The International Journal of Engineering and Science (IJES)
|| Volume || 13 || Issue || 5 || Pages || PP 23-34 || 2024 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Investigating the causes of uncertainty in emotion recognition from multimodal speech Inputs

Stewart Fohlo
Department of Computer Science
University of KwaZulu Natal
Durban, South Africa

----------ABSTRACT----------
The identification of emotions from speech is a challenging task due to the vague definition of emotion itself. This study employs a feature-based approach to address speech emotion recognition. The problem is formulated as a multi-class classification task, and two types of models are compared in terms of their performance. Eight manually crafted features are extracted from speech for both approaches. In the first method, six traditional machine learning classifiers are trained using the extracted features, while the second approach utilizes deep learning techniques, where a feed-forward neural network and an LSTM-based classifier are trained on the same features. Additionally, to mitigate communication ambiguities, text-based features are also included. The study evaluates the models in various settings and reports accuracy, f-score, precision, and recall. The results demonstrate that simpler machine learning models trained on a few handmade features can achieve comparable performance to the current state-of-the-art deep learning-based method for emotion recognition.

Keywords: *emotion recognition, multimodal, deep learning, multi-class classification, machine learning*

Date of Submission: 01-05-2024 Date of acceptance: 12-05-2024

I. Introduction

The ability to communicate is crucial for human survival, and we frequently encounter situations that are open to interpretation. For example, the statement "This is out of this world" could be uttered in either a joyful or a melancholy context. Humans typically resolve ambiguity or uncertainty with ease, as we are adept at comprehending information from various modalities, such as speech, text, and visual cues. In recent times, deep learning algorithms have been employed to tackle the task of Speech Emotion Recognition (SER), as evidenced by previous studies [1], [2], and [3].

The rise of deep learning has led many practitioners to rely solely on the power of these models, neglecting the use of domain knowledge to create meaningful features and develop models that are both effective and interpretable. This study examines the impact of hand-crafted features on SER and compares the performance of lighter machine learning models to data-intensive deep learning models. To improve the ability to resolve uncertainties, we also integrate features from the textual modality and explore the correlation between different modalities. We approach our task as a multi-class classification problem and employ two classes of models. Hand-crafted features are initially extracted from the time domain of the speech in the dataset and used to train the corresponding models in both approaches.

The first approach involves training traditional machine learning classifiers, such as Random Forests, Gradient Boosting, Support Vector Machines, Naive Bayes, and Logistic Regression. In the second method, emotions are recognized based on audio signals through the development of a Multi-Layer Perceptron and the use of an LSTM [4] classification algorithm.Different settings are employed to evaluate the models on the IEMOCAP [5] dataset, including Audio-only, Text-only, and a combination Audio and Text.The source code for the experiment can be found on Githubat the following link
https://github.com/stewartfohlo/speech_recognition_experiment .

The paper is structured as follows: Section II presents an overview of existing methods in the literature for speech emotion recognition. Section III provides information about the dataset used in this study and the pre-processing procedures carried out before feature extraction. Section IV outlines the proposed models and their implementation details. Results are presented in Section V, followed by the conclusion and suggestions for future work in Section VI.

II. Literature Review

This section presents a literature review of research carried out in the field of Speech Emotion Recognition (SER). Although not a new task, it has been extensively studied in the literature for a considerable time. Initially, a majority of the early approaches ([6] [7]) utilized Hidden Markov Models (HMMs) [8] for identifying emotions from speech. Recently, with the introduction of deep neural networks to the field, the state-of-the-art performance has significantly improved. For example, [3] and [9] utilize recurrent autoencoders to tackle the task. In addition, several different techniques have been suggested to integrate characteristics and features from various modalities in an efficient manner, such as Tensor Fusion Networks [10] and Low-Rank Matrix Multiplication [11], rather than simply combining them.

The objective of this work is to conduct a comparative analysis between 1) end-to-end deep learning models, and 2) lighter machine learning and deep learning models trained using hand-crafted features. Additionally, we explore the information present in multiple modalities and investigate how their combination affects performance.

III. The Dataset

This study utilizes the IEMOCAP dataset [5], which contains recorded conversations from ten speakers across five sessions. The dataset includes approximately twelve hours of audio and visual information along with transcriptions and is annotated with eight categorical emotion labels: anger, happiness, sadness, neutral, surprise, fear, frustration, and excited. Additionally, it includes dimensional labels for the activation and valence values from 1 to 5, which are not used in this study. The dataset has already been divided into multiple utterances per session, and each utterance file was further split into individual wav files for each sentence using the provided start and end timestamps. This resulted in approximately ten thousand speech audio files, which were used for feature extraction and classification with the two proposed models.

