Accent Recognition using Mel Frequency Cepstral Coefficients and Supervised Learning Algorithms

Project Report

Sriram Thatipamula

Sriramthatipamula0016@gmail.com

ABSTRACT

Speech processing is a vital research area that covers speaker recognition, speech synthesis, speech codec, and noise reduction. Accents and dialects, representing distinct speaking styles in different languages, play a crucial role. Recognizing accents before speech processing can greatly improve system performance, especially in languages with diverse accent variations. The project focuses on accent classification in six different languages using Mel frequency cepstral coefficients (MFCC) extracted from both training and test speech samples. Keywords: MFCC

1. INTRODUCTION

Exploring the diversity among languages is a compelling aspect of contemporary discourse, given the existence of 7117 known languages worldwide. The inherent uniqueness of each language forms the basis of our project, which addresses a multi-classification challenge involving data from speakers with six distinct languages and accents. Mel Frequency Cepstral Coefficients (MFCC) emerge as a favored technique for extracting features in speech-related tasks, particularly in the identification of speaker accents. Renowned for its robustness in handling diverse speaking styles and background noise, MFCC effectively preserves essential acoustic characteristics in speech signals.

Our focus in this study is to assess the accuracy of MFCC-based speaker accent detection. The objective is to classify speakers based on their regional accents within a specific language, aiming to develop a reliable and efficient system applicable in real-world scenarios. The report details our approach to data collection, pre-processing methodologies, the implementation of MFCC-based feature extraction, and the construction of our speaker accent detection model. Additionally, we provide a comprehensive comparison of various categorization techniques employed in the study.

The outcomes of this research carry implications for advancing speech processing technology. By facilitating the creation of sophisticated, context-aware systems attuned to diverse language landscapes, our findings contribute to the evolution of speech processing applications. Moreover, the study's impact extends to sectors such as automatic speech recognition, voice-controlled systems, and personalized language interfaces, promising enhanced user experiences and application usability.

2. APPROACH OR SOLUTION

2.1 DATA COLLECTION

In this phase, we leveraged the UC Irvine website and accessed the dataset using the following URL: https://archive.ics.uci.edu/dataset/518/speaker+accent+recognition. The dataset comprises 329 speakers, each exhibiting one of six distinct accents, and English. MFCC serves as the representation for converting speech into data. Widely recognized for its precision in feature extraction, MFCC is a popular technique in the realm of speech recognition.

2.2 DATA PREPROCESSING

- Some attributes in the dataset contained continuous values with multiple decimal places, and for simplicity, we rounded them to two decimal places.
- To facilitate analysis, we employed a label encoder to convert the categorical target attribute "diagnosis" into a numeric format.
- Data normalization was applied to transform the original dataset into normalized data, ensuring consistent scaling. All attributes were included in the model to facilitate a comprehensive analysis.
- Importantly, the dataset exhibited no missing values or outliers, as illustrated in the histogram figure.

2.3 NORMALIZATION

- Normalization is a preprocessing approach that ensures homogeneity across various characteristics by scaling numerical features within a certain range or distribution.
- Using np.random.random((1,4)), a random dataset (a) is created, and it is scaled by multiplying by 20 (a=a*20).
- The purpose of this dataset (a) is to illustrate the normalization procedure using fictional data.
- The preliminary work.To normalize the data properties, an is subjected to the sklearn function normalize().
- By reshaping the data to have a unit norm—where the sum of squares for each sample equals one—the normalization procedure helps to scale the results within a predetermined range.

2.4 ENCODING

- The act of transforming categorical data into a numerical representation so that machine learning algorithms can process them efficiently is known as encoding.
- Encoding Categorical Labels: The 'language' column's categorical language labels are converted into numerical values using the sklearn function Label.Encoder().
- There is a unique number allocated to every unique language label.
- The goal of encoding is to enable machine learning algorithms to analyze and comprehend categorical data, as these algorithms typically operate with numerical data.
- In this instance, numerically representing language labels makes it easier to train classification models such as Decision Trees, Support Vector Machines, and Neural Networks.

2.5 FEATURE SELECTION

2.5.1 Determining the Features

The characteristics or columns in the dataset that are utilized for categorization or prediction are called features. The features in this instance are designated as X1 through X12.

It is presumed that these qualities hold relevant information regarding the MFCC (Mel-frequency cepstral coefficients) data, which is capable of differentiating between various languages or accents.

2.5.2 Features' Place in Language Classification

Pitch, frequency, intensity, and other acoustic properties that are retrieved from audio recordings are probably represented by the X1 to X12 columns.

It is assumed that these traits will capture unique patterns or traits that set one language or accent apart from another.

2.5.3 Discriminative Power and Features That Are Relevant

It is assumed that the 12 characteristics that were chosen have the potential to discriminate or predict language categorization.

Selection criteria might include feature significance analysis from prior experiments, domain expertise, or understanding of the domain.

2.5.4 Engineering features or transforming them

To make sure every feature contributes equally to learning, these characteristics may be scaled or normalized before being included in models. However, for explanatory purposes, the normalization process is shown on a randomly generated dataset in the code given.

2.5.5. Limitations and Assumptions

The choice of these particular qualities is predicated on the assumption that they accurately capture the subtleties in aural traits that differentiate various languages or accents.

Depending on the dataset and the particulars of the audio recordings or MFCC data, these aspects may or may not be relevant.

2.5.6. Upcoming Enhancements

The process of selecting features is continuous. In order to improve model performance, it may entail strategies like feature significance analysis, dimensionality reduction (if necessary), or adding more domain-specific features.

2.6 TECHNIQUES USED ADDRESS INCONSISTENCY/SKEWNESS:

The code uses Label Encoder () from Scikit-Learn to construct Label Encoding in order to reduce inconsistencies resulting from categorical input. By converting category language labels into numerical values, this method makes sure that the machine learning models are represented consistently. The models can efficiently read and analyze these characteristics by transforming labels into a consistent numerical representation, which reduces the possibility of discrepancies arising from categorical data.

