A Comparative Analysis on Speech Emotion Classification Models

Imtiajul Islam

ID:2014–3–60–081

D.M. Kamruzzaman

ID:2015–1–60–128

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

East West University

Dhaka-1212, Bangladesh

September,2019

Declaration

I, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by me under the supervision of Md. Mohsin Uddin, Lecturer, Department of Computer Science and Engineering, East West University. I also declare that no part of this thesis/project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned	Signature
.....	.....
Md. Mohsin Uddin	**Imtiajul Islam**
Supervisor	**2014-2-60-081**
.....	.....
Dr. Taskeed Jabid	**D.M. Kamruzzaman**
Chairperson	**2015-1-60-128**

Abstract

Recently, speech recognition has been one of the most necessary domains in machine learning and deep learning. People tend to order a machine by speech more comfortably and so this field has emerged to fulfill this necessity. In this paper, we proceed towards a feature engineering-based approach for detecting emotion in speech domain. Handcrafted features from multiple audio files are inclined to feed into learning models Features extracted from text domain are also included for resolving ambiguity. We follow both machine learning and deep learning-based approach where extracted features are fed into six machine learning classifiers namely, Random Forest, Multinomial Naïve Bayes, Support Vector Machine, Logistic Regression and Gradient Boosting whereas deep learning consists of feed-forward neural network and LSTM based classifiers for feature extraction. For both domains, eight hand-crafted features are extracted. Finally, we compare machine learning model to deep learning model using accuracy, precision, recall and F-measure where we observe that shallow machine learning models achieve higher performance than renowned deep learning models for recognizing emotion.

Key Terms: multimodal speech emotion recognition, machine learning, deep learning.

Acknowledgements

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions with and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express my thankfulness other than simply listing those people who have contributed to this thesis itself in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express our deepest gratitude to the almighty for His blessings on us. Next, our special thanks go to our supervisor, "Md. Mohsin Uddin", who gave us this opportunity, initiated us into the field of "A Comparative Analysis on Speech Emotion Classification Models", and without whom this work would not have been possible. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our B.Sc. study were simply appreciating and essential. His ability to muddle us enough to finally answer our own question correctly is something valuable what we have learned and we would try to emulate, if ever we get the opportunity. Last but not the least, we would like to thank our parents for their unending support, encouragement and prayers.

There are numerous other people too who have shown us their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our heart and hope to find a more appropriate place to acknowledge them in the future.

Imtiajul Islam

September, 2019

D.M. Kamruzzaman

September, 2019

Table of Contents

Chapter 1

Introduction

1.1 Introduction

Since the beginning of mankind, communication system is growing day by day. People communicate with each other via text, speech and visual. Now, in this era of machines, machines learn from humans through text, image or speech. Ambiguity occurs in communication as people tend to say one thing through another domain and through tonal difference. Such as, “I am going to die” can be said under sad, fear or happy mode. Humans can deal with these ambiguities most of the times based on the circumstance. But for machines, it is a difficult task as they have to go through many phases of learning through machine learning or deep learning.

1.2 Motivation

For resolving ambiguity and capturing emotion in speech, many deep learning algorithms have emerged as in [1],[2],[3] and [23]. Many machine learning algorithms dwell too for this purpose ([4],[5]) and they give excellent result too. Although, people are trying to detect emotion in speech using deep learning as it is more robust in many cases. Here we try to compare deep learning models with machine learning models and thus try to clear the confusion of whether its’ intelligent to use deep learning or machine learning process for this purpose. Speech Emotion Recognition is highly demandable and applicable for training agents for prioritizing voice messages and replying with a proper feedback in call centers, online shopping portals etc.

1.3 Objective

In this paper, we try to resolve Speech Emotion Recognition (SER) through multi-class classification. We implement two classes of models from machine learning and deep learning. And then compare the performance of these two models by accuracy, recall, precision and f-score measurement. In one approach, we recall different machine learning models namely, Support Vector Machine, Naïve Bayes, Random forest, Logistic Regression and Gradient Boosting. In another approach, we focus on deep learning-based approach where the models are highly data-reliant. We implement a Multi-Layer Perceptron and LSTM [6] classifier to detect emotion in speech domain. Hand-crafted features extracted from audio signals are fed into the aforementioned classifiers. Textual information is combined too for eradicating ambiguity and understand the correlation between different modalities. But for LSTMs, we feed the classifier with only audio samples. *IEMOCAP* [7] dataset is used to evaluate the models for *Audio-only*, *Text-only* and *Audio + Text* settings.

1.4 Organization

In the upcoming part of the paper different sections of our work have been described. Chapter 2 contains related works on speech emotion recognition task that we have taken into account. Chapter 3 contains our methodology which consists of our problem definition, overview of the dataset we have used, data pre-processing steps, extracted features from audio and text and all the machine learning and deep learning models we have used. Chapter 4 includes our three types of experiment settings, implementation details and evaluation metrics. Chapter 5 concludes the work and chapter 6 is for holding references.

Chapter 2

Related Works

Some of the related works done on Speech Emotion Recognition (SER) have been discussed in this section. Emotion Recognition is not a new field in machine learning and many standard works have been done over time by many researchers. All of these works have opened new possibilities for future works and thus paved the way of building new models and epochs.

Some of the early works involve traditional machine learning based approaches in detecting emotion in speech. Many researchers have used Hidden Markov Model (HMM) [27] to identify emotion classes in speech domain ([25],[26]). Others used Bayesian Network model [28] [29], Support Vector Machines (SVM) [30], Multi-Classifier Fusion [31] Gaussian Mixture Model [32] to predict emotion in speech.

As deep learning has emerged, researchers were attracted more to it for emotion classification task. There are some novel works on speech emotion recognition based on deep learning. Like the researchers in [2] used different recurrent encoders namely audio recurrent encoder, text recurrent encoder, multimodal dual recurrent encoder and multimodal dual recurrent encoder with attention to predict emotion using audio and text. They used IEMOCAP [7] dataset for this task and showed accuracies ranging from 68.8% to 71.8%. Another group of researchers implemented CNN (Convolutional Neural Network) for this task [33]. They also used IMEOCAP [7] dataset and evaluated their model on text, spectrogram, MFCC and multiple different combination of these. Their overall accuracy ranged from 64.4% to 76.1%. CNN was also used broadly for SER ([34],[35]). But [33] showed the highest accuracy. Convolutional Neural Network and Recurrent Neural Network was also fused for emotion classification [36] which achieved a great f1-score and recall. In our work, we used simple concatenation of features from multiple domains. This concatenation was replaced by Tensor Fusion Network [37] and Low-Rank Matrix Multiplication [38] which were more efficient for combining features from different domains. All of these works extracted salient features from audio and text and many of them is state-of-the-art approach.

