



uses the audio file and classes them into male or female with the emotion using MFCC feature.

### C. Problem Objective

The aim of this project is to eliminate the Unprepared Interview phase of every student and make them ready to face any interview. The aspects used for checking and calculating the performance of the user in any interview are selected and measured using various ML models which give an accurate response to users response and answers.

This project helps out various number of students who struggle to face a interview and helps them to build confidence on the way they develop themselves ready foran interview..

## II. PROPOSED SYSTEM

### A. Existing Systems

The aim of the work in the base paper is to recognize and analyze the emotions displayed by the candidates using Convolution Neural Networks (CNN) to detect the confidence levels of a person. To improve the precision of the project, eye blink rate calculation to detect anxiety and eye gaze tracking to detect distraction are employed. The results are consolidated and presented to the candidate as a report which can effectively assist the interview candidates to prepare for their one on one interview preparations.

This uses a Facial Emotion Recognition Model which maps the data of the face of the user using facial points from Haar Cascades. The data from these facial points is collected for predicting emotion of the person in front of the camera.

### B. Proposed Model

The proposed model also uses emotion detection as one of the key features for interview analysis also while adding sentiment analysis as another feature. An Emotion Detection Model through Speech is used for predicting the Emotions of a person instead of a Facial Emotion Recogniser because of faster processing and accuracy SER models actually have.

The Emotional Detection Model is done with the help of Convolutional Neural Networks (CNN) and was achieved an accuracy of 72.2%. This helps in classifying both male and female voices with a set of 5 emotions for each user. The Sentiment analysis is done with the help of TextBlob package. This finds out various values like polarity, generality etc. when a string is passed an argument to its constructor. Using Polarity Value, we can find the Sentiment of the answer given by the user. We also have a scoring function which helps in identifying the correct answers and wrong answers which helps in judging the knowledge of the

user. These three features together contribute to Interview Analysis of the User.

### C. Datasets used

#### 1. SAVEE –

SAVEE (Surrey Audio-Visual Expressed Emotion)is an emotion recognition dataset. It consists of recordings from 4 male actors in 7 different emotions, 480 British English utterances in total. The sentences were chosen from the standard TIMIT corpus and phonetically-balanced for each emotion. This release contains only the audio stream from the original audio-visual recording. The data is split so that the training set consists of 2 speakers, and both the validation and test set consists of samples from 1 speaker, respectively.

#### 2. RAVDESS –

The RAVDESS is released under a Creative Commons Attribution license, so please cite the RAVDESS if it is used in your work in any form. Published academic papers should use the academic paper citation for our PLoS1 paper. Personal works, such as machine learning projects/blog posts, should provide a URL to this Zenodo page, though a reference to our PLoS1 paper would also be appreciated.<sup>[7]</sup>

## III. DESIGN

### A. Use Case Diagram

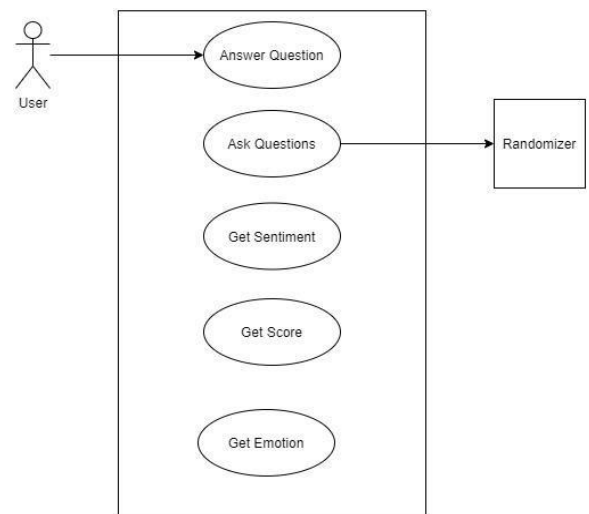
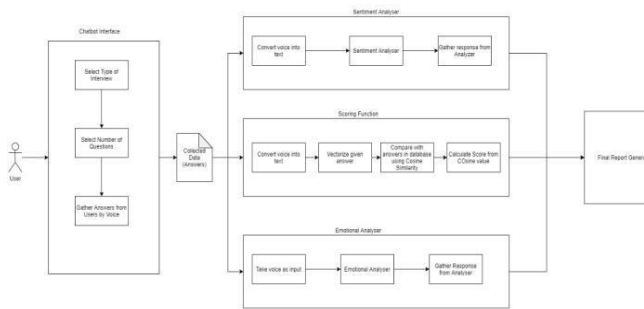


