International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Analysis of Recurrent Neural Networks-LSTM and 1D Convolutional Neural Network in Wake Word Detection System of Regional Dialects

Chaitra G. P.[1], Shylaja S. S.[2]

Submitted: 05/02/2024 **Revised:** 12/03/2024 **Accepted:** 20/03/2024

Abstract*:* This paper aims in comparative analysis of two deep neural network techniques -RNNs and 1D-CNN models in building the WWD system in Kannada for five various locations with their dialects in the state of Karnataka and they are:- Dharwad, Tulu, Dogg anal, Urban Kannada in addition the Kodagu region as well. The customized WWD system is built using Conv1d model with 97% accuracy compared to RNNs with 42.5% precision. The variation present with local dialects are finely specified by 1d CNN in analogous study along with RNN model in verifying on the dialect dataset on the labels impending which are contrasting and the implementation of Conv1d makes better predictions on the Idiom Dataset.

Keywords: *MFCC, CONV 1D, Dialect Identification, RNN(LSTM), Trainable Parameters, WWD.*

1. Introduction

With the development of technology in recent years, particularly in the fields of data science and artificial intelligence,
different Deep neural network algorithms are applied in creating voice-activated applications to give users an ideal experience [19].
RNNs are suitable for the tasks which has a sequence in the statements, which when compared with the short word sequences specially for wake words is challenging to implement. For Indian Languages each dialect holds its own word structure which is not ordered and to overcome such interferences CNNs play a very major role in extracting the audio features through MFCCs and on spatial relationship.
Indian population, majorly depends on agriculture as the main source of income. [12] has illustrated about analysing the various factors which influence the adoption of technology can be greatly guided by having a clear understanding of how farmers view contemporary facilities.
Mobile apps can become effective instruments for providing suitable specifics without the interference of the outside parties, to rural farmers about their needs in agricultural production [11].
The government and other commercial companies have released a number of agriculture-related apps which is based on chatbots and voice help. Among the apps is Krishify, which enables farmers to look for any topic associated with agriculture. Next, is Ag_Next which consolidates different services related to agriculture with the platform e-Nam [22]. FarmBee is another application, which is accessible in 10 various indian-languages, which brings in the information about the life cycle of the crop. Kumar [10]has illustrated challenges encountered by farmers in rightly utilizing using these applications.

[1] *computer science department, PES university Bangalore. Chaitragp@pesu.pes.edu*
[2] *computer science department, PES university Bangalore. Shylaja.sharath@pes.edu*

In India, over 1600 dialects are within the 120 major spoken languages spoken.
It is important to pay attention to the truth about the language gap if technology must advance to such a fine degree in order to meet the requirements of rural farmers [10]. For farmers to communicate with technology, a connection must be formed between them and the technology. This is only possible when information is disclosed to the farmers in their dialect [9]. This expresses an essential idea that the farming and rural communities may find relatable when using their regional dialect to communicate with the device.
Dialect identification is one of the difficult parts of speech recognition tasks. It is difficult with respect to separating the dialect from the elementary language of communication with respect to the order of words and complexity with phonetic systems that overlap [1]. [5] illustrates about the inefficiency in low-resource languages for modelling DID.
Variations are very minute in terms of utterance made for the same word with varied forms [4]. Few parameters have to be studied in terms of features related to prosodic including phonology, intonation and vocabulary along with grammar.
To build a wake word detection, completely depends on the type of data and the variation it has in terms of prosodic features, has to be focused. Dialects have their own sentence/word structure which depends completely on the speakers ethnic/social background. Capturing the details in the features plays a very critical role in deciding the DNN technique to be implemented. RNNs have proven to be effective in sequential structure of the sentences which is challenging when it comes to short word sequences in wake words with varied dialects with minimum of 3seconds [21]. The order is different in each dialect and differentiating them is the key of this research paper.

2. Literature Review

[25] Writers illustrate about universal decoder for phones which aid in building the WWD system. Primarily Two sets of data

consisting Persian spoken language is taken and trained on the syllables and tuning the parameters. Then, once the WWD system is built and reached to its precision LDA- Levenshtein Distance algorithm is used to compute the score for confidence of the output. It is used to get the high precision of the framework proposed by the Authors. Also, this technique is helpful in achieving high accuracies for noisy signals.

[26] Authors mention about CNN framework which is based on alignment loss for classification to spot the start of the keyword. Experiment also shoe the model accuracy is improved by 73% over traditional HMM-DNN models.