Chapter 3

Speech Emotion Recognition

3.1 Introduction

Speech Emotion Recognition is a challenging task for machine learning and deep learning. We have to understand the dataset clearly to extract salient features from it. We use IEMOCAP [7] dataset which consists of audio, visual data and text transcripts. As we are in the SER task, we work only on audio data and text transcripts. After extracting features of the audio samples from the dataset, we take in some features and discard others. This was done as some features have greater significance and some have less significance on the result. Also, if we take in the less significant features, the dataset overfits the models implemented. The task can be formalized as given a user utterance; we have to distinguish the class that the sentence belongs to. To understand the problem better, we first look up problem definition. Then we go through a simple overview of our dataset and the machine and deep learning models and finally evaluation of our result. Different learning models can be applied for the problem domain in which some performs better and some gives average performance.

3.2 Problem Definition

IEMOCAP is a great dataset to work on for emotion classification. We are getting more and more familiar with machines as technology evolves. We are likely to communicate more with speech than with text or image. So, it is important to recognize emotion in speech along with text. Detecting the right emotion plays a sophisticated role for interaction of human with machine. As we work on multimodal setting, we detect emotion in audio domain, text domain

and audio along with text domain. We implement different machine learning and deep learning models for this purpose and finally compare them based on evaluation metrics.

3.3 Dataset

In this work, we utilize *IEMOCAP* (Interactive Emotional Dyadic Motion Capture dataset) [7]. It is a multimodal dataset which contains utterances from multiple speakers. It contains huge audio visual data of nearly 12 hours which includes video, speech and text transcriptions. It contains eight nominal emotion labels, such as, happiness, anger, surprise, sadness, fear, excited, frustration and neutral. The dataset is divided into five sessions. Per session, the dataset is already divided into more than one utterance file. We gather wav file for each sentence by dividing the utterance files further more. This work was done by using the start and end timestamp which was provided for every transcribed sentence. By doing so, we finally have a total of ~10k audio files. These audio files are used for feature extraction. The extracted features are used for classification purpose.

3.4 Data Pre-processing

3.4.1 Audio

The dataset was unbalanced which was shown by doing a preliminary frequency analysis. “Happy” class was under-represented. To solve the issue, we combined “happy” and “excited” into “happy” as “excited” is pretty similar to “happy”. “Fear” and “surprise” classes were under-represented too and so we applied up-sampling technique for this issue. The emotion class “others” was merged with “neutral” class. Finally, we obtain six emotion classes from the eight classes and a total of 7837 emotion class examples. Here is a table showing the examples distribution per emotion class.

Table 1: Emotion sample distribution for each class

Class	Count
Angry	860
Happy	1309
Sad	2327
Fear	1007
Surprise	949
Neutral	1385
Total	7837

3.4.2 Text

Data preprocessing step for text data was pretty simple. We just lowercased each sample for normalization. Afterwards, we deleted all of the special symbols and stop-words. This is all for text pre-processing.

3.5 Feature Extraction

3.5.1 Audio Features

Pitch

Pitch is one of the most important features for audio. It represents the irrational frequency of the vocal cords during the sound productions. While many algorithms for calculating pitch signal exist, we use *auto-correlation of center clipped frames* [8] method which is very popular. The relation between the input signal x(n), and the center-clipped signal y(n) is,

$$y(n) = clc[x(n)] = \begin{cases} (x(n) - C_L), & x(n) \geq C_L \\ 0, & |x(n)| < C_L \\ (x(n) + C_L), & x(n) \leq -C_L \end{cases} \quad (1)$$

where C_L is the threshold which is normally half times mean value of input signal. Auto-correlation is calculated on center-clipped signal y(n).

Pitch is lower when the frequency of the signal is lower and higher when the frequency is higher. This is shown is figure 1.

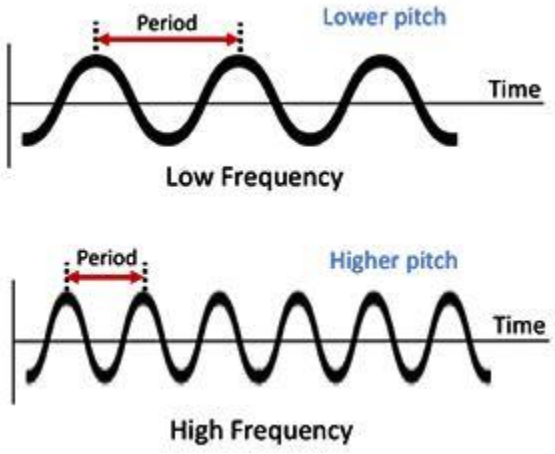


Figure 1: Pitch Comparison for Frequency Variation.

Harmonics

Harmonics are considered the source of the sound. Harmonics are considered to be any frequency that resides in a system along with the fundamental frequency.

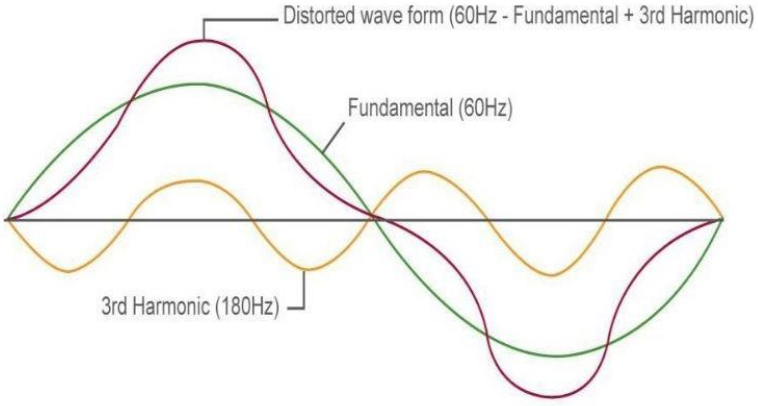


Figure 2: Distortion of Fundamental Frequency for Harmonics

We attempt to calculate harmonics by an approach described in [9] using a median based filter. Median filters replace the given sample in a signal with the median value of the signals. Given an input vector x(n) then y(n) is the output of a median filter of length l where l defines the number of samples over which median filtering takes place. Where l is odd, the median filter can be defined as:

$$y(n) = median\ \{x(n - k : n + k),\ k = (l - 1)/2\} \quad (2)$$

In cases where l is even, the median is obtained as the mean of the two values in the middle of sorted list. A harmonic-enhanced spectrogram frequency slice H_h can be obtained by median filtering frequency slice S_h.

$$H_i = M\{S_h, l_{harm}\} \quad (3)$$

Where l_{harm} is the length of the harmonic median filter *M*. The slices are then combined to give a harmonic enhanced spectrogram *H*. When we are angry or stressed, we create additional frequency spikes along with pitch ([10], [11]) which are seen in spectrogram. These signals from the spectrogram are referred to as harmonics and cross-harmonics.

