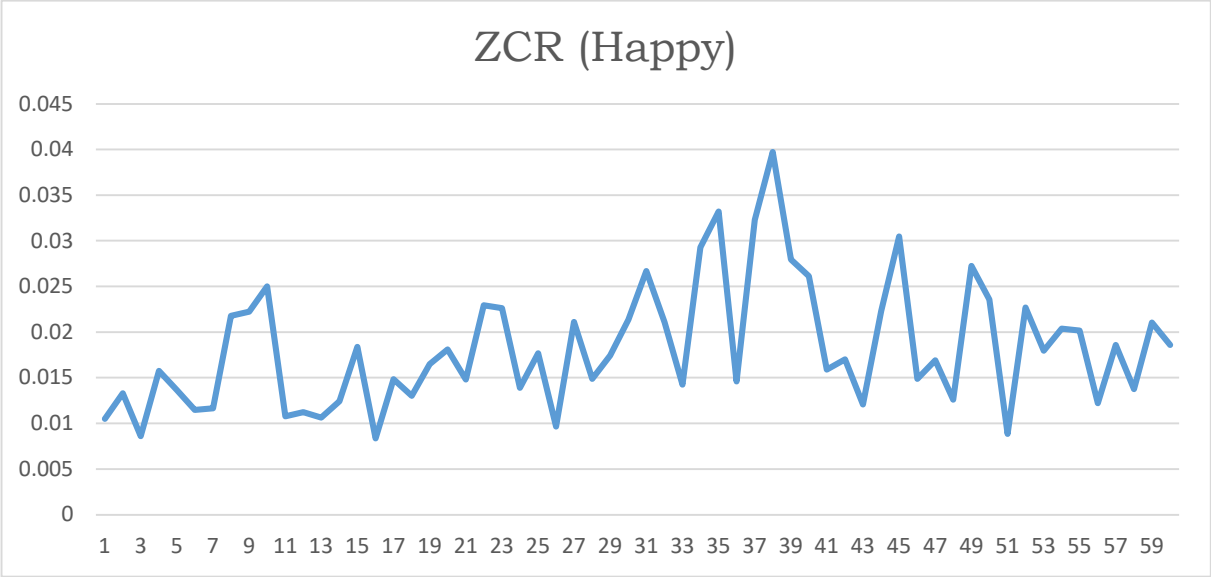

Fig 5.1.2.99: Plotting ZCR for 'Happy' Emotion Label

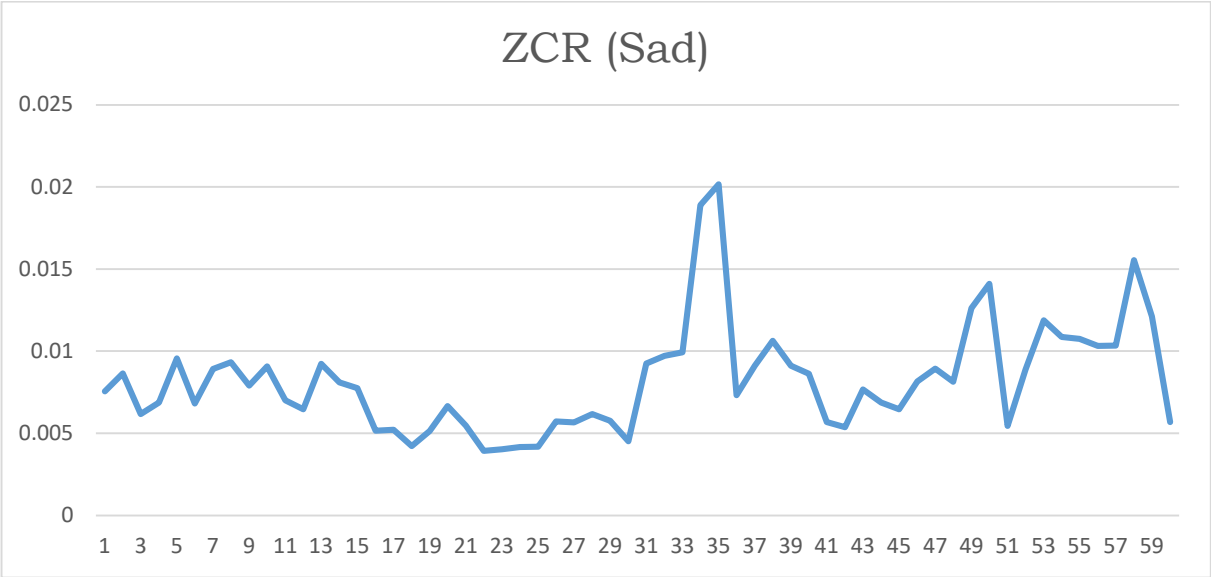

Fig 5.1.2.100: Plotting ZCR for 'Sad' Emotion Label

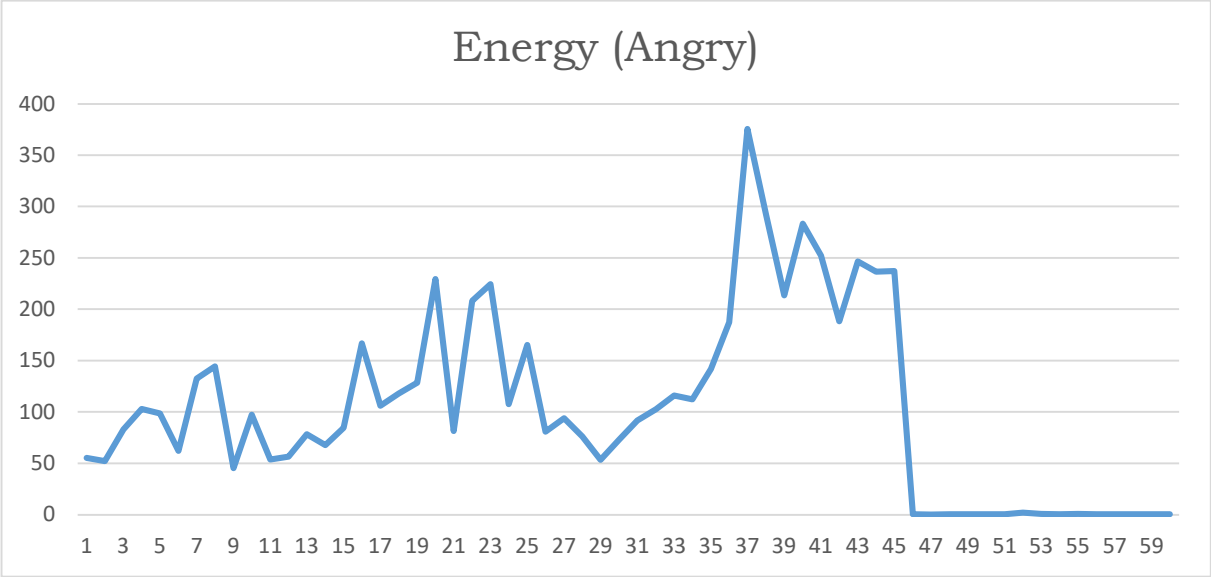

Fig 5.1.2.101: Plotting Energy for 'Angry' Emotion Label

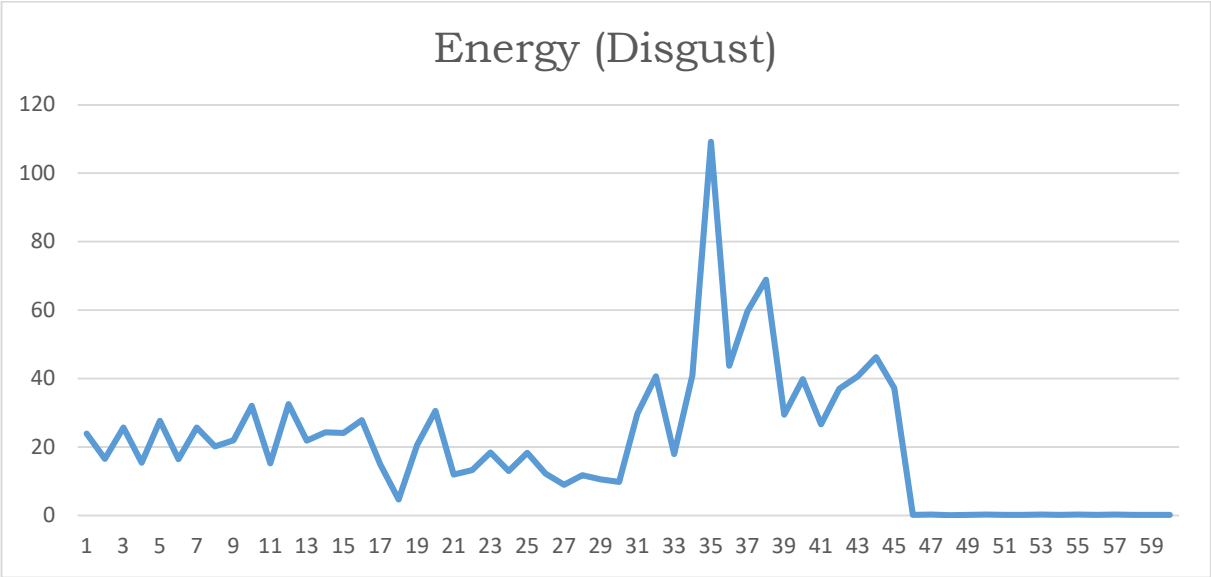

Fig 5.1.2.102: Plotting Energy for 'Disgust' Emotion Label

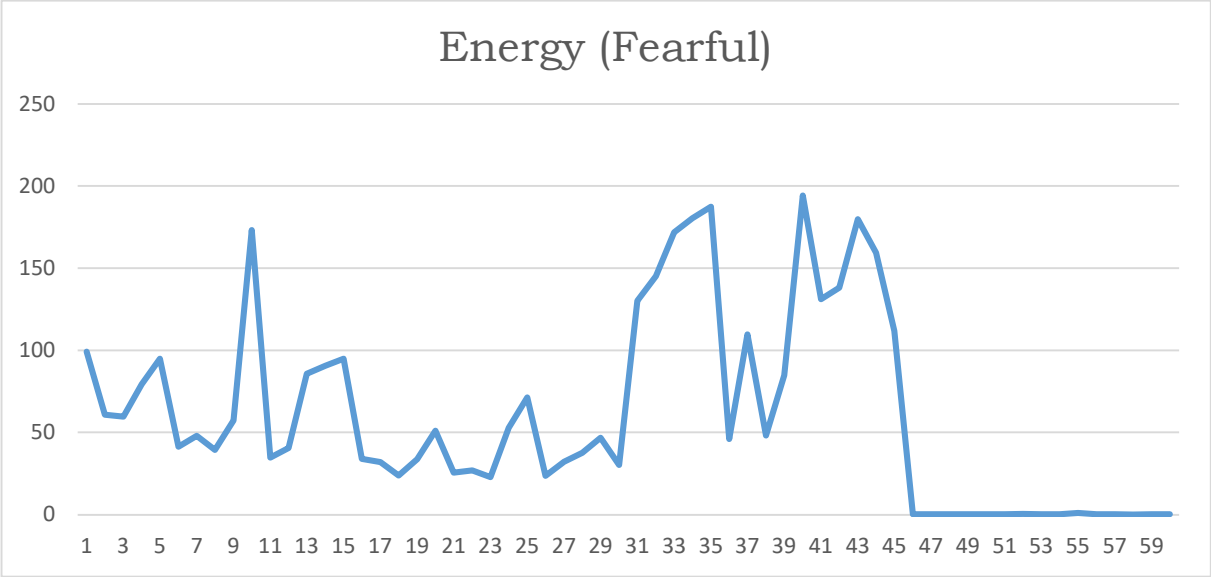

Fig 5.1.2.103: Plotting Energy for 'Fearful' Emotion Label

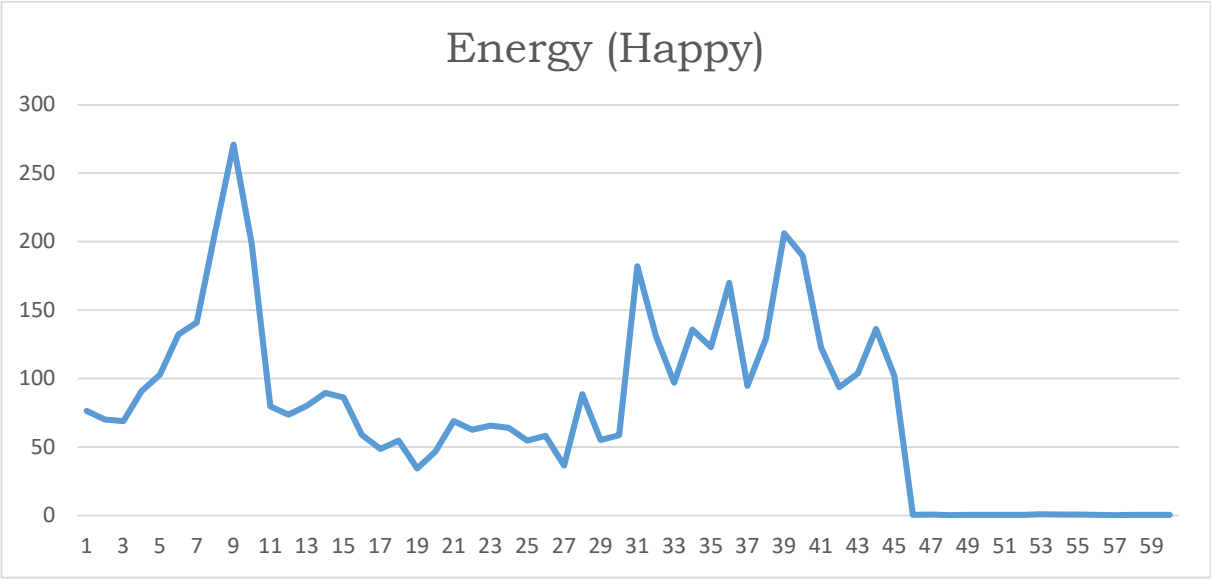

Fig 5.1.2.104: Plotting Energy for 'Happy' Emotion Label

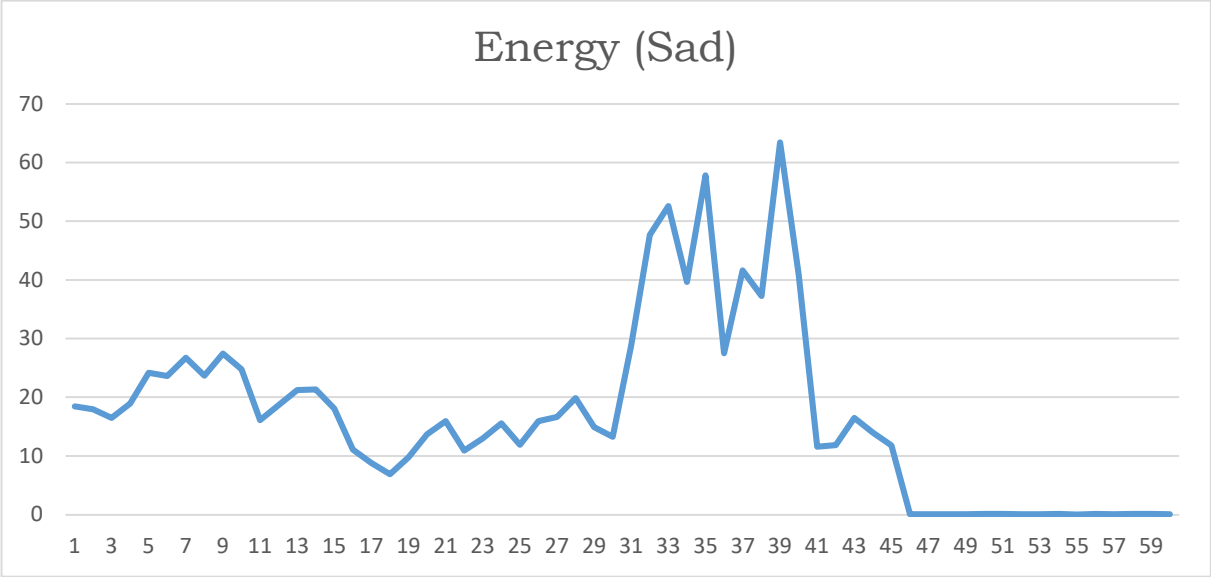

Fig 5.1.2.105: Plotting Energy for 'Sad' Emotion Label

Plots of EmoDB Dataset

Fig 5.1.2.106 – Fig 5.1.2.150 showcase plots of the variations in emotional expressions across all features within the EmoDB dataset. These visual representations provide insights into how different features respond to various emotions, offering a deeper understanding of the dataset's emotional dynamics.

