SpeechSentio: AI-powered speech therapy with emotion analysis

J Vijay Gopal
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
vijayjagadam@gmail.com

Dr. K. Sai Prasad
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
saiprasad.kashi@mlrinstitutions.ac.in

Amulya Behara
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
amulyabehara01@gmail.com

Pavithra Jasthi
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
jasthipavithra17@gmail.com

Abhinay Kunapuli
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
iamabhinay20@gmail.com

Lohitha Vasamsetty
Computer Science Engineering
Artificial intelligence and Machine learning
MLR Institute of Technology
Hyderabad, India
lohithav16@gmail.com

ABSTRACT

Delving into the intricate realm of stuttering detection and its integration with emotion recognition, this study offers an in-depth analysis of current developments and issues in these specialized fields. The level of difficulty of stuttering, a complicated speech disorder, is outlined and contemporary techniques are considered for detecting and measuring speech disfluencies. Recognizing the complex relationship between stuttering patterns and emotional expressions, and including emotion identification into the stuttering detection framework, is a major area of attention for the research. To precisely identify stuttering speech through acoustic analysis, the study looks into a variety of procedures, including signal processing techniques and machine learning algorithms.

Keywords:
Acoustic analysis, Emotion recognition, Machine learning algorithms, Signal processing, Speech disorder, Stuttering detection, Therapeutic interventions.

1 INTRODUCTION

Within the dynamic and ever-evolving domain of speech therapy, the distinguishing proof and treatment of speech obstacles, notably stuttering, have remained central focuses of continuous exploration and advancement. The perplexing nature of stammered speech presents challenges for conventional strategies of identification and intervention, requiring a shift towards more precise, scalable, and personalized approaches driven by advancements in innovation.

Recent research endeavors have yielded promising results by leveraging advanced [12] signal processing strategies to automatically perceive and classify occurrences of stuttered speech. Through meticulous analysis of speech data and the extraction of related highlights, researchers have accomplished notable success in recognizing distinct stammering patterns. [29]This nuanced understanding of stuttering not only improves diagnostic accuracy but moreover illuminates the advancement of focused on therapeutic interventions.Our research paper presents a novel approach by integrating stutter detection with emotion detection, a unique feature not explored in previous studies. By combining these two components into a cohesive model, we aim to provide a more comprehensive understanding of the emotional aspects associated with stuttering behavior, offering valuable insights for both clinical applications and psychological research.

Moreover, the integration of emotion recognition technology has presented a transformative measurement to stuttering treatment. By capturing and analyzing emotional nuances embedded inside speech, therapists can make a supporting and tailored environment that supports motivation and quickens advance for people experiencing speech therapy. This all-encompassing approach recognizes the exchange between emotional factors and speech fluency, cultivating a more comprehensive and compelling therapeutic experience.

As the landscape of speech treatment proceeds to evolve, there's a developing emphasis on tackling cutting-edge technology to customize treatment plans, provide immediate input, and optimize the generally efficacy of stammering therapy. Advancements such as SpeechSentio represent this trend, offering advanced devices and platforms that empower therapists to tailor interventions to person needs, monitor advance in real-time, and adjust procedures appropriately.

This convergence of technology and therapy holds immense guarantee for upgrading the quality of life for people grappling with speech obstacles. By combining logical insights into speech patterns with advancements in computational algorithms and artificial intelligence, groundbreaking solutions are paving the way for more productive, available, and individualized approaches to stammering detection and therapy.

Eventually, this integration of stuttering identification and speech emotion not only benefits people with speech obstacles but also moves the field of speech therapy into a new period of development and affect. When experts from different fields work together and leverage new technologies, it empowers people to break down communication barriers and achieve their greatest potential.

2 BRIEF INTRODUCTION OF MLP

Another essential element of the system's architecture, and namely, the emotion recognition block, is the Multilayer Perceptron. The MLP is trained on an emotion-based labelled speech dataset and uses complex feature extraction methods special for speech processing, such as Mel-frequency cepstral coefficients, that allow analysis of spectral details of speech signals. As a result, due to the detected sophisticated connections between these features and a certain set of emotions, the MLP conducts efficient testing of how emotions can be identified in unseen speech samples. This novelty

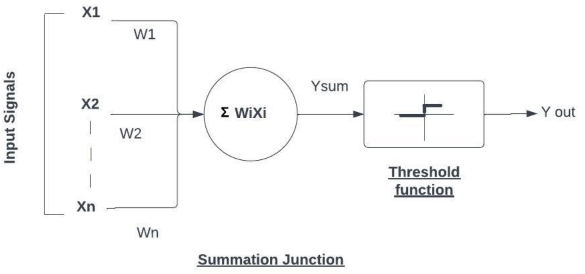


Figure 1: McCulloch-Pitts Model

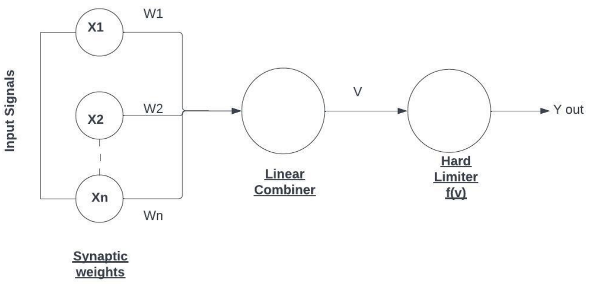


Figure 2: Rosenblatt's Perceptron

demonstrates the opportunity to test the usage of machine learning for emotion recognition tasks, demonstrating the system's potential for enhancing speech therapy interventions with emotion analysis capabilities.

