Hybrid Deep Learning Approach for Sentiment Classification of Malayalam Tweets

Soumya S
Department of Computer Applications
Cochin University of Science and Technology
Cochin, Kerala, India

Pramod K V
Department of Computer Applications
Cochin University of Science and Technology
Cochin, Kerala, India

***Abstract*—Social media content in regional languages is expanding from day to day. People use different social media platforms to express their suggestions and thoughts in their native languages. Sentiment Analysis (SA) is the known procedure for identifying the hidden sentiment present in the sentences for categorizing it as positive, negative, or neutral. The SA of Indian languages is challenging due to the unavailability of benchmark datasets and lexical resources. The analysis has been done using lexicon, Machine Learning (ML), and Deep Learning (DL) techniques. In this work, the baseline models and hybrid models of Deep Neural Network (DNN) architecture have been used for the classification of Malayalam tweets as positive, negative and neutral. Since, sentiment-tagged dataset for Malayalam is not readily available, the analysis has been done on the manually created dataset and translated Kaggle dataset. The hybrid models used in this study combine Convolutional Neural Networks (CNN) with variants of Recurrent Neural Networks (RNN). The RNN models are Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM) and Gated Recurrent Unit (GRU). All these hybrid models improve the performance of Sentiment Classification (SC) compared to baseline models LSTM, Bi-LSTM and GRU.**

Keywords*—*Bi-LSTM; CNN; NLP; Malayalam; Twitter

I. INTRODUCTION

Sentiment Analysis (SA) is one of the inevitable research domains in Natural Language Processing (NLP), ML and Linguistics. The public express their opinion about products, events, movies, political concerns, ideas, interests, and so on using various social media platforms like Facebook, Twitter, Blogs, and so forth. SA has a significant role in the automatic identification of sentiment hidden in the text [1]. SA classifies the text as positive, negative, or neutral based on the sentiment. SA has been done at different levels like sentence, document, and aspect levels. Since the Tweets are short messages with 280 characters long, sentence-level analysis is most suitable for SA of Tweets.

In the southern state of India, Malayalam is a prominent language spoken by Keralites and also in union territories like Lakshadweep and Puducherry. Most people in Kerala prefer Malayalam to express their opinion, ideas, comments, etc. The majority of youth in Kerala are using Twitter for expressing opinions. Thus the number of tweets is tremendously increasing over time. The Government has initiated the automation of regional languages such as automatic speech recognition, language translation, and character recognition to support the public. The policies made by Government can be modified or changed based on the opinion expressed by the people through social media platforms. Hence, the automation of SA of Malayalam is essential. SA of Indian languages has initiated in the year 2010. The first work in Indian languages reported for the Hindi language [3]. Research work on SA of Malayalam is at the beginning stage due to the lack of benchmark datasets and lexical resources. So far, only few works have been done on Malayalam based on ML, DL, and lexicon-based approaches [29]-[36]. Further important applications are in the field of business analytics, movie reviews, stock market prediction, and so forth.

Hinton et al. proposed different deep learning architectures [2]. The DL models are extensively applied in image processing, NLP, SA, and so on. The most generally used DL models are CNN, RNN, LSTM, and GRU.

The main objective of this study is to enhance the performance of the system and reduce computational costs. In order to achieve this, we extracted the best features by finetuning the hyperparameters and modeled the combined architecture of CNN and the varients of RNN. As a result, the sentiment classification accuracy is improved. In this paper, two different datasets have been analyzed using baseline models and hybrid models of DL architectures. The analysis shows that hybrid models performed better in SA of Malayalam tweets than baseline models.

Word embedding has a significant impact on text processing. Word embedding is the process of converting words or sentences into numerical vectors. These word vectors lie in different dimensions like 32, 64, 128, 300, etc. The words in the same context appear to lie near the same vector space. The word embedding is created either during the training process of embedding layers in the neural network or by using a pre-trained model. The available word embedding models are word2vec [4], BERT [5], fastText [6], etc. The pre-trained model available for the Malayalam language is fastText and BERT. Since the pre-trained model has not performed well in our dataset, the word vectors are formed during the training phase of the embedding layer. The contributions of this study are mentioned below:

- Two different datasets have been created for SA of Malayalam Tweets. Dataset I contains 6304 tweets, where 2907 are positive and 3397 are negative. The Dataset II includes 170000 tweets, among which 66357 are positive, 52798 are negative, and 50845 are neutral.

- Different DL methods like LSTM, Bi-LSTM and GRU have been applied for SA of tweets.
- Novel hybrid DL architectures have been developed by combining both CNN and varients of RNN models for the effective implementation of sentiment classification of Malayalam Tweets.

In the subsequent sections, Section II briefs the related works using hybrid deep learning models and SA of Malayalam language, whereas Section III is the proposed methodology of SA. Section IV describes the novel hybrid models. Section V is the experimental setup followed by results and discussion. Section VI concludes the work.

II. Related Work

The hybrid architecture combines different ML and DL algorithms for feature extraction and sentiment classification of datasets. The various works done using hybrid models are discussed as follows.

DL models are widely used in the analysis of social media content [7][8][9][10][11]. Hassan et al. [12] and Hedge et al. [13] proved that CNN and LSTM performed well in the analysis of short text messages . The recent study shows that the DL models such as CNN and RNN performed well in SA [7][8][14][15][16]. CNN combined with RNN shows potential improvement in the accuracy of SA of English [17]. Srinidhi et al. proposed the LSTM model combined with SVM for SC of IMDb dataset [18]. A similar architecture CNN combined with SVM was proposed by Akhtar et al. for SC of Hindi dataset [19]. SA on reviews/comments from e-commerce sites was proposed by Vo et al. using the hybrid architecture LSTM-CNN model [20]. The same model was proposed by Rehman et al. for SA of movie reviews [21]. All the above-mentioned works have been used single architecture for SC. Multiple DL models, CNN, LSTM and hybrid model CNN-LSTM was proposed by Kastrati et al. [22] . Facebook comments related to the COVID-19 pandemic were tested using this model. Pretrained models like fastText and BERT were utilized for word embedding [23]. The same hybrid model was used for SC of IMDb dataset, social media content and SMS spam detection for Arabic and English Messages [24][25]. The CNN-LSTM with fastText word embedding was proposed by Ombabi et al. [26] . SVM was used at the final layer for SC. A hybrid model CNN and Bi-LSTM were proposed for sentiment and emotions analysis of Chinese product reviews [27]. Pandey et al. proposed a hybrid DL model merged with CNN and Bi-LSTM for SA of Tweets [10]. Dang et al. suggested multiple hybrid DL models with CNN, LSTM, and SVM for the classification of tweets and review datasets [14]. Salur et al. proposed hybrid DL architecture for SC of Turkish dataset [28]. Both CNN and LSTM models were combined for feature extraction and have obtained better accuracy compared to baseline models. The works are done in Malayalam language using rule-based, lexicon-based, Fuzzy logic, ML, and DL methods are shown in Table I. All the works have been done on the dataset which is created by corresponding authors on different domains like a movie review, novel, Tweets, etc. Preprocessing is an important step in SA, which depends on the domain of the dataset. All the works in Malayalam have been used as a single model for SC. Here, we have used hybrid architecture which extracts better features and has shown potential improvement in the SA tasks. The proposed hybrid DL models and their evaluation are explained in consecutive sections.

