Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 18, no. 8, August 2022, pp. 545-551
DOI: 10.23940/ijpe.22.08.p2.545551

RNN LSTM-based Deep Hybrid Learning Model for Text Classification using Machine Learning Variant XGBoost

Sandhya Alagarsamy[a,*] and Visumathi James[b]

[a]*Department of Computer Science Engineering, Sathyabama Institute of Science and Technology, Chennai, 600100, India*
[b]*Department of Computer Science Engineering, Veltech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai, 600085, India*

Abstract

Text classification is an emerging area in Natural Language Processing (NLP). On the other hand, traditional text classification methods need to be improved due to the complexity and semantic nature in text. In this paper, we build a hybrid deep learning model using Deep Learning (DL) and Machine learning (ML) models. This work combines two traditional neural networks namely Long Short-Term Memory (LSTM) and Recurrent Neural Network (RNN) to extract the features from the text document. LSTM preserves the historical information for text sequences and extracts the features using the RNN structure. The extracted features are used to run on machine learning classification algorithms like AdaBoost and XGBoost to perform the final prediction. Thereby the proposed Deep Hybrid Model eliminates the fully connected classification layers from a typical Deep Learning model. The performance of proposed model is measured with other models and the results show that the deep hybrid model provides about 12% increased results in terms of accuracy in text classification.

Keywords: deep learning; machine learning; text classification; hybrid model; RNN; LSTM

(Submitted on June 3, 2022; Revised on June 29, 2022; Accepted on July 12, 2022)

1. Introduction

Text classification performs the classification of input text document. The source of data can be from any social media such as a Twitter database, Facebook or any website reviews. These data sources are classified into a positive and negative category. This is also known as sentiment analysis or classification. The traditional neural network (NN) techniques are framed to process the sentiment information and classify the text. However, the model maps the terms incorrectly with neighbour vectors and ends up in improper text classification for a large text corpus.

Many researchers were impressed by the abundant growth and development in the field of deep learning and started to explore various applications of text classification. In addition, various research groups are operating to upgrade this field from traditional methods to advanced deep neural network (DNN) methods. However, most of the work restricts them to use only one deep learning model to perform the classification task. Recently, as the number of internet users is increased at a rapid level, the text data growth rate is also increased. Hence there is a need for an advanced deep leaning model to handle the huge data and to perform accurate classifications. As a result, the hybrid learning model is evolving to handle the text classification efficiently under sentiment analysis.

Deep neural networks provide accurate results in the domain of nature language processing. Moreover, they provide excellent learning capabilities. Neural network models, like Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) are mainly used for text classifications [1]. The neural networks perform better than traditional learning methods. A typical CNN model fails in dealing with the long-time series due to the strong dependence between the contexts in text classification. Recurrent Neural Networks (RNNs) can handle the long-time series in an efficient way.

* Corresponding author.
E-mail address: sandhyalagar@gmail.com

The objective of this work is to propose a deep hybrid learning model for automatic classification of a large text corpus based on the sentiment present in it. The proposed hybrid model implementation is based on a RNN and LSTM deep learning algorithm along with a XGBoost machine learning variant. This hybrid model improves the accuracy in text classification. The Twitter airline sentiment database is accessed from kaggle to test the deep hybrid model. The experimental results shows that our hybrid model is efficient in enhancing the text classification accuracy and yields improved grouping results.

Deep Hybrid Learning (DHL) approach is a fusion method to merge the conventional or classical deep learning model with other learning algorithms. Based on the fusion mechanism, there exists two forms of DHL namely Early and Late Fusion. In Early Fusion, the fusion process will occur before feature extraction (FE). In Late Fusion, the fusion process will happen after feature extraction. The fusion can be either with another Deep Learning variant or Machine Learning variant. The DHL works efficiently for unstructured data and it does not require any manual Feature Extraction. The proposed work combines the advantages of RNN and LSTM for feature extraction and XGBoost to map the extracted features for final prediction. Finally a deep hybrid text classification model based on RNN-LSTM with Late Fusion using Machine Learning variant is proposed.