2.1 History of MLP

The multilayer perceptron highlights a wealthy history dating back to the 1940s when Warren McCulloch and Walter Pitts displayed the concept of fake neurons, laying the establishment for neural organize inquire about. In any case, it wasn't until the 1960s and 1970s that basic movements were made inside the outline of the perceptron, a single-layer neural system competent of twofold classification. In any case of the starting enthusiasm, the perceptron's restrictions in dealing with nonlinear issues driven to a decay in intrigued

The resurgence of intrigued in neural systems occurred within the 1980s with the introduction of the MLP, a more successful plan competent of learning complex designs through different layers of neurons. This breakthrough, coupled with the improvement of backpropagation, an compelling training methodology for altering network weights, revitalized the field of neural frameworks. Subsequently, MLPs have advanced with milestones in optimization methodologies, activation functions, and hardware capabilities, advancing to become a establishment of contemporary-era machine learning and laying the foundation for more progressed neural network structures.

2.2 Detailed Explanation of MLP

Multilayer Perceptrons represent a foundational cornerstone in the domain of artificial neural networks, offering a versatile solution extensively applied across various machine learning and pattern recognition endeavors. The multi-layered architecture of the MLP, which can be described as the system of interconnected nodes of input, hidden, and output levels, boasts very significant capabilities in recognizing any intricate details and correlations present in the training datasets. The components of feed-forward multi-layer perceptrons are structured in such a way that they are able to transmit information smoothly from the input nodes through hidden layers to the output nodes that process tasks like classification, regression, or clustering.

A multi-layer perceptron (MLP) is a computational device in the sense that the nodes mitigate to the machines by doing their transformation functions, giving weights to input data, and sending outputs to layers above. The hidden layers feature map the structures in complex data distribution spaces, the model can be enhanced through non-linear interactions and higher order features being captured. BP helps neural networks in learning faster during the training process by acting as a learning signal that drives changes in the model's input biases and weights. Such changes are fine-tuned in the direction of a shrinking gap in the differences between the actual outputs and the expected ones. Thus, the model parameters become optimized.

That is a nonlinear chain of activities, namely predictor generation, error computation, and in trajectory of update, for the parameters, which finally yields introduction of predictive capabilities for the MLP. Among a vast ensemble of neural network architectures, MLPs have been highly praised for their performance handling structured data and adaptability for representing non-linear linkages which extends its applicability in a broad range of domains, such as speech recognition and image classification.

Mathematical Foundations

. Activation Function: When all neurons within a multilayer perceptron employ a linear activation function, meaning a function that directly maps the weighted inputs to each neuron's output, linear algebra demonstrates the reduction of any number of layers to a simplified two-layer input-output model. However, in Multilayer Perceptrons (MLPs), various neurons utilize nonlinear activation functions, which were designed to emulate the firing frequency of biological neurons' action potentials. The activation functions that have been historically common are both sigmoids, and they can be described as:

$$y(v_i) = \tanh(v_i)$$

and

$$y(v_i) = \frac{1}{1 + e^{-v_i}}$$

The first is a hyperbolic tangent that ranges from -1 to 1, while the other is the logistic function, which is similar in shape but ranges from 0 to 1. Here y_i represents the output of the ith node (neuron), and v_i is the weighted sum of the input connections. In the realm of alternative activation functions, several proposals have emerged, encompassing options such as the rectifier and softplus functions. Additionally, more specialized activation functions, like radial basis functions utilized in radial basis networks, have been proposed, representing another class of supervised neural network models.

Notably, recent advancements in [8] deep learning have seen the widespread adoption of the rectified linear unit (ReLU) as a prominent solution to address numerical challenges associated with traditional sigmoid functions.

Learning:

Learning occurs within the perceptron through the modification of connection weights subsequent to the processing of each data point, contingent upon the observed error in the output in contrast to the anticipated outcome. We can represent the error in an output node j in the nth data point (training example) by $e_j(n) = d_j(n) - y_j(n)$, where $d_j(n)$ is the desired target value for the nth data point at node j, and $y_j(n)$ is the value produced by the perceptron at node j when the nth data point is given as an input.

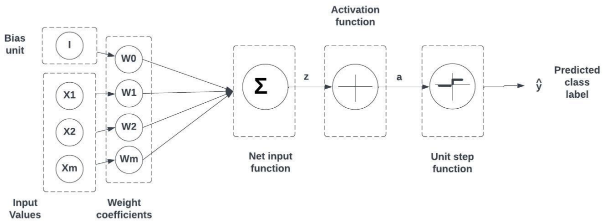


Figure 3: Multi-layer Perceptron

The node weights can then be adjusted to minimize the error in the entire output for the nth data point, given by $\mathcal{E}(n) = \frac{1}{2}\sum_{\text{output node } j} e_j^2(n)$.

Using gradient descent, the change in each weight w_{ij} is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$

where $y_i(n)$ is the output of the previous neuron i, and η is the learning rate. In this expression, $\frac{\partial \mathcal{E}(n)}{\partial v_j(n)}$ denotes the partial derivative of the error $\mathcal{E}(n)$ with respect to the weighted sum $v_j(n)$ of the input connections of neuron i.