[2] Mentioned about LSTM (Long Short-term Memory) technique in WWD, IN supporting the for a long time dependability within timestep present in the done on spectrogram. In backpropagation technique

present along with earlier components is taken considered as the input to a Neuron. The authors additionally outline the reasoning behind why LSTM were built-to take care of the diminishing gradient and blowing up issues in RNN's. LSTM containing procedures are well-suited for long sequential data.

[21] proposed the wake word speech recognition system with LSTM along with designing the scoring method using modified zero normalization. The The framework is capable of adapting and identifying multiple wake words with preference with relative improvement of 51.92%.

[18] This paper proposes a WWD system based on Transformers resulting in superior results compared to other algorithms on sequence modelling. The main agenda is about WWD which is a temporal model with short range, Transformers are not right option as they handle long sequence modelling. Attention mechanism is used in Transformers consisting long-term memory. It is composed of encoder_phase and decoder_phase which is attention-based, in which the encoder_phase encapsulates the specifics which is discovered in the initial sequence and the decoder_phase takes the resulted sequence and then results in individual output taking into account the prior results. Model is able to "attend" upon tokens that were previously created. Writers used a Lattice _Free Maximum Mutation_Information termed as LF_MMI procedure that consists of gradient_ stopping , anticipating the following data chunk, embedding-based techniques on locations present in sequence, also include layer_dependencies.

[4] Authors have proposed a system with recurrent neural network with transducers which can overcome the problem of overfitting in wake word spotting. It is a 3 step process-in the first step to address the overfitting issue a multitask training is included with CTC loss which is calculated for both Wake word spotting data and Automatic speech recognition data. Next, replaced by LSTM for sequence modelling. Further the model is improved by the technique of transfer learning.

For the dialect dataset in this research is not suitable to use RESNET as the dataset itself is small. [17] illustrates the approach used by Res2Net which is the superior version of ResNet. The capability in identifying WW of varying times has improved. Mob Voi datainfo-collection comprises of two WW which is having a false_rejection rate at approx. 12.0% in contrast with alternative frameworks. Residual networks as categorizing framework of widened responsive setting accelerates the identification ability with a smaller number of specifications. Primarily it is executed by mining the prominent characteristic through taking into account the locale-specific characteristics and then granting in

the features globally which are of the similar size in the present region with different lengths.

3. Method

This approach relies on multi-class classification issue which has the system where-in each input is related to a single class only. WW are employed so as to initiate communication which awakens the device in order to reciprocate the inquiries coming in. If the Device listens continuously to the conversation it may lead to the incompetence with security and also can cause-considerable amount of work for servers in handling every audio.

The Device begins to listen to the requests and when WW is discovered it awakens. There are five steps in the WWD system.

At First, we must get data ready through recording audio regarding a few seconds which contains WW with varied speech pattern(dialect) also audio containing other than the WW.

Second step, is to shift the raw audio data which is in the of waveform in time-domain, and if we need to more effectively assess and mine the pertinent features of audio_file. The transformation of waveform in time -domain to frequency-domain plays a critical role and in our work MFCCs are implemented. Every MFCCs are categorized and pickle files are used to save the labels and features for future usage. Depending on given dataset and issue at hand select the DL model resulting in the finest estimations. For WWD 1D Conv technique is best suitable for the varied data. Transform every MFCCs into 1-d array and give it in the form of input to convolution_layer1.

Next, we examine trained prototype for the purpose of prediction in which framework

hears the clip which is then categorized into a single classes, also results are written to .csv file.

This research has adopted same dataset collection and preprocessing technique for binary classification, multiclass classification and comparative analysis.

3.1. Data- Preparation

Information has been collected from five distinct Karnataka regions, predominantly from local areas with their own unique spoken forms in dialects. The locations comprises Karnataka Northern region Dharwad also Dogg anal location. South Karnataka regions comprises Kodagu, tulu_Kannada from the coast regions of Karnataka and Standard/Urban spoken Kannada from Bengaluru.

Sample consists of audio_file that has the audio recording of WW. The Audio has been captured in a short 3 sec uttering the word "Namaskara"with their related local dialects accompanying two or more words following the introductory phrase that are particular to the those regions. Non_WW, includes words/phrases other than wake words. Sounddevice is applied in capturing the speech followed by forming a Num_Py -array, next Scipy.io.wav helps in saving the Num_Py-array in the form of .wav format. Every dialect category has 100 clips consisting male and female speaking voices with varied ages. Augmentation of data is done which plays a critical role in creating the new samples by boosting the volume of audio file with changing speeds in the range of 0.7 - 1.4.