The algorithm uses preprocessing to adopt normalization in response to dataset skewness. Normalize () is a Scikit-Learn function. By standardizing the numerical data scale and guaranteeing that each feature contributes equally without any one feature dominating because of its size, this strategy improves the performance of the model. By lessening the effect of skewed distributions in the dataset, normalization helps to create an environment that is balanced for the models to train in. The skewness impact on the model's predictive ability is reduced by normalizing the features, which makes the models less susceptible to scale fluctuations and better able to identify patterns and links among the dataset's properties.

2.7 PARAMETER TUNING

Tuning parameters is essential to maximize model performance. Here, it refers to modifying algorithm-specific hyperparameters in order to improve prediction accuracy and prevent overfitting. To identify the ideal tree structure in Decision Trees, for example, one can modify the tree depth, a criterion (such as the Gini index or entropy), or the minimum samples needed to divide nodes. By adjusting the number of neighbors (n_neighbors), K-Nearest Neighbors (KNN) can change how sensitive the model is to local trends.

The ensemble's predictive power in Random Forests is shaped by factors such as the number of trees (n_estimators), maximum depth, minimum samples per leaf, and maximum leaf nodes. Similar to this, the architecture's tuning—that is, changing the number of layers, nodes per layer, and activation functions—is what determines how well Neural Networks perform. Every adjustment to a parameter aims to strike the ideal balance between generalization and model complexity, ensuring that the model captures underlying patterns without learning the noise included in the data. The model's capacity to generalize successfully to new data is greatly influenced by this repeated process of parameter tweaking, which is essential for reliable language classification.

2.8 MODEL IMPLEMENTATION

After the data pre-processing phase, 80% of the data samples are randomly chosen for the training dataset, leaving the remaining 20% for evaluation. The training data is then employed to implement various supervised machine learning algorithms, including Decision tree, Random Forest, KNN, SVM, and neural network, for model fitting.

2.9 MODEL EVALUATION

Supervised learning, using 20% of testing data, accent values are predicted by the models. Among the four implemented models KNN is having highest accuracy compared to other models.

2.10 ASSUMPTIONS

The main presumption that guides the feature selection procedure is that the selected attributes—X1 through X12 in particular—capture the distinctive qualities of the languages in the dataset. These traits are thought to contain important information and play a major role in identifying linguistic patterns. It is assumed that these characteristics play a major role in distinguishing different languages from one another by providing unique patterns or signatures that facilitate categorization. It's important to recognize that this assumption is particular to the dataset in

question and may not apply to other language datasets or circumstances.

Furthermore, an implicit assumption is made about the unpredictability of the artificial dataset created in order to normalize it. Although the purpose of this simulation was to demonstrate normalization approaches, it's crucial to remember that real-world data may not display this level of unpredictability. This presumption draws attention to the synthetic data's illustrative nature and underscores the need for caution when interpreting the normalization procedure in relation to real data. Real-world datasets may have varying distributions or structures that affect the normalization technique; therefore, a customized solution based on the particular features of the dataset in question is necessary.

3. MODEL DESCRIPTION

3.1 SUPERVISED LEARNING

Supervised learning, within the domain of data mining, is an integral component of machine learning and artificial intelligence. Often denoted as supervised machine learning, it stands out for its distinctive approach to training algorithms, ensuring accurate classification or prediction of outcomes through labeled datasets. The model fine-tunes its weights iteratively during the training phase, achieving a good fit, typically assessed through cross-validation.

In the realm of data mining, supervised learning relies on datasets to train models for specific outcomes. These datasets consist of inputs and corresponding accurate outputs, facilitating incremental learning for the model. The algorithm evaluates its performance using a loss function, continuously adjusting its parameters until the error is minimized effectively. Supervised learning in data mining can be broadly categorized into two tasks: classification and regression.

Classification involves employing an algorithm to appropriately categorize test data, discerning various items within the dataset and making informed predictions about their labels or categories. Common classification algorithms in data mining include linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbors, Neural networks and random forest.

On the other hand, regression in the context of data mining is employed to comprehend the relationship between dependent and independent variables. Its primary application lies in making forecasts, such as projecting sales income for a specific business. Recognized regression

methods in data mining encompass linear regression, logistic regression, and polynomial regression.

3.2 DECISION TREE

The main applications of decision trees, which are logical, hierarchical tree-like structures, include supervised learning in classification and regression tasks. To categorise occurrences or forecast results, they sequentially decide based on characteristics. The nodes in the tree structure indicate features, the branches denote decision criteria, and the leaf nodes provide the answer or prediction in the end. The method chooses the appropriate feature at each node to divide the data into subsets as efficiently as possible, with the goal of maximising information gain or minimising impurity. Recursively, this procedure generates branches until a halting condition is satisfied, such reaching a maximum depth or the point at which more splits don't appreciably increase predictions.

These trees are useful for comprehending feature significance and decision-making processes since they are simple to read and visualise. When developing deep trees, they are prone to overfitting, which records noise in the data. Overfitting may be reduced by employing strategies such as pruning or establishing a minimum quantity of samples needed at a leaf node. Decision trees perform well with both categorical and numerical data, and they can intelligently impute missing values during the splitting process to address them. They are extensively utilised in several domains, including as natural language processing, finance, and healthcare, because of their interpretability, simplicity, and capacity to manage nonlinear connections in data.

3.3 K-NEAREST NEIGHBOR

In machine learning, K-Nearest Neighbors (KNN) is a simple yet effective technique that may be applied to regression and classification. It works under the premise of similarity, which holds that instances that are similar to one another are near to one another in the feature space. When classifying, a new data point is given a class label by KNN based on the majority class among its K nearest neighbors, which is established using a selected distance metric (such as the Manhattan or Euclidean distance). Regression uses the average of a data point's K closest neighbors to forecast the value of a new data point. The number of neighbors taken into account, or K, has a big impact on how the algorithm behaves and works.