III. Proposed Methodology

In the previous works, lexicon-based, ML approaches like NB, SVM and RF, DL models like RNN, CNN, LSTM, BiLSTM and GRU have been applied for SA of Malayalam tweets [37][38][39]. Considering the significance of hybrid models in SA, three different hybrid architectures are developed in this work. As the first step of implementation, two different sentiment-tagged Malayalam datasets has been created. The pre-processing steps eliminate unnecessary information from the retrieved text. After that the feature vector is formed during the training phase of embedding layer. The feature vector is given to baseline as well as hybrid DL models. The sigmoid activation function for Dataset I and softmax for Dataset II are applied at the output layer in the baseline models and SVM is applied in the hybrid models. The proposed methodology for SC is shown in Fig. 1.

A. Dataset

Dataset I is created by retrieving Malayalam tweets using sentiment-oriented words [38]. Dataset II is created by translating the Kaggle dataset existing in English to Malayalam by using the Google document translator. The sample dataset is shown in Fig. 2. The sentiment distribution of the Dataset I and Dataset II are shown in Fig. 3 and Fig. 4, respectively. In Dataset I, 0 and 1 represents negative and positive sentiments, whereas, in the Dataset II, 0, 1, and 2 represent neutral, positive, and negative sentiment. The length of sentences under each category of Dataset I and Dataset II are shown in Fig. 5 and Fig. 6, respectively.

B. Preprocessing

The performance of the system has been improved by removing unnecessary information from the text. The following steps have been performed as part of data cleaning.

- Hyperlinks: Most of the tweets contain brief sentences followed by hyperlinks that do not provide meaningful information for SA. Hence all the hyperlinks have been removed from the corpus.
- Punctuations and Special Characters: Removed punctuations like :, , ',... etc. and special characters like @, $, #, etc. from the text.
- Stop Words: Stop words are often occurring in the input sentence, but are less information-oriented. Hence they are removed to reduce the vocabulary size.

Data cleaning has been done using regular expressions in Python language. Negative sentences are labeled with 0 and positive are labeled with 1. Preprocessing is an essential step in reducing the vocabulary size and removing unwanted information. Hyperlinks, special characters, punctuations, digits, foreign languages, etc. are removed by using the regular expression before extracting the features. Vocabulary is created by tokenizing the sentence based on Unicode standards within the range of 0D00–0D7F, which is shown in Fig. 7. Tokenized

TABLE I. SA OF MALAYALAM

Author	Domain with Size of Dataset	Preprocessing	Features	Classification Method	Accuracy
Mohandas et al. (2012) [29]	Novel	POS tagging	Tokens	SO-PMI-IR	63 %
Nair et al. (2014) [30]	Movie Review	Sandhi Splitter	Tokens, Negation	Rule Based	85 %
Anagha M et al.(2014) [31]	Movie Review	POS tagging	Wordnet	Lexicon	93.6 %
Anagha M et al. (2015) [32]	Movie Review	POS tagging	TnT tagger	Fuzzy Logic	91.6 %
Nair et al. (2015) [33]	Movie Review (30,000 tokens)	POS tagging	Tokens, Negation Intensifier	CRF, SVM	SVM: 91 %
Kumar et al. 2017 [34]	Tweets (12822)	Removing hyperlinks and punctuations	word embedding with different dimensions	LSTM, CNN	98.24 %
Rahul et al. (2018) [35]	Social Media	Removing hyperlinks and punctuations	POS tagging Positive and Negative Intensifier Negation	CRF, SVM	SVM: 52.75 %
Kumar et al. (2019) [36]	Tweets	Removing hyperlinks and punctuations	word embedding with different dimensions	RKS-RBF, LSTM CNN	86.5 % 89.3 %

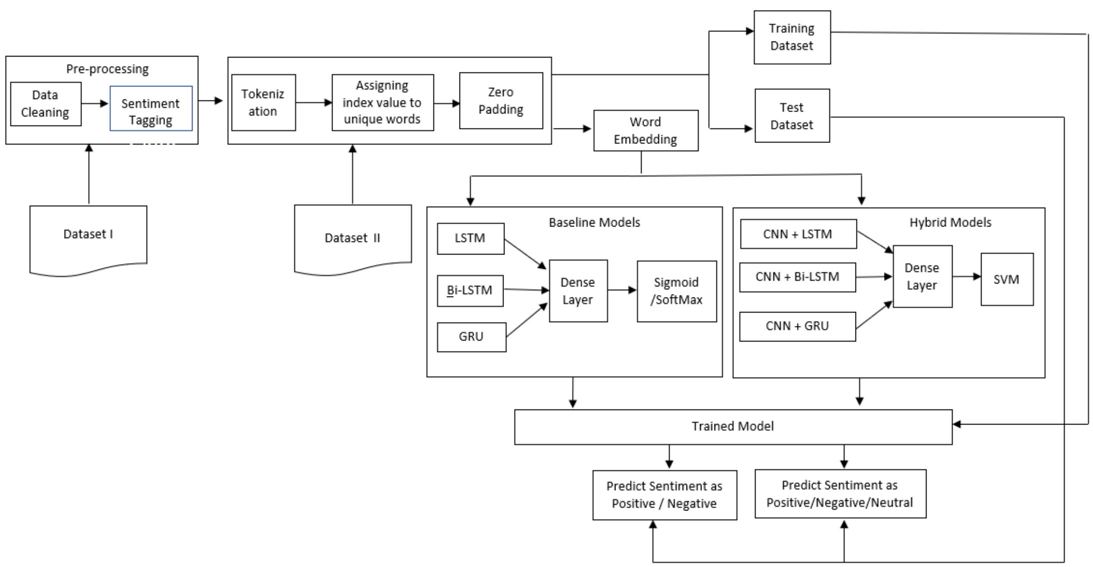

Fig. 1. Methodology for the Proposed System.