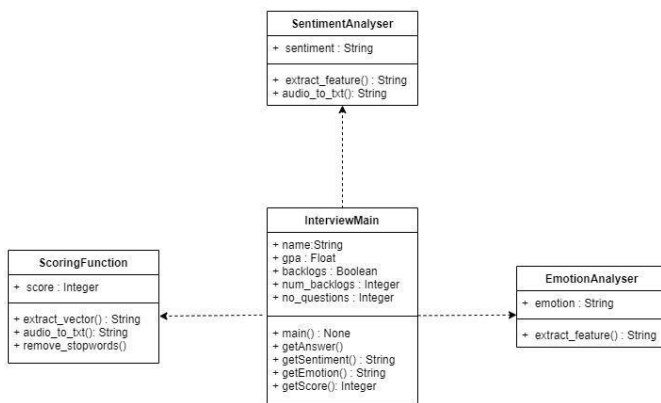
Fig.1 : Use Case Diagram

**B. Architecture Diagram:**



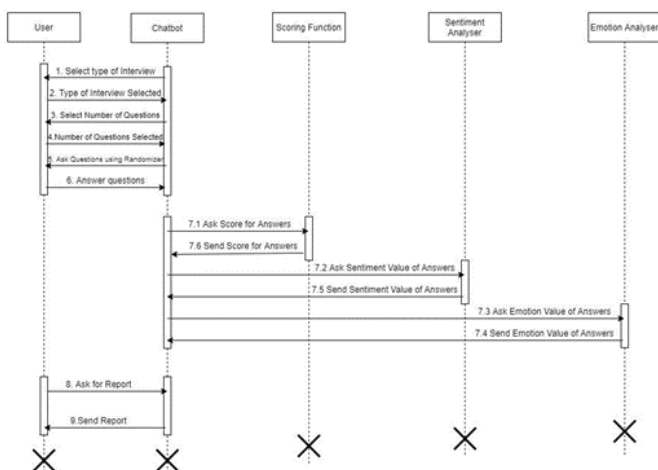
**Fig.2 : Architecture Diagram**

**C. Class Diagram:**



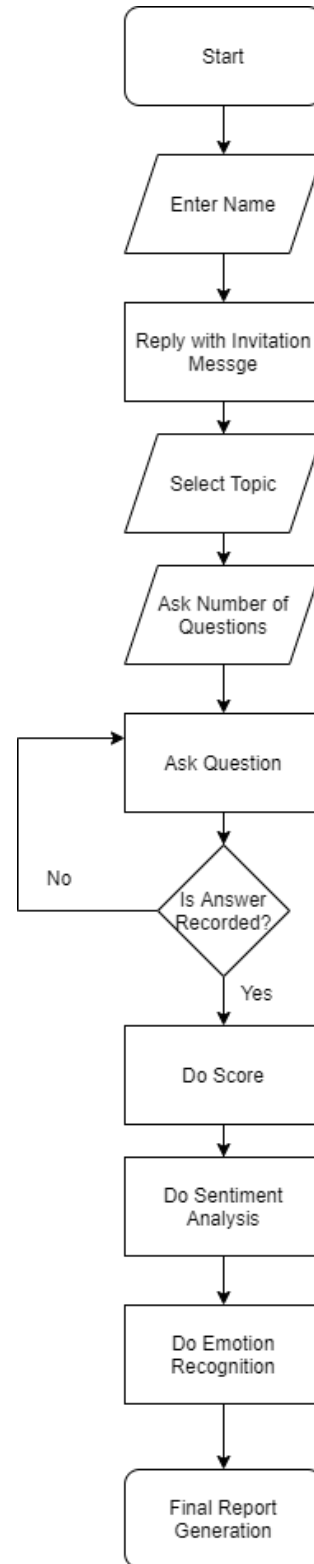
**Fig.3 : Class Diagram**

**D. Sequence Diagram:**



**Fig.4 : Sequence Diagram**

**E. Flow Chart:**



**Fig.5 : Flow Chart**

F. Collaboration Diagram:

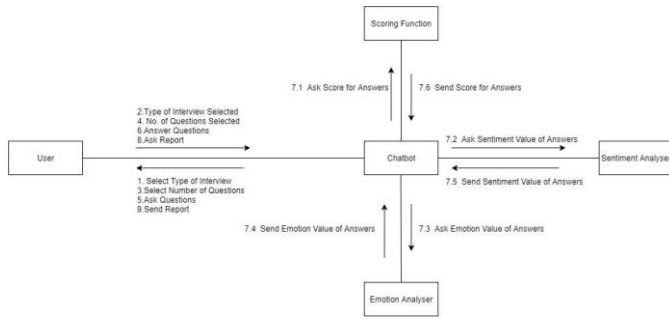


Fig.6 : Collaboration Diagram

IV. IMPLEMENTATION

The Implementation of the Interview Analysis Module is done with the help of three different systems like Sentiment Analysis, Emotion Recognition and Scoring Function.

A. Sentiment Analyser

Sentiment analysis is done with the help of a Python package called TextBlob which is used to calculate various values for a given statement.. A string is passed as argument to the constructor of the TextBlob Class and various values are generated for the string like polarity, generality etc. The polarity values shows about the sentimental value of the statement. The value rises in between -1.0 to +1.0 . A negative value implies a Negative sentiment and a positive value implies a Positive Statement whereas a neutral statement could imply both.

B. Emotion Recognizer

Detecting emotions is one of the most important marketing strategy in today’s world. You could personalize different things for an individual specifically to suit their interest. For this reason, we decided to do a project where we could detect a person’s emotions just by their voice which will let us manage many AI related applications. Some examples could be