Sentiment classification on Tamil and Telugu text using RNNs and Transformers

D. Sumathi

*Associate Professor*

*School of Computer Science & Engineering*

*VIT-AP University*

Amaravati, India

[Sumathi.research28@gmail.com](mailto:Sumathi.research28@gmail.com)

H. Subramani

*B. Tech Final Year*

*School of Computer Science &*

*Engineering*

*VIT-AP University*

Amaravati, India

subramani.18bcd7181@vitap.ac.in

B. Gowtham

*B. Tech Final Year*

*School of Computer Science & Engineering*

*VIT-AP University*

Amaravati, India

[gowtham.18bce7143@vitap.ac.in](mailto:gowtham.18bce7143@vitap.ac.in)

K. Naveen

*B. Tech Final Year*

*School of Computer Science & Engineering*

*VIT-AP University*

Amaravati, India

[naveen.18bce7323@vitap.ac.in](mailto:naveen.18bce7323@vitap.ac.in)

***Abstract*—People use social media to express their opinion, thoughts, and views about a target. Sentiment analysis is found to be a common application that prevails currently and is adopted by all business and social domains. Through this, the business could be progressed and the profit could be raised marginally. Yet, people have the habit of expressing their views in the form of figurative language in which the precise meaning of the word should not be considered and as an alternative, the contrary analysis of that view is considered. This is found to be a difficult task since it might not be accounted for either positive or negative. The performance of the sentiment analysis tasks could be improved by considering these kinds of sarcastic sentences. Several works have been done on forming a model and training the model to detect sarcasm in several languages like Hindi, English, Telugu, etc. Keeping these considerations, a model has been developed which takes the input in the form of bilingual text and does the training to detect the presence of sentiment in the text. The model that has been developed works for the bilingual text and it identifies the existence of sentiment Experimental results show that the best model has an overall accuracy of 81% on the test set.**

***Keywords— Sentiment Analysis, Tamil and Telugu, BiLSTM, IndicBert, Feature Encoder, Bilingual, RNN.***

1. Introduction

Sentiment analysis plays a significant role in many industries since it contributes much towards the growth of the industry. It is identified as a method where the users could be able to express their views. Opinions and thoughts on a product in the social media. This facilitates the business people to improve their growth by considering the posts done by the people. A sentiment could be identified as either positive or negative. Apart from this, a few people might represent their views in an indirect way that results in persistent task of finding the sentiment. Here, identification of the sentiment is found to be a persistent approach since it might not be known to the people. The presence of such information becomes the key role in changing the polarity of the text [1]. To represent their views and opinions, people make use of blogs, social networks, forums and other approaches and this results in the huge volume of the data. Business people could access the views and use it for further decisions. Apart from describing a view as positive or negative, some people use a distinctive mode of expression which consists of words that might be an opposite of what one must want to say. Among the various challenges that are in NLP, detection of sentiment with respect to the mixed language is found to be a difficult task since word embeddings, sentiment lexicon and other factors are dependent on language. Normally, during the building of a model, feature engineering is the significant task since the proper group of features must be detected to make the model run successfully. Additionally, features are found to be specific to language and dataset dependent. Hence, building models for different languages is a challenging task. Normally, people often use the English language to post their opinions. A few people try to post the reviews in their regional language. Building a model for the languages separately is a time-consuming task and there are no word embedding for the regional language, it inspired us to construct a model for Limited studies have been carried out to address the detection of sentiment in bilingual languages. The objective of this study is to review the major classifiers that could be used to classify the sentiment and evaluate its performance. Datasets that are available in various languages have been scarped up from various sources. Datasets that are found in kaggle and other social media are found to be specific to the languages. Hence, one of the objectives of this work is to create a dataset which could be used for further research analysis. Models that have been used to the detection of sentiment is associated with the corresponding languages. The second objective is to develop a model which takes the input in the form of bilingual text. Here, the bilingual text comprises both Telugu and Tamil text. The novel approach in this work is that a single model could be developed for the bilingual text and it identifies whether the text is positive or negative. The next section provides you an overview of various research works that has been carried out for several languages. The third section describes about the proposed model and its architecture. There is a separate section to discuss in detail about the results and the performance of the model. Finally, the work concludes with the future scope.