TABLE II. DATASET 1

Dataset I	
Sentiment	Number of Tweets
Positive	2907
Negative	3397
Total	6304

TABLE III. DATASET 2

Dataset II	
Sentiment	Number of Tweets
Positive	66357
Negative	52798
Neutral	50845
Total	170000

corpus is converted to a sequence of integers with variable length records and is represented in Fig. 8. The variable-length record is converted to fixed-length by zero-padding which is shown in Fig. 9. Word vector is formed during the training phase of embedding layer which takes embedding dimension, vocabulary size and maximum length of sentence as parameters. Embedding dimension takes different values like 64, 128 and 300, but it shows better performance when the dimension is set to 300. The 300 dimensional word vector is shown in Fig. 10.

IV. HYBRID ARCHITECTURE

In the related study, different hybrid models combined with CNN and LSTM show better results in SA of English languages. Here, we have used three different hybrid architectures for SA of Malayalam tweets. The word vector is created during the training phase of the embedding layer and then given to CNN and various RNN models. The features extracted by the CNN and RNN models are merged and corresponding output

Dataset I		
1	കലഹം നാലാംക്ലാസുകാരൻ മൂന്നാംക്ലാസുകാരനെ മണ്ണെണ്ണയൊഴിച്ച് തീകൊളുത്തി. (Riot A fourth-grader burned to death with kerosene)	0
2	അണലിയുടെ കടിയേറ്റ് ചികിത്സയിൽ കഴിയുന്ന വാവ സുരേഷിന്റെ ആരോഗ്യനില അതീവ ഗുരുതരം. (Vava Suresh is in serious condition after being bitten by a viper)	0
3	എഴുന്നേറ്റാലുടൻ ഒരുഗ്ലാസ് വെള്ളവും ഏറ്റവുമിഷ്ടമുള്ളൊരു പാട്ടും കേട്ടാൽ ശാരീരികവും മാനസികവുമായി ഉന്മേഷം ലഭിക്കും. (Getting up and drink a glass of water and listening to a favorite song will refresh you physically and mentally)	1
4	മൺസൂൺ നന്നായി പെയ്യുന്നു എന്നുള്ളത് വളരെ ശുഭകരമായ ഒരു വാർത്തയാണ് . (The good news is that the monsoon is getting better)	1
Dataset II		
1	ദൈവം നാളെ ഇന്ത്യയെ അനുഗ്രഹിക്കട്ടെ മോദി പ്രധാനമന്ത്രിയാകും (God bless India tomorrow Modi will be the Prime Minister)	0
2	മോഡിസ് ഇപ്പോൾ ഉപയോഗിക്കുന്ന അതേ വാക്കുകൾ ഇതിൽ നിന്ന് മാറ്റുന്നു (It replaces the same words that Modi now uses)	0
3	ആവേശമുണർത്തുന്ന ട്രെയിലർ ഗംഭീരമായ ശ്രമം കാത്തിരിക്കുന്നു നന്നായി കാണൂ കഠിനമായ ചോദ്യങ്ങൾ ചോദിക്കുന്നു ജനാധിപത്യം ശരിക്കും സംരക്ഷിച്ചു മോദി രാജ് എല്ലാ ആശംസകളും (Exciting trailer awaits great effort Watch well Ask tough questions Democracy is really protected Modi Raj All the best)	1
4	വിദേശ രാജ്യങ്ങളിലെ ഇന്ത്യൻ വംശജരെ മാന്യമായ ജീവിതം നയിക്കാൻ സഹായിച്ച മികച്ച നയതന്ത്ര മോദി അതാണ്. (Modi is the best diplomat who has helped Indians in foreign countries lead a dignified life.)	1
5	ജിഎസ്ടി പോലെ മോഡി കാലഘട്ടത്തിൽ ഉണ്ടാക്കിയ ബാങ്ക് അക്കൗണ്ടുകൾ നീക്കം ചെയ്യുന്നില്ല ആരാണ് ഈ വിഡ്ഢി പദ്ധതിക്ക് പണം നൽകുന്നത് . (Bank accounts created during the Modi era will not be removed like GST Who pays for this stupid scheme.)	2
6	വോട്ട് തേടുന്ന ഒരാളെ നിങ്ങൾ കാണുമ്പോഴെല്ലാം മോദിക്കെതിരെ അക്രമത്തിന് പ്രേരിപ്പിക്കുന്ന കുറ്റാരോപിതനായ ഗൗഡയ്ക്ക് ശക്തമായ അടി നൽകുക. (Whenever you see someone seeking votes, give a strong blow to Gowda, who is accused of inciting violence against Modi.)	2

Fig. 2. Sample Dataset.

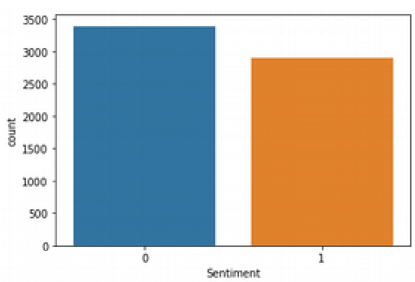

Fig. 3. Sentiment Distribution of Dataset I.

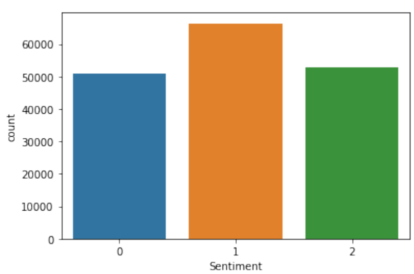

Fig. 4. Sentiment Distribution of Dataset II.

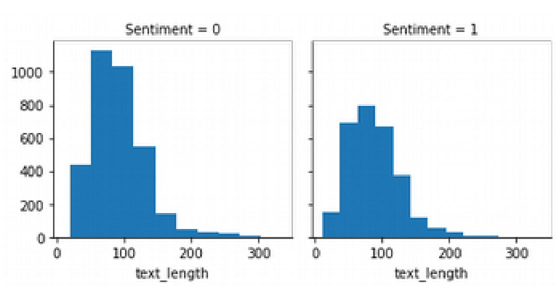

Fig. 5. Text Length of the Dataset I.