Every time when an audio is recorded it is stored inside a distinct file name which can later be beneficial in undertaking iteration every files. The audios were captured during on field visit during the survey. Same audio is used for the dataset preparation for precise outcomes.

3.2. Preprocessing

Two parameters are considered while recording: -save_path an empty directory which saves all audio files and another one is about how many the audio is captured. When all the speech is captured sample_rate is decided. [8] *Sample_rate is defined as rate of sampling audio/second. 44100 hertz is considered as the standard sample rate for the audio file*. A time_frame with 3sec is given to capture the audio.

The Librosa which is python library popularly implemented in analyzing speech data_collection. The preprocessing incorporates 3 main stages-

1. A raw_audio file is first loaded.
2. Then converted into .wav file and
3. Final stage is to snippet essential features out from the spectrum. *In* librosa the load() function assumes the path of the Num_Py as well as sample_rate of the speech. *librosa. Display()*is a function applied to visualize a spectrogram designed over matplot_lib.

Same process is implemented on other files. Following the preprocessing, files are categorized to their corresponding labels. Framing is done to all signals at intervals from 20 up-to 40ms since the audio signal will always fluctuate.

Once the audio files are labelled, next step is to mine pertinent pattern from the audio files. For this cause, -Mel Frequency Cepstral Coefficient is implemented as they result in more effective execution in acknowledging low frequency regions over high frequency regions. Also, is good at analyzing the resonances formed with the vocal tract. As a result, we can just identify the language position and ignore the noise. Variations in amplitude, pitch, length, speaker identification, and timber (resulting from the distinctiveness of each speaker's tone description) are among the minute but significant changes in speech signals that are noted in dialects.

MFCC gives important pieces of information pertaining to spectral bands for their changing rates. Changing the signal to frequency_ domain which is achieved by Fourier-transform in analyzing the two main features about spectral and power_elements in signal then using word embedding technique for the labels in one hot encoding.

To lower the dimensionality of the data, we employ the MFCC mean. It supports in eliminating fluctuating effects from either the instrument used for recording or by vocal_tract responses of the participants. Assigning labels from 0-6 for all the audios is done using Label encoding technique.

Next procedure is about framing pandas' data frame for the final data. This data frame is accessed in the course of
training period. Data_frame is stored in . pickle file in .csv format.

Pickling is the technique in Python which is used to store labelled dataset in its binary form for future applications[2]. We can send the pickled file over other platforms for viewers who may operate on equivalent data, in place with sending whole data leading to memory issues, in managing larger dataset. Also, it is helpful when data is lost to technical issues.

3.3. Design Architecture

Architecture of both RNN(LSTM) and 1D CNN has been customized for the dialect dataset.

In RNNs the learning happens at LSTM condition of the cell.

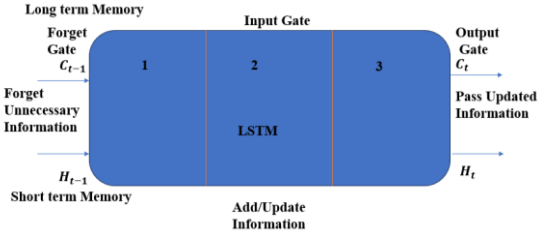


Fig. 1. LSTM Cell

LSTM Cell - Forget, Input Output Gate.

Forget_gate unlearns the unwanted information which are not relevant contextually.

$f_t = \sigma(x_{t^*}u_f + H_{t-1^*}w_f)$

$f_t = number\ or\ output\ between\ 0\ and\ 1$

$x_t = input\ from\ the\ current\ timestep$

$u_f = Weight\ metric\ with\ the\ input.$

$H_{t-1} = Hidden\ state\ from\ the\ previous\ step$

$w_f = weight\ metric\ with\ hidden\ state$

For overall cell condition:

$C_{t-1} *\ f_t = 0$ if $f_t = 0$ (forget entirely)

$C_{t-1} *\ f_t = 0$ if $f_t = 1$ (Remember everything)

Input_Gate:

$i_t = \sigma(x_t *u_f + H_{t-1} * w_i)$

N_t = tanh$(x_t *u_c + H_{t-1} * w_c)$

Tanh ranges from -1 to 1

Overall cell state:

$C_t = f_t * C_{t-1} + i_t * N_t$ (Update cell state with new information)