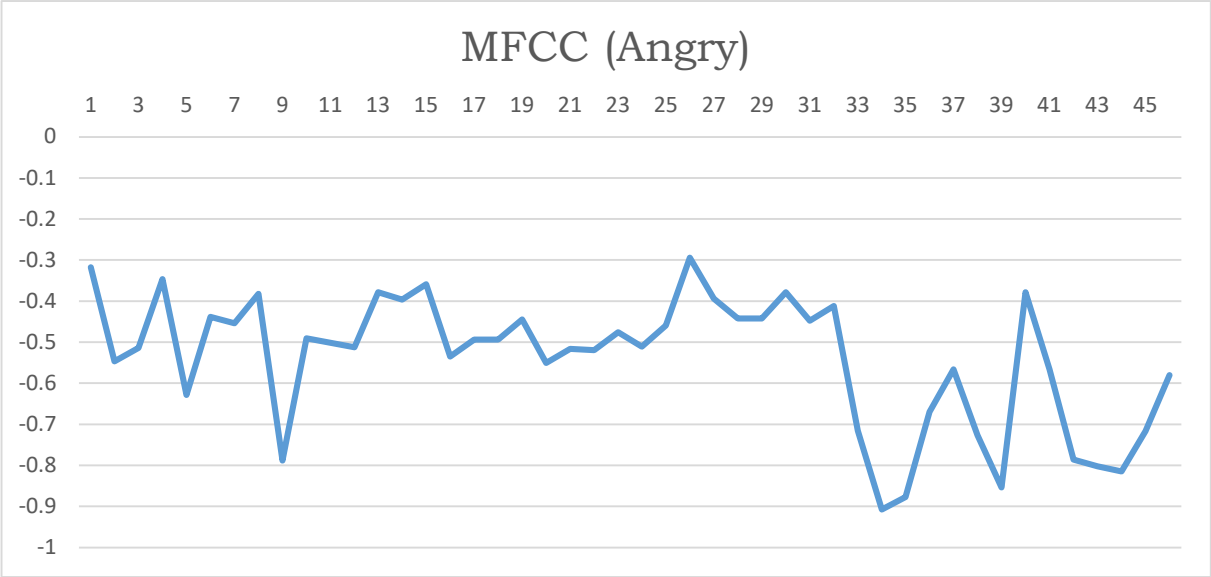

Fig 5.1.2.106: Plotting MFCC for 'Angry' Emotion Label

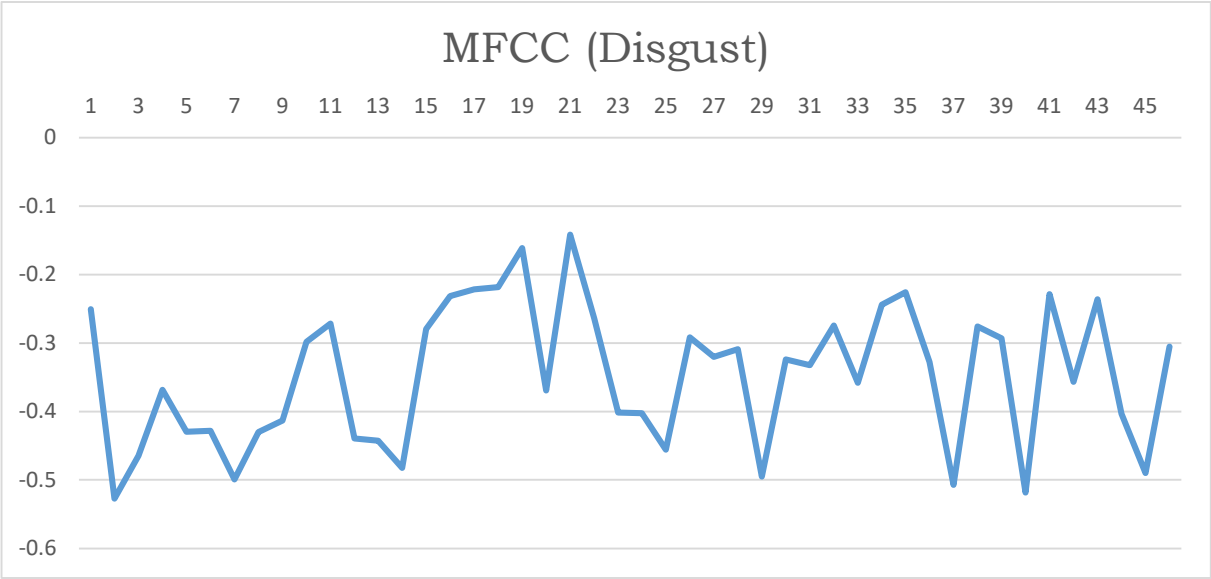

Fig 5.1.2.107: Plotting MFCC for 'Disgust' Emotion Label

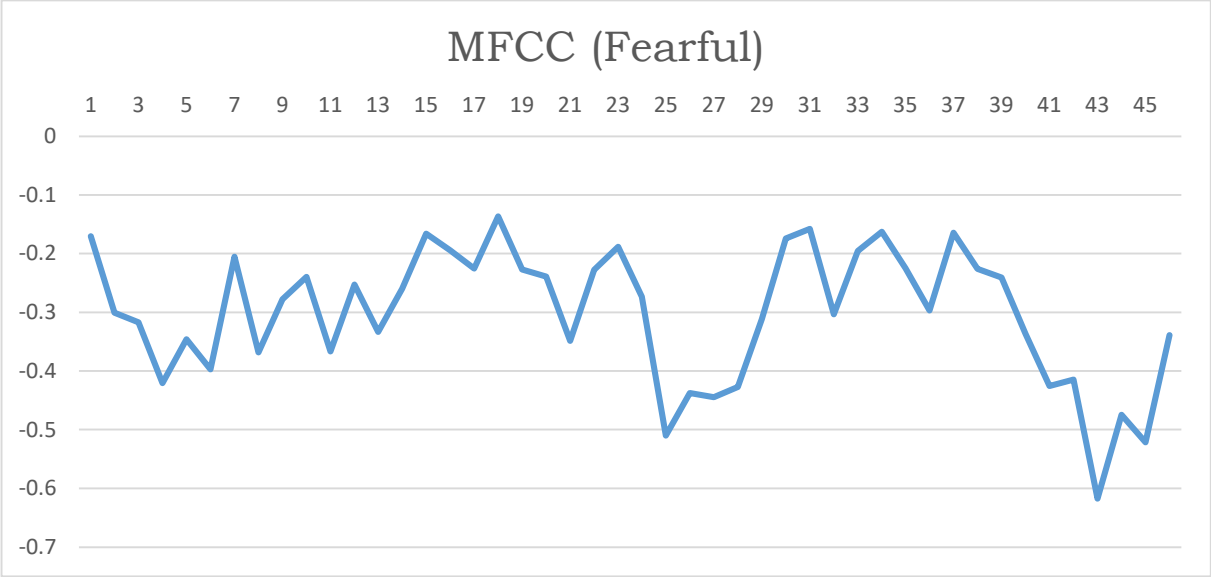

Fig 5.1.2.108: Plotting MFCC for 'Fearful' Emotion Label

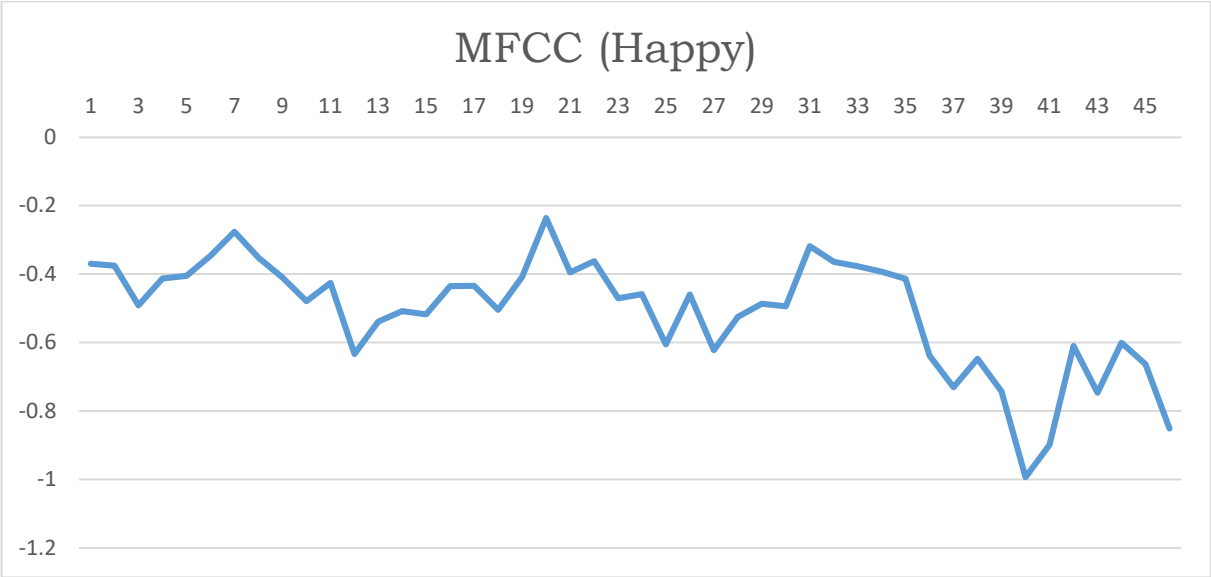

Fig 5.1.2.109: Plotting MFCC for 'Happy' Emotion Label

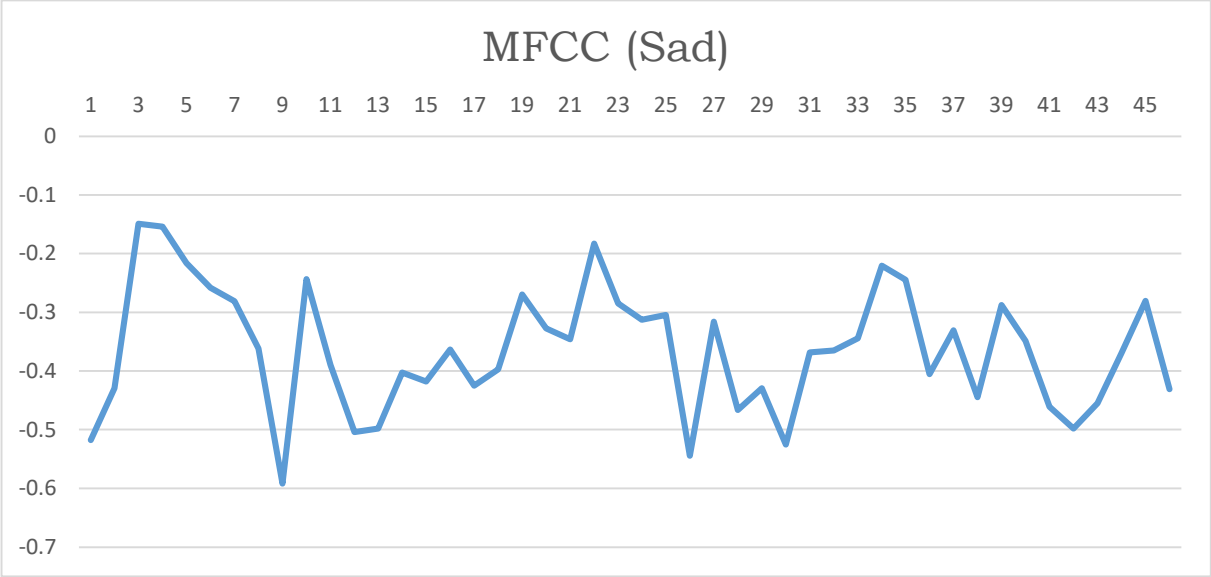

Fig 5.1.2.110: Plotting MFCC for 'Sad' Emotion Label

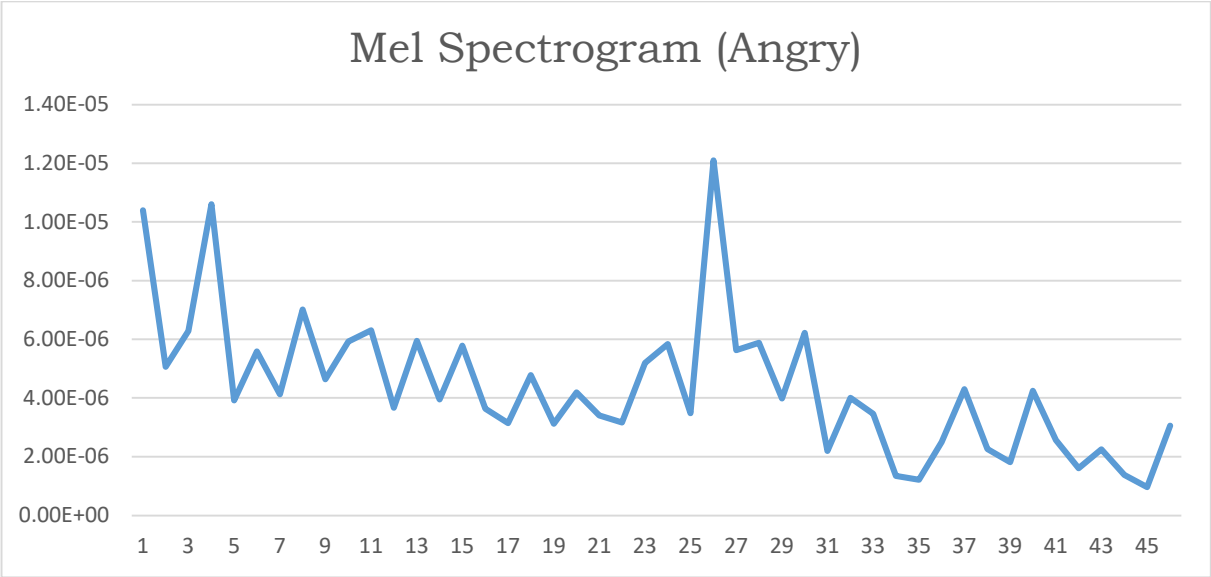

Fig 5.1.2.111: Plotting Mel Spectrogram for 'Angry' Emotion Label

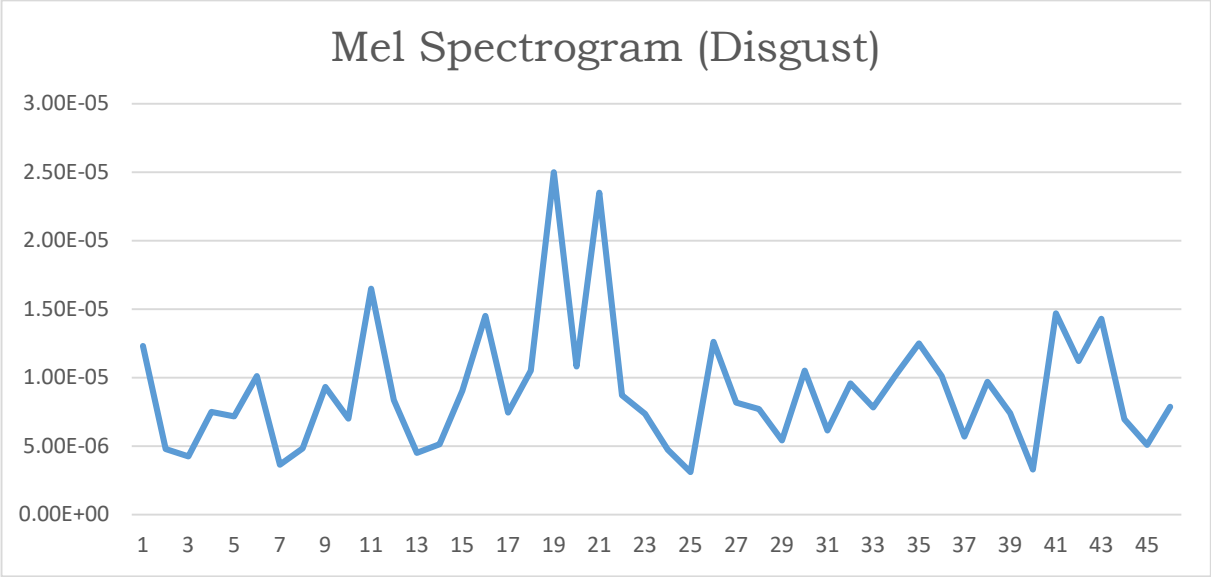

Fig 5.1.2.112: Plotting Mel Spectrogram for 'Disgust' Emotion Label

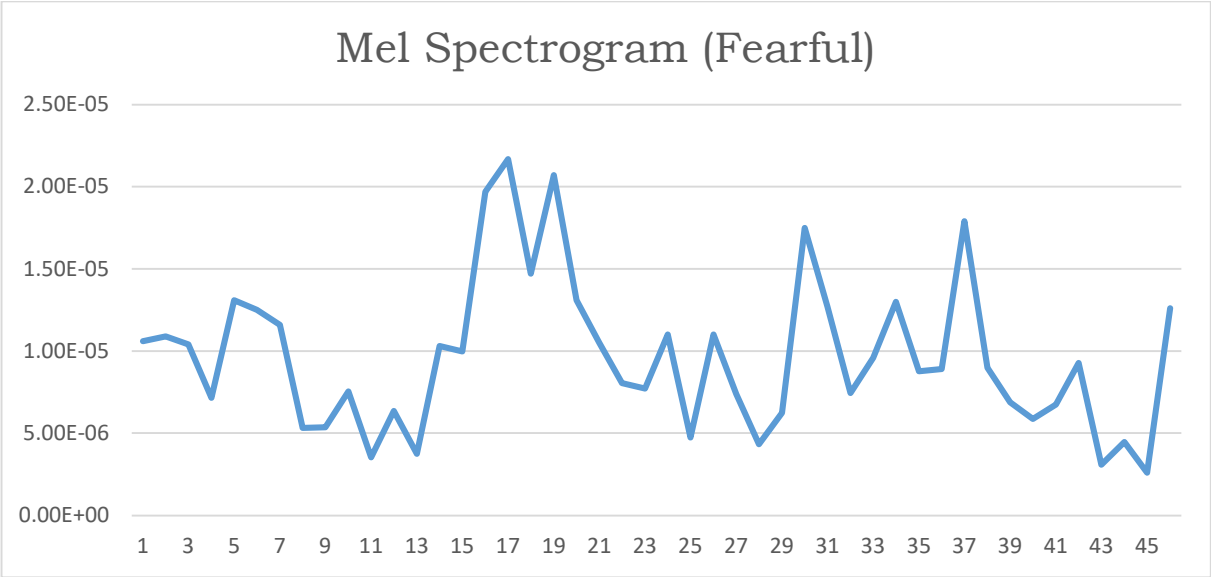

Fig 5.1.2.113: Plotting Mel Spectrogram for 'Fearful' Emotion Label

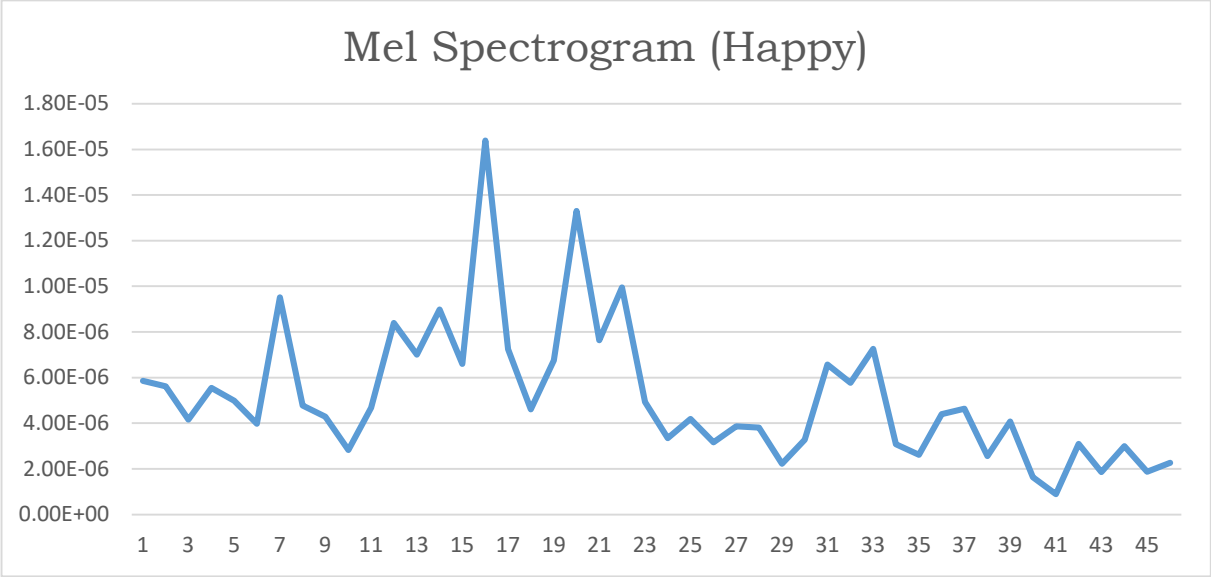

Fig 5.1.2.114: Plotting Mel Spectrogram for 'Happy' Emotion Label

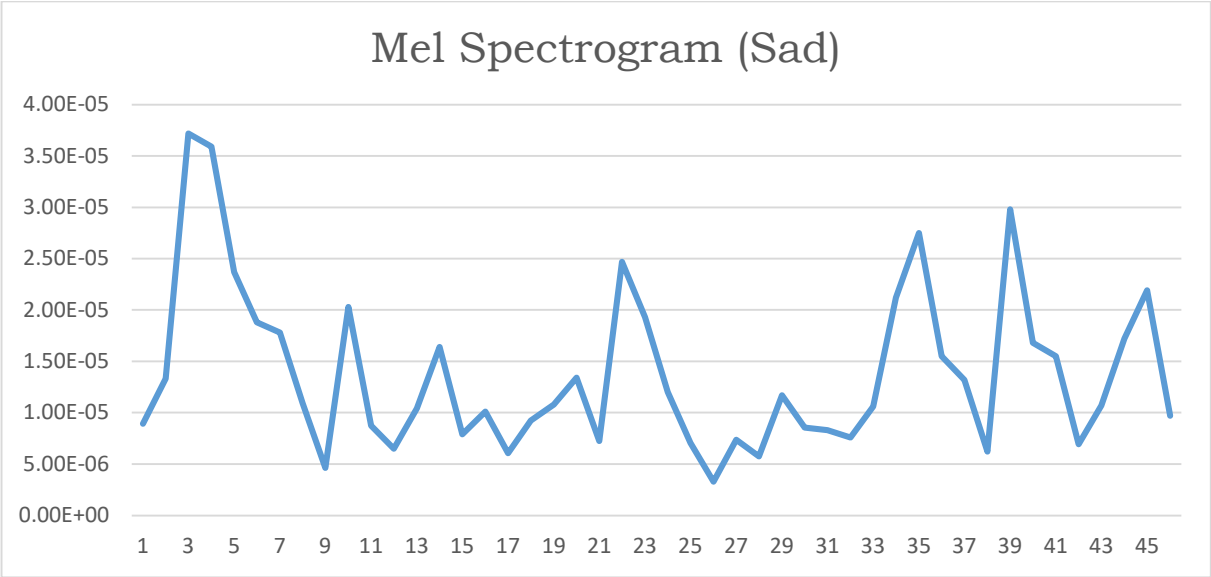

Fig 5.1.2.115: Plotting Mel Spectrogram for 'Sad' Emotion Label

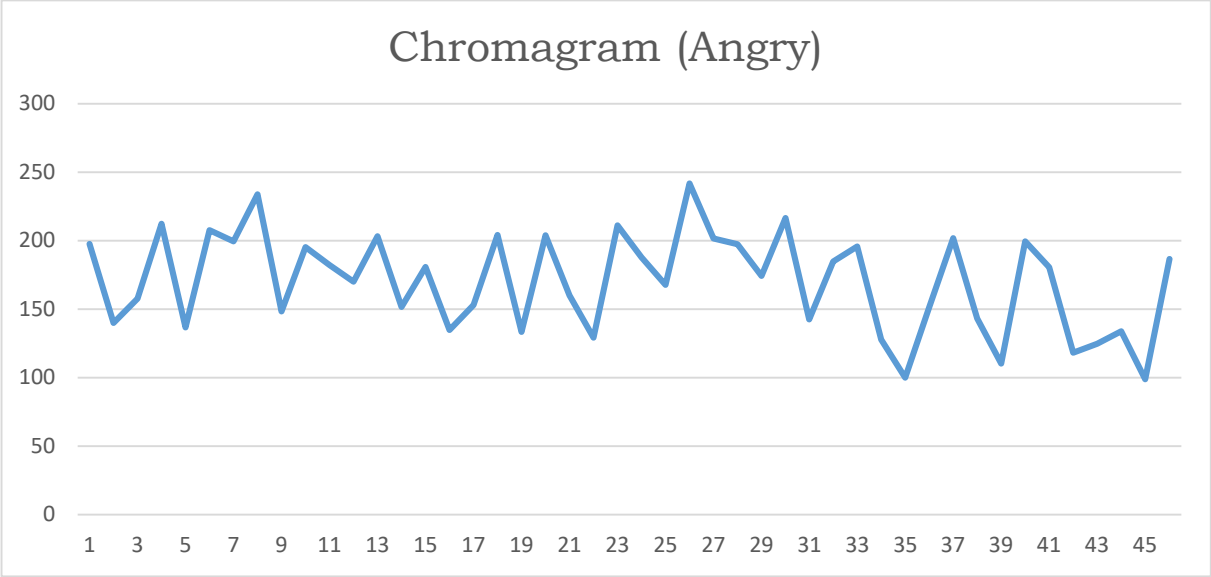

Fig 5.1.2.116: Plotting Chromagram for 'Angry' Emotion Label

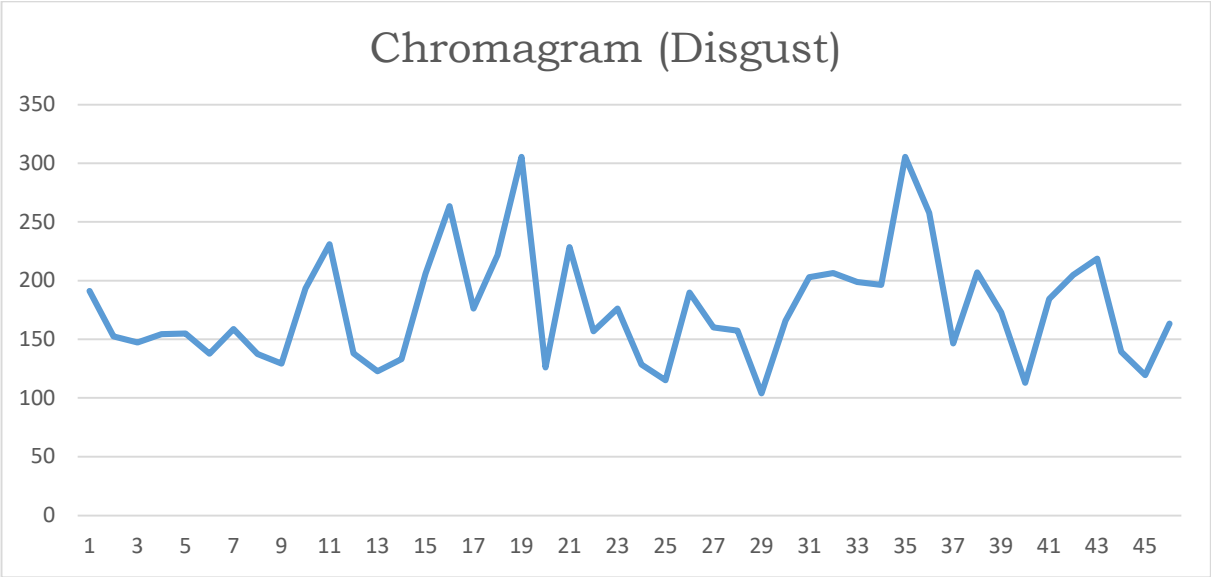

Fig 5.1.2.117: Plotting Chromagram for 'Disgust' Emotion Label

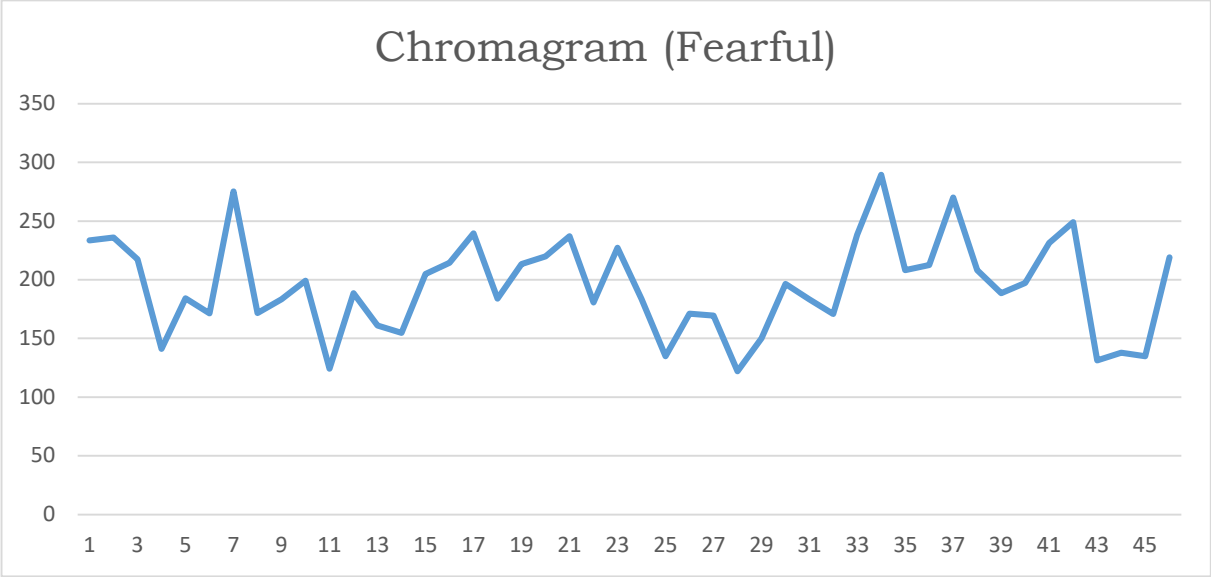

Fig 5.1.2.118: Plotting Chromagram for 'Fearful' Emotion Label

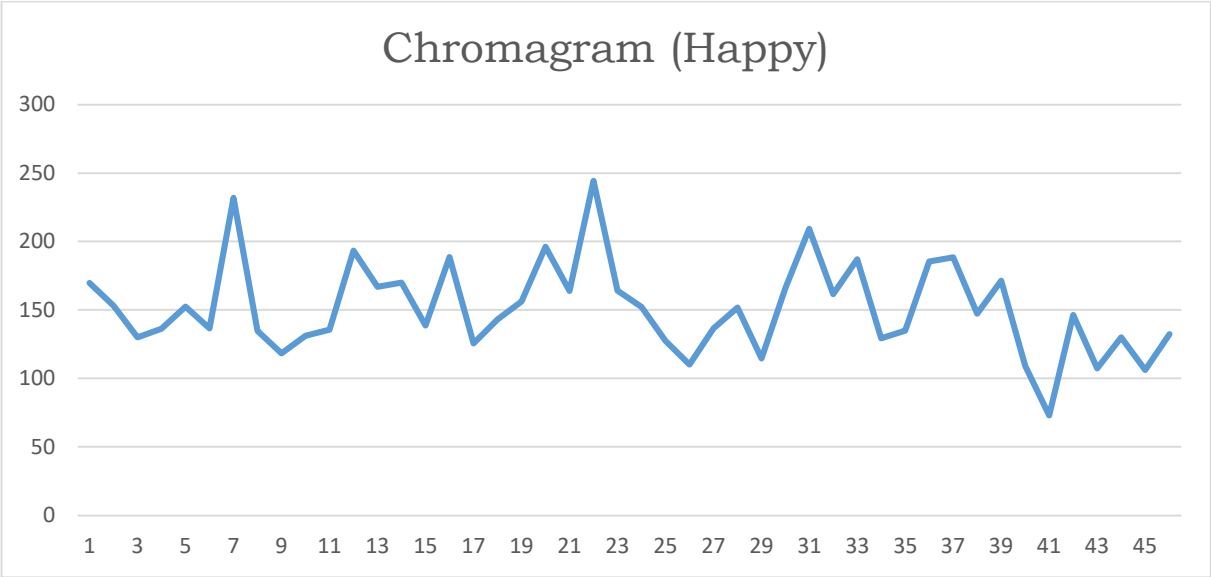

Fig 5.1.2.119: Plotting Chromagram for 'Happy' Emotion Label

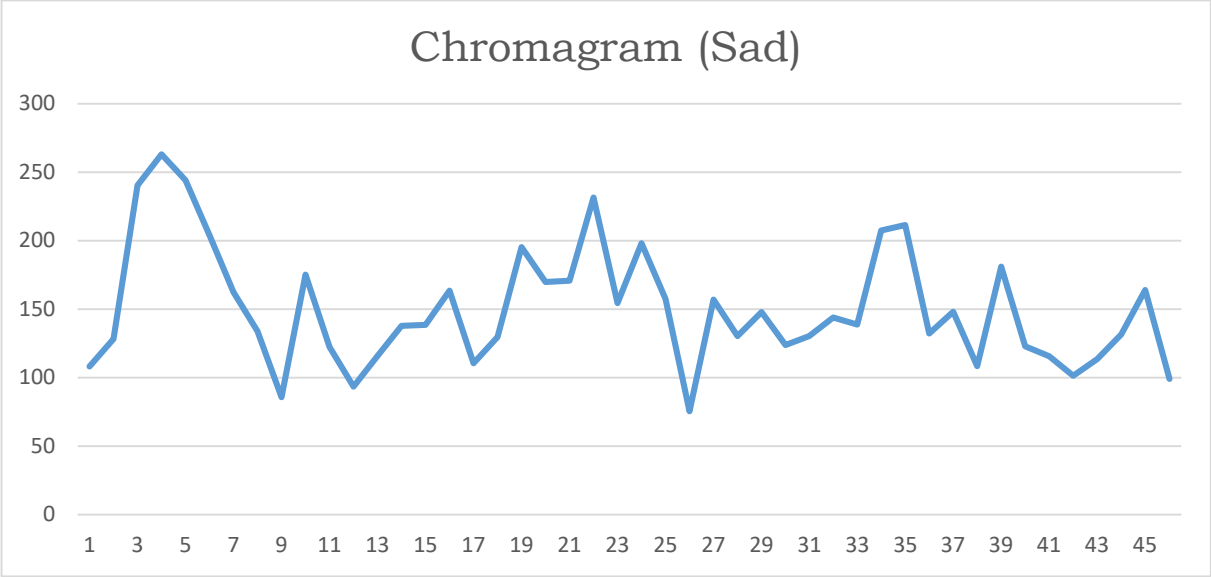

Fig 5.1.2.120: Plotting Chromagram for 'Sad' Emotion Label

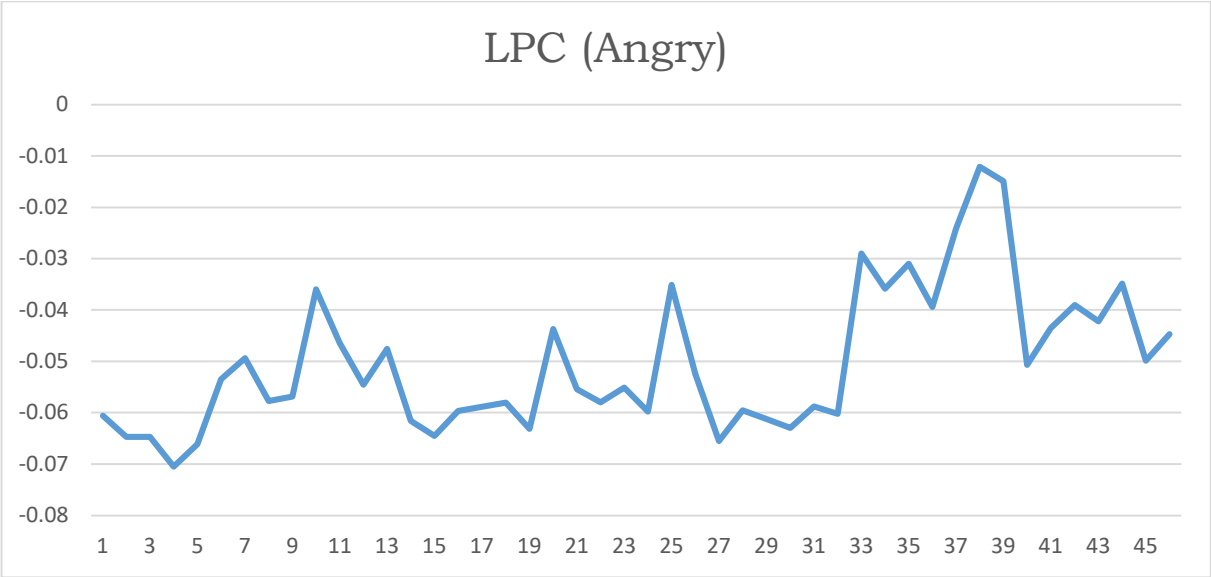

Fig 5.1.2.121: Plotting LPC for 'Angry' Emotion Label