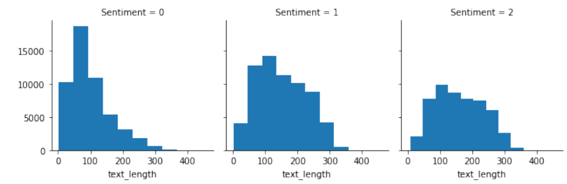

Fig. 6. Text Length of Dataset IIL.

```
['ഇന്നലെയും',
 'ഒരു',
 'ജീവൻ',
 'നഷ്ടപ്പെട്ടു',
 'കഴിഞ്ഞു',
 'ഈ',
 'നശിച്ച',
 'റോഡ്',
 'കാരണം',
 'മനഃപൂർവ്വമായ',
 'നരഹത്യക്ക്',
 'കേസ്',
 'എടുക്കണം',
 'സർ',
 'ഈ',
 'പൈശാചികർക്കെതിരെ',
 '.'],

['വ്യാജരേഖകൾ',
 'ചമച്ച്',
 'സഹകരണ',
 'ബാങ്കുകളിൽ',
 'നിന്നും',
 'ഒരു',
 'കോടിയിലധികം',
 'രൂപ',
 'തട്ടിപ്പ്',
 'നടത്തി',
 '.'],
```

Fig. 7. Tokenized Corpus.

```
[5869, 5870, 5871, 5126, 5712, 5015, 2943, 13],
[4620, 137, 5872, 2943, 5873, 680, 5874, 13],
[5875, 5876, 5877, 5878, 5879, 5880, 5881, 13],
[5882, 5883, 5884, 2943, 5885, 5042, 1974, 5830, 5886, 13],
[5310, 140, 5887, 2943, 5888, 5267, 5268, 5889, 5890, 13],
[640, 644, 414, 5874, 674, 414, 5891, 2350, 5668, 2943, 5892, 5893, 13],
```

Fig. 8. Index Value Assigned to Each Word in a Sentence.

```
array([[    0,     0,     0, ...,    11,    12,    13],
       [    0,     0,     0, ...,    25,    26,    13],
       [    0,     0,     0, ...,    28,    29,    13],
       ...,
       [    0,     0,     0, ...,  1266, 20118,    13],
       [    0,     0,     0, ...,   165,  3397,    13],
       [    0,     0,     0, ...,  1703,  2287,    13]], dtype=int32)
```

Fig. 9. Sequence after Zero Padding.

is given to a dense layer. Output of the dense layer is followed by a linear SVM. The three different hybrid models used in this study are as follows:

- Word Vector $->$ CNN + LSTM $->$ Dense layer $->$ SVM
- Word Vector $->$ CNN + Bi-LSTM $->$ Dense layer $->$ SVM
- Word Vector $->$ CNN + GRU $->$ Dense layer $->$ SVM

Feature extraction is done using the convolution layer of CNN whereas feature reduction is done using the max-pooling

```
array([[ 0.36599999, 0.058 , -0.37479998, ..., -0.3773 ,
        -0.44199998, 0.1278 ],
       [-0.01850001, 0.10969999, 0.0535 , ..., -0.1031 ,
        -0.32610001, -0.29970001],
       [ 0.77069998, 0.5072 , -0.54899997, ..., -1.50040001,
        -0.66089999, -0.4816 ],
       ...,
       [ 0.3575 , 0.2539 , -0.79189999, ..., 0.06089999,
        -0.2482 , 0.2326 ],
       [ 0.67430001, 0.0322 , 0.27570002, ..., -0.6824 ,
        -0.4373 , 0.3796 ],
       [ 0.46149999, 0.0568 , -0.12809999, ..., -1.1345 ,
        -0.7934 , -0.0933 ]])
```

Fig. 10. Word Embedding.

layer [9]. The convolutional layer uses filters for extracting important attributes from the data. Different filters are utilized for various applications with distinct kernel sizes. Kernel size represents the n-gram representation. ReLU is the commonly used activation function in convolutional layers. The pooling layer consolidates the output from the convolutional layer by selecting optimal data from the previous layer. Thus, reducing the dimension of feature vectors. The final layer of CNN is a fully connected neural network. LSTM consists of different elements, including an input gate, forget gate, memory cell, hidden state and an output gate [40]. LSTM removes long-term dependency but keeps some useful information. Bi-LSTM [41] creates the exact copy of LSTM in a backward direction also. The output of both forward and backward LSTM hidden states are combined at each step. GRU [42] is the simplified model of LSTM by removing the output gate. Thus, it reduces the complexity involved in LSTM architecture.

A. Evaluation Measures

The model is evaluated for making standard metrics like Precision, Recall, F1-score, and Accuracy using the confusion matrix [43]. The evaluation measures are formulated using the following equations.

Precision (positive classification) = $\frac{TP}{TP+FP}$
Precision (negative classification) = $\frac{TN}{TN+FN}$
Recall (positive classification) = $\frac{TP}{TP+FN}$
Recall (negative classification) = $\frac{TN}{TN+FP}$

Where, TP represents true positive, TN: true negative, FP: false positive, and FN: false negative.

F1-score = $\frac{2*precision*recall}{precision+recall}$

Accuracy is measured based on how many sentences are correctly classified among total sentences.

Accuracy = $\frac{TN+TP}{TP+TN+FP+FN}$

B. Proposed Hybrid Architecture

The details of the proposed hybrid models are given as follows:

Algorithm 1: HYBRID MODEL OF CNN AND LSTM

Input: S // S is a corpus which consists a set of sentences S_1, S_2, ... S_n
Output: Y // Labelled as 0 or 1 for Dataset I and 0, 1 or 2 for Dataset II
for each sentence S_i in S do
$W_i = Embedding(S_i)$ // Word vector in 300 dimensional space
end
for each W_i do
$C_i = CNN(W_i)$
$L_i = LSTM(W_i)$
end
for each C_i and L_i do
$O_i = Dense(C_i, L_i)$
end
for each O_i do
$Y_i = LinearSVM(O_i)$ // Linear SVM used as output layer activation function
end

Algorithm 2: HYBRID MODEL OF CNN AND BILSTM

Input: S // S is a corpus which consists a set of sentences S_1, S_2, ... S_n
Output: Y // Labelled as 0 or 1 for Dataset I and 0, 1 or 2 for Dataset II
for each sentence S_i in S do
$W_i = Embedding(S_i)$ // Word vector in 300 dimensional space
end
for each W_i do
$C_i = CNN(W_i)$
$B_i = BiLSTM(W_i)$
end
for each C_i and G_i do
$O_i = Dense(C_i, B_i)$
end
for each O_i do
$Y_i = LinearSVM(O_i)$ // Linear SVM used as output layer activation function
end

1) Model 1: Model 1 combines both CNN and LSTM. The word vector is given to CNN and LSTM parallelly. Filter size and kernel size of CNN are 32 and 2 respectively. The LSTM has 50 neurons. The merged output is given to a dense layer with 20 neurons. Further, the output is given to the linear SVM. SVM classifies the Dataset I as positive or negative, whereas Dataset II as positive, negative or neutral. The psudo-code representation of Model I is shown in Algorithm 1.