The structure of the paper is as follows: Section 2 discusses the popular methods and techniques followed in text classification. The most challenging works using machine and deep learning models are also listed. In section 3, the DHL framework is presented and various modules are discussed in detail with their internal operations. Section 4 provides detailed information on dataset, pre-processing steps and specification on kernel size. The results of the proposed model are discussed in section 5 and final findings are concluded in Section 6.

2. Related Works

Various methods and algorithms were there for processing the text document. The traditional methods produce results mostly for the smaller datasets [2,3]. When the semantic nature increases with text documents, the advanced and hybrid approach come into existence. Moreover the traditional embedding algorithm cannot do feature extraction for text documents in an efficient manner [4,5]. Therefore word embedding algorithms were used to process every word in the text document and then convert them into vector representation. While using deep learning models, the accuracy will improve compared to other machine learning models [6]. The CNN model has a few drawbacks like being unable to recollect historical data, time consuming in processing complex document, and having only minimal features being extracted.

To overcome the time and accuracy constraints, the layers of the models will be increased [7]. This leads to a complex network system and additional expense. The use of RNN produces better feature extraction than CNN [8,9]. LSTM with its gates available keeps historical data for efficient processing. Few Machine learning algorithms always provide accurate results through their prediction. Various flavours of hybrid models exist for efficient feature extraction and text processing. [10] implemented a Deep hybrid model to predict customer re-purchase behaviour and achieved maximum accuracy. [11] implemented a Deep hybrid model and proved that the features extracted this way had better representation. [12] proved in his work that the deep hybrid model works efficiently with 2Dimensional and 3Dimensional representation.

Various techniques in text classification originated from the simple term based methods to the advanced deep neural network methods [13,14]. The classification techniques based on the lexicon model are simple as they work based on the values assigned for each words [15]. However, the lexicon model provides less accuracy [16,17] when more similar words occur in the text corpus. After many experimental analyses [18], many researchers concluded that the models based on a machine learning approach provide higher accuracy than the lexicon approach. As the domain of NLP grows rapidly with neural network models, the major step forward is implementation of vector notations for the words [19]. Severyn [20] trained millions of Twitter data using a Word2Vec embedding algorithm. Mikolov [21] used a Continuous Bag Of Words (CBOW) method to learn the representation of words. Lauren [22] used a skip-gram model to create word embeddings for the input text. Recently the use of CNN and RNN with word embedding is done for efficient text classification of English language.

3. Deep Hybrid Model

The Deep Hybrid Model is the fusion of Deep Learning (DL) and Machine Learning. (ML). The fusion can be between deep learning and machine Learning or with deep learning and a modified model of deep learning [23]. Various flavours of Deep Hybrid Model can be defined and listed in Table 1. If the fusion happens before feature extraction, it is termed as early fusion. In late fusion, the model fusion occurs after feature extraction.

Table 1. Flavours of Deep Hybrid Learning

Type of fusion	Hybrid Structure	Variant Type
Late Fusion	DL+ Classical ML	Machine Learning
	DL+ Modified DL structure	Deep Learning
Early Fusion	DL and ML + DL / ML	Machine Learning
	DL1 and DL2 + Fully connected layer	Deep Learning

The deep hybrid model framework based on RNN-LSTM with ML variant is given in Figure 1. The three main parts of the model are as follows:

1) Word2vec embedding algorithm to process the words in the text.
2) RNN-LSTM model to extract the features.
3) XGBoost Machine Learning Variant for final prediction.

