Audio Based Sentiment Prediction Model

Dr. K. Anupriya[1] , Eduvulapati Chandana Sravya[2], Gudibandi Likitha Reddy[3], Jeldi Sathvika[4], Dr. K. Lavanya[5]

[1,5] Associate Professor, [2-4] B tech Students,
Lakireddy Bali Reddy college of Engineering, Mylavaram, JNTUK, AP, India.
Email id: smartykoneru@gmail.com , dr.lavanyakampa@gmail.com

Article Info
Page Number: 209 – 227
Publication Issue:
Vol. 71 No. 3 (2022)

Abstract

When something gets newly introduced among the people, it is important to know about people's opinion on that and this is where sentiment analysis comes into play. Model that's built for sentiment analysis helps to predict the sentiment of the content. The audios that we have collected are taken as the dataset for the system and our proposed model mainly overcomes the language barrier as the system involves converting of other language audio reviews to English text. The preprocessing, feature extraction are applied on the text before training the model. The data is split into testing and training data based on testsize parameter. Initially, we have carried out training phase using three supervised machine learning algorithms i.e., SVM, Random Forest, Multinomial NB and one neural network based deep learning algorithm LSTM used with RMSprop, Adamax optimizers. The training and stratified sampling is done and we have chosen the best algorithm based on the average of the accuracy results produced by system when performed against various test samples. SVM, Random Forest, Multinomial NB, LSTM with RMSprop, LSTM with Adamax have given accuracies of 80.02%, 77.08%, 76.49%, 72.36%, 68.24% on ration dataset and 86.67%, 81.67%, 81.67%, 74.17%, 60.84% on JVD dataset respectively. As SVM algorithm gave the best accuracy, we have selected it as the algorithm for our sentiment prediction model.

KeyWords: Support Vector Machine (SVM), Long Short Term Memory (LSTM), NaiveBayes (NB), Sentiment Analysis (SA), Random Forest Classifier (RFC), Convolutional Neural Networks (CNN), Weakly-Supervised Deep Embedding (WDE)

Article History
Article Received: *12 January 2022*
Revised: *25 February 2022*
Accepted: *20 April 2022*
Publication: *09 June 2022*

I. INTRODUCTION

Sentiment analysis or opinion mining as the name itself suggests, is used to extract the sentiment or opinion from the content referred. Various natural processing techniques and machine learning techniques can be combined together to predict the sentiment score by which the content or any review topics and can be classified under classes like positive, negative and neutral. Sentiment analysis helps to identify the sentiment or the tone conveyed in the sentence. Sentiment analysis needs to pick out and identify the proper opinions or facts from the sources. Several textual analysis techniques or preprocessing methods helps to perform the task of sentiment analysis more accurately. It is important to identify the high strength conveying sentiment words to identify correct polarity. As sentiment analysis involves opinion extraction, we can view and understand people's opinion on various things. The general examples include people giving reviews on products, which help the business unit to understand customer feelings on their products by performing analysis tasks which in turn helps them in making any changes accordingly to increase their sales. Sentiment analysis can also be helpful in identifying the important insights or patterns that have a major influence on opinions. SA is useful to attain a wider view of opinions behind the topics. Rather than just considering the likings or comments, sentiment analysis can help you to understand the actual emotion conveyed in the content. These general applications convey the importance of sentiment analysis in various fields.There is various techniques to perform sentiment analysis. For developing the sentiment classification models there are supervised machine learning algorithms (model trained with labelled data) like SVM, multinomial NaiveBayes, Random Forest etc and also unsupervised machine learning algorithms(itself identify the insights and features) like k-means clustering, neural networks etc. The sentiment prediction models can be built using any of these techniques.Now a days, social media is the widely used platform where people express their opinions on several things. There are many platforms like twitter, face book, instagram etc where people express their opinions widely in the form of tweets or any reviews. Performing analysis on millions of opinions expressed in such platforms helps you in identifying proper sentiments behind it as they provide us with large training data as well. But, the problem is that the people express their opinions generally in English texts and only educated people know how to use these platforms. The literacy percentage is India is 74.04% and literacy rate in Andhra Pradesh is 67.7% as per [1]. The remaining percentage is of illiterates i.e., uneducated people. Such uneducated people cannot express their opinions on such social platforms or cannot express their opinions in English texts which would have a greater influence or effect on the sentiment models as they are missed with opinions that may generate useful insights. There would be a major loss in opinion extraction because of such problem. The above context stated the language barrier problem as uneducated people could express their views only in their native language but not in English which is an obstacle faced in opinion expressing. There are some existing models that performed sentiment analysis on English text, Chinese text from [2], [3]. If this language barrier problem is resolved more useful insights can be generated, proper opinion extraction can be done. We have proposed a model which could overcome this language barrier problem where we have collected telugu audios from people about opinion on government schemes and used that as the dataset for our model. Most of the

people who receive benefits from schemes are illiterates who can express their opinion only in native language i.e., Telugu. Our model takes into consideration of also the reviews from illiterates which could help in ultimate opinion extraction which can also be helpful for government in improvising the schemes. The model involves conversion of other language audio review into English text which makes the language barrier problem to be resolved. For the model, we initially took algorithms like SVM, Multinomial NB, Random Forest, LSTM with RMSprop and Adamax optimizers into consideration for choosing the best suitable algorithm. We considered the average results for all the algorithms when tested against various samples and chosen SVM for training the proposed model as it has given the best results. The new model that we have proposed is trained using one of the Supervised Machine Learning Algorithm i.e., SVM which takes audio file as input and can predict the final sentiment from it.

The rest of the paper is described in the various sections which includes, The RELATED WORK is covered in Section 2. The proposed method is shown in Section 3. The results and evaluation is shown in Section 4. The study conclusion is described in Section 5.

2. RELATED WORK

Wei Zhao et al at [2], proposed a deep learning system for sentiment classification that uses ratings for initial training. The general opinion distribution from weakly tagged reviews is done, obtaining a characteristic value which attains high-level features. A neural network structure for WDE is developed using CNN and LSTM for prototyping text data. After sentence depiction, a classification layer is added which employs tagged sentences for final label prediction. Among frameworks, WDE-CNN contain less arguments than WDE-LSTM, but had difficulties in dealing long-term reliances that are well handled by WDE-LSTM, but it's less effective and needs more training data. But as a whole, WDE-LSTM gave the best accuracy of 87.9%.N. A. Osman et al at [4], proposed a recommender system using contextual info from opinion mining on reviews and likings. The system merges reviews with ratings matrix using opinion mining and also decreases data inadequacy. The conventional model can be improvised by inculcating with contextual elements. The model includes employing opinion words, quantifying positive and negative root words, applying some grammatical and contextual rules. The accuracy comparison is done between generated lexicons and general lexicons. The individual ratings are combined to obtain final rating. The RMSE and MAE are used as metrics to compare the models. Contextual information helps to avoid ambiguity. The proposed model has given the best RMSE and MAE values i.e., RMSE=4.01, MAE=3.77.Xing Fang et al at [5], proposed a system where sentiment classification is done both in terms of sentence and review levels. It entails POS tagging, negation handling, calculating sentiment score of tokenized words, based on the score, the word can be categorized under its sentiment. The system uses bag-of-words approach to count the sentiment for each word in sentence and finally classify to its label based on count value. To train the classifier, feature vector formation is done. ROC curves are drawn to select the best algorithm. In both human labelled and machine labelled texts, RFC outperformed NBC and SVM in terms of sentence-level classification. However, when it came to review-level classification, NBC and SVM outperformed RFC. On consideration of

