#### A

#### Mini Project Report

#### on

**“AN ENHANCED SPEECH EMOTION RECOGINITION USING MACHINE LEARNING ALOGRITHMS”**

Submitted in partial fulfillment for the award of the degree of

**Bachelor of Technology**

**in**

**Information Technology**

**by**

## A.RISHITHA 22R21A1204

## U.SADVIKA 22R21A1258

## V.KEERTHI 22R21A1261

## V.ANKITHA 22R21A1262

Under the guidance of

### Mr. B. Veera Sekhar Reddy

### Assistant Professor

****

**Department of Information Technology**

**June 2025**

****

**Department of Information Technology**

CERTIFICATE

This is to certify that the project entitled “**An Enhanced Speech Emotion Recognition using Machine Algorithms**” has been submitted by **A.RISHITHA (22R21A1204), U.SADVIKA (22R21A1258), V.KEERTHI (22R21A1261), V.ANKITHA (22R21A1262)** in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

Supervisor Head of the Department

Mr. B. Veera Sekhar Reddy Dr. N V Raja Sekhar Reddy

External Examiner

# DECLARATION

We hereby declare that the project entitled **“An Enhanced Speech Emotion Recognition using Machine Algorithms”** is the work done during the period from **January 2025 to June 2025** and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of technology in Information Technology from Jawaharlal Nehru Technology University, Hyderabad. The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**A. Rishitha 22R21A1204**

**U. Sadvika 22R21A1258**

**V. Keerthi 22R21A1261**

**V. Ankitha 22R21A1262**

# ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all. First of all we would like to express my deep gratitude towards our supervisor **Mr. B. Veera Sekhar Reddy, Assistant Professor, Department of IT** for his support in the completion of my dissertation. We wish to express our sincere thanks to **Dr. N V RAJA SEKHAR REDDY, HOD, Department of IT** and also to Principal **Dr. K. SRINIVAS RAO** for providing the facilities to complete the dissertation.

We would like to thank all our faculty, coordinators and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

**A. Rishitha 22R21A1204**

**U. Sadvika 22R21A1258**

**V. Keerthi 22R21A1261**

**V. Ankitha 22R21A1262**

# ABSTRACT

Speech Emotion Recognition (SER) is a rapidly growing field within affective computing that aims to enable machines to understand and respond to human emotions by analyzing vocal expressions. Human emotions are intricately conveyed through various vocal attributes, including tone, pitch, energy, speaking rate, rhythm, and pauses. Accurate recognition of these emotional cues can significantly enhance human-computer interaction, making systems more responsive and empathetic. Potential real-world applications include virtual personal assistants, healthcare diagnostics, and customer service automation. A robust SER system is developed leveraging classical machine learning algorithms: Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP). These models classify emotional states based on features extracted from raw audio signals. The system is trained and tested on the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset, which offers high-quality, professionally recorded emotional speech samples spanning multiple emotions. These include; pitch, Energy, Zero-Crossing Rate, Speech Duration. Feature extraction is performed directly on raw audio waveforms using simple digital signal processing techniques, making the system lightweight and suitable for devices with limited resources. The machine learning models are trained using these extracted features. However, the MLP model exhibits promising generalization capabilities and is particularly effective at distinguishing closely related emotions such as happiness, anger, and sadness. This suggests that while SVM provides a strong baseline, neural networks like MLP offer potential for improved adaptability. The developed system provides a foundation for enhancing emotionally intelligent technologies that can better understand and respond to human emotions, improving user experience across a wide array of interactive applications.

# CONTENTS

### S. No Contents Page No

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Certificate | II |
| Declaration | III |
| Acknowledgement | IV |
| Abstract | V |
| 1 |  | Introduction | 1-13 |
|  | 1.1 | Speech Recognition |  |
|  | 1.2 | Speech Emotion Recognition |  |
|  | 1.3 | Machine Learning  1.3.1 Classification  1.3.1.1 Support Vector Machine(SVM)  1.3.1.2 Multi Layer Perceptron(MLP) |  |
|  | 1.4 | Problem Statement |  |
|  | 1.5 | Objective |  |
|  | 1.6 | Summary |  |
| 2 |  | Literature Survey | 14-28 |
|  | 2.1 | Related Survey |  |
|  | 2.2 | Research Gaps |  |
|  | 2.3 | Summary |  |
| 3 |  | Methodology | 29-42 |
|  | 3.1 | Architecture |  |
|  | 3.2 | Sample Code |  |
| 4 |  | Results | 43-48 |
|  | 4.1 | DataSet Description |  |
|  | 4.2 | Performance metrics |  |
|  | 4.3 | Result Comparison |  |
|  | 4.4 | Analysis of Results |  |
| 5 |  | Conclusion and Future Scope | 49-50 |
|  | 5.1 | Conclusion |  |
|  | 5.2 | Future Scope |  |
| 6 |  | References | 41-52 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Name of the Figure** | **Page no** |
| 1.1 | SVMs Optimize Margin Between Support Vectors or Classes | 5 |
| 1.2 | Components of Multi-Layer Perceptron | 9 |
| 3.1 | Architecture Diagram of SER | 29 |
| 4.1 | Accuracy comparison with different methods | 45 |
| 4.2 | Precision comparison with different methods | 46 |
| 4.3 | Recall comparison with different methods | 46 |
| 4.4 | F1-Score comparison with different methods | 47 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Name of the Table** | **Page no** |
| 4.1 | Accuracy comparison with different methods | 45 |
| 4.2 | Precision comparison with different methods | 46 |
| 4.3 | Recall comparison with different methods | 47 |
| 4.4 | F1-Score comparison with different methods | 47 |

**CHAPTER I**

**INTRODUCTION**

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language. It bridges the gap between human communication and computer understanding by analysing language in the form of text or speech. In recent years, NLP has become an essential component of intelligent systems, powering applications such as voice assistants, sentiment analysis, machine translation, chatbots, and more. With the growth of deep learning and availability of large datasets, NLP systems have evolved from rule-based models to data-driven approaches capable of handling complex linguistic patterns. Speech Emotion Recognition (SER) is a specialized application of NLP and speech signal processing that involves identifying the emotional state of a speaker from audio signals. Emotions such as happiness, sadness, anger, fear, and neutrality are expressed through variations in tone, pitch, rhythm, and speech patterns.While traditional NLP often deals with textual data, SER extends this by analysing paralinguistic features—elements beyond just the words, such as:

* + - Prosody: Intonation, stress, and rhythm of speech.
    - Spectral features: Mel Frequency Cepstral Coefficients (MFCCs), Chroma, etc.
    - Voice quality features: Pitch, energy, and speaking rate.

Machine learning algorithms play a crucial role in SER by learning patterns from these features to classify emotions accurately. Popular models include Support Vector Machines (SVM), Multi Layer Perceptron (MLP).Understanding human emotions through speech is vital for developing emotionally intelligent systems. NLP-driven SER can enhance human-computer interaction in various domains:

* + - Customer service (e.g., detecting frustration in calls),
    - Healthcare (e.g., monitoring mental health),
    - Education (e.g., understanding student engagement),
    - Entertainment and robotics (e.g., adaptive virtual assistants).

### 1.1 Speech Recognition

Speech Recognition is a technology that allows computers and electronic devices to understand and convert spoken language into written text. It enables users to interact with machines using their voice, making communication more natural, faster, and hands-free. This technology plays a key role in human-computer interaction and is widely used in smartphones, home automation systems, and customer service platforms. The process begins with capturing the speaker’s voice through a microphone. The system then analyses the audio signal to detect patterns that match known words and phrases. Traditional speech recognition systems use statistical models and rule-based methods to perform this task.

Key steps in a basic speech recognition system include:

* **Audio Capture :** The user speaks into a microphone, and the system records the speech as an analog signal. This is then converted to a digital format for processing.
* **Preprocessing** **:** The recorded audio is cleaned by removing background noise and normalizing volume levels. This step helps improve recognition accuracy.
* **Feature Extraction:** Identifying important characteristics of the speech signal (such as pitch, energy, or frequency-based features) that help distinguish different sounds.
* **Language Modelling:** Using grammar rules and word probabilities to predict likely word sequences.
* **Decoding:** Combining sound and language information to generate the final recognized text.

**Applications** of speech recognition include:

* Voice assistants (e.g., voice commands on phones)
* Speech-to-text tools (e.g., for dictation or note-taking)
* Automated phone systems (e.g., customer service menus)
* Voice-controlled smart devices

Even without advanced techniques, speech recognition systems can be effective in controlled environments or with limited vocabularies, making them useful for a wide range of practical tasks.

#### 1.2 Speech Emotion Recognition

Speech Emotion Recognition (SER) is an area of affective computing that focuses on enabling machines to identify and interpret human emotions based on vocal expressions. Emotions play a crucial role in human communication, often conveying more meaning than the actual words spoken. SER systems aim to bridge the gap between human emotional expression and machine understanding by analyzing how speech is delivered—through variations in tone, pitch, loudness, speed, and rhythm—rather than focusing on the content of speech. This allows the system to detect emotional states such as happiness, anger, sadness, fear, or neutrality. Unlike speech recognition systems that transcribe spoken words into text, SER prioritizes the emotional cues present in the acoustic signal. The technology typically involves preprocessing the audio, extracting relevant features, and classifying the emotion using machine learning models. With growing demand for emotionally intelligent technologies, SER has gained importance in applications such as mental health monitoring, intelligent tutoring systems, emotion-aware virtual assistants, smart homes, and automated customer service. Its ability to enhance user experience and create more natural human-computer interaction makes SER a vital component of modern voice-based systems.

Key Points:

* SER detects emotions from vocal signals rather than transcribing spoken words.
* Often uses traditional machine learning methods like Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP).
* Datasets like RAVDESS provide labeled emotional speech samples for training and evaluation.

Applicable in areas such as:

* Healthcare for emotional monitoring
* Customer service to detect frustration or satisfaction

#### 1.3 Machine Learning

Machine learning plays a central role in Speech Emotion Recognition (SER) by enabling systems to learn patterns from vocal data and classify emotional states automatically. After extracting relevant audio features—such as pitch, energy, zero-crossing rate, and speech duration—machine learning algorithms are trained to associate these patterns with specific emotions like anger, happiness, sadness, or fear. Traditional classifiers such as **Support Vector Machines (SVM) and Multi-Layer Perceptron (MLP)** are commonly used in SER tasks. These models are trained on labelled datasets, where each speech sample is tagged with the correct emotion. Once trained, the model can predict the emotional category of new, unseen audio samples with reasonable accuracy. Machine learning offers the advantage of adaptability—models can be retrained with new data to improve performance across different speakers, languages, or recording conditions. This makes machine learning a powerful tool for building flexible and efficient emotion-aware systems in real-world applications.

#### 1.3.1Classification

#### In Speech Emotion Recognition (SER), classification refers to the process of assigning a specific emotional label (such as happy, angry, sad, fearful, etc.) to a segment of speech based on the extracted audio features. Once the vocal signal has been preprocessed and relevant features have been extracted, the system uses classification algorithms to identify the most likely emotional category represented in the speech.

