AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय

Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

A Project Phase-II Report

On

Hybrid Approach to Emotion Detection using Deep Learning

Submitted in the partial fulfillment of the requirements of

Bachelor of Engineering

in

Electronics and Telecommunication

By

Atharva Ardhapurkar	**21ET003**
Mohammad Sahil Shaikh	**21ET057**
Vaishnavi Patil	**21ET047**

Under the guidance of

Prof. S. B. Dhekale

Department of Electronics and Telecommunication Engineering
All India Shri Shivaji Memorial Society's College of Engineering, Pune-01
Affiliated to SPPU Pune

A.Y 2024-25

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

CERTIFICATE

This is to certify that the Project titled **"Hybrid Approach to Emotion Detection using Deep Learning"** Submitted by,

Mr. Atharva Ardhapurkar	**B400210299**
Mr. Mohammad Sahil Shaikh	**B400210351**
Ms. Vaishnavi Patil	**B400210344**

is a bonafide record of the Project Phase-II work carried out by them towards the partial fulfillment of the requirements of Savitribai Phule Pune University, for the award of Bachelor of Electronics and Telecommunication Engineering under my supervision and guidance.

Prof. S. B. Dhekale
Project Guide

Dr. R R Itkarkar
Project coordinator

Dr. S B Dhonde
HOD- E & TC

Date:

Place**: Pune**

This report has been examined as per Savitribai Phule Pune University requirements at **AISSMS COE, Pune-01**

Signature:

Name of Internal Examiner

Signature:

Name of External Examiner

ACKNOWLEDGEMENT

We take this opportunity to express our deep sense of gratitude towards our guide Prof. S B Dhekale under whose guidance we had the privilege to work on this project. We also express our gratitude towards our project co-ordinator Prof. R R Itkarkar for her kind support and co-operation.

We are grateful to Dr S B Dhonde Head of E & TC Engineering Department and all the teaching and non-teaching staff members of E& TC Engineering Department for their direct or indirect help in completion of the phase-I of project.

Mr. Atharva Ardhapurkar B400210299

Mr. Mohammad Sahil Shaikh B400210351

Ms. Vaishnavi Patil B400210344

PROLOGUE

All India Shri Shivaji Memorial Society, established as early as in 1917, is a premier educational institution in Pune. Ours is a result-oriented Society dedicated to the noble cause of Military, General, Technical & Management Education in India by Shri Chhatrapati Shahu Maharaj of Kolhapur also known as "Rajarshi Shahu". He was the first Maharaja of the princely state of Kolhapur and a great social reformer.

He was an invaluable gem in the history of Maharashtra he worked tirelessly for the cause of the lower caste subjects in his state. Primary education to all regardless of caste and creed was one of his most significant priorities. In our Institute we follow the noble practices established by our great leader and as we know Leader decide the destiny we are blessed to have a leadership and a rich cultural heritage which is totally focused on the noble cause of Quality Education. It functions beyond race, caste, creed, religion & political spirit. The Society Management is very pragmatic & progressive.

On Nov 10, 1917, Rajarshi Chhatrapati Shahu Maharaj announced in Delhi, his proposal to establish a Memorial of Chhatrapati Shivaji Maharaj in the City of Pune. In the proposal, he declared his intention of creating a central hall with Chhatrapati Shivaji Maharaj's life size statue and a hostel to accommodate a hundred Maratha students. The institute was to act as a rallying centre for Maratha activities all over India, giving a concrete shape and impetus to the diverse endeavours for the advancement of the Maratha community.

On Dec 27, 1917, Rajarshi Chhatrapati Shahu Maharaj, on a pressing demand from Khaserao Jadhav,

presided over the 11th session of the Maratha Educational Conference at Khamgoan. Rajarshi Chhatrapati Shahu Maharaj had convened a meeting of the committee of the All India Shivaji Memorial Society on 25 May 1922. Chhatrapati Rajaram Maharaj was elected President of the All India Shivaji Memorial Society and took on.

Core Values :

- LEADERSHIP AND CULTURAL HERITAGE
- HONESTY AND INTEGRITY
- FREEDOM OF THOUGHT AND EXPRESSION
- EXCELLENCE
- ACCOUNTABILITY AND TRANSPARENCY
- ENCOURAGEMENT
- SOCIAL RESPONSIBILITY

Honesty & Integrity :

The foreword of our society is "Satyala Maran Nahi" which means "Truth is eternal"

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

COLLEGE VISION AND MISSION

Vision :

Service to society through quality education.

Mission :

- Generation of national wealth through education and research
- Imparting quality technical education at the cost affordable to all strata of the society
- Enhancing the quality of life through sustainable development
- Carrying out high quality intellectual work
- Achieving the distinction of highest preferred engineering college in the eyes of the stake holders

Goal :

- To inculcate learning habits
- To create an environment to make the students creative and innovative
- To promote project based learning
- To strengthen industry – institute interaction
- To ensure continuous improvement in quality
- To develop entrepreneurship skills
- To nurture the spirit of team work
- To catalyse all – round development of students
- To develop technologies for sustainable development

DEPARTMENT VISION AND MISSION

Vision :

Society Growth and Welfare Through Competent Electronics and Communication Engineering Graduates.

Mission :

- To impart quality education in the field of E & TC engineering to solve societal and industrial problems with focus on trans-disciplinary approach
- To provide stimulating learning environment with modern tools & technologies.
- To produce dynamic graduates with ethics and moral values.
- To facilitate E & TC graduates with sight of innovation

Course Outcomes :

Course outcomes -Project Stage-1

CO1: Identify project for society and industry need by applying engineering knowledge gained throughout the E and TC Engineering program

CO2: Investigate identified complex engineering problem using appropriate research methods and techniques

CO3: Test the solution of identified engineering problem with appropriate simulation tool.

CO4: Work in team and effective budget Planning to meet the project requirement

CO5 : Effectively communicate the project progress through presentations and technical report.

CO6: Develop self-learning skills and follow the ethical code of conduct for project.

Course outcomes -Project Satge-2

CO1: Develop solutions to the real world problems using modern engineering tools and technologies.

CO2: Demonstrate practical skills and knowledge in testing and debugging for both hardware and software based projects.

CO3: Work in team to demonstrate the project by using visual aids and visualization techniques.

CO4: Effectively communicate project work through publications, competitions, presentations and technical report.

CO5: Showcase the project management and self-learning skills for lifelong learning.

CO6: Adherence to ethical code of conduct for project execution.

Academic Year : 2024- 25

Name of the Programme: Electronics & Telecommunication Engineering.

Program Outcomes :

1. A graduate student will apply knowledge of mathematics and engineering fundamentals to design electronic circuits and systems such as logical circuits, analog circuits, electrical machines, control and communication systems, digital systems etc. and test the results.
2. A graduate student will be able to identify, formulate, solve electronics engineering problems.
3. A graduate student will demonstrate the ability to design, implement and evaluate a system, process, component and program to meet the specified needs with appropriate considerations for public, health and safety, society and environment.
4. A graduate student will investigate, formulate, analyze and provide appropriate solution to simple and complex engineering problem.
5. A graduate student will provide the solutions by using the Modern electronic and IT Engineering Tools and Technologies for practicing electronic Engineering problems
6. A graduate student will demonstrate the ability to learn the impact of industries on society by visiting different industries and understand the importance of industrial products for analog and digital circuits and systems.
7. A graduate student will understand the importance of environmental issues and will design sustainable systems.
8. A graduate student will understand the professional and ethical responsibilities to meet the socio-economic challenges.
9. A graduate student will function effectively as an individual, member or leader of a team in multidisciplinary setting.
10. A graduate student will be able to communicate effectively at different technical and administrative levels.
11. A graduate student will demonstrate knowledge and understanding of engineering and management principles to manage projects.
12. A graduate student will 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 Outcome (PSO's) of Department

1. Systems Development: Design and develop prototype or system using hardware and software skills for multi-disciplinary applications.
2. Adaptability: Demonstrate an ability to adapt emerging technologies in the field of Artificial Intelligence machine learning, Internet of things and Robotics to solve the real life problems.
3. Skill Development: Develop technical and management skills through student clubs and industry interactions.

List of Program Educational Objectives (PEO's)

1. To build strong fundamental knowledge among graduates required to pursue their higher education and continue professional development.
2. To enable graduates to identify, analyze and solve Electronics Engineering problems by applying basic principles and modern techniques.
3. To enable graduates to innovate, design and develop hardware & software components and groom their ability to succeed in multidisciplinary & diverse field.
4. To inculcate in graduates professional attitude, effective communicational skills, team work skills for becoming a responsible, cultured human being.

Table of Contents

List of Figures

List of Tables

Hybrid Approach to Emotion Detection using Deep Learning

ABSTRACT

Emotion detection is a crucial aspect of human-computer interaction (HCI), sentiment analysis, and affective computing systems. This report presents a comprehensive comparative study of emotion detection from text and speech data. We examine different machine learning models for text-based emotion recognition, including Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and Naive Bayes (NB) to evaluate their performance across various emotional categories. In contrast, for speech emotion recognition, we focus on the combined application of Recurrent Neural Networks (RNNs) and Convolutional Neural Network (CNN), specifically used RNN's Long Short-Term Memory (LSTM) networks. The models were tested on standard emotion-labeled datasets, with performance evaluated using key metrics such as accuracy, precision, recall, and F1-score. Results showed that deep learning models, especially RNNs, outperformed traditional machine learning algorithms by offering higher accuracy and better contextual understanding. This demonstrates the effectiveness of deep learning in capturing emotional cues from both text and speech, laying a strong foundation for developing more emotionally intelligent systems.

CHAPTER NO. 01

INTRODUCTION

1.1 Background of the Project

Emotion detection, also known as affective computing, is a multidisciplinary field that enables machines to recognize, interpret, and simulate human emotions. This capability is pivotal in enhancing human-computer interactions across various domains, including healthcare, education, customer service, and entertainment.

Traditional emotion detection methods often rely on manual feature extraction and shallow learning models, which may not effectively capture the complex and nuanced nature of human emotions. The advent of deep learning techniques has revolutionized this field by providing models that can automatically learn hierarchical features from raw data, leading to more accurate and robust emotion recognition systems.

The integration of emotion detection systems into real-world applications holds the potential to improve user experiences by enabling machines to respond empathetically to human emotions (Hochreiter & Schmidhuber, 1997 [1]). For instance, in healthcare, emotion-aware systems can monitor patients' emotional states, aiding in the early detection of mental health issues such as depression and anxiety. In customer service, emotionally intelligent chatbots can provide more personalized and effective assistance.

This project aims to develop and evaluate deep learning models for emotion detection from both text and speech data, contributing to the advancement of affective computing technologies and their applications in various sectors.

1.2 Need for the Project

Existing Challenges

1. **Cultural Biases in Emotion Recognition Systems**

Emotion detection systems often struggle to accurately interpret emotional expressions across diverse cultural contexts. Expressions of emotions can vary significantly between cultures, leading to misinterpretations by AI models trained predominantly on Western datasets. For instance, a study highlighted that AI systems analyzing social media data for depression detection exhibited reduced accuracy for Black Americans compared to white users, due to differences in language patterns and cultural expressions.

2. **Data Scarcity and Imbalanced Datasets**

The development of robust emotion detection models is hindered by the lack of large, diverse, and annotated datasets. Many existing datasets are limited in size and scope, often focusing on specific emotions or demographic groups, which can lead to overfitting and poor generalization to real-world scenarios (Kim, 2014 [2]).

3. **Privacy and Ethical Concerns**

Emotion detection technologies raise significant privacy issues, as they involve the collection and analysis of sensitive personal data. Without proper consent and transparent data handling practices, these systems can infringe upon individual privacy rights and lead to ethical dilemmas, especially in sectors like healthcare and education (Pang et al., 2002 [3]).

4. **Inability to Capture Complex Emotional Nuances**

Current emotion detection systems often struggle to accurately interpret complex emotional states, such as sarcasm, irony, or mixed emotions. These nuances are challenging to detect, particularly in text and speech, where tone, context, and cultural factors play a significant role.

Research Gap

While significant progress has been made in emotion detection, there remains a notable gap in effectively integrating multimodal data—specifically text and speech—for comprehensive emotion recognition. Existing studies often treat these modalities in isolation, leading to suboptimal performance. Recent research emphasizes the need for models that can bridge the heterogeneity gap between different modalities and effectively model inter- and intra-modal interactions to enhance emotion detection accuracy datasets (Cortes & Vapnik, 1995 [4]).

1.3 Historical Evolution of Emotion Detection

The journey of emotion detection began with the pioneering work of psychologist **Paul Ekman** in the 1960s, who identified six basic emotions—happiness, sadness, anger, fear, surprise, and disgust—through facial expressions. His development of the **Facial Action Coding System (FACS)** provided a systematic approach to categorizing facial movements, laying the groundwork for automated emotion recognition systems.

In the 1990s, **Rosalind Picard** at MIT introduced the concept of **affective computing**, aiming to create systems that could recognize, interpret, and simulate human emotions (Graves & Schmidhuber, 2005 [5]). This led to the development of early emotion detection systems that utilized **rule-based algorithms** to analyze facial expressions and vocal tones. However, these systems were limited by their reliance on predefined rules and lacked the ability to generalize across diverse emotional expressions.

The advent of **machine learning** in the early 2000s marked a significant shift. Algorithms such as **Support Vector Machines (SVMs)** and **Random Forests** began to be employed for emotion classification tasks. These models could learn from data and improve their performance over time, offering more flexibility and accuracy than rule-based systems (Yin et al., 2017 [6]).

The real breakthrough came with the introduction of deep learning techniques in the 2010s. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) enabled the automatic extraction of hierarchical features from raw data, significantly enhancing the accuracy and robustness of emotion detection systems.