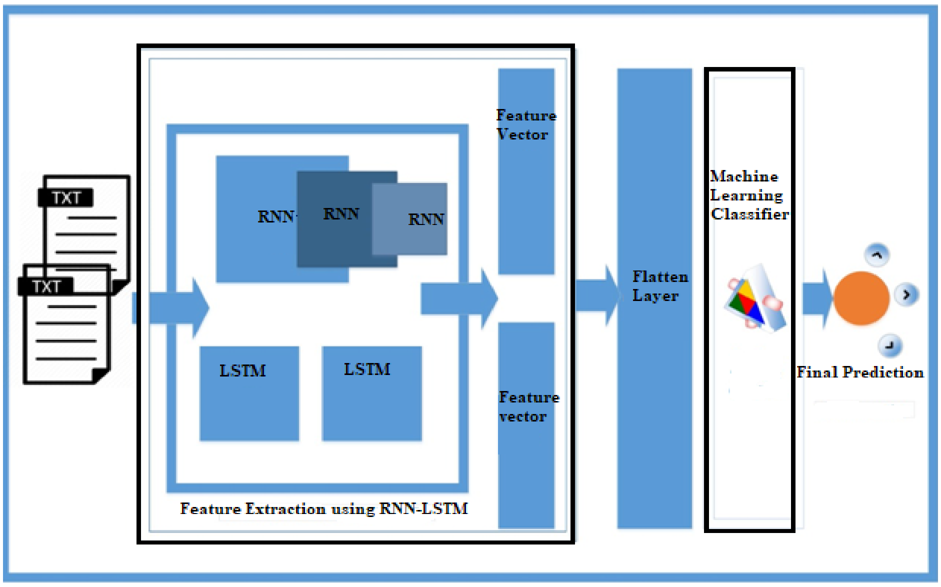

Figure 1. Architecture - RNN-LSTM + ML Variant

3.1. Word Embedding - Word2vec

While working with NLP, a word or phrase is used as the basic unit. Also, text is represented as vectors using various algorithms. One-hot representation algorithm is commonly used for word representation. While using the one-hot method, the similarity among the words cannot be represented. However, implementing one-hot algorithm in deep learning leads to problems like dimensional disaster due to its high dimension of input documents. Also, the degree of association between the words is not represented properly. The problems occurred by the one-hot representation method are replaced by word embedding algorithms. Various word embedding algorithms available are Embedding layer, Glove and Word2vec. In this work we used the three-layered neural network structure named word2vec embedding algorithm to represent the words in distributed manner. The main activity of word2vec is words to vector conversion.

Word2vec involve the Skip-Gram model and the CBOW model that has an input layer, a projection layer, and an output layer. The current word is used to predict the context in the Skip-Gram model, whereas the context is used in current word prediction in the CBOW model. The word2vec model performs the words into vector representation. Equation 1 is related to Input text document processing as in Fig 1. While processing the text document, each word in the document is represented in vector space. The vector representation consists of multiple words. Now, the tth word of the text is denoted as $a_t \in R^d$, where d is the word embedding vector dimension. Let the text length be T, so text input is denoted as in Equation 1.

$$A = [x_1, x_2, \dots x_T] \in R^{T*d} \quad (1)$$

In the randomly initialized word embedding, the vector needs to be updated while training using a neural network model. The dataset (Tweets.csv) used in this work is obtained from kaggle. The kaggle location is https://www.kaggle.com/crowdflower/twitter-airline-sentiment/Tweets.csv. This dataset has the customer tweets of particular airline. Various fields available in the dataset are tweet_id, tweet_text, sentiment_value, airline_id and so on. The input text document has two fields namely tweet_text and sentiment_value. After pre-processing, word2vec word embedding algorithm processes the input text document and converts the words present in the document into the vector representation. Semantically related words are represented very closely in the vector space and then passed to neural network based on RNN-LSTM.

3.2. RNN-LSTM

The neural network used in this hybrid model is RNN and LSTM. The interacting and interconnected network of neurons is known as RNN. Here the neurons were connected by the weights. This kind of network structure is suitable for varying size inputs and time series problems. In RNN, the information travels in a bidirectional manner which maintains the connection among the long sequence of data due to its internal memory. RNN is widely used to process the sequences of different length. As RNN is similar to feed-forward NN, it processes a huge volume of past data obtained from long text sequences.

LSTM network is an extension of RNN and capable of doing FE. The semantic inspection of long text sequences is also performed and rectifies the vanishing gradient problem. LSTM stores the old information in prolonged sequences effectively using the three control gates and memory cell. This structure prevents the old information loss occurred during training. This network model performs actions like read, write and delete the data through three gates namely Input, Forget and Output gate. The Input gate blocks any updates that may occur. The Forget gate deactivates a neuron of less importance according to the learned weight and the Output gate of the neuron yields the output.