IV. Methodology

In this segment, the steps for pre-processing the data are outlined, along with a thorough explanation of the features that were extracted and the two models that were utilized to address the classification issue.

Classification	Number of audio files
Sad	2327
Neutral	1385
Happy	1309
Angry	860
Fear	1007
Surprised	949

Table 1: Number of audio files for each class

1). Data Pre-Processing

i. *Audio:*After conducting a preliminary frequency analysis, it was discovered that the dataset was imbalanced, with "fear" and "surprise" being under-represented. To address this issue, up-sampling techniques were employed. Furthermore, examples from the "happy" and "excited" classes were merged into the "happy" class as they closely resembled each other, and the "happy" class was also under-represented. Additionally, examples that were classified as "others" were discarded as they were labelled as ambiguous even for humans. Following these operations, a total of 7837 audio files were obtained. Table I shows the final sample distribution for each emotion.

ii. *Text:*The transcriptions initially underwent a normalization process where they were converted to lowercase and any special symbols were eliminated.

2). Feature Extraction

i. *Pitch:*The significance of pitch lies in the fact that the waveforms generated by our vocal cords can vary according to our emotional state. Various algorithms have been developed to determine the pitch signal, with the autocorrelation of center-clipped frames being the most widely used method [12]. To clarify, the input signal y[n] is modified by center-clipping to produce a resultant signal, yclipped[n]:

$$y_{clipped}[n] = \begin{cases} y[n] - C_l, & \text{if } y[n] \geq C_l \\ 0, & \text{if } |y[n]| < C_l \\ y[n] + C_l, & \text{if } y[n] \leq C_l \end{cases} \quad (1)$$

As a rule, Cl; is approximately equal to half of the input signal's mean, and the brackets notation [·] indicates that the input signal is discrete. Autocorrelation is then computed for the resulting yclipped signal, which is subsequently normalized and used to identify peak values corresponding to the pitch of the original input signal y[n]. The utilization of center-clipping on the input signal was determined to yield clearer autocorrelation peaks.

ii. *Harmonics:*When expressing anger or undergoing stressful situations, there may be supplementary excitation signals present besides pitch ([13], [14]). These additional signals can be observed in the spectrum as harmonics and cross-harmonics, as shown in Figure 1. To determine the harmonics, we employ a median-based filter method outlined in [15]. Initially, a median filter is generated for a specific window size, denoted as l:

$$y[n] = median(x[n-k : n+k] | k = (l-1)/2) \quad (2)$$

where the value of l is always an odd number. However, in instances where l is even, the median is calculated as the average of the two middle values in the sorted list. Using this filter, we process the h-th frequency slice, denoted as Sh, of a given spectrogram S. This results in a harmonic-enhanced frequency slice, denoted as Hh:

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

In this case I is the I-th step, Iharm is the harmonic filter and M is the median filter.

iii.) *Speech Energy:*By measuring the energy of a speech signal, which is closely linked to its volume, we can detect specific emotions. The disparity in energy levels between an "angry" signal and a "sad" signal is demonstrated in Figure 2. To quantify the speech energy, we utilize the conventional Root Mean Square Energy (RMSE) representation, which is calculated using the following equation:

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

The calculation of RMSE is conducted on a frame-by-frame basis, and we extract both the mean and standard deviation as features.

iv.) *Pause:*We employ this feature to indicate the "silent" segment within the audio signal, which has a direct correlation to our emotional state. For example, when we are feeling excited, such as when we are angry or happy, we tend to speak rapidly, resulting in a low Pause value. The feature value is determined by the following expression:

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

where t indicates a carefully selected threshold of approximately 0.4 * E, and representing the RMSE.

v.) *Central Moments:*In conclusion, we utilize the average and standard deviation of the signal's amplitude to encompass a condensed overview of the input.