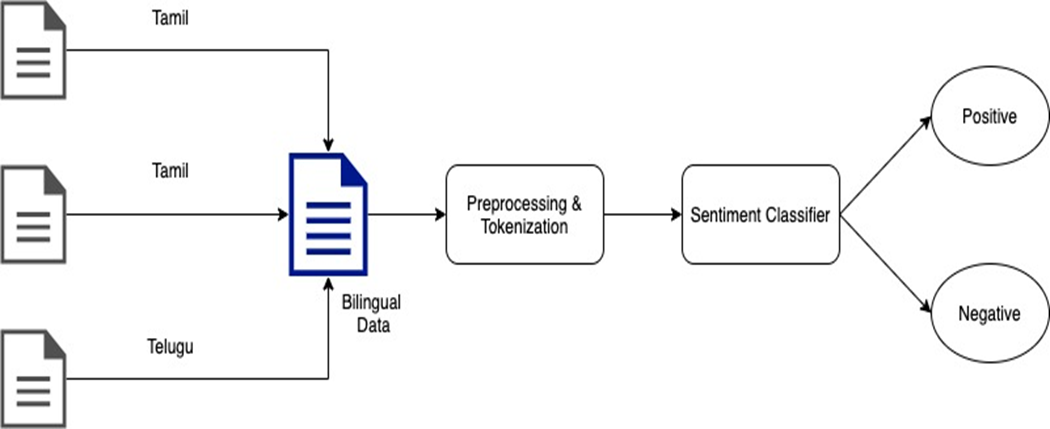
1. Literature Survey

This section refers to and provides an overview of the exploration in the field of sentiment examination for both Telugu and Tamil languages. Telugu SentiWordNet was proposed by the creators as "A Hybrid Learning Approach for Sentiment Classification in the Telugu Language"[2] which depends on the Telugu language. The supervised methodology which is dependent on the availability of training data is utilized to decide the number of audits of a Person written in the Telugu language. As the dataset is in an unstructured format with extra spaces, special characters, it is paramount to pre-process the dataset and are additionally dealt with. The precision of the framework is assessed by utilizing 20 percent of the dataset and their results showed the Accuracy of 85.04% in discovering sentiment related to Text. The authors proposed the Machine learning approach to classify the Sentiment value in Natural Language Processing in the Telugu language [3]. Language explicit difficulties including repetition of words and non-Telugu words for the most part found in Telugu text were dealt with and spider-monkey optimization is used (the bio-inspired model has its fitness based on its behaviour) to further develop the precision of the framework. The proposed framework to discover sentiment in the Telugu Language accomplished around 97% of the F-1 score. [3] Authors have utilized Supervised learning to deal with training a Deep Network on a small level of labelled data available. Word2Vec is used to replace words in vector format to understand the meaning. And then converted to the English language and used networks like recurrent neural networks and Naive Bayes. The main drawbacks are proper pre-processing of data, consistency of data, and data integrity is missing. They achieved an accuracy of 80.45 on the proposed framework. [4] Authors have extracted Tamil product reviews from YouTube, Twitter, Facebook, etc. We cannot assume the data integrity is maintained. All data pre-processing steps like stemming, lemmatization are performed and ensemble classification of Bagging (Tree bag, Random Forest) achieved 0.81 accuracy, and Stacking of (LDA, KNN, SVM) also obtained an accuracy of 0.81. [5] The authors used a raw corpus containing 7 lakh Telugu sentences to build Doc2Vec embeddings. Finally, a dataset was made of a random sampling of 1644 sentences from the raw corpus and labelled them manually. Sentences were directly sent to Doc2Vec without any pre-processing to get their corresponding embeddings. Later, these sentence embeddings were used to train Machine Learning algorithms (Naïve Bayes, Logistic Regression, Support Vector Machine, Multi-Layer Perceptron, Decision Trees, Random Forest, Adaboost Ensemble techniques) and evaluated models on 5 fold cross-validation. The F-score of 0.86 was achieved for binary classification of positive and negative classes and the F-score of 0.66 was achieved for ternary classification of positive, neutral, and negative classes. [6] Authors scraped data from social media and manually labelled it to build a dataset and named it as UJ\_Corpus dataset which contains Tamil sentences. After pre-processing the text, different embedding techniques like Bag of Words, TF, TF-IDF, Word2Vec, fast Text were applied. Later these sentences are fed into Lexicon-based classification algorithms, Machine Learning Algorithms like Linear SVM, Enhanced Gradient Boosting, Random Forest, Neural Network, K-Nearest Neighbours, logistic Regression and Multinomial Naïve Bayes and Hybrid Lexicon ML algorithm. The highest accuracy score of 0.79 was achieved on the test dataset of UJ\_corpus with a combination of fast Text and EGB classifier. This model further evaluated Tamil tweets of the SAIL dataset and achieved an accuracy score of 0.74. In another instance, the authors collected Tamil text from reviews, forums, blogs, discussions, etc., and manually labelled the dataset. Later, the word frequency of each word in a whole dataset is used to represent the sentence. Further, those sentence representations have been used to train Machine Learning Algorithms. Finally, the highest accuracy of 65 percent has been achieved [14]. One more includes the Tamil movie reviews dataset which was manually annotated, and later, features are extracted from sentences using TamilSentiWordNet. Further, those features are used to feed the Machine Learning Algorithms like SVM, Decision Tree, Naïve Bayes, etc., Finally, SVM performed better than other ML Algorithms and with an accuracy score of 0.75[15].