Output Gate:

$o_t = \sigma(x_t *u_o + H_{t-1} * w_o)$

$H_t = o_t$ * tanh(C_t)

Output = SoftMax(H_t)

Trainable Parameters:

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 40)                6720

 dropout_3 (Dropout)         (None, 40)                0

 dense_4 (Dense)             (None, 40)                1640

 dense_5 (Dense)             (None, 20)                820

 dropout_4 (Dropout)         (None, 20)                0

 dense_6 (Dense)             (None, 10)                210

 dropout_5 (Dropout)         (None, 10)                0

 dense_7 (Dense)             (None, 6)                 66

=================================================================
Total params: 9456 (36.94 KB)
Trainable params: 9456 (36.94 KB)
Non-trainable params: 0 (0.00 Byte)
```

Fig. 2. Summary of the RNN(LSTM) Model

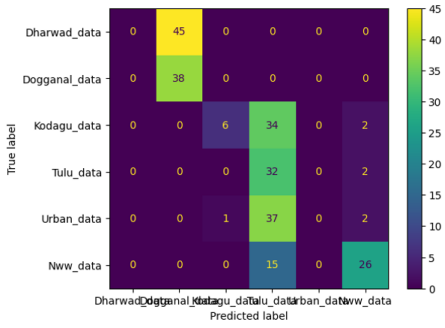
Total trainable parameters for the customized model for dialect dataset is 9456.

Dimension is considered by m x n where m is the input_vector dimension and n is number of hidden_state.

TP=(m x n) + n + (n x n = n(m+n+1)

LSTM unit has 4 set of parameters considered- *input_modulation gate, input forget and output gate*.

LSTM's trainable parameters is calculated — which is **4n(m+n+1)**.



RNN-LSTM				
Support	**Precision**	**Recall**	**F1-score**	**Support**
45	0.00	0.00	0.00	45
38	0.46	1.00	0.63	38
42	0.86	0.14	0.24	42
34	0.27	0.94	0.42	34
40	0.00	0.00	0.00	40
41	0.81	0.63	0.71	41
Loss:1.0432 Accuracy:0.4250				

Fig. 3. Confusion matrix representing predictions on Dialect Database with RNN-LSTM

Figure above depicts the Confusion matrix along with model classification report wherein it can be highlighted that with the structure similarity in dialects the model is misclassified the audio files.

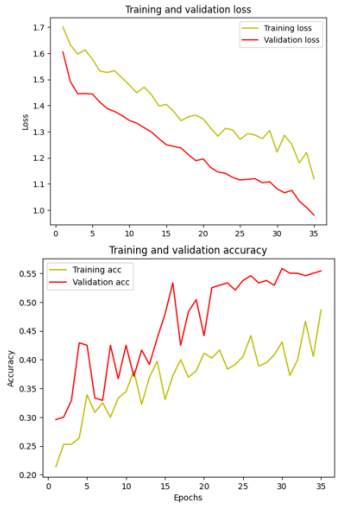


Fig. 4. Exhibits the loss and accuracy measures

3.4. 1 Dimensional CNN Design Architecture.

Given the dataset - the DLM which fits accurately is 1d CNNs. Shape of the input is considered 1-d format. Input to the convolutional layer is the MFCCs in the 1 dimensional array_format containing fourty distinct-Co-efficients. The shape of the input is represented as [40, 1]. The framework consist of 2 layers of convolutions – Conv_Layer_1 consist of 64 - filters and kernel_size of 3 and it is also in the form of one dimensional which follows activation_function mainly ReLU converting negative_values to 0 followed by max_pooling which of pool_size 2, subsequently the drop out of 0.25 of neurons. Conv_Layer_2 is made of 128 filters which is continued by ReLU- followed by max_pooling and drop out of 0.25 neurons. The output is then flattened and connected to Dense_layer consisting neurons of around 512 proceeded by drop out in neurons at 0.5 and SoftMax function for the last Dense_layer which encompasses six neurons which are the number of labels.

Padding-illustrated as 'same' encompass 0 at either ends of the input_array which gets output_shape with same dimension with shape of input

The shape of output is given:

(n + (2p-f /stride)) +1:- wherein

n. :- shape of input

p. :- zero at either ends in input.

f. :- size of the filter

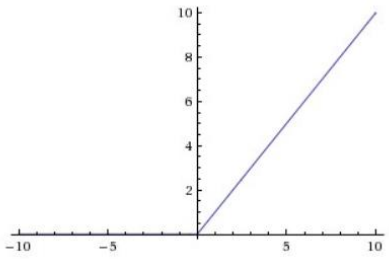
Stride: -mentioned - one

ReLU (Rectified linear unit), applied during Conv_layers returning zero on getting negative_value. With positive_ value :- take it as x, gives back the one(value).

Given as:

f (x) is given as max (0, x)