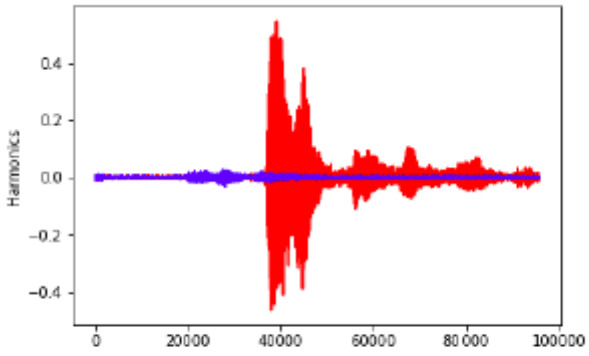


Fig. 1: Harmonics of angry (red) and sad (blue) audio signals

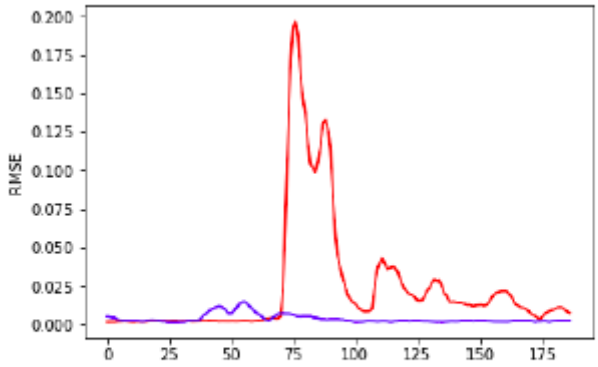


Fig. 2: RMSE plots of angry (red) and sad (blue) audio signals

3.) Text Features

i.) *Term Frequency-Inverse Document Frequency (TFIDF) :*This is a numerical metric that reveals the association between a word and a document in a corpus or collection. It comprises two components:

- *Term Frequency:*This represents the frequency of a word or token's occurrence in a document. The most straightforward approach is to use the raw count of a token in a document (such as sentences, in our specific case).
- *Inverse Document Frequency:*To mitigate the impact of commonly used language words such as "the," "a," and "an," this phrase is employed. Typically, idf for a term t and document D is defined as follows:

$$idf(t, D) = \log \frac{N}{|d \in D : t \in d|} \qquad (6)$$

The denominator demonstrates the number of documents that include the term t, and N represents the overall number of documents.

Ultimately, the TFIDF score for a term is obtained by multiplying the values of its TF and IDF.

4.) Machine Learning Models

In this section, we present a description of the machine learning-based classifiers employed in our work, which are, (XGB) Gradient Boosting, Support Vector Machines (SVM), Naive-Bayes, Random Forests (RF) and Logistic Regression.

a.) *Random Forest (RF):* RF utilizes an ensemble learning approach by creating numerous decision trees during training and returning the class that is the mode of the classes of individual trees. The RF algorithm employs two core principles: each decision tree predicts using a random subset of features, and each decision tree is trained with a subset of training samples via bootstrap aggregating. Finally, a majority vote of all the decision trees is taken to predict the class of a given input [16, 17].

b.) *Gradient Boosting (XGB):*XGB stands for eXtreme Gradient Boosting, which is a boosting algorithm that is implemented to train the model in a fast and parallelized way. Boosting is an ensemble classifier that combines a number of weak learners, typically decision trees, in a sequential manner using forward stagewise additive modeling. During the initial iterations, simple decision trees are learned, and as training proceeds, the classifier becomes more powerful because it focuses on instances where previous learners made errors.Upon completion of the training, the ultimate prediction is determined by a linear combination of the outputs from each of the individual learners, with weights assigned to each based on their performance [18].

c.) *Support Vector Machines (SVMs):*SVMs are machine learning models that utilize associated learning algorithms to classify and perform regression analysis on data. In the context of SVM training, a non-probabilistic binary linear classifier is constructed (although probabilistic classification can be achieved with methods such as Platt scaling [19]). The training process involves representing each example in the data as a point in space and mapping them in a way that maximizes the separation between the different categories with a clear gap (usually by minimizing the hinge loss).SVMs were initially designed for linear classification but can efficiently handle non-linear classification by employing the kernel trick [20], which implicitly maps their inputs into high-dimensional feature spaces.

d.) *Multinomial Naive Bayes (MNB):* Multinomial Naive Bayes is a member of the Naive Bayes family of classifiers, which are "probabilistic classifiers" that use Bayes' theorem under strong (naive) independence assumptions between features. When in a multinomial context, the feature vectors represent the occurrences of certain events, generated by a multinomial distribution (p1; : : : ; pn), where pi represents the probability of event I occurring.MNB is widely used in text-based document classification tasks [21], which involve multi-class classification problems.

e.) *Logistic Regression (LR):* Logistic Regression is typically used for binary classification problems [22], which have only two labels. In this work, LR is applied using a one versus rest approach, whereby 6 classifiers are trained for each class, and the predicted class with the highest probability is selected.