In the Proposed RNN-LSTM model, the multi-layer LSTM yields vector as output and the RNN model receives this vector as input. The RNN model is built over the LSTM model to further extract the property of the text input. The multi-layer LSTM extracts the feature from input sequence and produces the output as $\mathrm{S} = [\mathrm{s}_1, \mathrm{s}_2, .. \mathrm{s}_\mathrm{t}]^\mathrm{T}$ where s_t represents the feature vector of m-dimensional for the t th word of the text sequence. Also, the maximum vector distance equals to the nodes count of hidden layers present in LSTM. Count on steps for LSTM growth is denoted by “T“ and that equals to the length of text sequences.

The RNN receives matrix input represented as $\mathrm{S} \in \mathrm{R}^{\mathrm{m} * \mathrm{T}}$. The convolution filter used in RNN is represented as $\mathrm{F} \in \mathrm{R}^{\mathrm{i} * \mathrm{j}}$. The count on words in window is denoted as “i” and the word vector dimension is “j”. By using various filters with different parameters, maximum features are extracted. The extracted features are now passed to flatten layer where max pooling and activation functions are applied to obtain maximum probability.

In this Deep Hybrid model, RNN and LSTM were the deep neural networks used for extracting the features. After Feature Extraction, feature vectors were formed. The feature extraction part does not need any manual work to extract the desired features. The Deep Neural Network uses 2D convolution and max Pooling for efficient feature extraction. The layers count used in DNN depends on the Input complexity. Various parameters that support the feature extraction are Filter value, Kernel Size, and type of Activation Function. Now, the flattening process is performed using the flattened layer and feature vectors are formed. The next module will perform the final classification using the classical machine learning classifier.

3.3. Machine Learning Models

In this module suitable machine learning algorithms like Random Forest, Support Vector Machine, AdaBoost and XGBoost can be implemented to have the final prediction. In conventional machine learning models, the algorithm accuracy depends on few parameters like understanding and analysing the dataset and the ability to perform FE on the dataset.

The drawback in the final classification layer of a DL model being driven by fully connected neural network layers was it may result in over-fitting. This over-fitting scenario happens when we process with minimum data. Moreover, these models require an unnecessary usage of computational power and resources, which is not there in classical machine learning algorithms. To overcome this problem, a conventional machine learning algorithm is used for final prediction. Finally, this Machine Learning Classification algorithm is used to fit on the extracted features. The above mentioned issue can be rectified by combining Deep Learning and Machine Learning for obtaining the final prediction.

Since XGBoost performs quite well for structured data, the XGBoost machine learning variant is used in this work. Also, XGBoost provides consistent accuracy through its performance. The input dataset is split into train and test at the word embedding phase. The Train data is processed through the RNN-LSTM model and the Test data remains the same. Now, the model is built with the processed train data. Since it is a classification scenario, XGBClassifier is used in the model. Now, the train data is used to build the XGBoost model. The initial model M0 is defined to predict the target variable y, which will be associated with the residual. Now, new model g1 is fit to the residuals from the previous step. The upgraded version M1 is obtained by combining M0 and g1. M1 has less mean square error than M0. The model representation is given in Equation 2.

$$\mathrm{M}_1(\mathrm{x}) \ = \mathrm{M}_0(\mathrm{x}) + \mathrm{g}_1(\mathrm{x}) \tag{2}$$

The new model is built after residual g1. The process continues for multiple iterations until residuals have been minimized as much as possible. By this process, the model processes the test data and the final prediction is made. In this proposed work DL methods are used to generate features from the unstructured data and then classical ML approaches are used to build accurate classification models for the prediction. Thereby, using DHL we can acquire the benefits of both DL and ML in a more accurate manner with less expense. The machine learning algorithm used in this work is XGBoost for the prediction.

4. Experimental Analysis

To evaluate the model performance, the standard dataset is obtained from kaggle. The dataset (Tweets.csv) used in this work is obtained from kaggle. The kaggle location is https://www.kaggle.com/crowdflower/twitter-airline-sentiment/Tweets.csv. The dataset consists of customer reviews for a particular airline. The dataset consists of 14485 tweet comments. The data is pre-processed to remove stop words, punctuations and so on. After pre-processing, train and test data is split in the word embedding layer. Train data is processed by the RNN-LSTM model. The test data is reserved for testing using the machine learning model. Efficient feature extraction can be obtained from the pre-processed document. As per our proposed hybrid model, a few layers of RNN-LSTM will perform the feature extraction. Since we are processing a text document, word2vec embedding algorithm is used to perform word segmentation. Now each word in text is converted into a distinctive word index. The maximum text length is 80 and the minimum is 5. In order to facilitate the model performance, the model input length is fixed as 50. The sequences that are less than 50 are padded with zero and the sequence greater than 50 was truncated. The convolutional layer defined is 32 filters, kernel size 8, and ReLu Activation function. RNN used feedback loops for sequential processing and maintaining in memory. The output of this layer is passed to the LSTM layer with 128 neurons. Based on the probability function and loss function at the LSTM layer, the optimal features were extracted and passed to the Machine learning algorithm. The proposed hybrid model is evaluated with AdaBoost and XGBoost algorithm to yield the final prediction.