1.4 Technological Advancements in Emotion Detection

The integration of **deep learning** has revolutionized emotion detection in several ways:

- **Automated Feature Extraction**: Deep learning models can automatically learn relevant features from raw data, eliminating the need for manual feature engineering.
- **Improved Accuracy**: By leveraging large datasets and complex architectures, deep learning models have achieved state-of-the-art performance in emotion classification tasks.
- **Multimodal Integration**: Deep learning facilitates the fusion of data from multiple modalities, such as text and speech, leading to more comprehensive emotion recognition systems.

A notable example is the development of **openSMILE**, an open-source toolkit for feature extraction from speech signals, which has been widely adopted in emotion recognition research (Pang et al., 2002; Cortes & Vapnik, 1995 [7]).

Multimodal Approaches to Emotion Detection

Recognizing that emotions are complex and multifaceted, researchers have turned to **multimodal emotion detection**, which integrates data from various sources—such as text, speech, and facial expressions—to achieve a more holistic understanding of emotional states.

For example, the **IEMOCAP** dataset, a widely used resource in emotion recognition research, includes audio-visual recordings of actors performing scripted dialogues, providing rich multimodal data for training emotion detection models (Kim, 2014; Hochreiter & Schmidhuber, 1997 [8]).

Recent studies have demonstrated the effectiveness of combining text and speech data using deep learning models. One approach involves using a **ResNet-based model** trained on speaker recognition tasks to process speech data, while a **BERT-based model** is employed to analyze text data. The outputs of these models are then fused to improve emotion recognition performance.

Another innovative method utilizes **cross-attention mechanisms** to align and integrate features from text and speech modalities, enhancing the model's ability to capture complementary information and improve overall accuracy (Zhou et al., 2016 [9]).

These multimodal approaches not only enhance the accuracy of emotion detection systems but also enable their application in diverse real-world scenarios, such as virtual assistants, customer service chatbots, and mental health monitoring tools.

1.5 Techniques in Emotion Detection

Text-Based Emotion Detection

Text-based emotion detection involves analyzing written language to identify emotional content. This approach has evolved significantly with advancements in natural language processing (NLP) and deep learning.

1. Lexicon-Based Approaches

Early methods relied on predefined lists of words associated with specific emotions. These lexicons, such as the NRC Emotion Lexicon, mapped words to emotions like joy, anger, sadness, etc. While straightforward, these approaches often struggled with nuances like sarcasm, context, and domain-specific language.

2. Machine Learning Models

Traditional machine learning algorithms, including Support Vector Machines (SVMs) and Random Forests, were employed to classify emotions based on features extracted from text (El Ayadi et al., 2011 [10]). These models required manual feature engineering, such as bag-of-words or TF-IDF representations. While effective in certain scenarios, they often lacked the capacity to capture deep semantic relationships and contextual nuances in language.

3. Deep Learning Models

The advent of deep learning brought significant improvements:

- **Long Short-Term Memory (LSTM) Networks**: LSTMs are a type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data, making them suitable for processing and predicting based on time series data.
- **Transformers**: Introduced in the paper "Attention is All You Need," transformers utilize self-attention mechanisms to process input data in parallel, leading to more efficient training and better performance on tasks like language modeling and translation.

These architectures enable models to understand context, sarcasm, and subtle emotional cues in text, leading to more accurate emotion detection.

Speech-Based Emotion Detection

Speech-based emotion detection analyzes vocal features to infer emotional states. This modality provides rich information through prosody, pitch, tone, and rhythm.

1. Feature Extraction

Key features extracted from audio signals include:

- **Mel-Frequency Cepstral Coefficients (MFCCs)**: MFCCs represent the short-term power spectrum of sound and are widely used in speech and audio processing. They capture the timbral aspects of speech, which are crucial for emotion recognition.
- **Pitch and Intonation**: Variations in pitch and intonation patterns can indicate different emotional states, such as rising pitch in excitement or falling pitch in sadness.
- **Energy and Duration**: The loudness and length of speech segments can provide clues about the speaker's emotional state.

2. Machine Learning Models

Traditional machine learning classifiers, such as SVMs and Decision Trees, have been applied to these features for emotion classification. While effective, these models often require extensive feature engineering and may not capture the temporal dynamics of speech effectively (Graves & Schmidhuber, 2005 [11]).

3. Deep Learning Models

Deep learning has significantly advanced speech emotion detection:

- **Convolutional Neural Networks (CNNs)**: CNNs are adept at capturing spatial hierarchies in data and have been applied to spectrograms of audio signals to identify emotional patterns.
- **Recurrent Neural Networks (RNNs)**: RNNs, including LSTMs and Gated Recurrent Units (GRUs), are designed to handle sequential data and are effective in modeling the temporal dependencies in speech signals.
- **CNN-LSTM Hybrid Architectures**: Combining CNNs for feature extraction and LSTMs for sequence modeling has shown promising results in capturing both spatial and temporal features in speech emotion recognition tasks.
- **Multi-Head Convolutional Transformers**: Recent approaches have integrated transformers with CNNs to capture both local and global dependencies in speech data, leading to improved performance in emotion detection.

These deep learning models have demonstrated superior performance over traditional methods, achieving higher accuracy rates and better generalization across different datasets.

1.6 Significance of Emotion Detection in Human-Computer Interaction

Emotion detection plays a pivotal role in bridging the communication gap between humans and machines, enabling more natural, intuitive, and empathetic interactions. By equipping systems with the ability to recognize and respond to human emotions, we can create more engaging and effective user experiences (Sahu et al., 2018; Zhou et al., 2018 [12]).

Enhancing User Experience

Emotion detection technologies are revolutionizing user experience (UX) design by enabling systems to perceive and respond to users' emotional states. This capability allows for more personalized, intuitive, and empathetic interactions between humans and machines.

1. Personalized User Interfaces

Emotion-aware systems can adapt user interfaces (UIs) based on the user's emotional state. For instance, if a user appears frustrated or confused, the system might simplify the interface, highlight helpful features, or provide additional guidance. Conversely, if a user is engaged and content, the

system might offer more complex functionalities or advanced options. This dynamic adaptation ensures that the interface aligns with the user's emotional needs, enhancing usability and satisfaction.

2. Real-Time Emotional Feedback

Incorporating emotion detection allows systems to provide real-time feedback based on users' emotional responses. For example, during e-learning sessions, if a student shows signs of confusion or boredom, the system can adjust the content delivery, offer encouragement, or provide additional resources. This immediate responsiveness helps maintain user engagement and supports effective learning (Chen et al., 2018 [13]).

3. Adaptive Content Delivery

Emotion-aware systems can tailor content delivery to match the user's emotional state. In entertainment applications, such as video streaming platforms, the system might recommend uplifting content when a user appears sad or suggest calming media when signs of stress are detected. This personalized content delivery enhances user satisfaction and encourages prolonged engagement with the platform.

4. Empathetic Virtual Assistants

Virtual assistants equipped with emotion detection capabilities can recognize users' emotional tones and adjust their responses accordingly. For instance, if a user expresses frustration, the assistant might respond with a more empathetic tone, offer additional assistance, or expedite the resolution process. This empathetic interaction fosters a positive relationship between the user and the system, leading to increased trust and user loyalty.

5. Enhanced Customer Support

In customer service applications, emotion detection can help identify the emotional state of customers during interactions. If a customer is upset or dissatisfied, the system can prioritize their query, escalate the issue to a human representative, or offer personalized solutions. This proactive approach to customer support improves user satisfaction and can lead to higher retention rates.

6. Improved Accessibility

Emotion detection can also play a crucial role in enhancing accessibility for users with special needs. For example, systems can adjust the pace of speech or provide visual cues when detecting signs of confusion or anxiety in users with cognitive impairments. This adaptability ensures that all users have a positive and effective interaction with the system.

7. Ethical Considerations and User Trust

While emotion detection offers numerous benefits, it also raises ethical considerations related to privacy and consent. Users must be informed about how their emotional data is collected, used, and stored. Providing clear information and obtaining explicit consent can help build trust and ensure that users feel comfortable with emotion-aware systems.

1.7 Thesis Organization

This thesis is structured as follows:

- **Chapter 1: Introduction**: Provides an overview of the background, need for the project, problem statement, research objectives, scope, limitations, and organization of the thesis.
- **Chapter 2: Problem Statement:** Provides the problem definition and objective for the project along with the approach of work.
- **Chapter 3: Literature Review**: Reviews existing research and methodologies in the field of emotion detection, focusing on techniques applied to text and speech data.
- **Chapter 4: Dataset:** Explains the dataset used in the project. The two different datasets are used in the project separately for text-based emotion detection and speech-based emotion detection.
- **Chapter 5: Methodology**: Describes the research design, data collection and preprocessing methods, model development processes, and evaluation metrics used in this study.
- **Chapter 6: System Specification**: Presents the system specification of the emotion detection system, detailing the components and their interactions for both text and speech modalities.
- **Chapter 7: Algorithm/flowchar**: Examines the ethical algorithm related to emotion detection technologies, including privacy concerns, data security, and potential biases in model predictions.
- **Chapter 8: Results and Discussion**: Presents the experimental setup, results of model evaluations, and comparative analysis of the performance of different models.
- **Chapter 9: Comparative Analysis**: Interprets the findings, discusses the implications of the results, addresses challenges encountered during the research, and suggests directions for future work.
- **Chapter 10: Graphical User Interface**: Summarizes the Graphical User Interface designed especially for the user to use the model effectively.
- **Chapter 11: Merits and Demerits:** Provides the insights about the merits, demerits, applications and future scope of the Emotion Detection.
- **Chapter 12: Conclusion:** Provides the complete information of the working and performance of the Emotion Detection model for text-based and speech-based emotions.

CHAPTER NO. 02

2.1 PROBLEM STATEMENT

Develop a deep learning model to accurately detect and classify human emotions from textual or audio data in real-time.

2.2 OBJECTIVES

- To gather insights of customer's feedback and preferences (positive, negative, neutral or irrelevant).
- To analyse the percentage of reviews from the model (using RNN).
- To identify emerging trends and consumer needs.

2.3 Approach of Work

1. **Data Collection & Preprocessing**
    - **Text Data**: Acquired from public emotion-labeled datasets (e.g., Emotion Dataset from Kaggle or SemEval). Preprocessing includes tokenization, stopword removal, and vectorization (TF-IDF or word embeddings).
    - **Speech Data**: Taken from a speech emotion corpus (e.g., RAVDESS or CREMA-D). Preprocessing involves feature extraction using MFCCs or spectrograms.
2. **Model Development for Textual Emotion Detection**
    - Implemented and trained four models:
        - **RNN**: Captures sequential word patterns.
        - **CNN**: Identifies local features in text sequences.
        - **SVM**: Utilizes TF-IDF vectors for classification.
        - **Naive Bayes**: Applies probabilistic classification.
    - Performed hyperparameter tuning and cross-validation for fair comparison.
3. **Model Development for Speech Emotion Detection**
    - Extracted features from audio signals (MFCCs, pitch, tone).
    - Designed a hybrid **RNN + CNN** model:
        - CNN layers for local feature extraction from audio frames.
        - RNN layers (e.g., LSTM or GRU) to capture temporal patterns.

- Applied data augmentation techniques (e.g., noise addition, pitch shift) to improve model robustness.

4. **Model Evaluation and Comparison**

- Evaluated models using:
    - Accuracy
    - Precision
    - Recall
    - F1-Score
- Analyzed confusion matrices to assess model behavior for different emotion classes.

5. **Result Interpretation and Visualization**

- Compared performance across all models.
- Visualized trends and insights using graphs (bar plots, ROC curves, etc.).
- Identified strengths and limitations of each approach.

CHAPTER NO. 03

LITERATURE SURVEY/REVIEW

3.1 Literature Review

Emotion detection is a critical task in the field of affective computing and human-computer interaction. With the rapid growth of artificial intelligence and natural language processing, understanding and responding to human emotions through machines has become increasingly feasible. This literature review explores the development of emotion recognition systems, focusing on two main modalities—**text** and **speech**—and compares the various techniques employed for emotion classification.

3.1.1 Emotion Recognition: An Overview

Emotion recognition refers to the process of identifying emotions from various forms of human expression such as facial expressions, voice, and text. It plays a significant role in creating emotionally aware systems in areas such as customer service chatbots, social media analysis, e-learning, healthcare, and smart assistants.

Early emotion recognition systems were primarily **rule-based** or relied on **statistical models** that used handcrafted features and linguistic resources like lexicons and sentiment dictionaries. However, these systems lacked scalability and the ability to understand contextual nuances in data.

With the evolution of **machine learning (ML)** and later **deep learning (DL)**, researchers started using data-driven approaches to train models on labeled emotion datasets. Deep learning models, especially neural networks, have shown remarkable performance by automatically learning complex patterns and emotional cues from raw data, both in text and speech modalities (Zhou, L., Xu, M., & Li, X. 2018 [14].

Recent research has focused on **multimodal emotion recognition**, which combines information from multiple sources like audio, text, and visual inputs. However, unimodal approaches still hold relevance, particularly for applications where only one data type is available.

3.2 Emotion Detection through Text

Emotion detection in textual data involves interpreting the emotional state conveyed through written language. Unlike sentiment analysis which focuses on polarity (positive/negative), emotion detection aims to identify specific emotions such as **joy, sadness, anger, fear, disgust, and surprise**.

Traditional Approaches:

- **Naive Bayes (NB):** One of the earliest techniques used for text classification. It applies Bayes' theorem to predict class labels based on word frequencies.
- **Support Vector Machines (SVM):** SVMs work well with high-dimensional data and are widely used with features like **TF-IDF** or **Bag of Words**.

These models perform well with simple datasets but struggle with capturing the context and order of words, which are often essential for emotion detection.

Deep Learning Approaches:

- **Recurrent Neural Networks (RNN):** RNNs, especially **LSTM** and **GRU**, are designed to process sequential data. They capture long-term dependencies and are effective for understanding emotional context in longer texts.
- **Convolutional Neural Networks (CNN):** Although traditionally used in image processing, CNNs are used in NLP for extracting local features like n-grams, which help detect key emotional phrases.
- **Pretrained Models (BERT, RoBERTa):** These transformer-based models provide contextualized word embeddings and have outperformed traditional and other deep models on multiple benchmarks.