After training the aforementioned classifiers, an ensemble of the highest performing classifiers is utilized to compare against the current state-of-the-art for emotion recognition on the IEMOCAP dataset.

5.) Deep Learning Models

This section outlines the deep learning models that were used. Deep Neural Networks (DNNs) are typically trained end-to-end, allowing them to independently determine features. However, this approach can be time-consuming and computationally intensive. To reduce computational overhead, we directly input handcrafted features into these models and compare their performance with traditionally trained end-to-end models. We implemented two models in this study: the Multi-Layer Perceptron (MLP) and the Long Short-Term Memory (LSTM).

a.) Multi-Layer Perceptron (MLP): The Multi-Layer Perceptron (MLP) is a type of feed-forward neural network with at least three nodes: an input, a hidden, and an output layer. All nodes are connected to a non-linear activation function to stabilize the network during training. As the number of hidden layers increases, their expressive power improves to a certain extent. MLPs can distinguish data that is not linearly separable due to their non-linear nature.

b.) Long Short-Term Memory (LSTM): The Long Short-Term Memory (LSTM) was developed to capture long-range context in sequences. Unlike MLPs, it has feedback connections that allow it to decide which information is important. It comprises a gating mechanism with three types of gates: input, forget, and output. The equations for these gates are provided below:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (7)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (8)$$

$$o_t = \sigma_g(W_o x_t + U_o g_h t - 1 + b_o) \quad (9)$$

$$c_t = f \cdot c_{t-1} + i_t \cdot \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (10)$$

$$h_t = o_t \cdot \sigma_h(c_t) \quad (11)$$

The LSTM equations use initial values of c0 = 0 and h0 = 0. The dot (.) represents the element-wise product, t represents the time step, and xt refers to the input vector for the LSTM unit. The forget gate's activation vector is denoted by ft, the input gate's activation vector is denoted by it, and the output gate's activation vector is denoted by ot. The hidden state vector, ht, is typically used to map a vector from the feature space to a lower-dimensional latent space. The cell state vector is denoted by ct, and the weight and bias matrices (W;U and b) must be learned during training. Figure 3 shows that an LSTM cell can track hidden states at all time steps through the feedback mechanism.

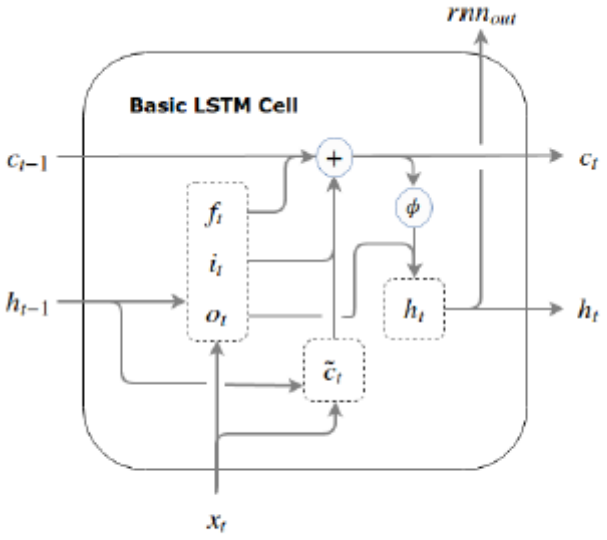


Fig. 3: Visualization of an LSTM cell

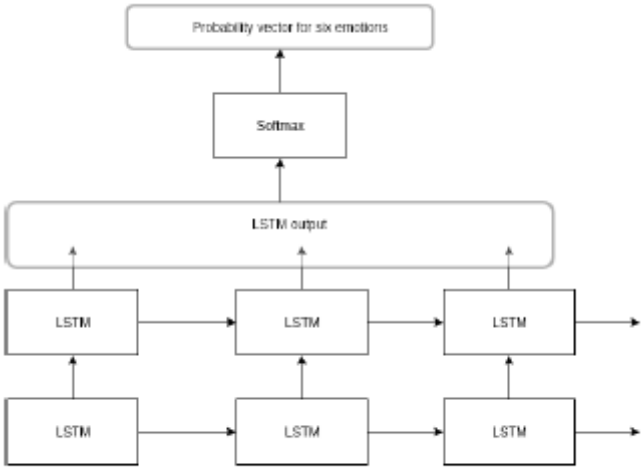


Fig. 4: LSTM classifier

The network used in this study is depicted in Figure 4. We input feature vectors into the network and apply a softmax layer to the output of the LSTM network, generating probability scores for the six emotion classes. Because we use feature vectors as input, there is no need for a decoder network to transform hidden to output space, thereby reducing the network's size.