2) Model II: The second hybrid architecture combines both CNN and Bi-LSTM model. All the other layers are the same as in Model I. The psudo-code representation of Model II is shown in Algorithm 2.

3) Model III: The third hybrid model combines both CNN and GRU. The merged output is given to the dense layer,

Algorithm 3: HYBRID MODEL OF CNN AND GRU

```
Input: S // S is a corpus which consists a set of
   sentences S_1, S_2, ... S_n
Output: Y // Labelled as 0 or 1 for Dataset I and 0, 1
   or 2 for Dataset II
for each sentence S_i in S do
W_i = Embedding(S_i) // Word vector in 300
   dimensional space
end
for each W_i do
C_i = CNN(W_i)
G_i = GRU(W_i)
end
for each C_i and G_i do
O_i = Dense(C_i, G_i)
end
for each O_i do
Y_i = LinearSVM(O_i) // Linear SVM used as output
   layer activation function
end
```

followed by linear SVM. The psudo-code representation of Model III is shown in Algorithm 3.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The implementation of this work is carried out using Google Colab [46] with Keras [44] and TensorFlow [45] libraries. The three different baseline DNN models (LSTM, BiLSTM, GRU) and three hybrid models (CNN+LSTM, CNN+BiLSTM, CNN+GRU) are applied to two different datasets for SC. The dataset is split into 80:20 where 80% dataset is used for training while 20% is used for testing. For Dataset I, the vocabulary size is set to 8353 and for Dataset II, the vocabulary size is set to 58846 by removing the foreign words and least frequent words present in the text. The maximum sequence length of Dataset I is 32 and of Dataset II is 151. Table II represents the optimal value of hyperparameters chosen during the training phase of both baseline and hybrid models on Dataset I. The percentage of rightly classified and wrongly classified datasets are shown in Fig. 11 and Fig. 12. respectively. The confusion matrices of hybrid models like CNN+LSTM, CNN+Bi-LSTM, and CNN+GRU for Dataset I are shown in Fig. 11(a), Fig. 11(b), and Fig. 11(c), and for Dataset II is shown in Fig. 12(a), Fig. 12(b) and Fig. 12(c) respectively. Precision, Recall, F1-score and accuracy are used as evaluation measures. Table III represents the evaluation measures on Dataset I. Table IV represents the loss and accuracy of training, validation and test dataset. The optimal value of hyperparameters on Dataset II is depicted in Table V. Table VI is the evaluation measures on Dataset II. Table VII represents the loss and accuracy of training, validation and test dataset of Dataset II. Table IV and Table VII show that the CNN + GRU got better prediction accuracy of 87.23% for Dataset I and CNN + BiLSTM got an accuracy of 74% for Dataset II. The model architecture for CNN + GRU for Dataset I and CNN + BiLSTM for Dataset II are shown in Fig. 13 and Fig. 14, respectively. The bar chart shown in Fig. 15 and Fig. 16 compares the accuracy of baseline models with hybrid models for Dataset I and Dataset

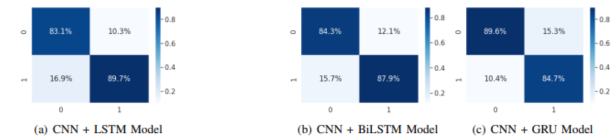

Fig. 11. Confusion Matrices of Hybrid Models on Dataset I.

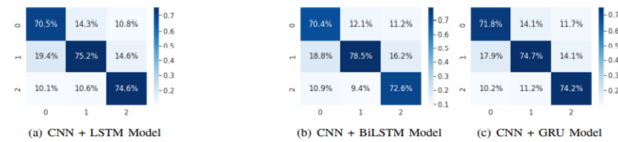

Fig. 12. Confusion Matrices of Hybrid Models on Dataset II.

II.

Discussion: Twitter is the most prominent platform for expressing opinions and suggestions about daily phenomena. Hence, tweets are considered for the SA. Both the baseline and hybrid DL models classified Dataset I as positive or negative and Dataset II as positive, negative and neutral. The best model is selected based on the accuracy of test dataset. The study of literature shows that the hybrid model performed well compared with baseline models for English and some Indian languages. This study analyzes that the hybrid models also show better prediction accuracy for the Malayalam language. Since the pre-trained vector fastText is not performed well on our datasets, the word embedding vector is created during the training phase of the embedding layer. After experimenting with various dimensions, the word vector is mapped to a 300-dimensional space for better prediction. Hybrid models improved the performance by nearly 2% to 3% compared with

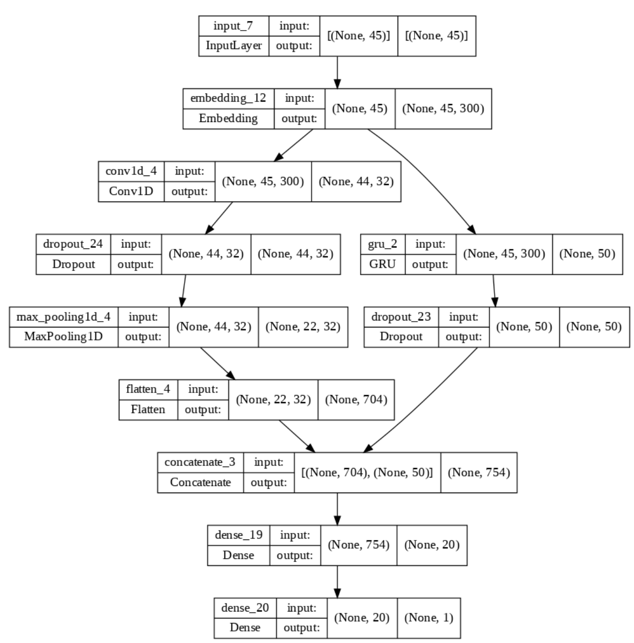

Fig. 13. Hybrid Architecture of CNN + GRU for Dataset I.