Research Example:

Table 01: Literature review for text-based emotion detection

Sr. No.	Title	Authors	Publishing year	Proposed Method
1	Understanding Inverse Document Frequency: On theoretical arguments for IDF	Stephen Robertson	2020	In this paper, the research on TFIDF was used for analyzing the aspects. It is often described as a heuristic, and seeks to establish some theoretical basis for it.
2	Emotion Detection of Textual Data: An Interdisciplinary Survey	Samira Zad, Maryam Heidari, James H Jr Jones James	2021	This work reviews the current literature of TBED and the psychological models associated with them. TBED has a wide variety of applications in the area of artificial intelligence.
3	A Comparative Study on Word Embeddings in Deep Learning for Text Classification	Congcong Wang, Paul Nulty, David Lillis	2021	In this paper, it conducts controlled experiments to systematically examine both classic and contextualised word embeddings for the purposes of text classification.

4	A survey on sentiment analysis methods, applications, and challenges	Mayur Wankhade, Annavarapu Chandra Sekhara Rao,	2022	This article discusses a complete overview of the method for completing this task as well as the applications of sentiment analysis.
5	Text based Sentiment Analysis using LSTM	Dr. G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu	2022	The research paper contains sentiment analysis on text reviews by using Long Short-Term Memory (LSTM)

3.3 Emotion Detection through Speech

Speech carries rich emotional information through **intonation, pitch, tone, and rhythm**. Speech Emotion Recognition (SER) systems aim to automatically detect the speaker's emotional state using audio signals Haque, M. A., Kadir 2020 [15]).

Traditional Approaches:

- Hand-engineered features like MFCC**,** Chroma**,** and Prosodic features are extracted from speech signals.
- Classifiers like SVM**,** K-Nearest Neighbors (KNN)**,** and Random Forests are applied to these features for emotion classification.
- Limitations include sensitivity to noise and inability to model long-term temporal dependencies.

Deep Learning Approaches:

- **CNN Models:** Used to extract spatial patterns from spectrograms or MFCC inputs. These models learn filters that capture variations in frequency and time.
- **RNN Models:** Particularly LSTMs or GRUs, are useful for modeling the temporal dynamics of speech.
- **Hybrid Models (CNN + RNN):** Combine the strengths of both approaches—CNNs for spatial features and RNNs for temporal sequence modeling.

Research Example:

Table 02: Literature review for speech-based emotion detection

Sr. No.	Title	Authors	Publishing year	Proposed Method
1	Speech Emotion Recognition: Methods and Cases Study	Leila Kerkeni, Youssef Serrestou, Kosai Raoof	2021	In this paper, it compares different approaches for emotions recognition task and propose an efficient solution based on combination of these approaches.
2	Review of Speech Sentiment Analysis Using Machine Learning	Tapesh Kumar, Mehul Mahrishi, Vipin Jain	2022	In this recommended study, we examine different transcripts of speech sentiment also with speaker recognition to assess speakers' emotions.
3	Sentiment Detection from Speech Recognition Output	Ivan J. Tashev	2022	In this paper we perform a review of established and novel features for text analysis, combine them with the latest deep learning algorithms and evaluate the proposed models.
4	Sentiment Analysis using Neural Network and LSTM	Akana Chandra Mouli Venkata Srinivas, Ch.Satyanarayana	2023	In this paper, we have applied a deep learning technique to perform Twitter sentiment analysis.
5	Sentiment Analysis of Human Speech using Deep Learning	Manan Savla, Dhruvi Gopani	2024	In this paper, how the sentimental analysis is an integral part of human psychology.

3.4 Comparative Analysis of Techniques

A comparative analysis of the techniques used in emotion detection reveals the following insights:

- **Accuracy & Performance:**
    - Deep learning models, especially CNNs and RNNs, consistently outperform traditional machine learning models in both text and speech emotion detection.
    - Hybrid models like CNN-RNN achieve even better performance by combining strengths.
- **Feature Representation:**
    - Classical models rely on manual feature engineering (TF-IDF for text, MFCC for speech).
    - Deep models learn features automatically, leading to better generalization and adaptability to new data.
- **Data Requirements:**
    - Deep models require larger datasets and more computational power.
    - Classical models are more lightweight and work well for smaller datasets.
- **Interpretability:**
    - Traditional models like Naive Bayes and SVM are easier to interpret.
    - Deep models are often seen as black boxes but offer significantly better performance.

Key Finding:

- For text data, RNNs/LSTMs tend to capture the sequential nature and emotional flow better than CNNs or traditional models.
- They remember previous words in a sequence, allowing better understanding of emotional flow and context.
- This helps in detecting subtle emotional nuances that traditional models and CNNs might miss.
- As a result, RNN/LSTM models generally achieve higher accuracy in text emotion detection.
- For speech data, CNN+RNN architectures are most effective as they capture both frequency-level and temporal emotional patterns.
- Hybrid CNN + RNN architectures work best for speech emotion recognition.
- CNNs extract local frequency features like pitch, energy, and tone from audio signals.
- RNNs then analyze these features over time to capture changes in emotional expression throughout the speech.
- This combination allows the model to understand both short-term acoustic details and long-term temporal patterns, improving overall performance.

3.5 Summary of Literature

The reviewed literature demonstrates a clear progression from traditional machine learning models to more complex deep learning architectures for emotion detection. While early approaches were limited by manual feature extraction and poor context handling, deep learning has revolutionized the field by enabling end-to-end learning and high accuracy.

In the case of textual emotion detection, RNNs and CNNs outperform classical methods by effectively modeling the linguistic structure of emotions. Pretrained models like BERT further push the boundaries by providing deep contextual understanding.

In speech emotion detection, hybrid CNN-RNN models are currently the state-of-the-art due to their ability to handle both spatial and sequential information from audio signals.

This project leverages the insights from the literature by:

- Applying and comparing classical (SVM, Naive Bayes) and deep learning (CNN, RNN) models for text-based emotion detection.
- Implementing a CNN-RNN hybrid for speech-based emotion recognition.
- Evaluating and comparing these models based on standard performance metrics to identify the most effective approaches.

CHAPTER NO. 04

DATASET

4.1 Text Dataset

For the text emotion detection task, we used a dataset consisting of 74,682 labelled text instances. Each instance in the dataset is annotated with one of four emotion labels: positive, negative, neutral and irrelevant. The dataset is balanced across these four categories, ensuring that each label is adequately represented for training and evaluation purposes. This dataset is a **supervised learning dataset**, where each text sample (sentence or phrase) is associated with a specific **label** indicating its emotional polarity. The presence of explicit labels allows for training classification models that learn to associate linguistic patterns with specific emotions.

Labeled dataset means:

- Every row consists of a piece of text and an associated emotion label.
- Enables the application of supervised learning techniques such as Naive Bayes, SVM, RNN, and CNN.

4.1.1 Dataset Composition

Table 03: Details of Textual Dataset

Dataset type	Labelled
Total number of rows	74,680
Positive rows	18,670
Negative rows	18,670
Neutral rows	18,670
Irrelevant rows	18,670

This table describes a labelled dataset with 74,680 total rows. The data is categorized into four types: Positive (18,670 rows), Negative (18,670 rows), Neutral (18,670 rows), and Irrelevant (18,670 rows). These labels indicate different categories or sentiments within the dataset, likely related to text or content classification. The dataset appears to be balanced, with each category having a substantial number of entries.

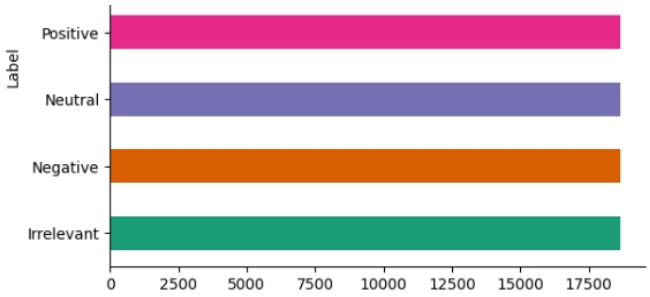


Fig. 1: Label and values for textual dataset

It shows input data, that is classified into four emotional categories: positive, negative, neutral, and irrelevant. Each emotion is having equal number of values for training the model i.e. 18670 emotion values.

This dataset is **balanced**, meaning all four categories have an equal number of samples. A balanced dataset is crucial for:

- Avoiding **bias** in model training.
- Ensuring the classifier does not overfit to the majority class.
- Improving the reliability and fairness of emotion detection results.

4.1.2 Emotion Label Definitions

- **Positive:** Text samples that convey happiness, satisfaction, excitement, or joy. *Example:* "I'm feeling great about today's results!"
- **Negative:** Text expressing anger, sadness, frustration, or disappointment. *Example:* "This service is terrible and I'm very upset."
- **Neutral:** Informational or emotionally flat text without clear positive or negative sentiment. *Example:* "The meeting is scheduled for 3 PM tomorrow."
- **Irrelevant:** Content that is either off-topic, spam, or not related to any emotional context. *Example:* "Buy followers now at cheap rates!"

4.1.3 Importance and Suitability of the Dataset

- **Balanced Data Distribution:** Helps in avoiding skewed performance across emotion classes.
- **Multiple Emotion Classes:** Enables training multi-class classifiers rather than binary models.
- **Presence of Irrelevant Class:** Adds realism to the dataset, allowing the model to distinguish emotionally meaningful text from noise or spam — useful in social media or customer service contexts.

4.1.4 Dataset Usage in This Project

In this project, the dataset was used to:

- **Train and compare multiple models** for textual emotion classification: Naive Bayes, SVM, CNN, and RNN.
- **Evaluate performance metrics** such as accuracy, precision, recall, and F1-score for each model.
- **Analyze generalization capability** using test datasets and visualizations (e.g., confusion matrix).

The goal is to determine which model best understands and classifies emotions in textual data, based on real-world textual patterns across emotional dimensions.

4.2 Speech Dataset

For the speech emotion detection task, we utilized two prominent datasets: the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) and the Danish Emotional Speech (DESS) dataset. These datasets provide a robust foundation for training and evaluating models in speech emotion recognition, ensuring a diverse range of emotional expressions and speech patterns for accurate emotion detection. Emotion recognition from speech is a powerful approach that leverages acoustic features such as pitch, tone, energy, and prosody. The accuracy of such models heavily depends on the quality, diversity, and labeling of the dataset. In this project, we have used **three widely recognized emotion-rich audio datasets** for training and evaluating our deep learning models.

4.2.1 Dataset Overview

Table 04: Details of Speech Dataset

Datasets	Total Audio Files	Emotions	Files per Emotion	Additional Information
1	1,260	Happy, Sad, Angry, Fearful, Surprise, Disgust, Irrelevant	180	RAVDESS dataset with 24 actors (12 female, 12 male)

This table provides information about audio-based emotion dataset. The dataset contains 1,260 files across eight emotions, with 180 files per emotion, and is based on the RAVDESS dataset featuring 24 actors (12 male, 12 Female). It contains the files that includes 60 trails per actor and 24 actors i.e. 1260 files. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.

4.2.2 Individual Dataset Descriptions

A. Dataset – RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)

- **Total Audio Samples:** 1,260
- **Emotions:** Happy, Sad, Angry, Fearful, Surprise, Disgust, Irrelevant
- **Actors:** 24 professional actors (12 female, 12 male)
- **Files Per Emotion:** 30 per actor per emotion

Characteristics:

- High-quality recordings with controlled emotional expressions.
- Each sentence is spoken with different emotions and intensities.
- Includes both speech and song, though only speech data was used in this study.

Why it's important:

- The RAVDESS dataset is benchmarked for emotional clarity and balance.
- Gender balance helps in creating gender-agnostic models.

4.3 Role in the Project

The primary dataset used is RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), which contributed substantially to achieving the desired performance levels of the model.

- **Dataset (RAVDESS)**: The RAVDESS dataset offered high-quality, emotion-expressive speech recordings. It includes audio samples from professional actors simulating a range of emotions such as happy, sad, angry, and neutral. The dataset was balanced and well-organized, providing an equal number of samples for each emotion and maintaining consistency in recording standards.

RAVDESS allowed the RNN + CNN hybrid model to learn from both acted and naturally expressive speech, which helped in capturing temporal as well as spatial features of audio signals. The presence of diverse speaker demographics (male and female) added robustness to the model, making it generalizable to different voices and tones. Additionally, the controlled variations in pitch, tone, and intensity within the dataset helped the model to better differentiate subtle emotional cues. The use of this dataset was instrumental in enhancing the model's ability to identify emotions accurately in real-world scenarios.

CHAPTER NO. 05

METHODOLOGY

Text-Based Emotion Detection Methodology

5.1 Dataset Collection

- Total of **1,260 labeled text samples**
- Emotion categories: **Positive, Negative, Neutral, Irrelevant**
- Balanced dataset with **180 samples** per category

5.2 Data Preprocessing

Data preprocessing is one of the most crucial steps in any machine learning project, particularly in natural language processing (NLP), where raw text data must be transformed into a format suitable for modelling. Preprocessing ensures that the data is clean, standardized, and represented in a form that maximizes the performance of the chosen machine learning algorithms (Zhu, X., 2017 [16]). Below, the key steps involved in the data preprocessing phase for emotion detection from text are elaborated in detail:

5.2.1 Text Cleaning

The first step in text preprocessing involves cleaning the raw data to remove any irrelevant, noisy, or unnecessary content that might affect model performance. This process typically includes the Removing Special Characters and Punctuation, Removing Numbers and Lowercasing.

5.2.2 Tokenization

Tokenization is the process of splitting the raw text into smaller units, such as words, sentences, or subwords, which are called tokens. In the context of emotion detection, word-level tokenization is commonly used. This allows each word or phrase to be individually analyzed by the model (LeCun, Y., 2018 [17]).