including call centers to play music when one is angry on the call. Another could be a smart car slowing down when one is angry or fearful. As a result this type of application has much potential in the world that would benefit companies and also even safety to consumers.

We built a Multi Perceptron model, LSTM model and CNN models. The MLP and LSTM were not suitable as it gave us low accuracy. As our project is a classification problem where we were categorize the different emotions, CNN worked best for us.

CNN model was the best for our classification problem. After training numerous models we got the best validation accuracy of 60% with 18 layers, softmax activationfunction, rmsprop activation function, batch size of 32 and 1000 epochs

In [51]: `model.summary()`

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 216, 128)	768
activation_1 (Activation)	(None, 216, 128)	0
conv1d_2 (Conv1D)	(None, 216, 128)	82048
activation_2 (Activation)	(None, 216, 128)	0
dropout_1 (Dropout)	(None, 216, 128)	0
max_pooling1d_1 (MaxPooling1D)	(None, 27, 128)	0
conv1d_3 (Conv1D)	(None, 27, 128)	82048
activation_3 (Activation)	(None, 27, 128)	0
conv1d_4 (Conv1D)	(None, 27, 128)	82048
activation_4 (Activation)	(None, 27, 128)	0
conv1d_5 (Conv1D)	(None, 27, 128)	82048
activation_5 (Activation)	(None, 27, 128)	0
dropout_2 (Dropout)	(None, 27, 128)	0
conv1d_6 (Conv1D)	(None, 27, 128)	82048
activation_6 (Activation)	(None, 27, 128)	0
flatten_1 (Flatten)	(None, 3456)	0
dense_1 (Dense)	(None, 10)	34570
activation_7 (Activation)	(None, 10)	0
Total params: 445,578		
Trainable params: 445,578		
Non-trainable params: 0		