Speech Energy

Speech energy is directly related to voice loudness. More loudness produces more energy. So, when a person is happy or angry, his speech energy is greater than of other emotional states. We use RMSE (Root Mean Square Energy) to calculate speech energy by the formula given below,

$$E = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \mathrm{y[i]}^2} \quad (4)$$

RMSE is calculated for each frame, *I* and both standard deviation and average is taken into account for features.

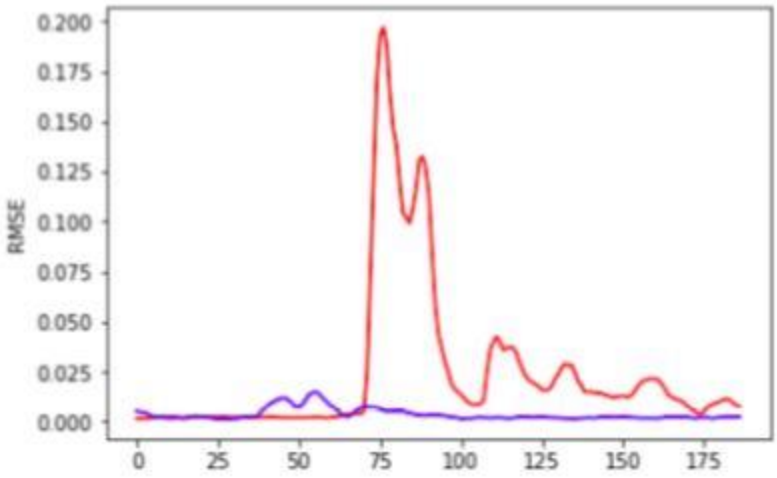


Figure 3: RMSE Plots For Sad (Blue) and Angry (Red) Audio Signals

Pause

Pause indicates the silent moments of speech signals. It helps us find emotion classes effectively as Pause is less in some emotion state whereas more in some other emotion state *i,e,* when we are "*excited"* or "*angry*", we speak fast and so the Pause is less. We calculate Pause by the formula below,

$$Pause = Pr(y[n] < t) \quad (5)$$

where t is a threshold of $0.4*E$, where E is the RMSE.

Central Moments

Central moment is a probability distribution of a variable using the variable's mean. We utilize both mean and standard deviation of the amplitude of the input audio signal in order to obtain effective information from the input signal.

Zero-Crossing Rate

The algorithm of zero-crossing rate [44] computes an audio signal. It is simply the rate of a signal amplitude crossing zero. It can be defined as the sign of signal changes from positive to zero to negative or from negative to zero to positive during a signal frame which is divided by total length of the frame. Noisy signal has higher zero-crossing rate. The feature is used mostly in both speech recognition and music industry. We took

standard deviation value of Zero crossing rate for our work.

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} 1\, R < 0(StSt - 1) \qquad (6)$$

where *s* is a signal of T length and $1_{R<0}$ is an indicator function.

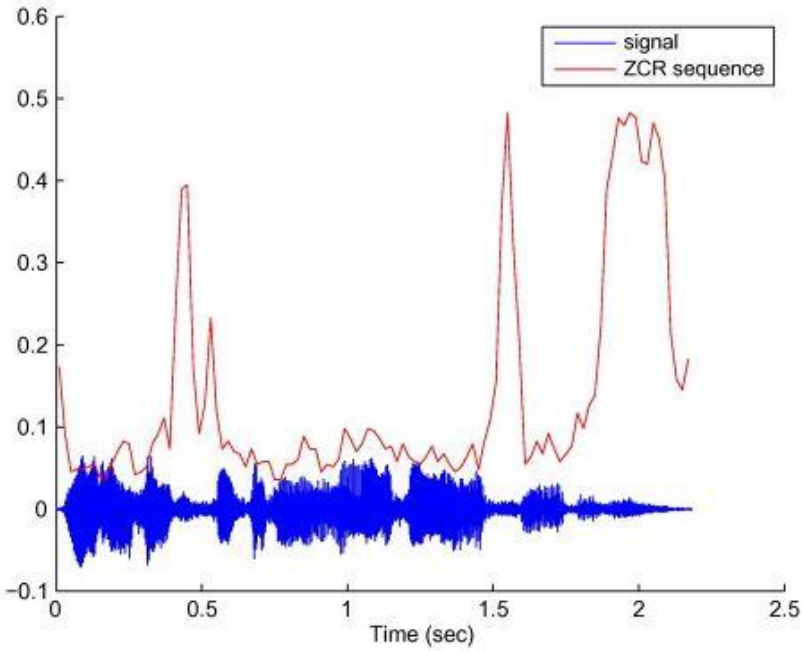


Figure 4: Zero-Crossing Rate Sequence Over Time

Mel Frequency Cepstral Coefficient

In audio recognition, Mel-frequency Cepstral Coefficient feature is highly used. Mel is calculated as

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700}) \qquad (7)$$

Normal frequency *f, Mel(f)* is the logarithmic scale. Mel covers the frequency range from 0 Hz to 20050 Hz.

MFCC determines the shape of human vocal tract tongue, teeth etc. When the structure is determined correctly, any speech can be accurately represented.

MFCC is used to convert signal residing in time domain into frequency domain. MFCC is

also used in music for genre classification, measuring audio similarity etc. In our work, we calculate the mean, standard deviation of MFCC. It is highly applicable for SER and so it is included as one of the features in our work.