3 EXISTING PROBLEMS

In the current landscape of speech therapy, there exist challenges related to the identification and management of speech obstacles, particularly within the context of stammering. Conventional methods face obstacles in terms of accuracy, versatility, and tailoring interventions to person needs. Manual assessment by speech-language professionals, whereas important, may need the ability to supply immediate insights into interesting speech patterns and the emotional aspects of speech challenges. With the fast advancement of technology, there's a growing request for more advanced and successful methodologies to help within the detection and treatment of stuttering.

Studies of recent times point to challenges in applying superior strategies to responses to circumstances of stuttering automatically and categorizing those occurrences. Features like the analysis of speech potentials and iteratively retrieving the significant features seem to gain recognition of stuttering manifestations and patterns. And to think of emotion recognition technology as part of a community setting, are opportunities to shape a supportive and personalized environment, which in this regard, would assist in increasing my motivation for the speech therapy and in turn attain a lot of progress.

By the future aimed in the field of speech therapy, the technology plays the vital role in developing the methods of building the treatment plans, really quick feedbacks and innovating the system of stuttering therapy. These solutions which appear to combine cutting-edge advances with a deep knowledge of speech variances and emotionality nuances, are able to cope with the complexities of stammering detection and treatment due to their effectiveness. With the mingling of tech and therapy, more efficacious outcomes can be achieved in the speech therapy among individuals with speech impairments and this field of treatment can be leveled to personalized and momentous interventions.

4 PROPOSED SYSTEM

4.1 Problem Statement

For speech therapy, currently, there are some problems with improving the criteria of stuttering diagnosis keeping in account individual features, and it also should consider emotions. Although the hand-made option still provides remarkable precision diagnostics, it is hard to classify a significant number of patients and provide immediate assessment of speech patterns subtleties. Moreover, another issue lies in the fact that the available solutions may not be efficient enough to deal with the emotional problems of the participants, it may not generate the necessary desire and progress in the rehabilitation process.

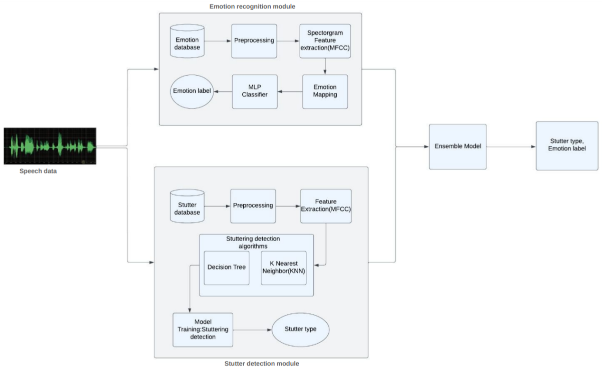


Figure 4: Proposed system architecture.

4.2 Objectives

- Immediate feedback on stutters: The system analyses your speech in real-time and tells you when you start with a stutter, thereby you can reply to the problem as soon as it appears and make progress more quickly.
- Creating a supportive environment: By allowing for emotional factors, its features will boost a more encouraging ambience which can benefit patients from a reduction of anxiety and will also motivate them during their therapy sessions.
- Tailored practice for faster improvement: A detailed examination of individual speech manner uniquely suited to a customized pronunciation exercise offers survival. In addition to this, it makes the learning process more effective, because users can play whenever they want instead of booking expensive classes.

4.3 Architecture

This speech recognition system adopts an innovative design for the recognition of emotions and stutters at the same time, which is realized by incorporating a two-tier architecture. The input data shared among the two sub-branches require specific preprocessing operations (various noise reduction procedures, removal of silence intervals, normalization) to correctly prepare a suitable speech signal. After the speech is processed emotion branch produces a spectrogram and then MFCC feature vectors are extracted and send to a completely connected classifier that matches speech patterns with emotions. The stutter branch utilizes similar MFCC extraction but employs algorithms like decision trees or K-Nearest Neighbors to analyze speech for disfluencies indicative of stutters. An optional ensemble model might combine these algorithms for potentially improved stutter detection. This approach allows researchers to not only recognize emotions but also analyze

5 IMPLEMENTATION

TensorFlow or Librosa can be used for preprocessing, spectrogram generation, and MFCC extraction. Emotion recognition likely employs an MLP trained with labeled emotional speech data, while stutter detection might use decision trees or KNN algorithms trained on labeled fluent/disfluent speech. These components would be integrated in Python to process speech input and output recognized emotions and detected stutters.

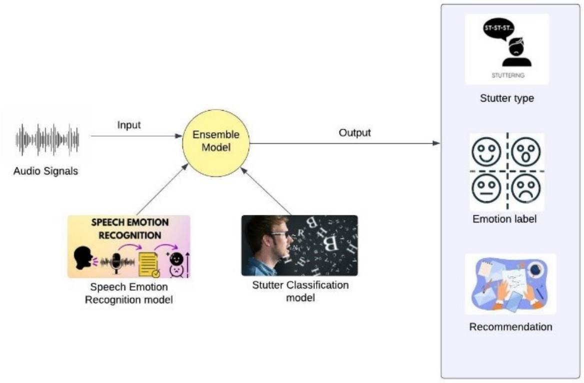


Figure 5: Representation of proposed system.