Graphically depicted as:



Max_pooling-applied on convolution_ layers. The Kernel transits on the feature_ map considering max_values from specific locations. Size of filter is specified during the Pooling process accounts for less compared to Feature-Map. Number of strides is 2 and compute the shape of output.

Dropout-layer-Implemented at Convolutional layers and dense layers. But has varied operations.

1. During the convolutional phase, dropout are employed at the rate of 0.2(lower) that aids in increasing the functionality without affecting the process of mining the prominent features.
2. Then, they are implemented at dense layer is implemented with the drop_out rate of 0.5 increasing precision of classification.

Softmax function-

$$S(y)_i = \frac{exp(y_i)}{\sum_{j=1}^{n} exp\,(y_j)}$$

y_i: $i\ th\ element\ of\ the\ input\ vctor$

y : input-vector of Softmax() consisting of n_ number of components with n-number of classes.

y_j: $term\ for\ normalization$

output ranges from 0 to 1.

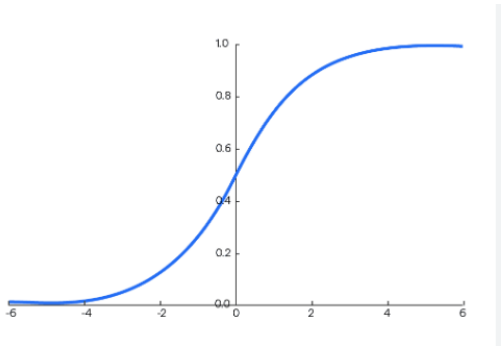
$exp(y_i)$:

$results\ in\ smaller\ value\ close\ to$ 0

but not 0.

Softmax() is implemented at outputs from dense-layer predominently during final stage of neural network in multi_class categorizing issue with n_number of classes. It gives back the output-vector which will be having the form in terms probability-ratings. Magnifies the maximum_value and lower_values are not considered.

Softmax graphical depiction:



Function which is the form of S in the graph ranges of 0 to 1 as 0.5 as midpoint. Output is given as 1in case of larger values and 0 when values are smaller or for negative values.

There is no set number for filters or layers that should be used; we should create the architecture based on whatever best fits the problem statement and produces the best expectations.

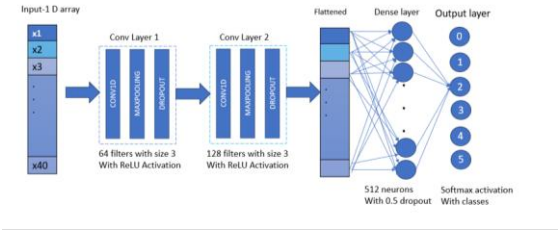


Fig. 5. Customized model architecture of 1D CNN

Trainable parameters Calculation:

1.Input-shape is termed with respect to X_train shape and its dimension.

2.Convolutional_Layer: Considering the weight metrices following expression is gives as:

(Shape related to filter length * number of filters in earlier layer + 1) * number of filters in present layer)).

3.Pooling_Layer: Gaining knowledge

is not taking place in this layer. L_specifications are zero.

4.FCT(fully connected layer):

(number of neurons in present layer * number of neurons in the earlier layer) + no: of neurons at present layer.

In summary below, there are around 683910 trainable parameters.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 40, 64)	256
max_pooling1d (MaxPooling1D)	(None, 20, 64)	0
dropout (Dropout)	(None, 20, 64)	0
conv1d_1 (Conv1D)	(None, 20, 128)	24704
max_pooling1d_1 (MaxPooling1D)	(None, 10, 128)	0
dropout_1 (Dropout)	(None, 10, 128)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 512)	655872
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 6)	3078

Total params: 683910 (2.61 MB)
Trainable params: 683910 (2.61 MB)
Non-trainable params: 0 (0.00 Byte)

Fig. 6. Summary of the 1D CNN Model

Model architecture is set then, compilation of the model.Two norms/specifics are taken into consideration: Categorical-cross_entropy function for loss the 'Adam' used as an optimizer which helps by controlling learning-rates in the whole training-period. Learning_rates depicts pace where weights are computed.

Training and Test data is divided in 60:40 ratio.