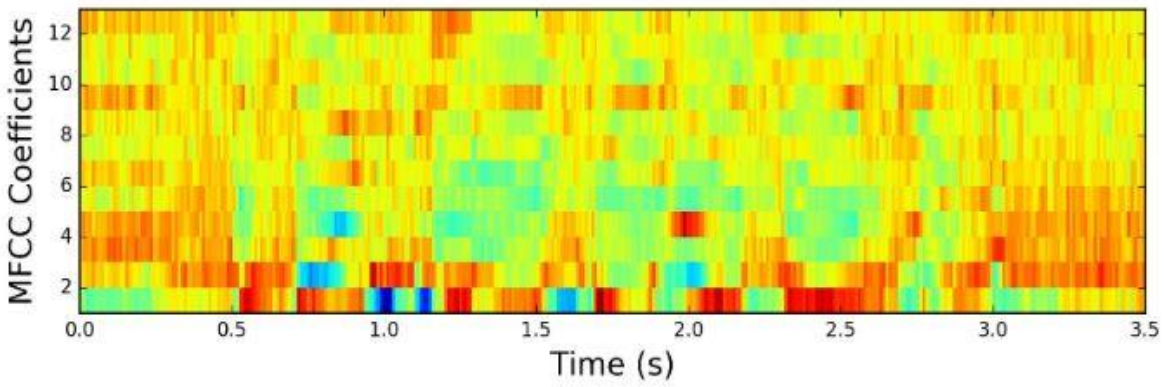


Figure 5: Features extracted from speech signal

Spectral roll-off

The spectral roll-off is concentrated as below frequency of 85% of the magnitude distribution of a spectrum. Skewness of a spectral shape is measured by spectral roll-off. It is mostly used to separate voice from unvoiced speech and music. An unvoiced speech contained high proportion of energy which leads to high frequency range of the spectrum. We used both mean and standard deviation of the spectral roll-off.

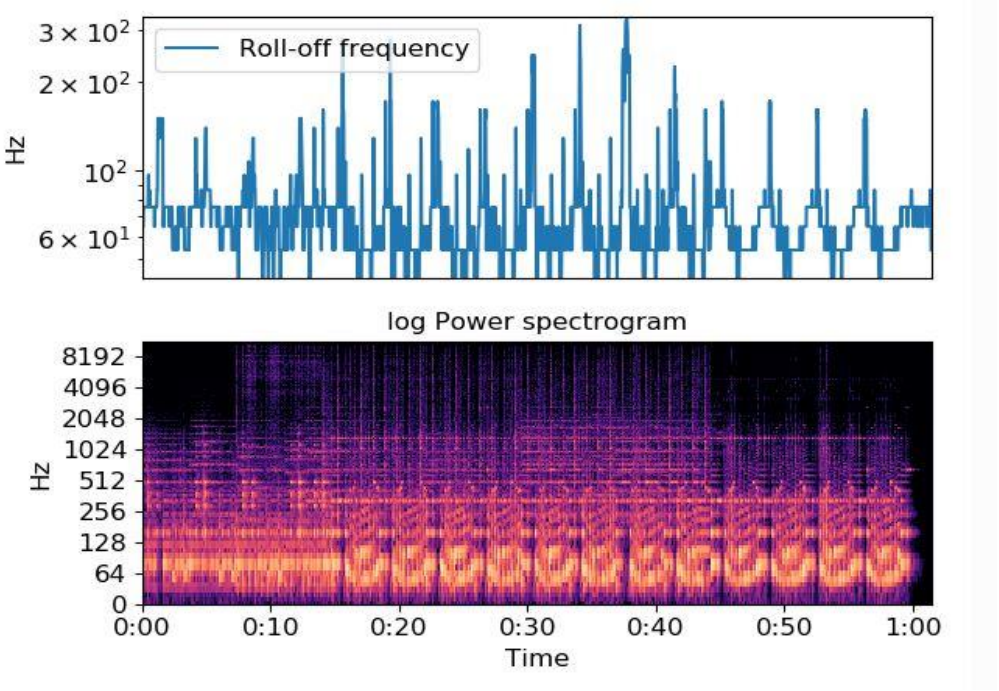


Figure 6: Roll-Off Frequency distribution over time

3.5.2 Text Features

Term Frequency-Inverse Document Frequency

This is the most popular and efficient measure to extract text features. It is a numerical statistic that reflects how important a word is to a document in a collection or corpus [12]. It consists of two parts.

Term Frequency

It compares a word or term with a document to show how many times the word has appeared on the document. The easiest way to do so is to do a *raw count* of the word in the document or sentence.

Inverse Document Frequency

The inverse document frequency is a measure of information volume provided by a word, i.e., whether if it's common or rare across all documents. It is the logarithmically measured inverse fraction of sentences that has the word in it (obtained by dividing the total number of sentences by the number of sentences that contain the word, and then

doing logarithm scaling of that quotient):

$$Idf(t,D) = log\ \frac{N}{|\{d \in D : t \in D\}|} \quad (8)$$

where N is the total number of sentences and $|\{d \in D : t \in D\}|$ is the number of sentences where t appears in.

3.6 Algorithms

3.6.1 Machine Learning Models

Machine learning has become one of the most popular method for classification or regression. Basically, it is a learning method for machines or computers. Machine learning is replete with its' models. We have
worked on five different machine learning models. They output great result for speech classification. Here we provide a short overview of the machine learning algorithms that we have taken into account for this task. Below is a short description of those.

Multinomial Naïve Bayes

Naïve Bayes is one of the simplest probabilistic classifiers available. It is computationally very easy to implement and also very efficient. It
assumes that given a class and feature vectors, the features are conditionally independent from one another. Then it tries to to assign a feature or text from the feature vector to the class. The posterior probability is calculated by Bayes formula, which is,

$$P(B) = \frac{P(A)P(A)}{P(B)} \quad (9)$$

Where A is a feature and B is the class label. Under multinomial setting,
the features are the word frequency of those words that are present in a
document or dataset. Multinomial Naïve Bayes is mostly used for
document classification or text categorization [13]. In work it is used for text only and

audio + text setting. However, its' inferior to the state-of-the-art support vector machine classifiers in terms of classification accuracy when it is applied for text categorization problems.

Random Forest

Random Forest [14] is an ensemble machine learning method. Ensemble learners consist of many weak classifiers to build a strong classifier. Random Forests are a collection of randomly created decision trees. The trees are weak classifiers. Each decision tree can have its' own rule for classification purpose. Random forest basically works on these two principles.

- Each tree produces an output class given a random subset of features [16].
- The train set for one decision tree is one subset of features. This is referred to as bootstrapping aggregating [17]

The majority of class is then chosen as the final prediction. Random forests are great to work with because they avoid over-fitting and the same model or collection of trees works great both for classification and regression.