Models	Accuracy	F1	Precision	Recall
RF	56.0	**56.0**	57.2	**57.3**
XGB	55.6	**56.0**	56.9	56.8
SVM	33.7	15.2	17.4	21.5
MNB	31.3	9.1	19.6	17.2
LR	33.4	14.9	17.8	20.9
MLP	41.0	36.5	42.2	35.9
LSTM	43.6	43.4	53.2	40.6
ARE (4-class)	56.3	-	54.6	-
E1 (4-class)	56.2	45.9	**67.6**	48.9
E1	**56.6**	55.7	57.3	**57.3**

a.) Audio setting

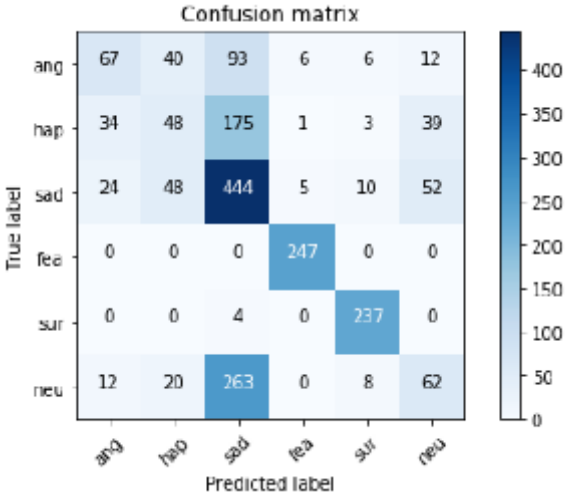
Models	Accuracy	F1	Precision	Recall
RF	62.2	60.8	65.0	62.0
XGB	56.9	55.0	70.3	51.8
SVM	62.1	61.7	62.5	**63.5**
MNB	61.9	62.1	**71.8**	58.6
LR	64.2	64.3	69.5	62.3
MLP	60.6	61.5	62.4	63.0
LSTM	63.1	62.5	65.3	62.8
TRE (4-class)	**65.5**	-	63.5	-
E1 (4-class)	63.1	61.4	**67.7**	59.0
E2	64.9	**66.0**	71.4	63.2

b.) Text Setting

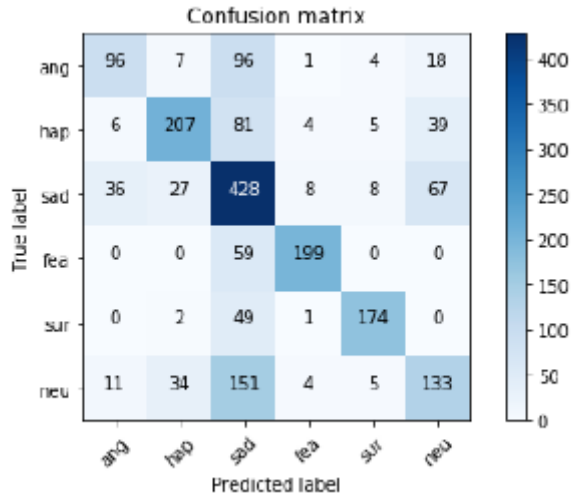
Models	Accuracy	F1	Precision	Recall
RF	65.3	65.8	69.3	65.5
XGB	62.2	63.1	67.9	61.7
SVM	63.4	63.8	63.1	65.6
MNB	60.5	60.3	70.3	57.1
MLP	66.1	68.1	68.0	69.6
LR	63.2	63.7	66.9	62.3
LSTM	64.2	64.7	66.1	65.0
MDRE (4-class)	**75.3**	-	71.8	-
E1 (4-class)	70.3	67.5	**73.2**	65.5
E2	70.1	**71.8**	72.9	**71.5**

c.) Audio and Text setting

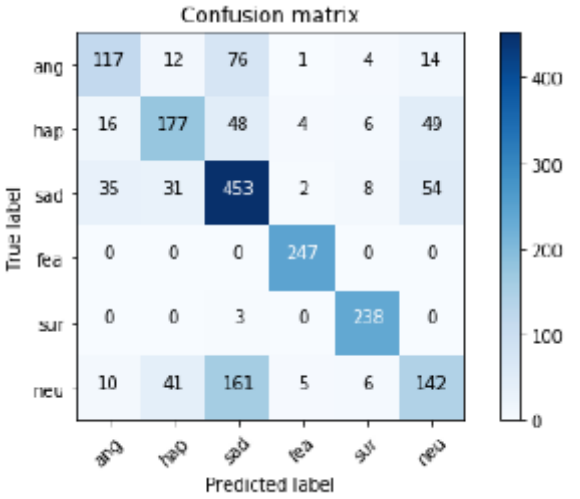
Fig. 5: The above results show performance of models where E1 is a combination of MLP, XGB and RF algorithms. E2, which is a combination of LR, MNB, MLP, XGB and RFalgorithms.