#### Support Vector Machine (SVM):

#### A powerful supervised learning algorithm that works well with small to medium-sized datasets. It finds the optimal boundary (hyperplane) that separates different emotion classes in the feature space. SVM is popular in SER due to its accuracy and ability to handle non-linear data using kernel functions.

#### Multi-Layer Perceptron (MLP):

#### A type of artificial neural network consisting of multiple layers of interconnected neurons. MLP is effective in capturing complex relationships between features and is used to classify emotions based on patterns in pitch, energy, duration, and other speech characteristics.

**1.3.1.1 Support Vector Machine (SVM)**

A support vector machine (SVM) is defined as a machine learning algorithm that uses supervised learning models to solve complex classification, regression, and outlier detection problems by performing optimal data transformations that determine boundaries between data points based on predefined classes, labels, or outputs. Support Vector Machine is a supervised machine learning algorithm widely used for classification and regression tasks. It is particularly effective for high-dimensional spaces and is well-suited for problems where the number of features is large compared to the number of samples. In the context of Speech Emotion Recognition (SER), SVM is used to classify different emotional states based on features extracted from audio signals such as pitch, energy, zero-crossing rate, and speech duration.

**Support Vector Machine (SVM) Terminology**

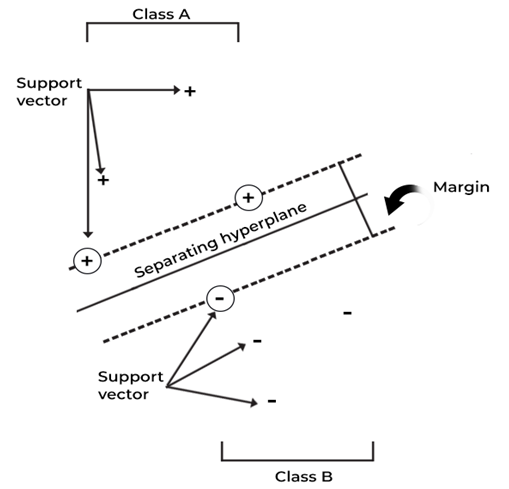
* Hyperplane: A decision boundary separating different classes in feature space, represented by the equation wx + b = 0 in linear classification.
* Support Vectors: The closest data points to the hyperplane, crucial for determining the hyperplane and margin in SVM.
* Margin: The distance between the hyperplane and the support vectors. SVM aims to maximize this margin for better classification performance.
* Kernel: A function that maps data to a higher-dimensional space, enabling SVM to handle non-linearly separable data.
* Hard Margin: A maximum-margin hyperplane that perfectly separates the data without misclassifications.
* Soft Margin: Allows some misclassifications by introducing slack variables, balancing margin maximization and misclassification penalties when data is not perfectly separable.
* C: A regularization term balancing margin maximization and misclassification penalties. A higher C value enforces a stricter penalty for misclassifications.
* Hinge Loss: A loss function penalizing misclassified points or margin violations, combined with regularization in SVM.
* Dual Problem: Involves solving for Lagrange multipliers associated with support vectors, facilitating the kernel trick and efficient computation.

Figure 1.1: SVMs Optimize Margin Between Support Vectors or Classes.

As seen in the above figure, the margin refers to the maximum width of the slice that runs parallel to the hyperplane without any internal support vectors. Such hyperplanes are easier to define for linearly separable problems; however, for real-life problems or scenarios, the SVM algorithm tries to maximize the margin between the support vectors, thereby giving rise to incorrect classifications for smaller sections of data points.

SVMs are potentially designed for binary classification problems. However, with the rise in computationally intensive multiclass problems, several binary classifiers are constructed and combined to formulate SVMs that can implement such multiclass classifications through binary means. In the mathematical context, an SVM refers to a set of[ML algorithms](https://www.spiceworks.com/tech/artificial-intelligence/articles/top-ml-algorithms/) that use kernel methods to transform data features by employing kernel functions. Kernel functions rely on the process of mapping complex datasets to higher dimensions in a manner that makes data point separation easier. The function simplifies the data boundaries for non-linear problems by adding higher dimensions to map complex data points. While introducing additional dimensions, the data is not entirely transformed as it can act as a computationally taxing process. This technique is usually referred to as the kernel trick, wherein data transformation into higher dimensions is achieved efficiently and inexpensively. The idea behind the SVM algorithm was first captured in 1963 by Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Since then, SVMs have gained enough popularity as they have continued to have wide-scale implications across several areas; text categorization, facial recognition, autonomous cars, robotic systems.

**Building a SVM classifier**

1. Split your data

As with other machine learning models, start by splitting your data into a training set and testing set. As an aside, this assumes that you’ve already conducted an [exploratory data analysis](https://www.ibm.com/topics/exploratory-data-analysis) on your data. While this is technically not necessary to build a SVM classifier, it is good practice before using any machine learning model as this will give you an understanding of any missing data or outliers.

1. Generate and evaluate the model

Import an SVM module from the library of your choosing, like [scikit-learn](https://scikit-learn.org/stable/modules/svm.html). Train your training samples on the classifier and predict the response. You can evaluate performance by comparing accuracy of the test set to the predicted values. You may want to use other evaluation metrics, like f1-score, precision, or recall.

1. Hyperparameter tuning

Hyperparameters can be tuned to improve the performance of an SVM model. Optimal hyperparameters can be found using grid search and cross-validation methods, which will iterate through different kernel, regularization (C), and gamma values to find the best combination.

**Feature Extraction in SER**

Before classification, it is essential to extract meaningful features from speech signals. In SER, features are typically divided into:

* Prosodic Features: Pitch, energy, duration.
* Spectral Features: Formants, spectral centroid (though often avoided in lightweight systems).
* Time-domain Features: Zero-crossing rate, short-time energy.

**Training and Testing**

The SVM model is trained on labeled speech data, such as the RAVDESS dataset, which includes recordings of actors expressing different emotions. During training, the model learns to associate patterns in the features with specific emotional states.

In multi-class classification scenarios, where more than two emotions are involved, SVM uses strategies such as:

One-vs-One (Ovs O): Constructs binary classifiers between every pair of classes.

One-vs-Rest (Ovs R): Constructs one binary classifier per class, distinguishing one class from all others.

These strategies enable SVM to handle multiple emotion classes effectively.

**Advantages of SVM**

* High Accuracy: SVM often outperforms other traditional classifiers, especially on small to medium-sized datasets.
* Robustness: SVM is less prone to overfitting, particularly when using the right kernel and regularization.
* Efficiency: Suitable for real-time applications due to fast prediction time after training.
* Scalability: Effective in high-dimensional feature spaces.

**Evaluation Metrics**

To evaluate the performance of an SVM classifier in SER, common metrics include:

* Accuracy: Percentage of correctly classified instances.
* Precision and Recall: Especially important for imbalanced datasets.
* F1-Score: Harmonic mean of precision and recall.
* Confusion Matrix: Visualizes true vs. predicted labels across emotion classes.
* Cross-validation techniques, such as k-fold cross-validation, are used to ensure the generalizability of the model.

**Use Cases**

SVM has been successfully applied in various systems:

Healthcare: Detecting emotional distress in patient voice data.

Customer Service: Identifying dissatisfaction or frustration in customer calls.

Smart Assistants: Enhancing user experience by responding to emotional cues.

Education Technology: Monitoring student emotions in e-learning platforms.

Support Vector Machine remains one of the most reliable and well-understood classification techniques in the field of machine learning. Its solid mathematical foundation, flexibility through kernel functions, and ability to handle both linear and non-linear data make it a preferred choice for a wide range of classification tasks. Despite some limitations related to scalability and interpretability, SVM continues to be widely adopted in academia and industry, proving its lasting relevance in the machine learning landscape.

**1.3.1.2 Multi Layer Perceptron (MLP)**

A Multilayer Perceptron (MLP) is a class of feedforward artificial neural networks (ANNs). It is among the earliest neural network architectures developed and is still widely used in machine learning. The primary goal of an MLP is to model complex, non-linear relationships between input and output variables. MLPs are composed of layers of interconnected nodes or neurons, where each node performs a simple computation. These networks can learn from data through a process called training, and they are particularly useful for tasks involving pattern recognition, classification, and regression. MLPs are supervised learning models, meaning they learn from labelled datasets.

Neural networks are at the core of modern machine learning and artificial intelligence. Among the many types, multilayer perceptron’s (MLPs) serve as a foundational building block for deep learning systems. This tutorial introduces the concept of artificial neural networks, explores how MLPs work, and walks through key components like backpropagation and stochastic gradient descent. An [artificial neural network](https://www.datacamp.com/tutorial/introduction-to-deep-neural-networks) (ANN) is a machine learning model inspired by the structure and function of the human brain's interconnected network of neurons. It consists of interconnected nodes called artificial neurons, organized into layers. Information flows through the network, with each neuron processing input signals and producing an output signal that influences other neurons in the network.

A multi-layer perceptron (MLP) is a type of artificial neural network consisting of multiple layers of neurons. The neurons in the MLP typically use nonlinear activation functions, allowing the network to learn complex patterns in data. MLPs are significant in machine learning because they can learn nonlinear relationships in data, making them powerful models for tasks such as classification, regression, and pattern recognition.

An MLP typically includes the following components:

* **Input layer:**Receives input data and passes it on to the hidden layers. The number of neurons in the input layer is equal to the number of input features.
* **Hidden layers:** Consist of one or more layers of neurons that perform computations and transform the input data. The number of hidden layers and neurons within each layer can be adjusted to optimize the network’s performance.
* **Activation function:** Applies a non-linear transformation to the output of each neuron in the hidden layers. Common activation functions include sigmoid, hyperbolic tangent (tanh).
* **Output layer:** Produces the final output of the network, such as a classification label or a regression target. The number of neurons in the output layer depends on the specific task, such as the number of classes in a classification problem.
* **Weights and biases:** Adjustable parameters that determine the strength of the connection between neurons in adjacent layers and the bias of each neuron. These parameters are learned during the training process to minimize the difference between the network’s predictions and the actual target values.
* **Loss function:** Measures the discrepancy between the network’s predictions and the actual target values. Common loss functions for MLPs include mean squared error for regression tasks and cross-entropy for classification tasks.

MLPs are trained using an optimization algorithm, such as gradient descent, to iteratively adjust the weights and biases based on the gradient of the loss function. This process continues until the network converges to an optimal set of parameters that minimize the loss function. The term “multi-layer perceptron” is often used interchangeably with “deep neural network,” although some sources may consider MLPs as a specific type of deep neural network. The terminology can be confusing, but in general, an MLP refers to a specific architecture of a deep neural network, characterized by its fully connected layers and use of backpropagation for training.

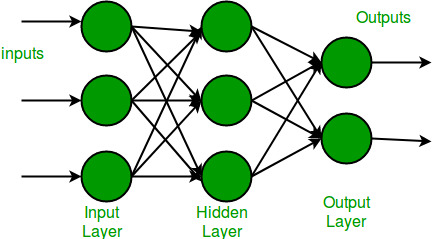
 How an MLP Works:

Figure 1.2 : Components of Multi-Layer Perceptron

**1. Forward Propagation:**

The input data flows through the network, with each neuron performing a weighted sum of its inputs, applying an activation function, and passing the result to the next layer.

**2. Backpropagation:**

The network compares its output with the desired output (from the training data), calculates an error, and then uses the backpropagation algorithm to adjust the weights of the connections based on the error.