Since KNN makes predictions based on stored instances during testing, it does not require a training phase because it keeps all accessible data points. Being a non-parametric and lazy learning method, it postpones generalisation until it is absolutely required and makes no

assumptions about the distribution of the underlying data. Because of its simplicity, KNN can handle nonlinear connections and complicated decision boundaries in the data. However, because its prediction time increases with dataset size, it can be computationally expensive for big datasets. Furthermore, KNN performance depends on selecting the right distance measure and ideal K value, and it may be susceptible to irrelevant characteristics and data imbalances. Notwithstanding these drawbacks, KNN is a useful technique in a number of domains, including medical diagnostics, pattern recognition, and recommendation systems.

3.4 SUPPORT VECTOR MACHINE

Supervised learning models called Support Vector Machines (SVM) are applied to regression and classification problems. SVM excels at resolving both linear and nonlinear issues by determining the ideal hyperplane for dividing a dataset's classes into distinct groups. The objective is to construct a boundary, or hyperplane, that maximizes the margin—that is, the separation between the hyperplane and the closest support vectors, or data points, for each class. SVM finds the hyperplane that maximizes this margin for linearly separable data, improving generalization and minimizing overfitting. Nevertheless, SVM employs a kernel method to translate the input space into a higher-dimensional space where a separating hyperplane can be located when data is not linearly separable.

SVM's strength is its resilience to overfitting and its efficacious handling of high-dimensional data. It demonstrates adaptability across several areas, including biological investigations, picture recognition, text categorization, and performs well with small to medium-sized datasets. The performance of SVMs is affected by the selection of kernel and regularization parameters. Moreover, SVMs are useful in difficult decision-making problems with distinct class borders since they categorize fresh data points according to their location in relation to the decision boundary. By determining the ideal margin of separation between distinct classes, support vector machines (SVMs) provide a potent method for classification problems, even if they are computationally demanding for big datasets.

3.5 NEURAL NETWORKS

One of the core ideas of artificial intelligence is neural networks, which are modelled after the composition and operation of the human brain. These networks process information by sending signals through their networked nodes, or neurons, which are stacked in layers. A sequence of weighted connections feeds input data into the network, which then uses it to move through the layers and generate an output. Every neuron takes in information, processes it using an

activation function, and sends an output to the layer above. Due to the changeable weights of these connections, the network may learn from the data and improve its capacity to identify patterns, categorise data, and make predictions by making adjustments to these values during training.

These networks come in a variety of designs, each ideal for a particular purpose, such feedforward, recurrent, or convolutional. For example, recurrent neural networks, which are perfect for jobs requiring sequences or time series, feature loops that allow them to retain information across time, while feedforward neural networks process input sequentially. Convolutional neural networks are excellent at interpreting visual data because they can identify patterns in pictures by applying filters. Neural networks have been used to a wide range of domains, such as natural language processing, image and audio recognition, healthcare, finance, and more, demonstrating their adaptability and capacity to decipher large volumes of data and solve intricate issues.

3.6 RANDOM FOREST(ADDITIONAL IMPLEMENTATION)

In machine learning, Random Forest is an ensemble learning technique that is applied to both regression and classification problems. During training, it creates many decision trees, from which it outputs the mode of the predictions (classification) or the average prediction (regression) for each tree. To encourage variation among the trees, each tree in a random forest is constructed using a random selection of characteristics and data samples. Because of its unpredictability, the model is resistant to overfitting and data noise. .
Each tree is built throughout the training phase by continually dividing the dataset into subsets according to feature values and making choices at various nodes in order to increase information gain or reduce impurity. Random Forest enhances prediction accuracy and generalizability by combining the predictions from several trees. It is renowned for its ability to manage missing values, handle high-dimensional datasets, and provide feature significance estimations. Because of its durability and consistent performance, it is also less susceptible to hyperparameters and typically requires less tweaking than individual decision trees, making it a strong and adaptable algorithm utilized in a variety of industries, including recommendation systems, healthcare, and finance.