5.1 Data Collection

The data has been acquired from an online platform named Kaggle. The dataset for speech fluency analysis, potentially containing 17 attributes and 4145 rows of speech samples. The attributes likely include factors influencing fluency, such as "PoorAudio" (potentially indicating poor audio quality), disfluency types like "Prolongation" (lengthened sounds), "Block" (blocking disfluencies), and "Sound Rep" (sound repetitions). These disfluency types might be measured as counts (number of occurrences) or durations. Another column might represent a speech fluency score, possibly measured on a 10-point scale (higher scores indicating greater fluency). Additionally, the data might include information about the speaker (e.g., age, gender, diagnosis) in separate columns. Analyzing these factors alongside the fluency scores could help researchers understand the relationships between specific disfluency types, audio quality, and overall speech fluency.

5.2 Data Pre-Processing

This speech recognition system uses a two-part design to identify emotions and detect stutters. The first thing before any analysis of the speech recording is the segment containing a speech goes through a few preparatory stages. This aims at the cleanliness of the device to assist in the extraction of significantly meaningful features (MFCCs) that provides accurate output. To clean the speech signal in the first step the multi-stage process is used. In first stage of reduction of noise input, the system engages the families of the undesired background noise which increases the general clarity of speech details. To achieve this, the extraction of actual speech content which comes after the silences between the speech attempts uses this method. Conclusively, audio normalization will equalize the volume levels across speech samples of varying degrees, therefore serve as an even ground to support the detailed analysis. The last, by far most essential, stage require computation of the whole Mel-frequency Cepstral Coefficients (MFCCs). Translated, MFCCs stands for Mel Frequency Cepstral Coefficients which define ’fingerprint’ of the sound by representing the energy distribution across various frequency ranges. This is the fundamental operation similar to that of the human ear perception. This modification can impact emotions and stutters, which might further impact the way speech expresses its energy across frequency. Through MFCCs the system not only detects the certain spectral features but also finding the hints for the system is able to hold the emotion or speech stutter. MFCCs extraction is a similar thing to data filtering which apart from the raw speech data it delivers with a great focus. These Pop-ups for now stress on the obscure modifications in speech that point out the feelings and stutters among others. However, for the training of the analysis branches these become the fundamental blocks that lead to the extraction of the temporal and spectral features that are used for the analysis step. The emotion recognition domain, uses a pre-tagged speech data classifier that discovers patterns among MFCCs and in the process, identifies emotions . Meanwhile, the stutter detection branch leverages algorithms like decision trees or K-Nearest Neighbors. These algorithms focus on how the MFCCs change over time (temporal variations), as stutters often involve hesitations, sound repetitions, or prolonged syllables. By analyzing these temporal patterns, the algorithms can potentially detect the presence of stutters within the speech signal. In essence, the preprocessing steps prepare the data, while MFCC feature extraction serves as a critical bridge between the raw speech signal and the higher-level tasks of emotion recognition and stutter detection.

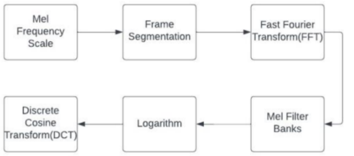


Figure 6: Flow diagram of MFCC

5.3 Building models with ML Algorithms

To create a model, the pre-processed data has been used. The data must be divided into two parts before a model can be built, with each part being utilized for the model’s construction and testing. Otherwise, it can result in overfitting. When a model can correctly predict learned data but not test data (or other data not included in the training set), this is known as overfitting. Typically, a ratio of 8:2 is considered for splitting dataset into training and test data. Following data splitting, an ML technique was used to train a model utilizing the training data. A variety of models have been constructed in the current study using a variety of machine learning methods, including RF, SVM, Gaussian NB, and LR. To maximize the DT’s precision, RF is employed.

- **Decision Tree**: Decision trees are machine learning algorithms that use a tree-like structure to classify data. They work by asking a series of yes-or-no questions based on specific features of the data. Speech analysis characteristics such as Mel-frequency cepstral coefficients (MFCCs) that are derived from speech signals can be used to train decision trees for stutter detection and emotion recognition in speech analysis. By analyzing these features at each decision node, the tree can learn to differentiate between fluent and disfluent speech segments for stutter detection. This allows the decision tree to efficiently classify speech samples based on the learned relationships between features and the desired outcome. In decision tree algorithms, entropy, Gini index, and information gain are key concepts utilized for feature determination and deciding the best split criteria. Here are the hypothetical concepts and equations for each of these measurements:

  5.3.1 ***Information Gain:*** The Information gain evaluates viability of a feature in decreasing entropy or impurity within the dataset. It makes a difference in selecting the leading feature for splitting the data. Figure shows the performance metrics of decision tree.
  Formula: $\mathrm{IG}(S,A) = \mathrm{Entropy}(S) - \sum_{v\in \mathrm{Values}(A)} \frac{|S_v|}{|S|} \times \mathrm{Entropy}(S_v)$
  Where:
  - $\mathrm{IG}(S,A)$ is the information gain by splitting dataset S on feature A.

- Values(A) are the possible values of feature A.
- S_v is the subset of instances in S for which feature A has value v.
- $|S|$ is the total number of instances in dataset S.