1. Methodology

In this research, we have created a custom dataset encompassing both Tamil and Telugu sentences and have applied various deep learning models to do sentiment classification. At first, we have taken datasets that are already available for sentiment classification in these two languages. For Tamil, we used Tamil Binary Classification 1K tweets Labels [7], ACTSEA [8] and for Telugu, we used ACTSA [9]. We have taken data of two labels, positive and negative from these datasets and have combined both Tamil and Telugu corpus to make a bilingual dataset. We wanted to create large enough bilingual data and so we decided to use a combination of datasets. Then we pre-process the data to remove noise such as tags, English words, URLs, etc., and tokenize them to feed it into different models which is explained in section 4.2. Finally, we train various deep learning models to do sentiment classification to predict whether each sentence belongs to a positive or negative label and calculate the accuracy for each of the models as detailed in section 4.3.

An overview of the proposed method is illustrated in Figure 1

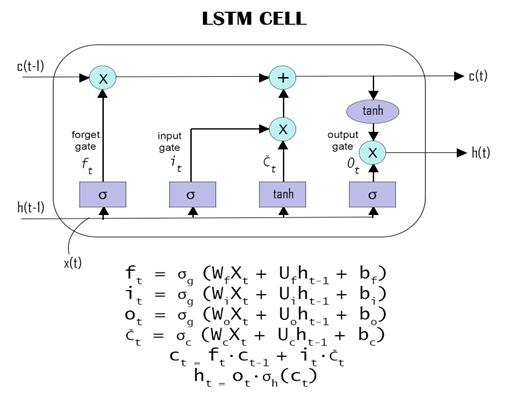


**Figure 1.** Architecture diagram of the proposed methodology

1. Experiment

In this section, we describe the data preparation, pre-processing techniques applied to the dataset and the different deep learning models used to perform sentiment classification.