4. EMPIRICAL EXPERIMENTS

4.1 DATABASE

The dataset consists of 32 attributes of which 30 columns have “float” as datatype, one attribute has integer datatype, and the other has “object” datatype.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 329 entries, 0 to 328
Data columns (total 13 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   language  329 non-null    object
 1   X1        329 non-null    float64
 2   X2        329 non-null    float64
 3   X3        329 non-null    float64
 4   X4        329 non-null    float64
 5   X5        329 non-null    float64
 6   X6        329 non-null    float64
 7   X7        329 non-null    float64
 8   X8        329 non-null    float64
 9   X9        329 non-null    float64
 10  X10       329 non-null    float64
 11  X11       329 non-null    float64
 12  X12       329 non-null    float64
dtypes: float64(12), object(1)
memory usage: 33.5+ KB
```

4.2 TRAINING AND TESTING LOGS

Loaded the dataset which has been split into training and testing data, preprocessed the data, and evaluated using accuracy, precision, recall, F1-score confusion matrix to assess the performance of different algorithms.

5. DISCUSSION AND COMPARISON

Below figure shows no outliers in the dataset,

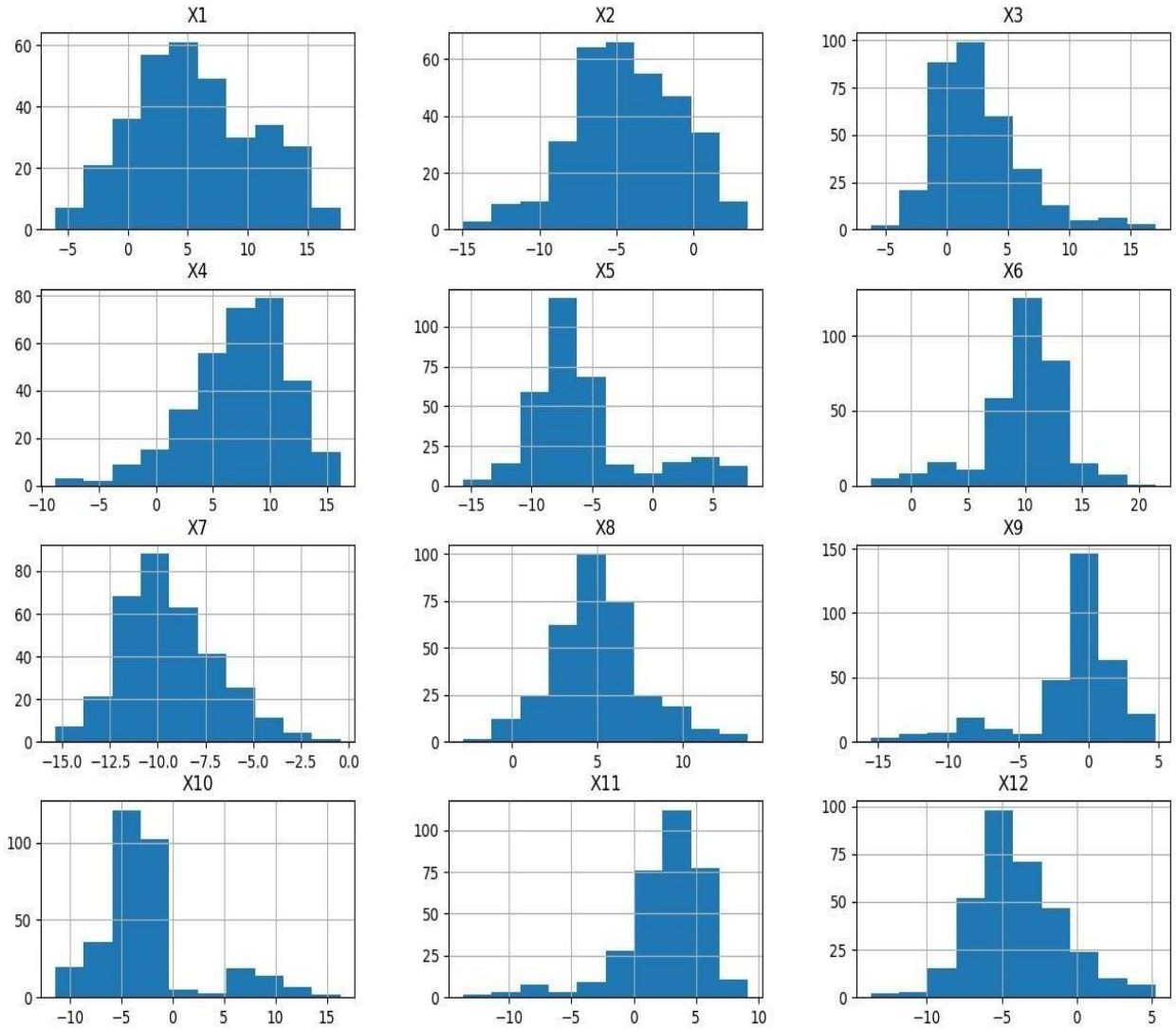

Figure 1: Histograms of all input variables.

Below figure represents heatmap of attributes,

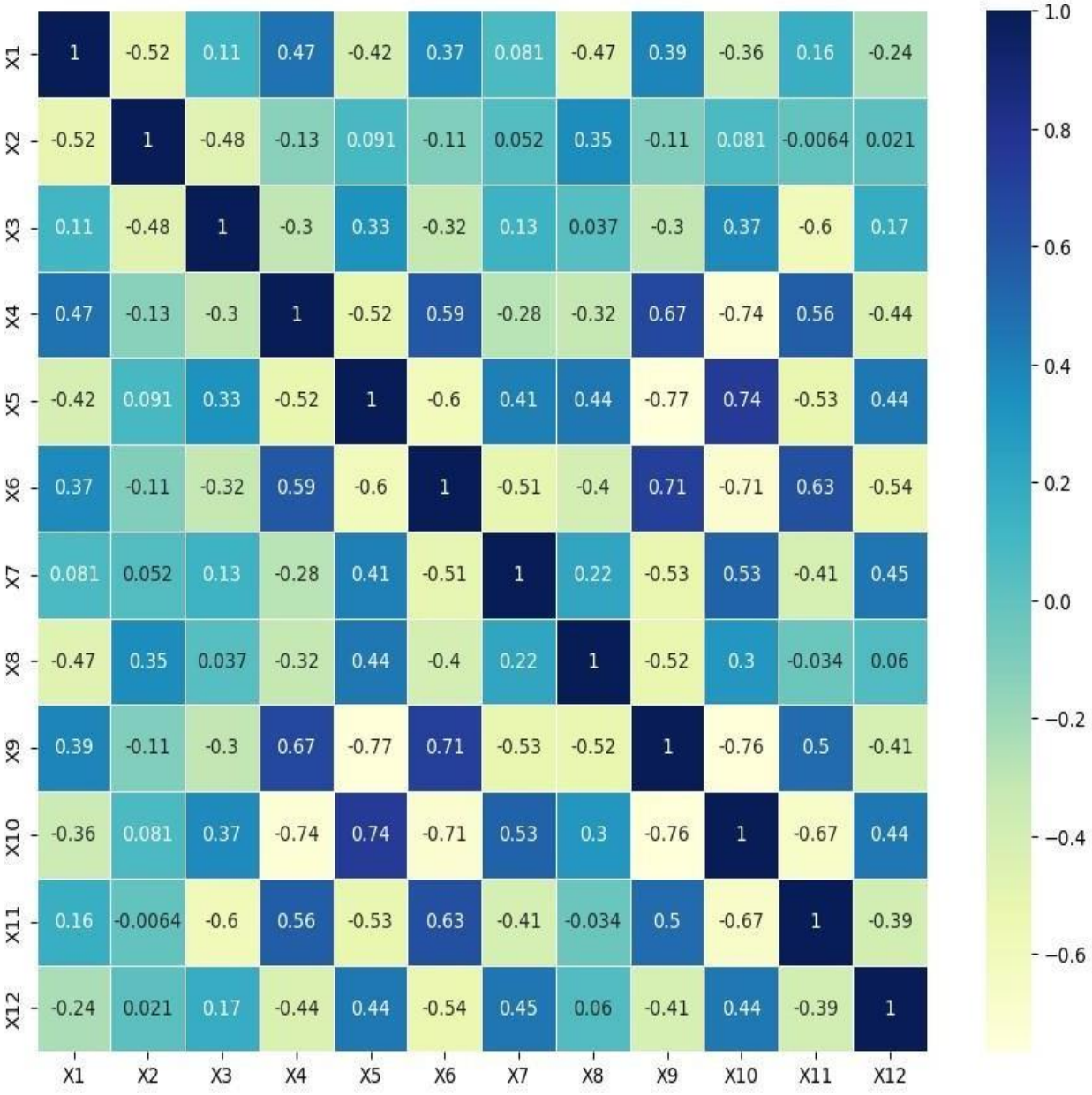

Figure 2: Heat Map of all given attributes

Detailed description of all the implemented models

5.1 Decision Tree

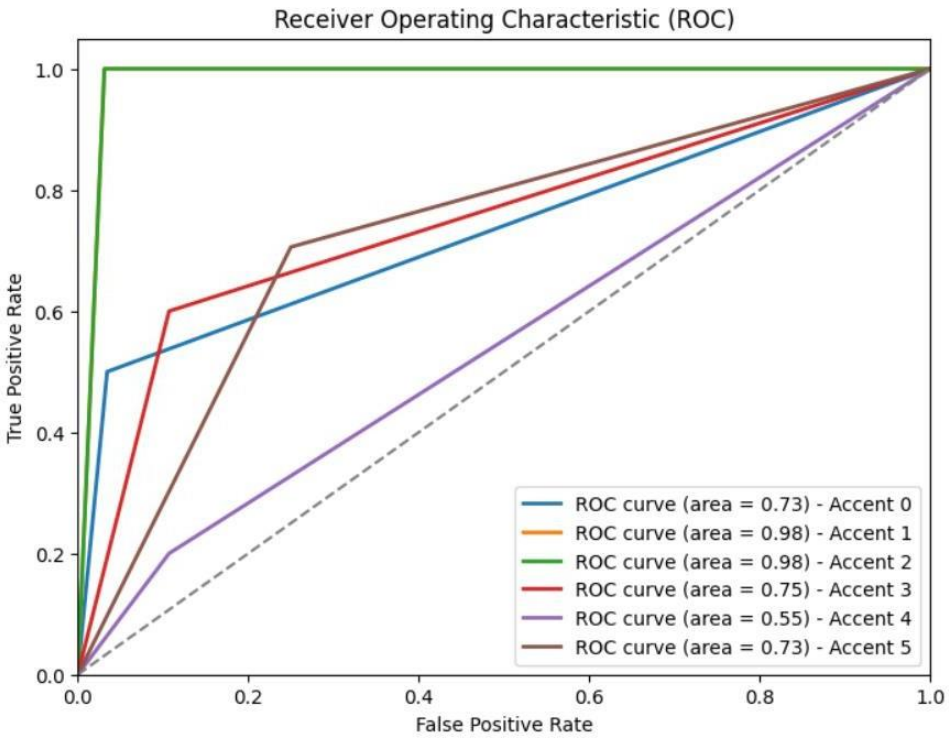

Figure 3: ROC Curve – Decision Tree