In Multi_class-classification issue, categorical cross-entropy is applied to calculate the loss.

Categorical-Cross-Entropy equation :

$$CE = -\sum_{i=1}^{i=N} y\_true_i \cdot log(y\_pred_i)$$

We are also applying function named to_categorical() which benefits the categorizing method in understanding the connectivity among the classes. It transforms the vectorofclasses into binaryclass of matrix.

Presence of minute fluctuations in dialects is hard in precisely depicting the optimal-weights and fine_tuning it for particular highlights which is reason for the variation in the waveform. Lower score, better execution by framework. Learning _rate and also convergence - quicker in contrast to other functions for loss such as -MSE. As there are smaller differences which are challenging in repairing weights must be modified correspondingly.

Upon the execution of these metrics, an allocated number of epochs is run before being stored in a format of hdf5 file for future usage. In this way, we may apply this pre-trained model to equivalent data-collections.

Training data is chunked down in batches. Learning_curves denotes state of training in every time_step . The more epochs in the model, the more accurate it becomes. As the accuracy goes up there is gradual decline in loss values.

Graphs depicts that the model performance is better in yielding estimations for the dataset.

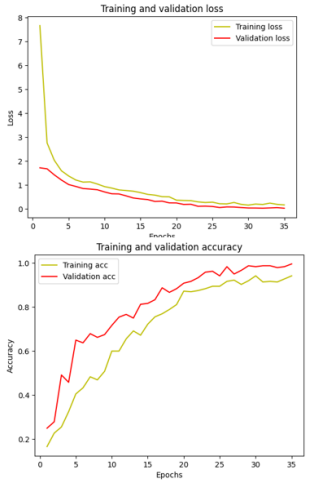
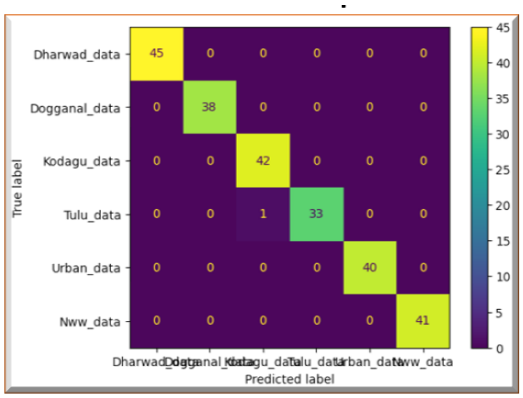


Fig. 7. Depicts the loss and accuracy curves

The framework fits the dataset well, as evidenced by the fact that both the training and validation losses drop and become stable at considerably earlier steps in the epoch.

Model examination is carried out using classification_report depicting the correct classification of audio files into their related labels.



Dialects	Labels	Precision	Recall	F1-score	Support
Dharwad	0	1.00	1.00	1.00	45
Dogganal	1	1.00	1.00	1.00	38
Kodagu	2	0.98	1.00	0.99	42
Tulu	3	1.00	0.97	0.99	34
Urban	4	1.00	1.00	1.00	40
Non-Wake Word	5	1.00	1.00	1.00	41

loss: 0.0354 - accuracy: 0.9917

Fig. 8. Confusion matrix with model classification report.

4. Result

The results from the suggested work with Conv1d model is one of the dialects in terms of label(class) is identified for the given audio file. Input is either through audio file or in terms of the utterance of the words/phrase which last at 3sec and then, the framework computes the label belonging to one of the classes with output written to .csv_file. Accuracy is 0.9917.

Other procedure involves the use of pre_trained frame work of the model where document of new audio is created by recording which is totally not seen in either training or in testing dataset. With the application of pre_trained framework stored in .hdf5 format- predictions are made then it is stored in .csv files.

Dialect diversity is shown in the dataset for the subject of our statement. Every dialect consist its own style and phrase for greeting. They are distinct w.r.t the usage of noun, pronoun and verb.

Figure 9 depicts the disintegration of Wake Word Sequence and the part of speech along with the place of order in the phrase.

The illustrated work in the papers in reference list [2],[8] and [18] are relied on key phrases -max_length of two words way too small and fluctuation is not noticed between different key_phrases in contrast to dialect data collected to serve the purpose of our research paper. As there is an absence of the order in the phrases challenges the view in applying RNNs(LSTM) to serve the motive of analysis.