5.3.2 *Entropy:*

Entropy measures the impurity or clutter in a dataset. A lower entropy value indicates a more homogeneous dataset, whereas a higher entropy esteem signifies a more diverse dataset.
The formula for entropy is $S = -\sum_{i=1}^{c} p_i \log_2(p_i)$.
Where:

- S represents the dataset.
- c denotes the number of classes.
- p_i represents proportion of instances belonging to class i in the dataset S.

5.3.3 *Pseudo code of Decision Tree classifier model for Sound Repetition*.

```
Begin
Filter rows in df where 'NoStutteredWords'
is not equal to 0
Filter resulting rows where 'NaturalPause'
is equal to 0
Filter resulting rows where 'Interjection'
is equal to 0
Filter resulting rows where 'Prolongation'
is equal to 0
Filter resulting rows where 'WordRep'
is equal to 0
Filter resulting rows where 'Block'
is equal to 0

Set values in 'SoundRep'
column >= 1.0 to 1.0

Extract features X from the
last 13 columns of df

Extract target variable y
from 'SoundRep' column

Splitting the data into train and test
data (Xtrain, Xtest, ytrain, ytest) with
a test size of 0.4 and random state 42

Initialize a Decision Tree Classifier with
 criterion as 'entropy' and random state 5

Train the classifier on the
training data (Xtrain, ytrain)

Perform predictions on the test data and
convert them to arrays (ypred)

Extract actual values from ytest and
convert them to arrays (actual)

Display the count of predicted and
actual values

End
```

5.3.4 *Gini Index:*

Gini index may be a degree of impurity utilized to assess how well a specific feature splits the information into classes. A lower Gini index demonstrates a more homogeneous dataset.
Formula: Gini$(S) = 1 - \sum_{i=1}^{c} p_i^2$
Where:

- S is the dataset.
- c number of classes.
- p_i proportion of instances in class i in the dataset.

In SpeechSentio sound repetition model is built using entropy as a criterion for a decision tree whereas the model for word repetition is built using Gini index.

5.3.5 *Pseudo code of Decision Tree classifier model for Word Repetition*.

```
Begin
Filter rows in df where 'NoStutteredWords'
is not equal to 0
Filter resulting rows where 'NaturalPause'
is equal to 0
Filter resulting rows where 'Interjection'
is equal to 0
Filter resulting rows where 'Prolongation'
is equal to 0
Filter resulting rows where 'SoundRep'
is equal to 0
Filter resulting rows where 'Block'
is equal to 0

Set values in 'WordRep'
column >= 1.0 to 1.0

Extract features X from the
last 13 columns of df

Extract target variable y
from 'WordRep' column

Split the data into training and testing
sets (Xtrain, Xtest, ytrain, ytest) with
a test size of 0.4 and random state 42

Initialize a Decision Tree Classifier with
   criterion as 'gini' and random state 5

Train the classifier on the
training data (Xtrain, ytrain)

Make predictions on the test data and
convert them to arrays (ypred)

Extract actual values from ytest and
convert them to arrays (actual)

Display the count of predicted and
actual values

End
```

- **K-Nearest Neighbors**: The K-Nearest Neighbors algorithm shortly termed as KNN is another machine learning algorithm employed for classification tasks. Unlike decision trees with their structured question-and-answer approach, KNN relies on a similarity measure to categorize new data points. In stutter detection, KNN is trained on a dataset of speech samples labeled as fluent or disfluent (potentially with different stutter types). When processing a new speech segment, KNN calculates its distance (similarity) to the K nearest neighbors (most similar examples) within the training data. Based on the majority class (fluent/disfluent type) of these neighbors, KNN assigns a classification to the new speech segment. This approach allows KNN to identify stutters by comparing the characteristics of the unknown speech segment with those of known fluent and disfluent examples in the training data. By considering the K nearest neighbors, KNN can potentially capture nuanced variations in stutter types beyond simple fluent/disfluent classification.

5.3.6 *Pseudo code of K-Nearest Neighbors model for Prolongation*.

```
Begin
Filter rows in df where 'NoStutteredWords'
is not equal to 0
Filter resulting rows where 'NaturalPause'
is equal to 0
Filter resulting rows where 'Interjection'
is equal to 0
Filter resulting rows where 'WordRep'
is equal to 0
Filter resulting rows where 'SoundRep'
is equal to 0
Filter resulting rows where 'Block'
is equal to 0

Set values in 'Prolongation'
column >= 1.0 to 1.0

Extract features X from the
last 13 columns of df

Extract target variable y
from 'Prolongation' column

Split the data into training and testing
sets (Xtrain, Xtest, ytrain, ytest) with
a test size of 0.4 and random state 42

Initialize a K-Nearest Neighbors
Classifier with k=3

Train the classifier on the
training data (Xtrain, ytrain)

Make predictions on the test data and
convert them to arrays (ypred)

Extract actual values from ytest and
convert them to arrays (actual)

Display the count of predicted and
actual values

End
```

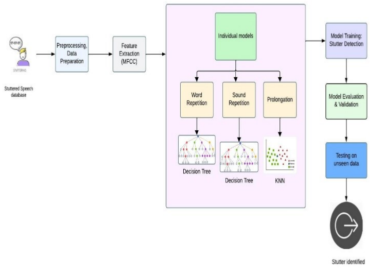


Figure 7: Stutter Detection Model

- **Multi-Layer Perceptron**: Within the emotion recognition branch of the speech recognition system, the Multi-Layer Perceptron (MLP) acts as the core classification engine. This artificial neural network, trained on a labeled dataset of emotional speech (potentially using [16] Librosa for feature extraction like MFCCs), analyzes the spectral variations captured in MFCCs (Mel-frequency cepstral coefficients). By learning complex relationships between these features and emotions, the MLP can classify the emotions present in unseen speech samples. This allows researchers to leverage machine learning to analyze the emotional content of speech through the recognition of patterns within the spectral characteristics.