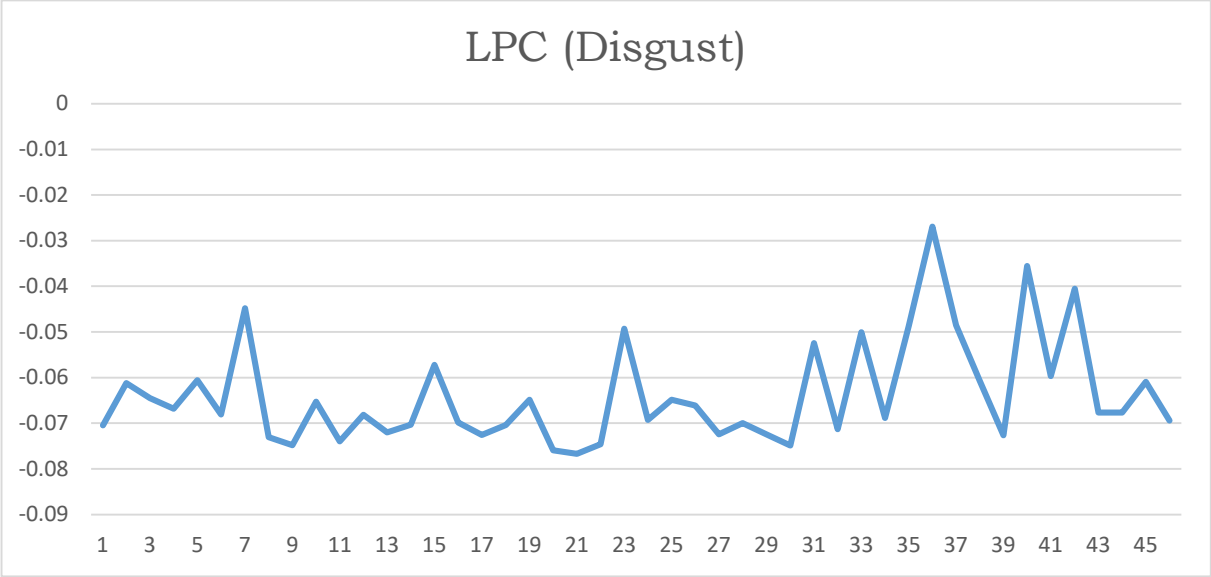

Fig 5.1.2.122: Plotting LPC for 'Disgust' Emotion Label

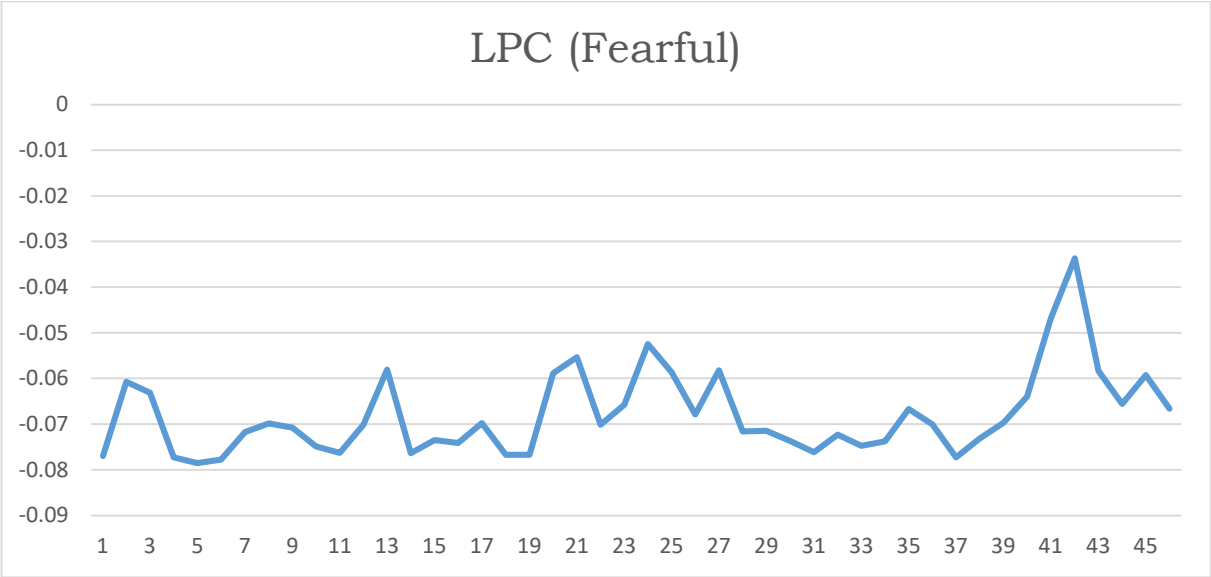

Fig 5.1.2.123: Plotting LPC for 'Fearful' Emotion Label

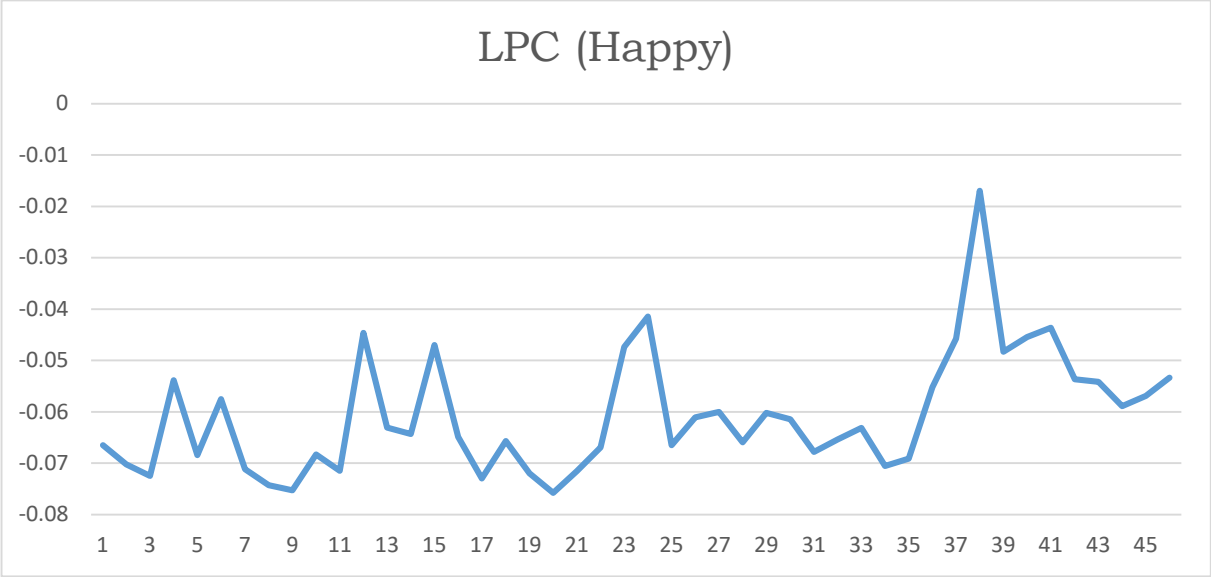

Fig 5.1.2.124: Plotting LPC for 'Happy' Emotion Label

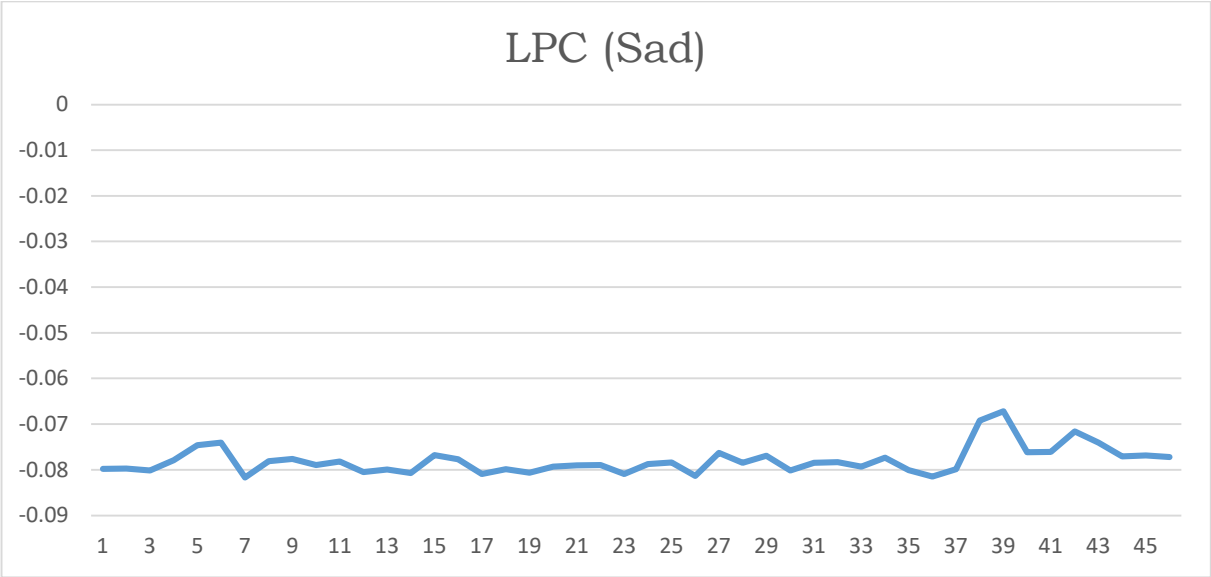

Fig 5.1.2.125: Plotting LPC for 'Sad' Emotion Label

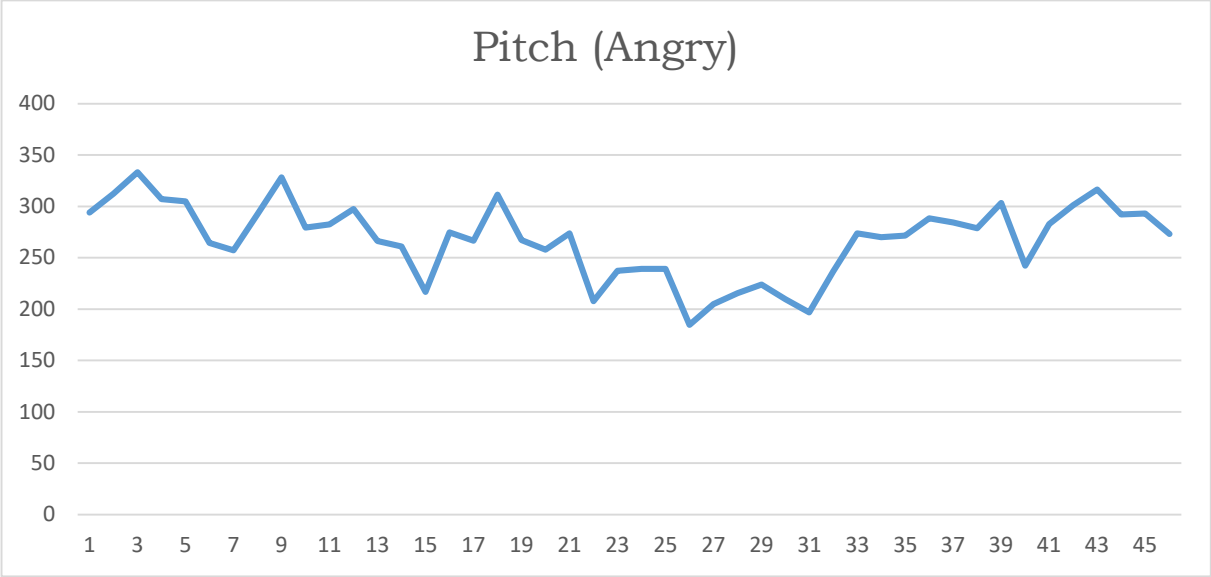

Fig 5.1.2.126: Plotting Pitch for 'Angry' Emotion Label

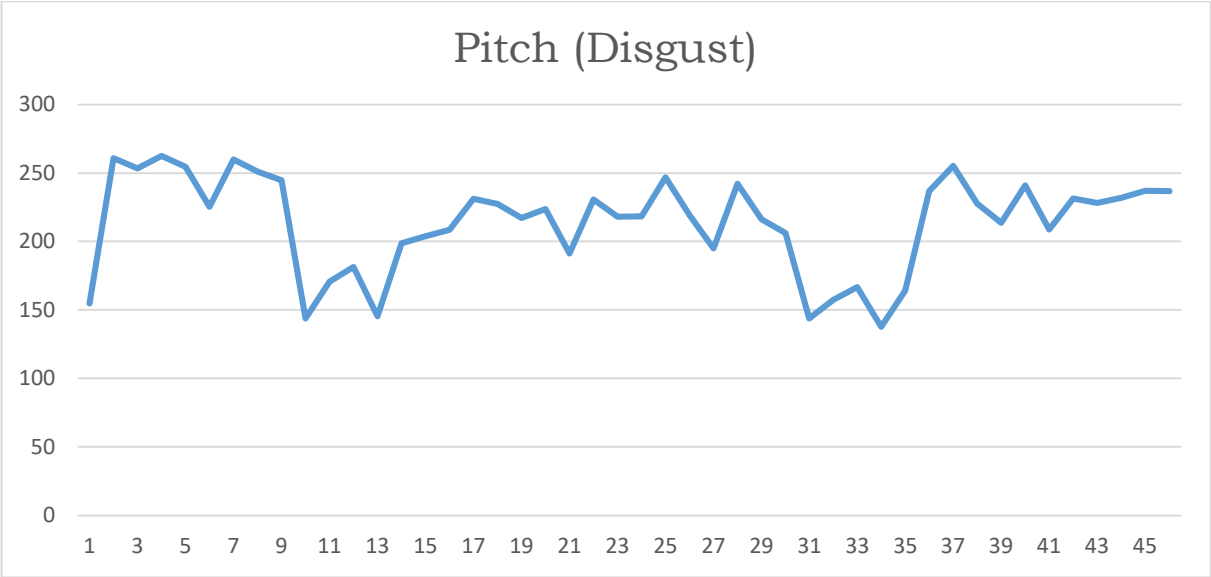

Fig 5.1.2.127: Plotting Pitch for 'Disgust' Emotion Label

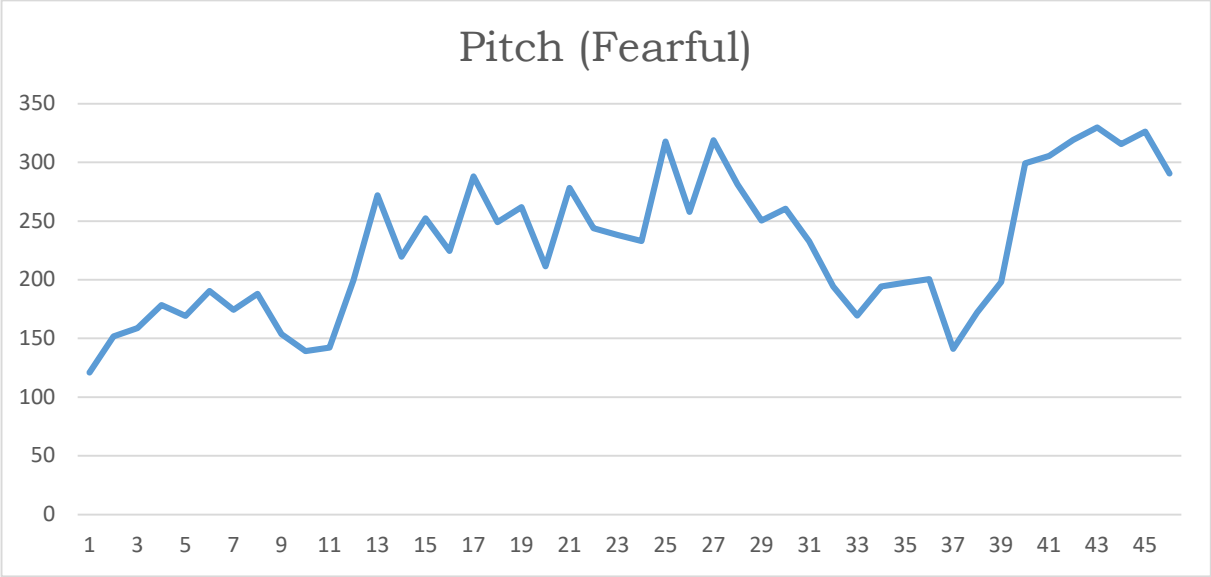

Fig 5.1.2.128: Plotting Pitch for 'Fearful' Emotion Label

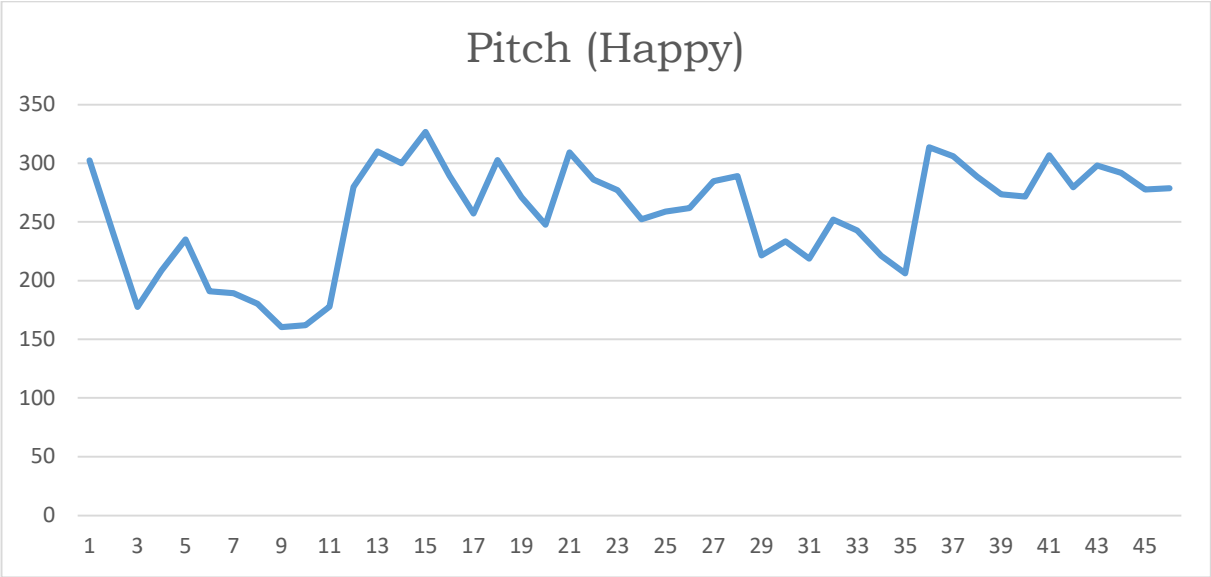

Fig 5.1.2.129: Plotting Pitch for 'Happy' Emotion Label

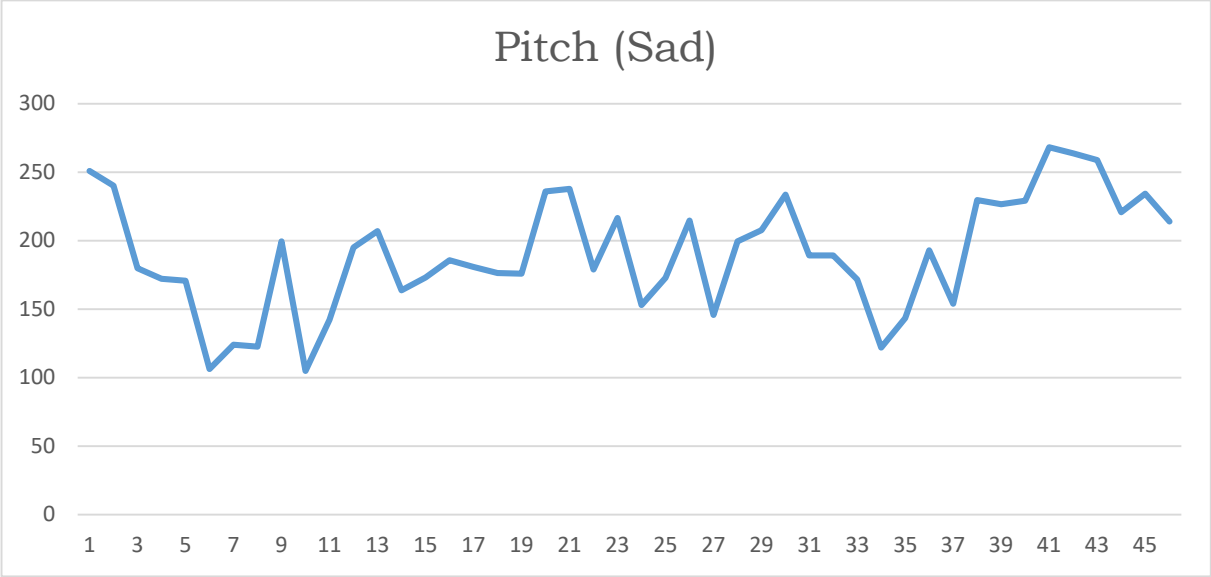

Fig 5.1.2.130: Plotting Pitch for 'Sad' Emotion Label

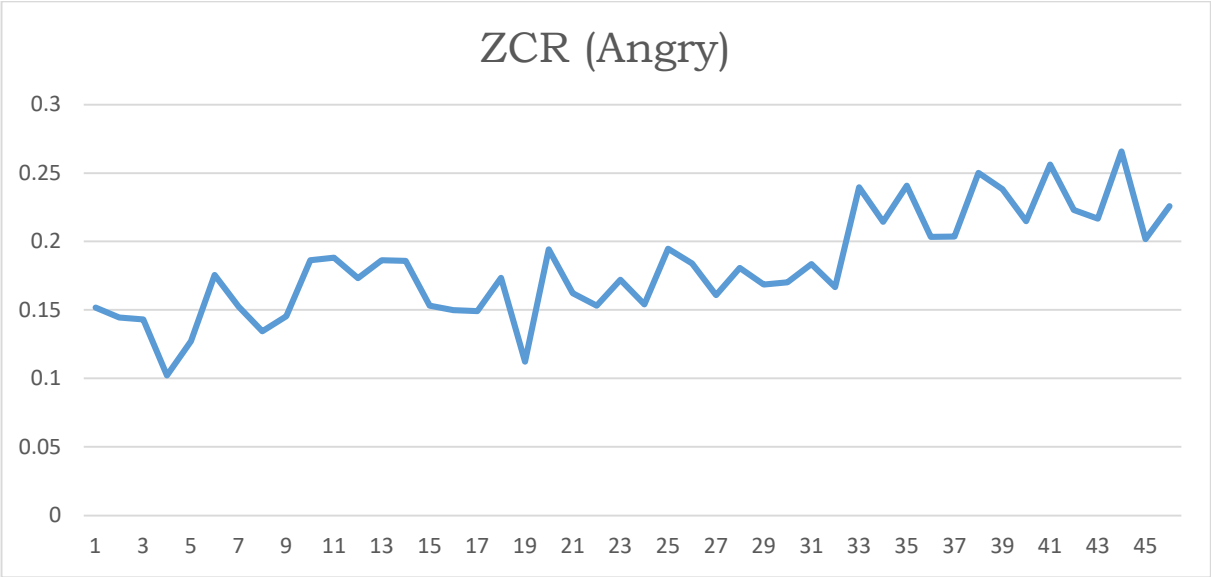

Fig 5.1.2.131: Plotting ZCR for 'Angry' Emotion Label

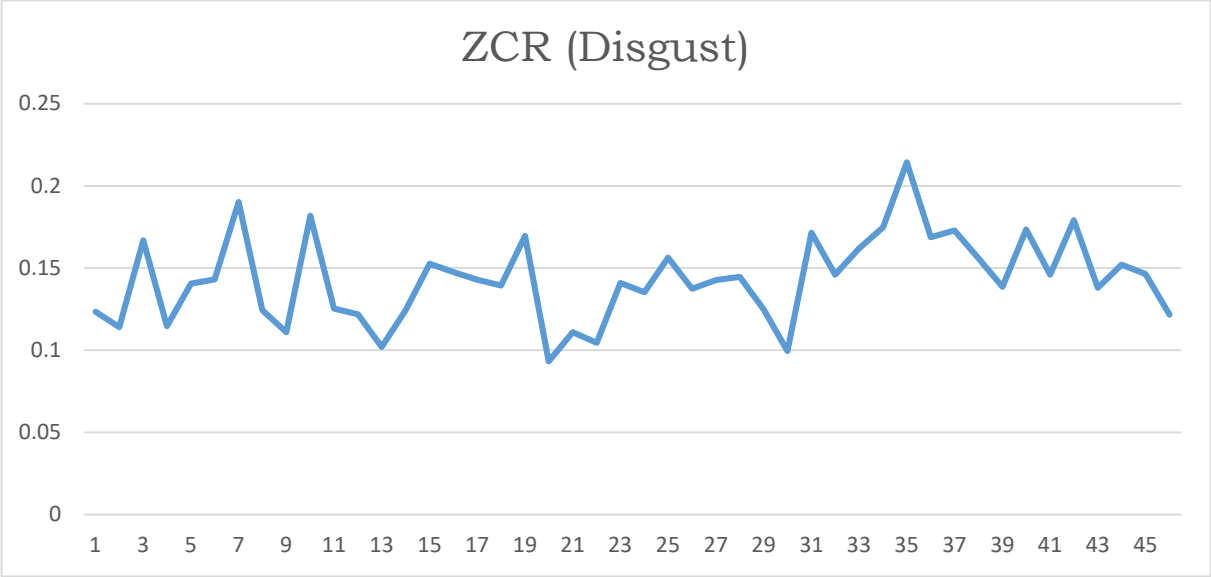

Fig 5.1.2.132: Plotting ZCR for 'Disgust' Emotion Label

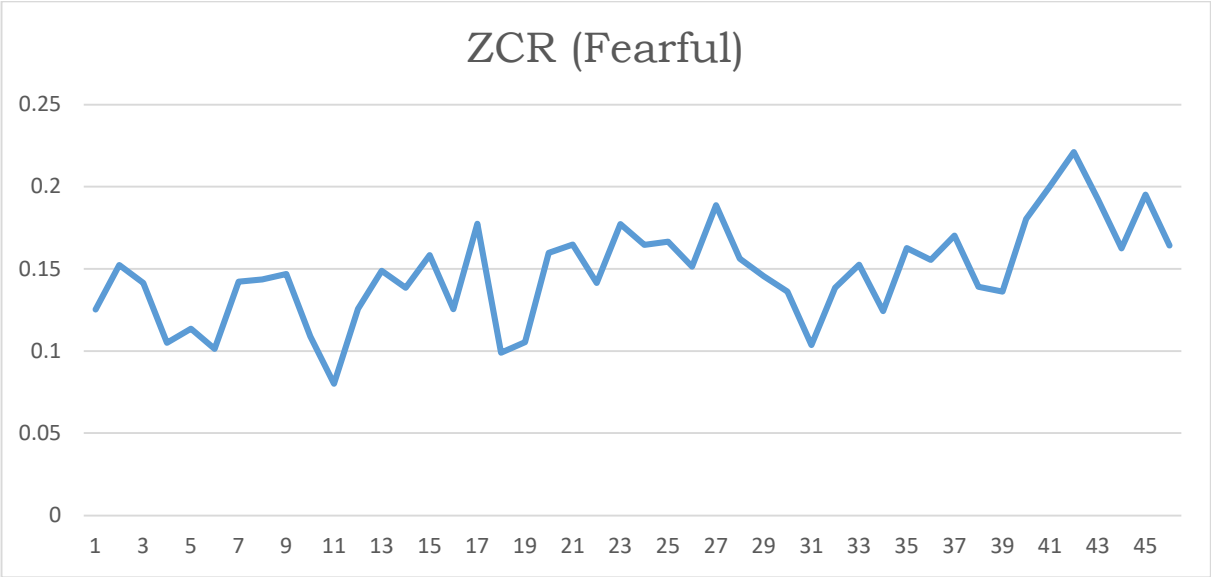

Fig 5.1.2.133: Plotting ZCR for 'Fearful' Emotion Label

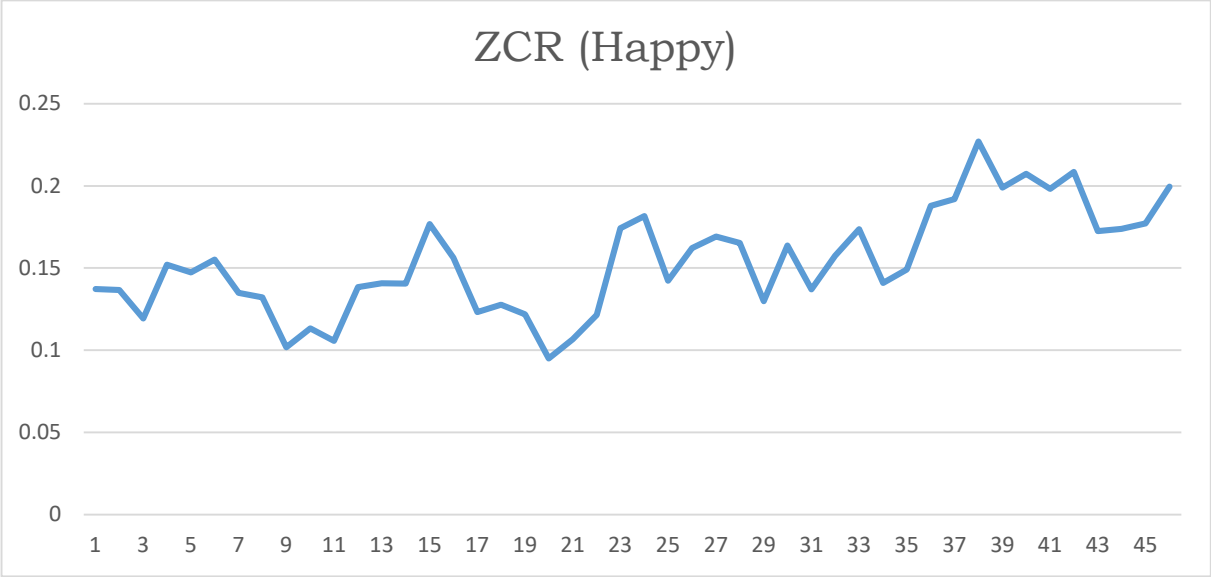

Fig 5.1.2.134: Plotting ZCR for 'Happy' Emotion Label

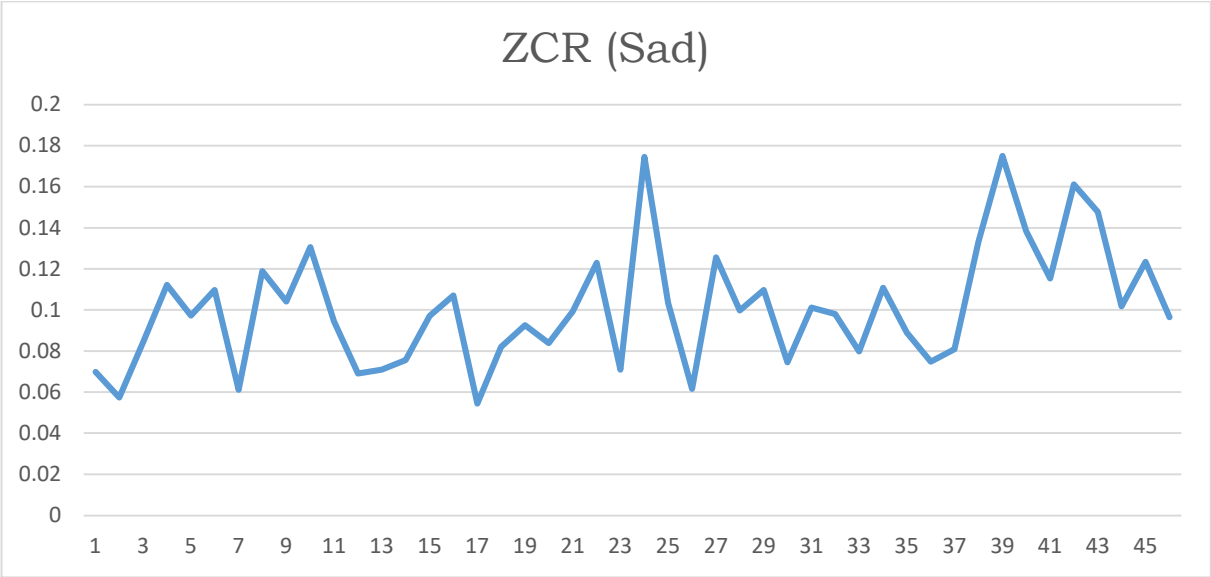

Fig 5.1.2.135: Plotting ZCR for 'Sad' Emotion Label

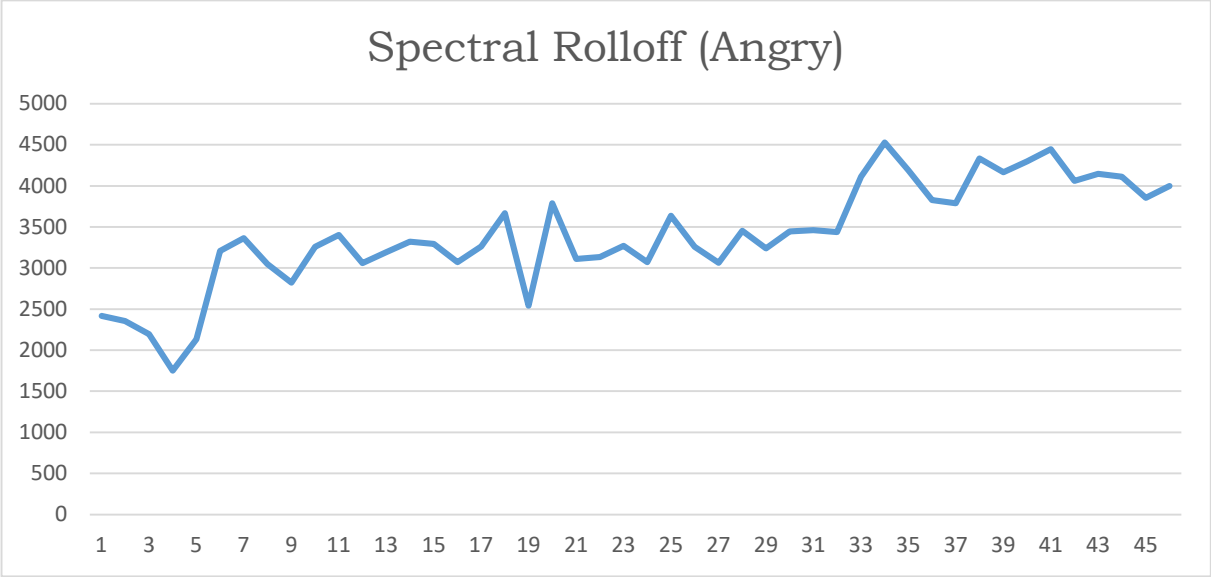

Fig 5.1.2.136: Plotting Spectral Rolloff for 'Angry' Emotion Label

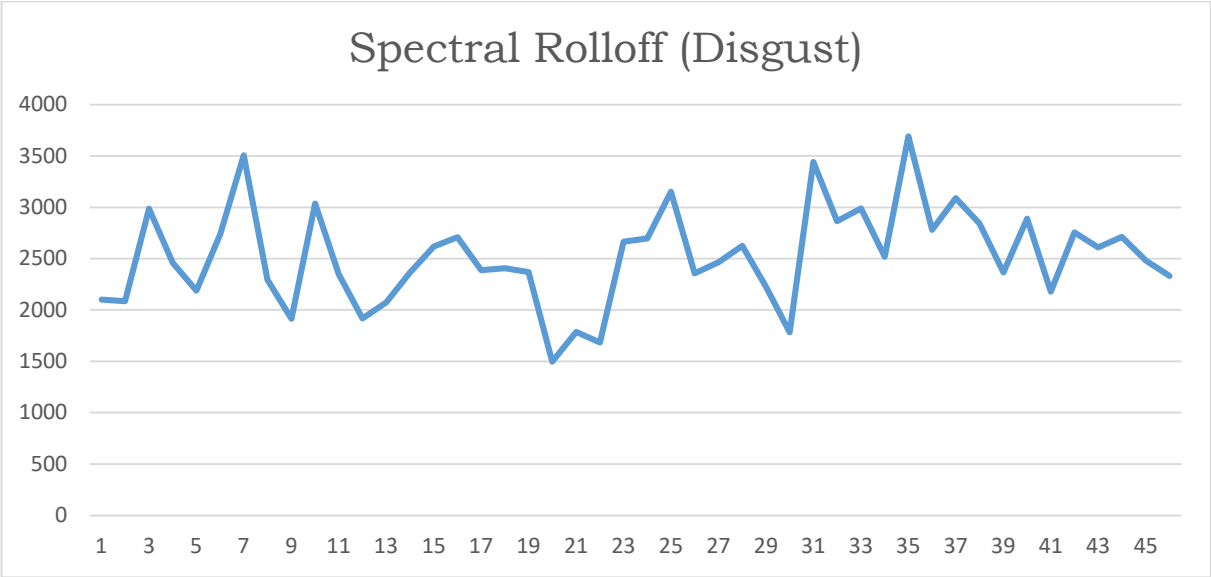

Fig 5.1.2.137: Plotting Spectral Rolloff for 'Disgust' Emotion Label

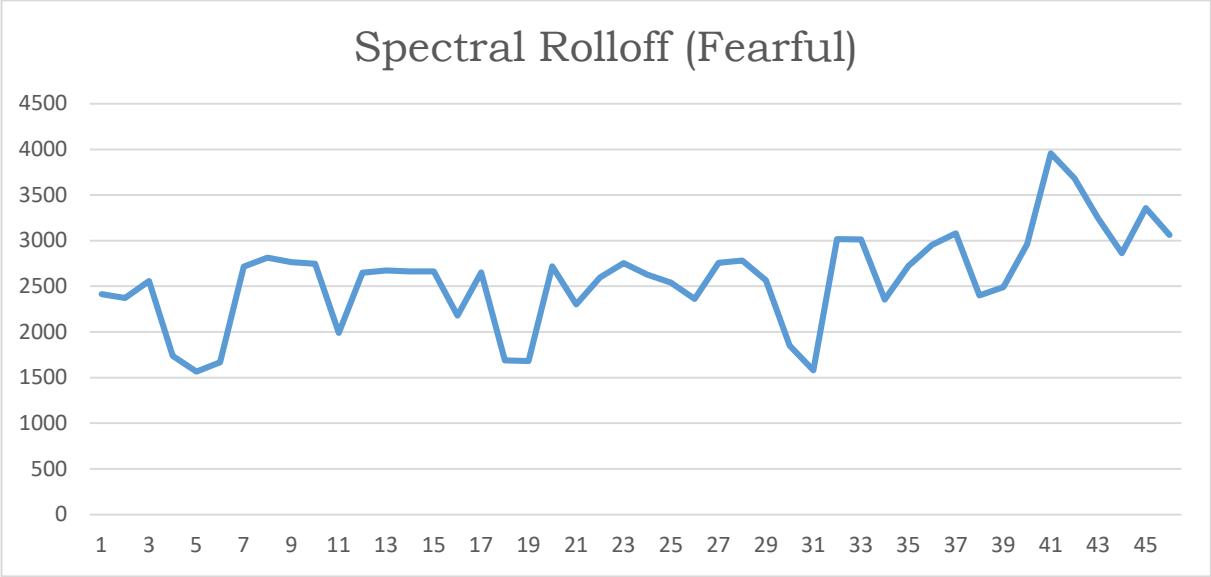

Fig 5.1.2.138: Plotting Spectral Rolloff for 'Fearful' Emotion Label

