Speech Emotion Recognition

Aditya Baranwal (B20EE001) Abhishek Rajora (B20CS002) Nakul Sharma (B20AI024)
Website: emotion-detection-dep, Github Repository: SER

Abstract:

This paper reports our methods, procedures and suggests further possible extensions in building a speech emotion recognizer. Exploring this relatively nascent problem, we explore several deep-learning-based approaches along with classical machine learning algorithms for an almost exhaustive analysis of the topic. In doing so, we utilize the Ravdess dataset and perform extensive analysis of various algorithms.

I. Introduction

Speech emotion recognition is an important problem receiving increasing interest from researchers due to its numerous applications, such as audio surveillance, E-learning, clinical studies, detection of lies, entertainment, computer games, and call centers. Nevertheless, this problem still remains a significantly challenging task for advanced machine learning techniques. One of the reasons for such a moderate performance is the uncertainty of choosing the right feature.
In this work, we explored various Speech emotion recognition classification algorithms on the dataset. We have the Ravdess dataset which contains 24 professional actors (12 female, 12 male), vocalizing two lexically-matched statements in a neutral North American accent. Speech emotions include calm, happy, sad, angry, fearful, surprised, and disgusted expressions. Each expression is produced at two levels of emotional intensity (normal, strong), with an additional neutral expression.
We deployed the Shallow MLP model at emotion-detection-dep. A reference at the bottom at the page.

Dataset Used :

Ravdess: A total of 1440 audio files were used as the dataset.
The training dataset contains 1440 files where each file represents a type of emotion associated to a particular actor:

- The filename consists of a 7-part numerical identifier
- These identifiers define the stimulus characteristics

The dataset has been split into train and test with a test size of 0.3

Speech dataset: The Ravdess dataset is an emotion type system that divides everyone into 8 distinct emotion with each expression produced at two levels of emotional intensity:

- Normal - Strong
- Neutral expression don’t have a strong level of intensity
- Two types of the sentence.

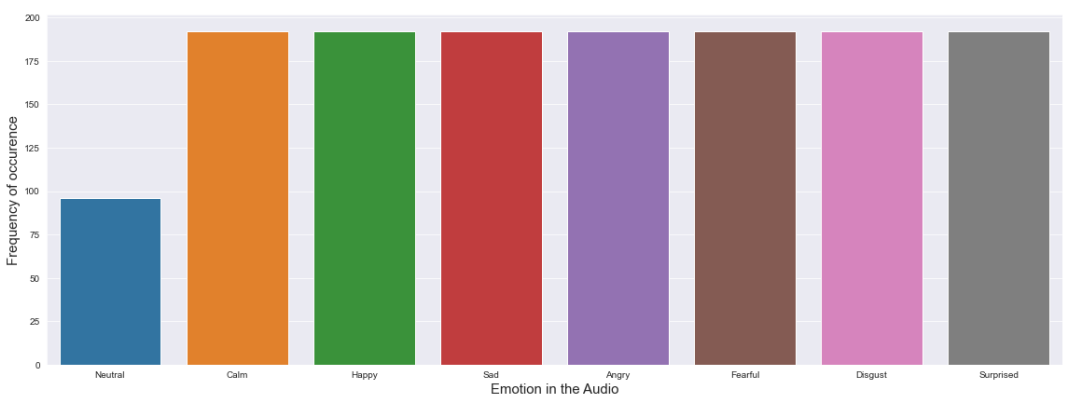

Fig. 1: The distribution of various emotions in the dataset.

II. Methodology

Overview:

The Classification Algorithm implemented in this project:

- SVM
- Logistic Regression
- Passive Aggressive Classifier
- Shallow MLP
- 1-D CNN
- KNN
- XGBoost
- Simple RNN
- LSTM

  We also make use of Data augmentation in preprocessing techniques including time rolling, varying the pitch, adding noise and time stretching.

Pipeline:

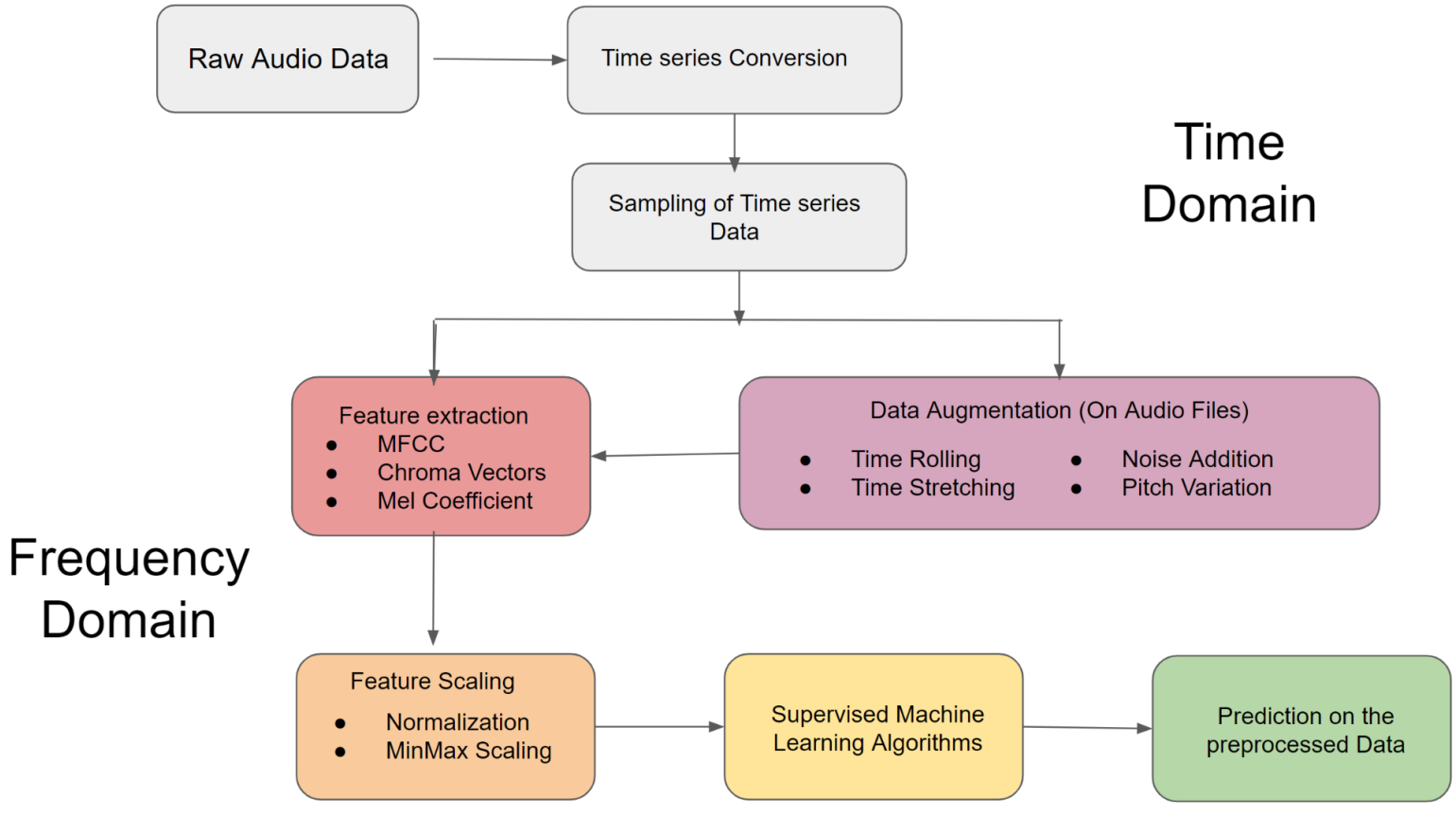

Fig. 2: end to end machine learning pipeline block diagram.

#Importing Modules and Analyzing Data

- We are provided with the voices of 24 actors each speaking two statements, 8 different times under a different emotional state, each emotion aside from neutral has two repetitions (higher emotional intensity in speech, lower emotional intensity while speech).

- Following the above, the dataset provides us with 1440 audio samples to work upon.
- We make use of the Librosa library for converting the audio data into time-series values and further converting it into the frequency domain.

Feature ( ) class

#Data Preprocessing

- Audio files are loaded into time-series vectors using the Librosa library
- Different frequency samplings are applied on the time-series vector to finally obtain Mel coefficients and Chroma Vectors.
- Fourier analysis is done on the same time series vector to obtain MFCCS
- ZCR is obtained using the Librosa library and the mean and variance of the time-series data are also extracted.