TABLE IV. VALUES OF HYPERPARAMETERS ON DATASET I

Model	Embedding Dimension	Number of Neurons in Dense Layer	Dropout	Optimizer	Kernel Size	Filter Size / Number of Neurons	Output Layer Activation Function	Loss Function
LSTM	300	20	0.4	Adam		50	sigmoid	binary_crossentropy
Bi-LSTM	300	20	0.4	Adam		50	sigmoid	binary_crossentropy
GRU	300	20	0.4	Adam		50	sigmoid	binary_crossentropy
CNN + LSTM	300	20	0.4	Adam	2	32, 50	linear SVM	squared_hinge
CNN + BiLSTM	300	20	0.4	Adam	2	32, 50	linear SVM	squared_hinge
CNN + GRU	300	20	0.4	Adam	2	32, 50	linear SVM	squared_hinge

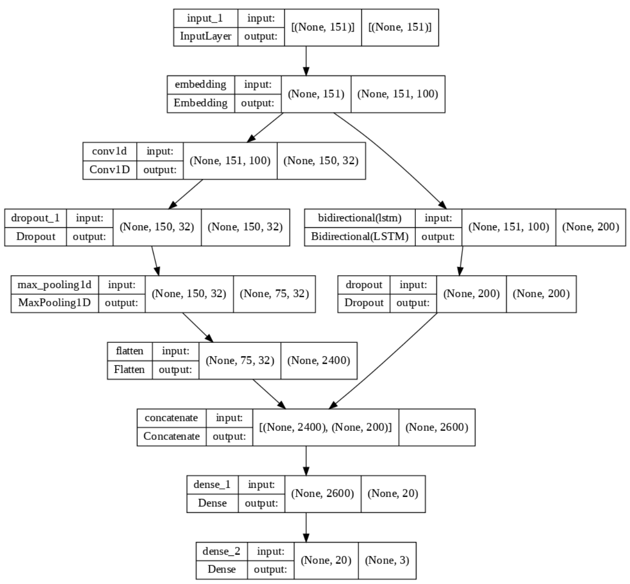

Fig. 14. Hybrid Architecture of CNN + BiLSTM for Dataset II.

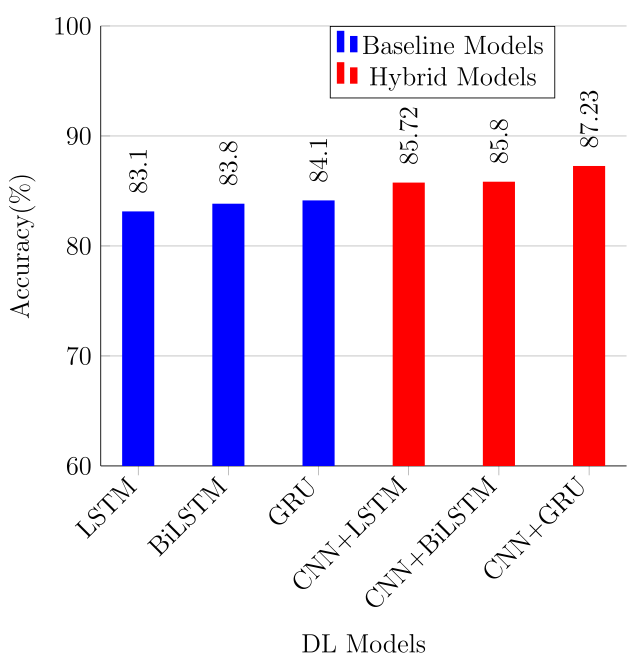

Fig. 15. Comparing Accuracy of DL Models on Dataset I.

TABLE V. EVALUATION MEASURES ON DATASET I

Model	Sentiment	Precision	Recall	F1-Score	Support
LSTM	Positive	0.84	0.78	0.81	581
	Negative	0.82	0.87	0.85	680
Bi-LSTM	Positive	0.80	0.87	0.83	581
	Negative	0.88	0.81	0.84	680
GRU	Positive	0.85	0.80	0.82	581
	Negative	0.83	0.88	0.86	680
CNN + LSTM	Positive	0.90	0.78	0.83	581
	Negative	0.83	0.92	0.87	680
CNN + Bi-LSTM	Positive	0.88	0.80	0.84	581
	Negative	0.84	0.91	0.87	680
CNN + GRU	Positive	0.85	0.88	0.86	581
	Negative	0.90	0.86	0.88	680

TABLE VI. TRAINING, VALIDATION AND TEST DATA ACCURACY OF DATASET I

Model	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
LSTM	0.0892	0.9586	0.4433	0.8571	0.5120	0.831
Bi-LSTM	0.1359	0.9482	0.3740	0.8552	0.4126	0.838
GRU	0.2068	0.9167	0.3659	0.8373	0.3217	0.8410
CNN + LSTM	0.1078	0.9548	0.5915	0.8373	0.4169	0.8572
CNN + BiLSTM	0.1146	0.9473	0.5917	0.833	0.4288	0.858
CNN + GRU	0.1191	0.9486	0.4917	0.8472	0.4414	0.8723

baseline models for Dataset I and 1% to 2% for Dataset II. The combined architecture of DNN models extracts better features rather than a single model. CNN extracts the local features and variants of RNN extracts the long-term features from the text data. The hyperparameters like the number of neurons in LSTM, BiLSTM and GRU are set to 50 for Dataset I and 100 for Dataset II after several trials with varying neurons. The number of neurons at dense layer is selected as 20 for both datasets. The sigmoid activation function is used at the output layer for Dataset I and Softmax for Dataset II. But, for both datasets, LinearSVM is selected as the best activation function for hybrid models. The loss function binary_crossentropy, categorical_crossentropy and squared_hinge are applied for sigmoid, softmax and SVM activation functions. Adam optimizer is used for all the models. For small dataset, GRU performed well and for large dataset Bi-LSTM shows better accuracy in baseline models. Therefore, the combined architecture of CNN and GRU got better prediction accuracy for Dataset I and CNN + BiLSTM shows better accuracy for Dataset II.