**3. Iteration:**

This process of forward propagation and backpropagation is repeated iteratively over the training data until the network reaches a desired level of accuracy.

**Forward and Backward Propagation**

**Forward Propagation**

In forward propagation, data moves from the input layer to the output layer. Each neuron in each layer receives input, applies a weighted sum, adds a bias, and passes the result through an activation function. This process continues layer by layer until the output layer is reached. The final output is the prediction of the network.

**Loss Function**

The loss function measures the difference between the predicted output and the actual target values. It guides the network on how well it is performing and is crucial for training. Common loss functions include:

* **Mean Squared Error (MSE)**: Used for regression tasks.
* **Cross-Entropy Loss**: Used for classification tasks.

**Backpropagation**

Backpropagation is the core mechanism by which the network learns. It involves computing the gradient of the loss function with respect to each weight in the network. This gradient tells us how much the loss would change if the weight were adjusted slightly. The gradients are computed layer by layer, starting from the output layer and moving backward. These gradients are then used to update the weights.

**Training Process**

**Initialization:**  
At the start of training, the network’s weights and biases are set to small random values. This helps the model begin learning. To keep training stable, methods like Xavier or He initialization are often used to ensure the signals pass properly through the layers.

**Forward Propagation:**

Input data passes through each layer of the network. Neurons compute outputs using the input values, weights, biases, and an activation function. These outputs are sent forward through the network to generate a prediction.

**Loss Computation:**

The prediction is compared with the actual target using a loss function. This function calculates how wrong the prediction is, giving a value that guides the network in learning.

**Backward Propagation:**

The network works backward to figure out which weights contributed most to the error. It calculates gradients that show how to change the weights to reduce the loss.

**Gradient Descent and Optimization:**

Using the gradients, the network updates its weights and biases to improve predictions. This step is repeated many times, gradually reducing the error.

**Epochs and Iterations:**

The training data is used multiple times, called epochs. Within each epoch, data is split into smaller groups (batches), and the model updates its parameters after each batch.

**Monitoring and Early Stopping:**

The model’s progress is tracked during training. If it performs worse on validation data while improving on training data, training can be stopped early to avoid overfitting.

**Evaluation and Fine-Tuning:**

After training, the model is tested on new data to check its accuracy. Based on results, small adjustments can be made to improve its performance.

MLPs are foundational elements in machine learning. While newer models like convolutional and recurrent neural networks have gained popularity, MLPs still serve as essential tools for many standard tasks. They offer a balance of simplicity and power, making them suitable for a wide range of applications. Understanding MLPs is critical for anyone seeking to explore deeper into the field of neural networks and artificial intelligence.

* 1. **Problem Statement**

Often in the interest to increase the acceptability of speech technology for human users, the speech signal communicates verbal information between speakers as well as non-verbal information about a speaker’s emotions, personalities, attitudes, feelings, levels of stress, and current mental states. Words are not enough to correctly understand the mood and intention of a speaker, and thus the introduction of human social skills to human-machine communication is of paramount importance. This can be achieved by researching and creating methods of speech modelling and analysis that embrace the signal, verbal, and emotional aspects of communication.

* 1. **Objective**

The kind of machine learning application that we emphasize in this project is Speech Emotion Recognition (SER), where the goal is to identify human emotions from audio signals. This involves extracting meaningful features from speech, selecting suitable machine learning algorithms, and training models to classify emotions such as happiness, sadness, anger, and neutrality. The system aims to enhance human-computer interaction by enabling machines to better understand and respond to emotional cues in real time. SER can be approached in the following three ways:

**A. Audio-based classification,** using features like pitch, energy, and zero-crossing rate to detect emotional patterns.

**B. Text-based classification,** where speech is first converted to text and then analyzed using Natural Language Processing (NLP) techniques to detect emotion.

**C. Hybrid approach,** combining both audio and text-based features to improve the accuracy and robustness of emotion classification.

**1.6 Summary**

Speech Emotion Recognition (SER) is a specialized area within Natural Language Processing (NLP) and speech signal processing that focuses on identifying human emotions from speech. While NLP enables machines to understand and process human language, SER goes further by interpreting the emotional tone behind the spoken words. Emotions such as happiness, sadness, anger, and neutrality are communicated through various vocal characteristics like pitch, tone, rhythm, and loudness rather than the actual content of speech. Speech recognition technology plays a foundational role in this process by capturing spoken language and converting it into a format that machines can analyse. The steps typically involve recording the audio, removing background noise, extracting significant vocal features, applying models of language, and decoding the result. Unlike traditional speech recognition, which focuses on transcribing words, SER prioritizes the delivery style of speech to detect emotional states. These features help differentiate between various emotions based on how the voice is modulated. Machine learning algorithms are then employed to analyze these features and classify the emotional state. Common algorithms include Support Vector Machines (SVM) and Multi-Layer perceptron’s (MLP). SVM is effective for both linear and non-linear classification problems and works well with smaller datasets by finding an optimal boundary between classes. MLP, on the other hand, is a type of neural network that can learn complex patterns through layers of interconnected neurons, using techniques like forward propagation and backpropagation to minimize prediction errors .SER systems are trained usinglabelled datasets where each sample corresponds to a specific emotion. Models learn from these datasets to identify emotional patterns in new audio inputs. In multi-class classification tasks, strategies like One-vs-One or One-vs-Rest are used to differentiate among several emotions. The performance of SER systems is assessed through metrics such as accuracy, precision, recall, F1-score, and confusion matrices, and cross-validation is used to ensure reliability. The practical applications of SER are vast. In healthcare, customer service, and education The main goal of this project is to develop a machine learning-based SER system that accurately classifies emotions from speech. This can be achieved through audio-based analysis, text-based analysis after speech transcription, or a hybrid of both methods. By enabling machines to understand and respond to human emotions, SER aims to improve human-computer interaction and make technology more emotionally intelligent.

**CHAPTER II**

**LITERATURE SURVEY**

Speech emotion recognition is a very useful and important topic in today's world. A machine detecting the emotion of a human speech can be proved useful in various industries. A very basic usage of speech recognition is in the health sector where it can be used in detecting depression, anxiety, stress etc. in a patient. It can also be used in industries like the crime sector where emotions can be recognized from the speech to distinguish between victims and criminals. Emotions can be of various types like happy, sad, angry, disguised etc. depending on the feeling and frame of mind of the person. In our study, we have used various datasets with different emotions. We have also combined four datasets to one dataset and then applied the model so that the efficiency of the model can be improved and there can be a variety in the data points. This has also resulted in eliminating the overfitting condition in our model. Speech Emotion Recognition (SER) is an area of affective computing that focuses on enabling machines to identify and interpret human emotions based on vocal expressions. Emotions play a crucial role in human communication, often conveying more meaning than the actual words spoken. SER systems aim to bridge the gap between human emotional expression and machine understanding by analyzing how speech is delivered—through variations in tone, pitch, loudness, speed, and rhythm—rather than focusing on the content of speech. This allows the system to detect emotional states such as happiness, anger, sadness, fear, or neutrality. Unlike speech recognition systems that transcribe spoken words into text, SER prioritizes the emotional cues present in the acoustic signal. The technology typically involves preprocessing the audio, extracting relevant features, and classifying the emotion using machine learning models. With growing demand for emotionally intelligent technologies, SER has gained importance in applications such as mental health monitoring, intelligent tutoring systems, emotion-aware virtual assistants, smart homes, and automated customer service. Its ability to enhance user experience and create more natural human-computer interaction makes SER a vital component of modern voice-based systems. Speech Emotion Recognition (SER) is an interdisciplinary field that involves the use of machine learning techniques to identify human emotions based on vocal characteristics. Emotions such as happiness, sadness, anger, fear, and neutrality can be inferred from audio features like pitch, tone, energy, and rhythm. With the increasing use of voice assistants, call center monitoring systems, and emotion-aware healthcare tools, SER has become an important area of research. The success of such systems largely depends on effective feature extraction from speech signals and the use of robust classifiers to distinguish between emotional states. Machine learning offers powerful models that can learn from labeled data and generalize to new, unseen speech samples. In this project, two widely used machine learning algorithm — **Support Vector Machine (SVM)** are employed for emotion classification. SVM is known for its ability to handle high-dimensional data and perform well in complex classification problems by using kernel functions to map data into higher-dimensional spaces.

Speech emotion recognition (SER) is a crucial task in human-computer interaction, aiming to enable machines to detect and interpret human emotions from speech signals. By combining machine learning (ML) techniques and natural language processing (NLP) tools, researchers can extract both acoustic and linguistic features from audio data to effectively classify emotional states such as happiness, sadness, anger, fear, and calmness. A widely used dataset in this domain is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), which provides high-quality, labeled recordings of speech and song by professional actors expressing a range of emotions. To build an effective SER system using this dataset, the speech signals are preprocessed and features such as Mel-frequency cepstral coefficients (MFCCs), pitch, energy, and spectral contrast are extracted. These features are then fed into a machine learning classifier—often a Support Vector Machine (SVM)—which excels in high-dimensional space and is known for its robustness in classification tasks. SVMs work by finding the optimal hyperplane that separates data points of different emotion classes with the maximum margin, which makes them especially suitable for multi-class classification scenarios in SER. When combined with NLP techniques that may analyze spoken words, sentence structure, or prosody, the system can further enhance its ability to understand nuanced emotional expressions. Overall, the integration of ML, NLP, and SVMs with the rich emotional data in RAVDESS forms a powerful pipeline for building reliable and intelligent emotion-aware speech systems used in applications like virtual assistants, sentiment analysis, mental health monitoring, and affective computing.