Contrasting analysis is performed with RNN model along with 1d CNNs to verify the accuracy.

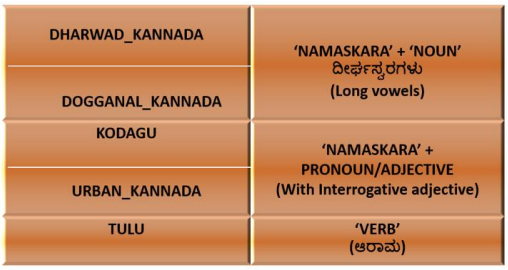


Fig. 9. Breakdown of wake word/sequence of words or phrase and their part od speech with their postion

As highlighted in the breaking down of words/phrases into linguistic segment it is concluded that speech with dialects appearing in Dharwad also in Dogganal dialects are similar yet they are not that identical. With respect to the order of their position of words present in terms of linguistic segment, RNN-LSTM framework misclassifies /makes the wrong forecast with labels [0,1] and [2,4].

It is evident from comparing the predicted labels that Conv1d works superior in comparison to the RNN-LSTM_framework.

Table shows training set prediction labels which are incorrectly estimated by RNN(-LSTM) _framework.

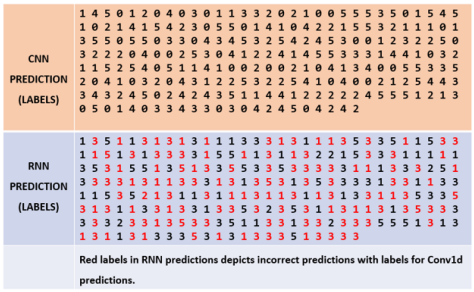


Fig. 10. Comparative evaluation of predicted lables

Conv1d model's overall implementation surpasses RNNs when it meets the objective, yielding significant outcomes.

Loss and accuracy check in metrics gives the trainable precision of models.

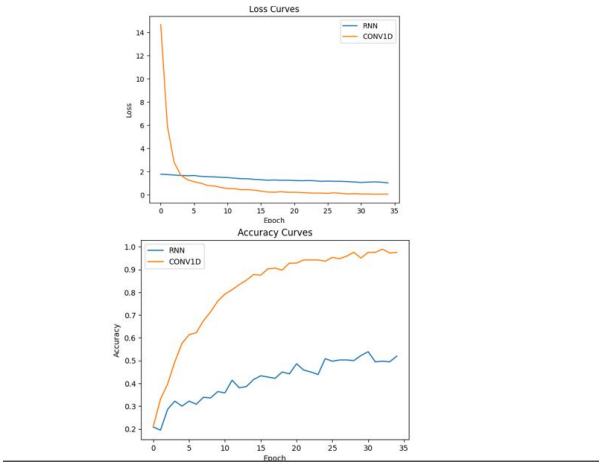


Fig. 11. Exhibits the loss and accuracy measures

5. Conclusion

With progress in AI and Data Science Dialect Identification assignment is accelerating in terms of different implementation in the field of NLP and automatic speech recognition tasks. This research paper focuses on the comparative analysis which is based on choosing the right model simple WWD system which is built is developed on major dialects within Karnataka. 1D CNN works effectively over the short-word sequences. Design architecture is developed to result in less error while predictions. Results shows that the concept of identifying correct dialect is purely relied on order of the words and not with sequence. Future work can be worked in field of TensorFlow_Quantization APIs which are more flexible in model deployment. The system should more focus on functionalities in Natural Language understanding in terms of extracting precise features from the Audio file.

References

[1] H. C. Das and U. Bhattacharjee, "Assamese Dialect Identification using," in IEEE World Conference on Applied Intelligence and Computing (AIC, 2022.

[2] Kumar, Rajath et al. “On Convolutional LSTM Modeling for Joint Wake-Word Detection and Text Dependent Speaker Verification.” Interspeech (2018)..

[3] H. Wang, M. Cheng, Q. Fu and M. Li, "The Dku Post-Challenge Audio-Visual Wake Word Spotting System," arXiv, 4 March 2023.

[4] Y. Tian, H. Yao, M. Cai, Y. Liu and Z. Ma, "Improving RNN Transducer Modeling for Small-Footprint Keyword Spotting," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 2021, pp. 5624-5628.

[5] J. Lee, K. Kim and M. Chung, "Korean Dialect Identification Based on Intonation Modeling," in 24th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA),, Singapore, 2021.

[6] Lokitha, Iswarya, Archana and A. Kumar, "Smart Voice Assistance for Speech disabled and Paralyzed People," in International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2022.

[7] Y. Wang, H. Lv, D. Povey, L. Xie and S. Khudanpur, "Wake Word Detection with Alignment-Free Lattice-Free MMI," INTERSPEECH 2020, 25-29 October 2020.