VI. CONCLUSION

The expeditious growth of social media content in regional languages have led to the importance of SA in native languages. The advancement of ML and DL models has improved the performance of NLP applications. Since general people in Kerala use their native language to express their suggestions and opinions in social media platforms like Twitter, Facebook,

TABLE VII. VALUES OF HYPERPARAMETERS FOR DATASET II

Model	Embedding Dimension	Number of Neurons in Dense layer	Dropout	Optimizer	Kernel Size	Filter Size/ Number of Neurons	Output Layer Activation function	Loss function
LSTM	300	20	0.4	Adam		100	Softmax	categorical_crossentropy
Bi-LSTM	300	20	0.4	Adam		100	Softmax	categorical_crossentropy
GRU	300	20	0.4	Adam		100	Softmax	categorical_crossentropy
CNN + LSTM	300	20	0.4	Adam	2	32 , 100	linear SVM	squared_hinge
CNN + Bi-LSTM	300	20	0.4	Adam	2	32, 100	linear SVM	squared_hinge
CNN + GRU	300	20	0.4	Adam	2	32, 100	linear SVM	squared_hinge

TABLE VIII. EVALUATION MEASURES ON DATASET II

Model	Sentiment	Precision	Recall	F1-Score	Support
LSTM	Positive	0.75	0.70	0.73	13271
	Negative	0.69	0.79	0.74	10560
	Neutral	0.70	0.65	0.68	10169
Bi-LSTM	Positive	0.74	0.71	0.73	13271
	Negative	0.73	0.76	0.74	10560
	Neutral	0.69	0.70	0.69	10169
GRU	Positive	0.73	0.73	0.73	13271
	Negative	0.74	0.78	0.76	10560
	Neutral	0.71	0.67	0.69	10169
CNN + LSTM	Positive	0.75	0.73	0.74	13271
	Negative	0.75	0.77	0.76	10560
	Neutral	0.70	0.70	0.70	10169
CNN + Bi-LSTM	Positive	0.78	0.71	0.75	13271
	Negative	0.73	0.78	0.75	10560
	Neutral	0.70	0.73	0.72	10169
CNN + GRU	Positive	0.75	0.75	0.75	13271
	Negative	0.74	0.76	0.75	10560
	Neutral	0.72	0.69	0.70	10169

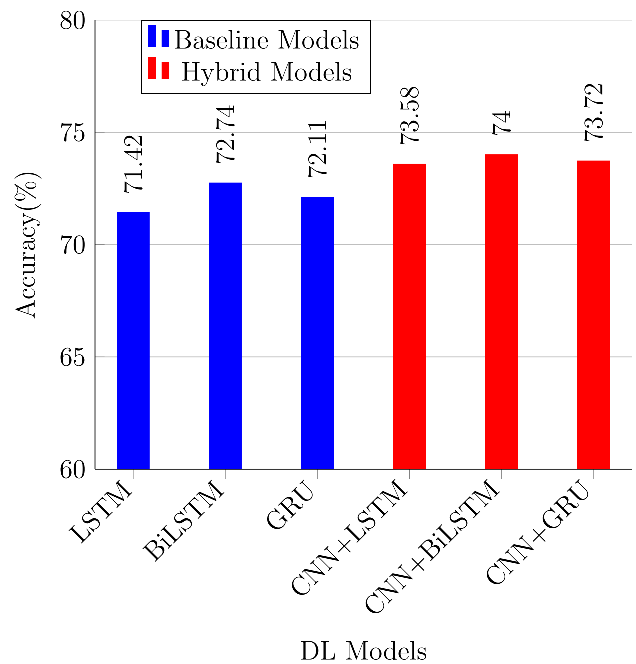

Fig. 16. Comparing Accuracy of DL Models on Dataset II.

TABLE IX. TRAINING, VALIDATION AND TEST DATA ACCURACY OF DATASET II

Model	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Test Loss	Test Accuracy
LSTM	0.1243	95.76	0.7845	73.12	0.8245	71.42
BiLSTM	0.2698	89.56	0.8296	73.36	0.7982	72.74
GRU	0.2841	93.13	0.8123	72.47	0.8379	72.11
CNN + LSTM	0.1191	94.81	0.7565	73.51	0.7842	73.58
CNN + Bi-LSTM	0.2513	88.50	0.5437	74.38	0.6184	74.0
CNN + GRU	0.2132	91.40	0.6231	73.85	0.7123	73.72

etc. The automation of SA in Malayalam is essential for analysis. Also, DL models like LSTM, BiLSTM and GRU prevail over other methods of SC of text datasets. This paper proposes SA of Malayalam tweets using hybrid DNN models. Three different baseline DNN models, namely, LSTM, BiLSTM, GRU and further hybrid models combined with CNN and variants of RNN, including LSTM, Bi-LSTM and GRU, have been used for SC of tweets as positive, negative, or neutral. The experiments were conducted on two different datasets. Hybrid DL models performed well in both datasets. The combined architecture of CNN and the variants of RNN extract the best features compared to a single model. The word vector is formed during the training phase and mapped into a 300-dimensional vector space to achieve semantically similar words in the same space. Among the hybrid models, CNN + GRU shows the highest accuracy of 87.23 % for Dataset I and CNN + BiLSTM show better performance for Dataset II with an accuracy of 74%.

The major challenge was lack of benchmark datasets and lexical resources in Malayalam language. Hence, SA was done only on two different datasets, which is insufficient to standardize the results. Future work is aimed to develop benchmark datasets to achieve standardization of text analysis in Malayalam language.

REFERENCES

[1] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. “Thumbs up? Sentiment classification using machine learning techniques.” arXiv preprint cs/0205070 2002.

[2] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets.” Neural computation vol. 18, no. 7, pp:1527-1554, 2006.

[3] Joshi, Aditya, A. R. Balamurali, and Pushpak Bhattacharyya. ”A fall-back strategy for sentiment analysis in hindi: a case study.” Proceedings of the 8th ICON 2010.

[4] Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781 2013.

[5] Devlin, Jacob, et al. “Bert: Pre-training of deep bidirectional transformers for language understanding.” arXiv preprint arXiv:1810.04805 2018.

[6] Grave, Edouard, et al. “Learning word vectors for 157 languages.” arXiv preprint arXiv:1802.06893 2018.

[7] Abid, Fazeel, et al. “Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter.” Future Generation Computer Systems Vol. 95, pp: 292-308, 2019.

[8] Alharbi, Ahmed Sulaiman M., and Elise de Doncker. “Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information.” Cognitive Systems Research vol. 54, pp: 50-61, 2019.

[9] Khattak, Asad Masood, et al. "Tweets classification and sentiment analysis for personalized tweets recommendation." Complexity 2020.