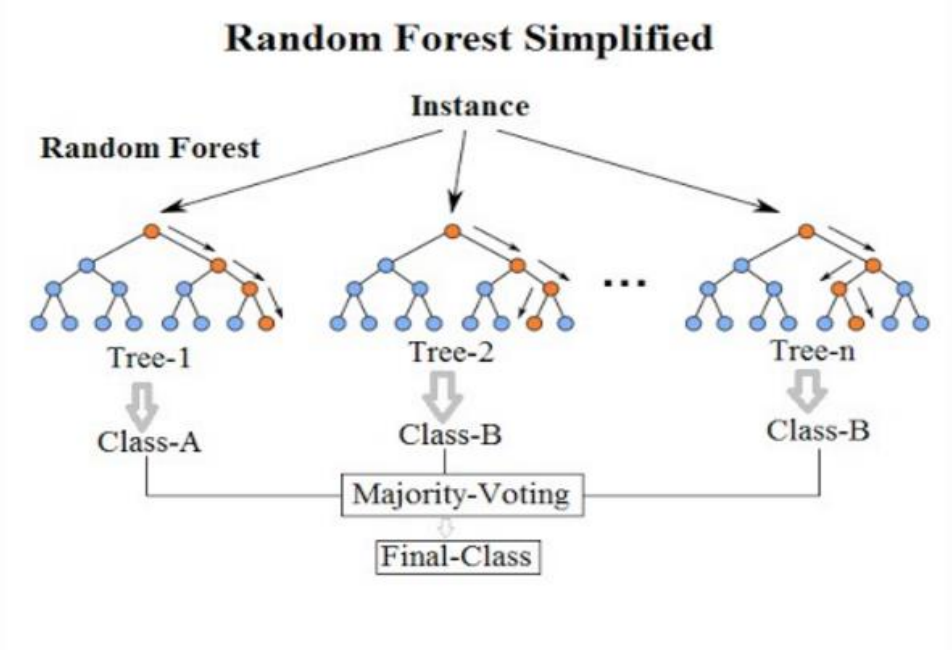


Figure 7: Random Forest Simplified

Gradient Boosting(xGB)

xGB is short for eXtreme Gradient Boosting [15]. Gradient Boosting is also an ensemble model based on some weak learners, mostly decision trees. Train data is fed into the classifier in a sequential, additive and gradual manner. On the output, it reminds the executor how many trees or iterations were needed or execution and whether any tree has zero significance in classifications or not. At the early stage, the learners are weak and become strong by doing iterations as each iteration learns from the mistakes made by the previous iterations. Finally, a strong prediction is achieved as final output.

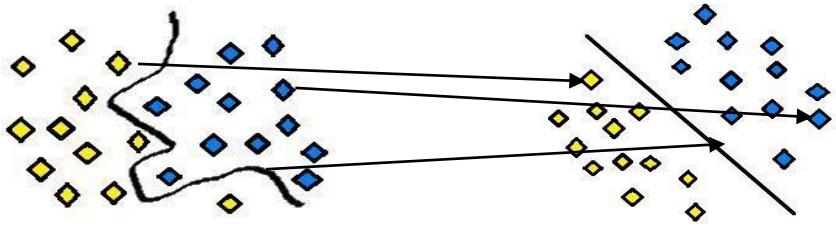
Support Vector Machine (SVM)

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships [18]. SVM employs an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into the following-

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \qquad (10)$$

This is the error function where C is constraint which multiplies the error variable.

Input space Feature space



SVM is capable of doing both classification and regression. It uses a technique called the kernel trick [20] to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. SVM is essentially non-probabilistic linear classifier but some methods such as, Platt Scaling [19] turns SVM into a probabilistic classifier.

Logistic Regression

Logistic regression [21] is a classification algorithm used to assign observations to a discrete set of classes. Logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). It is used to predict a binary outcome [22].

3.6.2 Deep Learning Models

Although deep learning-based models are robust, they take huge amount of time as well as huge computational resources. End to end training is applied in deep learning models. In our work, we implement two deep learning models.

Multi-Layer Perceptron (MLP)

MLP is a non-linear supervised feed-forward neural network. Its architecture allows it to have three nodes at least: input layer, output layer and hidden layer. Interleaving and backpropagation [24] is applicable in train phase for stabilizing the network. All the layers use nonlinear activation function except the input layer. Precision in output increases as the number of nodes in hidden layer increase. It can distinguish linearly separable data.

Long Short-Term Memory

Long Short-Term Memory networks usually just called LSTM [6] are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn.

LSTM utilizes gating mechanism where there are three different types of gates namely, input, output and forget gates. The equations necessary for LSTM are mentioned below,

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (11)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (12)$$

$$o_t = \sigma_g(W_o x_t + U_{ogh} t - 1 + b_o) \quad (13)$$

$$c_t = f . c_{t-1} + i_t . \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (14)$$

$$h_t = o_t . \sigma_h(c_t) \quad (15)$$

Wherec_o = 0 and h_o = 0 at initial step and [·] is the element-wise product, t is the time step, $x_t$is the LSTM unit input vector, f_t is the activation vector of forget gate, $i_t$is the activation vector of input gate, o_t refers to the activation vector of output gate, h_t is hidden state vector (which maps a vector toward the lower-dimensional latent space from a feature space), c_t is the vector of cell state and W,U are weight metrics and b is bias metric.Figure 2 shows the feedback mechanism of LSTM.

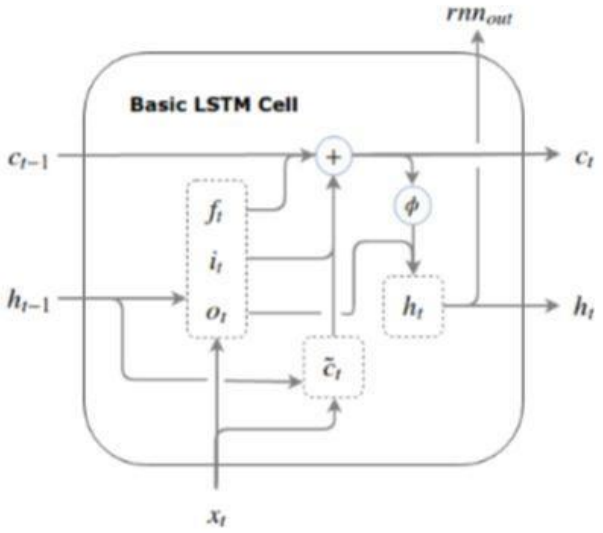


Figure 8: Basic feedback mechanism of LSTM cell

Chapter 4

Experiment

We work on three different settings to run our experiments. Three individual settings give three different kind of accuracy measures. It would be interesting if we could work with more settings. The settings are described in 4.1 (Experiment Settings). Feature vectors from these settings are fed into the machine learning classifiers where the class label is determined and later on comparison between different classifiers is inducted.