all cases, RFC gave the best average value. Lijuan Huang et al at [6], presented polymerization topic sentiment model (PTSM) which proved that assessing sentiment covered in the reviews is crucial. A data dictionary is built that uses mutual information to recognize features, and the features having high sentiment strength are used for sentiment words. Mutual information (MI) is a feature selection strategy and based on feature frequency, they would or would not be held. The MI metric determines whether a characteristic is positive or negative. The neural network (NN) model and support vector machine (SVM) model are the prediction models. The Mean Absolute Percentage Error (MAPE), mean squared error (MSE) and mean absolute error (MAE) measures demonstrates that PTSM is more exact than any other methods.

Imane El Alaoui et al at [7], developed a concept for creating a polarity dictionary and labeling tweets based on polarity-related attributes. For constructing dictionaries, most commonly used hashtags are classified into classes and preprocessing steps applied on such hashtags to get precise dictionaries. Other preprocessing for feature selection is proposed in various studies [15-18]. On fresh tweets, the data preprocessing is performed and Balancing is done to give importance to words that have strong polarity effect. Final rating for tweet is computed based on likes and retweets. For testing the system, it is compared with Google cloud prediction API and NaiveBayes classifier. The values of accuracy, precision, recall, and F-score as an average are considered for evaluation. In comparison to NaiveBayes and Google prediction API, the classification using the introduced model had an accuracy of 90.21 percent and 89.98 percent.S. Mahalakshmi et al at [8], considered Naive Bayes, Support Vector Machines, k-Nearest Neighbor, and Decision Tree as methodologies for document-level sentiment categorization. Data analysis and extraction of sentiment words from data is a tough task mainly when it incorporates reviews from various fields. The system uses several preprocessing techniques for classifying reviews into positive and negative polarity, to make it effective in terms of accuracy and training time. To choose the best strategy, a variety of models and preprocessing techniques are used and compared. As a result, the model with POS Tagging surpasses the others, and Multinomial NB and SVM classify the model accurately.Ning Jin et al at [9], introduced a multi-task training approach for multi-task multi-scale opinion prediction that employs a multi-scale convolutional neural network (CNN) and LSTM. When reviews from various jobs are included, categorization efficiency suffers. The model classifies such multitasking reviews using a multitask learning method in which local features (private features of a specific task) and global features (shared features among tasks) are extracted first, and then local and global encoding schemes are defined in which LSTM and CNN jointly perform sentence encoding and refine the encoder using an adversial multitask learning (ATML) structure. The accuracy and f1-score metrics reveal that the proposed model outperforms others because it appropriately evaluates and handles with text characteristics.Jaehun Park at [10], devised a unique online-review-based method for assessing relative customer satisfaction with cosmetics products and interpreting the factors that influence reviewers' positive and negative attitudes. The current study used sentiment analysis and statistical data analysis to build a systematic way of evaluating relative customer satisfaction with cosmetics companies and analyzed the causes of positive and negative sentiments using Term Frequency-Inverse Document Frequency analysis. The

proposed method is believed to be a new way to replace the questionnaire survey method, which has been widely utilized for analyzing customer happiness and is currently being used by cosmetics firms to realize or increase satisfaction with the brands that people evaluate. Ruba Obiedat et al at [11], offered a new hybrid evolutionary technique for analyzing people's attitudes toward different restaurants. The proposed method gathered about 3000 restaurant ratings and classifying them with the help of crowd sourcing. The problem of imbalanced data in the dataset was solved using oversampling techniques. They also used a hybrid optimization strategy that included PSO and SVM to identify the best weights as well as the k values of four distinct oversampling techniques to predict review sentiments. The research shows that the proposed PSO-SVM method is successful and outperforms the other methods in all of the tests.

Danushka Bollegala et al at [12], looked at three limitations that an embedding that may be used to train a cross-domain sentiment classification system had to meet. Using a benchmark dataset for cross-domain sentiment classification, they investigated the performance of the individual constraints as well as their combinations. Their results reveal that some of the proposed constraint combinations produce statistically comparable results to current cross-domain sentiment classification algorithms. Their proposed solution makes use of the label information in source domain reviews to develop embeddings that are sensitive to the application's end job, sentiment categorization. Danushka Bollegala et al at [13], proposed employing an automatically derived sentiment-sensitive thesaurus to create a cross-domain sentiment classifier. To deal with the feature miss-match problem in cross-domain sentiment classification, we compute the relatedness of features and develop a sentiment-sensitive thesaurus using labeled data from various source domains and unlabeled data from source and target domains. Next, we use the thesaurus we've constructed to expand feature vectors for a binary classifier throughout the train and test phases. L1regularization is used to pick a relevant subset of characteristics. On a benchmark dataset, the proposed method greatly outperforms numerous baselines and produces good results compared to previously described cross-domain sentiment classification algorithms.

Azwa Abdul Aziz et al at [14], explained about performance of sentiment model when applied on cross domain datasets .While the results are promising especially with in domain sentiments, there is no guarantee the model provides the same performances against the real-time data due to the diversity of new data. SML performance decreases when applied to cross-domain datasets because new features are appeared in different domains. The model is proposed such that to increase the performance even when applied on the cross domain datasets and they also implemented contextual approaches which construct relationship between words and sources by constructing a tree structure identified as Hierarchical knowledge tree and they have used Multinomial Naive Bayes and Random Forest Classifier which both produced similar results in terms of average accuracy.

3. Proposed Methodology

The complete idea about the proposed method is shown in the Figure 1.It consist of the different stages which includes: Data Collection, Conversion from audio to text, Translation

into English text, Split data set, Preprocessing, Feature Extraction, Classification Model. The complete idea about the proposed method is shown in Figure 1.

3.1. Dataset Collection and Conversion of Audios to English Text

We have collected audios in telugu language from people on two government schemes and undergo text conversion and act as the dataset for our system. The conversion of audios to text is done in a series of steps:

Generally for Speech Recognition, the input files should be in wav file format in default. So we used the subprocess module in python to initiate ffmpeg conversion program which converts the audios to wav extension files.