[10] Pandey, Avinash Chandra, Dharmveer Singh Rajpoot, and Mukesh Saraswat. "Twitter sentiment analysis using hybrid cuckoo search method." Information Processing & Management vol. 53, no.4, pp: 764-779, 2017.

[11] Vateekul, Peerapon, and Thanabhat Koomsubha. "A study of sentiment analysis using deep learning techniques on Thai Twitter data." 2016 13th international joint conference on computer science and software engineering (JCSSE). IEEE, 2016.

[12] Hassan, A., & Mahmood, A. "Deep learning approach for sentiment analysis of short texts." In 2017 3rd international conference on control, automation and robotics (ICCAR) pp. 705-710, 2017.

[13] Hegde, Siddhanth U., A. S. Zaiba, and Y. Nagaraju. "Hybrid cnn-lstm model with glove word vector for sentiment analysis on football specific tweets." 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). IEEE, 2021.

[14] Dang, Cach N., María N. Moreno-García, and Fernando De la Prieta. "Hybrid Deep Learning Models for Sentiment Analysis." Complexity 2021.

[15] Martín, Carlos Alberto, et al. "Using deep learning to predict sentiments: case study in tourism." Complexity 2018.

[16] Monika, R., S. Deivalakshmi, and B. Janet. "Sentiment analysis of US airlines tweets using LSTM/RNN." 2019 IEEE 9th International Conference on Advanced Computing (IACC). IEEE, 2019.

[17] Chen, Tao, et al. "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN." Expert Systems with Applications vol. 72, pp: 221-230, 2017.

[18] Srinidhi, H., G. M. Siddesh, and K. G. Srinivasa. "A hybrid model using MaLSTM based on recurrent neural networks with support vector machines for sentiment analysis." Engineering and Applied Science Research vol. 47, no. 3, pp: 232-240, 2020.

[19] Akhtar, Md Shad, et al. "A hybrid deep learning architecture for sentiment analysis." Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 2016.

[20] Vo, Quan-Hoang, et al. "Multi-channel LSTM-CNN model for Vietnamese sentiment analysis." 2017 9th international conference on knowledge and systems engineering (KSE). IEEE, 2017.

[21] Rehman, Anwar Ur, et al. "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis." Multimedia Tools and Applications vol. 78, no.18, pp: 26597-26613, 2019.

[22] Kastrati, Zenun, et al. "A deep learning sentiment analyser for social media comments in low-resource languages." Electronics vol. 10, no. 10, pp: 1133, 2021.

[23] Kaur, Harleen, et al. "A proposed sentiment analysis deep learning algorithm for analyzing COVID-19 tweets." Information Systems Frontiers vol. 23, no. 6, pp: 1417-1429, 2021.

[24] Ghourabi, Abdallah, Mahmood A. Mahmood, and Qusay M. Alzubi. "A hybrid CNN-LSTM model for SMS spam detection in Arabic and english messages." Future Internet vol. 12, no.9 pp: 156, 2020.

[25] Yenter, Alec, and Abhishek Verma. "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis." 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). IEEE, 2017.

[26] Ombabi, Abubakr H., Wael Ouarda, and Adel M. Alimi. "Deep learning CNN–LSTM framework for Arabic sentiment analysis using textual information shared in social networks." Social Network Analysis and Mining vol. 10, no. 1, pp: 1-13, 2020.

[27] Zhou, Kai, and Fei Long. "Sentiment analysis of text based on CNN and bi-directional LSTM model." 2018 24th International Conference on Automation and Computing (ICAC). IEEE,2018.

[28] Salur, Mehmet Umut, and Ilhan Aydin. "A novel hybrid deep learning model for sentiment classification." IEEE Access 8 pp: 58080-58093, 2020.

[29] Mohandas, Neethu, Janardhanan PS Nair, and V. Govindaru. "Domain specific sentence level mood extraction from Malayalam text." 2012 International Conference on Advances in Computing and Communications. IEEE, 2012.

[30] Nair, Deepu S., et al. "SentiMa-sentiment extraction for Malayalam." 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2014.

[31] Anagha, M., et al. "Lexical resource based hybrid approach for cross domain sentiment analysis in Malayalam." Int J Eng Sci 15 (2014): 18-21.

[32] Anagha, M., et al. "Fuzzy logic based hybrid approach for sentiment analysisl of malayalam movie reviews." 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES). IEEE, 2015.

[33] Nair, Deepu S., et al. "Sentiment Analysis of Malayalam film review using machine learning techniques." 2015 international conference on advances in computing, communications and informatics (ICACCI). IEEE, 2015.

[34] Kumar, S. Sachin, M. Anand Kumar, and K. P. Soman. "Sentiment analysis of tweets in malayalam using long short-term memory units and convolutional neural nets." International Conference on Mining Intelligence and Knowledge Exploration. Springer, Cham, 2017.

[35] Rahul, M., R. R. Rajeev, and S. Shine. "Social Media Sentiment Analysis For Malayalam." International Journal of Computer Sciences and Engineering, E-ISSN (2018): 2347-2693.

[36] Kumar, S., M. Anand Kumar, and K. P. Soman. "Identifying Sentiment of Malayalam Tweets Using Deep Learning." Digital Business. Springer, Cham, pp: 391-408, 2019.

[37] Soumya, S., and K. V. Pramod. "Fine Grained Sentiment Analysis of Malayalam Tweets Using Lexicon Based and Machine Learning Based Approaches." 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE). IEEE, 2021.

[38] Soumya, S., and K. V. Pramod. "Sentiment analysis of Malayalam tweets using machine learning techniques." ICT Express vol. 6, no. 4, pp: 300-305, 2020.

[39] Soumya, S., and K. V. Pramod. "Sentiment analysis of Malayalam tweets using different deep neural network models-case study." 2019 9th International Conference on Advances in Computing and Communication (ICACC). IEEE, 2019.

[40] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation vol. 9, no. 8, pp: 1735-1780, 1997.

[41] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with deep bidirectional LSTM." 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013.

[42] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 2014.

[43] Su, Louise T. "Evaluation measures for interactive information retrieval." Information Processing & Management vol. 28, no.4, pp: 503-516, 1992.

[44] Francois, C. H. O. L. L. E. T. "Keras: The Python deep learning library." keras. io (2015).

[45] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).

[46] Google. Google Colab. Accessed: Jan. 10, 2020. [Online]. Available: https://colab.research.google.com