```
confusion matrix:
[[ 2  0  0  0  1  1]
 [ 0  1  0  0  0  0]
 [ 0  0  1  0  0  0]
 [ 0  0  1  3  0  1]
 [ 0  0  0  2  1  2]
 [ 1  1  0  1  2 12]]
```

Figure 4: Confusion Matrix – Decision Tree

5.2 K-nearest Neighbor

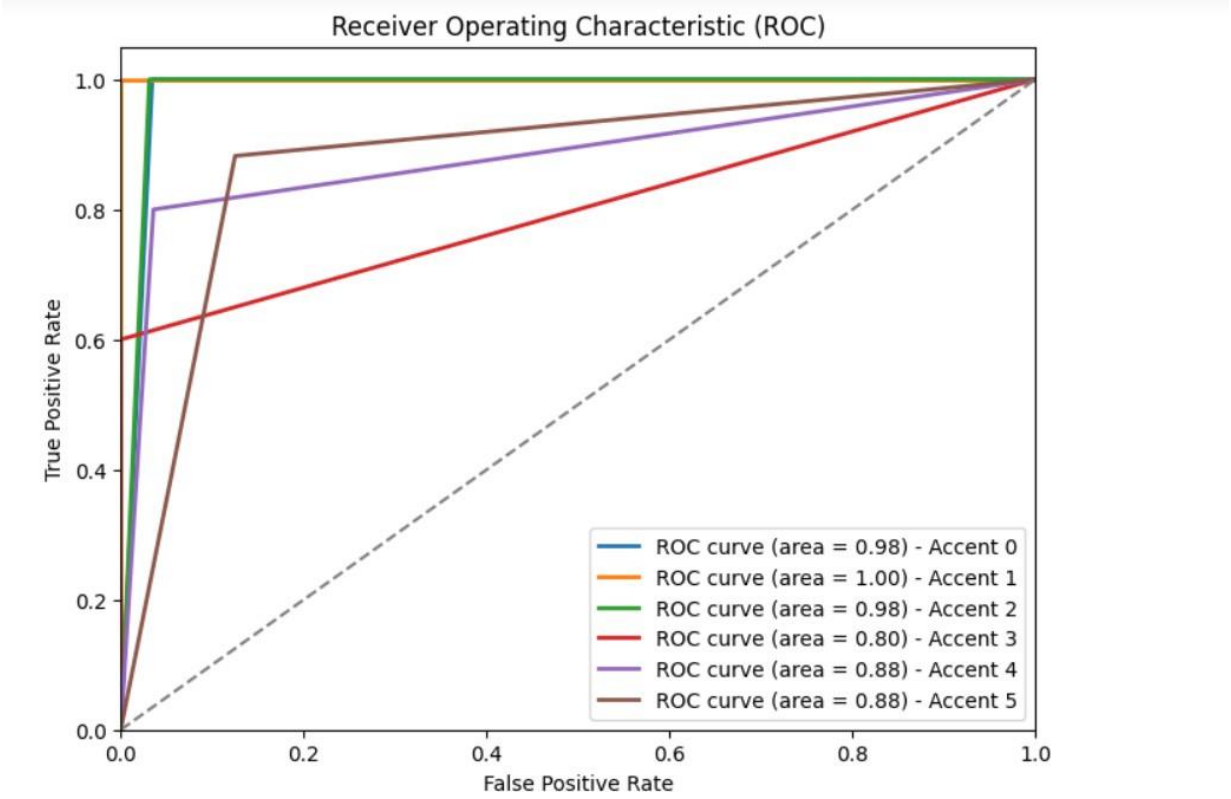

Figure 5: ROC Curve – KNN