E1, Audio Setting



E2, Text Setting



E2: Audio and Text Setting

Fig. 6: Shows Confusion Matrices of the models where ensemble E1 is a combination of MLP, XGB and RF algorithms and ensemble E2 being a combination of MNB, RF, XGB and LR algorithms.

6.) Experiments carried out

The following is a description of the three experimental settings we utilized:

- Audio only: Our classifiers were trained solely on the audio feature vectors that were previously described.
- Text only: Our classifiers were trained only on the text feature vectors (TFIDF vectors).
- Audio and Text: In this setting, we combined the feature vectors from both modalities. There are various methods available for effectively fusing vectors from multiple modalities; however, we simply merged the feature vectors from audio and text to produce the combined feature vectors. The aim of this experiment was to determine the amount of information contained within each modality and to investigate how combining the feature vectors influenced the model's performance.

7.) Implementation steps

This section provides an overview of the implementation details utilized in this study.

- The LSTM classifiers discussed earlier were implemented using PyTorch [26].
- To regularize the hidden space of the LSTM classifiers and enhance network robustness, we incorporated a shut-off mechanism called dropout [27]. Dropout involves excluding a fraction of neurons from the final prediction, thereby preventing overfitting
- For the machine learning classifiers (RF, XGB, SVM, MNB, and LR) and the MLP, we employed the Python libraries scikit-learn and xgboost [24] [25].
- To process the audio files and extract features, we employed the Python library librosa [23].

To ensure a fair comparison, we randomly divided our dataset into train (80%) and test (20%) sets, maintaining the same split for all experiments. The LSTM classifiers were trained on an Intel Core vPro CPUs to expedite processing. Training was stopped when no improvement in validation performance was observed for more than 10 epochs. Each epoch refers to one iteration over all the training samples. Different batch sizes were employed for different models. For detailed hyperparameters of all models in the three experiment settings, please refer to the released repository.

8.) Evaluation of Metrics

In this section, we begin by providing an overview of the evaluation metrics utilized and present the results for the three experiment settings.

a) Accuracy: This metric represents the percentage of correctly classified test samples.

b) Precision: This measure indicates the proportion of correct predictions out of all the predictions made, considering the ground truth (i.e., labels). Below is the formula:

$$Precision = \frac{tp}{tp + fp} \qquad (12)$$

c) Recall: This measure reflects the number of correct labels present in the predicted output. Below is the formula:

$$Precision = \frac{tp}{tp + fn} \qquad (13)$$

In the formulas, tp, fp, and fn represent true positive, false positive, and false negative, respectively. These values can be derived from the confusion matrix.

d) F-score: This metric is defined as the harmonic mean of precision and recall. Unlike accuracy, F-score provides a more normalized measure of a model's predictive power.

To evaluate the performance of our best models, we compare them with the current state-of-the-art models mentioned in [2]. The state-of-the-art models employ three types of recurrent encoders: Audio-, Text-, and Multimodal Dual-Recurrent Encoders, referred to as ARE, TRE, and MDRE, respectively. It is important to note that [2] focuses on classifying four emotions, namely angry, happy, sad, and neutral, whereas our study involves six emotions. To ensure a fair comparison between our method and theirs, we also conduct experiments considering the four classes (models with code 4-class in Figure 5).

V. Results

In this section, we present the performance analysis of the models described in Section IV. Based on Figure 5, it is evident that our simpler and lighter ML models either outperform or achieve comparable results to the heavier state-of-the-art models on this dataset.

a) Audio only results:
The results for this setting are particularly intriguing. The performance of LSTM and ARE highlights that deep models require extensive information to learn features, as the LSTM classifier trained on eight-dimensional features exhibits significantly lower accuracy compared to the end-to-end trained ARE. However, neither of them surpasses the lighter E1 model (Ensemble of RF, XGB, and MLP) trained on the eight-dimensional audio feature vectors. Examination of the confusion matrix (Fig. 6a) reveals that the most challenging aspect for the model is detecting "neutral" or distinguishing between "angry," "happy," and "sad" emotions.