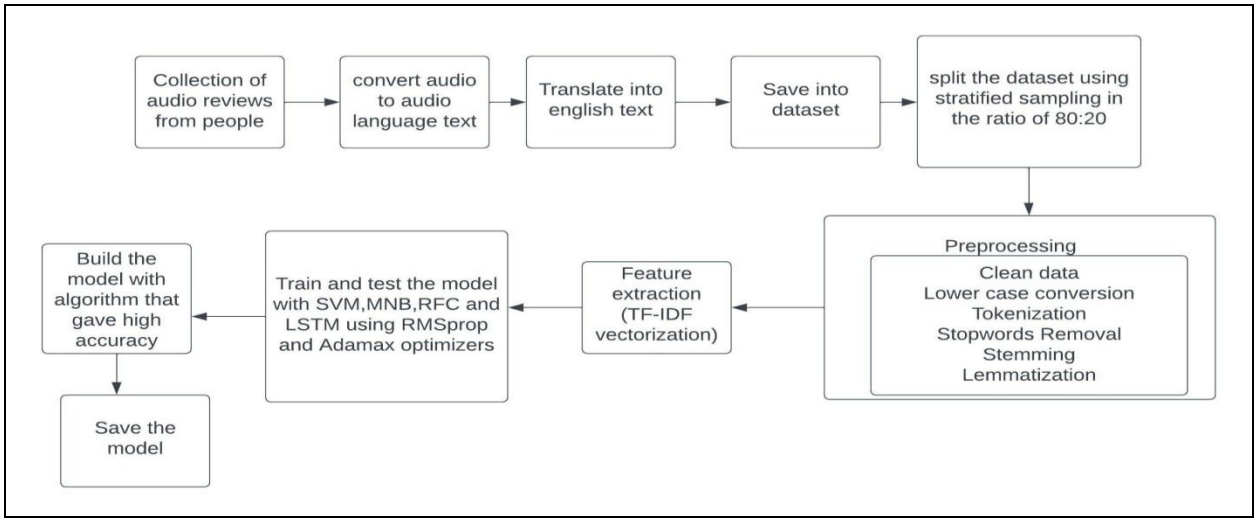

Figure 1: The proposed Method overview

Initially for detecting the language from the collected we used Recognizer() method under the speech Recognition module that can be able to hear and detect the words spoken. The Audio Segment method under pydub allows you to extract the recognized audio but sometimes there may be issues if they are large audio files so split onsilence() method allows to split the audio file into chunks where output from chunks of an audio can be combined as a whole to get the sentence from it. Later, the intermediate sentence generated should be converted into English text which can be carried out using Google Translator method.

3.2. Preprocessing

As the entire sentence does not carry the sentiments, it is necessary to eliminate the unnecessary words which help in finding proper weighted sentiments and also increasing the overall accuracy of the system. Text preprocessing allows you to remove the noisy data and it becomes handy to perform sentiment classification task. The preprocessing task can be carried out using nltk package in python which has methods for carrying out the preprocessing tasks which include:

- Cleaning: As the symbols etc. do not add any sentiment value, they can be removed
- Tokenization: Under this step, the sentence is broke down into a sequence of words.

- Stopwords Removal: As the stopwords doesn't have any effect on sentiment, they will be removed. nltk package already has the set of stopwords in English so, removing such words would be an easy task.
- Stemming: Stemming helps to obtain the root form of the words but as it blindly stems based on the rules, it has some disadvantages which would be solved by lemmatization.
- Lemmatization: It is also similar as stemming but it checks the correctness of final root word obtained in terms of vocabulary i.e., such that meaningful words are generated.

The output of this step will be stored in a CSV file which will be later used for training phase.

3.3. Feature Extraction

For any classifier, Most of the algorithms expect the data to be in numerical format i.e., a certain numerical weight to the words which specifies the need for text to get converted into a vector. To achieve this, we have used TF-IDF vectorizer where

$$IDF_i=\log(n/df_i)$$

where n is total number of documents, df_i is document frequency, IDF_i is inverse domain frequency, i refers to current word .The final TF-IDF score can be obtained by the product of TermFrequency with its corresponding IDF,

TF-IDF score=TF * IDF

The output of this step gives a feature numerical value to the words.

3.4. Training the model

We supply the preprocessed text reviews which are manually labeled as positive or negative for training the model. For training purpose, we have considered several algorithms which include Supervised Machine Learning algorithms like SVM, Multinomial NB, RandomForest and neural network based algorithm like LSTM applied along with RMSprop, Adamax as optimizers. These algorithms could be able to predict the labels of testing data based on the knowledge or features learnt from the testing data.

3.4.1. Support Vector Machine (SVM):

SVM is a supervised machine learning algorithm which is mostly used the classification tasks. It uses a dimensional space for classifying the sentences to their respective classes. It constructs the best line (or) hyper plane that separates the dimensional space into classes based on the features learnt from training data. As the model learns the features of respective classes, whenever a new dataitem occurs, it classifies the dataitem to its respective category.

3.4.2. Multinomial NaiveBayes (MNB):

The Multinomial Naive Bayes algorithm is a Bayesian learning method in Natural Language Processing (NLP) which uses Bayes theorem. This MNB mainly uses the concept of conditional probability based on which it predicts the label of a text i.e., classify to its respective class. For given data it checks the possibility for belongingness to each class and maps the sample data to the most probable class. The Naive Bayes method can be used for text analysis and for classifying instances to their corresponding classes. Multinomial Naive Bayes specifies need for conversion of text to a vector format as it requires the feature

vectors i.e., words being transformed to a local feature value.

3.4.3. Random Forest Classifier (RFC):

Random Forest algorithm is one of the supervised machine learning algorithms which can be used for classification tasks. It constructs several decision trees based on the feature values of data in the subsets generated from the data supplied for training. It combines several classifiers to predict the final output. Whenever a new data item or new instance occurs, the predictions are generated by the individual decision trees and highly predicted class is observed and selected as the final label for the new instance.

3.4.4. Long Short Term Memory (LSTM):

LSTM is an refined version of Recurrent Neural Network (RNN). The LSTM solves the problem of vanishing gradient in RNN where gradient value becomes weaker each time during back propagation in calculating the weights which makes it unreliable for learning which is solved by LSTM. LSTM allow you to feed input sequences to a network, and make predictions. If proper layers are used, the network would be able to predict actual meaning in the context and map to its most precise class. LSTM generally uses various gates each meant for particular task i.e., forget gate meant for deciding on data relevancy and calculating cell state, input gate meant for updating cell state and checking importance of information, cell state meant for keeping all gained information. Based on all this, the LSTM could classify the data to its corresponding class.