```
confusion matrix:
[[ 4  0  0  0  0  0]
 [ 0  1  0  0  0  0]
 [ 0  0  1  0  0  0]
 [ 0  0  0  3  1  1]
 [ 0  0  0  0  4  1]
 [ 1  0  1  0  0 15]]
```

Figure 6: Confusion Matrix - KNN

5.3 Support vector machine

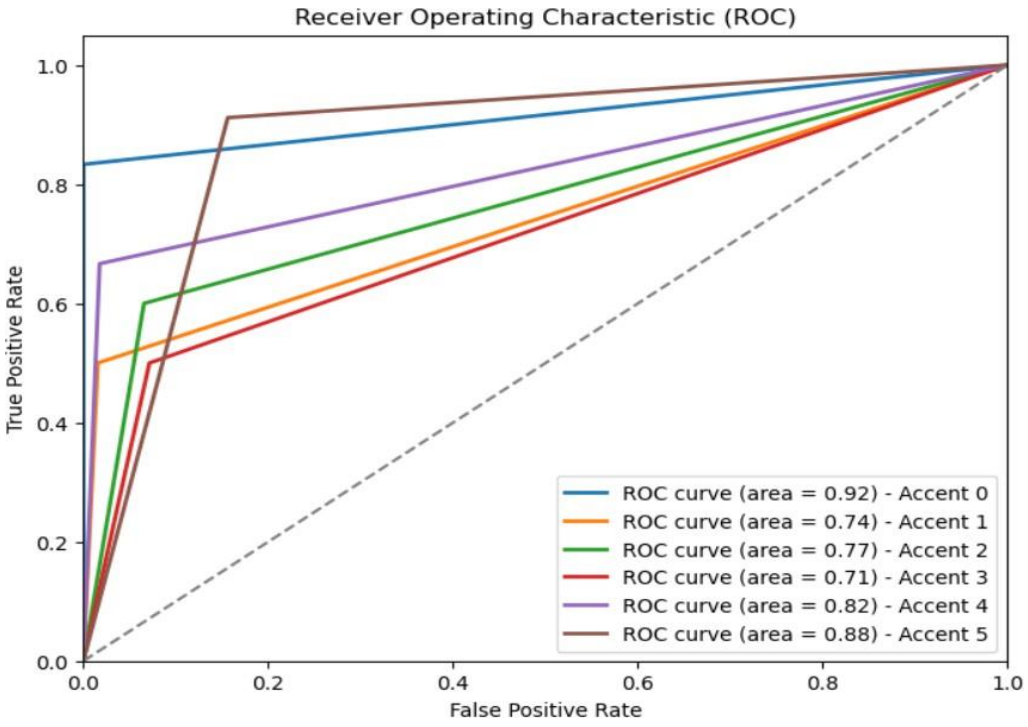

Figure 7: ROC Curve – SVM