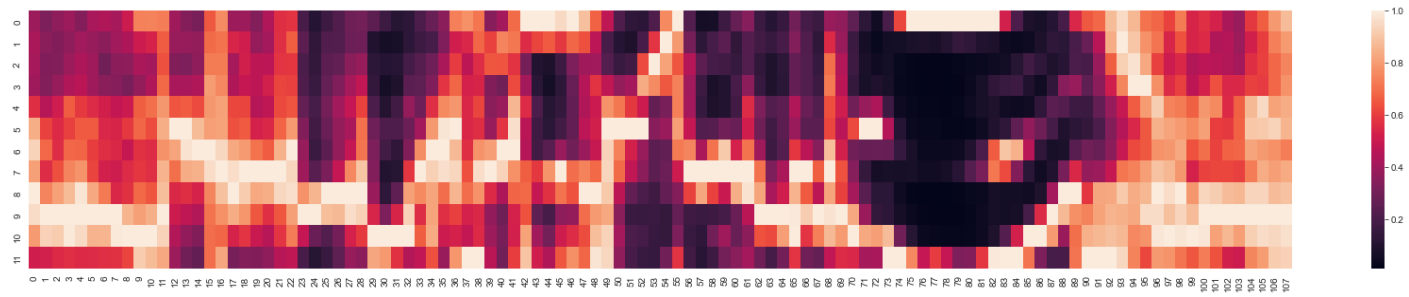

Fig3.: Visualized Chromagram.

#Augmenting data

- Four types of audio augmentation methods are created, to make the trained model less sensitive to trivial changes in the audio
- ***These are***
  - ***Time rolling:*** Audio is shifted in the time domain
  - ***Time stretching:*** Parts of the audio are stretched and resampled in the time domain
  - ***Pitch Variation:*** Pitch of the given audio is scaled or refactored
  - ***Noise Addition:*** A certain proportion of random noise is added to the audio provided.
- The feature extraction process is repeated for this newly transformed data.

$$\mathrm{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Fig. 4: The mathematical formulation of MEL coefficients.

#Classification Models

Features taken into consideration include Chroma vector, MFCC [Fig. 2], and Mel scale. Classification algorithms that were implemented for this project:

- SVM: In SVM, data points are plotted into n-dimensional graphs which are then classified by drawing hyperplanes.
  - Simple SVM with RBF and Polynomial kernel
  - Degrees range from 2 to 5.
  - Gamma is set to auto and C is kept at 1e+3 for the rbf kernel and to 1e-3 for the linear kernel.

- Logistic Regression: Logistic Regression model is widely used for binary classification but modified multinomial logistic regression can be used for multi-class classification.
    - Changing logistic regression from binomial to multinomial probability requires a change to the loss function used to train the model (e.g. log loss to cross-entropy loss), and a change to the output from a single probability value to one probability for each class label
    - Simple Logistic Regression is used

- Passive-Aggressive: Passive-Aggressive algorithms are somewhat similar to a Perceptron model because they do not require a learning rate. The input data comes in sequential order and the machine learning model is updated sequentially, as opposed to conventional batch learning, where the entire training dataset is used at once.
    - Simple classification model is used
    - Step constant is held at 1 while keeping tolerance = 1e-3
    - n estimators are kept at 1000

- Shallow MLP: Shallow neural networks consist of only 1 or 2 hidden layers. Understanding a shallow neural network gives us an insight into what exactly is going on inside a deep neural network.
    - The number of epochs is kept to 100. Batch Size is set to 128
    - Learning date 1e-3
    - Categorical cross-entropy loss is used as an evaluation metric

- 1-D CNN: CNN classification algorithm can learn from the raw time series data directly, and in turn does not require domain expertise to manually engineer input features
    - The activation function used is Relu
    - Categorical cross-entropy (as depicted in Fig. 5) loss is used as an evaluation metric

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

Fig. 5: The cross-entropy function.

- KNN(k-nearest neighbors): KNN has supervised algorithms that classify data on the basis of distance from similar points. Here k is the number of nearest neighbors to be considered in the majority voting process.
    - n-neighbors are varied from 1 to 50
    - Distance weighting is applied

- XGBoost: XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.
    - XGBoostClassifier with a max depth of 3 is used
    - n estimators are kept at 500
    - Learning rate is set to 0.2

- LSTM: Unlike standard feedforward neural networks, LSTM has feedback connections. It can process entire sequences of data.
    - We use an LSTM to work on the extracted vectors and treat them as pseudo temporal data as we have concatenated the features in the order **MFCC->Chroma->Mel** which is our decided order of importance and the application of LSTM keeps the MFCC context prevalent throughout the vector and so on.

#Custom Audio Prediction

****Note:** All the steps are briefly explained and visualized in the Notebook.*

III. Evaluation of Models

Results and Analysis:

The Accuracy of Models according to testing data is arranged below:

The models implemented were evaluated using techniques like - Classification report: precision, recall, F-1 score, and the number of support vectors in the appropriate case. The score for each model shown below is the weighted average score of class-wise predictions.