4.1 Experiment Settings

Audio-Only

For this setting, all of the machine learning and deep learning classifiers were trained with the audio feature vectors which were Pitch, Harmonics, Pause, Speech Energy, Central Momments, Zero Crossing Rate, Spectral Roll-Off and Mel-Frequency Cepstral Coefficient.

Text-Only

For this setting, all except the LSTM classifiers were trained with text feature vectors. We used TF-IDF vectors in this case. Text features were not included in LSTMs because it creates complexity and overfits the model.

Audio with Text

For this setting, feature vectors obtained from the audio modality and text modality were fused in together. There are some extensive researches to fuse vectors efficiently from multiple different modalities like Tensor Fusion Network [37] and Low-Rank Matrix Multiplication [38]. In our case, we just concatenate the feature vectors obtained from audio signals and text data.

4.2 Implementation Details

In this section of the chapter, details of our implementation is described.

- We have used Python 3.7 to conduct all our experiments.
- We have used librosa [39], a library of Python, in order to fetch the audio files and to extract salient features from those.
- We have used scikit-learn [40] as well as xgboost [41], which are some of the most popular libraries of Python on machine learning. We used these libraries in order to implement the machine learning classifiers namely, Random Forest, Extreme Gradient Boosting, Support Vector Machine, Multinomial Naïve Bayes, and Logistic Regression. Multi-Layer Perceptron, the deep learning classifier is also trained with the help of these libraries.
- We have used PyTorch [42], a deep learning library for implementing LSTM classifiers which is described previously.
- For regularizing the hidden layer of LSTM classifiers, we have used a mechanism, named dropout [43], where a certain number of neurons or nodes are not utilized for making the final prediction. This method is proven to be useful for robustness increase of the deep neural network as well as preventing overfitting.
- We randomly do a 80/20 split on our dataset where 80% of the data are in training and 20% of them are in testing. We do not include any test data from outer environment because of the feature extraction and computational overhead complexity. This split is applied to all the machine learning and deep learning models.
- We stop the training when there is no performance improvement for >10 epochs. Here, one epoch is denoted as one iteration on all of the train samples.
- Batch sizes were different for all the different models.

4.3 Evaluation Metrics

In this part of the chapter, different evaluation metrics used for measuring our performance are discussed.

Accuracy

This indicates the test sample percentage which are correctly classified. Given a set of data points of the same quantity, the set can be said accurate if their average is close to the true value of the measured quantity. But classification accuracy can be wrong or misleading if observations of each class is an unequal number or if there is greater than two classes in the dataset.

Precision

Precision is also called positive predictive value. Precision informs that among all predictions, how many predictions are actually present in the labels. From confusion matrix, precision can be denoted by the formula,

$$Precison = \frac{tp}{tp + fp} \qquad (16)$$

Recall

This measure informs how many correct labels are present in the predicted output. Recall has the ability to gather all of the instances that are relevant in a dataset. Recall can be expressed by the formula,

$$Recall = \frac{tp}{tp + fn} \qquad (17)$$

F-score

F-score is the harmonic mean of precision and recall. F-score was taken as accuracy measure as accuracy alone can be misleading. Also because F-score is more normalized than accuracy.

In the formulas above, tp, fn, and fp denotes true positive, false negative and false positive respectively. These values are derived from the confusion matrix.

Chapter 5

Results

5.1 Introduction

In this chapter, we discuss about the results of our work. Results are particularly interesting for each of settings namely, audio only, text only and audio + text setting. We discuss about results of these and compare the results with current state-of-the-art for this dataset. Additionally, we compare the results of the different models that we have employed.

5.2 Audio Only

Results are actually interesting for this setting. Performance of LSTM and ARE reveals that deep models indeed need a lot of information to learn features as the LSTM classifier trained on eight-dimensional features achieves very low accuracy as compared to the end-to-end trained ARE. However, E1 model (Ensemble of RF, XGB and MLP) which was trained on the eight-dimensional audio feature vectors performs poor in terms of accuracy but great in precision (Table 2). A look at the confusion matrix (Fig. 9) reveals that detecting “neutral” or distinguish in between “angry”, “happy” and “sad” is the most difficult for the model.

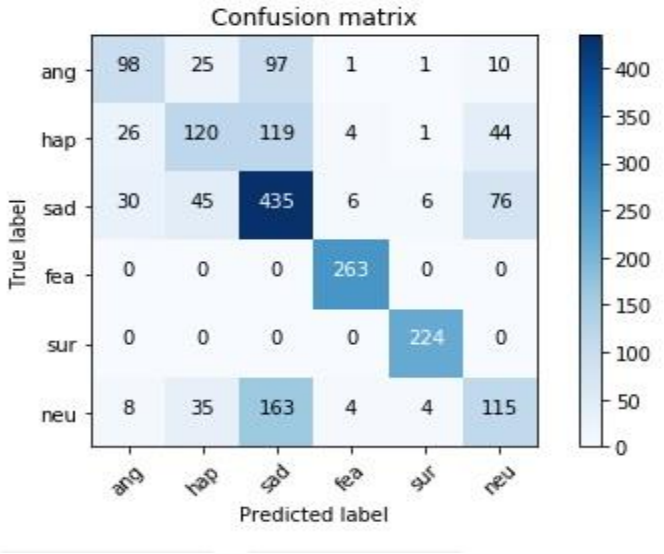


Figure 9: (E1: Ensemble (RF+XGB+MLP))

Models	Accuracy	F1	Precision	Recall
RF	63.5	64.4	67.2	63.8
XGB	63.3	65.2	66.5	64.6
MNB	31.2	9.8	14.4	17.7
SVM	58.1	58.5	60.8	58.2
LR	37.4	28.3	40.4	28.9
MLP	58.1	58.5	60.8	58.2
LSTM	51.20	50.69	57.8	49.51
ARE(4-class)	56.3	-	54.6	-
E1(4-class)	56.2	45.9	67.6	48.9
E1	64.0	65.1	67.3	64.7

Table 2: Audio data

5.3 Text Only

We observe that the performance of all the models for this setting is similar. This could be attributed to the richness of TFIDF vectors known to capture word-sentence correlation. We see from the confusion matrix (Fig. 10) that our text-based models are able to distinguish the six emotions fairly well along with the end-to-end trained TRE. We observe that “sad” is the toughest for textual features to identify very clearly. We see that for text data, our model outperforms both ARE and TRE in terms of accuracy and precision.