Fig.7 : Model Summary of SER Model

C. Scoring Function:

Scoring Function is used to grade the results between the answer given by the user and answer present in the database. The database used consists of various topics regarding different types of interviews in the field of computer science. A set of questions implying the topic are collected from the database along with its answer. These answers are then compared with the answer given by the user using cosine similarity and the final score for the answer given by the user is generated.

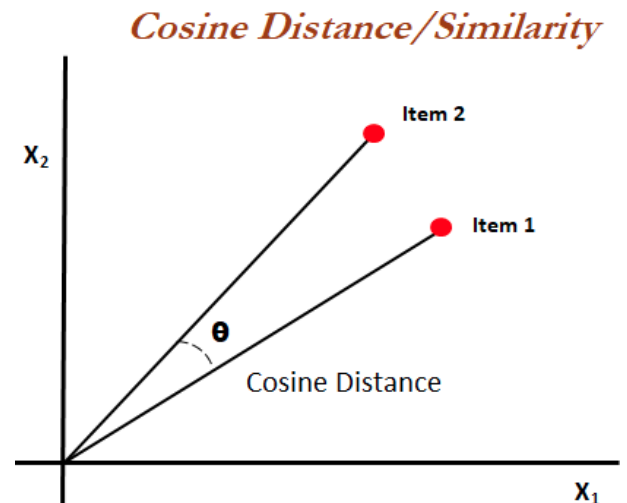


Fig.8 : Cosine Similarity of Two Items

#### D. Testing

Testing is done with the help of a module called pytest. The PyTest module helps in unit testing directly from the command line interface. This is a very short but powerful testing package which helps in unit testing and also testing multiple units at a single time.

The pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries.

```
PS X:\study_pytest> py.test
===== test session starts =====
platform win32 -- Python 3.7.0, pytest-4.0.2, py-1.7.0, pluggy-0.8.0
rootdir: X:\study_pytest, inifile:
plugins: xdist-1.25.0, forked-0.2
collected 2 items

test_sample1.py F. [100%]

===== FAILURES =====
_____ test_file1_method1 _____

@pytest.mark.set1
def test_file1_method1():
    x=5
    y=6
    assert x+1 == y,"test failed"
> assert x == y,"test failed because x=" + str(x) + " y=" + str(y)
E   AssertionError: test failed because x=5 y=6
E   assert 5 == 6

test_sample1.py:9: AssertionError
===== 1 failed, 1 passed in 0.04 seconds =====
PS X:\study_pytest>
```

Fig.9 : Sample PyTest Module Interface

#### V. RESULTS

All the three different systems when implemented together with the help parallel programming in Python generate great results. We have shown results of various modules used in our Interview analysis system individually and together.

```
[ ] # loading json and creating model
from keras.models import model_from_json
json_file = open('model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
# load weights into new model
loaded_model.load_weights("saved_models/Emotion_Voice_Detection_Model.h5")
print("Loaded model from disk")

# evaluate loaded model on test data
loaded_model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
score = loaded_model.evaluate(x_test, y_test, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], score[1]*100))

Loaded model from disk
acc: 72.73%
```

Fig.10 : Accuracy of the Model used for SER

```
Enter your name:Sugun
COMP: Hey Sugun !
COMP: Select topic you want to get interviewed
Select:
1: "OOPs Interview Questions"
2: "Python Interview Questions"
3: "Data Structures Questions"
  Topics selected:2
COMP: Enter number of Questions (max 10):
3
COMP: What are functions in Python?
USER: set of statements that are executed in a particular order to obtain a specific result
COMP: What are Python packages?
USER: a group of classes and functions which are used for later purposes
COMP: How do you write comments in python?
USER: written using inverted commas and backward slashes
1/1 [=====] - 0s 6Sms/sample
1/1 [=====] - 0s 5ms/sample
1/1 [=====] - 0s 5ms/sample
['Positive', 'Neutral', 'Neutral']
[2, 1, 3]
['male_calm', 'male_calm', 'male_calm']
```