3.5 Need for optimization:

There are some arguments which could improvise the performance of any model and one amongst those is optimization algorithms. Generally, there is something called learning rate which could tell about how well the model could learn from the training data. If the learning rate is good, then the performance of the model could be efficient. These optimization algorithms improve such learning rate which makes the system to learn faster. Optimization algorithms help in decreasing the objective function i.e., points in decreasing the difference between the expected and predicted values. The optimizers help you to improve the overall accuracy of the system. RMSprop and Adamax are some of the most frequently used optimizers.

3.5.1. RMS prop optimizer:

The optimizer that we have used here is RMS prop which is to stabilize the optimization i.e., decrease the number of functional computations required to reach the optima, or to improve the overall capability of the optimization algorithm, giving better final results .RMS prop is similar to gradient descent algorithm with momentum that uses adaptive learning rate which conveys that the learning rate changes over time. Gradient descent is used in machine learning to find the function's parameter values that minimize a cost function (decrease the difference in predicted and expected values). The equations below show how the gradients are calculated for the RMSprop where momentum is denoted by beta and generally set to 0.9.

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2$$
$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db^2$$
$$W = W - \alpha \cdot \frac{dw}{\sqrt{v_{dw}} + \epsilon}$$
$$b = b - \alpha \cdot \frac{db}{\sqrt{v_{db}} + \epsilon}$$

3.5.2. Adamax Optimizer:
Adamax is the improved version to Adam version of gradient descent which helps in accelerating overall optimization process. Adam (Adaptive Movement Estimation) uses separate step size for each input which may lead to rapid decrease in value. Adamax itself adapts different learning rate for each argument in optimization process.
All these steps are applied on both the datasets i.e., on both government schemes.

4. Results and Discussion

4.1. Metrics

4.1.1. Accuracy: Accuracy is the ratio of the correct predictions made by the model to the total number of samples. Accuracy is the major parameter considered for evaluating the performance of the models i.e., the higher the accuracy is, the best the model would be.
Accuracy= (No. of correct predictions/ Total no.of samples)* 100 %

$$\text{Accuracy} = \frac{True\ positives + True\ negatives}{True\ positives + True\ negatives + False\ positives + False\ negatives}$$

4.1.2. Precision: Precision is the ratio of correct positive predictions to total number of positive predictions.

$$Precision = \frac{True\ positives}{Truepositives + Falsepositives}$$

4.1.3. Recall: Recall is the ratio of correct positive predictions to the total number of positive samples.

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

where True positive is the correct positive class prediction by the model
True negative is the correct negative class prediction by the model
False positive is the incorrect positive class prediction by the model
False negative is the incorrect negative class prediction by the model

4.1.4. F1- score: F1-score is a measure calculated based on precision and recall.

$$F1\ score = 2 * (Precision * Recall) / (Precision + Recall)$$

4.2. Results Evaluation: The study consider two data sets Ration and JVD which were created by our own. To design data sets considered our college (i.e., Lakireddy bali reddy college of engineering) students as test data samples.Total 10 students were participated in

this study. After reading the datasets applied translation and preprocessing techniques. By using the test size parameter the entire dataset will be split into testing and training data. We have taken the test size as 0.2 i.e., splits up into 80% training data and 20% testing data. We have applied or used stratified sampling while running the system against various test samples.In the case of the stratified sampling, Firstly, the selection of testing set randomly from the entire training set is one of the very important task that should be considered while building a model i.e., the testing data should be selected in such a way that it represents the whole training data. If not, then as the testing data doesn't cover the entire population, the model performance can't be estimated correctly. Stratified sampling is refined version of random sampling as it selects the testing data in such a way that all the population gets represented and checks such that all cases in the entire samples are into consideration.

Table 1: Average values of metrics given by algorithms when tested on 10 samples

Algorithm	**Accuracy**	**Precision**		**Recall**		**F1-score**	
		Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
SVM	80.02	0.74	0.86	0.87	0.75	0.80	0.80
MNB	77.08	0.79	0.75	0.79	0.78	0.78	0.75
RFC	76.49	0.82	0.70	0.77	0.78	0.79	0.72
LSTM with RMSprop	72.36	0.89	0.54	0.70	0.82	0.78	0.63
LSTM with Adamax	68.24	0.93	0.40	0.64	0.88	0.76	0.54

We have considered 10 test cases for each dataset with stratified sampling for evaluating the performances of model trained with different algorithms. The metrics values i.e., Accuracy, Precision, Recall, F1-score of all the testcases are generated for each algorithm. The average values of these metrics are computed for each algorithm and they are compared. We have chosen the algorithm that gave the best results in both datasets for developing the sentiment prediction model.The complete resuts over Ration data sets related to various classifiers is described in Table 1 over few samples. The accuracy of each algoirhtm in individual 10 cases is shown in Table 2. Similary The Precision, Recall,F1-Score of each algoirhtm in individual 10 cases is shown in Table 3-5.

In Table-1, For the considered 10 test cases, the average metric values for different algorithms are calculated based on the individual metric values produced for each test case.The SVM accuracy is produced promising compared to others.

Table 2: Accuracy values of algorithms on each testcase

Algorithm	Accuracy									
	Test 1	**Test 2**	**Test 3**	**Test4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**	**Test 9**	**Test 10**
SVM	88.2	70.6	82.4	88.2	76.5	76.5	82.4	82.4	70.6	82.4
MNB	76.5	82.4	76.5	88.2	70.6	76.5	82.4	64.7	76.5	76.5
RFC	76.5	70.6	58.8	82.4	88.2	70.6	76.5	82.4	82.4	76.5
LSTM with RMSprop	76.5	88.2	64.7	58.8	76.5	70.6	82.4	70.6	76.5	58.8
LSTM with Adamax	64.7	70.6	64.7	70.6	58.8	70.6	70.6	64.7	64.7	82.4

In Table-2, we can observe the individual accuracy values given by the algorithms when applied on each of the generated stratified test samples. From the Overall ten cases two methods of LSTM produced promising results compared to the other methods especially in the Test 7-10.

Table 3: Precision values of algorithms on each testcase

	Precision									
	SVM		**MNB**		**RFC**		**LSTM with RMS prop**		**LSTM with Adamax**	
Test Samples	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**
Test 1	0.78	1.00	0.67	0.88	0.78	0.75	0.89	0.63	1.00	0.25
Test 2	0.67	0.75	0.67	1.00	0.89	0.50	0.78	1.00	0.89	0.50
Test 3	0.78	0.88	0.78	0.75	0.89	0.25	0.78	0.50	1.00	0.25
Test 4	0.89	0.88	0.89	0.88	0.78	0.88	0.89	0.25	1.00	0.38
Test 5	0.78	0.75	0.67	0.75	0.89	0.88	0.89	0.63	0.78	0.38
Test 6	0.78	0.75	1.00	0.50	0.67	0.75	1.00	0.38	1.00	0.38
Test 7	0.78	0.88	0.89	0.75	0.89	0.63	1.00	0.63	1.00	0.38
Test 8	0.67	1.00	0.67	0.63	0.78	0.88	0.89	0.50	0.89	0.38
Test 9	0.56	0.88	0.78	0.75	0.89	0.75	1.00	0.50	0.78	0.50
Test 10	0.78	0.88	0.89	0.63	0.78	0.75	0.78	0.38	1.00	0.63

In Table-3, we can observe the individual precision values of each class label given by the algorithms when applied on each of the generated stratified test samples.