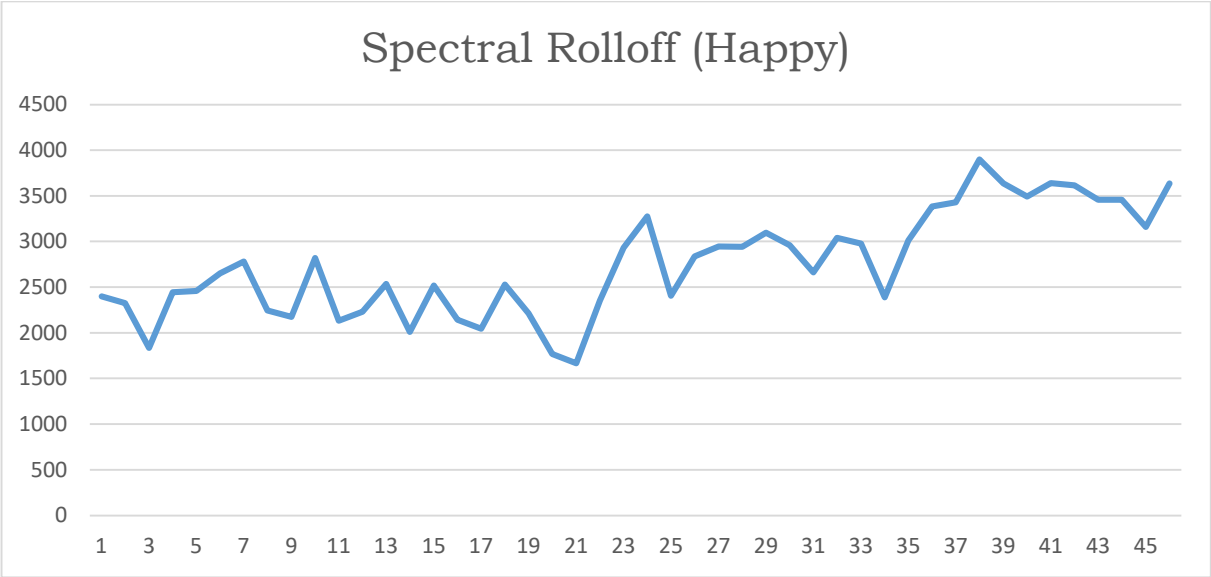

Fig 5.1.2.139: Plotting Spectral Rolloff for 'Happy' Emotion Label

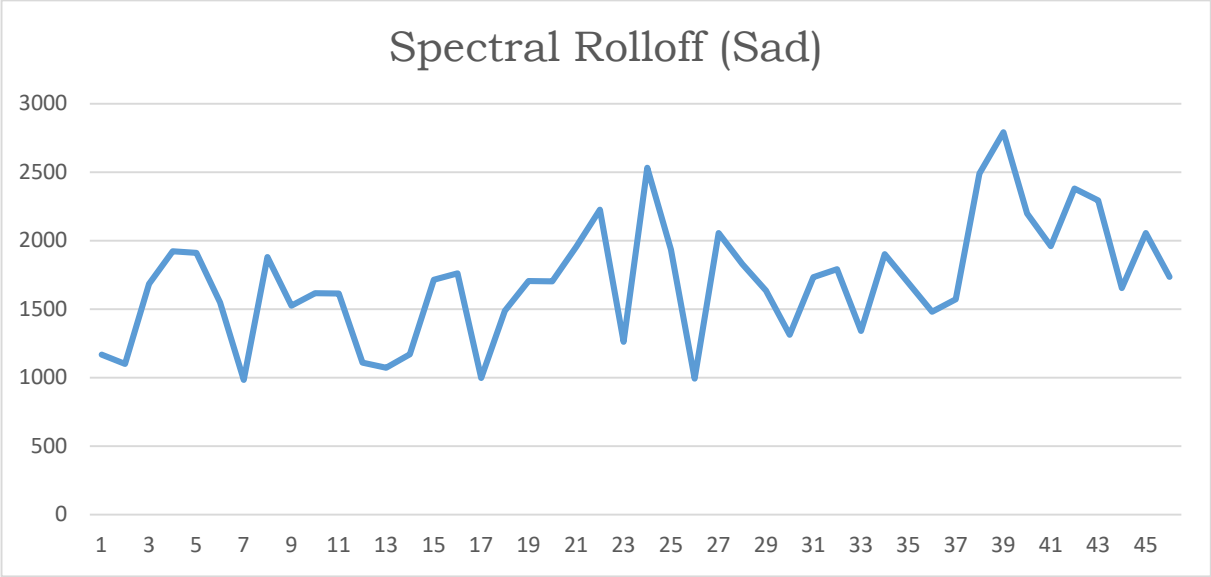

Fig 5.1.2.140: Plotting Spectral Rolloff for 'Sad' Emotion Label

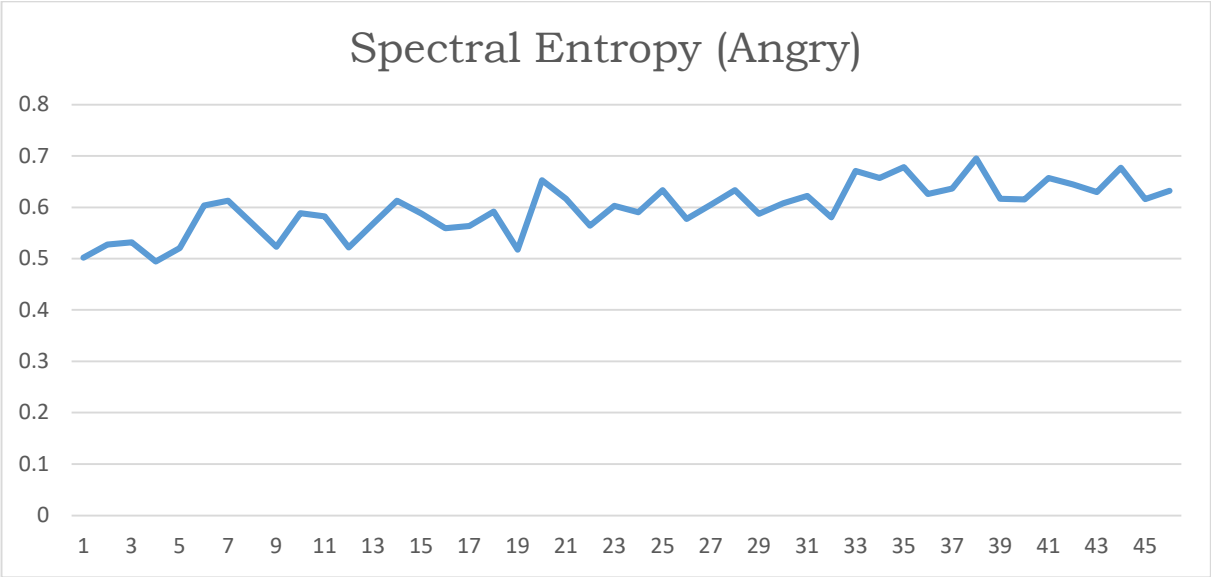

Fig 5.1.2.141: Plotting Spectral Entropy for 'Angry' Emotion Label

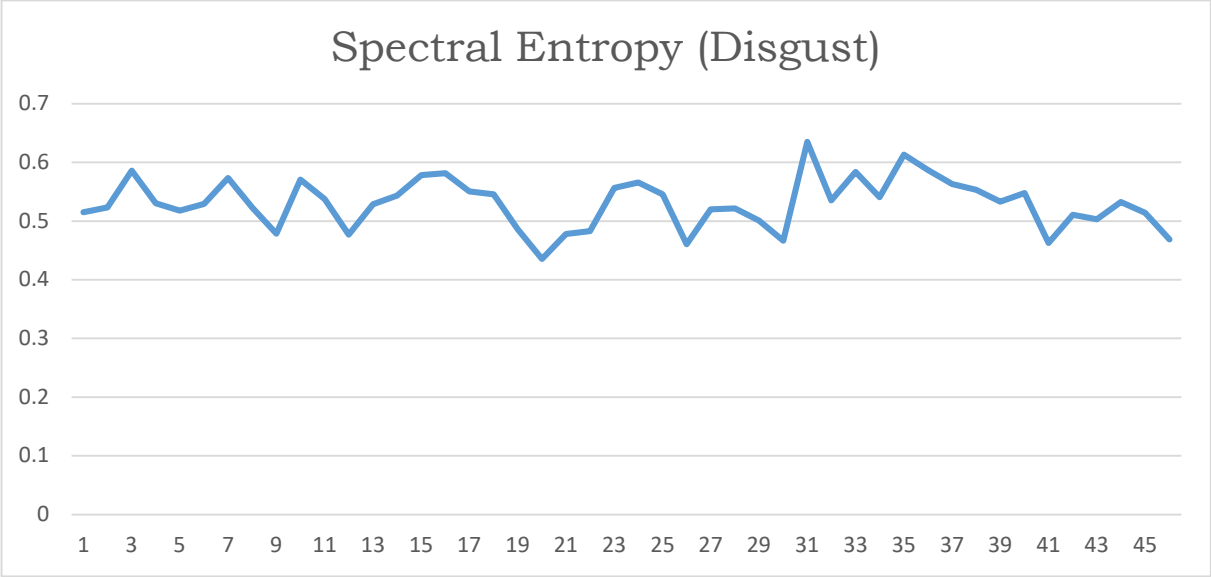

Fig 5.1.2.142: Plotting Spectral Entropy for 'Disgust' Emotion Label

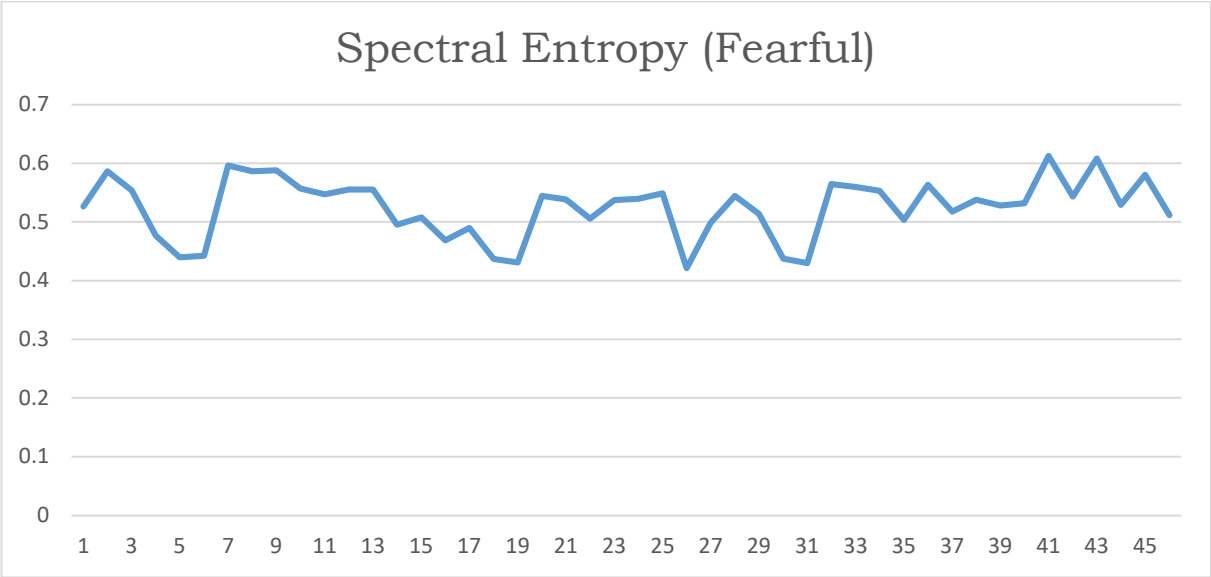

Fig 5.1.2.143: Plotting Spectral Entropy for 'Fearful' Emotion Label

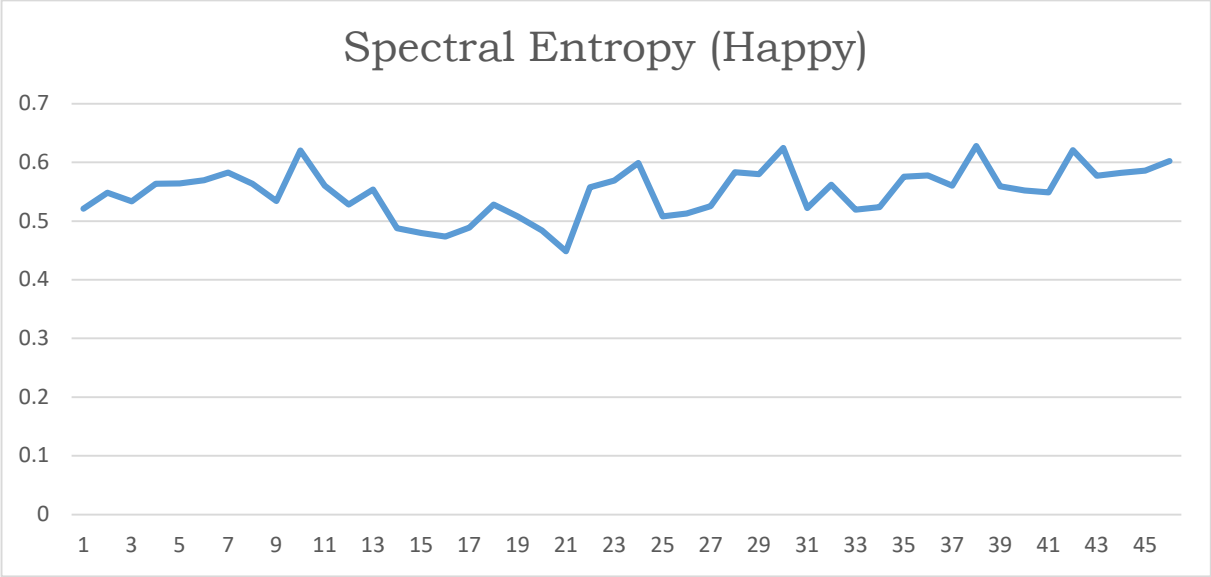

Fig 5.1.2.144: Plotting Spectral Entropy for 'Happy' Emotion Label

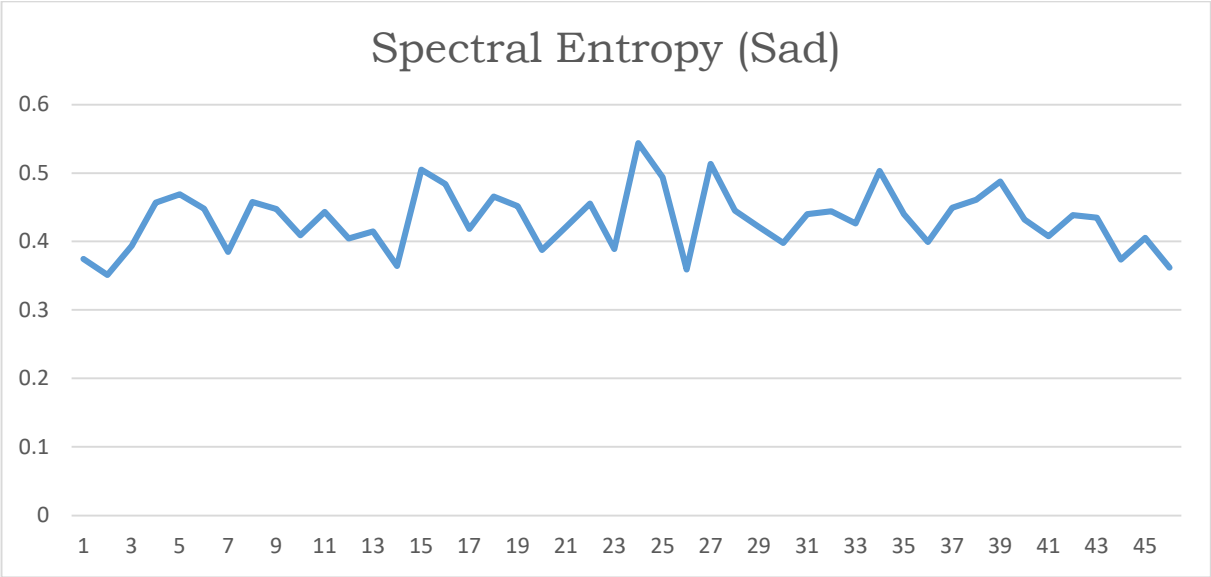

Fig 5.1.2.145: Plotting Spectral Entropy for 'Sad' Emotion Label

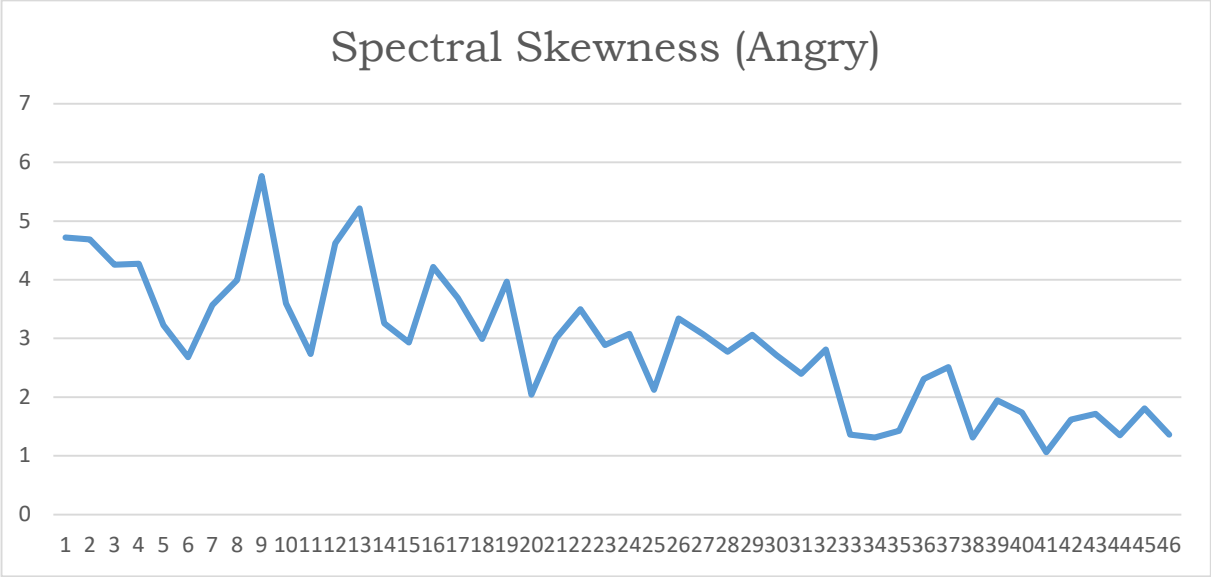

Fig 5.1.2.146: Plotting Spectral Skewness for 'Angry' Emotion Label

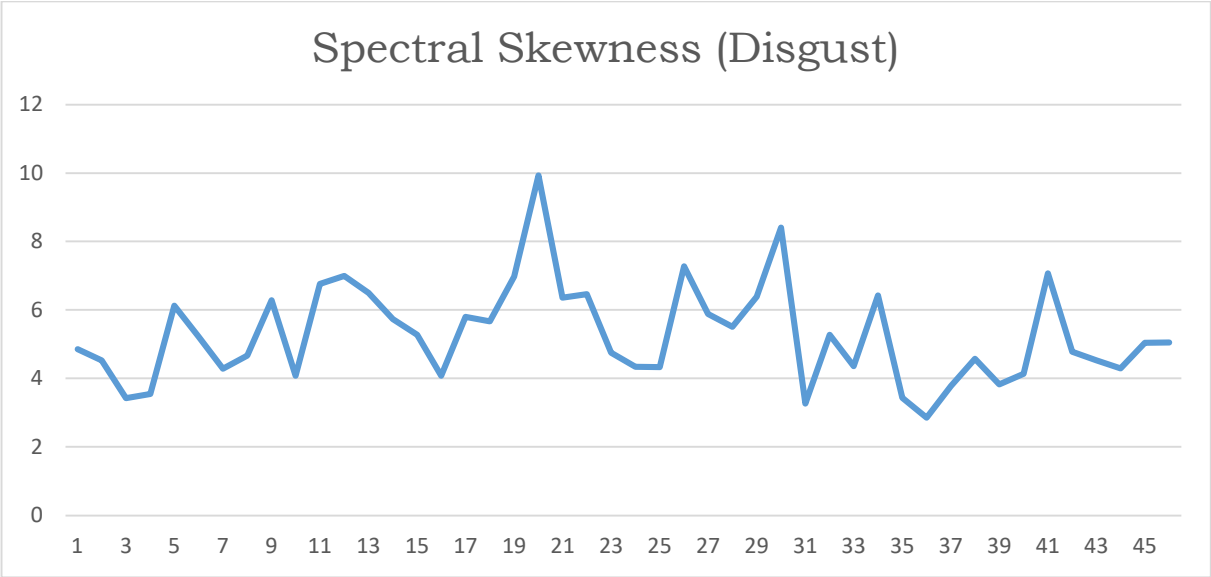

Fig 5.1.2.147: Plotting Spectral Skewness for 'Disgust' Emotion Label

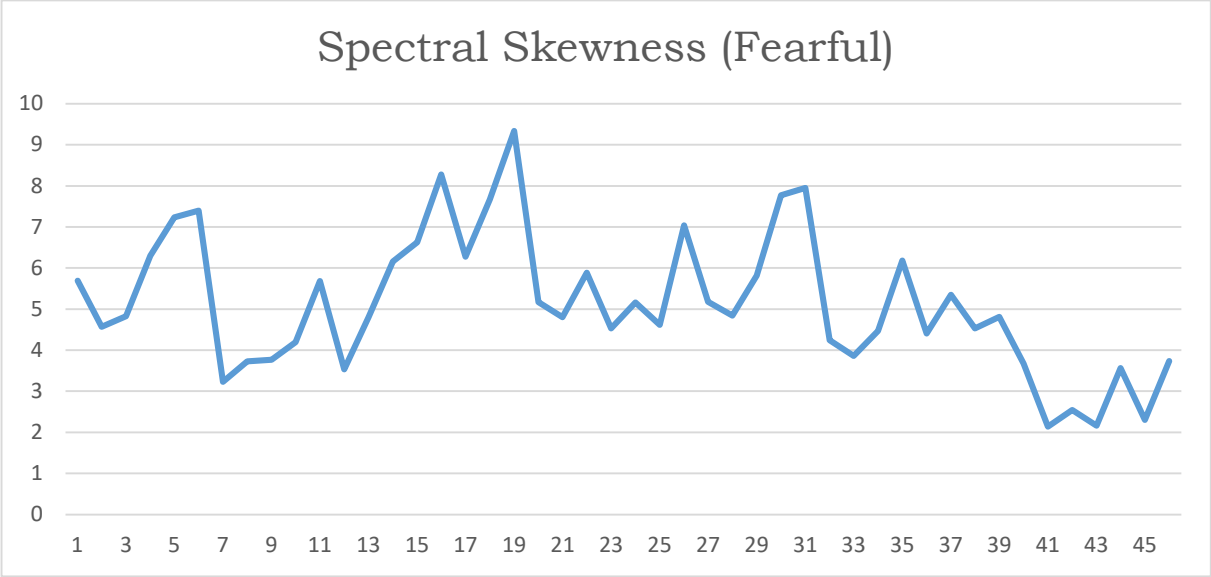

Fig 5.1.2.148: Plotting Spectral Skewness for 'Fearful' Emotion Label

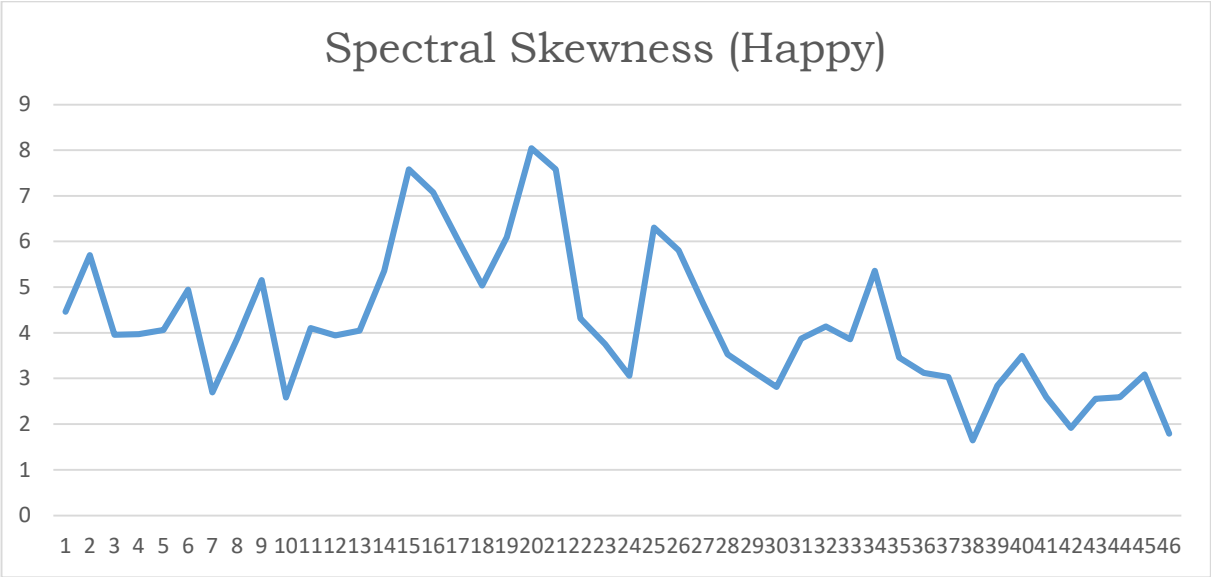

Fig 5.1.2.149: Plotting Spectral Skewness for 'Happy' Emotion Label

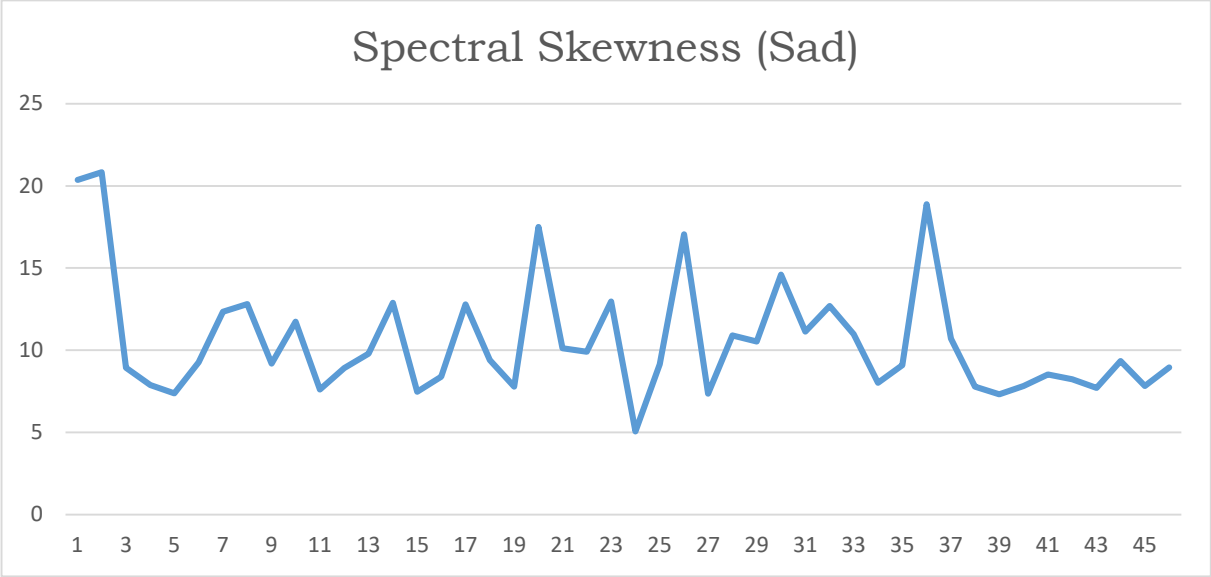

Fig 5.1.2.150: Plotting Spectral Skewness for 'Sad' Emotion Label

Plots of CREMA-D Dataset

Fig 5.1.2.151 – Fig 5.1.2.195 showcase plots of the variations in emotional expressions across all features within the CREMA-D dataset. These visual representations provide insights into how different features respond to various emotions, offering a deeper understanding of the dataset's emotional dynamics.