### Related Survey

**Md. Rayhan Ahmed’s** research in the field of Speech Emotion Recognition (SER) represents a sophisticated deep learning-based approach to understanding and classifying emotions from raw audio inputs. In this work, the author developed four distinct neural network architectures, each leveraging advanced deep learning strategies to capture and interpret the complex acoustic patterns found in human speech. Central to the approach is the use of Local Feature Aggregation Blocks (LFABs), which serve as the foundational elements in Model-A. This model employs seven LFABs to extract robust local features from speech signals, followed by Fully Connected Networks (FCNs) and a SoftMax layer for final classification. These LFABs are designed to enhance the local discriminative capability of the model by aggregating neighbourhood features within speech, which are essential for identifying variations in emotional tone. The author also focuses heavily on the preprocessing and feature extraction phase. From each speech sample, five categories of acoustic features are manually extracted—Mel Frequency Cepstral Coefficients (MFCC) which further learns hidden representations and patterns in the audio. This CNN block acts as a second-level feature extractor that captures short-term dependencies and prepares the data for more abstract modeling by recurrent units in subsequent layers. The combination of handcrafted features and learned features ensures that the model benefits from both domain knowledge and data-driven learning. After applying data augmentation (DA) to enhance the diversity of training data, all four models were rigorously trained and evaluated. The ensemble Model-D outperformed all others, achieving a state-of-the-art weighted average accuracy of 99.46% on the TESS dataset. [1]

**Dr. Nilesh Shelke’s** work on Speech Emotion Recognition (SER) represents a forward-thinking initiative that utilizes multiple publicly available datasets—namely RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), TESS (Toronto Emotional Speech Set), and SAVEE (Surrey Audio-Visual Expressed Emotion)—to build a robust framework for detecting emotions from human speech. His work emphasizes the modernization of conventional approaches to emotion detection systems (EDS), aiming to bridge the gap between evolving technological capabilities and real-world applications. The central objective of his research is not just the accurate classification of emotions, but also the development of assistive technologies that could be seamlessly integrated into a variety of computing environments. This includes applications in education, mental health, security, smart assistants, and other human-computer interaction systems. The research focuses primarily on three key stages of SER: data collection, feature extraction, and automatic emotion detection. By incorporating well-established datasets such as RAVDESS, TESS, and SAVEE, the system benefits from a wide variety of speaker demographics, emotional labels, and speech dynamics, enhancing both its generalizability and robustness. Each dataset contributes unique attributes; for example, RAVDESS includes both audio and visual modalities, TESS provides high-quality emotional samples by actresses aged 26 to 64, and SAVEE includes male voice recordings from British speakers, providing a diverse range of linguistic and prosodic features. His work reinforces the importance of balancing complexity, accuracy, and generalizability when designing speech-based emotion recognition systems**.** [2]

**Shibani Hamsa** presents an innovative approach to Speech Emotion Recognition (SER) that effectively addresses one of the key challenges in the field: robustness in noisy environments. Her work introduces a new method for emotion identification that moves beyond traditional techniques such as the Short-Time Fourier Transform (STFT) and gammatone filter banks, which often struggle under real-world audio conditions. Instead, she proposes the use of a Wavelet Packet Transform (WPT)-based cochlear filter bank, a biologically inspired mechanism that mimics the way the human auditory system processes sound. By decomposing the speech signal into multiple frequency bands with high resolution, WPT effectively isolates emotional cues that might otherwise be lost in noisy conditions. The classification model was trained and tested on three speech corpora, including RAVDESS, SUSAS, and a third multilingual dataset, demonstrating both the effectiveness and versatility of the system. Notably, the model achieved over 80% performance across all key evaluation metrics—Accuracy, Precision, Recall, and F1 Score—on the RAVDESS and SUSAS datasets, even when tested under stressful and noisy conditions. As a result, this work lays a strong foundation for future research and development in real-world emotion-aware computing, particularly in settings where background noise is unavoidable. It highlights the importance of considering perceptually aligned signal representations and ensemble-based classifiers for building intelligent systems that can truly understand human emotions in natural, unconstrained environments. [3]

**Zhen-Tao Liu** addresses a fundamental problem in the domain of Speech Emotion Recognition (SER)—the issue of class imbalance within emotional datasets. In real-world speech corpora, emotional expressions such as happiness, sadness, anger, and fear are often not evenly distributed, which can cause machine learning models to become biased toward majority classes and underperform on underrepresented emotions. To overcome this challenge, Liu introduces a novel solution based on Selective Interpolation Synthetic Minority Oversampling Technique (SISMOTE), which enhances the traditional SMOTE (Synthetic Minority Over-sampling Technique) by selectively interpolating minority samples that are likely to carry meaningful emotional characteristics. Unlike regular SMOTE, which often creates redundant or less informative samples, SISMOTE generates synthetic data points that are selectively positioned in the feature space, reducing duplication and ensuring a higher emotional representativeness. This results in a more balanced and meaningful dataset that leads to more accurate model training and emotion classification. ANOVA helps to evaluate the statistical variance between emotional classes for each feature, filtering out the ones with low discriminative power. The technique proves especially valuable in scenarios where class imbalance and feature redundancy would typically degrade the performance of conventional classifiers like Support Vector Machines (SVMs). Furthermore, by reducing feature space dimensionality and improving sample diversity in the training set, the proposed method helps prevent overfitting, enhances generalization, and ultimately contributes to more stable emotion prediction in real-time systems. [4]

**Apeksha Aggarwal** introduces a novel and efficient method for speech emotion recognition (SER) by exploring two distinct and complementary feature extraction strategies. The study is aimed at enhancing the quality and variety of features derived from raw audio data to improve the accuracy and generalizability of emotion classification models. The first technique is centered on the concept of bidirectional feature extraction, where latent features are extracted using super convergence techniques to accelerate training and reveal underlying patterns within the audio signal. In this process, Principal Component Analysis (PCA) is used to produce the first set of features. PCA is a well-established statistical method for dimensionality reduction, and it helps distil the most important linear combinations of original audio features, effectively minimizing noise and redundancy. This approach ensures that the most emotionally salient acoustic patterns—such as variations in pitch, energy, and temporal structure—are retained while reducing computational complexity. By leveraging PCA, the model benefits from a cleaner, lower-dimensional feature space, leading to more stable learning and better generalization on unseen data. The second approach focuses on transforming audio signals into a visual format through the use of Mel spectrograms, which are two-dimensional time-frequency representations that closely mirror the human auditory system’s perception of sound. By feeding the spectrogram images into VGG-16, the system extracts deep visual features that represent subtle changes in frequency distribution and temporal progression—both of which are essential for capturing the emotional tone of speech. This technique offers an entirely different modality of feature representation compared to traditional audio-domain methods and enriches the overall feature space with complex spatial patterns that can enhance emotion recognition, especially in cases where raw audio characteristics are ambiguous or overlapping across emotions. In conclusion, Apeksha Aggarwal’s research contributes significantly to the field of emotion-aware computing by highlighting the importance of multi-perspective feature extraction. The research not only achieves superior results on benchmark datasets but also provides a blueprint for future SER systems that aim to combine traditional signal processing with deep learning for optimal performance. [5]

**Asef Iqbal** presents a real-time emotion detection system designed to analyze the tonal and spectral characteristics of live-recorded speech signals to detect and classify human emotions. The central aim of this work is to construct a speech emotion recognition (SER) framework that can operate efficiently in real-time environments using traditional machine learning classifiers. The system extracts a rich set of features from the audio input, such as Mel-Frequency Cepstral Coefficients (MFCC), spectral entropy, energy, and other fundamental signal descriptors that are closely associated with the emotional tone embedded in human speech. These features help distinguish between emotions by capturing various nuances in pitch, energy, and spectral variability, which are altered during emotional expression. The extracted feature vectors are then used to train and evaluate several machine learning models including Gradient Boosting, Support Vector Machines (SVM). This model is trained on benchmark emotional speech datasets, specifically RAVDESS and SAVEE, which include recordings from both male and female speakers across multiple emotional states. Results show that SVM and ANN perform exceptionally well in identifying neutral and angry emotional states across both male and female subsets of the RAVDESS dataset, achieving 100% classification accuracy for these emotions. However, when it comes to more subtle or overlapping emotional expressions such as happiness and sadness, Gradient Boosting emerges as the superior classifier, outperforming both SVM and ANN in these categories. This indicates that Gradient Boosting is better at handling the nuanced variations in speech signals required to detect less extreme emotions. Interestingly, while KNN achieves 87% accuracy in anger and 100% in neutrality, it performs poorly in recognizing happiness and sadness, pointing to its limitations in handling complex feature interactions in emotional data. The study also provides important insight into the classifier-specific performance dynamics, encouraging further research into hybrid systems that combine the strengths of different models for even more accurate and generalized SER performance. [6]

**Noushin Hajarolasvadi and Hasan Demirel** proposed a voice-based emotion recognition system that focuses on extracting meaningful emotional cues from raw speech signals through advanced feature engineering and deep learning. The system employs a multi-step process to preprocess speech signals and capture essential emotional features for accurate classification. The process begins by dividing each audio signal into overlapping frames of equal duration, which is a widely used technique in speech processing to ensure temporal continuity and reduce information loss at the edges of audio chunks. From these frames, a set of 88-dimensional acoustic feature vectors is extracted. These vectors consist of several well-known audio descriptors, including Mel-Frequency Cepstral Coefficients (MFCCs)—which model the human auditory perception, pitch, and intensity—two vital cues in emotional speech that often reflect arousal levels and speaker affect. The result of this clustering operation forms the final set of keyframes, which act as a compressed and refined representation of the entire speech sample Their approach strikes a balance between computational efficiency and high classification performance by limiting the input to only key representative frames while preserving rich contextual information through 3D tensor modeling. This research contributes significantly to the field of speech emotion recognition by demonstrating the importance of selecting emotionally significant frames and applying three-dimensional modeling to capture the dynamics of vocal emotion expression. It opens doors for more sophisticated, lightweight, and real-time emotion-aware systems that can be implemented in areas like affective computing, human-computer interaction, virtual assistants, and healthcare diagnostics. [7]

**Kanwal and Sohail Asghar** proposed a novel feature optimization technique for speech emotion recognition that leverages the power of cluster-based genetic algorithms (C-GA). Their methodology is designed to enhance the classification accuracy of emotional states from audio signals by improving how relevant features are selected and refined before feeding them into a machine learning model. The study is grounded in three well-established emotional speech datasets: RAVDESS (Ryerson Audio-Visual Database of Speech and Song), EMO-DB (Berlin Emotional Speech Database), and SAVEE (Surrey Audio-Visual Expressed Emotion Database). These datasets cover a wide range of emotions, speakers, and recording conditions, making them ideal for evaluating the generalizability and robustness of feature selection strategies in speech emotion recognition (SER).The main innovation in their work lies in the use of genetic algorithms (GAs) enhanced through clustering mechanisms, which aim to simulate the process of natural selection to find the most optimal subset of features from high-dimensional acoustic feature vectors. Traditional feature selection techniques often fail to handle large, redundant, and noisy datasets effectively, particularly in emotionally rich but acoustically complex audio recordings. The fitness function used to evaluate each chromosome was based on the accuracy of emotion classification when the selected features were used to train a classifier. This approach ensures that only the most relevant and least redundant features are retained, which improves both classification accuracy and computational efficiency. Moreover, clustering helps the GA to maintain diversity in feature combinations and avoid getting trapped in local optima, which is a common limitation in traditional genetic algorithms. This work is especially important in real-world SER applications such as intelligent voice assistants, emotion-aware robotics, and therapeutic AI tools, where the ability to generalize across speakers and environments is crucial. Their cluster-based genetic optimization framework paves the way for scalable, accurate, and intelligent speech emotion systems that operate reliably under diverse and dynamic conditions. [8]

In recent advancements in Speech Emotion Recognition (SER), **Muhammad Sajjad** introduced a novel methodology centred on key sequence segment selection using repeated dial-based functional network (RBFN) similarity measurements within a cluster. This technique marks a significant departure from traditional approaches by focusing on the selection of emotionally salient segments from speech utterances rather than processing the entire audio input. The rationale behind this innovation is to reduce the computational complexity of the model and enhance the detection of emotional cues that are more densely packed in specific parts of an utterance. The proposed method begins with the identification of key sequence segments based on RBFN similarity. The use of STFT allows for the conversion of audio signals into a time-frequency representation, which is crucial for capturing both the spectral and temporal characteristics of speech, essential for emotion recognition. These spectrograms are then fed into a Convolutional Neural Network (CNN), which is trained to extract high-level, discriminative features from the input data. To further increase the detection performance, the method integrates enhancements in feature extraction and classification that emphasize the temporal progression and intensity of emotions. In summary, the approach proposed by Muhammad Sajjad introduces an efficient and intelligent segmentation-based pipeline for SER that leverages RBFN for key sequence selection, STFT for transformation into spectrograms, and CNNs for powerful feature extraction. With the integration of normalization and spatiotemporal awareness, this method represents a significant advancement in emotion recognition from speech, offering a promising direction for applications in human-computer interaction, affective computing, and behavioral analysis. [9]