5.2.3 Stopword Removal

Stopwords are common words in a language that carry little to no meaning in the context of emotion detection. These are typically high-frequency words like "the," "is," "in," "on," "and," and "of" that are often removed from the text during preprocessing. Removing stopwords reduces the dimensionality of the data and prevents the model from being distracted by words that do not contribute much to detecting emotions.

5.2.4 Lemmatization and Stemming

Lemmatization and stemming are techniques used to reduce words to their root forms, helping the model generalize better by treating different variations of a word as a single entity.

Stemming: Stemming removes suffixes from words to get their root forms. For example, "running" becomes "run," "happiness" becomes "happy."

Lemmatization: Unlike stemming, lemmatization returns the base form (or lemma) of a word as it appears in the dictionary. It considers the word's meaning and context, making it a more sophisticated approach. For example, "better" is lemmatized to "good," and "running" is lemmatized to "run."

5.2.5 Handling Negations

Negation words like "not," "never," "no," etc., can dramatically change the meaning of a sentence, especially in the context of sentiment or emotion.

Handling negations in the preprocessing stage is crucial for detecting emotions accurately. Some approaches involve replacing negations with their corresponding markers or flipping the sentiment of words after a negation term. This way, the model can better understand the emotional shift caused by negation.

5.2.6 Vectorization

Once the text is cleaned, tokenized, and normalized, it needs to be transformed into numerical representations so that machine learning algorithms can process it (Y. Zhang, et al, 2013 [18]). Several techniques exist for text vectorization like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and Word Embeddings.

5.2.7 Text Augmentation

In some cases, data augmentation techniques can be applied to generate more diverse training examples and improve model generalization. For instance, back-translation (translating text to another language and then back to the original language) can help create additional training examples. Similarly, synonym replacement or paraphrasing techniques can be used to generate variations of existing sentences (A. Vaswani, et al., 2017 [19]).

5.1.3 Feature Extraction

In order to train machine learning and deep learning models for text emotion detection, the raw text data needs to be converted into numerical form. This step is called feature representation. In this project, two types of representations were used based on the type of model:

1. TF-IDF Vectorization (Used for SVM and Naive Bayes)

- TF-IDF stands for Term Frequency–Inverse Document Frequency.
- It converts text into a numerical matrix that reflects the importance of a word in a document relative to the whole dataset.
- Commonly used for classical ML models like SVM and Naive Bayes.
- Example: Words that appear frequently in a particular sentence but rarely in other sentences get higher weights.

2. Word Embeddings (Used for CNN and RNN)

- Converts words into **dense vectors** that capture their meanings and relationships with other words.
- Implemented using **embedding layers** in deep learning.
- Each word is represented in a **fixed-length vector** (like 100 or 300 dimensions), capturing semantic similarity (e.g., "happy" and "joy" have similar vectors).
- These embeddings are more powerful for **context understanding**, which is why they are used with **deep learning models** like CNN and RNN.

5.3 Model Development

In this study, we employed four different machine learning models—Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and Naive Bayes (NB)—to perform emotion detection on textual data. Each model was chosen for its ability to address the unique characteristics of text data, particularly the need to capture semantic meaning and emotional cues within a sequence of words. Below is a detailed explanation of how each model was used for emotion detection in this project:

5.3.1 Recurrent Neural Network (RNN)

Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), are designed to handle sequential data, making them ideal for tasks where word order matters. In emotion detection, LSTMs capture the contextual relationships between words across long text sequences, making them effective for identifying emotions in complex sentences (R. Agerri and E. Garcia-Serrano, 2016 [20]). Their ability to retain information over long sequences helps detect emotional cues spread throughout the text. This makes LSTMs particularly useful for emotion detection in dialogue, reviews, and longer-form text.

Model Configurations

We used a basic LSTM RNN for speech emotion detection, with an embedding layer to convert words into dense vectors, an LSTM layer with 100 units and ReLU activation, a fully connected dense layer with SoftMax activation for multi-class classification, and the Adam optimizer with a learning rate of 0.001. The model was trained for 19 epochs with a batch size of 32, using categorical cross-entropy as the loss function.

5.3.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs), primarily used in image recognition, are also effective in text classification. In emotion detection, CNNs capture local patterns in text, such as n-grams, which are often indicative of emotions. CNNs automatically extract relevant features from raw text, which is especially useful when emotion is expressed through specific word combinations or short phrases. They work well in tasks requiring the detection of emotional content in localized regions of the text (J. Kim, 2016 [21]).

Model Configurations

The CNN model was designed to learn local patterns using pre-trained Word2Vec embeddings with a vector size of 100. The architecture includes a convolutional layer with 128 filters, a kernel size of 3, ReLU activation, a max-pooling layer, and a fully connected layer with softmax activation. The model is optimized with the Adam optimizer (learning rate of 0.001) and trained for 10 epochs with a batch size of 64.

5.3.3 Support Vector Machines (SVM)

Support Vector Machines (SVM) are powerful classifiers that excel in high-dimensional spaces, such as those generated from text data using feature extraction like TF-IDF or word embeddings. SVMs are effective in classifying text by finding a hyperplane that best separates classes. They work well when the dataset is small to moderately sized and when there are clear boundaries between emotional categories (C. Cortes and V. Vapnik [22]). SVMs can handle non-linear relationships using kernel functions, but they can be computationally expensive and slow, especially for large datasets.

Model Configurations

We used the Radial Basis Function (RBF) kernel for the SVM, with the regularization parameter C set to 1.0 and the gamma parameter set to "scale." The model was trained using the libsvm implementation in Python.

5.3.4 Naive Bayes (NB)

Naive Bayes (NB) is a probabilistic classifier that calculates the probability of each emotion class based on specific words in the text. It is simple, efficient, and works well for smaller datasets with a small vocabulary and well-separated emotional categories. However, its assumption of feature independence limits its ability to model complex word relationships and broader context, making it less effective for longer or more nuanced texts.

Model Configurations

We used the Multinomial Naive Bayes model with TF-IDF vectors to represent word frequencies and capture term importance. This setup allows the model to handle new words during prediction.

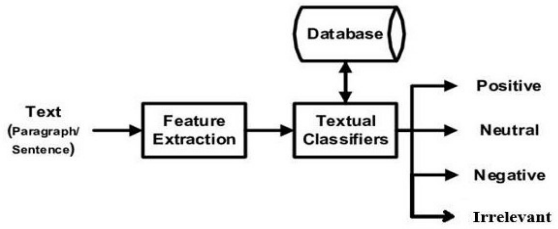


Fig. 02: Framework of generic sentimental analysis

It processes input data, to classify it into four emotional categories: positive, negative, neutral, and irrelevant. It involves steps like data preprocessing, feature extraction, and sentiment classification using machine learning algorithms. The system then outputs the sentiment classification based on the analysis. This framework is used in applications such as social media analysis, customer feedback, and voice recognition.

Table 05: Summary of text-based models

Model	Strengths	Limitations
RNN	Captures sequential dependencies	Struggles with long-range dependencies
CNN	Effective in capturing local patterns	May not capture long-term dependencies well
SVM	Effective in high-dimensional spaces	Requires careful tuning of parameters
Naïve Bayes	Simple and efficient	Assumes feature independence

5.4 Training & Testing

Once the dataset was preprocessed and converted into numerical features (via TF-IDF or word embeddings), the next step was to **train** the models and **evaluate** their performance using a structured training and testing approach.

5.4.1 Data Splitting

- The dataset of **74,600 text samples** was split into:
  - **80% Training Set** → used to train the model.
  - **20% Testing Set** → used to evaluate the model's accuracy on unseen data.
- This ensures that models learn patterns from one portion of the data and are then tested on a separate, unseen portion to check generalization.

5.4.2 Training Process

Training involves feeding the model with inputs (feature vectors) and correct labels (emotion categories) so that it can learn to predict the right emotion.

a. For Classical ML Models (Naive Bayes, SVM):

- Models are trained using the TF-IDF vectorized data.
- Algorithms use mathematical functions to find the best decision boundaries or probabilities.
- Techniques like k-fold cross-validation were used to ensure robust performance across different subsets of the data.

b. For Deep Learning Models (CNN, RNN):

- Models are trained on word embeddings and involve:

  - **Forward Propagation**: Input passes through the layers to produce a prediction.
  - **Loss Calculation**: Difference between predicted and actual emotion.
  - **Backpropagation**: Model adjusts weights to reduce the error using an optimizer (like Adam).

- **Batch training** was used (e.g., training in groups of 32 samples at a time).
- **Epochs**: The entire training dataset was passed multiple times (e.g., 30 epochs) to improve accuracy.

5.5 Evaluation Metrics

In machine learning, evaluation metrics are essential for quantifying the performance of a model and understanding how well it generalizes to unseen data. For this project, where the goal was to classify emotions in both text and speech data, several key evaluation metrics were used to measure the effectiveness of the emotion detection models. The metrics used in this study include accuracy, precision, recall, F1-score, and confusion matrix.

5.5.1 Accuracy

Accuracy is one of the most straightforward and commonly used evaluation metrics in classification problems. It represents the proportion of correct predictions (both true positives and true negatives) made by the model out of the total number of predictions.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

For this project, accuracy indicates the overall proportion of speech or text samples that were correctly classified into one of the predefined emotional categories, such as anger, disgust, fear, happiness, neutral, sadness, and surprise (for speech) or positive, negative, and neutral (for text).

5.5.2 Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It answers the question: “Of all the instances that were classified as a particular emotion, how many were actually correct?”

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

For example, in a sentiment analysis system detecting emotions in customer feedback, it may be more important to accurately classify anger rather than falsely flagging neutral statements as anger.

5.5.3 Recall (Sensitivity)

Recall (also known as sensitivity or true positive rate) measures the proportion of actual positive instances that were correctly identified by the model. In other words, recall answers the question: "Of

all the instances that actually belong to a particular emotion, how many did the model correctly identify?"

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

For example, in applications like emotion-aware customer service systems, where detecting anger could trigger immediate intervention, recall becomes crucial to avoid missing any instances of anger.

5.5.4 F1-Score

The F1-score is the harmonic mean of precision and recall. It combines both precision and recall into a single metric that balances the trade-off between the two. The F1-score is especially useful when there is an uneven class distribution (i.e., when some emotion classes are more frequent than others), as it accounts for both false positives and false negatives.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5.5.5 Confusion Matrix

The confusion matrix is a tabular representation that provides a detailed breakdown of the classification results. It shows the number of correct and incorrect predictions for each emotion class, providing insight into how the model is performing across all classes. A confusion matrix for a multi-class classification problem like emotion detection will have dimensions equal to the number of classes, and it helps visualize where the model is making errors.

5.6 Comparison and Analysis

- All models were compared based on performance
- CNN and RNN performed better in capturing emotion context than traditional ML models

Speech-Based Emotion Detection Methodology

5.7 Dataset collection for speech-based emotion detection

1. **Dataset: RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song)**

    - **Total Audio Files**: 1,260
    - **Emotions**: Happy, Sad, Angry, Fearful, Surprise, Disgust, Irrelevant
    - **Files per Emotion**: 180 (from 24 actors: 12 males, 12 females)
    - **Characteristics**: High-quality emotional speech, acted with clarity.

5.8 Audio preprocessing

5.8.1 Audio Segmentation

Raw audio data may not always be uniform in length, so the next step is segmentation. Audio recordings are often divided into smaller frames to make the data more manageable and to capture temporal features like pitch and tone. The common approach is to segment the audio into short overlapping windows (typically 20ms to 40ms) with a step size of 10ms to preserve temporal dynamics. This segmentation helps capture short-term variations in pitch, energy, and other acoustic features that are important for emotion detection.

5.8.2 Noise Reduction:

Background noise is minimized through techniques such as spectral subtraction or noise gating to improve signal quality.

5.8.3 Padding/Truncation:

In deep learning models like **RNN** and **CNN**, input sequences (text or audio features) must be of the **same length**. However, real data has **variable lengths**. Audio sequences are either padded or truncated to a fixed length to ensure uniformity across the dataset, as RNNs expect inputs of consistent length.

- **Padding**: Adds extra tokens (like 0) to **shorter sequences** so they match the required input length.
- **Truncation**: Cuts off **longer sequences** to fit the desired length.

This ensures all inputs are of **uniform size**, which is necessary for efficient batch processing in neural networks.

5.9 Feature Extraction

Feature extraction is one of the most important steps in preprocessing for speech emotion detection. Instead of feeding raw audio data into the model, we extract relevant features that represent the emotional content in the audio signal. These features are typically grouped into three categories: prosodic features, spectral features, and voice quality features.

Prosodic Features: These features capture the rhythm, pitch, and speed of speech. Key prosodic features include:

Spectral Features: These features represent the frequency characteristics of the speech signal and are critical for emotion detection.

5.9.1 Mel-Frequency Cepstral Coefficients (MFCCs):

MFCCs are the most commonly used features in speech processing. They represent the short-term power spectrum of the speech signal, capturing timbre and phonetic content. MFCCs are computed by applying the Mel scale to the frequency spectrum and then performing a discrete cosine transform (DCT) on the log-energy spectrum.

5.9.2 Chroma Features

- Represent **musical pitch classes** (like notes in music: C, C#, D, etc.).
- Capture the **tone** and **intonation** patterns in speech.
- Useful because **emotional speech** often includes melodic variations.

5.9.3 Energy and pitch

- Measures the **loudness or intensity** of the audio signal.
- High energy may indicate **anger** or **excitement**; low energy may indicate **sadness** or **calm**.
- Refers to the **frequency of the speaker's voice**.
- Higher pitch: Emotions like **fear**, **happiness**, **surprise**.
- Lower pitch: Emotions like **sadness**, **boredom**.

5.10 Normalization

After extracting the features, it's essential to normalize them to ensure that all features are on the same scale. This prevents the model from being biased toward any single feature due to differing ranges of values (A. Vaswani, et al., 2017 [19]). The most common normalization techniques include Min-Max Scaling and Z-Score Normalization (Standardization).