Table 4: Recall values of algorithms on each testcase

Test Samples	Recall									
	SVM		MNB		RFC		LSTM with RMSprop		LSTM with Adamax	
	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
Test 1	1.00	0.80	0.86	0.70	0.78	0.75	0.73	0.83	0.60	1.00
Test 2	0.75	0.67	1.00	0.73	0.67	0.80	1.00	0.80	0.67	0.80
Test 3	0.88	0.78	0.78	0.75	0.57	0.67	0.64	0.67	0.60	1.00
Test 4	0.89	0.88	0.89	0.88	0.88	0.78	0.57	0.67	0.64	1.00
Test 5	0.78	0.75	0.75	0.67	0.89	0.88	0.73	0.83	0.58	0.60
Test 6	0.78	0.75	0.69	1.00	0.75	0.67	0.64	1.00	0.64	1.00
Test 7	0.88	0.78	0.80	0.86	0.73	0.83	0.75	1.00	0.64	1.00
Test 8	1.00	0.73	0.67	0.63	0.88	0.78	0.67	0.80	0.62	0.75
Test 9	0.83	0.64	0.78	0.75	0.80	0.86	0.69	1.00	0.64	0.67
Test 10	0.88	0.78	0.73	0.83	0.78	0.75	0.58	0.60	0.75	1.00

In Table-4, we can observe the individual recall values of each class label given by the algorithms when applied on each of the generated stratified test samples.

Table 5: F1-score values of algorithms on each testcase

Test Samples	F1-Score									
	SVM		MNB		RFC		LSTM with RMSprop		LSTM with Adamax	
	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
Test 1	0.88	0.89	0.75	0.78	0.78	0.75	0.80	0.71	0.75	0.40
Test 2	0.71	0.71	0.80	0.84	0.76	0.62	0.88	0.89	0.76	0.62
Test 3	0.82	0.82	0.78	0.75	0.70	0.36	0.70	0.57	0.75	0.40
Test 4	0.89	0.88	0.89	0.88	0.82	0.82	0.70	0.36	0.78	0.55
Test 5	0.78	0.75	0.71	0.71	0.89	0.88	0.80	0.71	0.67	0.46
Test 6	0.78	0.75	0.82	0.67	0.71	0.71	0.78	0.55	0.78	0.55
Test 7	0.82	0.82	0.84	0.80	0.80	0.71	0.86	0.77	0.78	0.55
Test 8	0.80	0.84	0.67	0.63	0.82	0.82	0.76	0.62	0.73	0.50
Test 9	0.67	0.74	0.78	0.75	0.84	0.80	0.82	0.67	0.70	0.57
Test 10	0.82	0.82	0.80	0.71	0.78	0.75	0.67	0.46	0.86	0.77

In Table-5, we can observe the individual F1-score values of each class label given by the algorithms when applied on each of the generated stratified test samples.
From Table 1-5, we can observe the metric values given by each algorithm considered when applied on the 10 test cases (testing sets) generated from stratified sampling on ration dataset and SVM gave the best accuracy.

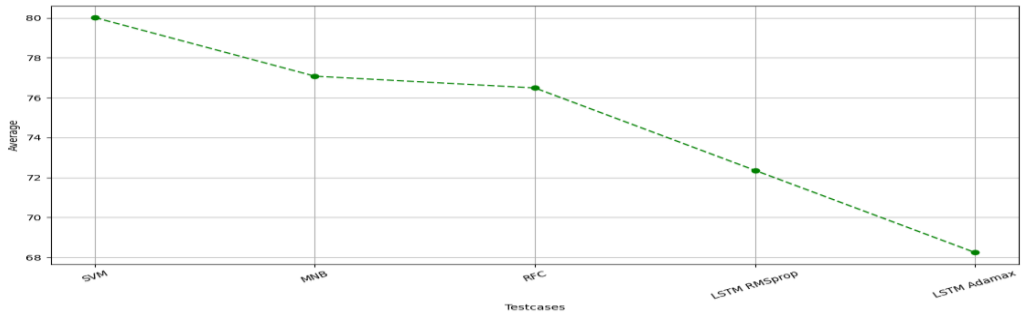

Figure 2: Average accuracies of algorithms on RATION dataset.

Figure-2 represents the average accuracy values by each of the individual algorithms when applied on the testsets generated from ration dataset and we can see that SVM gave the best accuracy.

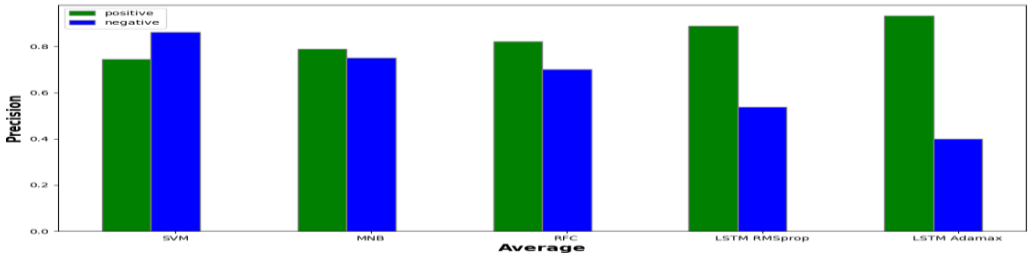

Figure 3: Average precision values of algorithms on RATION DATASET

Figure-3 represents the average precision values of each class label by each of the individual algorithms when applied on the testsets generated from ration dataset.