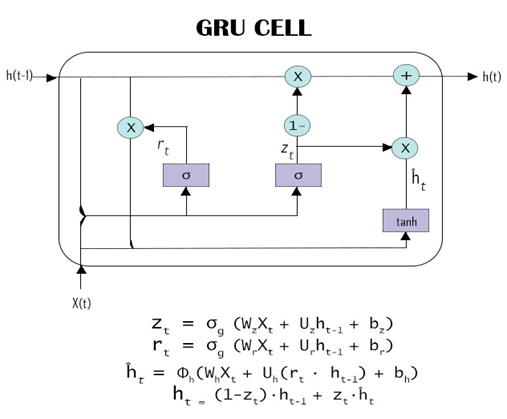
1. ***Data Preparation****:* Our goal was to create a bilingual language model that can classify positive and negative sentences from both Tamil and Telugu languages. To do this, we have used publicly available sentiment classification datasets [7][8][9] for these two languages. Apart from positive and negative labels, these datasets contained various other labels like happy, sad, anger, fear. Here we consider all the sentences labelled under happy as positive and sad, anger, fear, and negative. Finally, we have combined all these sentences and corresponding labels into one single dataset to do further pre-processing.
2. ***Data Pre-processing****:* In this research, we wanted to train a model that uses only sentences from Tamil and Telugu languages to classify positive and negative labels. The dataset comprises the sentences made of only Tamil and Telugu languages. This dataset includes URLs, handlers, emojis, special characters, and numbers. Duplicates have been removed from the dataset. A vocabulary of words from our dataset has been constructed. During this task, So, we pre-processed the dataset which contains only sentences from Tamil and Telugu languages without words from any other languages, URLs, handlers, emojis, special characters, and numbers. We have also removed duplicates from the dataset. We then went on to build a vocabulary of words from our dataset with a minimum word frequency of 3 and sentences are padded with zeros in each mini-batch to get a sentence of fixed length which is equal to the length of the sentence with maximum words in the mini-batch. An unknown token is used to represent a word whose frequency in the dataset is less than 3.
3. ***Models****:* Bidirectional LSTM + CNN Long Short-Term Memory [10] or LSTMs for short are an evolution of Recurrent neural networks, which solves the long-term dependency or vanishing gradient problem in RNNs. At any timestamp t, an LSTM unit has two states, a hidden state ht,and cell state ct. It also uses a gating mechanism in which different gates control the flow of information through the network. A single LSTM unit is depicted in figure 2. Firstly, a forget gate ft is computed using the output from the previous cell ht-1 and the present state input xt, which helps to decide which information from the cell state ct has to be thrown away. Then we have the input gate it which takes care of the addition of new information and modification into the cell state. The cell state ct then contains the information which flows through the network and the gates help them to make updates to hold relevant information at each timestamp. Here we use a Bidirectional LSTM which has two hidden layers. A BiLSTM can use information from both the past and future. It processes the input data in both forward and backward directions. The output at any timestamp is obtained by combining the results of both forward and backward layers. Once the data is pre-processed, it is then fed into the trainable Embedding layer which creates a feature vector of dimension 64 for each word in the sentence. Here we use a sequence length of 64 and a batch size of N=8 per mini-batch. The output of the embedding layer is then passed on to the BiLSTM which has a hidden unit's size of 128 and 2 layers. The final output of the BiLSTM after applying a softmax activation is of shape = (N, sequence\_length, hidden\_size) where N is the batch size. This output is then passed on to a 1D CNN with kernel size=1 which moves as a sliding window over the output sequence of BiLSTM. The resultant output is then of shape = (N, sequence\_length). This output is then passed on to a dense layer with 64 units and then a final output layer with 2 units and a softmax activation to predict the probabilities of positive and negative labels. We have used a categorical cross-entropy loss and adamax optimizer with a learning rate initially set to 3\*10-4. The learning rate was reduced by 0.5 after every two epochs to have a smooth convergence of loss function.



**Figure 2**. A Single LSTM unit

1. *Bidirectional GRU*

Gated Recurrent units [11] or GRUs are a simpler alternative to LSTMs which have fewer parameters to train than LSTMs and are thus computationally less expensive to train. Unlike an LSTM, GRUs contain a candidate hidden state and only have two gates which are the update gate and reset gate. A GRU unit is depicted in figure 3. The reset gate rt is responsible for short-term memory where it controls how much of the previous state is used to compute the new state whereas the update gate controls long-term memory which determines how much of the previous state has to be remembered. They are both computed using the present state input Xt and the hidden state of the previous timestamp ht-1. A sigmoid activation function is applied at the end to each of them. Then the candidate hidden state is computed using the current input Xt, reset gate rt, and the previously hidden state ht-1 and tanh non-linearity is added at the end. This is then used along with the update gate zt and previous hidden state to find the current hidden state at t. Also, we are using bidirectional GRUs like BiLSTM which has both forward and backward blocks. Here also, we have a trainable embedding later in which the sentences are fed in initially with a batch size of 8. The embedding layer computes a feature vector of dimension=34 for each word and here the weights of the embedding layer are initialized with a uniform distribution of range 0.5 to 0.5. The output of this is then fed to the bidirectional GRUs with a hidden unit size of 16 and 2 layers. The hidden state of the last layer of GRU which is of shape = (N,2, hidden size) is resized and then passed on to a final dense layer. This final layer has a softmax activation and two units which compute the probability of positive and negative labels. Again, we use the same categorical cross-entropy but the optimizer we used here was Adam with a learning rate of 3\*10-5 which is also reduced by 0.5 after every 2 epochs.