b) Text only results:
We observe that the performance of all the models in this setting is similar. This can be attributed to the effectiveness of TFIDF vectors in capturing word-sentence correlation. The confusion matrix (Fig. 6b) demonstrates that our text-based models, along with the end-to-end trained TRE, are capable of distinguishing the six emotions quite well. However, it is worth noting that identifying "sad" emotions is relatively more challenging for textual features.

c) Audio and Text results:
Combining audio and text features provides a significant approx. 15% improvement across all metrics. This clearly indicates the strong correlation between text and speech features. It is noteworthy that in this case, the recurrent encoders exhibit slightly better accuracy, albeit at the expense of precision. The lower performance of E1 can be attributed to the simplistic fusion method (concatenation) used, as simple concatenation for an ML model may still retain many modality-specific connections instead of desired inter-modal connections. The promising outcome here is that the fusion of features from both modalities effectively resolves the ambiguity observed in modality-specific models, as depicted in Fig. 6c. We can conclude that textual features aid in the accurate classification of "angry" and "happy" classes, while audio features enhance the model's ability to detect "sad" emotions.

Overall, it can be concluded that our simple ML methods demonstrate remarkable robustness by achieving comparable performance, despite being designed to predict six classes instead of four in previous works.

A. Key features to explore when addressing uncertainty in speech emotion recognition:
In this section, we explore the dominant factors influencing predictions in this classification task. We specifically selected the XGB model for this analysis and ranked the eight audio features. Our findings reveal that the Harmonic feature, which directly correlates with signal excitation, has the greatest impact. Surprisingly, the "silence" feature, associated with pauses, is nearly as influential as the standard deviation of the autocorrelated signal (linked to pitch). The relatively minor contribution of central moments is understandable, considering the diverse nature of the signal, as a global or coarse feature would struggle to capture its subtleties.