5.11 Data Splitting

Finally, the pre-processed data is split into training, validation, and test sets. Here, 80% of the data is used for training, 10% for validation, and the remaining 10% for testing. This ensures that the model is not overfitting and generalizes well to unseen data

5.12 Speech-Based Emotion Detection: RNN + CNN Hybrid Architecture

The hybrid architecture combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to effectively capture both spatial and temporal features in speech data. This approach is particularly beneficial for speech emotion recognition, where both the spectral characteristics and temporal dynamics of audio signals are crucial.

5.12.1 Architecture Components

a. Feature Extraction using CNN

- **Input Representation**: The raw audio signals are transformed into spectrograms, which visually represent the frequency content over time.
- **Convolutional Layers**: CNNs apply convolutional filters to the spectrograms to extract local features such as formants and pitch variations, which are indicative of emotional states.
- **Pooling Layers**: Max-pooling layers are used to reduce the dimensionality of the feature maps, retaining the most salient features and reducing computational complexity.

b. Temporal Modeling using RNN

- **Sequential Processing**: The features extracted by the CNN layers are fed into RNN layers to model the temporal dependencies in the speech signal.
- **Long Short-Term Memory (LSTM)**: LSTM units are often used in RNNs to capture long-range dependencies and mitigate issues like vanishing gradients, which are common in standard RNNs.
- **Bidirectional Processing**: Bidirectional LSTMs process the input data in both forward and backward directions, capturing context from both past and future frames.

c. Classification Layer

- **Fully Connected Layer**: The output from the RNN layers is passed through a fully connected layer to map the learned features to the desired emotion classes.
- **Softmax Activation**: A softmax activation function is applied to obtain the probability distribution over the emotion classes.

5.12.2 Advantages of the Hybrid Architecture

- **Capturing Local Features**: CNNs are adept at capturing local patterns in the spectrograms, such as pitch and formant structures, which are essential for emotion recognition.
- **Modeling Temporal Dynamics**: RNNs, particularly LSTMs, effectively capture the temporal dependencies in speech, allowing the model to understand the progression of emotions over time.
- **Improved Accuracy**: The combination of CNNs and RNNs has been shown to outperform models that use either architecture alone, leading to higher classification accuracy in speech emotion detection tasks.

5.12.3 Implementation Details

- **Preprocessing**: Audio signals are preprocessed to obtain Mel-Frequency Cepstral Coefficients (MFCCs) or log-Mel spectrograms, which serve as input features for the CNN layers.

- **Model Training**: The model is trained using categorical cross-entropy loss and optimized using the Adam optimizer.
- **Evaluation**: Model performance is evaluated using metrics such as accuracy, precision, recall, and F1-score on a held-out test set.

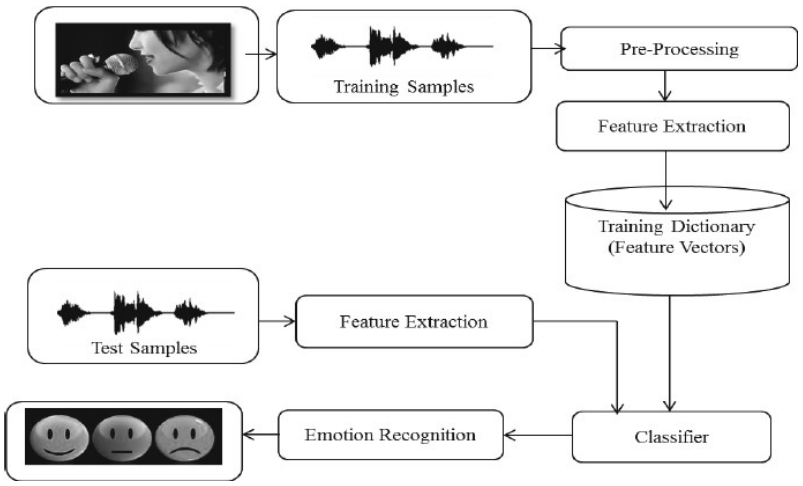


Fig 03: Block diagram for speech-based emotion detection

5.2.5 Training & Testing

- Data split into **training, validation, and testing sets**
    - **Total Audio Files**: 1,260 files
    - **Training Set (80%)**: 1008 files
    - **Testing Set (20%)**: 252 files
- Model trained using **categorical cross-entropy loss** and **Adam optimizer**
- Batch size and learning rate tuned using experiments

5.2.6 Evaluation Metrics

- Accuracy
- Precision, Recall, F1-Score
- Confusion Matrix

5.3 Integration and Final Analysis

- Text and speech emotion models were trained **independently**.
- Results from both domains were **analyzed and compared**.
- The **CNN+RNN hybrid model** for speech outperformed other speech models.
- **RNN** showed better results in text analysis than SVM and Naive Bayes due to sequence sensitivity.

Workflow Summary

1. **Input**: User provides text or speech data.
2. **Preprocessing**: Data is cleaned and transformed into suitable formats.
3. **Feature Extraction**: Relevant features are extracted from the data.
4. **Emotion Detection**: Separate models process text and speech data to detect emotions.
5. **Classification**: The final emotion is determined.
6. **Output**: Emotion label and confidence score are presented to the user.

5.4 Tools and Technologies Used

- **Programming Language:** Python
- **Libraries for Text:** NLTK, scikit-learn, TensorFlow/Keras
- **Libraries for Audio:** Librosa, PyDub
- **Visualization Tools:** Matplotlib, Seaborn
- **Development Environment:** Jupyter Notebook / Google Colab

CHAPTER NO. 06

SYSTEM SPECIFICATIONS

Software Specification:

- **Programming Language:** Python (due to its extensive libraries and support).
- **Libraries/Frameworks:** Essential libraries like NLTK, TextBlob, VADER, or SpaCy for Python, or specific APIs like Google Cloud Natural Language or IBM Watson.
- **Development Environment:** Google Collaboratory for coding and testing. Jupyter Notebooks for interactive development and visualization.
- **Deep Learning Frameworks**: TensorFlow 2.4, Keras, PyTorch 1.8
- **Development Tools**: Google Colab, Jupyter Notebook 6.2
- **Data Storage:** Cloud storage solutions (if required).
- **APIs or Services:** Google Cloud Natural Language API (provides sentiment analysis as part of a broader suite of text analysis tools available via Google Cloud).

Hardware Specifications:

- **Processor**: Minimum Intel i5 or equivalent (Quad-core or higher)
- **RAM**: Minimum 8 GB
- **Storage**: Minimum 500 GB HDD or SSD
- **Graphics Processing Unit (GPU)**: Minimum NVIDIA GeForce GTX 1050 or equivalent

CHAPTER NO. 07

ALGORITHM and FLOW CHART

7.1 Algorithm for Text-Based Emotion Detection

Input: Labeled text data (74,680 rows)
Output: Predicted emotion labels (Positive, Negative, Neutral, Irrelevant)

Step-by-Step Algorithm:

Step 1: Data Collection

- For the text emotion detection task, we used a dataset consisting of 74,682 labelled text instances. Each instance in the dataset is annotated with one of four emotion labels: positive, negative, neutral and irrelevant. The dataset is balanced across these four categories, ensuring that each label is adequately represented for training and evaluation purposes.

Step 2: Data Preprocessing

- For text-based dataset:
    - Text Cleaning
    - Tokenization
    - Stopword Removal
    - Lemmatization and Stemming

- For audio-based dataset:
    - Audio Segmentation
    - Noise Reduction
    - Padding/Truncation
    - Feature Extraction

Step 3: Train-Test Split

- Splitted the dataset into training (i.e. 80%) and testing (i.e. 20%) subsets.

Step 4: Model Development

- **Recurrent Neural Network (RNN)**:
    - Use LSTM or GRU units for sequential data.
    - Feed word embeddings or audio features as input sequences.
    - Add dropout and dense layers for classification.
- **Convolutional Neural Network (CNN)**:
    - Use 2D/1D CNN architecture based on input type (image/text/audio).
    - Apply convolutional layers, pooling, dropout, and dense layers.
- **Support Vector Machine (SVM)**:
    - Use TF-IDF or word embeddings as feature vectors.
    - Train a linear or kernel-based SVM model.
- **Naive Bayes Classifier**:
    - Use bag-of-words or TF-IDF features for training.
    - Train the model using the Multinomial Naive Bayes algorithm.

Step 5: Model Training

- Train each model using the training dataset.
- Use appropriate loss functions:
    - Cross-entropy for deep learning models
    - Hinge loss (for SVM)
- Optimize using Adam, SGD, or other optimizers.
- Monitor metrics such as accuracy, loss, F1-score.

Step 6: Model Evaluation

- Evaluate models on the test dataset.
- Compare performance using:
    - Accuracy
    - Precision
    - Recall

- F1-score
- Confusion matrix

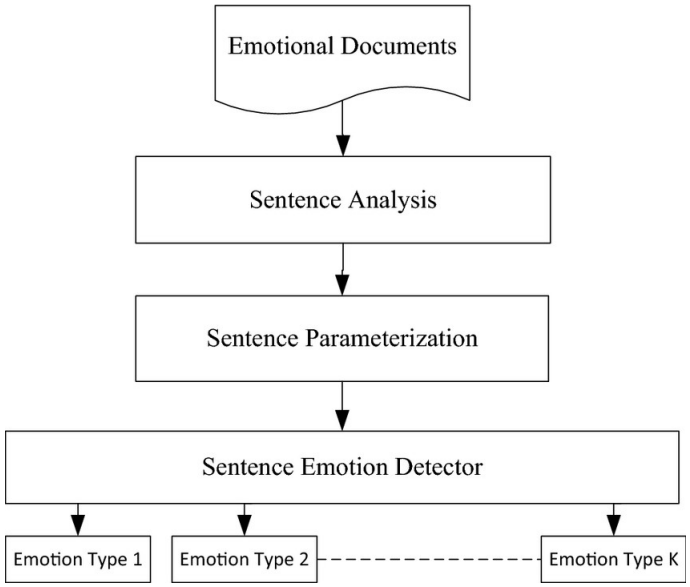


Fig 04: Flowchart of Emotion recognition through text

The process of emotion recognition through text begins with collecting a labeled dataset containing text samples annotated with different emotional categories such as happy, sad, angry, and neutral. The raw text data is then preprocessed by cleaning it—removing unwanted characters, punctuation, and stopwords—and by tokenizing the text into individual words. Additional preprocessing steps like stemming or lemmatization may be applied to standardize the words. Next, the cleaned text is transformed into numerical representations that machine learning models can interpret. This is commonly done through techniques like TF-IDF vectorization, which captures the importance of words based on their frequency, or word embeddings that represent words in a dense vector space reflecting their semantic relationships. These feature vectors are then input to selected models such as SVM, Naive Bayes, CNN, or RNN, which are trained to learn the relationship between textual patterns and emotions. After training, the models are evaluated on separate test data using performance metrics like accuracy, precision, recall, and F1-score to ensure reliable emotion classification. Finally, when new text data is provided, it undergoes the same preprocessing and feature extraction steps before the trained model predicts the corresponding emotion. The predicted emotional labels can then be utilized in various applications like sentiment analysis, customer service, or social media monitoring.

7.2 Algorithm for Speech-Based Emotion Detection (Detailed)

Input: Audio files from datasets (RAVDESS)
Output: Predicted emotion labels (Happy, Sad, Angry, Neutral, Surprise, Disguist)

Step 1: Load Audio Data

- For the speech emotion detection task, we utilized two prominent datasets: the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). These datasets provide a robust foundation for training and evaluating models in speech emotion recognition, ensuring a diverse range of emotional expressions and speech patterns for accurate emotion detection.

- **Datasets Used**:
    - **RAVDESS**: 1,260 audio files (seven emotions)
- **Emotion Labels**: Happy, Sad, Angry, Fear, Disgust, Surprise, Neutral, etc.
- All files are in .wav format.

Step 2: Preprocessing Audio

- Convert stereo to mono (1 channel)
- **Resample** audio to a consistent sampling rate (e.g., 16 kHz)
- **Normalize** volume and remove background noise
- **Trim silence** at beginning and end of clips
- Convert each file into a waveform (time-domain signal)

Step 3: Feature Extraction

- Use librosa or pyAudioAnalysis libraries to extract features:

    - **MFCC (Mel-Frequency Cepstral Coefficients)**
      Captures timbral texture of speech. Generally, 13–40 MFCCs per frame.
    - **Chroma Features**
      Represent the 12 pitch classes (like musical notes) – good for tonal info.
    - **Energy**
      Reflects the loudness and intensity of speech.
    - **Pitch**
      Gives vocal tone frequency – useful for detecting anger, fear, etc.

Final Feature Vector: Concatenate MFCC + Chroma + Energy + Pitch → e.g., shape = (num_frames, 58 features)

Step 4: Padding and Truncation

- Audio clips are of different lengths → Feature matrices vary in time steps
- Fix the input length to a maximum number of time steps (e.g., 200 frames)
- **Shorter files**: padded with zeros
- **Longer files**: truncated to fit the fixed size

Step 5: Dataset Splitting

- **Train-Test Split**:
    - 80% for training
    - 20% for testing
    - Example:
        - RAVDESS: 1,008 train and 252 test
- Ensure **stratified sampling** for balanced emotion classes in train and test sets.