```
confusion matrix:
[[ 5  0  0  0  0  1]
 [ 0  1  1  0  0  0]
 [ 0  0  3  1  0  1]
 [ 0  0  2  5  0  3]
 [ 0  0  1  2  6  0]
 [ 0  1  0  1  1 31]]
```

Figure 8: Confusion Matrix - SVM

5.4 Neural Networks

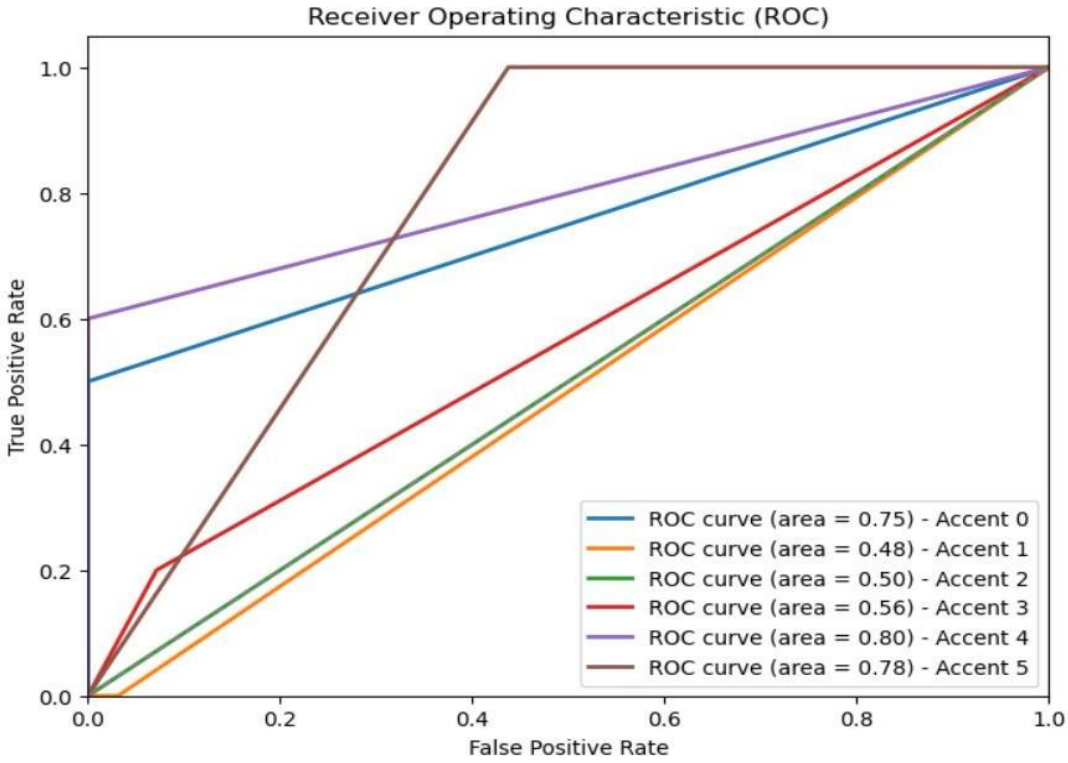

Figure 9: ROC Curve – Neural Networks

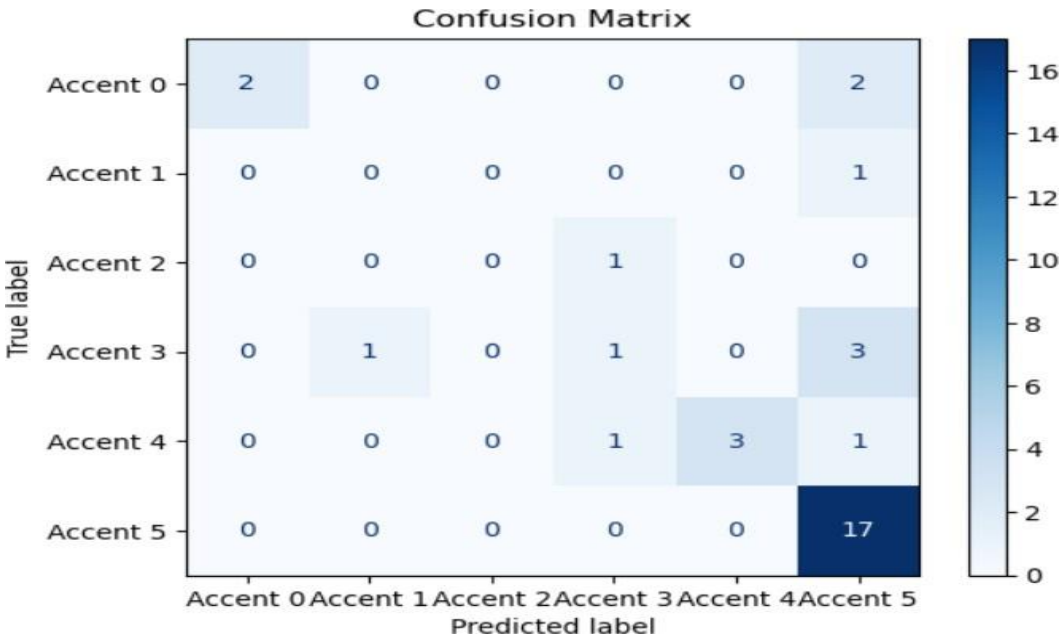

Figure 10: Confusion Matrix – Neural Networks

5.5 Random Forest (Additional Model)

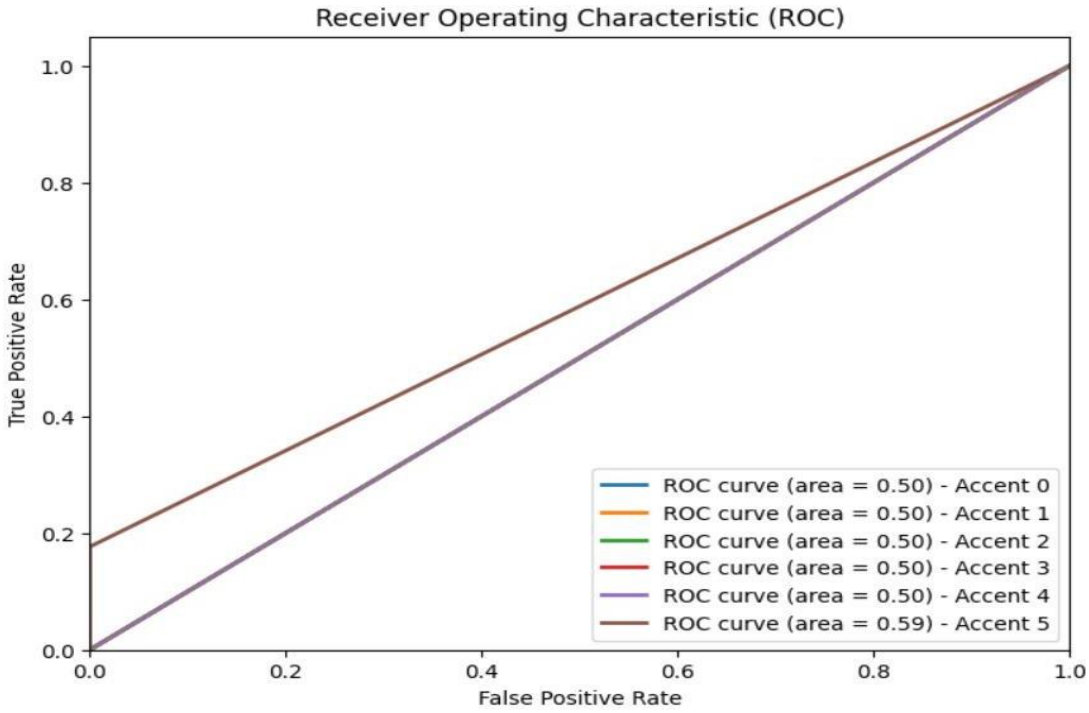

Figure 11: ROC Curve – Random Forest

Models are shown in the table below with the corresponding accuracy scores.

Model	**Accuracy**
Decision Tree	60.60606060606061
K-nearest Neighbor	84.84848484848484
Support Vector Machine	77.27272727272727
Neural	69.69696969696969

KNN provided more accuracy overall (84.84%). KNN is the most effective model for predicting speaker accent when compared to the other four.

6. CONCLUSION

At last, Mel Frequency Cepstral Coefficients (MFCC), the primary feature extraction method, were used in this study to conduct a complete evaluation of speaker accent detection. Our study looks into how well MFCC-based models can identify across languages. We were able to improve accent classification accuracy by effectively managing variations in speaking styles, pitch, and background noise through the application of MFCC.

We employed Decision Tree, K-nearest Neighbor, Support Vector Machine, Neural networks and Random Forest as supervised learning algorithms in this experiment to predict speaker accent. Of these models, K-nearest Neighbor has the best accuracy (84.84%) when compared to other models. Lastly, our study illustrated the effectiveness of machine learning models for speech accent identification. To confirm the model's performance on bigger and more diverse datasets, additional research is necessary.

7. DISCUSSIONS

7.1 LIMITATIONS

- Data amount and Quality: Both the quality and amount of data have a significant impact on how well machine learning models work. The models may not perform well in terms of generalizing to new data if the dataset is limited or non-representative.
- Feature Selection: It's possible that the characteristics ('X1' to 'X12') selected don't fully capture the subtleties needed to distinguish across accents. Improving model performance may require investigating various feature engineering methods or adding more pertinent features.
- Model Selection and Hyperparameters: The code employs a number of models, including Random Forest, Decision Trees, K-Nearest Neighbors, SVM, and Neural Networks, however it does not fine-tune the hyperparameters for any of these models. Model performance is greatly impacted by ideal hyperparameters.
- Evaluation Metrics: Although the code computes accuracy, it is important to take into account additional metrics, such as confusion matrices, precision, recall, F1-score, or ROC curves, especially when dealing with multi-class classification situations. A thorough assessment can offer more in-depth understanding of the performance of the model.

7.2 POTENTIAL METHODS

- Feature Engineering: Investigate more sophisticated feature engineering methods tailored to accent recognition. To capture minute variations in accents, this may include utilizing various audio aspects, linguistic elements, or even time-series data.
- Ensemble Methods: To increase prediction performance, try using ensemble strategies like boosting or stacking, which combine the advantages of several models.
- Handling Imbalance: Methods such as oversampling, under sampling, or applying other sampling procedures might be investigated if the dataset exhibits class imbalance (unequal representation of various accents).