**Figure 3**. A Single GRU unit

1. *BiLSTM with Attention Mechanism*

In this method, the output of each BiLSTM layer is passed on to an attention layer that uses attention weights which tells the model how much attention has should be paid for each word in a sentence. The attention layer thus takes in a context c as input which is the sum of attention weights multiplied by the activation of the BiLSTM layer. The output of this is passed on to the final dense layer which predicts the label. The number of hidden units used here for the BiLSTM is 128 with an input sequence length of 50. We used the same loss function and optimizer as in the case of BiLSTM+CNN explained above with the same learning rate and learning rate decay.

1. *IndicBERT*

The final model we experimented with was IndicBERT [12], which is a pre-trained ALBERT [13] based model. This model was pre-trained on 12 Indian languages and can perform various NLP tasks on these languages. ALBERT is an auto-encoding transformer model which is light and also good at understanding the input sentences which makes it better suited for any word or sentence classification tasks. We have used the hugging face transformers library to load the tokenizer and pre-trained IndicBERT model and then used it on our data to perform the sentiment classification task.

1. Results

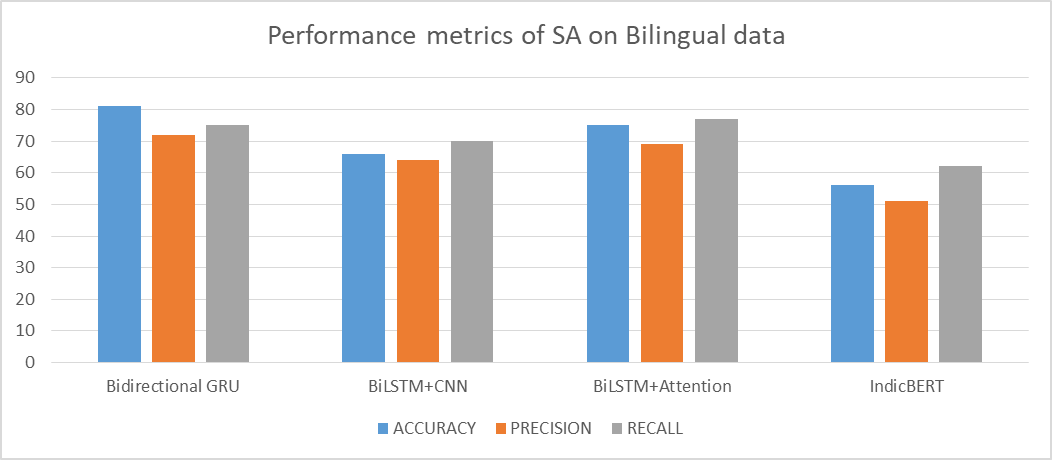
The dataset we created to train our models contained around 2000 Tamil and Telugu sentences and their corresponding sentiment as labels. We have split the dataset into an 80:20 ratio for training and testing. The results in Table 1 show the accuracy, precision, and recall obtained by our models on the test data. Accuracy is nothing but the number of correct predictions over the entire observation. Precision measures the rate of false positives while recall measures false negatives against true positives. The formulas for all three are given below.

Accuracy=

(1)

Precision = (2)

Recall = (3)



**Figure 4:** Performance comparison of various models

From the table, we can infer that Bidirectional GRUs have the highest accuracy while IndicBERT has the lowest accuracy rate. The other two models have given average results. GRUs have fewer parameters compared to LSTMs and IndicBERT and are thus faster to train. GRUs need fewer data to generalize and since we do not have a huge language corpus to train, it is understandable that Bidirectional GRU has performed better than BiLSTM. Also, IndicBERT was benchmarked with an accuracy of 61% on ACTSA sentiment analysis dataset, which is one of the datasets we used to build our bilingual dataset. But IndicBERT gives lower accuracy than that in our results which shows that it is not generalized well on our bilingual dataset. Apart from that, using an attention mechanism to BiLSTM seems to give better results because of its ability to pay attention to nearby contexts.