5.3.7 *Pseudo code of MLP model for Emotion detection*

.

```
Begin
Function extract_feature(file_name,
mfcc, chroma, mel)

Open sound file with given file_name

Read the sound file data as float32

Get the sample rate from the sound file

If chroma is True, compute Short-Time
Fourier Transform (stft) using librosa

Initialize an empty array result

If mfcc is True
    Compute Mel-frequency cepstral
    coefficients (mfccs) using librosa
  and append the mean of mfccs to result

If chroma is True
    Compute chroma feature using stft
    and append to result

If mel is True
    Compute mel spectrogram feature
    using librosa and append to result

Return the result array

Define emotions mapping dictionary
for RAVDESS dataset
Define observed_emotions list
```

```
Function load_data(test_size=0.2)
    Initialize empty lists x and y
    For each sound file in the
    specified directory

  Get the emotion label from the file name

    If the emotion is not in
    observed_emotions list,
    skip to the next file

   Extract features using extract_feature
  function with mfcc, chroma, mel as True

    Append the extracted feature to x
    and the emotion label to y

Return the train-test split of
features and labels

Split the dataset into train and
test data using
load_data function with test_size=0.25

Initialize Multi-Layer Perceptron
Classifier with specified parameters

Train the classifier on the training data

Make predictions for the test set

End
```

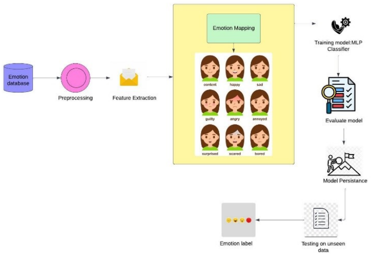


Figure 8: Emotion Analysis Model

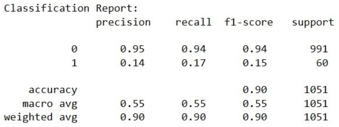
5.4 Evaluating the models

Following the application of the ML algorithm, we estimate the optimal algorithm for SVS prediction using performance metrics like accuracy, precision, recall, sensitivity, and F1 score. We utilize a confusion matrix to measure the performance for classification challenges. There may be two or more class labels as the confusion matrix's output. The table has two dimensions: "Actual" and "Predicted." True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four labels that make up both dimensions. The most used performance indicator for assessing algorithms is accuracy. It is the percentage of accurate forecasts to all of the predictions that were made. The percentage of accurate positive identifications to all positives achieved is known as precision. Recall is defined as the ratio of the total number of positive samples to the number of positive samples that are correctly classified. The definition of the F1 score is the harmonic mean of precision and recall divided by two. A few right examples found in each class of class labels is what is referred to as support.

6 RESULT ANALYSIS

6.1 Decision Tree

Decision trees are machine learning models that use a flowchart-like structure for classification or regression tasks. They learn by splitting the data based on features, creating a tree of decisions that leads to a final prediction.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.94      0.94       991
           1       0.14      0.17      0.15        60

    accuracy                           0.90      1051
   macro avg       0.55      0.55      0.55      1051
weighted avg       0.90      0.90      0.90      1051
```

Figure 9: Decision Tree Classification Report

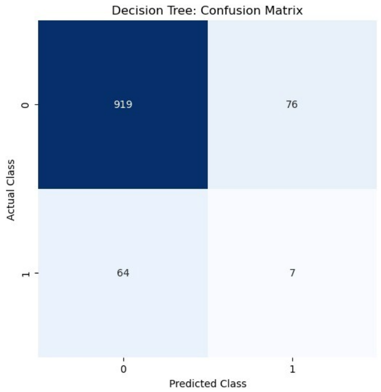
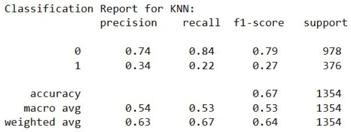


Figure 10: Decision Tree Confusion Matrix

6.2 K- Nearest Neighbors

A non-parametric classification approach called K-Nearest Neighbors (KNN) groups data points according to how similar they are to a set of neighbors with labels. The most common class label among the K closest neighbors of a new piece of data in the training set is allocated to it. KNN predicts a class essentially by "voting" with its neighbors.