5. Results and Discussion

To evaluate the RNN-LSTM based hybrid model for text classification, the prediction accuracy is used to measure the proposed models performance. Accuracy is the most commonly used metric in text classification. The classification accuracy is measured as the percentage of the total number of text in the correct category. The validation and test accuracy obtained using the proposed model is evaluated and examined with other models. The comparison is listed in Table 2.

Table 2. Experimental Results of various Models

Model	Validation	Test
CNN	81.24%	83.22%
RNN	83%	87%
CNN-LSTM	82.77%	86.91%
RNN-LSTM	89.54%	91.34%
RNN-LSTM + XGBoost	92.77%	96.87%

From Table 2 we can infer that the hybrid model that used RNN-LSTM along with the ML algorithm XGBoost gave maximum accuracy when compared to other models listed. The proposed hybrid model is evaluated with both AdaBoost and XGBoost for performance evaluation metrics like Accuracy, Precision, Recall, F1 Score and AUC Score. The obtained performance metrics values were listed in Table 3.

Table 3. Performance Metrics Evaluation

Performance Metrics	AdaBoost	XGBoost
Accuracy	93.8 %	98.4%
Precision	0.8819	0.9404
Recall	0.8792	0.9348
F1 Score	0.8804	0.9370
AUC Score	0.8792	0.9348

From various performance metrics evaluated for ML algorithms AdaBoost and XGBoost, it is inferred that XGBoost provides more accuracy than AdaBoost while implemented in the Hybrid Model. Also, the hybrid model achieved maximum accuracy. The training and validation accuracy based on number of epoch is shown in Figure 2. Figure 3 shows the training loss and validation loss for each epoch.