TABLE I. Performance Of Each Model On The Test Set

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ACCURACY** | **PRECISION** | **RECALL** |
| **Bidirectional GRU** | 81 | 72 | 75 |
| **BiLSTM+CNN** | 66 | 64 | 70 |
| **BiLSTM+Attention** | 75 | 69 | 77 |
| **IndicBERT** | 56 | 51 | 62 |

1. Conclusion

Many real-world applications, such as review analysis and recommendation systems, benefit from sentiment analysis. It becomes more challenging, when there is a lot of noise in the data and when it's gathered through social media. India a multilingual country populated by multilingual individuals; non-native English speakers who communicate in more than one language. People would like to post tweets in their native languages. Detecting analysis from the regional languages is more complex as there is a requirement of corpus of the corresponding language. The work has focused on various models and at the end, IndicBert has been deployed to detect the sentiment analysis. Accuracy obtained through Bidirectional GRU is high when compared with the other models.

References

[1] Rojas-Barahona LM (2016) Deep learning for sentiment analysis. Lang Linguist Compass 10(12):701–719.

[2] S. Tammina, "A Hybrid Learning approach for Sentiment Classification in Telugu Language," 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), 2020, pp. 1-6, doi: 10.1109/AISP48273.2020.9073109.

[3] Palli Suryachandra, P. Venkata and Subba Reddy, 2020. Machine Learning Approach to Classify the Sentiment Value of Natural Language Processing in Telugu Data. Journal of Engineering and Applied Sciences, 15: 3593-3598.

[4] Bharti, S., Naidu, R., & Babu, K.S. (2021). Dynamic SentiPhraseNet to Support Sentiment Analysis in Telugu.

[5] Mukku, S.S., Choudhary, N., & Mamidi, R. (2016). Enhanced Sentiment Classification of Telugu Text using ML Techniques. SAAIP@IJCAI.

[6] S. Thavareesan and S. Mahesan, "Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation," 2019 14th Conference on Industrial and Information Systems (ICIIS), 2019, pp. 320-325, doi: 10.1109/ICIIS47346.2019.9063341.

[7] Kracekumar, “Tamil Binary Classification 1K tweets Labels V1.” Kaggle, 2020, doi: 10.34740/KAGGLE/DSV/1226691.

[8] Jenarthanan, R., Senarath, Y., & Thayasivam, U. (2019). ACTSEA: Annotated Corpus for Tamil & Sinhala Emotion Analysis. 2019 Moratuwa Engineering Research Conference (MERCon).

[9] Mukku, Sandeep Sricharan and R. Mamidi. “ACTSA: Annotated Corpus for Telugu Sentiment Analysis.” (2017). Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems ([EMNLP-2017 2017](https://generalizablenlp.weebly.com/)), pp 54-58.

[10] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. Neural Comput 1997; 9 (8): 1735–1780. doi: <https://doi.org/10.1162/neco.1997.9.8.1735>

[11] Chung, Junyoung, Çaglar Gülçehre, Kyunghyun Cho and Yoshua Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” ArXiv abs/1412.3555 (2014): n. pag.

[12] Kakwani, D., Kunchukuttan, A., Golla, S., N.C. G., Bhattacharyya, A., Khapra, M.M. and Kumar, P., 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. Accepted by Findings of EMNLP 2020

[13] Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma and Radu Soricut. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.” ArXiv abs/1909.11942 (2020): n. pag.

[14] S. J. Arunselvan, M. Kumar, A., and Soman, K. P., “Sentiment analysis of tamil movie reviews via feature frequency count”, International Journal of Applied Engineering Research, vol. 10, no. 20, pp. 17934-17939, 2015.

[15] Predicting the Sentimental Reviews in Tamil Movie using Machine Learning Algorithms**,** Shriya Se\*, R. Vinayakumar, M. Anand Kumar and K. P. Soman DOI:[10.17485/ijst/2016/v9i45/106482](https://dx.doi.org/10.17485/ijst/2016/v9i45/106482).

.