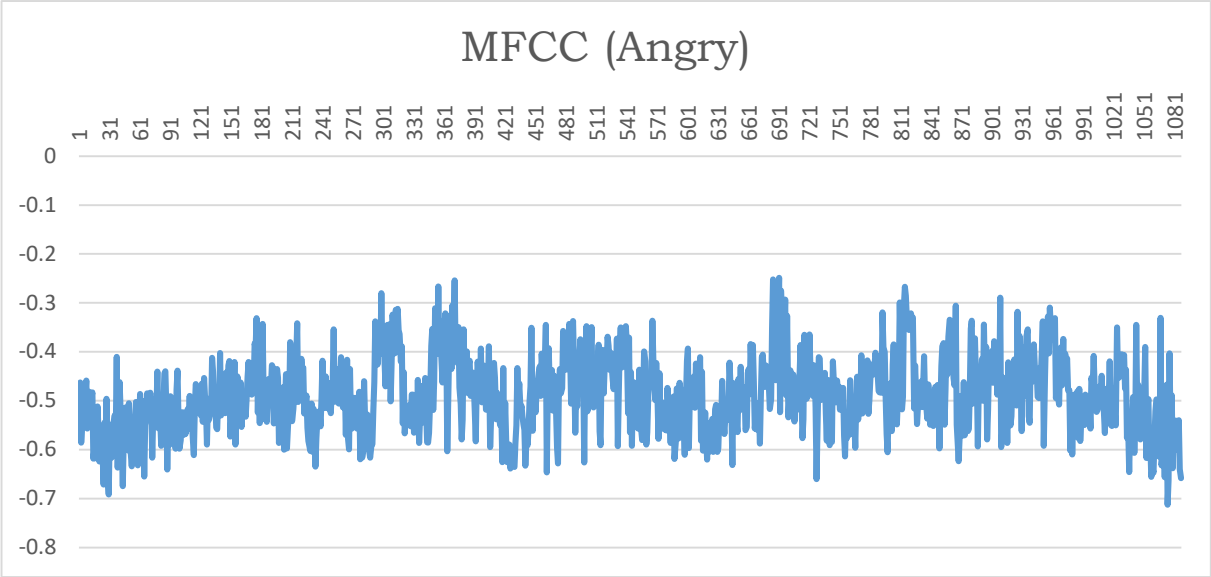

Fig 5.1.2.151: Plotting MFCC for 'Angry' Emotion Label

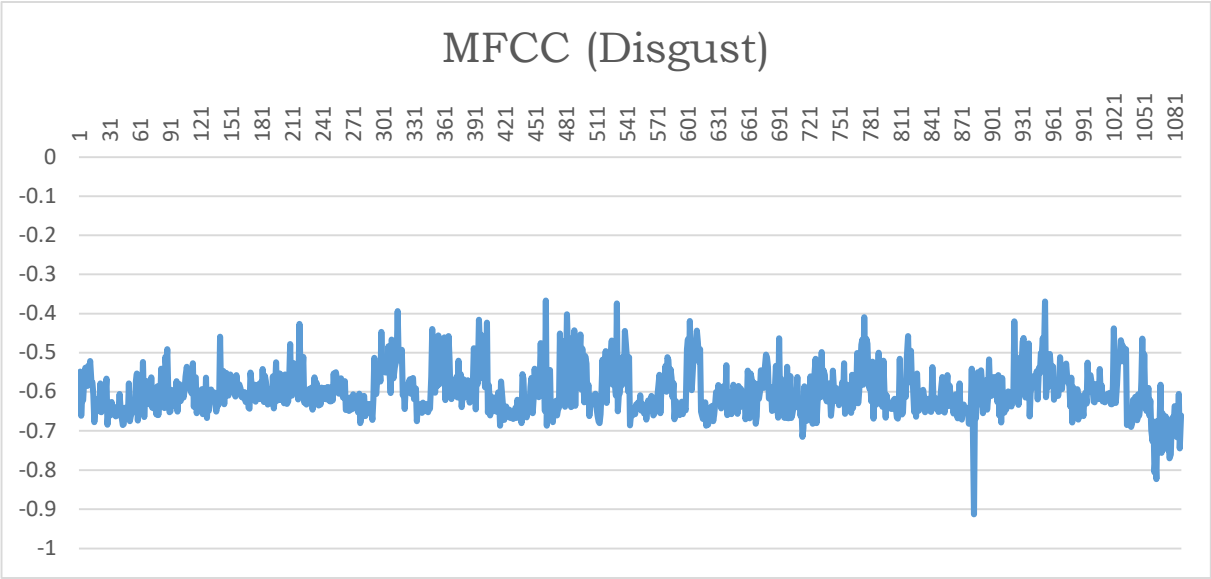

Fig 5.1.2.152: Plotting MFCC for 'Disgust' Emotion Label

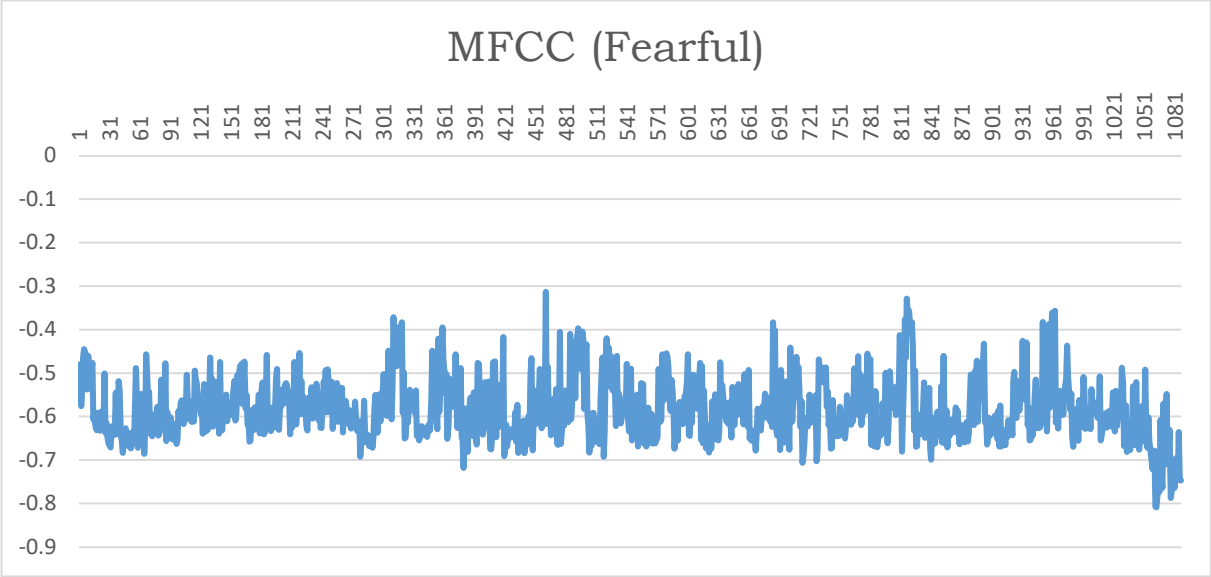

Fig 5.1.2.153: Plotting MFCC for 'Fearful' Emotion Label

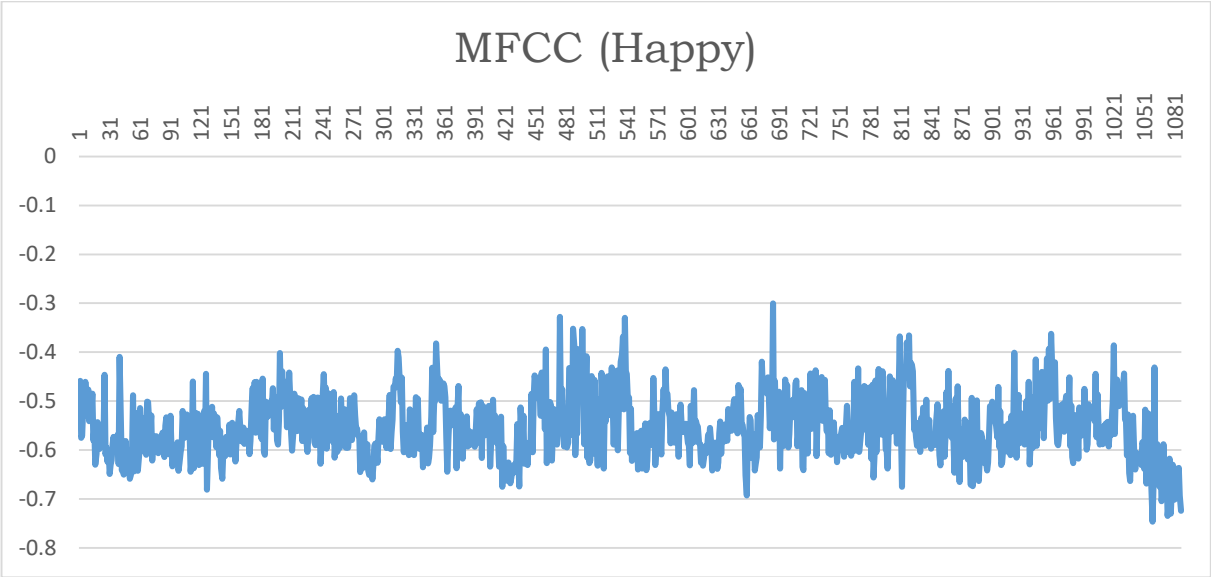

Fig 5.1.2.154: Plotting MFCC for 'Happy' Emotion Label

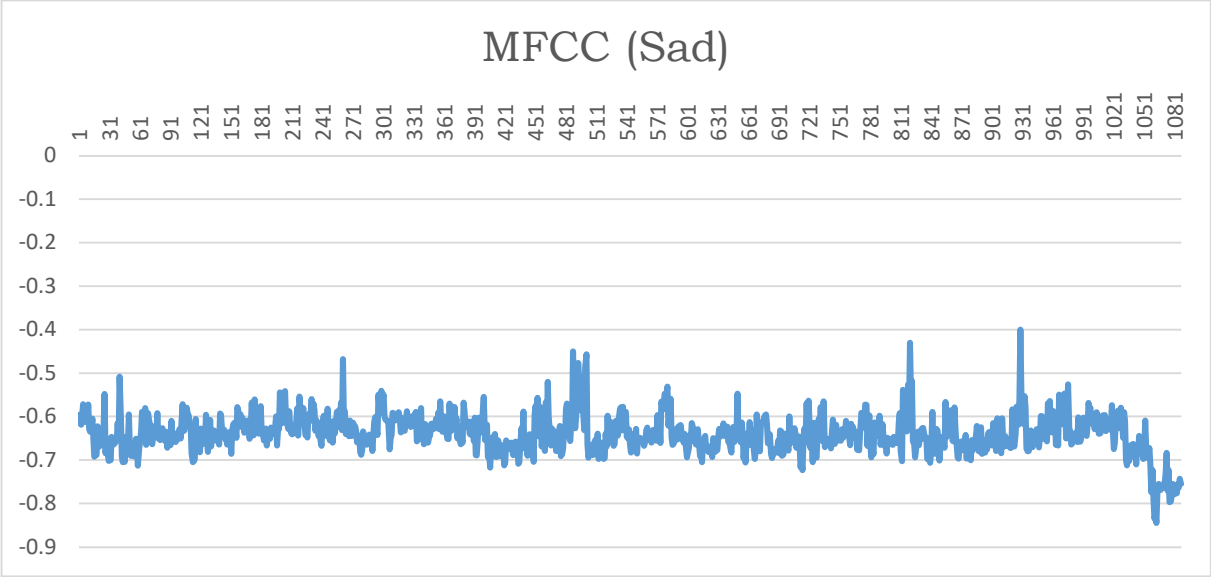

Fig 5.1.2.155: Plotting MFCC for 'Sad' Emotion Label

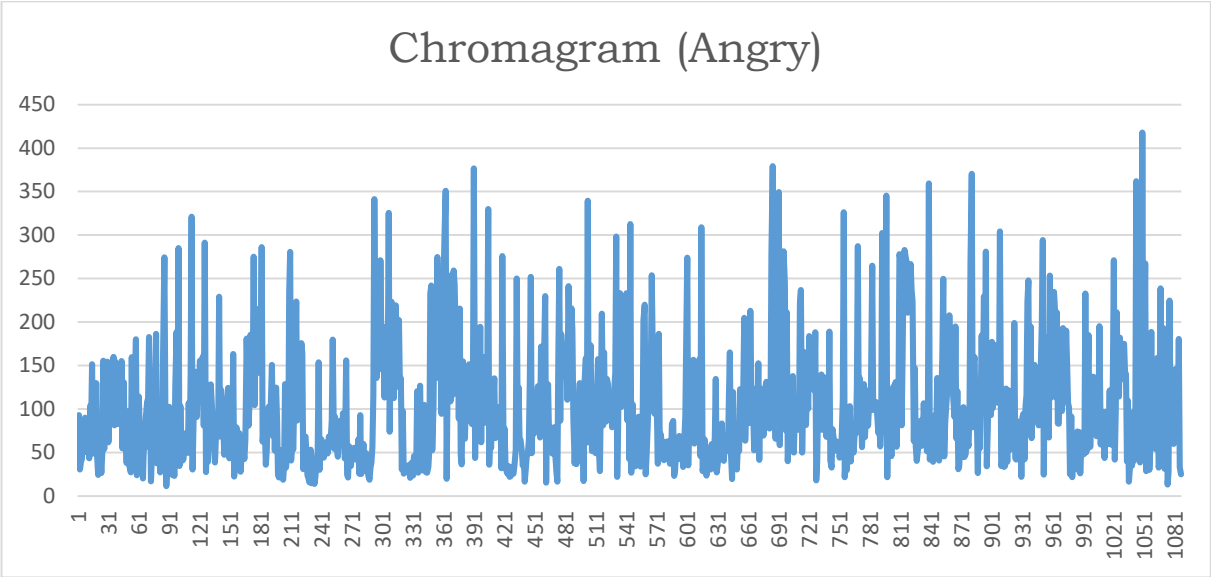

Fig 5.1.2.156: Plotting Chromagram for 'Angry' Emotion Label

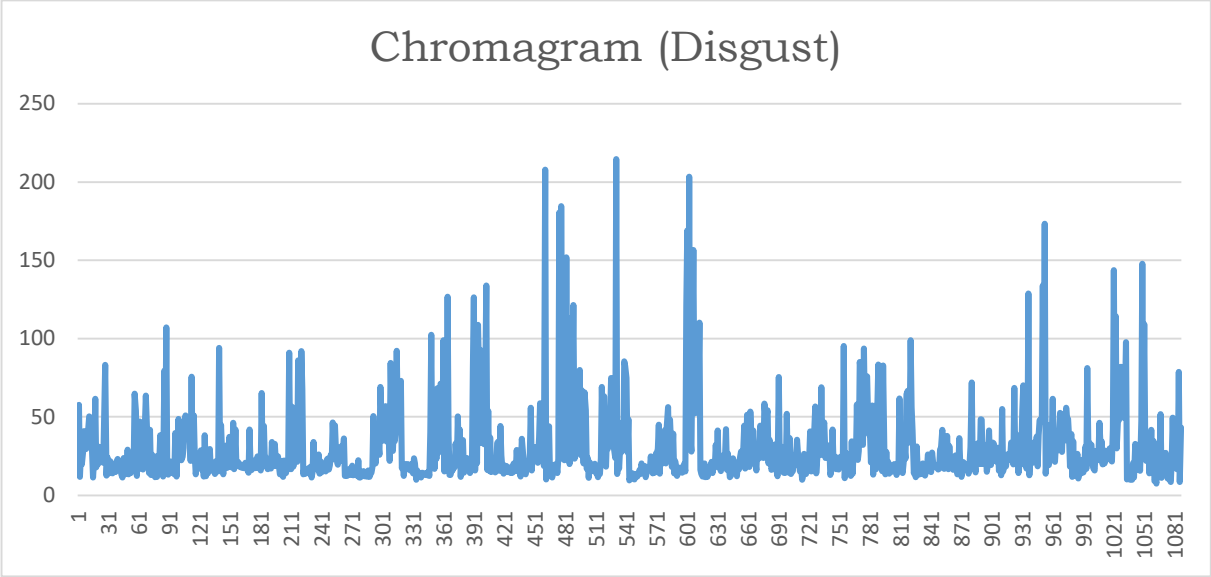

Fig 5.1.2.157: Plotting Chromagram for 'Disgust' Emotion Label

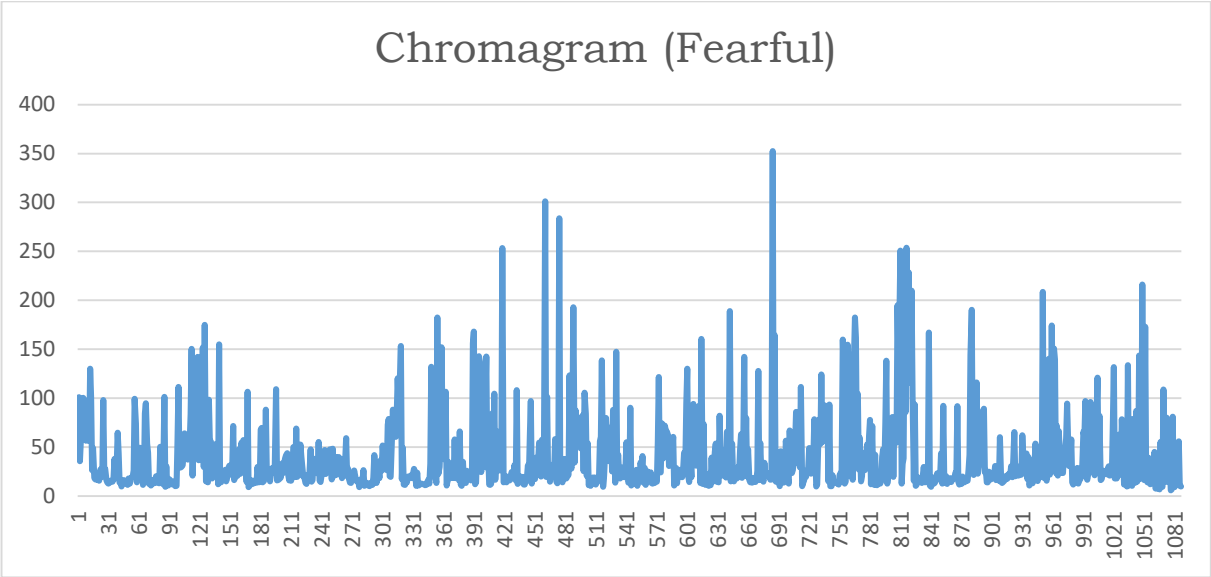

Fig 5.1.2.158: Plotting Chromagram for 'Fearful' Emotion Label

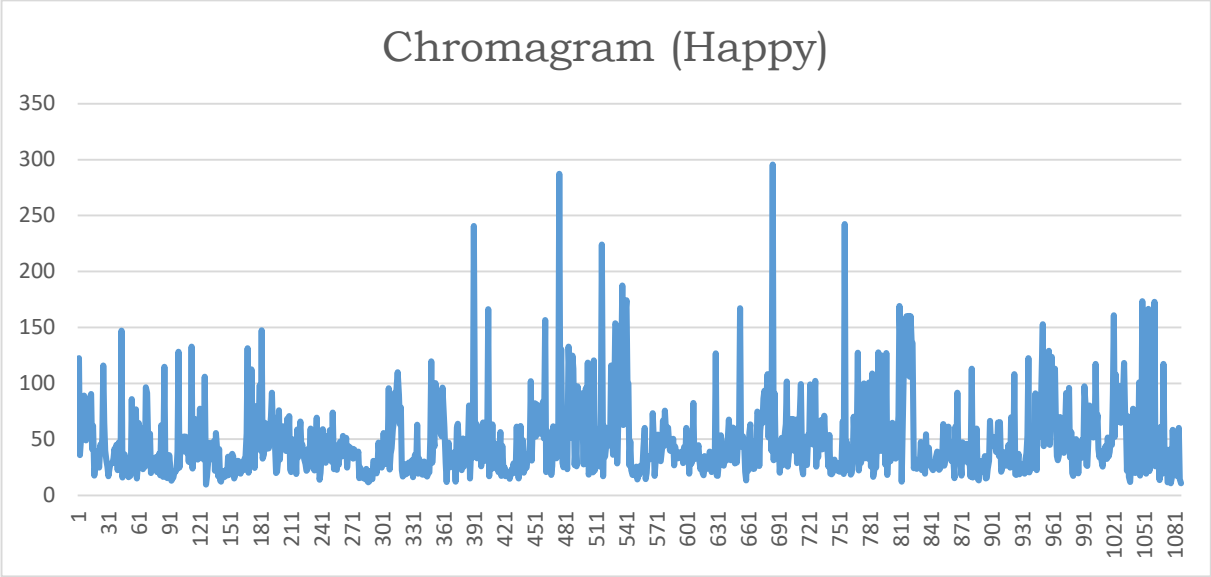

Fig 5.1.2.159: Plotting Chromagram for 'Happy' Emotion Label

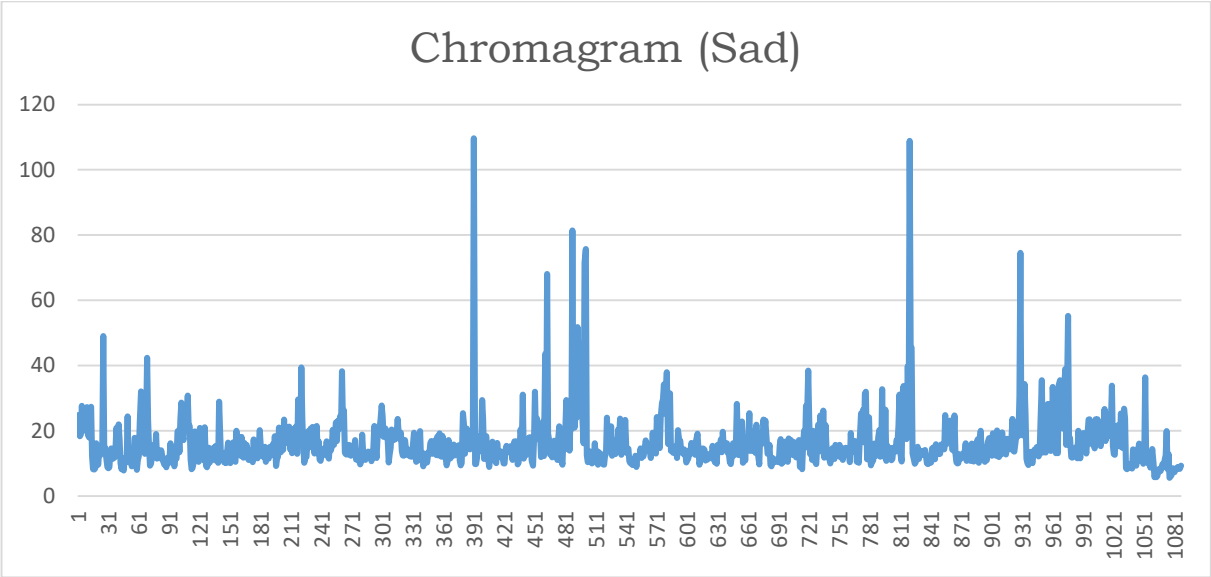

Fig 5.1.2.160: Plotting Chromagram for 'Sad' Emotion Label

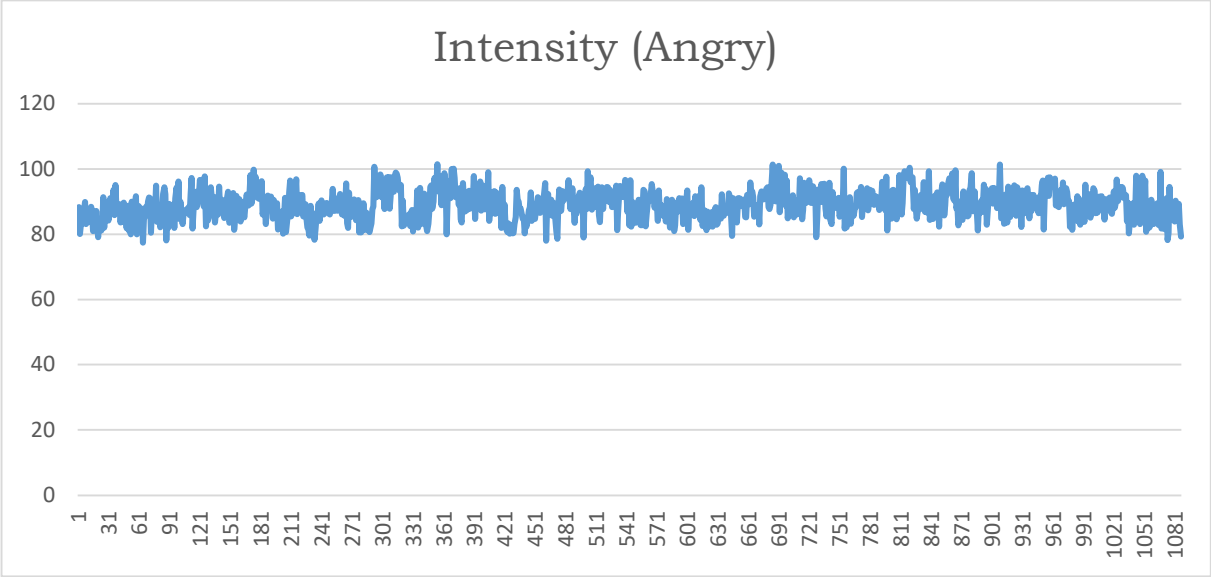

Fig 5.1.2.161: Plotting Intensity for 'Angry' Emotion Label

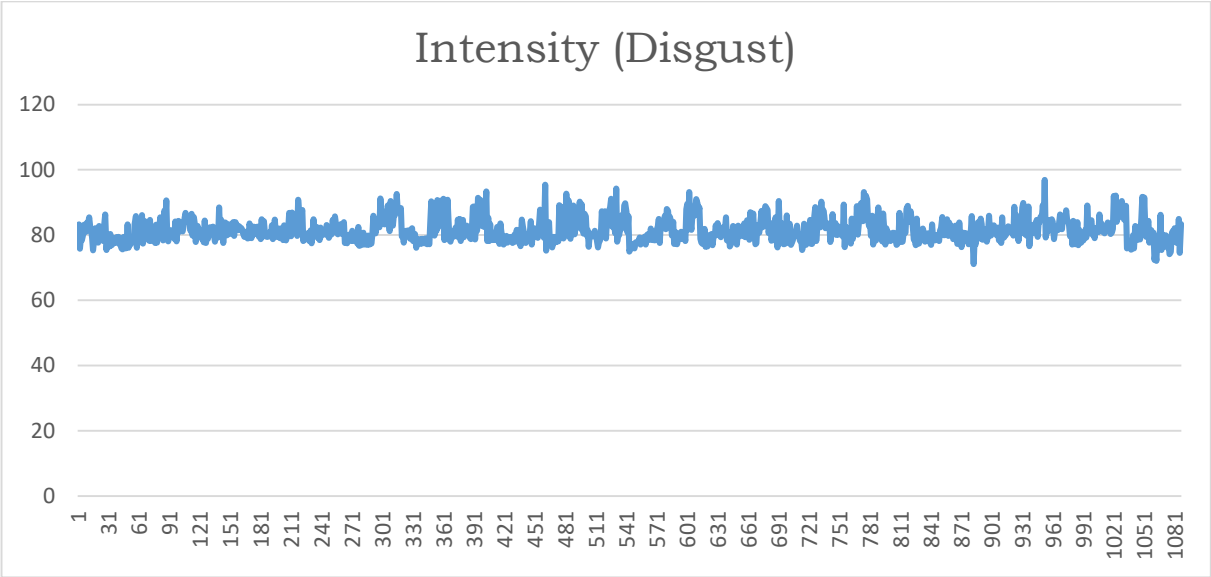

Fig 5.1.2.162: Plotting Intensity for 'Disgust' Emotion Label

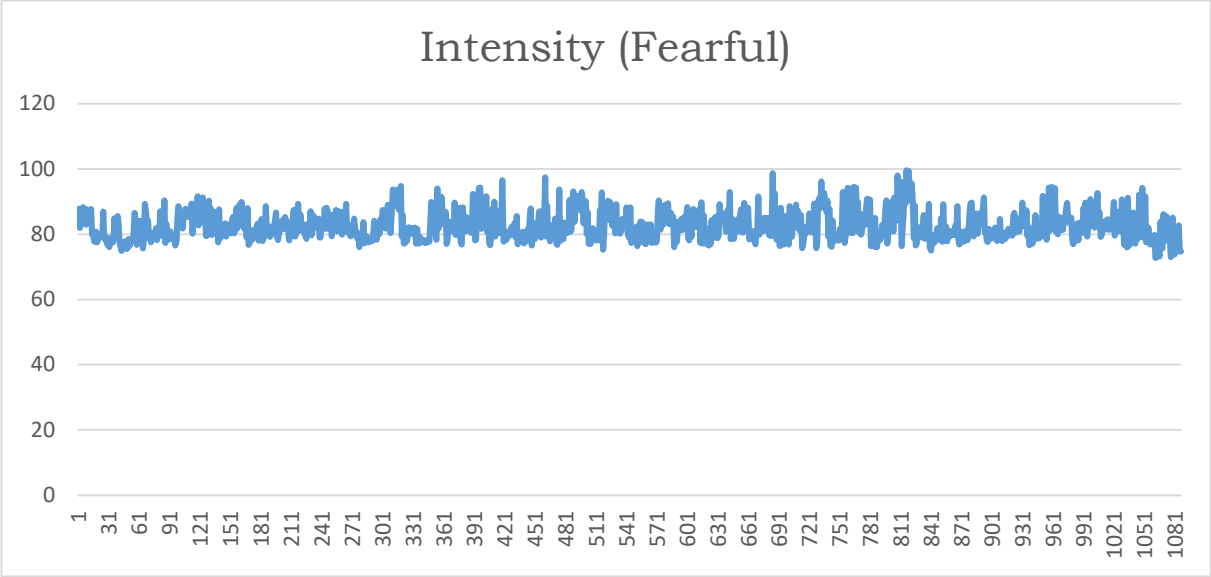

Fig 5.1.2.163: Plotting Intensity for 'Fearful' Emotion Label

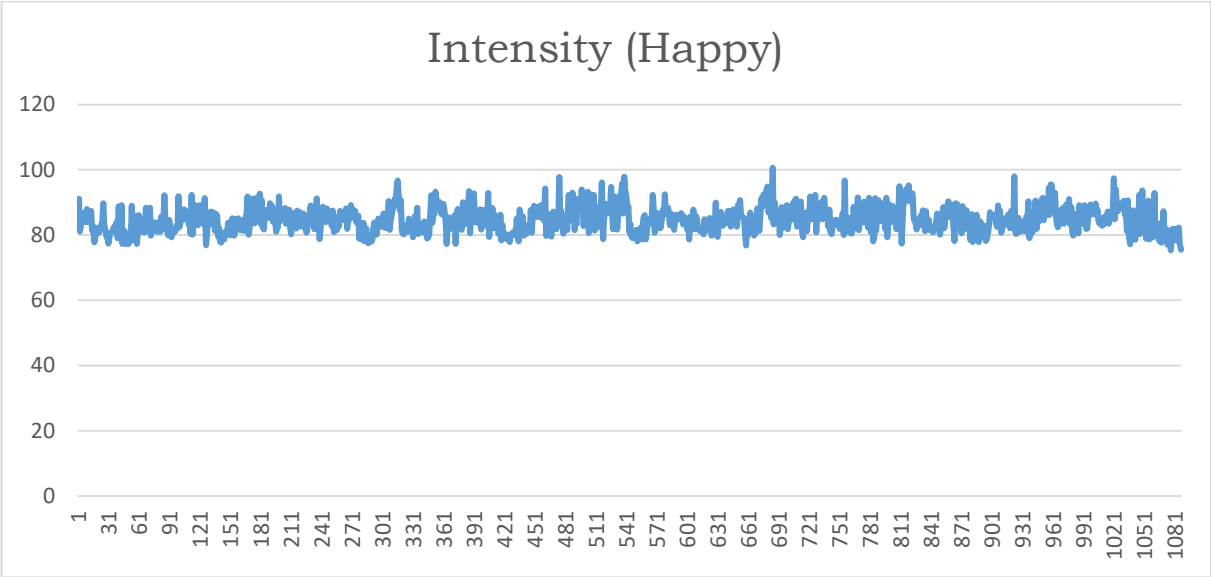

Fig 5.1.2.164: Plotting Intensity for 'Happy' Emotion Label

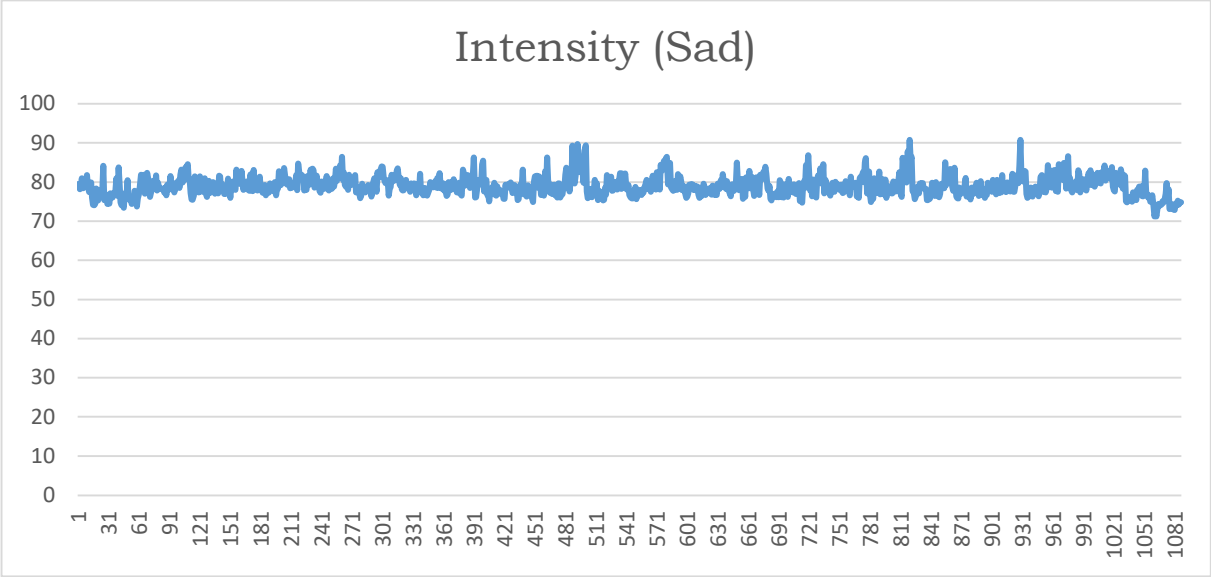

Fig 5.1.2.165: Plotting Intensity for 'Sad' Emotion Label

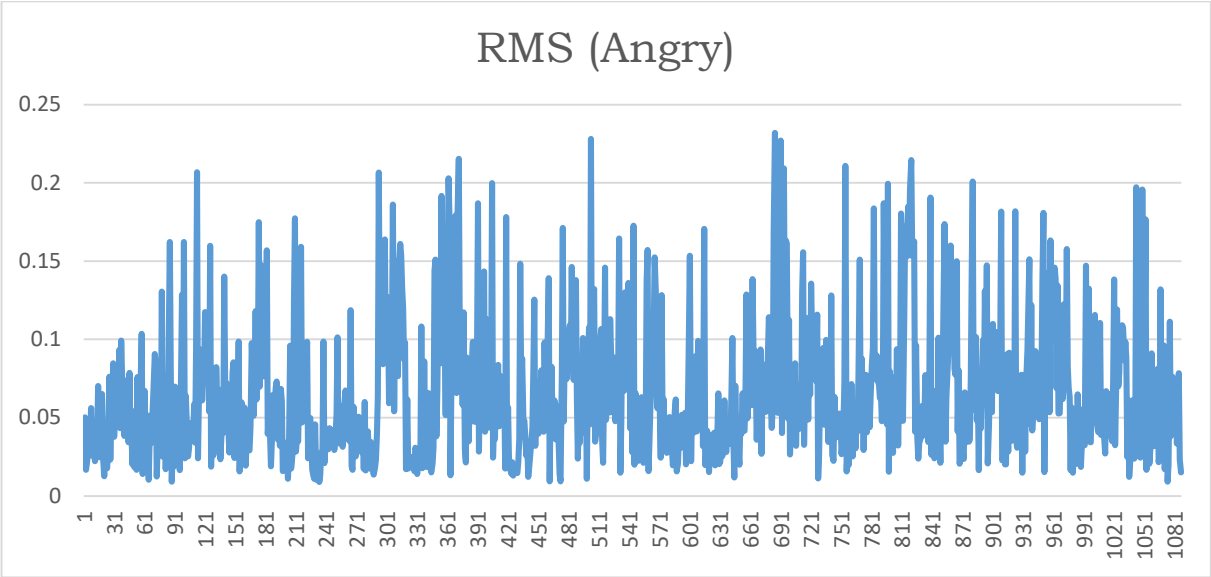

Fig 5.1.2.166: Plotting RMS for 'Angry' Emotion Label

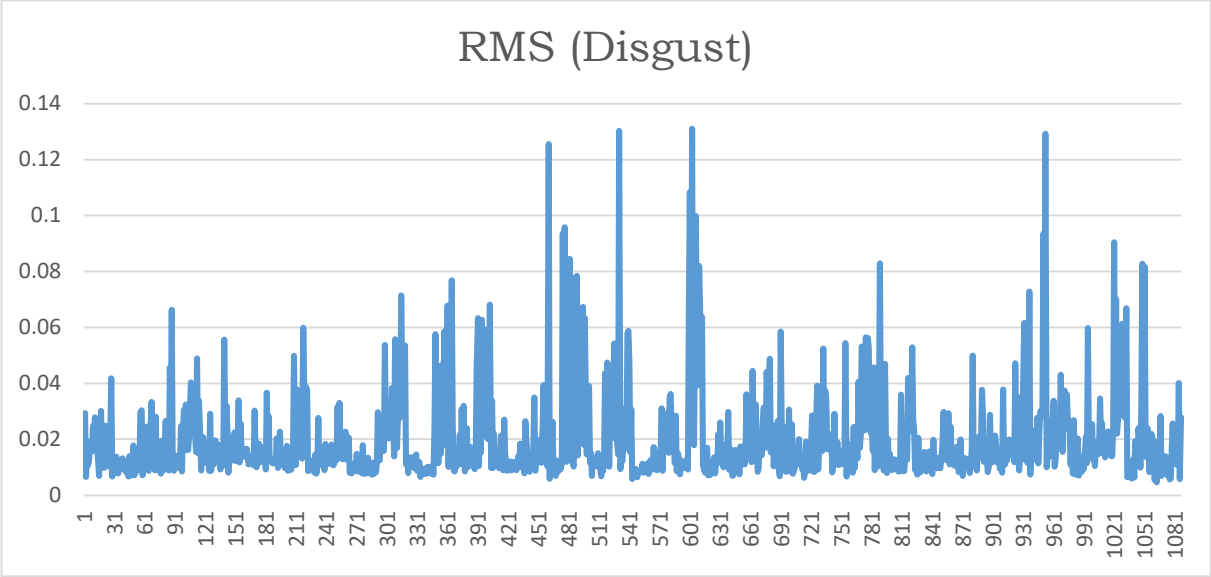

Fig 5.1.2.167: Plotting RMS for 'Disgust' Emotion Label

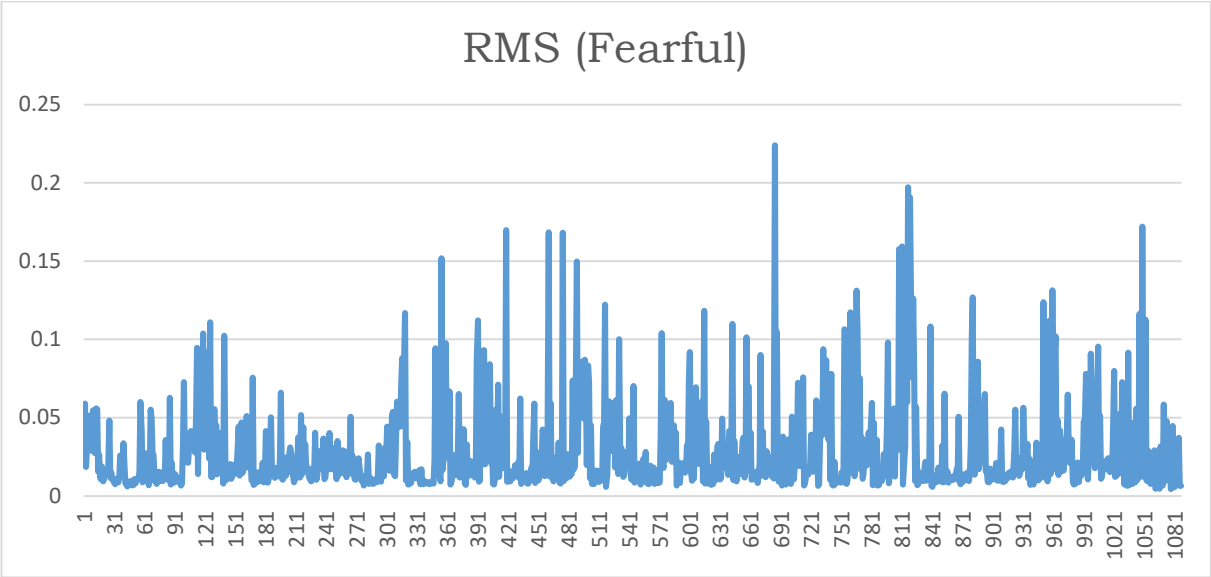

Fig 5.1.2.168: Plotting RMS for 'Fearful' Emotion Label

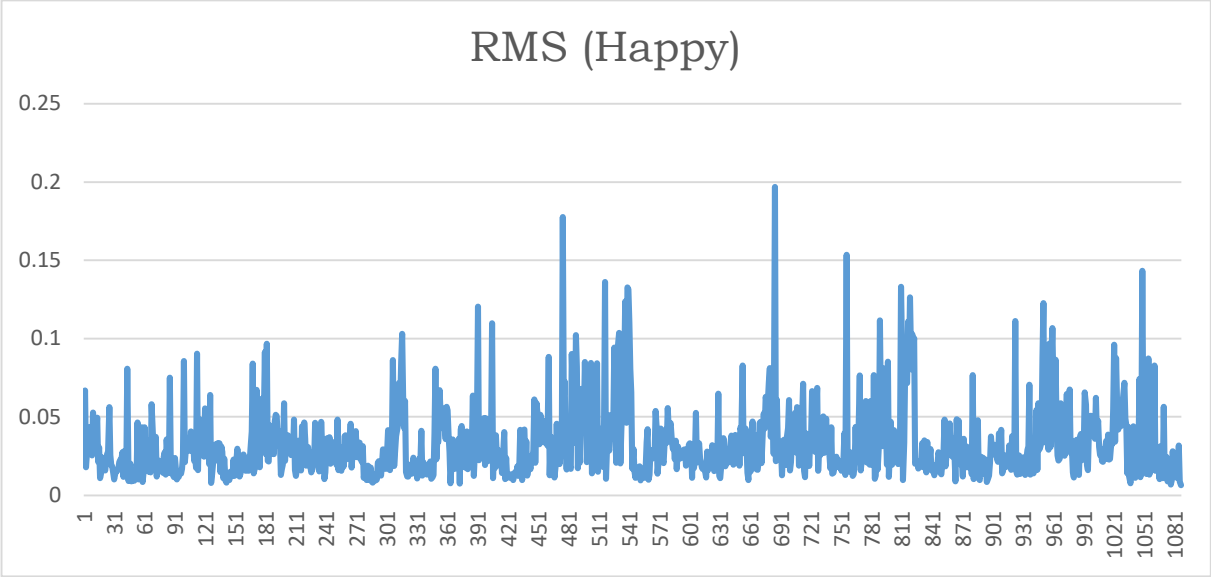

Fig 5.1.2.169: Plotting RMS for 'Happy' Emotion Label

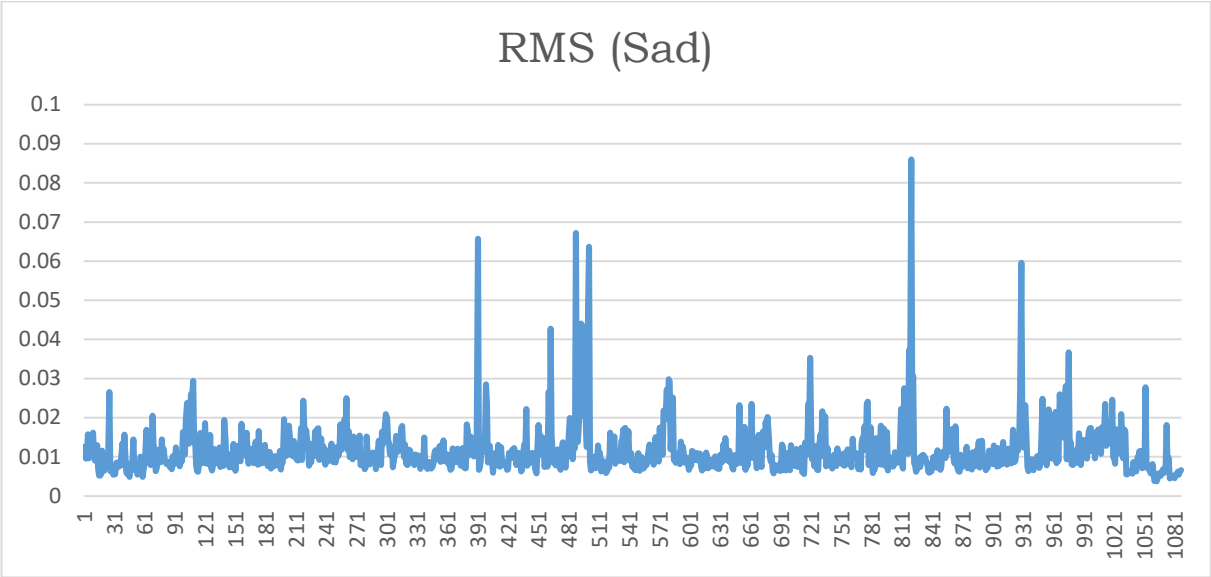

Fig 5.1.2.170: Plotting RMS for 'Sad' Emotion Label

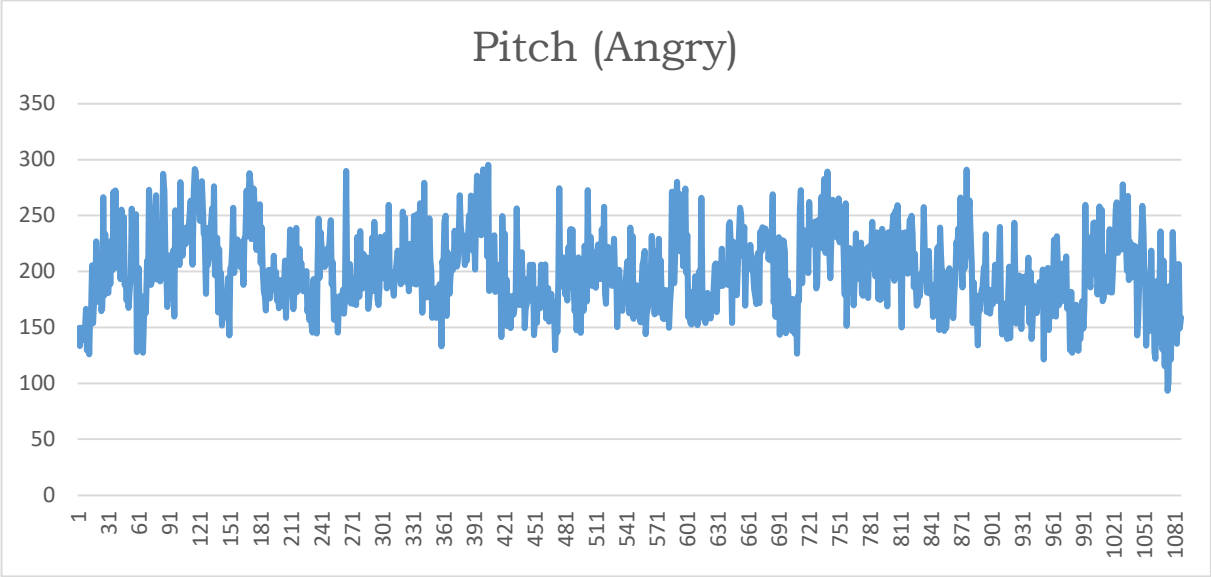

Fig 5.1.2.171: Plotting Pitch for 'Angry' Emotion Label

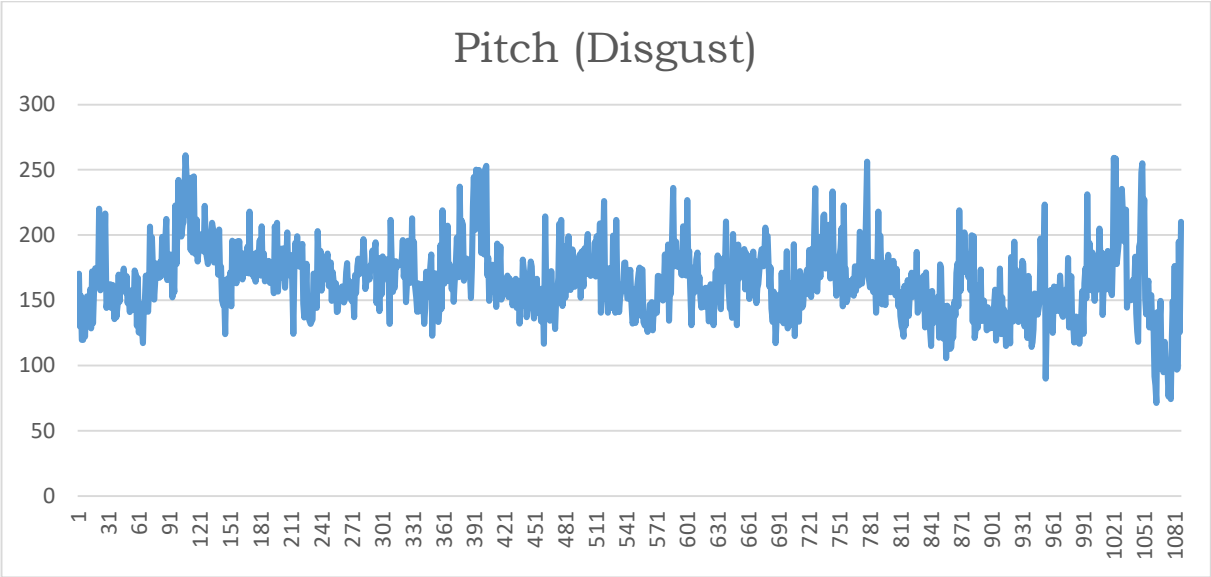

Fig 5.1.2.172: Plotting Pitch for 'Disgust' Emotion Label

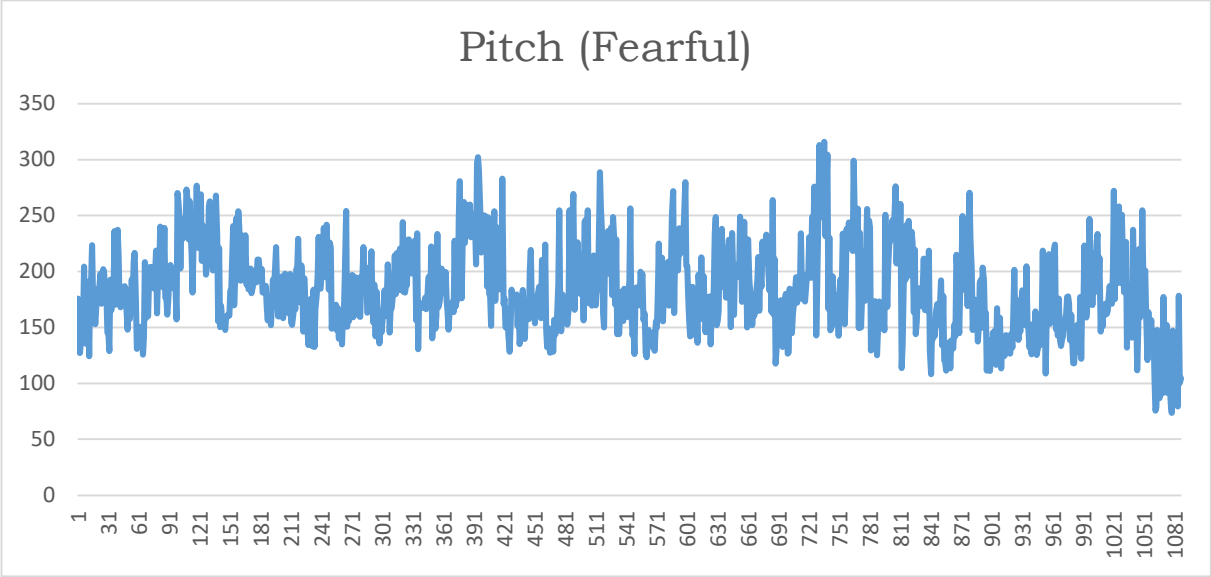

Fig 5.1.2.173: Plotting Pitch for 'Fearful' Emotion Label

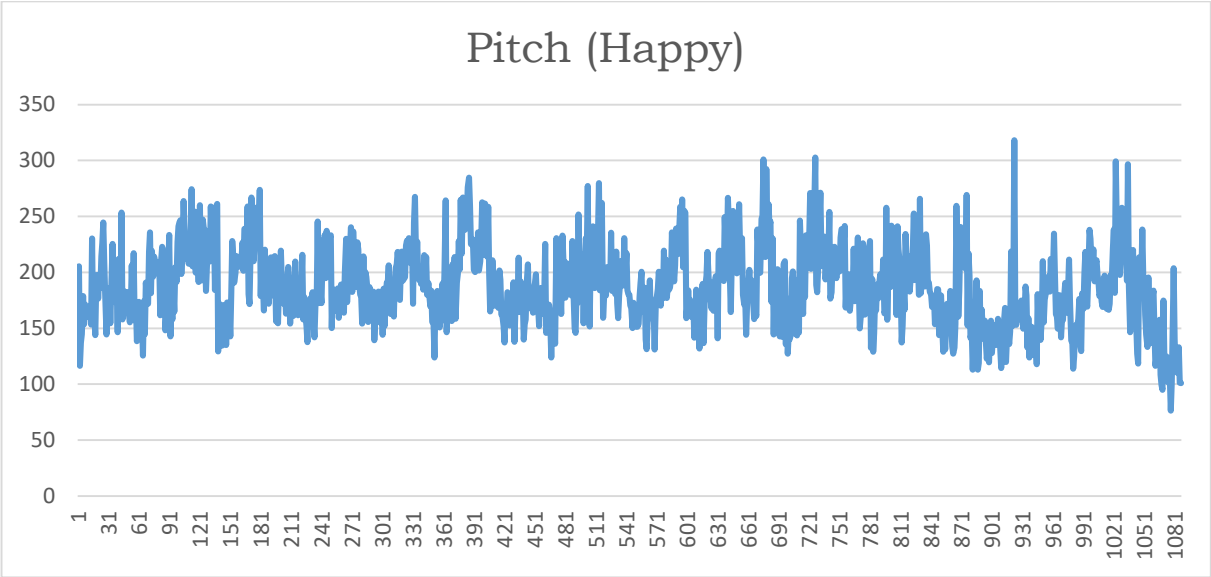

Fig 5.1.2.174: Plotting Pitch for 'Happy' Emotion Label

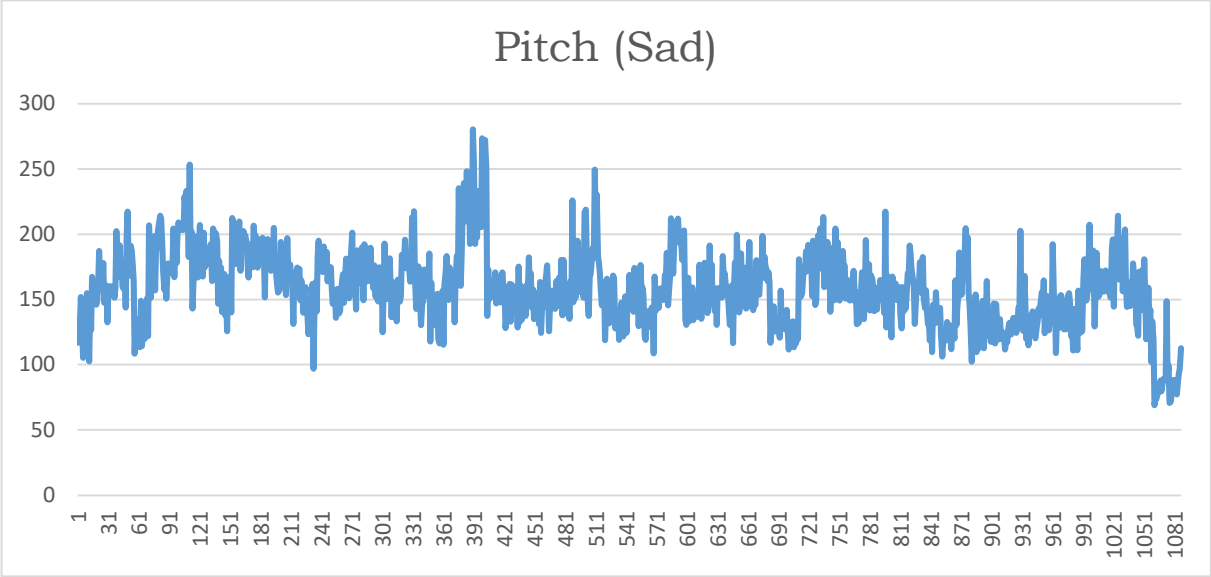

Fig 5.1.2.175: Plotting Pitch for 'Sad' Emotion Label

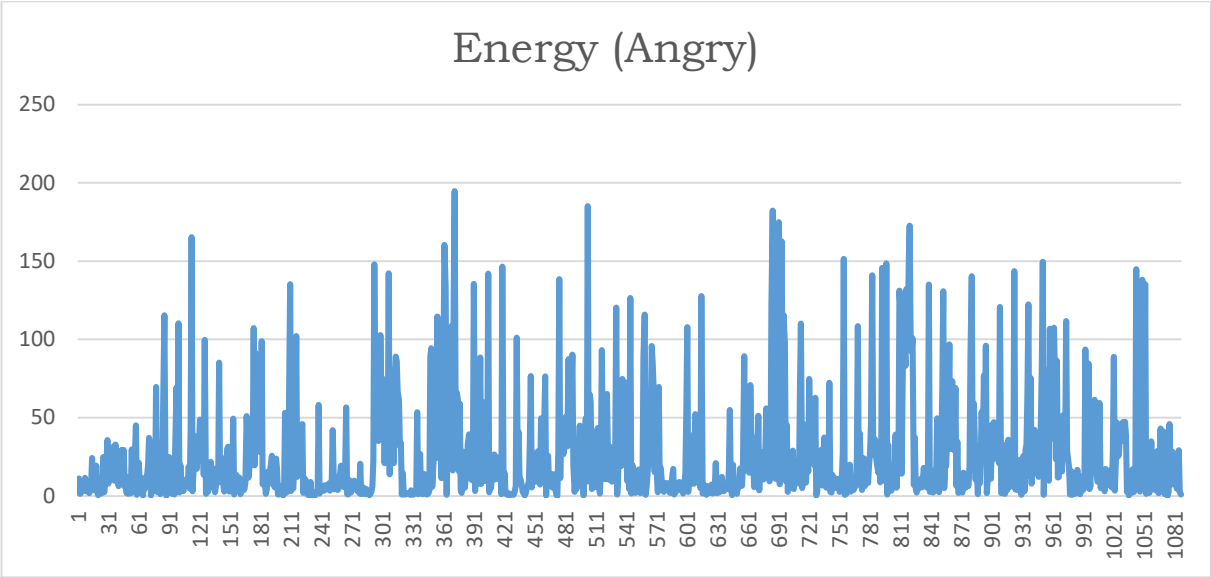

Fig 5.1.2.176: Plotting Energy for 'Angry' Emotion Label

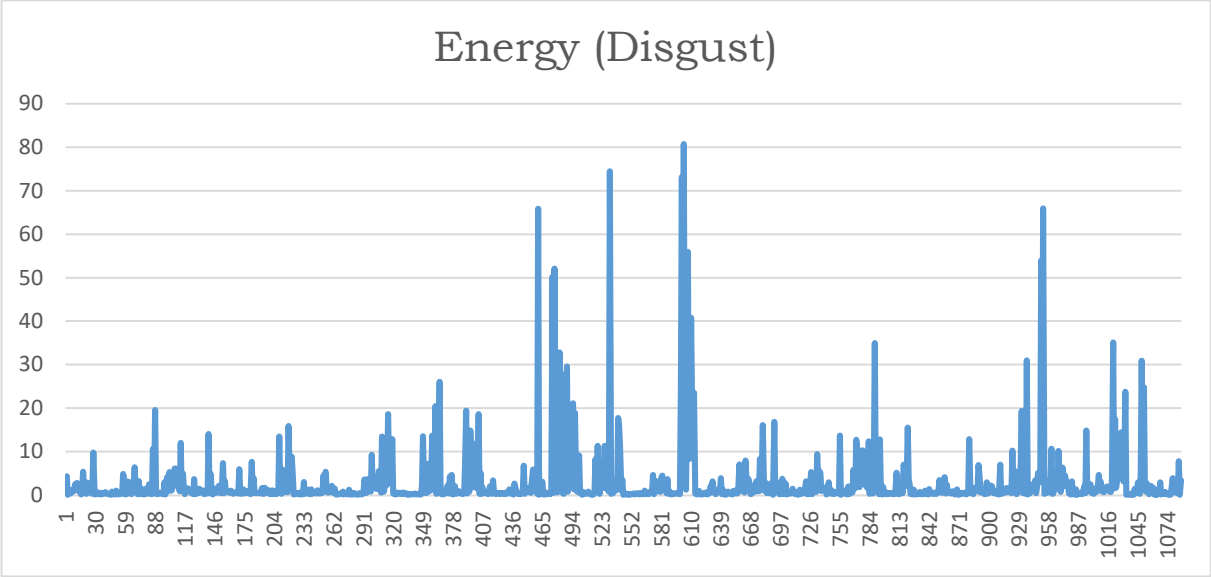

Fig 5.1.2.177: Plotting Energy for 'Disgust' Emotion Label

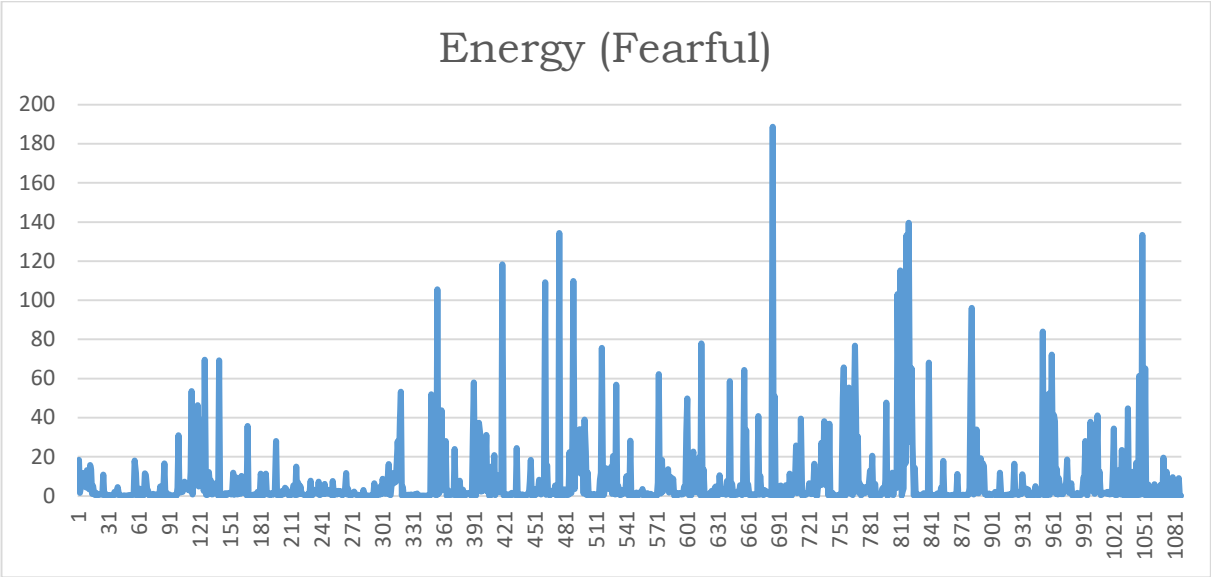

Fig 5.1.2.178: Plotting Energy for 'Fearful' Emotion Label

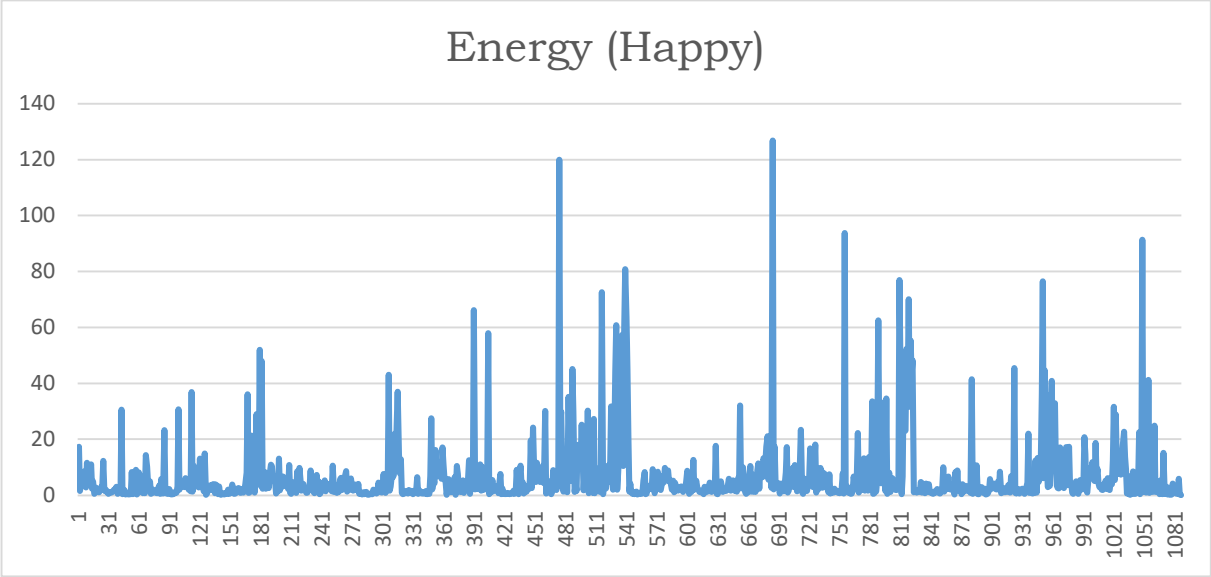

Fig 5.1.2.179: Plotting Energy for 'Happy' Emotion Label

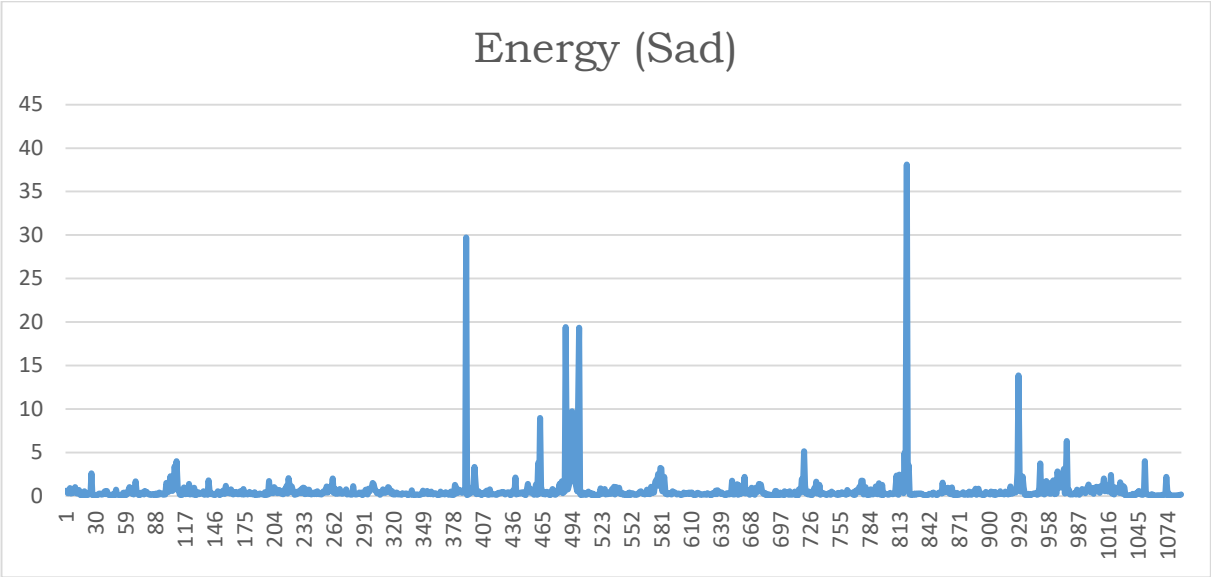

Fig 5.1.2.180: Plotting Energy for 'Sad' Emotion Label

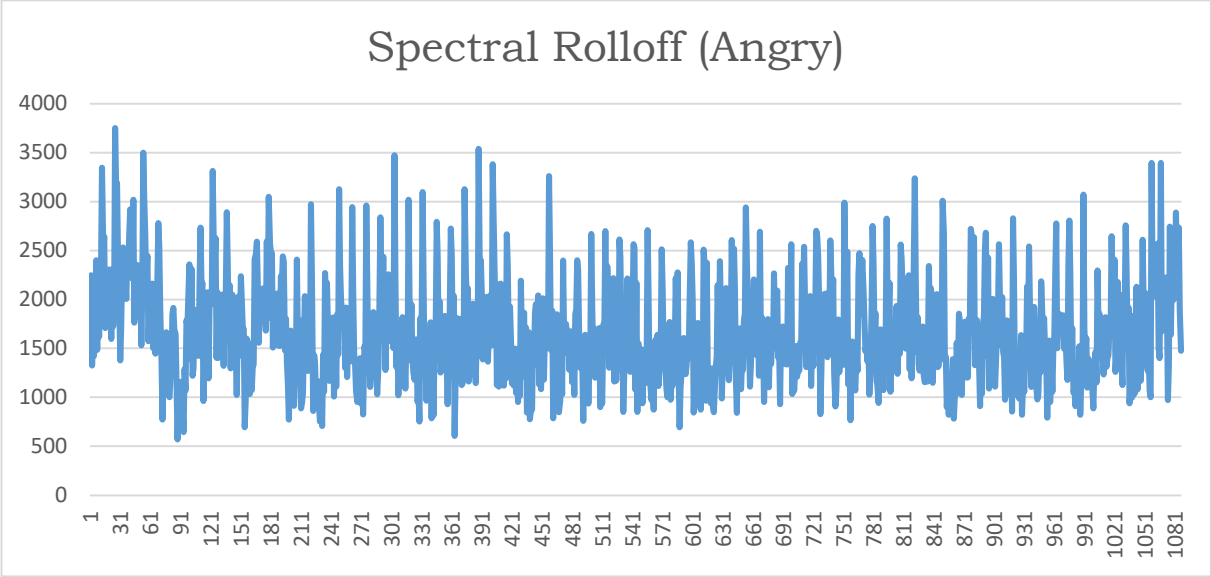

Fig 5.1.2.181: Plotting Spectral Rolloff for 'Angry' Emotion Label

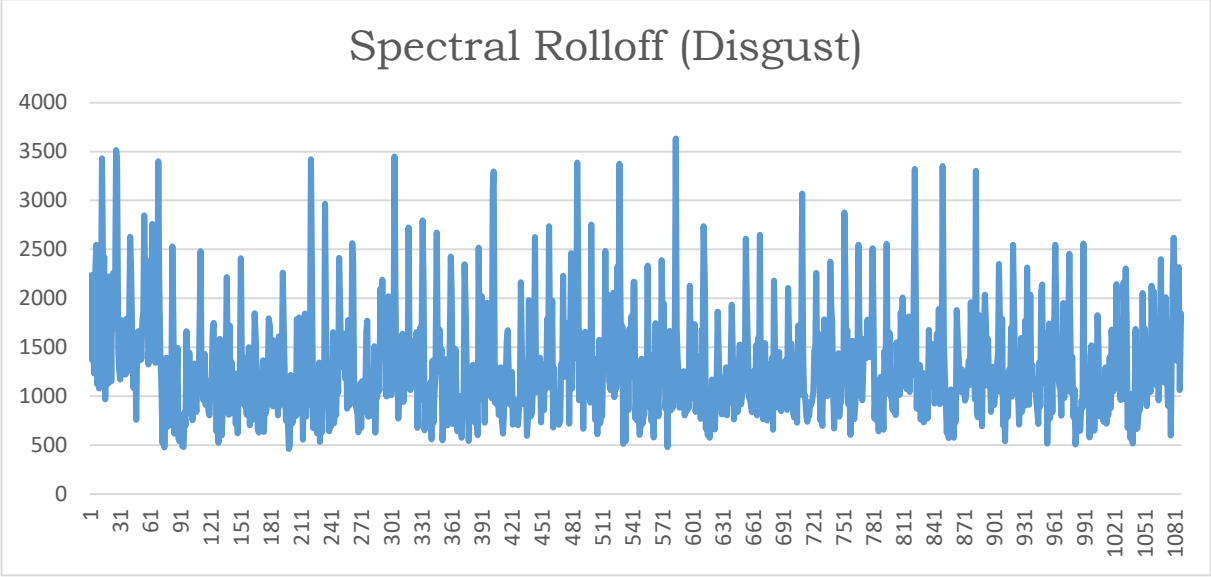

Fig 5.1.2.182: Plotting Spectral Rolloff for 'Disgust' Emotion Label

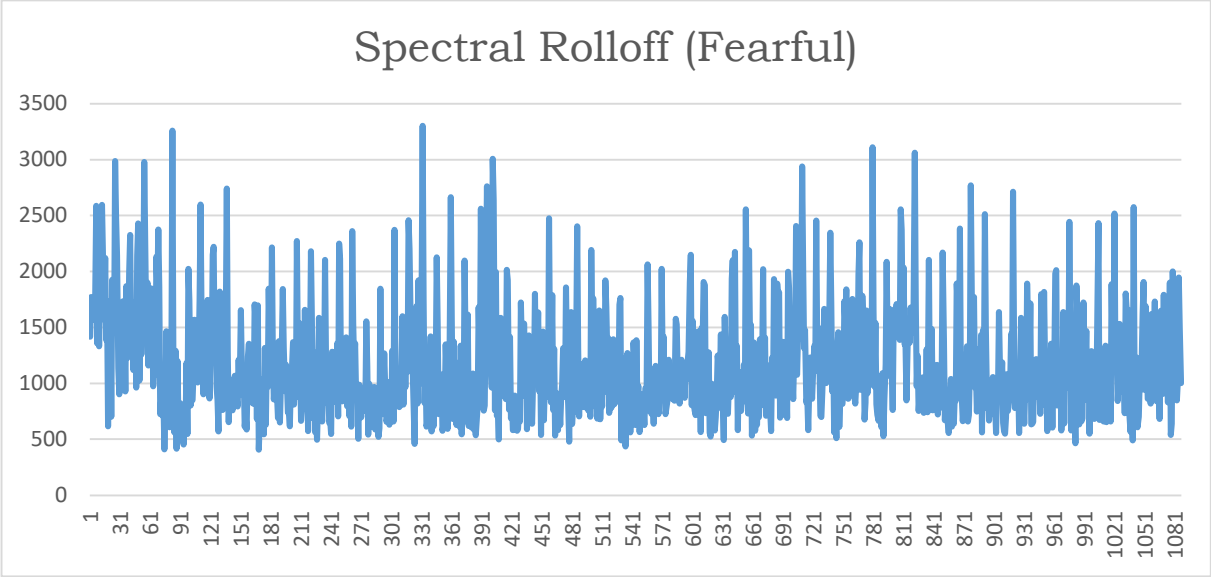

Fig 5.1.2.183: Plotting Spectral Rolloff for 'Fearful' Emotion Label

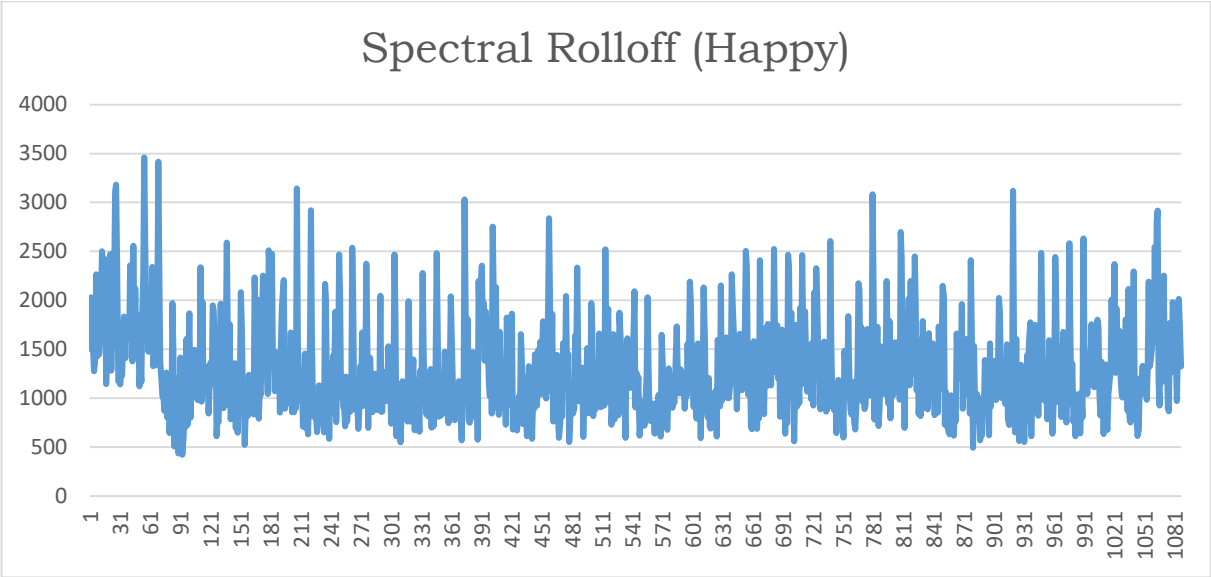

Fig 5.1.2.184: Plotting Spectral Rolloff for 'Happy' Emotion Label