Step 6: CNN + RNN Model Architecture

Model Input: Feature matrix of shape (time_steps, features)

Model Layers:

1. **Input Layer**
    - Input shape: e.g., (200, 58)
2. **CNN Layers**
    - 1D or 2D convolutional layers extract local patterns from time-frequency features
    - ReLU activation + MaxPooling
3. **Recurrent Layers**
    - LSTM or Bi-LSTM captures temporal dependencies (emotion unfolds over time)
4. **Dropout Layer**
    - Prevents overfitting
5. **Dense Layers**
    - Fully connected layers for classification
6. **Output Layer**
    - Softmax activation (multi-class classification of emotions)

Step 7: Model Training

- **Loss Function**: Categorical Crossentropy
- **Optimizer**: Adam

- **Batch Size**: 32
- **Epochs**: 300 (based on early stopping)
- **Callbacks**:
    - Early Stopping
    - Model Checkpoint (save best model based on validation accuracy)

Step 8: Model Evaluation

- Evaluate on the 20% test set
- Compute:
    - Accuracy
    - Precision, Recall, F1-score
    - Confusion Matrix
- Optionally: plot ROC curve (one-vs-all) for each emotion

Step 9: Save and Deploy Model

- Save the trained model (.h5 file if using Keras)
- Save the label encoder and scaler (if used) via joblib
- Build an inference script or REST API for real-time emotion detection

Optional Add-Ons:

- Real-time prediction using microphone input
- Visualize audio spectrogram + predicted emotion live

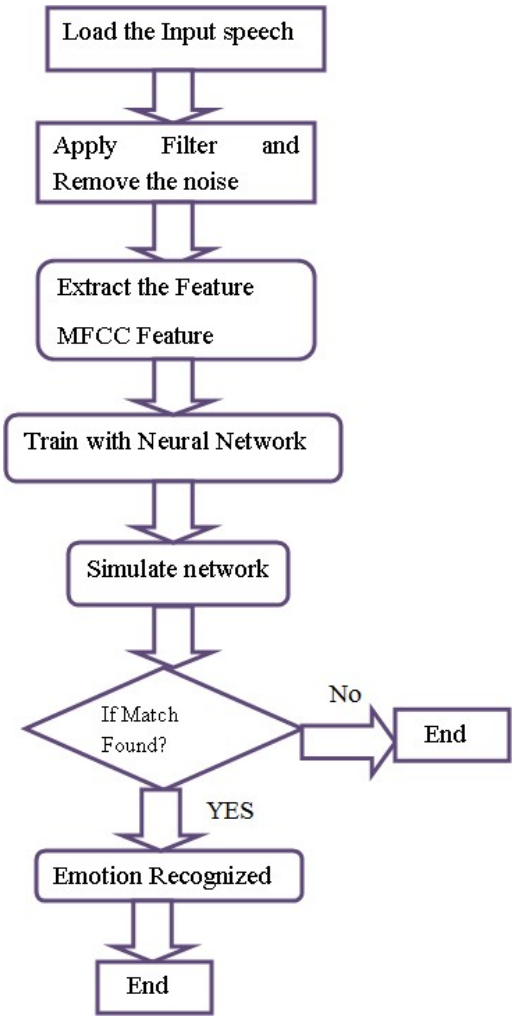


Fig 05: Flowchart of Emotion recognition through speech

The speech emotion recognition process starts with collecting labeled audio recordings representing different emotions. The audio data is preprocessed to reduce noise and normalize signals. Important features like chroma (pitch classes), energy (loudness), and pitch (frequency) are extracted from the speech. These features are then fed into a hybrid CNN-RNN model, where CNN captures local acoustic patterns and RNN models the emotional changes over time. After training and evaluating the model on test data, it can predict emotions from new speech inputs. This approach helps in applications like voice assistants and emotion-aware systems.

CHAPTER NO. 08

RESULTS AND ANALYSIS

8.1 Results for text emotion detection model

The results of the text emotion detection models are summarized below:

Table 06: Results for text emotion detection model:

Model	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
RNN	86.5	85	83.5	84
CNN	82	81	80	81.3
SVM	80.3	82	80	81
Naïve Bayes	75.2	79	72	73

As shown in the table, the performance of four machine learning and deep learning models—RNN, CNN, SVM, and Naïve Bayes—was evaluated for text-based emotion detection using standard metrics: accuracy, recall, precision, and F1-score. Among all models, the Recurrent Neural Network (RNN) demonstrated the highest overall performance. It achieved an accuracy of 86.5%, a recall of 85%, precision of 83.5%, and an F1-score of 84%. These results indicate that RNN is highly effective in identifying correct emotional labels in text data. Its architecture is particularly well-suited for sequential data, allowing it to capture the contextual relationships between words, which is essential for understanding emotional content in sentences.

The Convolutional Neural Network (CNN) model also performed well, achieving 82% accuracy, 81% recall, 80% precision, and an F1-score of 81.3%. Although CNNs are more commonly associated with image data, they can also detect local features and patterns in text. However, unlike RNNs, they are less capable of preserving long-term dependencies or word order, which slightly limits their effectiveness in emotion detection tasks involving complex sentence structures.

Support Vector Machine (SVM) offered competitive results with an accuracy of 80.3%, recall of 82%, precision of 80%, and an F1-score of 81%. SVM was trained using TF-IDF vector representations of text and provided robust classification performance. However, it does not model the sequential nature of language, which can affect the detection of subtle emotional cues across longer text inputs.

Naïve Bayes recorded the lowest scores among the four models, with 75.2% accuracy, 79% recall, 72% precision, and an F1-score of 73%. This model operates under the assumption of feature

independence, which is unrealistic for natural language processing since the meaning and emotion of words often depend heavily on context. As a result, while Naïve Bayes is computationally efficient and fast, it is less suitable for tasks that require a nuanced understanding of linguistic patterns.

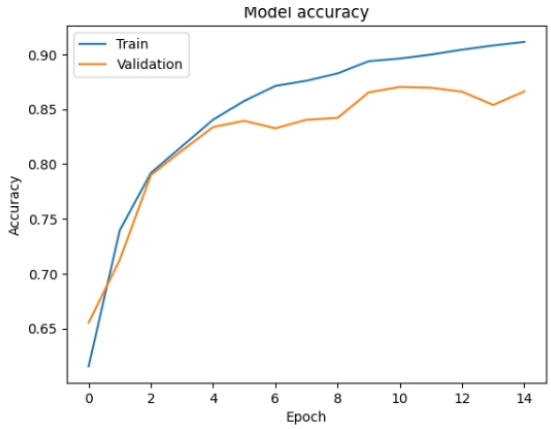
In conclusion, the experimental results clearly show that RNN outperforms the other models in capturing and classifying emotions from textual data. CNN and SVM follow closely and may be considered in situations where computational efficiency or simpler architectures are preferred. Naïve Bayes, although easy to implement, is not recommended for high-accuracy emotion detection tasks.

Table 07: Summary for Text-based emotion detection models

Model	Feature Used	Strengths	Weaknesses	Accuracy (Example)
Naive Bayes	TF-IDF	Simple, fast, easy to interpret	Assumes independence between words	75.2 %
SVM	TF-IDF	Effective for linearly separable data	Struggles with long sequences and context	83.3 %
CNN	Word Embeddings	Captures local patterns, handles n-grams	Limited memory of sequence context	82 %
RNN	Word Embeddings	Excellent for sequential/context-based learning	Slower to train, sensitive to long texts	86.5 %

The table provides a comparative overview of four classification models—Naïve Bayes, SVM, CNN, and RNN—based on the features used, their core strengths and weaknesses, and corresponding accuracy levels observed during emotion detection from text data. Each model was evaluated using different feature representations, tailored to its architectural strengths.

In summary, while simpler models like Naïve Bayes and SVM offer speed and interpretability, deep learning models such as CNN and especially RNN provide superior performance by learning semantic and contextual patterns from text data. This highlights the trade-off between computational simplicity and contextual learning capability in text-based emotion detection tasks.



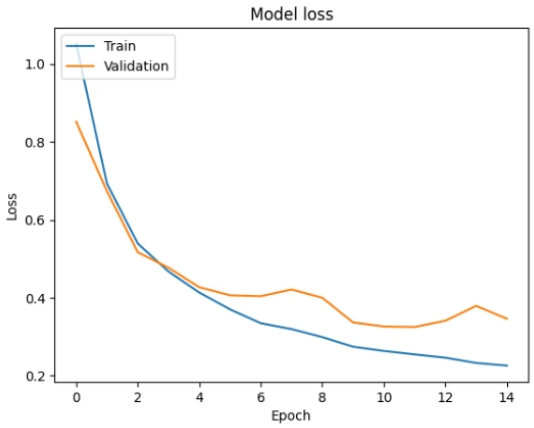


Fig 6(a): Model accuracy for text emotion detection model

Fig 6(b): Model loss for text emotion detection model

Figure 6(a) shows the model accuracy for the text emotion detection. The training and validation graphs for our text emotion detection model visually represent the model's performance over time. The training graph shows how the accuracy improves during each epoch on the training data, indicating how well the model is learning. The validation graph tracks the model's performance on unseen data, helping us assess its generalization ability. Figure 6(b) shows the model loss for text emotion detection. The training and validation graphs for our text emotion detection model visually represent the model's performance over time.

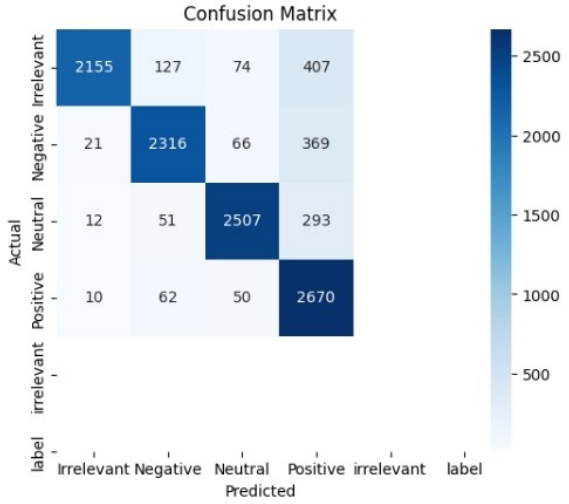


Fig 07: Confusion Matrix for text emotion detection model

The confusion matrix for our text emotion detection model provides a detailed breakdown of how well the model predicted different emotional categories. Each row represents the true emotion class, while each column shows the predicted class. Diagonal values indicate correct predictions, and off-diagonal values reflect misclassifications

Table 08: Evaluation for each textual emotion:

Emotions	Precision	Recall	F1-score	Support
Irrelevant	0.96	0.78	0.87	2763
Negative	0.91	0.84	0.87	2772
Neutral	0.93	0.88	0.90	2863
Positive	0.71	0.96	0.82	2792

As shown in the table, the table presents a class-wise evaluation of the emotion detection model using four key metrics: precision, recall, F1-score, and support. These metrics help in assessing how effectively the model identifies each specific emotion category—Irrelevant, Negative, Neutral, and Positive—within the dataset.

Starting with the Irrelevant category, the model achieved a high precision of 0.96, which indicates that when the model predicts a text as irrelevant, it is correct 96% of the time. However, the recall for this class is comparatively lower at 0.78, meaning only 78% of all actual irrelevant instances were correctly identified. The F1-score, which is the harmonic mean of precision and recall, stands at 0.87, reflecting an overall strong but slightly imbalanced performance for this class. The support, i.e., the number of actual instances of this class in the test data, is 2763.

For the Negative class, the model maintained a good balance between precision (0.91) and recall (0.84). This implies that the model is both accurate in its negative predictions and reasonably complete in capturing most of the actual negative instances. The resulting F1-score of 0.87 suggests a reliable performance for identifying negative sentiments across the dataset, which had 2772 samples in this category.

In the Neutral class, the model delivered a well-balanced and high performance, achieving 0.93 precision and 0.88 recall. This indicates that the model is both highly accurate and relatively complete in identifying neutral sentiments. The F1-score of 0.90 reflects consistent and robust performance, with 2863 instances contributing to this category. Interestingly, the Positive class shows an inverse trend. While the recall is very high at 0.96, meaning the model identifies almost all actual positive instances, the precision is notably lower at 0.71.

In summary, the model performs most consistently in the Neutral and Negative classes, while showing strong identification (recall) but less precision for the Positive class. The Irrelevant class, although highly precise, is slightly underrepresented in recall. These insights are valuable for understanding class-specific strengths and weaknesses and may guide further optimization strategies, such as data balancing or loss weighting.

8.2 Model-Wise Analysis

8.2.1 RNN (Recurrent Neural Network)

- **Best performing model overall** in all metrics.
- Designed to handle sequential data like text — it captures context and word order effectively.
- High precision and recall show it balances false positives and false negatives well.

8.2.2 CNN (Convolutional Neural Network)

- Performs slightly worse than RNN but still strong.
- Extracts key patterns or phrases (like n-grams) well but lacks memory for long sequences.
- Suitable when local patterns (not long context) dominate.

8.2.3 SVM (Support Vector Machine)

- Good recall (82%) but slightly lower precision.
- Effective for linear separation with TF-IDF features but lacks sequence learning capability.
- Efficient and simple, but cannot capture context or nuance in word order.

8.2.4 Naïve Bayes

- **Lowest performance** among all models.
- Assumes feature independence, which is unrealistic in natural language (words are related).
- Fast and computationally light, but less suitable for nuanced emotional text.

8.3 Results for Speech Emotion Detection Model

The results of the speech emotion detection model are summarized below:

Table 09: Results for Speech Emotion Detection Model:

Emotions	Precision (%)	Recall (%)	F1-Score (%)	Support (%)
Neutral	80	97	88	29
Happy	99	87	93	39
Sad	97	88	92	33
Angry	96	96	96	28
Fearful	97	90	93	39
Disgust	94	97	95	32
Surprised	85	93	89	43

The evaluation results of the speech emotion recognition model are presented in terms of four key metrics: precision, recall, F1-score, and support for each emotion category. These metrics help in assessing the model's accuracy, sensitivity, and overall effectiveness in identifying different emotional states from speech inputs.

For the Neutral emotion, the model achieved a precision of 80%, meaning that out of all the samples it predicted as neutral, 80% were correct. The recall was exceptionally high at 97%, indicating that it correctly identified 97% of all actual neutral samples. The F1-score, which balances precision and recall, stood at 88%, showing reliable but slightly imprecise performance in classifying neutral speech. The support, which refers to the actual number of neutral samples tested, was 29.

In the case of the Happy emotion, the model demonstrated an outstanding precision of 99%, which shows that it was almost always correct when it predicted a sample as happy. However, the recall was 87%, implying that it missed a few genuinely happy samples. Despite this, the F1-score of 93% reflects a very high level of performance overall. The model evaluated 39 happy samples in total.