Models	Accuracy	F1	Precision	Recall
RF	64.0	63.1	65.7	64.3
XGB	56.4	55.4	**72.6**	51.3
MNB	61.0	61.1	**71.7**	57.5
SVM	61.0	62.8	65.4	61.4
LR	63.1	63.6	68.7	61.5
MLP	62.9	63.2	63.2	64.9
TRE(4-class)	**65.5**	-	63.5	-
E1(4-class)	63.1	61.4	**67.7**	59.0
E2	63.4	65.0	**70.0**	62.2

Table 3: Text data

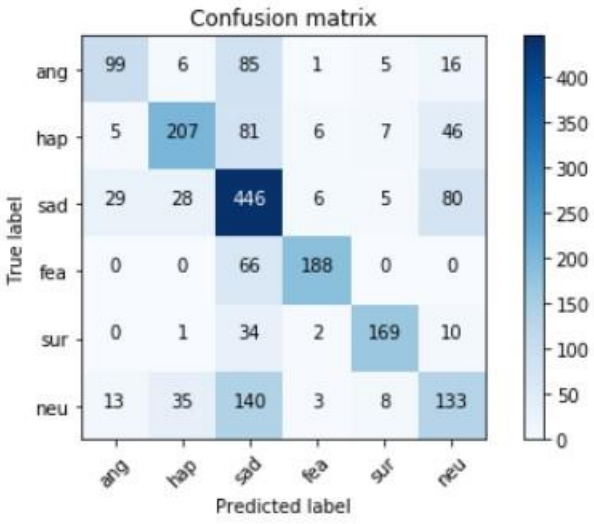


Figure 10: (E2: Ensemble (RF+XGB+MLP+MNB+LR))

5.4 Audio + Text

We see that combining audio and text features gives us a boost for all the metrics. This is clear evidence of the strong correlation between text and speech features. Also, this is the only case when the recurrent encoders seem to perform slightly better in terms of accuracy but at the cost of precision. The lower performance of E1 maybe be attributed to the trivial fusion method (concatenation) we use as simple concatenation for an ML model would still

contain a lot of modality-specific connections instead of the desired inter-modal connections. The promising result here is that combining features from both the modalities indeed helped to resolve the ambiguity observed for modality-specific models as shown in Fig. 6c. We can say that the textual features helped incorrect classification of "angry" and "happy" classes whereas the audio features enabled the model to detect "sad" better.

Models	**Accuracy**	**F1**	**Precision**	**Recall**
RF	65.6	65.8	**71.0**	65.0
XGB	65.4	66.0	69.3	65.3
MNB	58.5	58.0	69.7	54.4
SVM	65.7	66.6	66.4	67.4
LR	35.9	26.9	36.3	28.2
MLP	68.1	70.5	69.8	70.5
MDRE(4-class)	**75.8**	-	71.8	-
E1(4-class)	70.3	67.5	73.2	65.6
E2	71.4	72.4	74.1	71.9

Table 4: Audio + Text Data

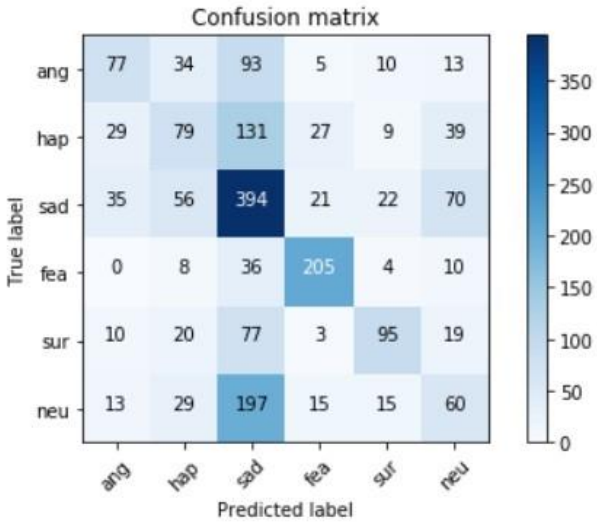


Figure 11: (E2: Ensemble (RF+XGB+MLP+MNB+LR))

5.5 Most Important Features

In this section, we investigate which features contribute the most during prediction in this classification task. We chose the XGB model for this study and rank the eight audio features. We see that Harmonic, which is directly related to the excitation in signals, contributes the most. It is interesting to see that “silence” attributing to Pause, is almost as significant as standard deviation of the auto-correlated signal (related to pitch). The low contribution of central moments is expected as a signal is very diverse and a global/coarse feature would be unable to identify the noises present in it.

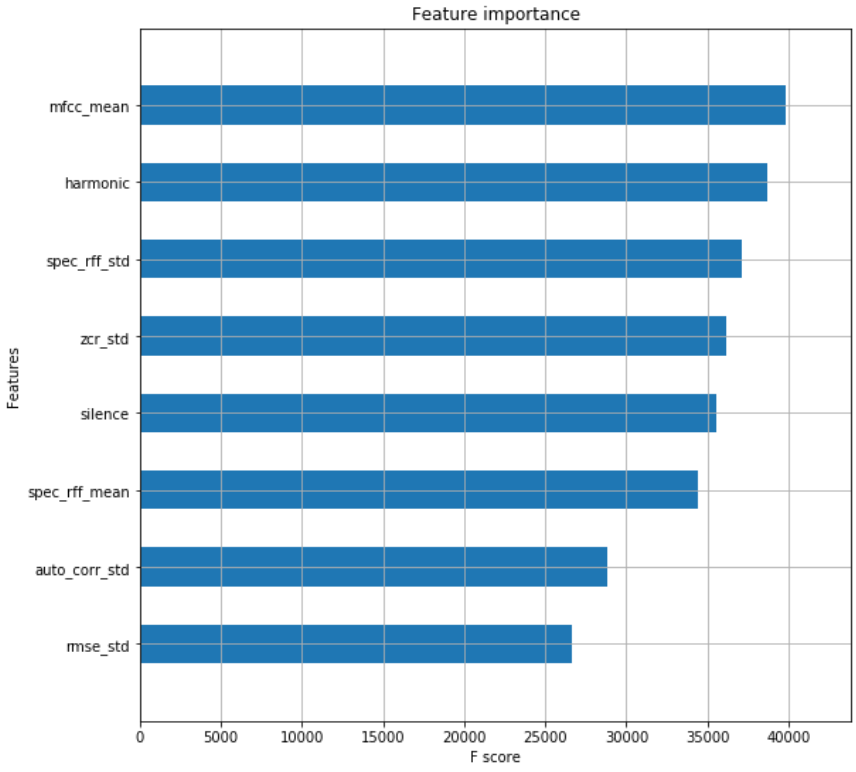


Figure 12: Most Important Audio Feature

Chapter 7

6.1 Conclusion and Future Work

In our work, we handle the task of speech emotion recognition and learn the contribution of different modalities towards ambiguity resolution on the IEMOCAP dataset. This is a challenging task as speech data contains more noise and additional information than text data or visage. This task was carried out with many models. We compare both, machine learning and deep learning based models. We only extract a handful of time-domain features from audio signals. If we extracted more features, it could've led to better result in terms of accuracy. We have not used feature selector of python which could have showed more improvement in feature extraction. Using PCA could have helped in dimension reduction. Also, better fusion methods such as TFN [37] and LMF [38] could be employed for combining speech and text vectors more effectively. We also want to work on Transfer Learning and Reinforcement Learning Method in future. It would also be interesting to see the scaling in the performance of machine learning models v/s deep learning models if we include more data.