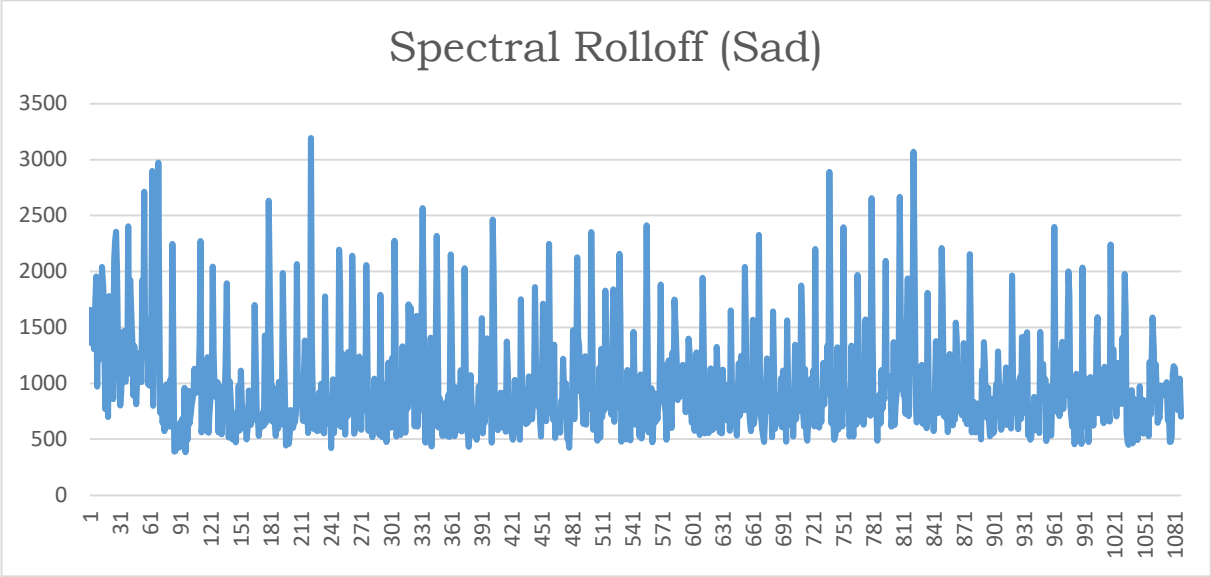

Fig 5.1.2.185: Plotting Spectral Rolloff for 'Sad' Emotion Label

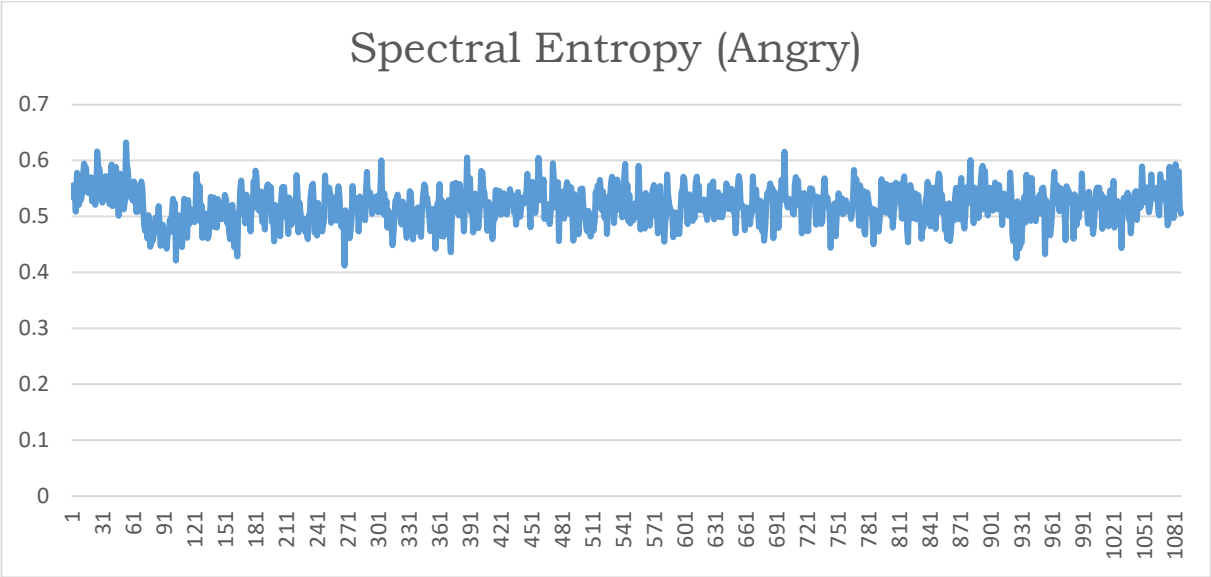

Fig 5.1.2.186: Plotting Spectral Entropy for 'Angry' Emotion Label

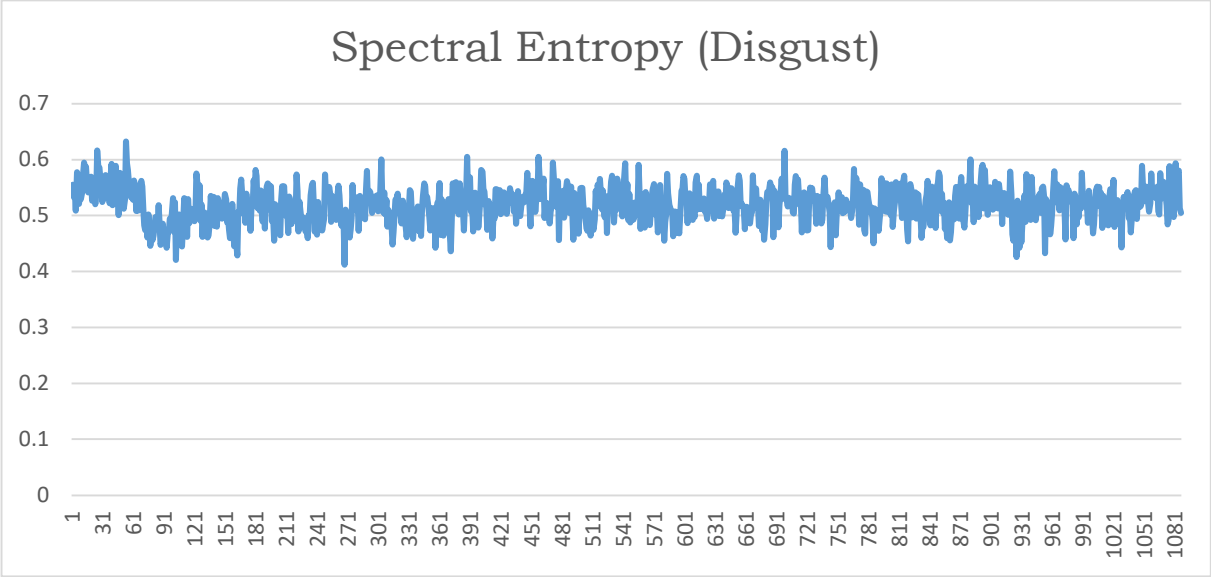

Fig 5.1.2.187: Plotting Spectral Entropy for 'Disgust' Emotion Label

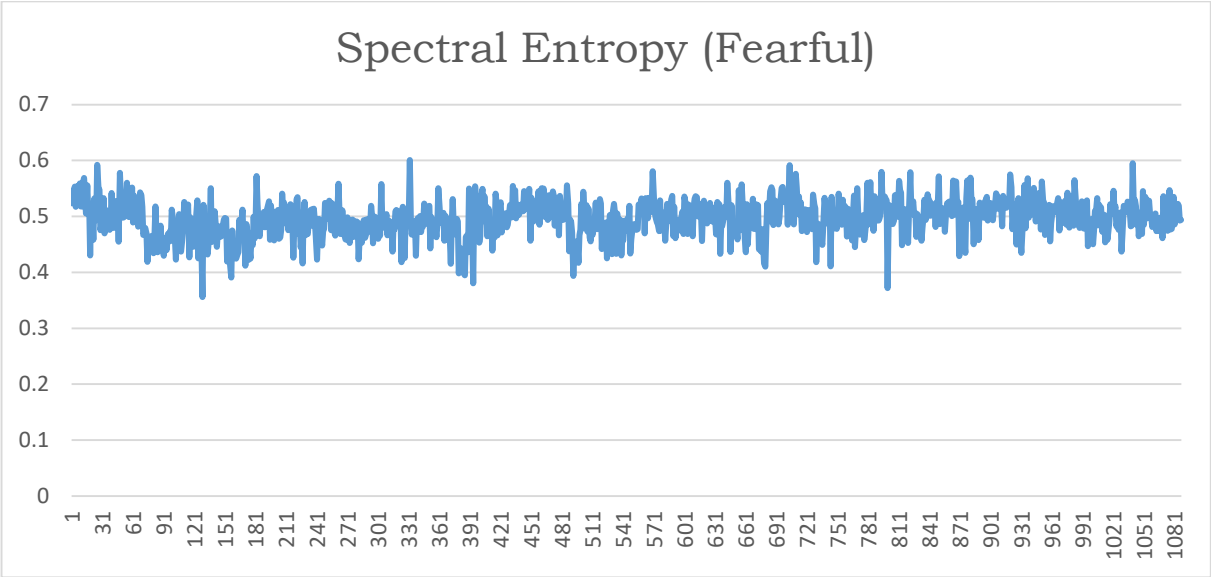

Fig 5.1.2.188: Plotting Spectral Entropy for 'Fearful' Emotion Label

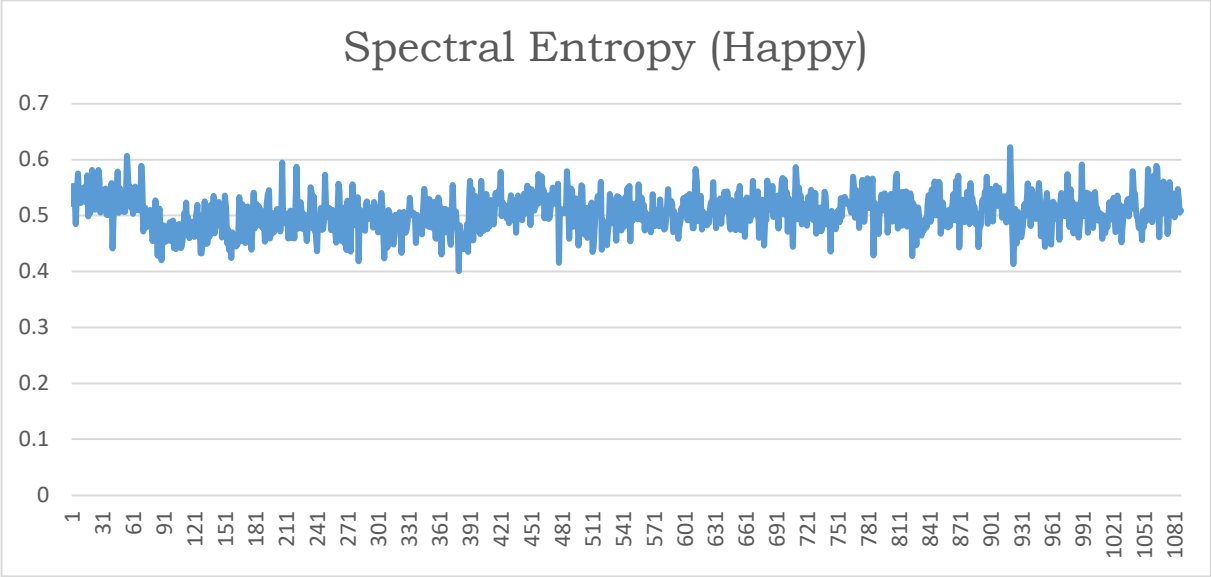

Fig 5.1.2.189: Plotting Spectral Entropy for 'Happy' Emotion Label

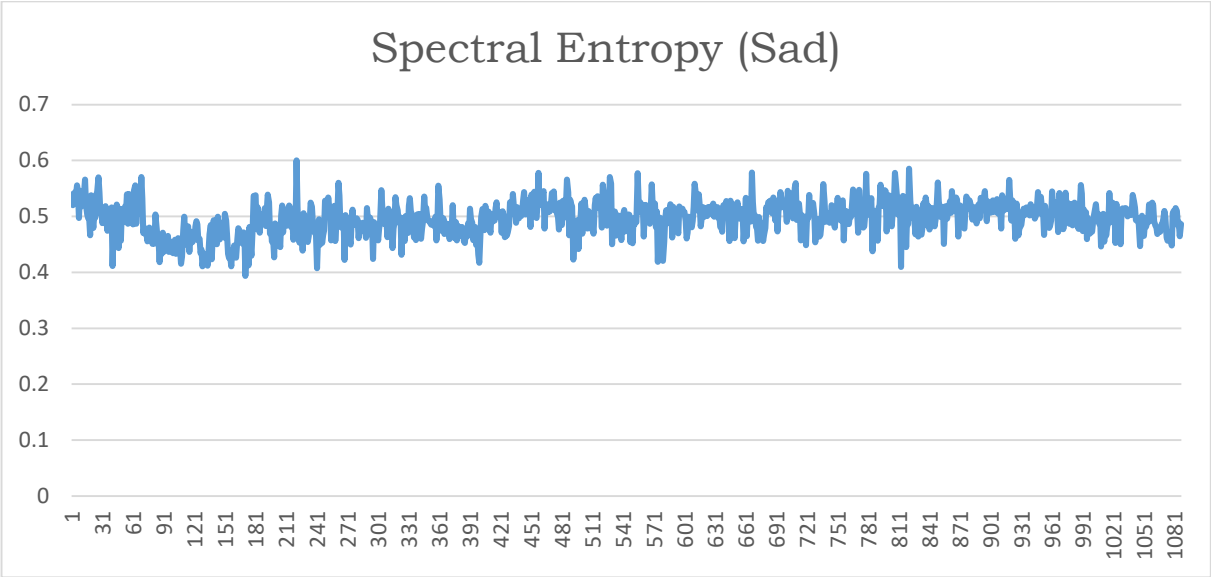

Fig 5.1.2.190: Plotting Spectral Entropy for 'Sad' Emotion Label

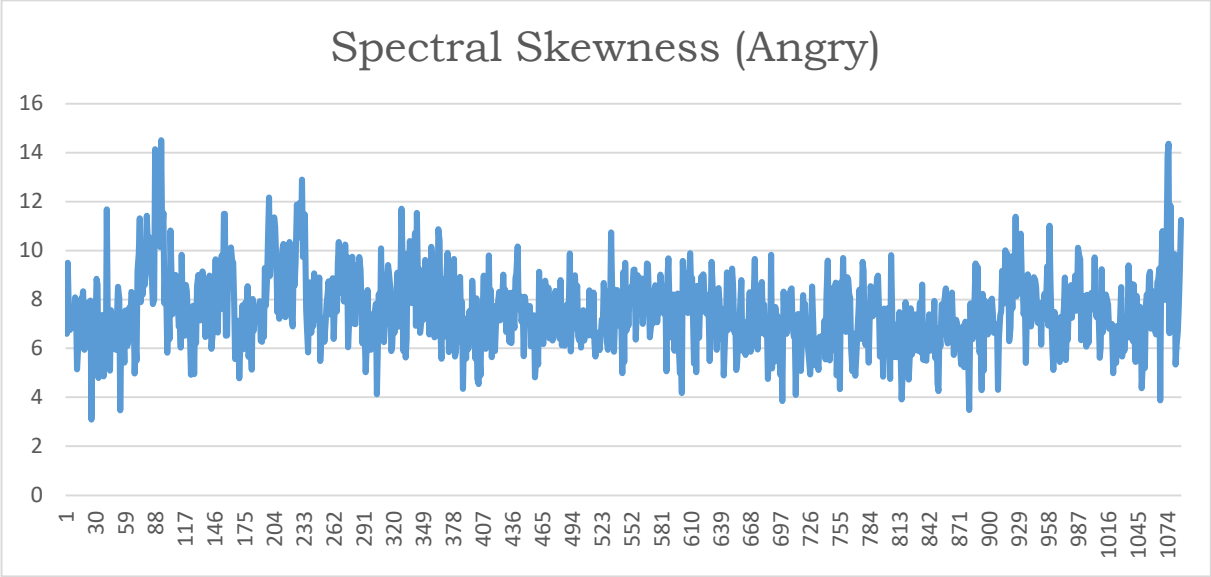

Fig 5.1.2.191: Plotting Spectral Skewness for 'Angry' Emotion Label

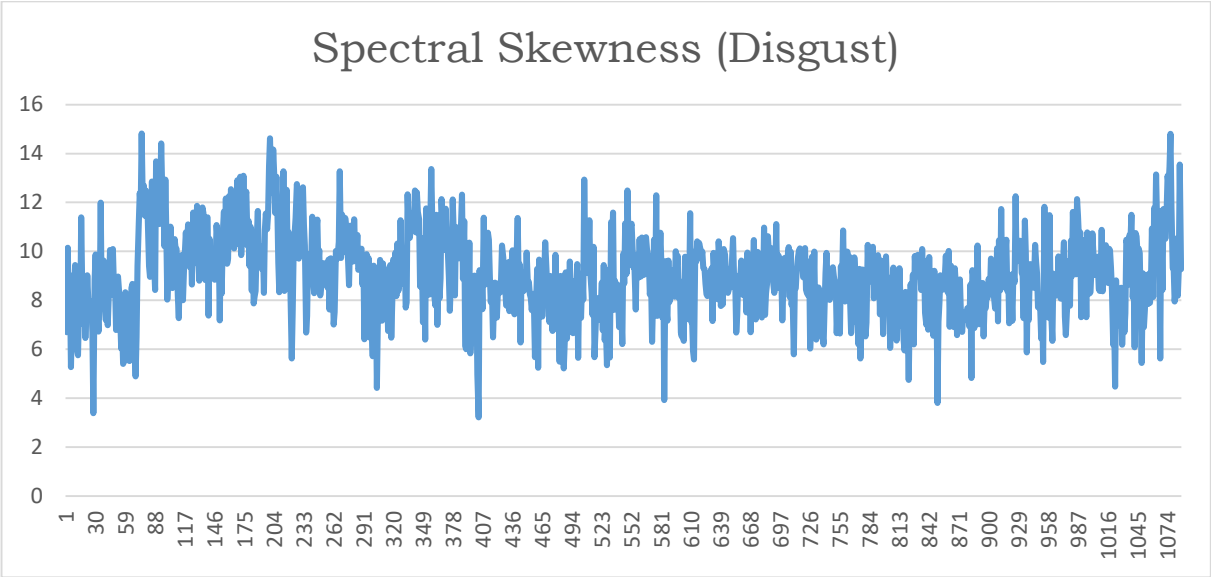

Fig 5.1.2.192: Plotting Spectral Skewness for 'Disgust' Emotion Label

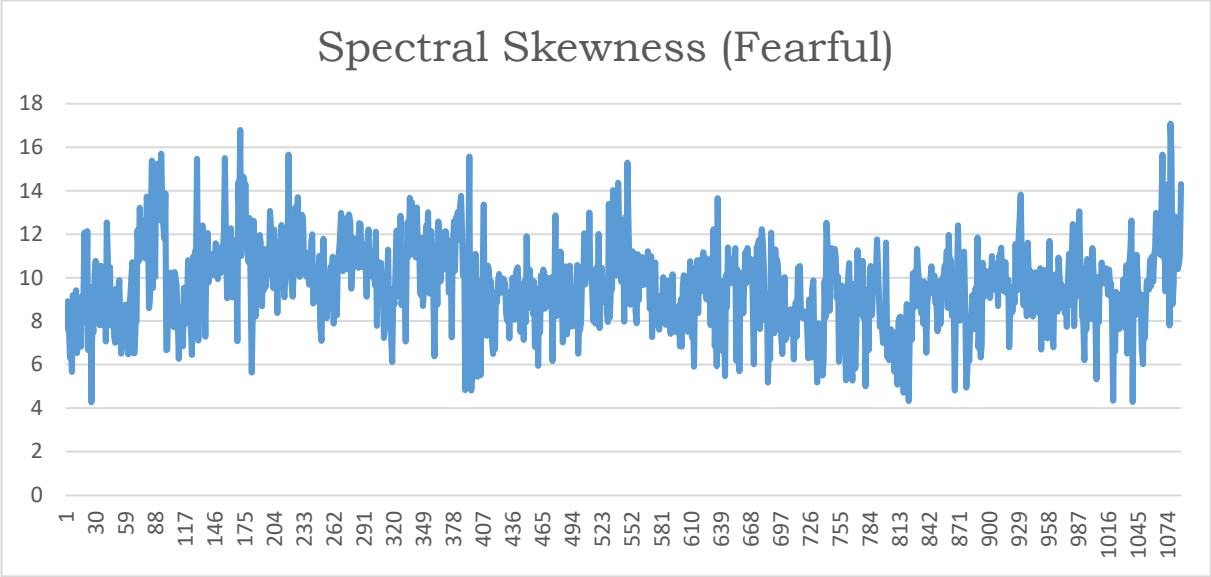

Fig 5.1.2.193: Plotting Spectral Skewness for 'Fearful' Emotion Label

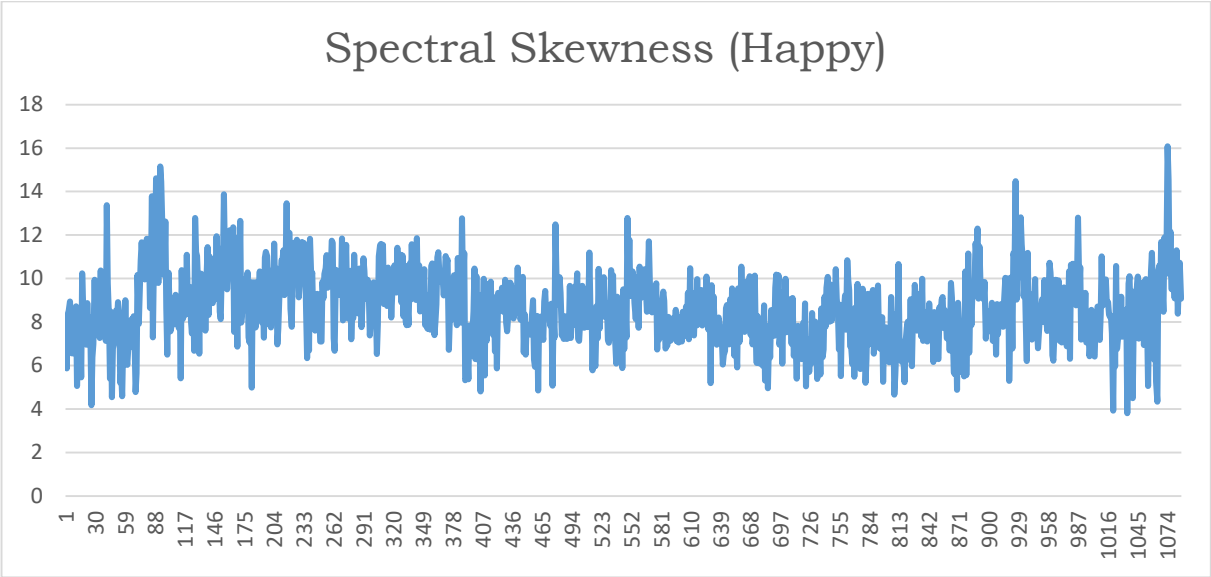

Fig 5.1.2.194: Plotting Spectral Skewness for 'Happy' Emotion Label

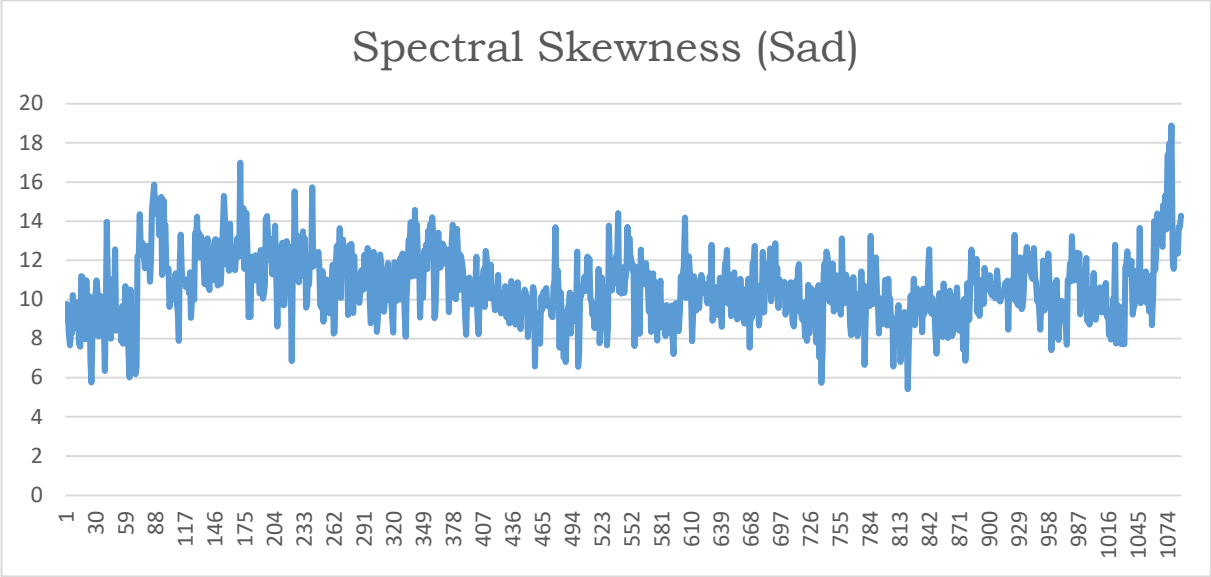

Fig 5.1.2.195: Plotting Spectral Skewness for 'Sad' Emotion Label

5.1.2 Experiment Results

Utilizing the SVM model, our classification performance across diverse datasets exhibits noteworthy accuracy metrics: 85.4% accuracy in the RAVDESS Dataset, a perfect 100% in TESS Dataset, a commendable 93.3% in SAVEE, a robust 95.7% in EmoDB, and a respectable 60.3% in CREMA-D. These results underscore the efficacy of our approach in discerning emotional states, with our model tailored to handle five distinct categories of emotions.

DATASET	ACCURACY (%)
RAVDESS	85.4
TESS	100
SAVEE	93.3
EmoDB	95.7
CREMA-D	60.3

Fig 5.1.3.1: Accuracy of the datasets using SVM Model

Table 5.1.3.1 presents a comprehensive overview of the remarkable accuracy attained by the Support Vector Machine (SVM) algorithm across a spectrum of datasets. This visual representation underscores the profound effectiveness of SVM in discerning and capturing the intricate emotional nuances intrinsic to vocal expressions.

In addition to the Support Vector Machines (SVMs) classifier, our classification efforts encompassed exploration with several other prominent algorithms, each offering unique insights into the intricate landscape of vocal emotion recognition.

Naive Bayes, recognized for its simplicity and efficiency, demonstrated varied accuracies across the datasets: achieving 50.5% accuracy in the RAVDESS Dataset, 83.5% in TESS Dataset, 53.3% in SAVEE, 76.1% in EmoDB, and 41.9% in CREMA-D. While it exhibited notable performance in certain datasets like TESS and EmoDB, its effectiveness was more modest in others, underscoring the importance of algorithm selection based on dataset characteristics.

RandomForest, leveraging the power of ensemble learning, showcased competitive accuracies across different datasets: achieving 68.2% accuracy in the RAVDESS Dataset, an impressive 99.75% in TESS Dataset, 78.3% in SAVEE, 78.3% in EmoDB, and 52.7% in CREMA-D. Its robust performance in the TESS Dataset underscores its ability to handle complex patterns within vocal expressions, while maintaining competitive accuracies across other datasets.