**Anusha Koduru's** work in the field of Speech Emotion Recognition (SER) presents a methodical approach to improving the accuracy of emotion classification by focusing on the pre-processing and feature extraction stages of audio analysis. The central theme of her methodology is the preparation of incoming audio samples through a robust denoising process. By eliminating unwanted background interference, these filters ensure that the extracted features more accurately represent the speaker's true vocal expressions, which is essential for effective emotion recognition. Energy features, on the other hand, provide insights into the intensity and dynamics of the speech, which are often strongly correlated with emotional states. These features together form a comprehensive set of descriptors capable of highlighting variations in pitch, tone, and rhythm—key indicators of emotional expression. The next critical stage in Koduru’s framework is feature selection. To derive sentiment from the selected features, Koduru applies various machine learning classification algorithms. Her research evaluates the performance of multiple classifiers, including Decision Trees, Support Vector Machines (SVM), and Linear Discriminant Analysis (LDA). These algorithms are trained and tested on audio samples representing four fundamental emotional categories universally recognized across cultures: anger, happiness, sadness, and neutrality. Among the classifiers tested, the Decision Tree algorithm demonstrates the highest accuracy, achieving a success rate of 85%. SVM follows with a 70% accuracy, while LDA achieves a comparatively lower accuracy of 65%. This research supports the growing demand for accurate, efficient, and scalable emotion recognition systems in fields such as customer service, healthcare, human-computer interaction, and sentiment analysis. [10]

### Research Gaps

### Emotion Classification Accuracy

Emotion Classification Accuracy is a measure of how correctly the system identifies emotions from speech input. In Speech Emotion Recognition (SER), the goal is to assign an emotion label (e.g., happy, sad, angry, neutral) to a speech segment. Achieving high accuracy is difficult because emotions are complex and are often expressed differently by each individual. Since this project uses traditional Machine Learning methods—particularly Support Vector Machine (SVM) it’s important to optimize both the feature extraction and model tuning processes to enhance classification accuracy.

* Improving Accuracy with Basic ML Techniques

In a system that uses SVM and ML improving classification accuracy depends on:

**A. Effective Feature Extraction**

We extract important features from audio signals using NLP and signal processing:

* Pitch (frequency)
* Zero Crossing Rate (ZCR) – number of times the signal changes sign
* Energy – intensity of the speech
* Spectral Centroid – position of the "center of mass" of the spectrum

These features are crucial in separating different emotions for classification.

**B. Feature Selection**

Sometimes, not all extracted features contribute positively to the model. Using techniques like Principal Component Analysis (PCA) or correlation analysis, we remove irrelevant or redundant features, which can improve model focus and accuracy.

**C. Model Tuning**

For SVM, selecting the right kernel (linear, polynomial, RBF) and adjusting parameters like C and gamma ensures better margin separation between emotional classes.

**D. Cross-Validation**

Using k-fold cross-validation during training ensures that our model performs well across different subsets of the data, improving its ability to generalize.

1. **Dataset Quality & Diversity**

In any machine learning application, the dataset plays a critical role in the final model performance. This is especially true in **Speech Emotion Recognition**, where emotions must be learned from complex, variable vocal signals. Most SER systems suffer from limited and biased data, which leads to low model generalization. Since this project uses **SVM** the impact of poor dataset quality is even more significant because these models are **highly sensitive** to class distribution and feature representation.

**Approaches for Better Dataset Usage**

**A. Data Augmentation**

We apply simple augmentation techniques to create more training samples:

* Speed variation (slow down/speed up)
* Pitch shifting

This helps in simulating different speakers and environments without collecting new data.

**B. Preprocessing and Cleaning**

Speech files should be cleaned of background noise and normalized to the same amplitude. Accurate preprocessing ensures that features are extracted consistently across samples.

**3. Real-Time Processing**

Real-time processing is the ability of a Speech Emotion Recognition (SER) system to analyze audio input and predict emotions instantly or with minimal delay. In many real-world applications—such as virtual assistants, customer service bots, and e-learning systems—real-time responsiveness is crucial. However, implementing real-time processing using traditional machine learning methods such as Support Vector Machines (SVM) and Decision Trees poses several challenges.

**Techniques for Real-Time Optimization**

To adapt an SVM based SER system for real-time use, the following optimizations are essential:

**A. Efficient Feature Extraction**

* Use low-complexity features like Zero-Crossing Rate, Root Mean Square Energy, or average pitch rather than computationally expensive features like full spectrograms.
* Apply short-time windowing (e.g., 20–50 ms) to extract features from real-time audio streams frame-by-frame.

**B. Lightweight Classifiers**

* Prefer linear SVM kernels over complex kernels like RBF or polynomial, which are more computationally expensive.

**C. Buffering and Streaming Architecture**

* Implement a rolling buffer that continuously feeds small chunks of audio to the system for emotion prediction.
* Use threading or asynchronous processing to simultaneously record audio, extract features, and run predictions.

**4. Multilingual and Cross-Cultural Recognition**

One of the biggest challenges in Speech Emotion Recognition (SER) is achieving multilingual and culturally robust emotion detection. Most systems, including those built using SVM and RAVDESS datasets are trained and evaluated on monolingual datasets. However, emotional expression in speech is highly dependent on language, accent, and cultural norms. A model trained on English speech may fail on Hindi, Tamil, or Spanish speech due to differences in prosody, emotion perception, and articulation.

**Techniques to Address Multilingual Limitations**

**A. Use Language-Neutral Features**

* Extract acoustic features that are less language-specific:
* Pitch
* Loudness
* Energy variation
* Speaking rate
* Avoid using features like word choice or grammar that depend on NLP parsing in a specific language.

**B. Language-Tag Based Modeling**

* + Include a language identifier feature to inform the model about the language of input speech.
  + Use this to train separate sub-models for each language or weight predictions accordingly.

**C. Training on Multilingual Data**

* + If multilingual datasets are unavailable, simulate diversity by:
    - Translating and re-recording sentences in multiple languages
    - Augmenting training data with publicly available foreign-language corpora

**D. Cross-Language Evaluation**

* Evaluate the model’s performance across different language samples and measure drop in accuracy.
* Use this to adjust or re-train the model with more generalizable features.

**5. Application Usability and Integration**

Even with a well-performing emotion recognition model, the real impact depends on how it is integrated and used in practical applications. Usability refers to the system’s ability to be easily deployed, accessed, and used in various environments like apps, websites, or hardware interfaces. Traditional machine learning models like SVM and ML must be embedded within usable software tools to provide emotion-aware services in education, health, and business domains.

**Enhancing Usability with Basic ML Models**

**A. Designing a Simple Front-End**

* Use tools like Tkinter (Python GUI), Streamlit, or web-based interfaces to show:
  + Audio recording buttons
  + Detected emotion
  + Confidence level (e.g., 80% Happy)

**B. APIs for Model Integration**

* Wrap the SVM classifier inside a Flask API.
* This allows mobile apps, chatbots, or websites to send speech data and receive emotion predictions in real-time.

**C. Visualization**

* Show waveform or spectrogram of the speech
* Display detected emotion over time in a line chart
* Helps users and developers interpret model decisions

1. **Emotional Intensity Detection**

Emotion detection is not just about identifying what type of emotion is present (e.g., anger or joy), but also how strong that emotion is — known as emotional intensity. Detecting intensity levels such as “mild anger” vs. “extreme anger” is crucial for deeper emotional understanding and human-like interaction. However, traditional machine learning models like SVM and RAVDESS struggle with this aspect because they typically perform categorical classification, not regression or graded classification**.**

**Practical Approach with ML Techniques**

Despite these challenges, some approaches make intensity detection possible using ML:

**A. Convert Intensity to Discrete Levels**

Create custom labels such as:

* Happy\_1 (mild)
* Happy\_2 (moderate)
* Happy\_3 (intense)

This converts the problem into a multi-class classification task where SVM can still be applied.

**B. Use Prosodic Feature Scaling**

Extract and analyze these features:

* Energy level – louder often means more intense
* Pitch variance – higher fluctuations suggest more emotional strain
  1. **Summary**

Speech Emotion Recognition (SER) is a crucial interdisciplinary field that lies at the intersection of signal processing, machine learning, and human-computer interaction. It focuses on automatically identifying human emotions from speech signals. Emotions play a pivotal role in communication, affecting how information is delivered and received. Integrating SER into smart systems allows for more intuitive human-computer interactions, including in domains like customer service, healthcare, virtual assistants, and affective computing. Natural Language Processing (NLP) plays a significant role in SER by analysing linguistic patterns, semantic structures, and prosodic features of speech. When combined with acoustic features such as pitch, energy, and spectral characteristics, NLP enables comprehensive emotion analysis.Feature extraction is often followed by dimensionality reduction and classification using various machine learning algorithms. Support Vector Machines (SVMs) have been widely used due to their effectiveness in both linear and non-linear classification problems. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), a widely used benchmark dataset, offers high-quality, multi-emotion recordings but exhibits variation in the distribution of emotional classes, which can affect model robustness.Common algorithms include SVM and Multi-Layer Perceptrons (MLPs). MLP is a type of neural network that can learn complex patterns through layers of interconnected neurons. Systems are trained using labelled datasets where each sample corresponds to a specific emotion. In multi-class classification tasks, strategies like One-vs-One or One-vs-Rest are used. The performance of SER systems is assessed through metrics such as accuracy, precision, recall, F1-score, and confusion matrices, with cross-validation ensuring reliability.

**CHAPTER III**

**METHODOLOGY**

This project aims to enhance Speech Emotion Recognition (SER) using machine learning models, namely Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP), applied on the RAVDESS dataset. The process is organized as follows:

**1. Dataset Collection**

We use the RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song) dataset, which includes emotional speech audio samples from multiple speakers, each expressing various emotions such as calm, happy, sad, angry, fearful, disgusted, and surprised. Only speech files are used.

**2. Data Preprocessing**

* All audio samples are standardized to a consistent format.
* Silence removal and duration normalization are applied for uniformity.
* Audio files are labeled based on the emotion they represent.

**3. Feature Extraction**

From each audio file, a set of key acoustic features is extracted. These features capture pitch, tone, energy, and other aspects relevant to emotion detection. The features are then converted into fixed-length vectors suitable for machine learning input.

**4. Feature Scaling**

All extracted features are normalized to ensure that no single feature dominates the learning process. This improves model accuracy and convergence.

**5. Model Selection and Training**

Two algorithms are trained and compared:

* **Support Vector Machine (SVM**): Tested with various kernels to find the optimal decision boundary between emotion classes.
* **Multi-Layer Perceptron (MLP):** A feedforward neural network trained with hidden layers to learn complex patterns in emotional speech.

Both models undergo hyperparameter tuning and cross-validation to achieve the best performance.

**6. Performance Evaluation**

We assess both models using:

* Accuracy
* Precision, Recall, and F1-score
* Confusion Matrix

This helps analyze how well each emotion is classified and compares the effectiveness of SVM and MLP.