Chapter 6

References

[1] Huang, Z., Dong, M., Mao, Q., & Zhan, Y. (2014, November). Speech emotion recognition using CNN. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 801-804). ACM.

[2] Yoon, S., Byun, S., & Jung, K. (2018, December). Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)* (pp. 112-118). IEEE.

[3] Han, K., Yu, D., &Tashev, I. (2014). Speech emotion recognition using deep neural network and extreme learning machine. In *Fifteenth annual conference of the international speech communication association*.

[4] Pan, Y., Shen, P., & Shen, L. (2012). Speech emotion recognition using support vector machine. *International Journal of Smart Home, 6*(2), 101-108.

[5] Casale, S., Russo, A., Scebba, G., & Serrano, S. (2008, August). Speech emotion classification using machine learning algorithms. In *2008 IEEE international conference on semantic computing* (pp. 158-165). IEEE.

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

[7] Busso, C., Bulut, M., Lee, C. C., Kazemzadeh, A., Mower, E., Kim, S., ... & Narayanan, S. S. (2008). IEMOCAP: Interactive emotional dyadic motion capture database. *Language*

resources and evaluation, *42*(4), 335.

[8] Sondhi, M. (1968). New methods of pitch extraction. *IEEE Transactions on audio and electroacoustics*, *16*(2), 262-266.

[9] Fitzgerald, D. (2010). Harmonic/percussive separation using median filtering.

[10] Teager, H. M., & Teager, S. M. (1990). Evidence for nonlinear sound production mechanisms in the vocal tract. In *Speech production and speech modelling* (pp. 241-261). Springer, Dordrecht.

[11] Zhou, G., Hansen, J. H., & Kaiser, J. F. (2001). Nonlinear feature based classification of speech under stress. *IEEE Transactions on speech and audio processing*, *9*(3), 201-216.

[12] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, pp. 133-142).

[13] Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004, December). Multinomial naive bayes for text categorization revisited. In *Australasian Joint Conference on Artificial Intelligence* (pp. 488-499). Springer, Berlin, Heidelberg.

[14] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[15] Chen, T., He, T., Benesty, M., Khotilovich, V., & Tang, Y. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1-4.

[16] Amit, Y., Geman, D., & Wilder, K. (1997). Joint induction of shape features and tree classifiers. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11), 1300-1305.

[17] Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

[18]Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology*, *24*(12), 1565.

[19] Platt, John. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." *Advances in large margin classifiers* 10.3 (1999): 61-74.

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

[21] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). *Logistic regression*. New York: Springer-Verlag.

[22] King, G., & Zeng, L. (2001). Logistic regression in rare events data. *Political analysis*, *9*(2), 137-163.

[23] Morgan, N., & Bourlard, H. (1990, April). Continuous speech recognition using multilayer perceptrons with hidden Markov models. In *International conference on acoustics, speech, and signal processing* (pp. 413-416). IEEE.

[24] Widrow, B., & Lehr, M. A. (1990). 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, *78*(9), 1415-1442.

[25] Nwe, T. L., Foo, S. W., & De Silva, L. C. (2003). Speech emotion recognition using hidden Markov models. *Speech communication*, *41*(4), 603-623.

[26] Nogueiras, A., Moreno, A., Bonafonte, A., &Mariño, J. B. (2001). Speech emotion recognition using hidden Markov models. In *Seventh European Conference on Speech Communication and Technology*.

[27] Eddy, S. R. (1996). Hidden markov models. *Current opinion in structural biology*, *6*(3), 361-365.

[28] Ververidis, D., &Kotropoulos, C. (2008). Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. *Signal Processing, 88*(12), 2956-2970.

[29] Mao, X., Chen, L., & Fu, L. (2009, March). Multi-level speech emotion recognition based on HMM and ANN. In *2009 WRI World congress on computer science and information engineering* (Vol. 7, pp. 225-229). IEEE.

[30] Hu, H., Xu, M. X., & Wu, W. (2007, April). GMM supervector based SVM with spectral features for speech emotion recognition. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07* (Vol. 4, pp. IV-413). IEEE.

[31] Wu, C. H., & Liang, W. B. (2010). Emotion recognition of affective speech based on multiple classifiers using acoustic-prosodic information and semantic labels. *IEEE Transactions on Affective Computing, 2*(1), 10-21.

[32] Neiberg, D., Elenius, K., & Laskowski, K. (2006). Emotion recognition in spontaneous speech using GMMs. In *Ninth international conference on spoken language processing*.

[33] Tripathi, S., Kumar, A., Ramesh, A., Singh, C., &Yenigalla, P. (2019). Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions. *arXiv preprint arXiv:1906.05681*.

[34] Mao, Q., Dong, M., Huang, Z., & Zhan, Y. (2014). Learning salient features for speech emotion recognition using convolutional neural networks. *IEEE transactions on multimedia, 16*(8), 2203-2213.

[35] Huang, Z., Dong, M., Mao, Q., & Zhan, Y. (2014, November). Speech emotion recognition using CNN. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 801-804). ACM.

[36] Lim, W., Jang, D., & Lee, T. (2016, December). Speech emotion recognition using

convolutional and recurrent neural networks. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)* (pp. 1-4). IEEE.

[37] Zadeh, A., Chen, M., Poria, S., Cambria, E., &Morency, L. P. (2017). Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

[38] Liu, Z., Shen, Y., Lakshminarasimhan, V. B., Liang, P. P., Zadeh, A., &Morency, L. P. (2018). Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*.

[39] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (Vol. 8).

[40] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... &Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, *12*(Oct), 2825-2830.

[41] Chen, T., &Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acmsigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

[42] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... &Lerer, A. (2017). Automatic differentiation in pytorch.

[43] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., &Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

[44] Barnett, J. T., & Kedem, B. (1991). Zero-crossing rates of functions of Gaussian processes. *IEEE Transactions on Information Theory*, *37*(4), 1188-1194.