Fig.11 : Execution of the Final Module

```
===== test session starts =====
platform win32 -- Python 3.7.7, pytest-6.2.4, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\Sugun\Wishesh\Desktop\Project Files
collected 9 items

interface.py::test_perfect_score PASSED [ 11%]
interface.py::test_regular_score PASSED [ 22%]
interface.py::test_negative_sentiment PASSED [ 33%]
interface.py::test_positive_sentiment PASSED [ 44%]
interface.py::test_neutral_sentiment PASSED [ 55%]
interface.py::test_get_question PASSED [ 66%]
interface.py::test_get_answer PASSED [ 77%]
interface.py::test_male_voice PASSED [ 88%]
interface.py::test_female_voice PASSED [100%]

===== 9 passed, 13 warnings in 4.29s =====
C:\Users\Sugun\Wishesh\Desktop\Project Files>
```

Fig 12: Execution of the Testing using Pytest Module

#### VI. CONCLUSION

The Interview Analysis System uses various models and implementation of these systems have thoroughly implemented in our system. The Sentiment Analysis using the TextBlob package helps in finding the sentiment implied by the interviewee on the interviewer while the SER model helps in detecting any tension in Interviewee answers. Also, the answers are verified with the help of our Scoring Function. This helps in preparation of students for their upcoming interviews without any issue

#### ACKNOWLEDGMENT

We would like to take this opportunity to express our deep gratitude to the members who assisted us directly and indirectly for the completion of this project work from Dhanekula Institute of Engineering and Technology.

#### REFERENCES

- [1] Emotion Detection for Interview Analysis using CNN S.Subhal , L. Paul Jasmine Rani2 , Abisha B. 3 , Afra M. 4 , International Journal of Grid and Distributed Computing Vol. 13, No. 1, (2020), pp. 1005-1010
- [2] B. Schuller, G. Rigoll and M. Lang, "Hidden Markov model-based speech emotion recognition," 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., 2003, pp. II-1, doi: 10.1109/ICASSP.2003.1202279.
- [3] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." *Proceedings of the workshop on language in social media (LSM 2011)*. 2011.
- [4] M. Parmar, B. Maturi, J. M. Dutt and H. Phate, "Sentiment Analysis on Interview Transcripts: An application of NLP for Quantitative Analysis," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1063-1068, doi: 10.1109/ICACCI.2018.8554498.
- [5] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." *Ain Shams engineering journal* 5.4 (2014): 1093-1113.
- [6] El Ayadi, Moataz, Mohamed S. Kamel, and Fakhri Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases." *Pattern recognition* 44.3 (2011): 572-587.
- [7] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PLoS ONE* 13(5): e0196391. <https://doi.org/10.1371/journal.pone.0196391>.

ABOUT THE AUTHORS

	<p><b>Padmini Chattu</b> MTech (Phd), Number of Publications: 8, ISTE, IFERP, ISSE, IAENG Dhanekula Institute of Engineering and Technology</p>
	<p><b>Sugun Wishesh Nadipalli,</b> Bachelor of Technology, Member of IFERP, Dhanekula Institute of Engineering and Technology</p>
	<p><b>Komal Raghava Polavarapu</b> Bachelor of Technology, Dhanekula Institute of Engineering and Technology</p>
	<p><b>Chinmayee Nasika</b> Bachelor of Technology, Dhanekula Institute of Engineering and Technology</p>
	<p><b>Tummala Sri Sahithi</b> Bachelor of Technology, Dhanekula Institute of Engineering and Technology</p>