[8] T.-H. Tsai and P.-C. Hao, "Customized Wake-Up Word with Key Word Spotting using Convolutional Neural Network," in IEEE, 2019.

[9] S. Sarkar, B. Kumar and S. Kumar, "Mobile Applications for Indian Agriculture and Allied Sector:An Extended Arm for Farmers," International Journal of Current Microbiology and Applied Sciences, vol. 10, no. 3, 2021.

[10] M. Tzudir, S. Baghel, P. Sarmah and S. R. M. Prasanna, "Analyzing RMFCC Feature for Dialect Identification in Ao, an Under-Resourced Language," 2022.

[11] N. C. Diaz, N. Sasaki, T. W, Tsusaka and S. Szabo, "Factors affecting farmers’ willingness to adopt a mobile app in the marketing of bamboo products," Science Direct, vol. 11, 2021.

[12] R. K. Raman, D. K. Singh, U. Kumar and S. Sarkar, "Agricultural Mobile Apps for Transformation of Indian Farming," ReserachGate, vol. 07, no. 04, April 2021.

[13] S. G. Mane and K. R.V, "Design and Development of Mobile App for Farmers," International Journal of Trend in Scientific Research and Development (IJTSRD), pp. 179-182, 2019.

[14] R. Kumar, "Farmers’ Use of the Mobile Phone for Accessing Agricultural Information in Haryana: An Analytical Study," Open Information Science, 7 April 2023.

[15] K. D. M, and S. K. R. M, "FARMER’S ASSISTANT using AI Voice Bot," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), pp. 527-531, 2021.

[16] Z. Dan, Y. Zhao, X. Bi and Q. Ji, "Multi-Task Transformer with Adaptive Cross-Entropy Loss for Multi-Dialect Speech Recognition," MDPI, 8 OCTOBER 2022.

[17] R. Z. Qiuchen Yu, "Wake Word Detection Model Based on Res2Net," JOURNAL OF LATEX CLASS FILES, vol. 10, no. 10, 30 September 2022.

[18] Y. Wang, H. Lv, D. Povey, L. X. and S. Khudanpur, "WAKE WORD DETECTION WITH STREAMING TRANSFORMERS," in IEEE, Toronto, Canada, 2021.

[19] D. Landmann, C. Lagerkvist and V. Otter, "Determinants of Small Scale Farmers’ Intention to Use Smartphones for Generating Agricultural Knowledge in Developing Countries: Evidence from Rural India," The European Journal of Development Research, 10 August 2020.

[20] M.L.Dhore and M. Dhakate, "Insurance Value Chain Chatbot for Farmers," in ResearchGate, 2022.

[21] C. Li, L. Zhu, S. Xu, P. Gao and B. Xu, "Recurrent Neural Network Based Small-footprint Wake-up-word Speech Recognition System with a Score Calibration Method," 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 2018, pp. 3222-3227.

[22] V. Ribeiro, Y. Huang, Y. Shangguan, Z. Yang, L. Wan and M. Sun, "Handling the Alignment for Wake Word Detection:A Comparison Between Alignment-Based, Alignment-Free and Hybrid Approaches," in Accepted to Interspeech 2023, 2023.

[23] C. R. Kinkar and Y. K. Jain, "AN OVERVIEW OF MODERN ERA SPEECH RECOGNITION MODEL," International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 9, September 2021.

[24] T.Cynthia and C. Newton, "Voice Based Answering Technique for Farmers in Mobile Cloud Computing," International Journal of Scientific Research in Computer Science Applications and Management Studies, vol. 7, no. 3, 13 JULY 2020.

[25] D. Rostami and Y. Shekofteh, "A Persian Wake Word

Detection System Based on the Fine Tuning of A Universal Phone Decoder and Levenshtein Distance," 2023 9th International Conference on Web Research (ICWR), Tehran, Iran, Islamic Republic of, 2023, pp. 35-40.

[26] M. S. R. M. C. P. P. D. N. Arnav Kundu, "HEiMDaL: Highly Efficient Method for Detection and Localization of wake-words," in Audio and Speech Processing-ICASSP, 2023.