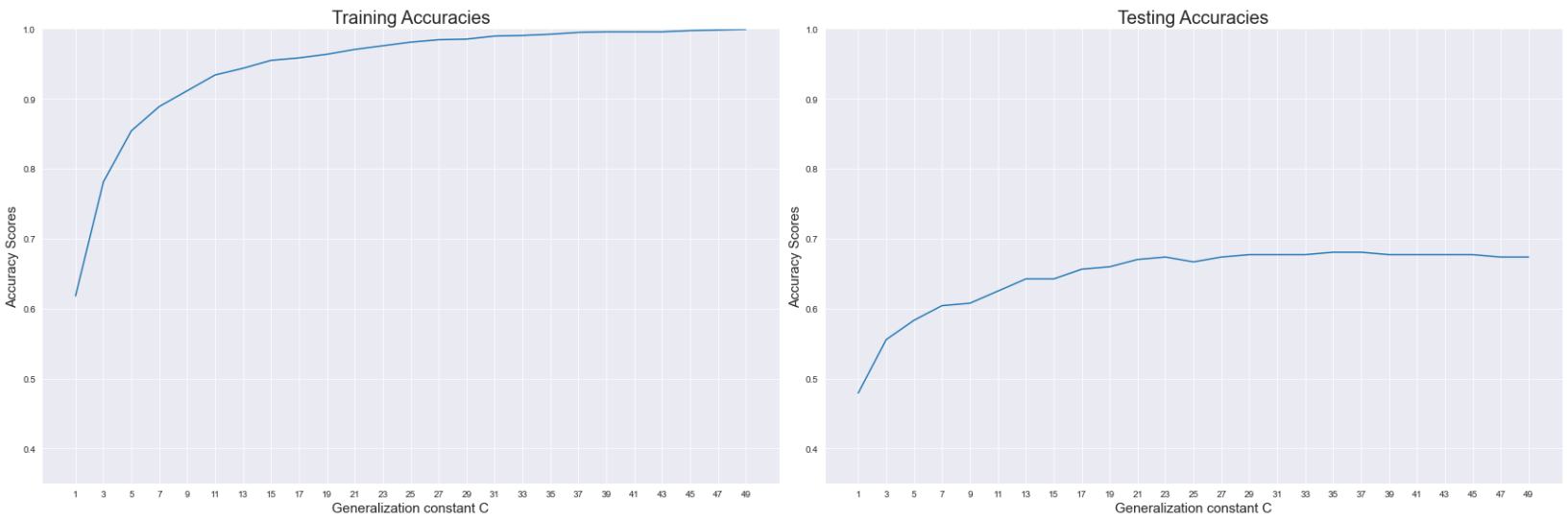

Fig. 6: Performance of SVM with variation of C.

Table 1: The performance comparison of original approaches.

	Precision	Recall	F1 Score	Accuracy Score
SVM	0.63	0.63	0.63	64%
Logistic Regression	0.51	0.51	0.51	52%
Passive Aggressive	0.44	0.43	0.42	45%
Shallow MLP	0.60	0.58	0.59	58%
1-D CNN	0.50	0.50	0.50	50%
KNN	0.64	0.63	0.63	64%
XGBoost	0.63	0.64	0.63	65%
LSTM	0.64	0.63	0.64	62%

Table 2: The performance comparison on augmented data.

	Precision	Recall	F1 Score	Accuracy Score
SVM	0.64	0.64	0.64	64%
Logistic Regression	0.50	0.49	0.49	50%
Passive Aggressive	0.43	0.43	0.42	43%
Shallow MLP	0.69	0.69	0.68	69%
1-D CNN	0.61	0.62	0.61	61%
KNN	0.64	0.63	0.63	64%
XGBoost	0.63	0.63	0.63	64%
LSTM	0.65	0.66	0.65	66%

Evaluation of Models:

As observed from the Table above the models performing the best on the objective SER are Shallow MLP and SVM(Tuned) and hence these approaches become the most viable algorithms to determine the emotional contexts from speech.

IV. Conclusion

In this work, we explored the challenging task of speech emotion recognition using various machine learning algorithms from the classical and modern paradigms on the Ravdess dataset. We found out that, amongst the algorithms being analyzed herein, the classical ML approaches discussed compete very closely with the modern approaches like 1-D CNN and MLPs.
But we would argue that more-modern deep learning based approaches like CNN-X (Shallow CNN), CNN14, MME, etc could easily outperform the classical algorithms due to their ability to learn distinguished representations from relationships underlying the data.

Front-end Development