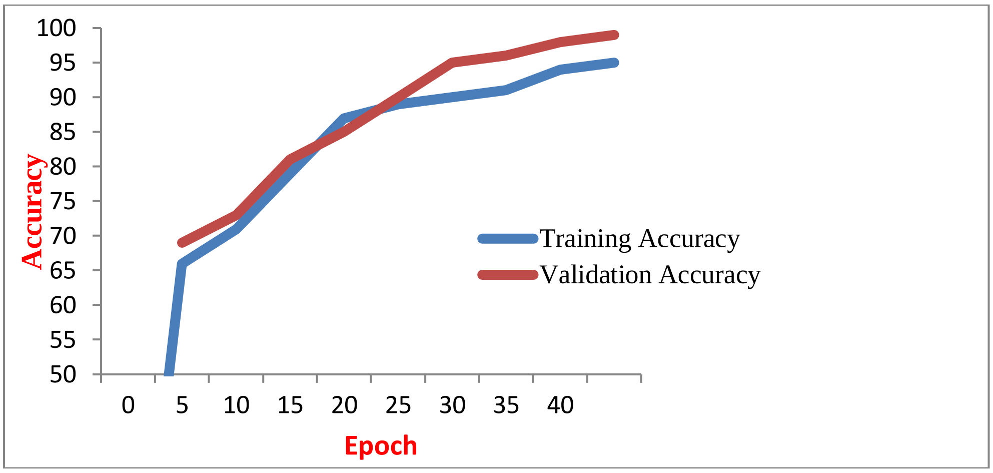

Figure 2. Hybrid Model - Accuracy Curves

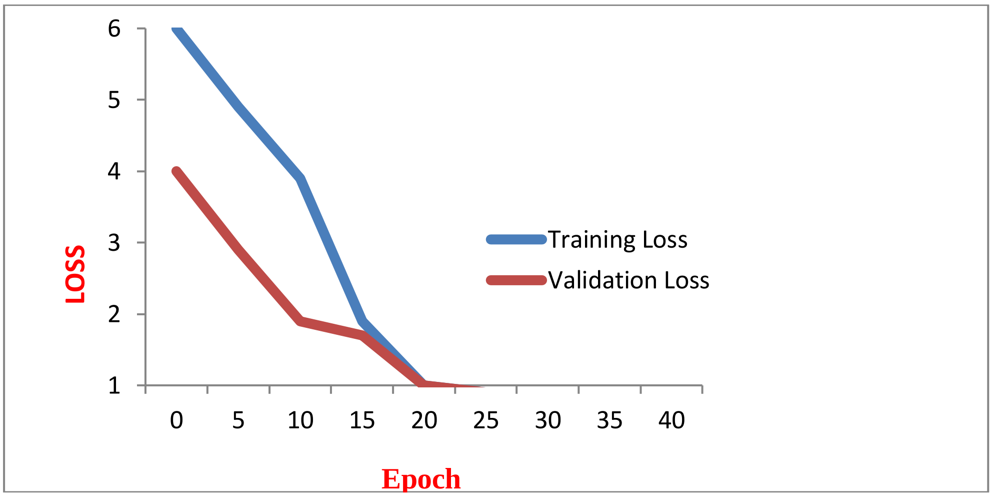

Figure 3. Hybrid Model - Loss Curves

From Figure 2 and Figure 3 we conclude that, as the number of epoch increases, the accuracy increases and the loss is reduced. Hence the hybrid model with XGBoost is suitable for efficient prediction of text data.

6. Conclusion and Future work

This work proposed a RNN-LSMT based hybrid model with a ML variant for text classification. The model uses the advantage of LSTM and preserves historical information in long text thereby resolving the vanishing gradient problem. On the other hand, RNN is used for FE of the text. The model uses Machine Learning variant XGBoost to process the extracted features and makes the final prediction. Since the model proposed in this paper used the ML variant, it eliminates the fully connected classification layers from a typical Deep Learning model. Hence the proposed model works with less cost and effectively improved the text classification accuracy. The experimental results prove the validity of the model. As the length and complexity of text documents increase dynamically, the complexity in text classification will be challenging. Therefore, the next level of model such as the Attention based model can be combined with a Deep Learning model to form an advanced hybrid model for accurate classification of long text sequences. Also, this work can be extended to incorporate the Wikipedia and Wordnet databases to this hybrid framework to achieve the accuracy in text classification for a large and dynamic text corpus with minimum epoch value.