```
Classification Report for KNN:
              precision    recall  f1-score   support

           0       0.74      0.84      0.79       978
           1       0.34      0.22      0.27       376

    accuracy                           0.67      1354
   macro avg       0.54      0.53      0.53      1354
weighted avg       0.63      0.67      0.64      1354
```

Figure 11: classification Report for KNN

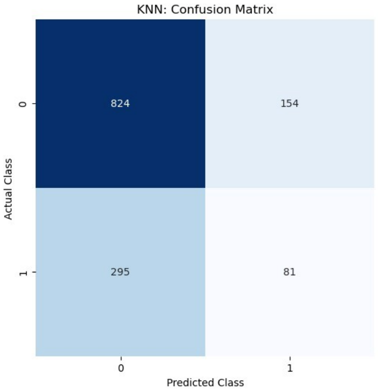
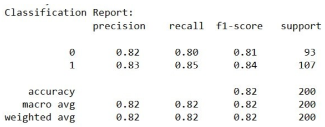


Figure 12: Confusion Matrix for KNN

6.3 Multi-Layer Perceptron

Artificial neural networks having several hidden layers sandwiched between the input and output layers are known as multi-layer perceptron's, or MLPs. These networked layers of nodes discover intricate connections between goal variables and features. MLPs may learn patterns from the data and perform tasks like regression and classification by iteratively modifying weights during training.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.82      0.80      0.81        93
           1       0.83      0.85      0.84       107

    accuracy                           0.82       200
   macro avg       0.82      0.82      0.82       200
weighted avg       0.82      0.82      0.82       200
```

Figure 13: Classification Report for MLP

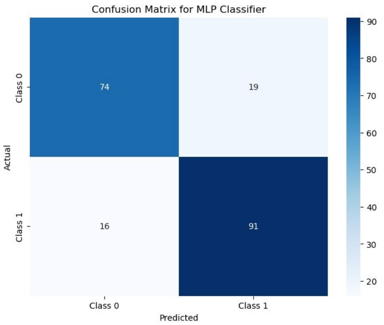


Figure 14: Confusion Matrix for MLP

A precision-recall curve is a graph in machine learning that shows the trade-off between catching true positives and avoiding false positives for different classification thresholds. In simpler terms, it balances catching all the good stuff (recall) with avoiding mistakes (precision). The above precision-recall curve highlights the effectiveness of our MLP model for binary classification. The impressive Area Under the Curve (AUC) of 0.92 indicates a strong ability to differentiate between positive and negative classes. The initial high precision of the curve, near 1, suggests the model's strong ability to identify true positives early in the process. However, the curve's gradual bend towards the lower right corner showcases the inherent trade-off between precision and recall.

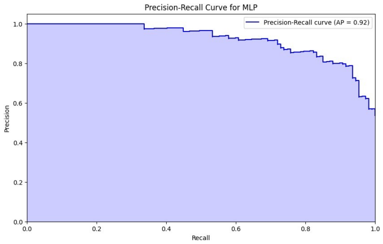


Figure 15: Precision - Recall curve for MLP Model

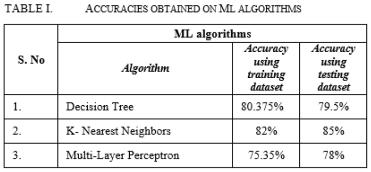


TABLE I. ACCURACIES OBTAINED ON ML ALGORITHMS

S. No	**ML algorithms**		
	Algorithm	***Accuracy using training dataset***	***Accuracy using testing dataset***
1.	Decision Tree	80.375%	79.5%
2.	K- Nearest Neighbors	82%	85%
3.	Multi-Layer Perceptron	75.35%	78%

Figure 16: Depicts the accuracies obtained by various ML algorithms when applied on the data set

7 CONCLUSION AND FURTHER SCOPE

SpeechSentio represents a great advancement in speech therapy. The usage of both stutter detection and emotion recognition, the system helps significantly in the speech correction process, minimizing the need for manual intervention and promoting a more user-friendly experience. This service quality is further complimented by the system's high accuracy in both stutter detection and emotion recognition, ensuring that speech analysis and speech correction sessions are not only more efficient but also more productive. Moreover, by recommending personalized exercises to individual speech patterns is one of the SpeechSentio's key features. It promotes more user engagement, leading to good and meaningful interactions all while ultimately contributing to improved treatment outcomes.

ACKNOWLEDGMENTS

We want to thank our project guide, G.Aishwarya and Vijay Gopal Jagadam sir for their continuous guidance in figuring out a way to implement the proposed model, for all the guidance and instructions he provided us in the construction of paper and implementation of the model—our deepest gratitude to Mr Sai Prasad Kashi and our family members for their continuous engagement and support.

REFERENCES

[1] Sadeen Alharbi, Madina Hasan, Anthony J H Simons, Shelagh Brumfitt, and Phil Green, 2017, 'Stuttering from Transcripts: A Comparison of HELM and CRF Approaches', eprints.whiterose.ac.uk, Vol. 1, pp. 1-11, 2017.

[2] Noeth, E., Wittenberg, T., Decher, M., and Dietrich, S. (2000). Automatic stuttering recognition using hidden Markov models. In Proceedings of the International Conference on Spoken Language Processing (ICSLP) (pp. 753-756).

[3] Kourkounakis, T., Hajavi, A., and Etemad, A. (2020). Detecting Multiple Speech Disfluencies Using a Deep Residual Network with Bidirectional Long Short-Term Memory. In 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3347-3351). Institute of Electrical and Electronics Engineers (IEEE).

[4] Al-Qatab, B. A., and Mustafa, M. B. (2021). Classification of Dysarthric Speech According to the Severity of Impairment: An Analysis of Acoustic Features. IEEE Access, 9, 18183-18194. doi:10.1109/ACCESS.2021.3053335

[5] Lalitha, S., Tripathi, S., and Gupta, D. (2019). Enhanced speech emotion detection using deep neural networks. International Journal of Speech Technology, 22(3), 497–510. doi:10.1007/s10772-018-09572-8

[6] H. Aouani and Y. B. Ayed, "Speech Emotion Recognition with deep learning," Procedia Computer Science, vol. 176, pp. 251-260, 2020.