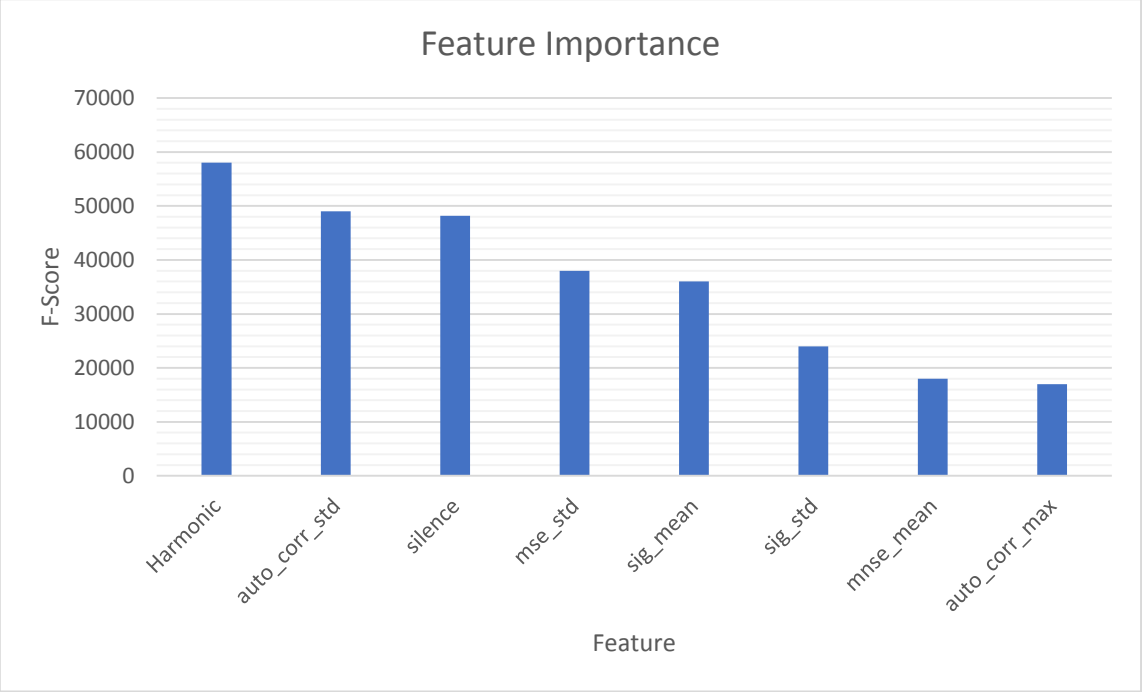


Fig. 7: Most critical audio features for speech emotion recognition

VI. Conclusion and Future Work

In this study, our focus is on speech emotion recognition and examining how different modalities contribute to resolving ambiguity using the IEMOCAP dataset. We compare Machine Learning (ML) and Deep Learning (DL) models and demonstrate that even lighter and more interpretable ML models can achieve performance comparable to DL models. Furthermore, we highlight that ensembling multiple ML models can lead to performance improvements. Our feature extraction process primarily involves selecting a limited set of time-domain features from audio signals. However, incorporating additional frequency-domain features such as Mel-Frequency Cepstral Coefficients (MFCC), Spectral Roll-off, and additional time-domain features like Zero Crossing Rate (ZCR) could enhance the richness of the audio feature space. Furthermore, exploring advanced fusion methods such as TFN and LMF for combining speech and text vectors could improve effectiveness. It would also be intriguing to examine how the performance of ML models versus DL models scales with the inclusion of more data.

References

[1]. Z. Huang, M. Dong, Q. Mao, and Y. Zhan, "Speech emotion recognition using Convolutional Neural Networks (CCN)," in Proceedings of the 22nd ACM international conference on Multimedia, pp. 801–804, ACM, 2014.

[2]. S. Yoon, S. Byun, and K. Jung, "Multimodal speech emotion recognition using audio and text," in 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 112–118, IEEE, 2018.

[3]. K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in 15th annual conference of the international speech communication association, 2014.

[4]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780,- 1997.

[5]. C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," Language resources and evaluation, vol. 42, no. 4, p. 335, 2008.

[6]. A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. Mari˜no, "Speech emotion recognition using hidden markov models," in 7th Conference on Speech Communication and Technology, 2001.

[7]. B. Schuller, G. Rigoll, and M. Lang, "Hidden Markov model-based speech emotion recognition," in 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., vol. 2, pp. II–1, IEEE, 2003.

[8]. L. R. Rabiner and B.-H. Juang, "An introduction to hidden markov models," ieeeassp magazine, vol. 3, no. 1, pp. 4–16, 1986.

[9]. Y. Kim, H. Lee, and E. M. Provost, "Deep learning for robust feature generation in audiovisual emotion recognition," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3687–3691, IEEE, 2013.

[10]. A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor fusion network for multimodal sentiment analysis," arXiv preprint arXiv:1707.07250, 2017.

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

[12]. M. Sondhi, "New methods of pitch extraction," IEEE Transactions on audio and electroacoustics, vol. 16, no. 2, pp. 262–266, 1968.

[13]. H. Teager and S. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," in Speech production and speech modelling, pp. 241–261, Springer, 1990.

[14]. G. Zhou, J. H. Hansen, and J. F. Kaiser, "Nonlinear feature based classification of speech under stress," IEEE Transactions on speech and audio processing, vol. 9, no. 3, pp. 201–216, 2001.

[15]. D. Fitzgerald, "Harmonic/percussive separation using median filtering," 2010.

[16]. Y. Amit, D. Geman, and K. Wilder, "Joint induction of shape features and tree classifiers," IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 11, pp. 1300– 1305, 1997.

[17]. L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123–140, 1996.

[18]. J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning, vol. 1. Springer series in statistics New York, 2001.

[19]. J. Platt et al., "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," Advances in large margin classifiers, vol. 10, no. 3, pp. 61– 74, 1999.

[20]. C. Cortes and V. Vapnik, "Support-vector networks," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[21]. A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," in Australasian Joint Conference on Artificial Intelligence, pp. 488–499, Springer, 2004.

[22]. G. King and L. Zeng, "Logistic regression in rare events data," Political analysis, vol. 9, no. 2, pp. 137–163, 2001.

[23]. B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in Proceedings of the 14th Python in Science Conference, pp. 18–25, 2015.

[24]. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., "Scikit-learn: Machine learning in python," No. Oct, pp. 2825–2830, 2011.

[25]. T. Chen, "Scalable, portable and distributed gradient boosting (gbdt, gbrt or gbm) library, for python, r, java, scala, c++ and more. runs on single machine, hadoop, spark, flink and dataflow," 2014.

[26]. Facebook, "Pytorch," 2017.

[27]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.

[28]. S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE transactions on acoustics, speech, and signal processing, vol. 28, no. 4, pp. 357–366, 1980.

[29]. F. Gouyon, F. Pachet, O. Delerue, et al., "On the use of zero-crossing rate for an application of classification of percussive sounds," in Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy, 2000.