References

1. Fan, Y., Lu, X., Li, D., and Liu, Y. Video-based emotion recognition using CNN-RNN and C3D hybrid networks, *ACM International Conference on Multimodal Interaction*. ACM, pp. 445-450, 2016.
2. Fan, Y. Chen, Y., Lin, K., and Huang, C. Using deep neural networks to evaluate the system reliability of manufacturing networks, *International Journal of Performability Engineering*, vol. 17, no. 7, pp. 600-608, 2021.
3. Mikolov, T., Karafiat, M., Burget, L., and Cernocky, J. Recurrent neural network based language model, The 11th *International Speech Communication Association*, Japan, pp. 235-241, 2010.
4. Rao, A. and Spasojevic, N. Actionable and political text classification using word embeddings and LSTM, *Computer Science,* ArXiv, abs/1607.02501, 2016.
5. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E. Hierarchical attention networks for document classification, In *Proceedings of the 2016 Conference of the North American Chapter Of the Association for Computational Linguistics:* Human Language Technologies, San Diego, CA, USA, 12–17 June, pp. 1480-1489, 2016.
6. Goldberg, Y. Natural language processing based on deep learning, Beijing, *Mechanical Industry Press*, 2018.
7. Zhen, W. and Mao-ting, G. Design and implementation of image recognition algorithm based on convolutional neural networks, *Modern Computer*, vol. 20, pp. 61-66, 2015.
8. Zoph, B. and Le, Q. Neural architecture search with reinforcement learning, *ArXiv*, doi:10.48550/ARXIV.1611.01578, 2017.
9. Zhang, S., Li, X., Zong, M. and Zhu, X. Learning k, for KNN Classification, *Acm Transactions on Intelligent Systems & Technology*, vol. 8, no. 3, pp. 43, 2017.
10. Kim, J. Ji, H. Oh, S. Hwang, S. Park, E., and Pobil, A. A deep hybrid learning model for customer repurchase behavior, *Journal of Retailing and Consumer Services*, vol. 59, 2021, 102381,ISSN 0969-6989, https://doi.org/10.1016/j.jretconser.2020.102381, 2021.
11. Younas, K. Z., Niu, Z., Nyamawe, A., Haq, I. A deep hybrid model for recommendation by jointly leveraging ratings, reviews and metadata information", *Engineering Applications of Artificial Intelligence*, vol. 97, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2020.104066, 2021.
12. Li, J., Wu, C., Song, R., Xie, W., and Gao, X. Deep hybrid 2-D-3-D CNN based on dual second-order attention with camera spectral sensitivity prior for spectral super-resolution, *In IEEE Transactions on Neural Networks and Learning Systems*, doi: 10.1109/TNNLS.2021.3098767, 2021.
13. Ma, Y. Peng, H., Khan, T., Cambria, E., and Hussain, A. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis, *Cognitive Computation*, vol. 10, pp. 639-650, 2018.
14. Qin, P., Xu, W., and Guo, J. An empirical convolutional neural network approach for semantic relation classification, *Neurocomputing*, vol. 190, pp. 1-9, 2016.
15. Zhang, H. Gan, W. and Jiang, B. Machine learning and lexicon based methods for sentiment classification: a survey, *In Proceedings of the 11th Web Information System and Application Conference*, Tianjin, pp. 262-265, 2014.
16. Giatsoglou, M., Vozalis, M., Diamantaras, K., Vakali, A. Sarigiannidis, G. and Chatzisavvas, K. Sentiment analysis leveraging emotions and word embeddings, *Expert Systems with Applications*, vol. 69, pp. 214-224, 2017.
17. Ren, Y. Zhang, Y. Zhang, M. and Ji, D. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings, *In Proceedings of 30th AAAI Conference on Artificial Intelligence*, Phoenix, pp. 3038-3044, 2016.
18. Ravi, K. and Ravi, V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications, *KnowledgeBased Systems*, vol. 89, pp. 14-46, 2015.
19. Araque, O., Corcuera-Platas, I. Sánchez-Rada, J. and Iglesias, C. Enhancing deep learning sentiment analysis with ensemble techniques in social applications, *Expert Systems with Applications*, vol. 77, pp. 236-246, 2017.
20. Severyn, A. and Moschitti, A. Twitter sentiment analysis with deep convolutional neural networks, *In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, pp. 959-962, 2015.
21. Mikolov, T. Grave, E. Bojanowski, P. Puhrsch, C. and Joulin, A. Advances in Pre-training distributed word representations, *In Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, pp. 52-55, 2018.
22. Lauren, P., Qu, G. Zhang, F. and Lendasse, A. Discriminant document embeddings with an extreme learning machine for classifying clinical narratives, *Neurocomputing*, vol. 277, pp. 129-138, 2018.
23. Zhou, F. Jin, L. and Dong, J. A review of convolutional neural networks, *Chinese Journal of Computers*, vol. 40, no. 6, pp. 1229-1251, 2017.

Sandhya Alagarsamy has completed the Bachelor's degree in Computer Science and Engineering from SASTRA Deemed University, Master degree in Information Technology from Sathyabama University, Chennai, India. She is persuing her doctorate in the field of Big Data Analytics in Sathyabama University. She has 5 years of Industry experience and 7 years of teaching experience. She has published more than 5 papers in conferences and journals. Her current areas of interest include Bigdata, Artificial Intelligence and Deep learning.

Dr. Visumathi James has completed the Bachelor's degree in Computer Science and Engineering from Manonmaniam Sundaranar University, Master and Ph.D. degree in Computer Science and Engineering from Sathyabama University, Chennai, India. She is working as Professor in the department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India. She has 20 years of teaching experience. She has published more than 75 papers in conferences and journals. Her current areas of interest include Network Security, Data mining, Bigdata, Cloud Computing and Artificial Intelligence.