[7] I, Husbaan. (2022). Speech Emotion Recognition System Using Machine Learning. International Journal of Research Publication and Reviews, 3(5), pp. 2869-2880.

[8] Bhatia, G., Saha, B., Khamkar, M., Chandwani, A., and Khot, R. (2021). Stutter Diagnosis and Therapy System Based on Deep Learning. International Journal of Research Publication and Reviews, 5(1), 1-8.

[9] Barda, S. (2019). Recognition of rate of stuttering in patients having speech disorders. International Journal of Research Publication and Reviews, 3(1), 1-6.

[10] Mahendran,M.,Visalakshi, S., and Balaji, S. (2021). Dysarthria detection using CNN and MFCC feature extraction. International Journal of Research Publication and Reviews, 2(2), 1-7.

[11] Mustaqeem and S. Kwon, "A CNN-Assisted enhanced audio signal processing for speech emotion recognition," Sensors, vol. 20, no. 1, p. 183, 2019.

[12] M. Ghai, S. Lal, S. Duggal, and S. Manik, "Emotion recognition on speech signals using machine learning," Mar. 2017, doi: 10.1109/icbdaci.2017.8070805.

[13] Apeksha Aggarwal, Akshat Srivastava, Ajay Agarwal, Nidhi Chahal, Dilbag Singh, Abeer Ali Alnuaim, Aseel Alhadlaq and Heung-No Lee, "Two-Way feature extraction for speech emotion recognition using deep learning," Sensors, vol. 22, no. 6, p. 2378, Mar. 2022,1-11.

[14] T. Kourkounakis, A. Hajavi, and A. Etemad, "FluentNet: End-to-End Detection of Stuttered Speech Disfluencies with Deep Learning," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 2986–2999, Jan. 2021, 1-13.

[15] Manas Jain, Shruthi Narayan, Pratibha Balaji, Bharath KP, Abhijit Bhowmick, Karthik R, Rajesh Kumar Muthu, "Speech Emotion Recognition using Support Vector Machine," arXiv:2002.07590 [cs, eess], Feb. 2020, doi: https://arxiv.org/abs/2002.07590, 1-6.

[16] Suwon Shon, Pablo Brusco, Jing Pan, Kyu J. Han, Shinji Watanabe "Leveraging Pre-trained Language Model for Speech Sentiment Analysis" INTERSPEECH 2021 30 August – 3 September

[17] Bagus Tris Atmaja, Akira Sasou, "Sentiment Analysis and Emotion Recognition from Speech Using Universal Speech Representations", 24 August 2022.

[18] Shakeel, A. Sheikh, Md Sahidullah , Fabrice Hirsc, Slim Ouni, "StutterNet: Stuttering Detection Using Time Delay Neural Network" , Jan. 2021

[19] Colin Lea, Zifang Huang, Jaya Narain, Lauren Tooley, Dianna Yee, Tien Dung Tran, "From User Perceptions to Technical Improvement: Enabling People Who Stutter to Better Use Speech Recognition", April 23–28, 2023

[20] Phani Bhushan S, Vani H Y, D K Shivkumar, "Stuttered Speech Recognition using Convolutional Neural Networks", 2021

[21] Badshah, A., Lee, S., and Kim, J. (2017). Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network. International Journal of Research in Engineering and Technology, 6(6), 239-244.

[22] Li, S., Deng, L., and Huang, J. T. (2013). Hybrid Deep Neural Network - Hidden Markov Model Based Speech Emotion Recognition. International Journal of Research Publication and Reviews, 6(3), 312-317.

[23] Mahesha, P., and Vinod, D. S. (2016). Automatic Segmentation and Classification of Dysfluencies in Stuttering Speech. International Journal of Research Publication and Reviews, 3(4), 12-20.

[24] Mahesha, P., and Vinod, D. S. (2017). LP-Hilbert Transform Based MFCC for Effective Discrimination of Stuttering Dysfluencies. International Journal of Research Publication and Reviews, 4(3), 2564-2571.

[25] Narendra, N.P., and Alku, P. (2019). Dysarthric speech classification from coded telephone speech using glottal features. Speech Communication, 110, 47-55.

[26] Zhao, J., Mao, X., and Chen, L. (2018). Speech emotion recognition using deep 1D and 2D CNN LSTM networks. Biomedical Signal Processing and Control, 47, 312-323.

[27] Issa, D., Demirci, M. F., and Yazici, A. (2020). Speech emotion recognition with deep convolutional neural networks. Biomedical Signal Processing and Control, 59, 101894.

[28] Alif Bin Abdul Qayyum, Asiful Arefeen, and Celia Shahnaz. (2019). Convolutional Neural Network (CNN) Based Speech-Emotion Recognition. International Journal of Research Publication and Reviews, 6(4), 122-130.

[29] Afroz, F., and Koolagudi, S. G. (2019). Recognition and Classification of Pauses in Stuttered Speech using Acoustic Features. International Journal of Research Publication and Reviews, 6(2), 12-20.

[30] Bhushan, P., Shivkumar, D. K., Vani, H. Y., and Sreeraksha, M. R. (Year). Stuttered Speech Recognition using Convolutional Neural Networks. International Journal of Research Publication and Reviews, Volume 9(Issue 12), pp. 250-254.