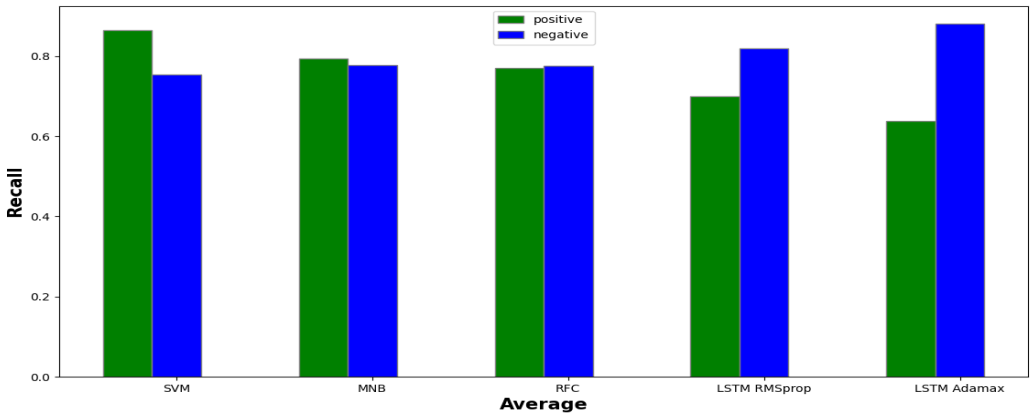

Figure 4: Average recall values of algorithms on RATION Dataset

Figure-4 represents the average recall values of each class label by each of the individual algorithms when applied on the testsets generated from ration dataset.

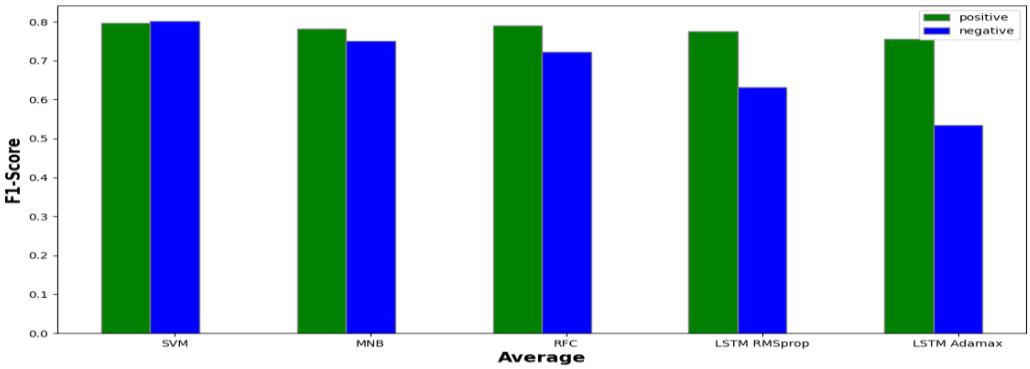

Figure 5: Graph representing the average F1-score values of algorithms on RATION Dataset

Figure-5 represents the average f1-score values of each class label by each of the individual algorithms when applied on the testsets generated from ration dataset.
Similarly the same analysis done over the other data sets known as JVD. The complete resuts over this data sets related to various classifiers is described in Table 6 over few samples. The accuracy of each algoirhtm in individual 10 cases is shown in Table 2. Similary The Precision, Recall,F1-Score of each algoirhtm in individual 10 cases is shown in Table 7-1.

Table 6: Average values of metrics given by algorithms when tested on 10 samples

Algorithm	Accuracy	Precision		Recall		F1-score	
		Positive	Negative	Positive	Negative	Positive	Negative
SVM	86.67	0.78	0.93	0.90	0.86	0.83	0.89
MNB	81.67	0.64	0.94	0.92	0.79	0.73	0.86
RFC	81.67	0.70	0.90	0.84	0.81	0.75	0.85
LSTM with RMSprop	74.17	0.92	0.61	0.65	0.93	0.75	0.72
LSTM with Adamax	60.84	0.80	0.47	0.52	0.80	0.63	0.57

In Table-6, For the considered 10 test cases, the average metric values for different algorithms are calculated based on the individual metric values produced for each test case.

Table 7: Accuracy values of algorithms on each testcase

Algorithm	Accuracy									
	Test 1	Test 2	Test 3	Test4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10
SVM	66.70	83.30	83.30	91.70	100.00	91.70	83.30	83.30	91.70	91.70
MNB	83.30	66.70	91.70	83.30	91.70	75.00	75.00	75.00	83.30	91.70

RFC	83.30	83.30	91.70	83.30	66.70	75.00	66.70	83.30	91.70	91.70
LSTM with RMSprop	83.30	66.70	58.30	66.70	75.00	75.00	75.00	83.30	91.70	66.70
LSTM with Adamax	50.00	66.70	50.00	58.30	75.00	66.70	58.30	41.70	66.70	75.00

In Table-7, we can observe the individual accuracy values given by the algorithms when applied on each of the generated stratified test samples.

Table 8: Precision values of algorithms on each testcase

	Precision									
	SVM		**MNB**		**RFC**		**LSTM with RMSprop**		**LSTM with Adamax**	
Test Samples	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**
Test 1	0.60	0.71	0.60	1.00	0.80	0.86	1.00	0.71	0.80	0.29
Test 2	0.60	1.00	0.60	0.71	0.80	0.86	1.00	0.43	0.80	0.57
Test 3	0.80	0.86	0.80	1.00	0.80	1.00	1.00	0.29	1.00	0.14
Test 4	1.00	0.86	0.60	1.00	0.60	1.00	0.80	0.57	0.60	0.57
Test 5	1.00	1.00	1.00	0.86	0.40	0.86	1.00	0.57	1.00	0.57
Test 6	0.80	1.00	0.40	1.00	0.80	0.71	1.00	0.57	0.80	0.57
Test 7	0.80	0.86	0.60	0.86	0.40	0.86	0.80	0.71	0.80	0.43
Test 8	0.60	1.00	0.40	1.00	0.80	0.86	1.00	0.71	0.40	0.43
Test 9	0.80	1.00	0.60	1.00	0.80	1.00	0.80	1.00	0.80	0.57
Test 10	0.80	1.00	0.80	1.00	0.80	1.00	0.80	0.57	1.00	0.57

In Table-8, we can observe the individual precision values of each class label given by the algorithms when applied on each of the generated stratified test samples.

Table 9: Recall values of algorithms on each testcase

	Recall									
	SVM		**MNB**		**RFC**		**LSTM with RMSprop**		**LSTM with Adamax**	
Test Samples	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**
Test 1	0.60	0.71	1.00	0.78	0.80	0.86	0.71	1.00	0.44	0.67
Test 2	1.00	0.78	0.60	0.71	0.80	0.86	0.56	1.00	0.57	0.80
Test 3	0.80	0.86	1.00	0.88	1.00	0.88	0.50	1.00	0.45	1.00

Test 4	0.83	1.00	1.00	0.78	1.00	0.78	0.57	0.80	0.50	0.67
Test 5	1.00	1.00	0.83	1.00	0.67	0.67	0.63	1.00	0.63	1.00
Test 6	1.00	0.88	1.00	0.70	0.67	0.83	0.63	1.00	0.57	0.80
Test 7	0.80	0.86	0.75	0.75	0.67	0.67	0.67	0.83	0.50	0.75
Test 8	1.00	0.78	1.00	0.70	0.80	0.86	0.71	1.00	0.33	0.50
Test 9	1.00	0.88	1.00	0.78	1.00	0.88	1.00	0.88	0.57	0.80
Test 10	1.00	0.88	1.00	0.88	1.00	0.88	0.57	0.80	0.63	1.00

In Table-9, we can observe the individual recall values of each class label given by the algorithms when applied on each of the generated stratified test samples.