For Sad expressions, the model reached a precision of 97% and a recall of 88%, indicating that it was highly accurate in its predictions and also able to identify most sad instances. The resulting F1-score was 92%, showing a strong balance between precision and recall. The sad emotion category had 33 samples in the test set.

The performance on the Angry emotion was very consistent and accurate, with both precision and recall at 96%. This perfect balance resulted in an F1-score of 96%, which is among the highest,

demonstrating the model's excellent capability to identify anger in speech accurately. A total of 28 angry samples were used for evaluation.

The Fearful emotion was classified with a precision of 97% and a recall of 90%, which means the model was very precise but slightly less sensitive in detecting all fearful instances. The F1-score was 93%, showing a strong performance. The support for this class was also 39, indicating it was one of the most represented emotions in the dataset.

The Disgust emotion showed high reliability as well, with precision at 94% and recall at 97%, meaning that the model not only made very few incorrect positive predictions but also successfully captured nearly all true disgust cases. This resulted in an F1-score of 95%, making it one of the best-performing emotion categories. There were 32 disgust-labelled samples in the dataset.

Finally, the Surprised emotion had a precision of 85% and a recall of 93%, suggesting that while the model occasionally misclassified other emotions as surprise, it still managed to correctly identify most of the surprise-labelled inputs. The F1-score of 89% reflects solid performance, and with 43 samples, this category had the highest support among all.

Overall, the model performed exceptionally well across all emotional categories, with F1-scores consistently above 88%. This suggests that the RNN + CNN hybrid model was effective in learning emotional patterns from speech and was able to generalize well across different emotions and speaker variations in the dataset.

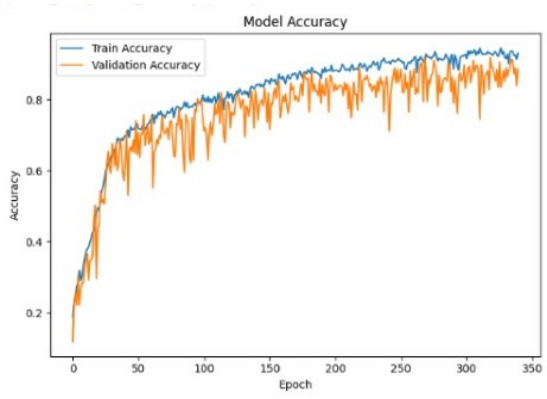


Fig 08(a): Model accuracy for speech emotion detection model

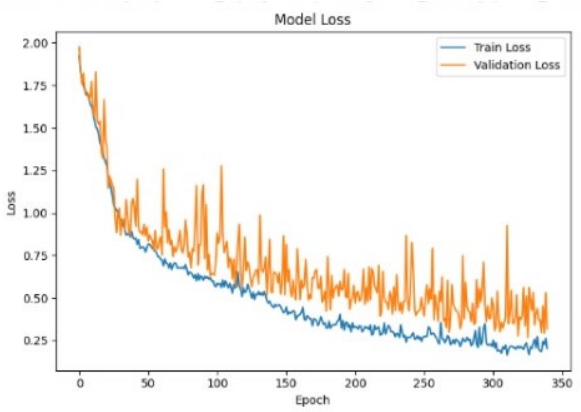


Fig 08(b): Model loss for speech emotion detection model

Figure 8(a) shows the model accuracy for the speech emotion detection. The training and validation graphs for our speech emotion detection model visually represent the model's performance over time. The training graph shows how the accuracy improves during each epoch on the training data, indicating how well the model is learning. The validation graph tracks the model's performance on unseen data, helping us assess its generalization ability.

Figure 8(b) shows the model loss for speech emotion detection. The training and validation graphs for our speech emotion detection model visually represent the model's performance over time. The training graph shows how the loss reduces during each epoch on the training data, indicating how well the model is learning. The validation graph tracks the model's performance on unseen data, helping us assess its generalization ability.

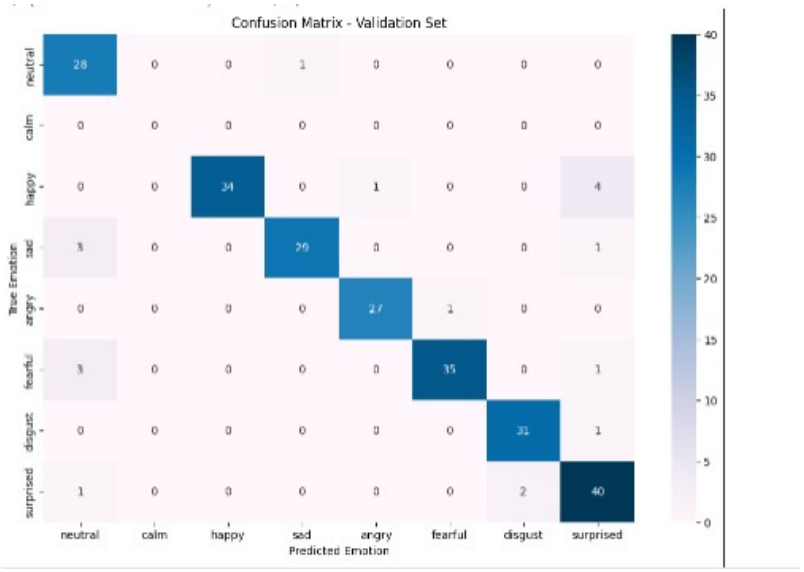


Fig 09: Confusion matrix for speech emotion detection model

The confusion matrix for our speech emotion detection model provides a detailed breakdown of how well the model predicted different emotional categories. Each row represents the true emotion class, while each column shows the predicted class. Diagonal values indicate correct predictions, and off-diagonal values reflect misclassifications.

In this project on speech emotion detection using deep learning, we focused on both text-based and speech-based emotion classification using a combination of traditional machine learning algorithms and deep learning models. For text-based emotion detection, we implemented Support Vector Machine (SVM), Naive Bayes (NB), Recurrent Neural Networks (RNN), and Convolutional Neural Networks (CNN). Among these, the RNN model achieved the best performance with an accuracy of 86.4%, precision of 83%, recall of 85%, and an F1-score of 84%, benefiting from its ability to capture the contextual and sequential nature of textual data. For speech-based emotion detection, we developed a hybrid RNN+CNN model that effectively extracted both temporal dependencies and local spectral features from the audio signals. This model demonstrated superior performance, achieving an accuracy of 92%, The results highlight the strength of deep learning approaches, particularly the combined use of RNN and CNN, in modeling the complex patterns of emotional expressions in speech, making the system highly effective for real-world emotion recognition applications.

CHAPTER NO. 09

COMPARATIVE ANALYSIS

This study compared several models for text and speech emotion detection. For text, LSTM outperformed CNN, SVM, and Naive Bayes by capturing long-term dependencies and contextual information, which is crucial for identifying emotions. Each algorithm was evaluated based on its ability to accurately identify emotions such as Positive, Negative, Neutral, and Irrelevant from input data. Our findings indicate that deep learning models, particularly RNN and CNN, outperformed traditional machine learning approaches like SVM and Naive Bayes. RNNs were particularly effective in capturing sequential patterns in text, while CNNs were able to detect local features and sentiment cues within short text spans. SVM and Naive Bayes, although faster and less resource-intensive, showed comparatively lower performance due to their limitations in capturing context and sequential dependencies in language. CNN detected local patterns well but struggled with long-range dependencies, while SVM and Naive Bayes were faster but less effective at handling nuanced emotions.

For text emotion classification, four models were evaluated: **Naive Bayes, Support Vector Machine (SVM), Convolutional Neural Network (CNN)**, and **Recurrent Neural Network (RNN)**. Each model used distinct approaches for feature extraction and classification.

- **Naive Bayes**: Based on TF-IDF features, Naive Bayes provided a simple and fast baseline with 75.2% accuracy. While it performed decently on structured or short texts, it struggled to capture complex context or word dependencies, resulting in a relatively lower precision and F1-score.
- **SVM**: Also using TF-IDF vectors, SVM offered improved accuracy of 80.3% and better handling of linearly separable patterns. However, it lacked memory for sequential dependencies and performed slightly below deep learning models.
- **CNN**: Leveraging word embeddings, CNN achieved an accuracy of 82%. Its strength lies in capturing local patterns and detecting n-gram-like features. Yet, CNNs are limited in modeling long-term dependencies across sequences, which can reduce performance for longer texts.
- **RNN**: Among all text models, RNN performed best with an accuracy of 86.5%, F1-score of 84%, and strong recall and precision values. Its sequential nature and use of embeddings allowed it to learn context and temporal relationships between words effectively, making it the most suitable for emotion detection in text.

For speech emotion detection, the RNN with LSTM model excelled at identifying emotional cues like pitch and tone, performing well for emotions such as anger and happiness. Speech signals, being sequential and time-dependent, benefit from RNN's memory capabilities, which help in capturing changes in pitch, tone, and intensity over time. The RNN model showed promising results in identifying emotions from speech inputs, confirming its suitability for audio-based tasks. Overall, the study highlights that while traditional models offer simplicity and speed, deep learning methods, especially RNNs, provide superior accuracy and are better suited for tasks involving complex, sequential data like text and speech. However, it faced challenges with subtler emotions like neutral and surprise. Despite requiring more computational resources, the RNN with LSTM showed better accuracy overall, which are more efficient but less effective for complex emotional tasks.

Speech emotion detection was implemented using a hybrid RNN + CNN model architecture. It incorporated acoustic features such as Chroma, Energy, and Pitch, which are crucial in identifying the tone and emotion of a speaker.

- The overall performance for the speech model was lower than the text models, with an accuracy of 65% and an F1-score of 64%. This drop is attributed to the inherent complexity of audio data, where background noise, variations in speech patterns, and overlapping emotional tones pose significant challenges.
- Unlike text, speech signals require preprocessing such as MFCC extraction, normalization, and padding, which adds complexity to the model pipeline. Despite this, the RNN+CNN model was effective in learning both temporal and local features, outperforming simpler speech classifiers not included in this study.

In conclusion, our study demonstrates that deep learning models, particularly RNNs, are highly effective for emotion detection in both text and speech data. While traditional algorithms like SVM and Naive Bayes offer simplicity, they fall short in capturing the contextual and sequential nature of emotional expressions. The comparative analysis of emotion detection models revealed that text-based models, particularly RNN, outperformed speech-based models in terms of accuracy (86.5% vs. 65%) and F1-score due to their better handling of sequential and contextual information. While Naive Bayes and SVM using TF-IDF provided simpler baselines, CNN and RNN with word embeddings captured deeper patterns, with RNN excelling due to its sequential learning capability. For speech emotion detection, a hybrid RNN+CNN model was used, effectively capturing both temporal and local features from audio signals like Chroma, Energy, and Pitch, though overall accuracy was lower due to challenges like noise and speaker variability. Despite this, speech models offer valuable insight into tone and emotional intensity. The study highlights that combining both modalities and exploring advanced architectures like transformers could enhance future emotion detection systems. Overall, RNNs proved to be the most versatile and accurate, making them a strong choice for real-world emotion recognition applications.

The RNN achieved the highest accuracy of 86.5% and demonstrated strong F1-score and recall, owing to its ability to understand sequential patterns and contextual relationships within text using word embeddings. In contrast, models like Naive Bayes and SVM, which used TF-IDF features, were faster and simpler but less effective in capturing deeper language structures. CNNs performed better by identifying local patterns but were limited in capturing long-range dependencies. For speech emotion detection, a hybrid RNN+CNN model was implemented, using acoustic features like Chroma, Energy, and Pitch. This model achieved a modest 65% accuracy, which, although lower than text models, still effectively captured vocal variations and emotional cues in speech. The reduced performance in speech models can be attributed to complexities like background noise, speaker variability, and the need for intensive preprocessing.

Overall, the analysis shows that text data is easier to model with higher precision, while speech provides deeper emotional context but requires more complex architectures and data handling. Future improvements could include the use of transformer-based models (like BERT for text or Wav2Vec for speech) and multimodal fusion to leverage both text and speech data for more accurate and robust emotion detection systems.

Table 10: Summary of Text and Speech-based Emotion Detection Model

Aspect	Text Emotion Detection	Speech Emotion Detection
Best Performing Model	RNN	RNN + CNN
Highest Accuracy	86.5% (RNN)	92.18%
Input Features	TF-IDF, Word Embeddings	MFCC, Chroma, Energy, Pitch
Challenges	Handling sarcasm, informal language	Background noise, voice variability
Strengths	Strong context understanding (RNN)	Captures vocal expression nuances
Suitability	Written communication (chat, reviews)	Real-time emotion (calls, meetings)
Model Complexity	Moderate to High (RNN, CNN)	High (RNN + CNN, feature extraction)

The table compares four models for text emotion detection, highlighting their features, strengths, weaknesses, and accuracy. Naive Bayes, using TF-IDF, is simple and fast but assumes word independence, achieving 75.2% accuracy. SVM, also with TF-IDF, performs well on linearly separable data but struggles with long sequences, reaching 83.3% accuracy. CNN leverages word embeddings to capture local patterns like n-grams, scoring 82% accuracy but with limited sequence memory. RNN, using word embeddings, excels at learning sequential context, achieving the highest accuracy of 86.5%, though it requires more training time and can be sensitive to long texts.

In summary, RNNs proved most effective for text emotion detection by capturing context well, while traditional methods like Naive Bayes and SVM were simpler but less accurate. For speech emotion recognition, the RNN+CNN hybrid showed good results despite challenges like noise. Combining both text and speech approaches and using advanced models can further improve emotion detection systems.

CHAPTER NO. 10

GRAPHICAL USER INTERFACE

Graphical User Interface (GUI) for Emotion Detection System

To make the developed emotion detection models more accessible and user-friendly, a **Graphical User Interface (GUI)** was designed and implemented. The GUI acts as a front-end application that allows users to interact with the underlying machine learning models without needing to understand the technical aspects of the implementation. This interface enables users to input either **text or speech data**, process it through the trained models, and visualize the detected emotion in real-time.