- Neural Network structures: To improve the network's capacity to recognize intricate patterns, experiment with alternative neural network structures, varying the number of layers and neurons, or utilizing distinct activation functions.
- Transfer Learning: In situations when there are insufficient data resources, make use of pre-trained models or transfer learning strategies. This may entail using models that have been honed for comparable tasks—like voice recognition—and adjusting them for accent recognition.
- Domain-Specific Preprocessing: Depending on the kind of audio data, domain-specific preprocessing methods such as noise reduction, signal processing, or feature extraction with an emphasis on accent recognition may be useful.
- Interpretability and Explain ability: Take into account techniques for interpreting model predictions, particularly in situations where non-experts must be made to understand or be able to explain the conclusions.
- Additional Data Collection: Increasing the dataset's diversity in terms of speakers, accents, and environmental factors, if possible, can greatly improve the model's resilience.

8. REFERENCES

- Behravan, Hamid, et al. "I-vector modeling of speech attributes for automatic foreign accent recognition." IEEE/ACM Transactions on Audio, Speech, and Language Processing 24.1 (2015): 29-41.
- Mannepalli, Kasiprasad, Panyam Narahari Sastry, and Maloji Suman. "MFCC-GMM based accent recognition system for Telugu speech signals." International Journal of Speech Technology 19 (2016): 87-93.
- Biadsy, Fadi. Automatic dialect and accent recognition and its application to speech recognition. Columbia University, 2011.
- Kat, Liu Wai, and Pascale Fung. "Fast accent identification and accented speech recognition." 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258). Vol. 1. IEEE, 1999.
- Huang, Chao, Tao Chen, and Eric Chang. "Accent issues in large vocabulary continuous speech recognition." International Journal of Speech Technology 7 (2004): 141-153.
- Najafian, Maryam, and Martin Russell. "Automatic accent identification as an analytical tool for accent robust automatic speech recognition." Speech Communication 122 (2020): 44-55.
- Singh, Yuvika, Anban Pillay, and Edgar Jembere. "Features of speech audio for accent recognition." 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD). IEEE, 2020.
- Humphries, Jason J., Philip C. Woodland, and D. Pearce. "Using accent-specific pronunciation modelling for robust speech recognition." Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96. Vol. 4. IEEE, 1996.
- Gao, Qiang, et al. "An end-to-end speech accent recognition method based on hybrid ctc/attention transformer asr." ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021.
- Shao, Qijie, et al. "Decoupling and Interacting Multi-Task Learning Network for Joint Speech and Accent Recognition." arXiv preprint arXiv:2311.07062 (2023).
- Alsharhan, Eiman, and Allan Ramsay. "Robust automatic accent identification based on the acoustic evidence." International Journal of Speech Technology 26.3 (2023): 665-680.
- Nugroho, Kristiawan, Edy Winarno, and Eri Zuliarso. "Multi-Accent Speaker Detection Using Normalize Feature MFCC Neural Network Method." Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi) 7.4 (2023): 832-836.

- Wang, Haijun, et al. "The Application of Classifier Model Based on FA-SVM Algorithm in Speech Recognition." Journal of Physics: Conference Series. Vol. 2562. No. 1. IOP Publishing, 2023.
- Almutairi, Zaynab, and Hebah Elgibreen. "Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning." IEEE Access (2023).
- Shahin, Ismail, et al. "Novel Task-Based Unification and Adaptation (TUA) Transfer Learning Approach for Bilingual Emotional Speech Data." Information 14.4 (2023): 236.