Table 10: F1-score values of algorithms on each testcase

	F1-score									
	SVM		**MNB**		**RFC**		**LSTM with RMSprop**		**LSTM with Adamax**	
Test Samples	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**	**positive**	**negative**
Test 1	0.60	0.71	0.75	0.88	0.80	0.86	0.83	0.83	0.57	0.40
Test 2	0.75	0.88	0.60	0.71	0.80	0.86	0.71	0.60	0.67	0.67
Test 3	0.80	0.86	0.89	0.93	0.89	0.93	0.67	0.44	0.63	0.25
Test 4	0.91	0.92	0.75	0.88	0.75	0.88	0.67	0.67	0.55	0.62
Test 5	1.00	1.00	0.91	0.92	0.50	0.75	0.77	0.73	0.77	0.73
Test 6	0.89	0.93	0.57	0.82	0.73	0.77	0.77	0.73	0.67	0.67
Test 7	0.80	0.86	0.67	0.80	0.50	0.75	0.73	0.77	0.62	0.55
Test 8	0.75	0.88	0.57	0.82	0.80	0.86	0.83	0.83	0.36	0.46
Test 9	0.89	0.93	0.75	0.88	0.89	0.93	0.89	0.93	0.67	0.67
Test 10	0.89	0.93	0.89	0.93	0.89	0.93	0.67	0.67	0.77	0.73

In Table-10, we can observe the individual f1-score values of each class label given by the algorithms when applied on each of the generated stratified test samples.

From Table 6-10, we can observe the metric values given by each algorithm considered when applied on the 10 test cases (testing sets) generated from stratified sampling on JVD dataset and SVM gave the best accuracy.

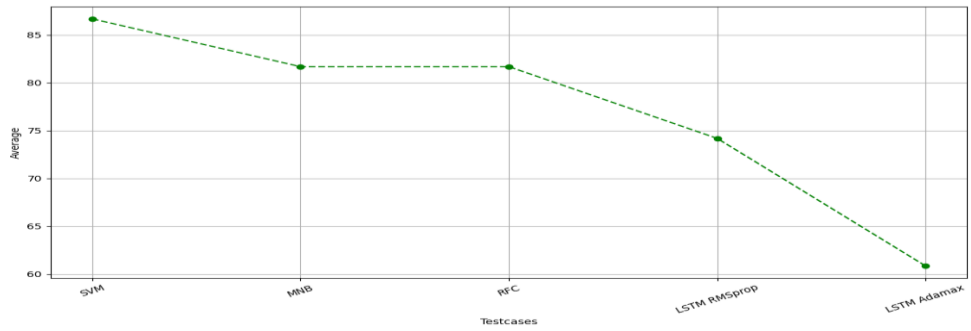

Figure 6: Average accuracies of algorithms on JVD Dataset

Figure-6 represents the average accuracy values by each of the individual algorithms when applied on the testsets generated from jvd dataset and we can see that SVM gave the best accuracy.

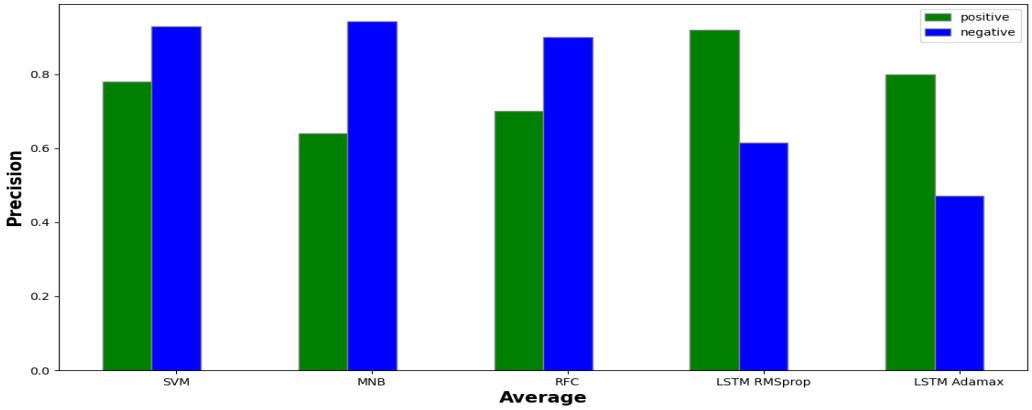

Figure 7: Graph representing the average precision values of algorithms on JVD Dataset

Figure-7 represents the average precision values of each class label by each of the individual algorithms when applied on the test sets generated from JVD dataset.

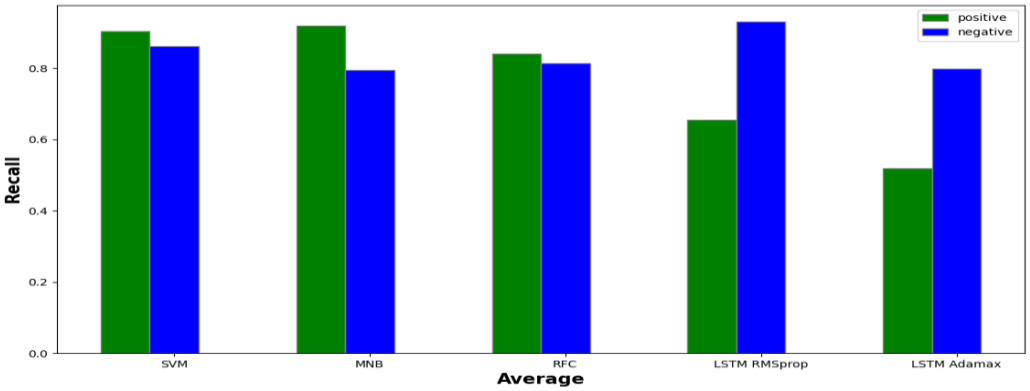

Figure 8: Average recall values of algorithms on JVD Dataset

Figure-8 represents the average recall values of each class label by each of the individual algorithms when applied on the test sets generated from jvd dataset.