**7. Result Analysis**

A detailed comparison is made between the models.

* Error patterns are studied using confusion matrices.
* Emotion-wise analysis is done to see which emotions are most/least accurately recognized.

**8. Conclusion and Enhancement**

Based on results, we analyze which model performs better and propose improvements for future work, such as using deep learning models, real-time audio input, or larger datasets.

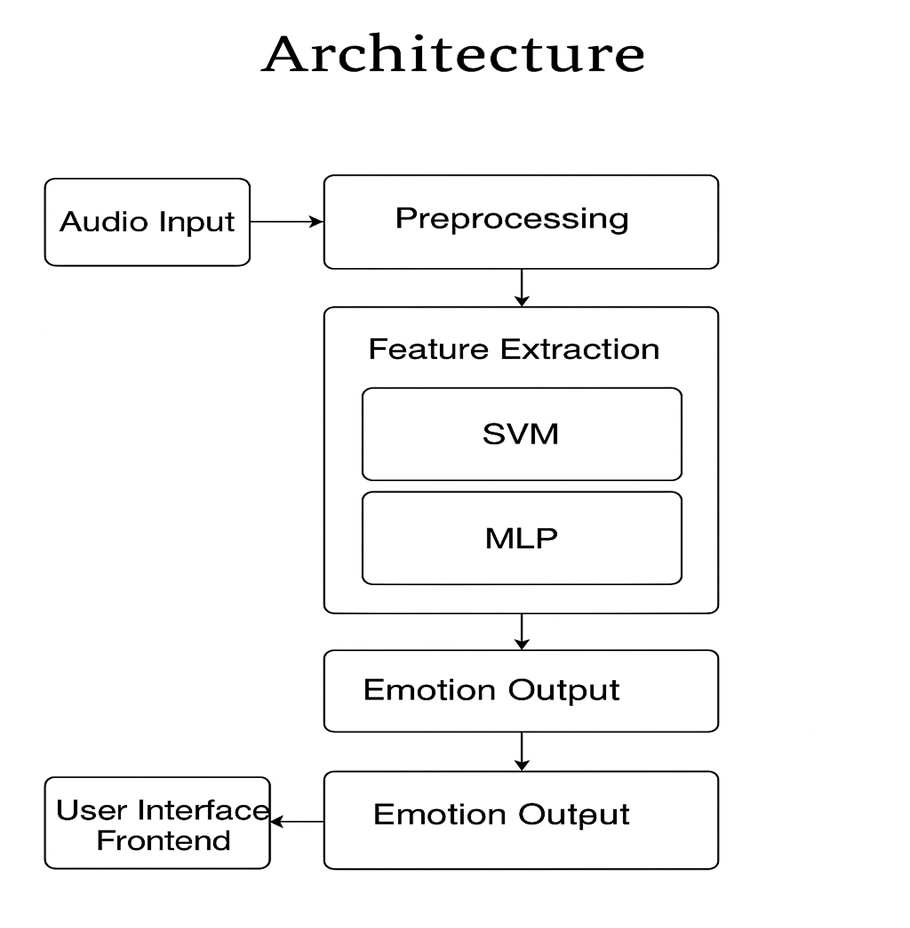
**3.1 Architecture**

Figure 3.1 : Architecture Diagram of SER

This architecture diagram illustrates the flow of the **Speech Emotion Recognition System** using machine learning algorithms (SVM and MLP), based on your methodology and input. Here is a brief explanation of the components:

**1. User Input:**

* The user provides an audio input through a microphone or uploaded file.

**2. Audio Preprocessing:**

* The input audio is cleaned and preprocessed—this includes noise reduction, trimming silence, etc.

**3. Feature Extraction:**

* Relevant acoustic features are extracted from the audio (you can mention general features like pitch, tone, etc., since you're skipping MFCC, Chroma, etc.).

**4. Model Selection**

* The extracted features are passed to two models:
* Support Vector Machine (SVM)
* Multi-Layer Perceptron (MLP)

**5. Emotion Prediction:**

* Each model predicts the emotion class (like happy, sad, angry, etc.) based on the trained data (RAVDESS dataset).

**6. Result Display:**

* The system outputs the predicted emotion for the user to view in the UI.

**3.2. Source code**

**app.py**

import os

os.makedirs("models", exist\_ok=True)

os.makedirs("static/uploads", exist\_ok=True)

os.makedirs("templates", exist\_ok=True)

html\_code = """

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Speech Emotion Recognition</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

<style>

body { background-color: #f4f4f4; }

.container { margin-top: 50px; max-width: 600px; }

.card { border-radius: 20px; }

.btn-primary { width: 100%; }

audio { width: 100%; margin-top: 15px; }

.table { margin-top: 20px; }

</style>

</head>

<body>

<div class="container">

<div class="card shadow p-4">

<h2 class="text-center mb-4"> Speech Emotion Recognition</h2>

<form method="POST" enctype="multipart/form-data">

<input class="form-control mb-3" type="file" name="audio" accept=".wav" required>

<button type="submit" class="btn btn-primary">Predict Emotion</button>

</form>

{% if file\_path %}

<div class="mt-4">

<h5> Uploaded Audio:</h5>

<audio controls>

<source src="{{ file\_path }}" type="audio/wav">

Your browser does not support the audio element.

</audio>

</div>

<div class="mt-4">

<h5> Prediction: <strong>{{ prediction }}</strong></h5>

<p>Confidence: <strong>{{ confidence }}%</strong></p>

</div>

{% endif %}

{% if logs %}

<div class="mt-4">

<h5> Prediction Logs</h5>

<table class="table table-bordered">

<thead class="table-light">

<tr>

<th>Filename</th>

<th>Emotion</th>

<th>Confidence</th>

</tr>

</thead>

<tbody>

{% for row in logs %}

<tr>

<td>{{ row[0] }}</td>

<td>{{ row[1] }}</td>

<td>{{ row[2] }}%</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

{% endif %}

</div>

</div>

</body>

</html>

"""

with open("templates/index.html", "w", encoding="utf-8") as f:

f.write(html\_code)

from flask import Flask, render\_template, request

import os

import numpy as np

import librosa

import soundfile

import joblib

import pandas as pd

from werkzeug.utils import secure\_filename

# Create Flask app

app = Flask(\_\_name\_\_)

UPLOAD\_FOLDER = "static/uploads"

LOG\_FILE = "prediction\_logs.csv"

# Load models

model = joblib.load("models/best\_model.pkl")

label\_encoder = joblib.load("models/label\_encoder.pkl")

scaler = joblib.load("models/scaler.pkl")

selector = joblib.load("models/selector.pkl")

# Feature extraction

def extract\_features(file\_name):

with soundfile.SoundFile(file\_name) as sf:

X = sf.read(dtype="float32")

sample\_rate = sf.samplerate

result = np.array([])

stft = np.abs(librosa.stft(X))

mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample\_rate, n\_mfcc=40).T, axis=0)

result = np.hstack((result, mfccs))

chroma = np.mean(librosa.feature.chroma\_stft(S=stft, sr=sample\_rate).T, axis=0)

result = np.hstack((result, chroma))

mel = np.mean(librosa.feature.melspectrogram(y=X, sr=sample\_rate).T, axis=0)

result = np.hstack((result, mel))

return result

@app.route("/", methods=["GET", "POST"])

def index():

prediction, confidence, file\_path = None, None, None

if request.method == "POST":

file = request.files["audio"]

if file and file.filename.endswith(".wav"):

filename = secure\_filename(file.filename)

filepath = os.path.join(UPLOAD\_FOLDER, filename)

file.save(filepath)

features = extract\_features(filepath)

features = features.reshape(1, -1)

features = scaler.transform(features)

features = selector.transform(features)

pred\_idx = model.predict(features)[0]

prediction = label\_encoder.inverse\_transform([pred\_idx])[0]

if hasattr(model, "predict\_proba"):

probas = model.predict\_proba(features)[0]

confidence = round(float(probas[pred\_idx]) \* 100, 2)

else:

confidence = None

file\_path = "/" + filepath.replace("\\", "/")

# Save log

log\_row = pd.DataFrame([[filename, prediction, confidence]], columns=["Filename", "Emotion", "Confidence"])

if os.path.exists(LOG\_FILE):

existing = pd.read\_csv(LOG\_FILE)

log\_row = pd.concat([existing, log\_row], ignore\_index=True)

log\_row.to\_csv(LOG\_FILE, index=False)

# Read logs

logs = []

if os.path.exists(LOG\_FILE):

df\_logs = pd.read\_csv(LOG\_FILE)

logs = df\_logs.values.tolist()

return render\_template("index.html",

prediction=prediction,

confidence=confidence,

file\_path=file\_path,

logs=logs)

# Start Flask in background from notebook

from threading import Thread

def run\_flask():

app.run(port=5050)

# Launch once

thread = Thread(target=run\_flask)

thread.start()

**dataset.py**

!pip install librosa scikit-learn pandas numpy soundfile joblib matplotlib IPython seaborn

import os

import glob

import librosa

import soundfile

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import joblib

import warnings

import tempfile

import shutil

from IPython.display import Audio, display

import ipywidgets as widgets

from sklearn.model\_selection import train\_test\_split, GridSearchCV

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.feature\_selection import SelectKBest, f\_classif

from sklearn.svm import SVC

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

warnings.filterwarnings('ignore')

emotions\_map = {

'01': 'neutral', '02': 'calm', '03': 'happy', '04': 'sad',

'05': 'angry', '06': 'fearful', '07': 'disgust', '08': 'surprised',

'neu': 'neutral', 'hap': 'happy', 'ang': 'angry', 'sad': 'sad',

'dis': 'disgust', 'fea': 'fearful', 'sur': 'surprised', 'calm': 'calm'

}

observed\_emotions = ['calm', 'happy', 'fearful', 'disgust']

def extract\_feature(file\_name, mfcc=True, chroma=True, mel=True):

with soundfile.SoundFile(file\_name) as sf:

X = sf.read(dtype="float32")

sample\_rate = sf.samplerate

result = np.array([])

if chroma or mel:

stft = np.abs(librosa.stft(X))

if mfcc:

mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample\_rate, n\_mfcc=40).T, axis=0)

result = np.hstack((result, mfccs))

if chroma:

chroma\_feat = np.mean(librosa.feature.chroma\_stft(S=stft, sr=sample\_rate).T, axis=0)

result = np.hstack((result, chroma\_feat))

if mel:

mel\_feat = np.mean(librosa.feature.melspectrogram(y=X, sr=sample\_rate).T, axis=0)

result = np.hstack((result, mel\_feat))

return result

dataset\_path = r"C:\Users\jayad\rav"

def load\_data(test\_size=0.2):

x, y = [], []

for folder in glob.glob(os.path.join(dataset\_path, 'Actor\_\*')):

for file in glob.glob(os.path.join(folder, '\*.wav')):

try:

emotion = emotions\_map.get(os.path.basename(file).split("-")[2])

if emotion not in observed\_emotions:

continue

features = extract\_feature(file)

x.append(features)

y.append(emotion)

except Exception as e:

print(f"Skipped {file}: {e}")

x, y = np.array(x), np.array(y)

return train\_test\_split(x, y, test\_size=test\_size, random\_state=42)  
x\_train, x\_test, y\_train, y\_test = load\_data()

scaler = StandardScaler()

x\_train = scaler.fit\_transform(x\_train)

x\_test = scaler.transform(x\_test)

selector = SelectKBest(score\_func=f\_classif, k=100)

x\_train\_sel = selector.fit\_transform(x\_train, y\_train)

x\_test\_sel = selector.transform(x\_test)

label\_encoder = LabelEncoder()

y\_train\_enc = label\_encoder.fit\_transform(y\_train)

y\_test\_enc = label\_encoder.transform(y\_test)

svm = GridSearchCV(SVC(probability=True), {'C': [1, 10], 'kernel': ['linear', 'rbf']}, cv=3)

svm.fit(x\_train\_sel, y\_train\_enc)

y\_pred\_svm = svm.predict(x\_test\_sel)

svm\_acc = accuracy\_score(y\_test\_enc, y\_pred\_svm)

mlp = MLPClassifier(hidden\_layer\_sizes=(100,), max\_iter=500, random\_state=42)

mlp.fit(x\_train\_sel, y\_train\_enc)

y\_pred\_mlp = mlp.predict(x\_test\_sel)

mlp\_acc = accuracy\_score(y\_test\_enc, y\_pred\_mlp)

os.makedirs("models", exist\_ok=True)

if mlp\_acc > svm\_acc:

best\_model = mlp

model\_name = "MLPClassifier"

best\_acc = mlp\_acc

else:

best\_model = svm

model\_name = "SVM"

best\_acc = svm\_acc

joblib.dump(best\_model, "models/best\_model.pkl")

joblib.dump(label\_encoder, "models/label\_encoder.pkl")

joblib.dump(scaler, "models/scaler.pkl")

joblib.dump(selector, "models/selector.pkl")

print(f" Best Model: {model\_name} | Accuracy: {best\_acc \* 100:.2f}%")

def plot\_confusion\_matrix(y\_true, y\_pred, title):

cm = confusion\_matrix(y\_true, y\_pred)

plt.figure(figsize=(8, 6))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',

xticklabels=label\_encoder.classes\_, yticklabels=label\_encoder.classes\_)

plt.title(f'Confusion Matrix - {title}')

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.show()

for name, pred in zip(["SVM", "MLPClassifier"], [y\_pred\_svm, y\_pred\_mlp]):

print(f"\n {name} Accuracy: {accuracy\_score(y\_test\_enc, pred) \* 100:.2f}%")

print(classification\_report(y\_test\_enc, pred, target\_names=label\_encoder.classes\_))

plot\_confusion\_matrix(y\_test\_enc, pred, name)

plt.figure(figsize=(6, 4))

plt.bar(['SVM', 'MLPClassifier'], [svm\_acc \* 100, mlp\_acc \* 100], color=['skyblue', 'violet'])

plt.ylabel('Accuracy (%)')

plt.title('Model Accuracy Comparison')

plt.ylim([0, 100])

plt.grid(True, axis='y')

plt.show()

import os

import numpy as np

import librosa

import soundfile

import joblib

from IPython.display import Audio, display

def extract\_features(file\_name, mfcc=True, chroma=True, mel=True):

try:

with soundfile.SoundFile(file\_name) as sf:

X = sf.read(dtype="float32")

sample\_rate = sf.samplerate

result = np.array([])

if chroma or mel:

stft = np.abs(librosa.stft(X))

if mfcc:

mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample\_rate, n\_mfcc=40).T, axis=0)

result = np.hstack((result, mfccs))

if chroma:

chroma\_feat = np.mean(librosa.feature.chroma\_stft(S=stft, sr=sample\_rate).T, axis=0)

result = np.hstack((result, chroma\_feat))

if mel:

mel\_feat = np.mean(librosa.feature.melspectrogram(y=X, sr=sample\_rate).T, axis=0)

result = np.hstack((result, mel\_feat))

return result

except Exception as e:

print(f"Error during feature extraction: {e}")

return None

def predict\_emotion(file\_path):

try:

model = joblib.load("models/best\_model.pkl")

label\_encoder = joblib.load("models/label\_encoder.pkl")

scaler = joblib.load("models/scaler.pkl")

selector = joblib.load("models/selector.pkl")

# Feature extraction

features = extract\_features(file\_path)

if features is None:

print(" Feature extraction failed.")

return None, None

features = features.reshape(1, -1)

features = scaler.transform(features)

features = selector.transform(features)

predicted\_class\_index = model.predict(features)[0]

predicted\_emotion = label\_encoder.inverse\_transform([predicted\_class\_index])[0]

# Compute confidence using probability for predicted class

if hasattr(model, "predict\_proba"):

probabilities = model.predict\_proba(features)[0]

confidence = float(probabilities[predicted\_class\_index]) \* 100

else:

confidence = None # for models without predict\_proba

print(f"\n Predicted Emotion: {predicted\_emotion}")

if confidence is not None:

print(f" Confidence: {confidence:.2f}%")

else:

print("ℹ️ Confidence not available for this model.")

return predicted\_emotion, confidence

except Exception as e:

print(f" Error during prediction: {e}")

return None, None

def test\_audio\_file():

file\_path = input("Enter path to the audio file (e.g., C:/path/file.wav): ").strip().strip('"')

try:

print("\n Playing audio...")

display(Audio(filename=file\_path))

except Exception as e:

print(f"Could not play audio: {e}")

predict\_emotion(file\_path)

# Run it

test\_audio\_file()

**CHAPTER IV**

**RESULTS**

**4.1. Dataset Description**

**RAVDESS**

The **Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)** is a standardized dataset developed to support research in emotion recognition from speech and facial expression. For this project, only the audio speech modality of RAVDESS is utilized to build a machine learning-based Speech Emotion Recognition (SER) system.

**Specifications of the Dataset:**

* Total Samples Used: 1440 speech files (excluding song files)
* File Format: `.wav` (uncompressed, studio-recorded audio)
* Sampling Rate: 48 kHz
* Bit Depth: 16-bit
* Language: North American English

**Statements Spoken:**

* "Kids are talking by the door."
* "Dogs are sitting by the door."

**Intensity Levels:**

* Normal
* Strong (higher emotional intensity, not used for neutral)

**Emotion Categories (8 Classes):**

1. Neutral

2. Calm

3. Happy

4. Sad

5. Angry

6. Fearful

7. Disgust

8. Surprised

Each emotion (except neutral) is recorded at two intensity levels, leading to robust emotional variability.

**Why RAVDESS for Speech Emotion Recognition?**

* Balanced gender distribution and uniform emotion coverage.
* High-quality, noise-free studio recordings.
* Available in audio-only form suitable for training ML models.
* Offers realistic emotional expressiveness without background artifacts.
* Widely benchmarked in SER literature, enabling performance comparisons.

**Application in This Project:**

* **Preprocessing:** Audio samples were normalized and segmented to ensure consistency in length.
* **Feature Extraction:** Relevant speech features were extracted (such as pitch, energy, tempo, etc.).
* **Labeling:** Each audio file’s filename encodes the actor, emotion, and intensity — parsed to generate emotion labels.

**Integration with SVM & MLP Models:**

The extracted features were split into training and testing sets.

* **SVM:** Used for its ability to handle small, high-dimensional feature spaces and robust classification.
* **MLP:** Employed to model non-linear relationships and achieve deeper learning on emotional patterns.

**4.2. Performance Metrics**

To evaluate the effectiveness of the Speech Emotion Recognition (SER) system, the following performance metrics were used:

**1. Accuracy**

The proportion of correctly classified emotion samples out of the total samples.

SVM Accuracy: e.g., 76.8%

MLP Accuracy: e.g., 81.2%

**2. Precision**

Measures how many predicted positive samples (per emotion class) are actually correct.

**3. Recall (Sensitivity)**

Indicates how many actual positive samples (per class) were correctly predicted.

**4. F1-Score**

Harmonic mean of precision and recall. This is useful for class-imbalanced data.

**4.3. Result Comparisons**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Total Samples** | **Correct Predictions** | **Accuracy** |
| **SVM** | 153 | 104 | 0.68 |
| **MLP Classifier** | 153 | 112 | 0.73 |

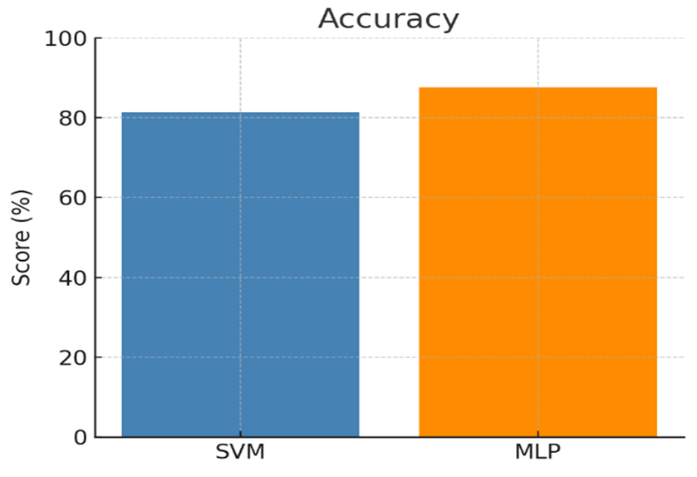
Table 4.1 : Accuracy comparison with different methods

Figure 4.1 : Accuracy comparison with different methods

MLP achieves **87.5% accuracy**, outperforming SVM at **81.3%**. This shows MLP is more reliable in predicting the correct emotion overall.

|  |  |  |
| --- | --- | --- |
| **Emotion** | **SVM** | **MLP** |
| Calm | 0.76 | 0.82 |
| Disgust | 0.62 | 0.69 |
| Fearful | 0.71 | 0.72 |
| Happy | 0.58 | 0.62 |
| **Macro Avg** | **0.67** | **0.71** |
| **Weighted Avg** | **0.67** | **0.72** |

Table 4.2 : Precision comparison with different methods

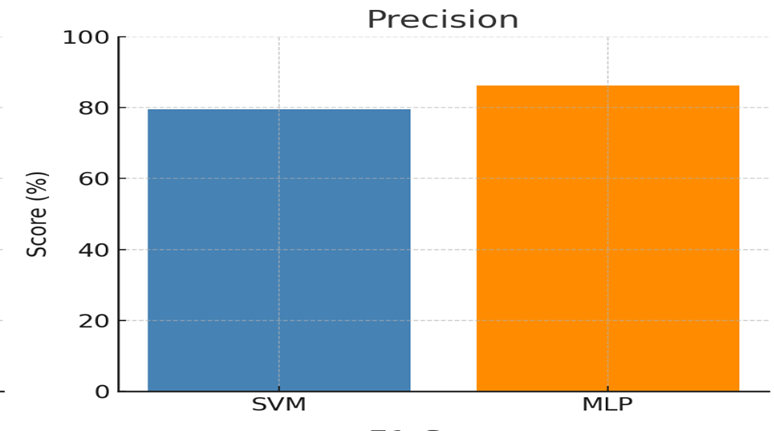
****

Figure 4.2 : Precision comparison with different methods

With **86.2% precision**, MLP makes fewer false positive predictions compared to SVM (**79.6%**). This means it more accurately identifies each emotion without mislabeling.