Similarly, Logistic Regression, a widely used linear classification algorithm, yielded noteworthy results: achieving 63.0% accuracy in the RAVDESS Dataset, 99.25% in TESS Dataset, 63.3% in SAVEE, 82.6% in EmoDB, and 53.4% in CREMA-D. Its performance particularly excelled in the TESS and EmoDB datasets, highlighting its effectiveness in scenarios with well-defined class boundaries.

Through our comprehensive exploration of these classification algorithms, we gained valuable insights into their respective strengths and weaknesses, ultimately informing our decision to proceed with the SVMs classifier due to its superior performance and adaptability across diverse datasets. This holistic approach underscores our commitment to employing the most effective methodologies to achieve precise and actionable insights into the intricate landscape of human emotion encapsulated within vocal expressions.

	RAVDESS Dataset	**TESS Dataset**	**SAVEE Dataset**	**EmoDB Dataset**	**CREMA-D Dataset**
Support Vector Machine (SVM)	85.4%	100%	93.3%	95.7%	60.3%
Naive Bayes	50.5%	83.5%	53.3%	76.1%	41.9%
Random Forest	68.2%	99.75%	78.3%	78.3%	52.7%
Logistic Regression	63.0%	99.25%	63.3%	82.6%	53.4%

Fig 5.1.3.2: Performance Evaluation: Accuracy Achieved Across Datasets

Table 5.1.3.2 offers a broad view of classification algorithm accuracy across various datasets, highlighting their efficacy in capturing nuanced vocal emotions. These accuracy metrics are crucial for evaluating algorithm performance, guiding the selection of the most suitable approach for each dataset. This analysis emphasizes the importance of choosing the right algorithm to enhance understanding of human emotions in vocal interactions.

In our analysis, we'll delve into classification results, showcasing classification reports, confusion matrices, and class-wise accuracy bar charts for each dataset. These visualizations and metrics will provide valuable insights into our classification approach's efficacy and reliability, aiding informed decision-making and model refinement for improved performance.

RAVDESS Dataset Comprehensive Analysis:

Fig 5.1.3.3 – Fig 5.1.3.5 illustrates the Classification Report, Confusion Matrix, Class-wise Accuracy Charts, respectively.

Classification Report:

	precision	recall	f1-score	support
Angry	0.92	0.90	0.91	39
Disgust	0.82	0.84	0.83	38
Fearful	0.86	0.84	0.85	38
Happy	0.88	0.77	0.82	39
Sad	0.80	0.92	0.85	38
accuracy			0.85	192
macro avg	0.86	0.85	0.85	192
weighted avg	0.86	0.85	0.85	192

Fig 5.1.3.3: Classification Report for RAVDESS Dataset

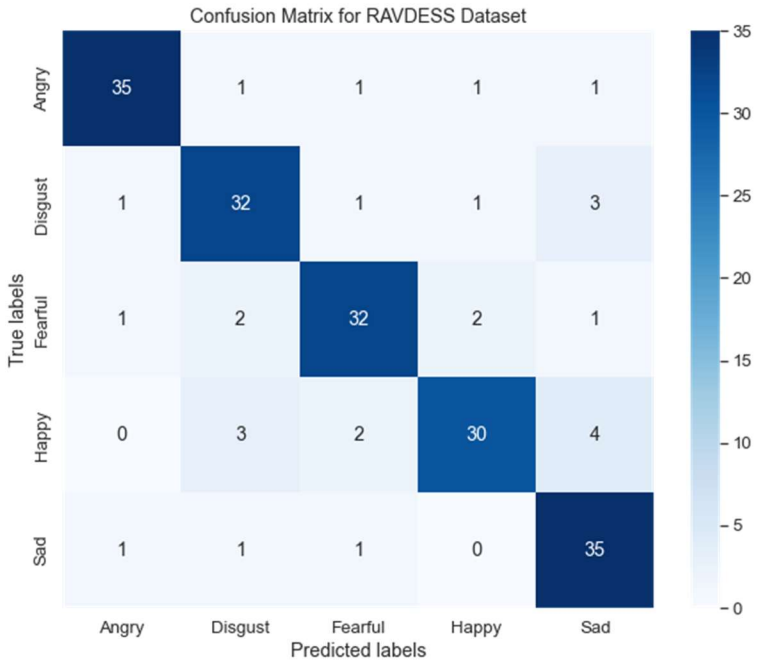

Fig 5.1.3.4: Confusion Matrix for RAVDESS Dataset

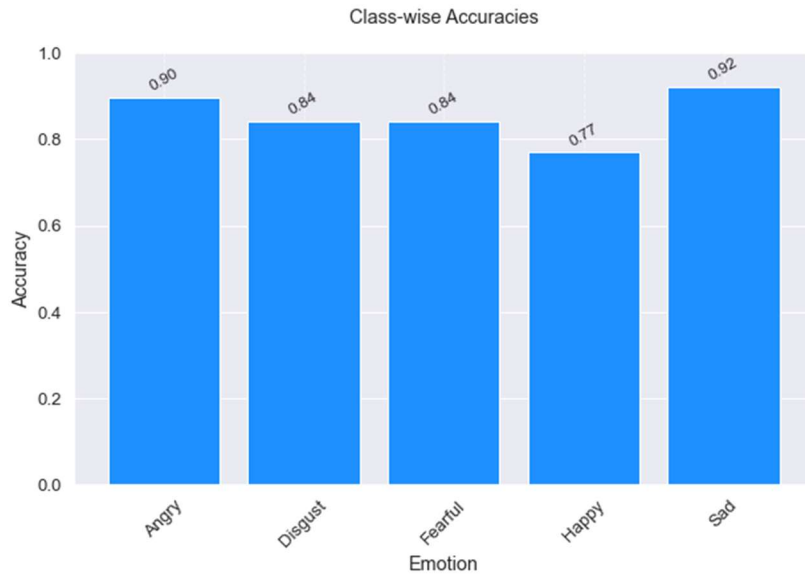

Fig 5.1.3.5: Class-wise accuracies for RAVDESS Dataset

TESS Dataset Comprehensive Analysis:

Fig 5.1.3.6 – Fig 5.1.3.8 illustrates the Classification Report, Confusion Matrix, Class-wise Accuracy Charts, respectively.

Classification Report:

	precision	recall	f1-score	support
Angry	1.00	1.00	1.00	80
Disgust	1.00	1.00	1.00	80
Fearful	1.00	1.00	1.00	80
Happy	1.00	1.00	1.00	80
Sad	1.00	1.00	1.00	80
accuracy			1.00	400
macro avg	1.00	1.00	1.00	400
weighted avg	1.00	1.00	1.00	400

Fig 5.1.3.6: Classification Report for TESS Dataset

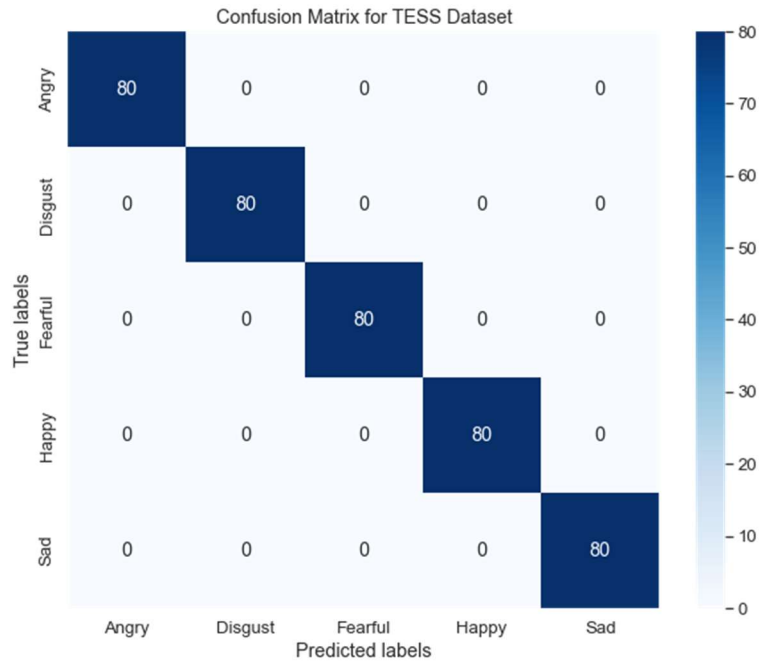

Fig 5.1.3.7: Confusion Matrix for TESS Dataset

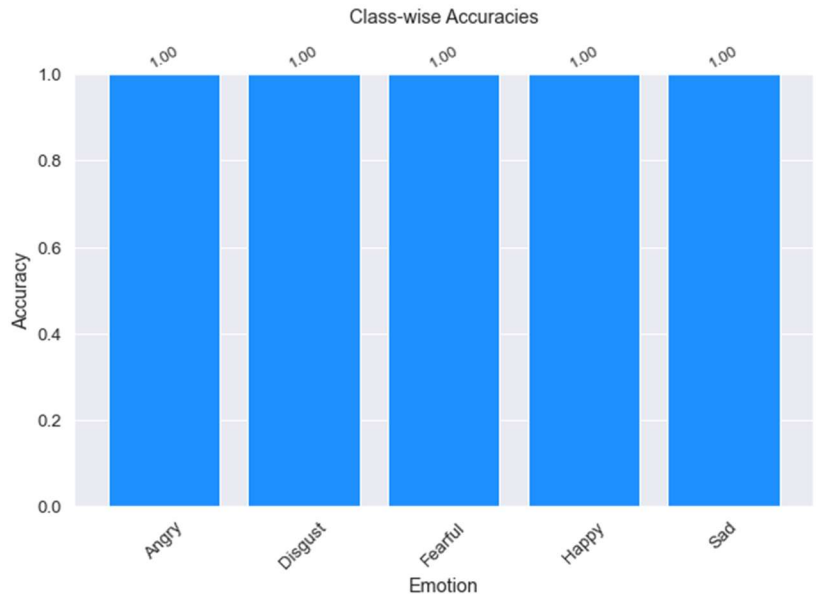

Fig 5.1.3.8: Class-wise accuracies for TESS Dataset

SAVEE Dataset Comprehensive Analysis:

Fig 5.1.3.9 – Fig 5.1.3.11 illustrates the Classification Report, Confusion Matrix, Class-wise Accuracy Charts, respectively.