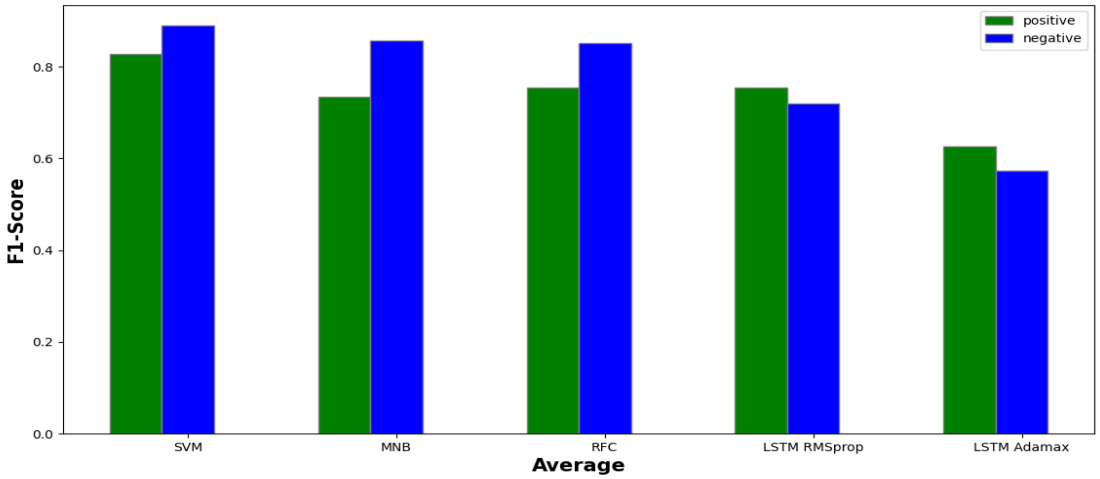

Figure 9: Average f1-score values of algorithms on JVD Dataset

Figure-9 represents the average f1-score values of each class label by each of the individual algorithms when applied on the test sets generated from JVD dataset.

From the table values and the figures generated, among the considered algorithms as SVM gave the best results in case of both the datasets, the sentiment prediction model is developed using the SVM algorithm for best results.

5. Conclusion and Future Scope

On an average, as SVM gave the best results when applied on both datasets, the sentiment prediction model is developed using the SVM algorithm. The model would be trained using the SVM algorithm and would be fed with an audio as input, which undergo all the steps like audio to English text conversion, preprocessing, feature extraction and the proposed model classifies the input under positive (or) negative i.e., predict the sentiment polarity of the input. As having many audios provide us with more number of features and insights for the model, we plan on collecting large dataset in the future to extend the performance of the sentiment prediction model as it gets supplied with large training data. In future we plan to develop the model which can give way better accuracy even when applied on cross-domain datasets.

REFERENCES

[1] https://www.indiaonlinepages.com/population/literacy-rate-in-india.html
[2] W. Zhao *et al.*, "Weakly-Supervised Deep Embedding for Product Review Sentiment Analysis," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 185-197, 1 Jan. 2018, doi: 10.1109/TKDE.2017.2756658.
[3] G. Xu, Z. Yu, H. Yao, F. Li, Y. Meng and X. Wu, "Chinese Text Sentiment Analysis Based on Extended Sentiment Dictionary," in *IEEE Access*, vol. 7, pp. 43749-43762, 2019, doi: 10.1109/ACCESS.2019.2907772.
[4] Osman, Nurul. (2020). Contextual Sentiment Based Recommender System to Provide Recommendation in the Electronic Products Domain. International Journal of Machine Learning and Computing. 9. 10.18178/ijmlc.2019.9.4.821.

[5] Fang, X., Zhan, J. Sentiment analysis using product review data. *Journal of Big Data* **2,** 5 (2015). https://doi.org/10.1186/s40537-015-0015-2.
[6] L. Huang, Z. Dou, Y. Hu and R. Huang, "Textual Analysis for Online Reviews: A Polymerization Topic Sentiment Model," in *IEEE Access*, vol. 7, pp. 91940-91945, 2019, doi: 10.1109/ACCESS.2019.2920091.
[7] El Alaoui, I., Gahi, Y., Messoussi, R. *et al*. A novel adaptable approach for sentiment analysis on big social data. *J Big Data* **5,** 12 (2018). https://doi.org/10.1186/s40537-018-0120-0
[8] Mahalakshmi, S. & Elango, Sivasankar. (2015). Cross Domain Sentiment Analysis Using Different Machine Learning Techniques. 10.1007/978-3-319-27212-2_7.
[9] N. Jin, J. Wu, X. Ma, K. Yan and Y. Mo, "Multi-Task Learning Model Based on Multi-Scale CNN and LSTM for Sentiment Classification," in *IEEE Access*, vol. 8, pp. 77060-77072, 2020, doi: 10.1109/ACCESS.2020.2989428.
[10] J. Park, "Framework for Sentiment-Driven Evaluation of Customer Satisfaction With Cosmetics Brands," in *IEEE Access*, vol. 8, pp. 98526-98538, 2020, doi: 10.1109/ACCESS.2020.2997522.
[11] R. Obiedat et al., "Sentiment Analysis of Customers' Reviews Using a Hybrid Evolutionary SVM-Based Approach in an Imbalanced Data Distribution," in IEEE Access, vol. 10, pp. 22260-22273, 2022, doi: 10.1109/ACCESS.2022.3149482.
[12] D. Bollegala, T. Mu and J. Y. Goulermas, "Cross-Domain Sentiment Classification Using Sentiment Sensitive Embeddings," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 398-410, 1 Feb. 2016, doi: 10.1109/TKDE.2015.2475761.
[13] D. Bollegala, D. Weir and J. Carroll, "Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 8, pp. 1719-1731, Aug. 2013, doi: 10.1109/TKDE.2012.103.
[14] A. Abdul Aziz and A. Starkey, "Predicting Supervise Machine Learning Performances for Sentiment Analysis Using Contextual-Based Approaches," in IEEE Access, vol. 8, pp. 17722-17733, 2020, doi: 10.1109/ACCESS.2019.2958702.
[15] K. Lavanya, L. S. S. Reddy and B. Eswara Reddy, "Modelling of Missing Data Imputation using Additive LASSO Regression Model in Microsoft Azure", Journal of Engineering and Applied Sciences,2018,Vol 13,Special Issue 8,pp:6324-6334.
[16] K. Lavanya, G.V.Suresh," An Additive Sparse Logistic Regularization Method for Cancer Classification in Microarray Data", The International Arab Journal of Information Technology, Vol. 18, No. 2, March 2021.https://doi.org/10.34028/iajit/18/10, ISSN: 1683-3198E-ISSN: 2309-4524.
[17] K.Lavanya, K. Harika, D. Monica, K. Sreshta,"Additive Tuning Lasso (AT-Lasso): A Proposed Smoothing Regularization technique for Shopping Sale Price Prediction", International Journal of Advanced Science Technology, Vol 29,No 05,pp 878-886,2020.
[18] Lavanya K., Reddy, L., & Reddy, B. E. (2019). Distributed Based Serial Regression Multiple Imputation for High Dimensional Multivariate Data in Multicore Environment of Cloud. International Journal of Ambient Computing and Intelligence (IJACI), 10(2), 63-79. doi:10.4018/IJACI.2019040105.