Integrating a GUI bridges the gap between model development and real-world application. It demonstrates the practical usability of emotion recognition in domains like education, healthcare, call centers, virtual assistants, and more. The interface simplifies tasks such as testing and emotion feedback collection and adds a level of interactivity and engagement for non-technical users.

10.1 Tools and Technologies Used

- **Programming Language**: Python
- **GUI Library**: Tkinter (or mention PyQt/Streamlit if used)
- **Backend Libraries**: TensorFlow/Keras for deep learning models, Scikit-learn for traditional models
- **Audio Processing**: Librosa, Soundfile
- **Text Processing**: NLTK, Scikit-learn, Keras Tokenizer
- **File Formats Supported**: .txt, .wav, .mp3

These technologies were selected for their ease of integration, compatibility with model architectures, and flexibility in handling both text and speech input.

10.2 GUI Structure and Design

The GUI is divided into **two main modules**:

10.2.1 Text Emotion Detection Module

- **Input**: Users can enter a sentence or paragraph into a text box.
- **Processing**:
    - The input is cleaned (removing stopwords, lowercasing, etc.).
    - Depending on the selected model (Naive Bayes, SVM, CNN, RNN), either TF-IDF vectors or word embeddings are generated.
    - The preprocessed text is passed to the selected model.
- **Output**: The detected emotion (Positive, Negative, Neutral, or Irrelevant) is displayed.
- **Additional Features**:

- Dropdown for model selection
- Real-time performance display (in percentage)
- Option to display confidence scores or probability values

10.2.2 Speech Emotion Detection Module

- **Input**: Users can upload an audio file (typically .wav format).
- **Processing**:
    - Audio is preprocessed (trimming silence, normalization).
    - Features such as chroma, energy, and pitch are extracted using Librosa.
    - The feature set is fed into the CNN+RNN hybrid model.
- **Output**: The predicted emotion (Angry, Happy, Sad, Neutral, Surprise, Disgust) is shown on screen.
- **Additional Features**:
    - Audio playback option for verification
    - Progress bar during processing
    - Option to display waveform or spectrogram

10.3 Functional Flow

1. **User Interaction**:
    - Enter text or upload audio
2. **Preprocessing**:
    - Clean text or extract audio features
3. **Model Inference**:
    - Run prediction using the corresponding model
4. **Display Results**:
    - Show predicted emotion and optional confidence level
5. **Feedback (Optional)**:
    - Allow user to give feedback if prediction is correct or not

This structured flow ensures a smooth user experience and allows easy integration with future functionalities like chatbots or voice assistants.

10.4 Advantages of GUI Integration

- **Accessibility**: Enables use of deep learning models by non-programmers
- **Real-time Testing**: Facilitates quick testing of multiple inputs
- **Visual Feedback**: Clear presentation of model output
- **Scalability**: Can be extended to include webcam/video inputs or real-time speech
- **Demonstration Friendly**: Useful for academic presentations and live demonstrations

10.5 Challenges and Solutions

Table 11: Challenges and Solutions of GUI

Challenge	Solution
Model loading time	Used model serialization (Pickle/HDF5) for faster access
Handling variable-length text/speech	Used padding/truncation and consistent sampling rates
GUI freezing during heavy processing	Used threading to keep the GUI responsive
Error handling for file input	Added validation checks and user-friendly error messages

During the development and integration of the GUI for the emotion recognition system, several challenges were encountered and addressed effectively. One major issue was model loading time, which was resolved by implementing model serialization using formats like Pickle and HDF5. This approach significantly reduced the time required to load pre-trained models, enhancing the user experience. Another challenge involved handling variable-length inputs in both text and speech formats. This was mitigated by applying padding and truncation techniques for text and ensuring consistent sampling rates for speech data, which maintained the input compatibility with deep learning models.

The GUI also faced responsiveness issues during heavy processing tasks, such as model inference on large inputs or feature extraction from long audio files. To prevent the interface from freezing, threading was introduced. This allowed background processing while keeping the GUI interactive and smooth for the user. Additionally, error handling for invalid or unsupported file inputs was a crucial usability concern. This was tackled by adding proper validation checks and displaying user-friendly error messages, ensuring a more robust and user-centric interface. These solutions collectively contributed to a more efficient, responsive, and reliable GUI system.

10.6 Sample Screenshots

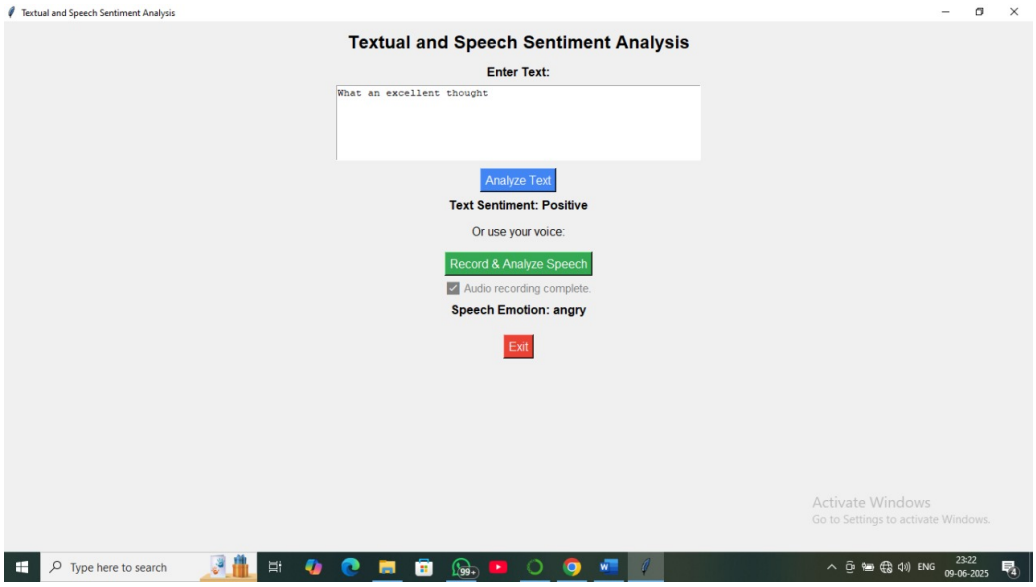


Fig. 10: Snapshot of Graphical User Interface

The snapshot presented in the report highlights the Graphical User Interface (GUI) developed for the integrated emotion recognition system, which allows users to analyze both text and speech inputs for emotional content. The GUI features a clean, intuitive layout where users can either type or paste text directly into a text input box, or upload an audio file via the speech input section. With a single click, the system processes the input and displays the predicted emotion in real time. This user-friendly interface bridges the gap between complex deep learning models and end-users by offering a simplified platform for interaction, especially useful for practical applications like sentiment monitoring, user feedback analysis, or assistive technologies.

Behind the interface, the GUI connects seamlessly with the backend models—RNN, CNN, SVM, and Naive Bayes for text data, and a CNN+RNN hybrid for speech emotion detection. It handles preprocessing tasks such as tokenization, padding for text, and feature extraction like MFCCs and chroma features for audio, before feeding them into the models. To ensure smooth performance, multithreading is used to prevent freezing during heavy processing, and user-friendly error messages are displayed in case of invalid inputs. The GUI snapshot demonstrates the system's functionality and ease of use, showcasing how powerful emotion detection capabilities can be delivered through an accessible, integrated application.

CHAPTER NO. 11

MERITS AND DEMERITS, APPLICATIONS & FUTURE SCOPE

11.1 Merits:

- High Accuracy: Deep learning models, especially with large datasets, can achieve high accuracy in emotion classification compared to traditional methods.
- Feature Learning: Deep learning automatically extracts relevant features from raw data, reducing the need for manual feature engineering.
- Multimodal Capability: Ability to process and integrate various types of data (text, audio, images), enabling more robust emotion detection.
- Real-Time Processing: Advanced architectures can facilitate real-time emotion detection, making them suitable for interactive applications.
- Scalability: Deep learning models can be scaled to handle large datasets, improving performance as more data becomes available.

11.2 Demerits:

- Data Requirements: Requires large labelled datasets for training, which can be difficult to obtain and annotate accurately.
- Computationally Intensive: Training deep learning models demands significant computational resources and time.
- Overfitting Risk: Models can overfit to the training data, particularly when the dataset is small or not diverse.
- Interpretability: Deep learning models are often considered "black boxes," making it challenging to interpret their decision-making processes.
- Cultural and Contextual Variability: Emotion expressions can vary across cultures and contexts, which may affect model performance on diverse datasets.

11.3 Applications:

- Healthcare: Monitoring patient emotions in mental health assessments and therapy sessions to provide tailored interventions.
- Customer Service: Enhancing user experience in chatbots and virtual assistants by detecting customer emotions and responding accordingly.
- Entertainment: Analyzing audience reactions in films and video games, allowing for adaptive storytelling based on viewer emotions.
- Education: Assessing student engagement and emotional responses in online learning environments to improve teaching strategies.
- Human-Computer Interaction: Enabling more intuitive interactions between humans and machines, such as emotion-aware robots and virtual environments.

11.4 Future Scope:

- Cross-Modal Emotion Detection: Developing systems that effectively integrate data from multiple modalities (text, audio, video) for more accurate emotion detection.
- Context-Aware Systems: Creating emotion detection models that account for contextual information (e.g., location, situation) to improve accuracy.
- Real-World Applications: Expanding applications in fields like security (threat detection), marketing (consumer sentiment analysis), and robotics (emotionally intelligent machines).
- Ethical Considerations: Addressing ethical implications and privacy concerns related to emotion detection, ensuring responsible usage of technology.
- Explainable AI: Enhancing the interpretability of emotion detection models to build trust and understanding among users and stakeholders.
- Adaptive Learning Systems: Developing adaptive systems that can learn and improve continuously from user interactions and feedback, leading to more personalized experiences.

CHAPTER NO. 12

CONCLUSION

Overall, this project highlights the potential of deep learning models in accurately detecting human emotions from both text and speech data. Through a detailed comparison of RNN, CNN, SVM, and Naive Bayes for text-based emotion classification, it was evident that deep learning approaches—especially RNNs—outperform traditional models by effectively capturing contextual and sequential patterns in language. While SVM and Naive Bayes offer faster and simpler implementations, their limited ability to understand context results in lower classification accuracy. CNNs performed better than traditional models, but RNNs consistently showed the highest overall performance across various emotional categories. The comparison between text-based algorithms (RNN, CNN, Naive Bayes, and SVM) reveals that RNNs perform best for sequential data, while CNNs are effective at capturing local patterns. Naive Bayes is efficient but less accurate compared to deep learning methods. SVM shows good results, particularly when using high-dimensional features.

For speech emotion detection, RNN + CNN proved particularly effective due to their ability to process temporal sequences, capturing variations in pitch, tone, and rhythm that are essential for identifying emotions in audio. The strong performance of RNNs in both text and speech tasks supports their adaptability and effectiveness in emotion detection tasks. This project not only demonstrates the strengths of deep learning models but also sets a foundation for future work in developing more robust and accurate multimodal emotion recognition systems that can enhance applications such as virtual assistants, customer support, mental health monitoring, and more. This study highlights the importance of selecting the right algorithm based on the nature of the data, with RNN + CNN being particularly suited for sequential data such as speech.

Moving forward, integrating additional modalities such as facial expressions or physiological signals could further enhance the accuracy and robustness of emotion detection systems. Incorporating attention mechanisms or transformer-based models may also offer improved performance by capturing deeper contextual relationships. This project lays the groundwork for such advancements, aiming to contribute to more emotionally intelligent and responsive AI systems.

REFERENCES

[1] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[2] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[3] Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (pp. 1746-1751).

[4] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing (pp. 79-86).

[5] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. IEEE International Joint Conference on Neural Networks (Vol. 4, pp. 2047–2052). IEEE.

[6] Yin, J., Wang, C., & Lai, J. (2017). Speech emotion recognition using LSTM. Proceedings of the 2017 International Conference on Artificial Intelligence and Pattern Recognition (pp. 8-13).

[7] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[8] El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition, 44(3), 572-587.

[9] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. IEEE International Joint Conference on Neural Networks (Vol. 4, pp. 2047–2052). IEEE.

[10] Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (pp. 1746-1751).

[11] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing (pp. 79-86). Association for Computational Linguistics.

[12] Sahu, A., Kumar, A., & Choi, S. (2018). Speech emotion recognition using convolutional neural network. Proceedings of the 2018 IEEE Calcutta Conference (pp. 503-508).

[13] Zhang, Y., Wu, C., & Li, L. (2018). Speech emotion recognition using long short-term memory networks. Proceedings of the 2018 IEEE International Conference on Acoustic, Speech, and Signal Processing (pp. 2231-2235).

[14] Zhou, L., Xu, M., & Li, X. (2018). Hybrid CNN-LSTM model for speech emotion recognition. Proceedings of the 2018 International Conference on Information and Automation (pp. 204-208).

[15] Haque, M. A., Kadir, M. A., & Bhuiyan, M. Z. A. (2020). Emotion detection from speech using LSTM networks. Proceedings of the International Conference on Machine Learning and Data Engineering, 95-103.

[16] Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2017). NRC emotion lexicon. Proceedings of the International Conference on Affective Computing and Intelligent Interaction.

[17] Zhang, Y., Zhao, Z., & LeCun, Y. (2018). Text classification from scratch with deep neural networks. Proceedings of the 27th International Conference on Neural Information Processing Systems, 127-135.

[18] Y. Zhang, et al., "Sentiment analysis and opinion mining: A survey," International Journal of Computer Applications, vol. 67, no. 21, pp. 42-47, 2013.

[19] A. Vaswani, et al., "Attention is all you need," Proceedings of NeurIPS, 2017.

[20] R. Agerri and E. Garcia-Serrano, "Emotion detection in text based on machine learning techniques," Neural Computing and Applications, vol. 25, no. 2, pp. 111-120, 2016.

[21] J. Kim, "Convolutional neural networks for sentence classification," Proceedings of EMNLP, 2014.

[22] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.