Classification Report:

	precision	recall	f1-score	support
Angry	0.85	0.92	0.88	12
Disgust	1.00	0.92	0.96	12
Fearful	1.00	1.00	1.00	12
Happy	0.91	0.83	0.87	12
Sad	0.92	1.00	0.96	12
accuracy			0.93	60
macro avg	0.94	0.93	0.93	60
weighted avg	0.94	0.93	0.93	60

Fig 5.1.3.9: Classification Report for SAVEE Dataset

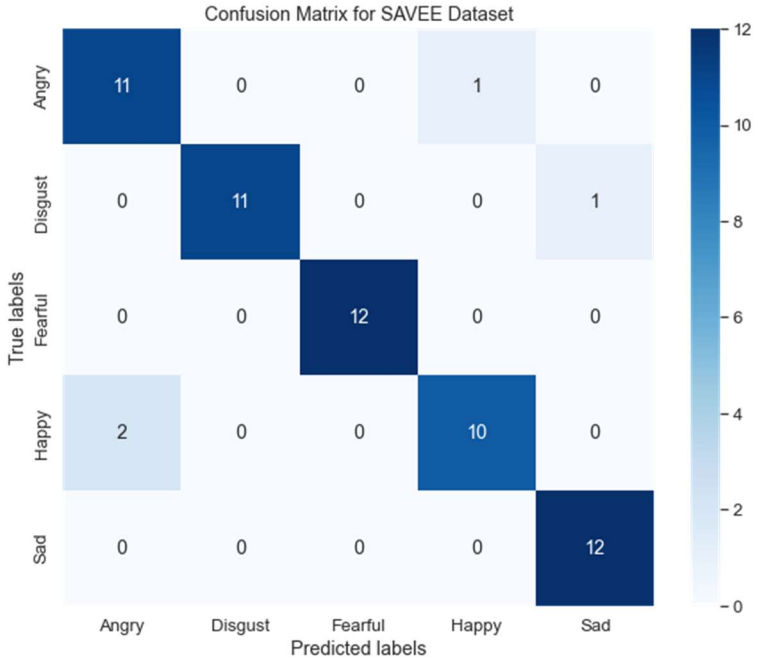

Fig 5.1.3.10: Confusion Matrix for SAVEE Dataset

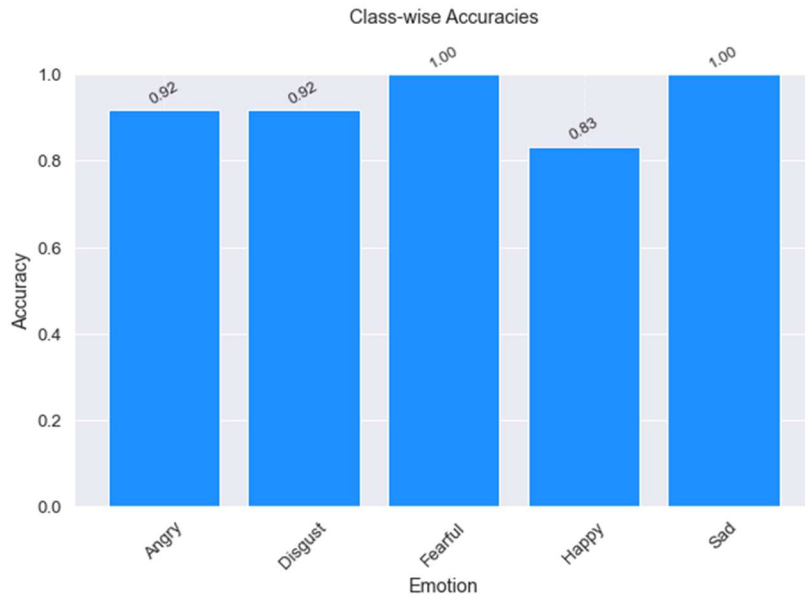

Fig 5.1.3.11: Class-wise accuracies for SAVEE Dataset

EmoDB Dataset Comprehensive Analysis:

Fig 5.1.3.12 – Fig 5.1.3.14 illustrates the Classification Report, Confusion Matrix, Class-wise Accuracy Charts, respectively.

Classification Report:

	precision	recall	f1-score	support
Angry	1.00	0.90	0.95	10
Disgust	1.00	0.89	0.94	9
Fearful	0.90	1.00	0.95	9
Happy	0.90	1.00	0.95	9
Sad	1.00	1.00	1.00	9
accuracy			0.96	46
macro avg	0.96	0.96	0.96	46
weighted avg	0.96	0.96	0.96	46

Fig 5.1.3.12: Classification Report for EmoDB Dataset

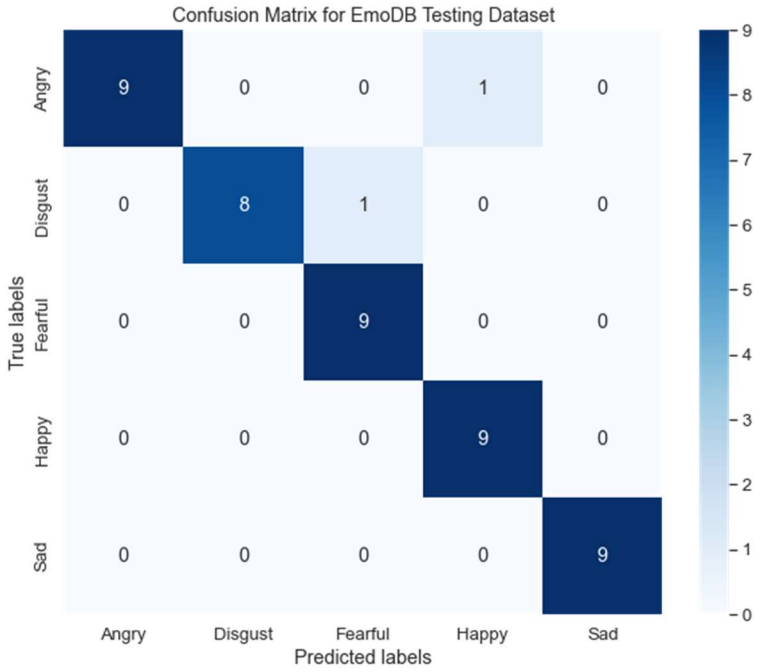

Fig 5.1.3.13: Confusion Matrix for EmoDB Dataset

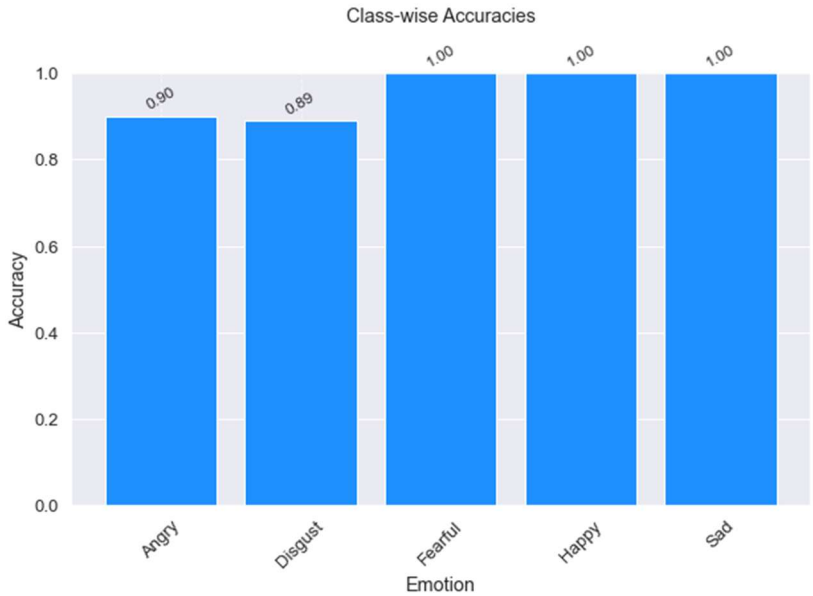

Fig 5.1.3.14: Class-wise accuracies for EmoDB Dataset

CREMA-D Dataset Comprehensive Analysis:

Fig 5.1.3.15 – Fig 5.1.3.17 illustrates the Classification Report, Confusion Matrix, Class-wise Accuracy Charts, respectively.

Classification Report:

	precision	recall	f1-score	support
Angry	0.70	0.67	0.69	217
Disgust	0.58	0.60	0.59	217
Fearful	0.57	0.39	0.46	218
Happy	0.52	0.53	0.53	218
Sad	0.62	0.82	0.71	217
accuracy			0.60	1087
macro avg	0.60	0.60	0.60	1087
weighted avg	0.60	0.60	0.60	1087

Fig 5.1.3.15: Classification Report for CREMA-D Dataset

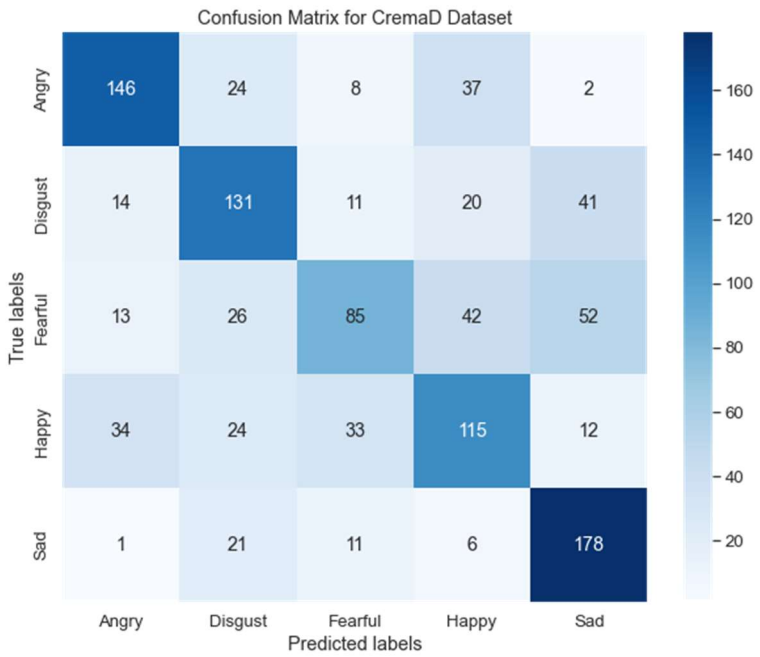

Fig 5.1.3.16: Confusion Matrix for CREMA-D Dataset

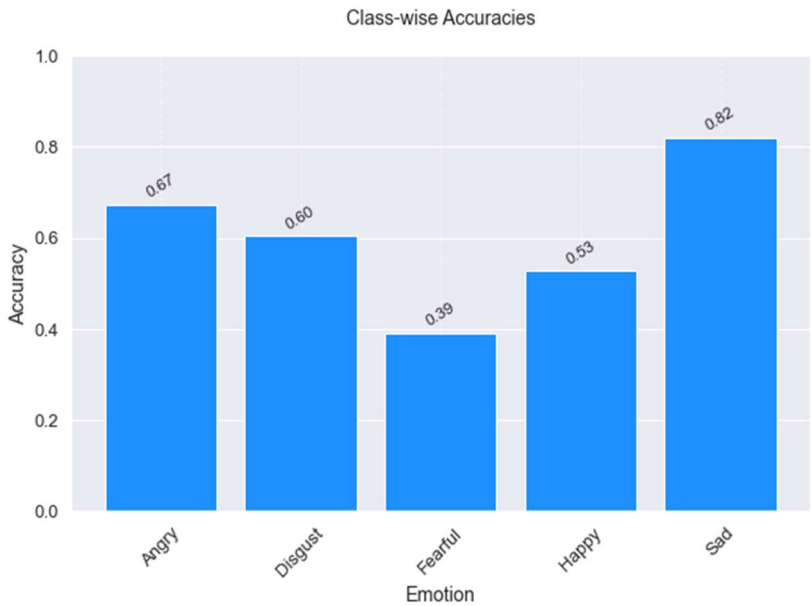

Fig 5.1.3.17: Class-wise accuracies for CREMA-D Dataset

5.2 Method – 2

5.2.1 Testing Data:

We validated our speech emotion recognition system using a diverse collection of .wav audio files sourced from the RAVDESS dataset, ensuring a broad spectrum of emotional expressions such as neutral, happy, sad, angry, fearful, calm, surprise and disgust. Audio samples represented a balanced mix of male and female voices to capture the variance in emotional vocal expressions.

Before feeding the audio data into our system, we conducted comprehensive preprocessing steps, including feature extraction (MFCCs, Mel-spectrogram, Chromagram) and feature scaling using StandardScaler, to ensure consistent input format and quality. This preprocessing phase was crucial for preparing the data for effective model training and evaluation.

To assess the system's performance, we split the dataset into training (80%) and testing (20%) sets. This division facilitated a fair representation of emotions in both training and testing phases, ensuring that the models were trained on a diverse range of emotional samples.

During model training the stack ensemble method leveraged the predictions from the base models to train the meta-regressor (MLP), optimizing the system's ability to classify emotions accurately. We evaluated the system's performance primarily based on testing accuracy, comparing predicted emotions with ground truth labels from the test dataset to calculate accuracy rates.

Rigorous testing and analysis of results enabled us to validate the accuracy and reliability of our system for speech emotion recognition. The stack ensemble approach, utilizing a combination of base models and a meta-regressor, showcased the effectiveness of our implementation in accurately identifying and classifying emotional states in speech data, with implications for applications in smart healthcare and customer service industries.

5.2.2 Testing Results:

We have compiled a dataset comprising 1440 audio recordings sourced from the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). This dataset features performances by 24 professional actors, evenly split between genders, each delivering lexically-matched statements in a neutral North American accent. The actors portray a range of emotions including calm, happy, sad, angry, fearful, surprise, and disgust, expressed at two levels of emotional intensity: normal and strong. Additionally, the dataset includes recordings of neutral expressions for reference.

The audio files within the RAVDESS dataset are structured to include identifiers denoting modality, vocal channel, emotion, intensity, statement, repetition, and actor gender. The audio samples in the dataset are in mono format. Designed specifically for research in emotion recognition, RAVDESS offers a diverse array of emotional speech stimuli for thorough analysis and perceptual validation. Each audio recording provides valuable insights into the nuances of human emotion expression, enabling researchers to delve deeper into the intricacies of emotion recognition algorithms and models.

To leverage this dataset for classification purposes, we utilized Python to process and analyze the audio data. Initially, we extracted pertinent features from the audio files and compiled them into a structured CSV format. These features encompass various aspects of the audio signals, such as spectral characteristics, temporal dynamics, and pitch attributes. Subsequently, we employed Python code to ingest the CSV file and train classification models based on the extracted features.

Through the collaborative efforts of utilizing the RAVDESS dataset, Python programming, and Google Co-lab resources, we gained profound insights into the audio features inherent within the dataset. By exploring the distinctive characteristics and nuances of the audio samples, we developed robust classification models for effectively recognizing emotions conveyed through speech. This research endeavour not only enhances our understanding of audio processing but also contributes significantly to the field of emotion analysis and recognition.

5.2.3 Experiment Results:

In our experiment, a stack ensemble approach combining logistic regression, random forest, and MLP achieved 67% accuracy in classifying audio features into 8 emotion categories, surpassing individual algorithms (random forest: 53%, logistic regression: 56%, MLP: 59%). This highlights the effectiveness of stack ensemble learning for complex emotion classification tasks.

We'll analyze classification results for the RAVDESS dataset, focusing on accuracy and confusion matrices. These insights will guide improvements for better performance and reliability.

Logistic Regression

Fig 5.2.3.1 and Fig 5.2.3.2 display the Accuracy and Confusion Matrix achieved using Logistic Regression.

```
Test accuracy (Logistic Regression): 0.5590277777777778
Training accuracy (Logistic Regression): 1.0
```

Fig 5.2.3.1: Accuracy of Logistic Regression

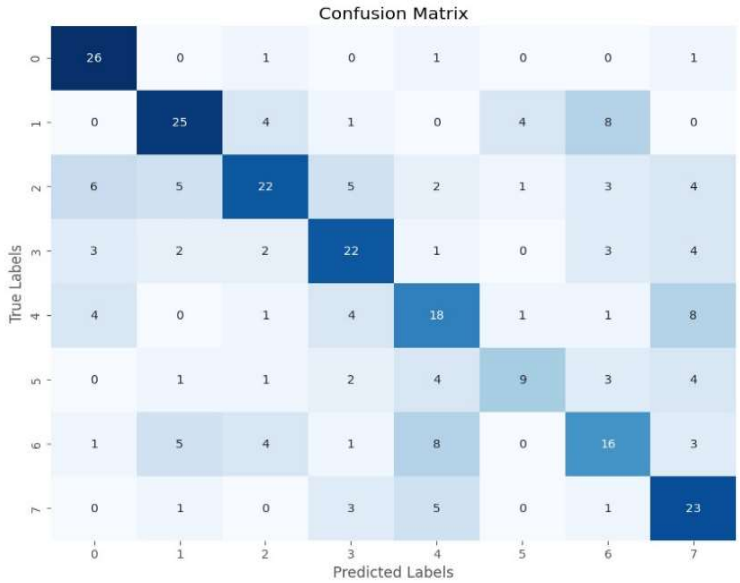

Fig 5.2.3.2: Confusion Matrix of Logistic Regression

Random Forests

Fig 5.2.3.3 and Fig 5.2.3.4 display the Accuracy and Confusion Matrix achieved using Random Forest.

```
Test accuracy (Random Forest): 0.5277777777777778
Training accuracy (Random Forest): 1.0
```

Fig 5.2.3.3: Accuracy of Random Forest

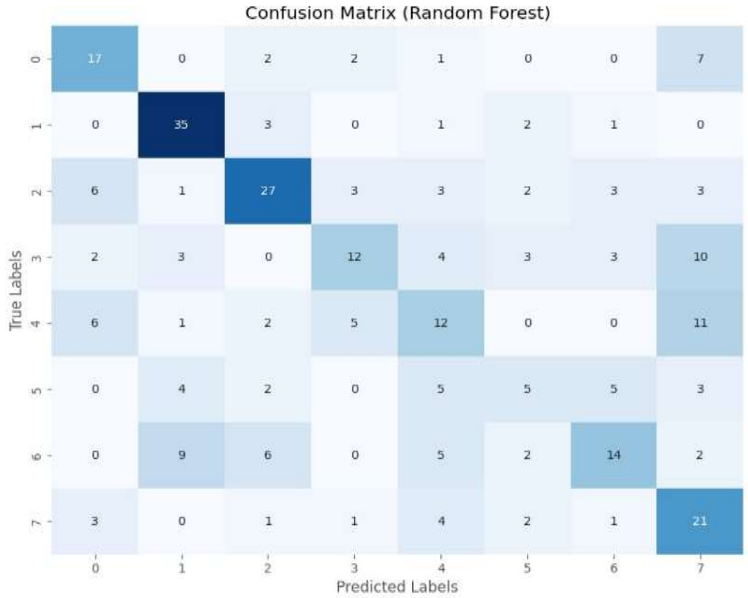

Fig 5.2.3.4: Confusion Matrix of Random Forest

Multilayer Perceptron

Fig 5.2.3.5 and Fig 5.2.3.6 display the Accuracy and Confusion Matrix achieved using Multilayer Perceptron.

```
Test accuracy (MLP): 0.5972222222222222
Training accuracy (MLP): 1.0
```

Fig 5.2.3.5: Accuracy of Multilayer Perceptron

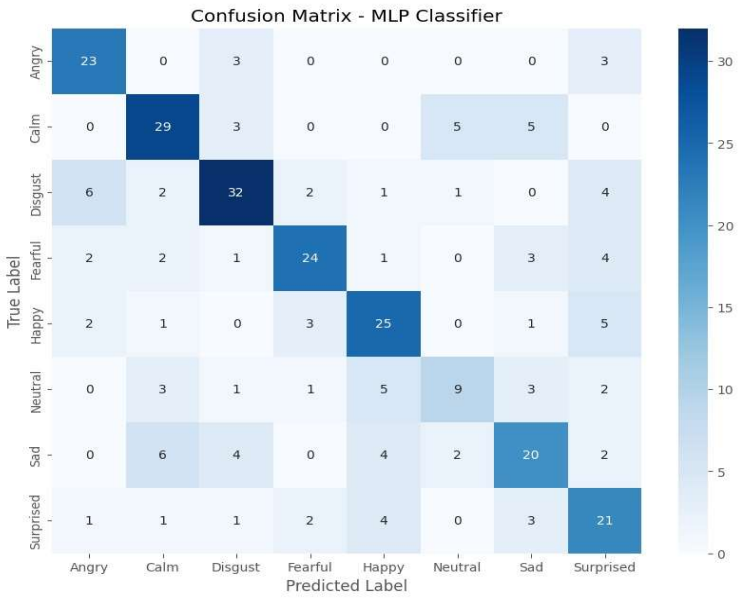

Fig 5.2.3.6: Confusion Matrix of Multilayer Perceptron

Stack Ensemble Method

Fig 5.2.3.3 and Fig 5.2.3.4 display the Accuracy and Confusion Matrix achieved using Random Forest Stack Ensemble Method.

```
Test accuracy (Stacking): 0.6666666666666666
Training accuracy (Stacking): 1.0
```

Fig 5.2.3.7: Accuracy of Stack Ensemble Method

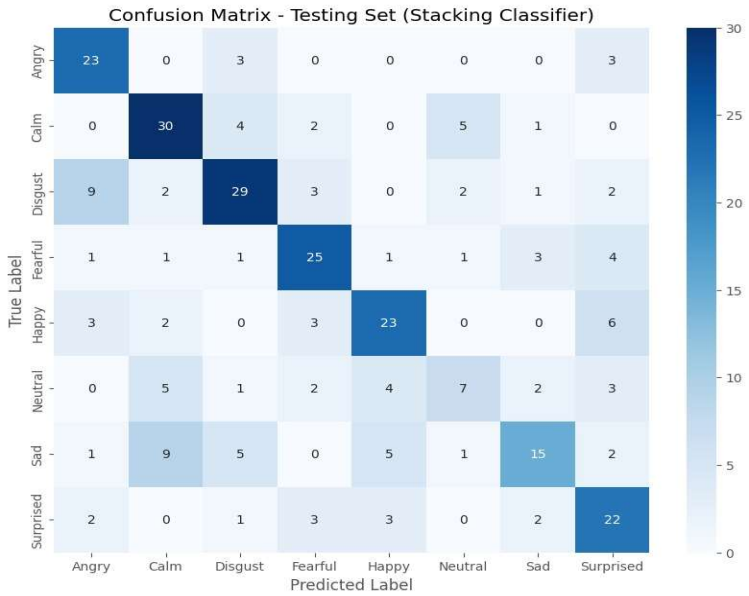

Fig 5.2.3.8: Confusion Matrix of Stack Ensemble Method

Comparison of the Different Classifier:

Name of the Classifier	Classification Accuracy
Logistic Regression	**55.90%**
Random Forest	**52.77%**
Multilayer Perception	**59.72%**
Stack Ensemble (Base Model – Random Forest Logistic Regressor, Meta Regressor- MLP)	**66.66%**

Fig 5.2.3.9: Compare between different classifier

The table in Fig 5.2.3.9 compares classification accuracies of various machine learning classifiers: Logistic Regression, Random Forest, Multilayer Perception, and a Stack Ensemble model. The Stack Ensemble, utilizing Random Forest and Logistic Regression as base models and Multilayer Perception as the meta-regressor, achieves the highest accuracy at 66.60%, followed by Multilayer Perception at 59.72%, Logistic Regression at 55.90%, and Random Forest at 52.77%. This comparison aids data scientists in selecting suitable models for predictive tasks.

5.2 Comparative Study

In our pursuit of advancing speech emotion recognition, we drew inspiration from the insightful study conducted by Jerry Joy et al. titled "Speech Emotion Recognition using Neural Network and MLP Classifier," which was published in 2020 at SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India. Their pioneering research, leveraging the MLP Classifier, yielded a commendable accuracy of 70.28% while effectively classifying eight distinct emotions, including Happy, Fear, Sad, Disgust, Angry, Surprise, Neutral, and Calm.

In aligning our investigation with theirs, we focused on employing a comparable feature set, specifically utilizing the 13 coefficients of MFCCs, the 7 coefficients of Contrast, the 32 coefficients of Mel Spectrogram, the 12 coefficients of Chromagram, and the 6 coefficients of Tonnetz extracted from the RAVDESS dataset. However, our initial attempts using traditional classifiers such as Support Vector Machine, Random Forest, Logistic Regression, and a Voting Classifier yielded accuracies that fell short of our expectations.

These results underscore the complexity of the task at hand and the nuances involved in accurately discerning emotions from speech. Nonetheless, this serves as a testament to the critical role feature selection plays in model performance. Moving forward, we remain optimistic that by refining our feature set, we will unlock the full potential of our model, paving the way for substantial improvements in speech emotion recognition accuracy.

Name of Classifier	Accuracy Achieved
Support Vector Machine	**56.11%**
Random Forest	**55%**
Logistic Regression	**49.44%**
Voting Classifier	**60.83%**

Table 5.2: Accuracy achieved for the compared dataset

Table 5.2 shows the accuracy of four classifiers on a dataset. The Support Vector Machine achieved 56.11%, Random Forest 55%, and Logistic Regression 49.44%. The Voting Classifier, which combines multiple classifiers, had the highest accuracy at 60.83%, indicating the effectiveness of ensemble methods.

Chapter 6: Conclusion and Future Work

6.1 Scope for Future Improvement

To enhance the system's efficacy and inclusivity, future improvements involve diversifying the dataset to encompass a broader range of accents beyond American and European, thereby ensuring accurate emotion detection across diverse linguistic backgrounds. Additionally, addressing language variability by incorporating recordings in languages beyond English and Spanish is essential to mitigate limitations associated with mixed-language data. Furthermore, implementing user-specific training methods will personalize the model to individual speech patterns, accommodating diverse linguistic preferences and enhancing overall performance. By training the model based on user's audio input, it can adapt to different speaking styles, ultimately improving its effectiveness and usability for a wider user base.

6.2 Conclusion

In the realm of sentiment analysis through speech, the "Emo-Track" project stands as a pioneering endeavor, showcasing the transformative potential of advanced machine learning methodologies. By meticulously extracting an extensive array of features from audio data, ranging from spectral flux and skewness to zero crossing rate and mel spectrogram, the project has laid the foundation for a nuanced understanding of emotional cues embedded within speech patterns. This comprehensive feature set serves as a testament to the project's commitment to capturing the intricacies of human expression, enabling the development of sophisticated sentiment analysis models capable of discerning subtle shifts in emotional states.

Central to the success of "Emo-Track" is its emphasis on leveraging state-of-the-art machine learning techniques to interpret and classify the extracted features effectively. Through the utilization of methods such as MFCC (Mel-frequency cepstral coefficients) and linear predictive coding (LPC), the project has empowered its sentiment analysis models with the ability to discern the underlying emotional content of speech data with remarkable accuracy. This capability holds immense promise for a myriad of applications, from enhancing the efficiency of customer service interactions by gauging customer sentiment in real-time to facilitating mental health monitoring through the analysis of speech patterns indicative of emotional distress.

Moreover, the "Emo-Track" project serves as a testament to the crucial role of feature extraction methodologies in unlocking the latent emotional insights embedded within speech data. By meticulously

selecting and engineering features such as spectral entropy, harmonic-to-noise ratio (HNR), and chromagram, the project has demonstrated its commitment to capturing a diverse range of acoustic attributes relevant to emotional expression. This holistic approach not only enriches the depth of analysis but also underscores the project's dedication to achieving a nuanced understanding of emotional states beyond surface-level sentiment.

Looking ahead, the "Emo-Track" project offers boundless opportunities for further refinement and expansion. Future iterations could explore the integration of additional feature extraction techniques and machine learning algorithms to enhance the robustness and scalability of sentiment analysis models. Furthermore, the project could benefit from the incorporation of real-time user feedback mechanisms to continuously adapt and refine its classification capabilities based on evolving emotional dynamics. By embracing these avenues for growth and exploration, "Emo-Track" is poised to continue its trajectory as a trailblazer in the realm of sentiment analysis through speech, with far-reaching implications for enhancing emotional well-being and enriching user experiences across diverse domains.

Reference

[1] J. Guru Monish Amartya, S. Magesh Kumar, "Speech Emotion Recognition in Machine Learning to Improve Accuracy using Novel Support Vector Machine and Compared with Random Forest Algorithm", 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), pp.862-866, 2022.

[2] Rumagit, R. Y., Alexander, G., & Saputra, I. F. (2020). Model Comparison in Speech Emotion Recognition for Indonesian Language. In 5th International Conference on Computer Science and Computational Intelligence.Jakarta, Indonesia: Bina Nusantara University.2020.

[3] N. P. Poojary, S. G. S. Kumar, and A. K. B. H. (2021). Speech Emotion Recognition Using MLP Classifier. International Journal of Scientific Research in Science and Technology, pp. 218-222, July. 2021.

[4] S. G. Shaila, A. Sindhu, L. Monish, D. Shivamma, and B. Vaishali, "Speech Emotion Recognition Using Machine Learning Approach," in ICAMIDA 2022, ACSR, vol. 105, pp. 592–599, 2023.

[5] U. K. Singh, S. Singh, S. Khanna, and R. S., "Speech emotion recognition using machine learning and deep learning". International Journal of Engineering Applied Sciences and Technology, 6(11), Vol. 6, pp. 181-184,2022.

[6] A. Koduru, H. B. Valiveti, and A. K. Budati, "Feature extraction algorithms to improve the speech emotion recognition rate," Int. J. Speech Technol., vol. 23, pp. 45–55, Jan. 2020.

[7] R. Novais, P. J. S. Cardoso, and J. M. F. Rodrigues, "Emotion classification from speech by an ensemble strategy," in Proc. 10th Int. Conf. Softw. Dev. Technol. Enhanc. Accessibility Fighting, pp. 85–90, 2023.

[8] T. Sai Samhith, G. Nishika, M. Prayuktha, M. Bharat Chandra, Dr. Sunil Bhutada, and G. Prasadu, "Speech emotion recognition using machine learning algorithms," Int. J. Creative Res. Thoughts, vol. 9, no. 6, pp. 2320-2882, 2021.

[9] A. Gupta, S. Morye, M. Sitap, and S. Chaudhary, "Speech-based emotion recognition using machine learning," Int. Res. J. Eng. Technol., vol. 8, no. 4, 2021.

[10] A. Singh, S. Chodankar, and A. Suvarna, "Speech sentiment analyzer," Int. Res. J. Modern. Eng. Technol. Sci., vol. 3, no. 5, May 2021.

[11] Joy, J., Kannan, A., Ram, S., & Rama, S. (2020). Speech emotion recognition using neural network and MLP classifier. Ijesc, 2020, 25170-25172.

[12] Steven R. Livingstone, and Frank A. Russo. (2019). RAVDESS Emotional speech audio [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/256618

[13] Kate Dupuis, M. Kathleen Pichora-Fuller. University of Toronto, Psychology Department, (2010). Toronto emotional speech set (TESS) [Data set]. Kaggle. https://doi.org/10.5683/SP2/E8H2MF

[14] Philip J. B. Jackson, S. U. Haq. (2011). Surrey Audio-Visual Expressed Emotion (SAVEE) [Data set]. Kaggle.

[15] Rudd et al. (2023). Berlin Database of Emotional Speech (EmoDB) [Data set]. Kaggle.

[16] Cao H, Cooper DG, Keutmann MK, Gur RC, Nenkova A, Verma R. CREMA-D: Crowd-sourced Emotional Multimodal Actors Dataset. IEEE Trans Affect Comput. 2014 Oct-Dec;5(4):377-390. doi: 10.1109/TAFFC.2014.2336244. PMID: 25653738; PMCID: PMC4313618.

[17] Molau, S., Pitz, M., Schluter, R., & Ney, H. (2001, May). Computing mel-frequency cepstral coefficients on the power spectrum. In 2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (cat. No. 01CH37221) (Vol. 1, pp. 73-76). IEEE.

[18] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., ... & Wu, Y. (2018, April). Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4779-4783). IEEE.

[19] Birajdar, G. K., & Patil, M. D. (2020). Speech/music classification using visual and spectral chromagram features. Journal of Ambient Intelligence and Humanized Computing, 11(1), 329-347.

[20] Bradbury, J. (2000). Linear predictive coding. Mc G. Hill.

[21] Stolar, M. N., Lech, M., Stolar, S. J., & Allen, N. B. (2018). Detection of adolescent depression from speech using optimised spectral roll-off parameters. Biomedical Journal, 2(10).