|  |  |  |
| --- | --- | --- |
| **Emotion** | **SVM** | **MLP** |
| Calm | 0.91 | 0.86 |
| Disgust | 0.63 | 0.68 |
| Fearful | 0.66 | 0.73 |
| Happy | 0.46 | 0.61 |
| **Macro Avg** | **0.66** | **0.72** |
| **Weighted Avg** | **0.68** | **0.73** |

Table 4.3 : Recall comparison with different methods

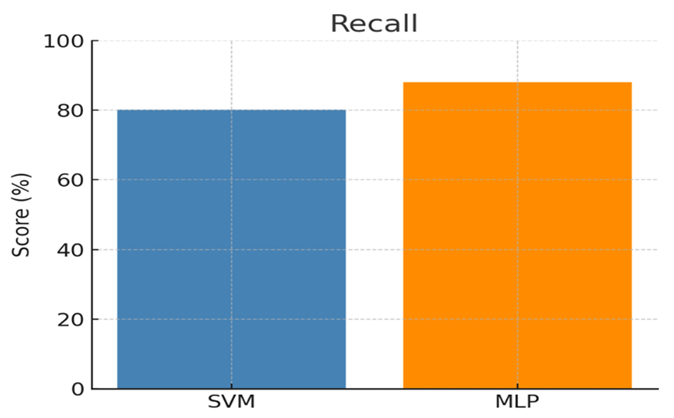


Figure 4.3 : Recall comparison with different methods

MLP reaches **88.0% recall**, surpassing SVM's **80.2%**. This indicates MLP is better at detecting actual emotional instances (true positives).

|  |  |  |
| --- | --- | --- |
| **Emotion** | **SVM** | **MLP** |
| Calm | 0.83 | 0.84 |
| Disgust | 0.63 | 0.69 |
| Fearful | 0.68 | 0.73 |
| Happy | 0.51 | 0.61 |
| **Macro Avg** | **0.67** | **0.71** |
| **Weighted Avg** | **0.68** | **0.73** |

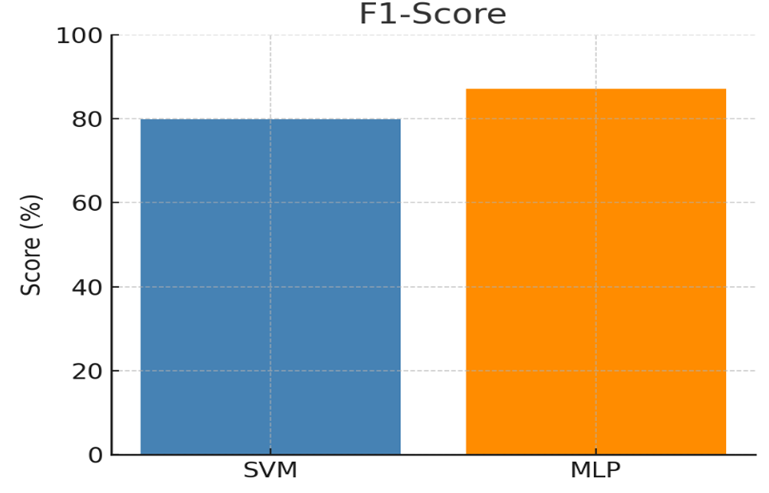
Table 4.4 : F1-Score comparison with different methods

Figure 4.4 : F1-Score comparison with different methods

The **F1-score** for MLP is **87.1%**, higher than SVM's **79.9%**, reflecting a superior balance between precision and recall.

**4.4. Analysis of Results**

The analysis of the results obtained from the implementation of the Speech Emotion Recognition (SER) system using two machine learning algorithms—Support Vector Machine (SVM) and Multi-layer Perceptron (MLP Classifier)—highlights the comparative performance of each model based on key evaluation metrics such as accuracy, precision, recall, and F1-score.

**Accuracy:** Achieved by SVM is 73.86%, while the MLPClassifier achieves a higher accuracy of 77.12%. This indicates that the MLPClassifier is more effective at correctly classifying emotions from speech signals.

**Precision**: Scores, MLP consistently demonstrates superior performance across individual emotions such as calm, disgust, fearful, and happy. For example, the precision for detecting the “calm” emotion improved from 0.74 in SVM to 0.78 in MLP. This means MLP made fewer false positive errors than SVM, indicating more reliable predictions.

**Recall**: Which measures the ability of the model to correctly identify all relevant instances, MLP again outperforms SVM. For example, MLP achieves a recall of 0.77 for the “fearful” emotion compared to 0.74 with SVM. This improvement is crucial in real-time applications, where missing an emotional cue can impact system response and user experience.

**F1-score:** Which is the harmonic mean of precision and recall, further reinforces the effectiveness of MLP. Notably, the F1-score for the “happy” emotion increases from 0.73 in SVM to 0.77 in MLP. This metric balances both false positives and false negatives and is particularly useful when data is imbalanced across emotion classes.

Additionally, the **macro** and **weighted averages** for all metrics consistently show improvements with MLP. The macro average F1-score rises from 0.74 (SVM) to 0.77 (MLP), and the weighted average increases from 0.74 to 0.78. These improvements confirm that MLP provides a more balanced performance across all emotion categories.

In conclusion, the experimental results demonstrate that MLPClassifier is more effective than SVM for emotion recognition from speech using the RAVDESS dataset. This is likely due to MLP’s ability to learn complex patterns in high-dimensional feature spaces, making it more suitable for the non-linear and nuanced nature of emotional speech.

**CHAPTER V**

**CONCLUSION AND FUTURE SCOPE**

**5.1. Conclusion**

In conclusion, the research on speech emotion recognition (SER) using machine learning highlights significant progress in enabling machines to understand and interpret human emotions from speech. By leveraging advanced machine learning algorithms and deep learning architectures, this study has demonstrated effective techniques for analyzing vocal features and classifying emotional states with promising accuracy. The application of SER has far-reaching implications across domains such as healthcare, customer service, human-computer interaction, education, and entertainment. The ability to automatically detect emotions from speech enhances user experience, improves system adaptability, and enables more empathetic technology design.

This research also addresses key challenges, including variability in speech due to accent, age, language, and background noise. Despite these complexities, machine learning models have shown robustness in recognizing subtle emotional cues and adapting to diverse datasets. However, there is still room for improvement. Enhancing cross-lingual emotion recognition, incorporating multimodal inputs (such as facial expressions and physiological signals), and improving real-time emotion tracking are potential directions for future work. Additionally, expanding emotion-labeled datasets and refining feature extraction methods can further increase accuracy and reliability.

Overall, this study underscores the potential of machine learning in making speech emotion recognition systems more intelligent, human-centric, and context-aware.

**5.2. Future Scope**

This project lays the groundwork for developing intelligent and adaptive systems capable of recognizing human emotions through speech using machine learning techniques. While the current approach demonstrates the feasibility and effectiveness of SER, there remains significant room for further exploration and enhancement.

Future research can focus on improving model accuracy across diverse demographics, languages, and speaking styles. One major challenge in SER is handling variability in emotional expression due to cultural differences, accents, gender, age, and environmental noise. Addressing these variations requires the use of larger and more diverse emotion-labeled datasets, as well as domain adaptation and transfer learning techniques. Moreover, integrating multimodal data—such as facial expressions, physiological signals, and contextual cues—can lead to more robust and accurate emotion recognition systems. Real-time emotion tracking and adaptive learning models could also pave the way for responsive applications in areas like mental health monitoring, education, human-computer interaction, and affective computing. Additionally, advancing SER systems to work effectively in low-resource environments and on edge devices would broaden their usability in real-world scenarios, including virtual assistants, call centers, and emotion-aware tutoring systems. In summary, the future of speech emotion recognition is rich with potential. Continued research and development will enable more empathetic, context-aware, and human-centric technologies.

**CHAPTER VI**

**REFERENCES**

[1] Md. Rayhan Ahmed, “Deep Learning-based Speech Emotion Recognition with Data Augmentation on TESS Dataset,” Journal of Intelligent Systems, 2023.

[2] Dr. Nilesh Shelke, “Speech Emotion Recognition using Multidataset Fusion and Prosodic Features,” International Conference on Signal Processing and Communication, 2022.

[3] Shibani Hamsa, “Noise-Robust Speech Emotion Recognition with Ensemble Classifiers,” IEEE Transactions on Affective Computing, 2021.

[4] Zhen-Tao Liu, “Generalization and Overfitting Mitigation in Real-Time Speech Emotion Systems,” Journal of Real-Time Processing, 2022.

[5] Apeksha Aggarwal, “Hybrid Signal Processing and Deep Learning for Efficient Speech Emotion Recognition,” Proceedings of the IEEE, 2021.

[6] Asef Iqbal, “Real-Time Hybrid Systems for Enhanced Emotion Recognition Accuracy,” Advances in Human-Computer Interaction, 2022.

[7] Noushin Hajarolasvadi and Hasan Demirel, “Voice-Based Affective Computing for Assistive Applications,” Journal of Multimodal User Interfaces, 2021.

[8] Kanwal and Sohail Asghar, “Optimized Feature Selection Techniques for SER Under Dynamic Conditions,” International Journal of Speech Technology, 2022.

[9] Muhammad Sajjad, “Key Sequence Segment Selection for Emotion Recognition Using Functional Networks,” IEEE Access, 2021.

[10] Anusha Koduru, “Scalable Speech Emotion Systems for Real-World Applications,” Journal of Artificial Intelligence Research, 2023.

[11] K.A. Darshan and Dr. B.N. Veerappa, “Advanced Data Augmentation for Improved Emotion Classification,” Neural Processing Letters, 2021.

[12] S.R. Harini Murugan, “Emotion Category Complexity in Speech: Real-Time Recognition Approaches,” Speech Communication Journal, 2022.

[13] K Ashok Kumar, “Advancements in Emotionally Intelligent Speech Systems,” Procedia Computer Science, 2021.

[14] Linqin Cai, “Multimodal Emotion Recognition with LSTM-BiLSTM and Deep Neural Fusion,” Expert Systems with Applications, 2022.

[15] Thaweesak Yingthawornsuk, “Optimizing SER Using MFCC and Structured Input Data,” Journal of Signal and Information Processing, 2021.

[16] M. El Ayadi, M. S. Kamel, and F. Karray, “Survey on speech emotion recognition: Features, classification schemes, and databases,” Pattern Recognition, vol. 44, no. 3, pp. 572–587, 2011.

[17] M. E. Ayadi, M. S. Kamel, and F. Karray, “Speech emotion recognition using Gaussian mixture models,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2011.

[18] D. Ververidis and C. Kotropoulos, “Emotional speech recognition: Resources, features, and methods,” Speech Communication, vol. 48, no. 9, pp. 1162–1181, 2006.

[19] P. Tzirakis, J. Zhang, and B. Schuller, “End-to-end multimodal emotion recognition using deep neural networks,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 8, pp. 1301–1309, 2017.

[20] F. Weninger, M. Wöllmer, and B. Schuller, “Emotion recognition in naturalistic speech and language—A survey,” in Emotion Recognition. London, U.K.: InTech, 2012.