A Research Paper on Negation Handling: Sentiment Analysis Using Super Ensemble Method in Deep Learning

Sarita Bansal Garg [1] *and V. V. Subrahmanyam* [2]

Abstract

Sentiment analysis, a vital technique for comprehending the ideas and attitudes represented in natural language text includes negation handling as a key component. The technique of automatically identifying and categorising the polarity of the sentiment expressed in a text, which might be positive, negative, or neutral, is known as sentiment analysis. Negation handling is tricky too because words like "not", "no", and "never" can flip the polarity of a mood, making it challenging to recognise the sentiment being represented. Therefore, in this research paper, a model is proposed using the super ensemble technique which can correctly comprehend the sentiments expressed in the reviews and give approximately 96% accuracy.

Keywords : **Sentiment Analysis, Negation Handling, Super Ensemble Method, Two Level Ensemble**

I. INTRODUCTION

In the last decade and so, we have seen a worldwide rise in the use of sentiment analysis due to the rapid growth in technology related to computation power and storage. The rise in the use of social media is also one of the major factors contributing towards the increasing usage of sentiment analysis. It is a field dedicated to Natural Language Processing which tends to analyse the sentiments of the users expressing his/her opinion about any topic related to product, person etc. on social media. Views of the users can be in any form and may lack proper structure of the sentences. So, sentiment analysis needs to correctly detect the sentiments from these opinions and categorize them into different categories (e.g., happy, sad, neutral etc.). However, such high-level classification is possible only if we can correctly detect and categorize the sentiments expressed through them.

Even though sentiment analysis has been extensively researched in academia and used in industry, it is still a difficult undertaking. Finding negations in views and correctly classifying them in accordance with the sentiments they reflect are two of the biggest issues in the field of sentiment analysis. In this research, we suggested an ensemble learning technique that combines the outputs of these models with the stacking ensemble method and uses deep learning models as the foundation classifier. In fact, the suggested method starts by building and training five deep learning models, including CNN, LSTM, and different combinations of BiLSTM. Next, a neural network model is utilised as a meta-learner to aggregate the outputs of base classifiers. The proposed model is tested on the complete sentences of Stanford Sentiment Treebank Dataset. This model with an accuracy of 95.6% accuracy outperformed the benchmark which had 85.4% accuracy, and it was tested on the same dataset developed using Recursive Neural Tensor Network. To the best of our knowledge, this is the first instance in which a meta-learner is used to combine various deep learning models in a task involving negation-based sentiment analysis.

Paper Submission Date : March 12, 2023 ; Paper sent back for Revision : March 20, 2023 ; Paper Acceptance Date : March 23, 2023 ; Paper Published Online : June 5, 2023

[1] S. B. Garg *(Corresponding Author), Associate Professor,* Maharaja Agrasen Institute of Management Studies, PSP Area, Sector - 22, Rohini, Delhi. Email : saritabansal2607@gmail.com ; ORCID iD : https://orcid.org/0000-0001-5663-7156

[2] Prof. V. V. Subrahmanyam, *Professor,* School of Computer & Information Science, Block 3, Indira Gandhi National Open University, Maidan Garhi, Delhi. Email : vvsubrahmanyam@ignou.ac.in ; ORCID iD : https://orcid.org/0000-0003-4872-4828

DOI : https://doi.org/10.17010/ijcs/2023/v8/i3/172862

II. RELATED WORK

Deep Learning, Machine Learning, and sentiment dictionaries are some of the basic current text sentiment analysis techniques. The accuracy of the results of the sentiment classification depends on how well the sentiment dictionary is constructed, which is a need for the approach based on sentiment dictionaries [1]. Machine learning-based sentiment analysis necessitates the manual selection of complex features [2], [3] and feature selection affects the final categorization. Deep learning-based sentiment analysis techniques that can automatically extract features are frequently employed to address text sentiment analysis issues.

Convolutional Neural Network (CNN) model was used by [4] for aspect-based sentiment classification. For aspect-based sentiment analysis, [5] suggested an Attention-based Long Short-Term Memory (LSTM) model. When different components are given as input, the attention mechanism can focus on various parts of a sentence. For aspect-based sentiment categorization, [6] employed an attention-based Gated Recurrent Unit (GRU) model. For the purpose of identifying negation cues, [7] used the LSTM Deep Learning approach.

Convolutional models have the drawback of not accounting for the word order in a sentence, which prevents them from deriving accurate meanings from sentences. When using data with images, CNN performs well. The authors [8] designed and applied a two stage Convolutional Neural Network-based deep learning framework to address the unbalanced problem for traffic data and to identify the vehicle type. To address the issue of an uneven data set, data augmentation with balanced sampling is used in the first stage. In the second stage, parameters are learned from the expanded training data set, and an ensemble of CNN models with various architectures is built. To identify photos based on the initial predictions of a single model, maximum majority voting is used. Due of the odd number of models in the ensemble, situations with identical votes won't happen. The results of the experiment demonstrate that when compared to the baseline methods for vehicle classification, the proposed model was effective in raising the mean precision to some amount.

This issue with CNN models can be resolved by recurrent (memory-based) models, such as LSTM or GRU, which can learn long-term and syntactic dependencies. On the other hand, memory-based algorithms are unable to carry out reliable analysis when a sentence's word order changes [9].

The authors of [10] addressed the issue of domain adaptation for sentiment categorization using deep learning techniques and ensemble learning. In particular, they extract text properties using a convolutional neural network (CNN) and then utilise a random forest classifier to forecast the future. The authors employ two datasets from distinct domains, one from product reviews and one from movie review to assess the effectiveness of their methodology. They suggested a framework for representing high level features across domains and minimising generalisation error. For feature learning and representation, a marginalised stacked de-noising autoencoder (mSDA) constituted of m closed-form de-noising layers was utilised. The ensemble approaches and several learners were taken into consideration, and results were obtained by utilising the mSDA. The suggested framework has shown to be significantly better.

Authors of [11] suggested a brand-new ensemble approach for categorising the modality of medical photos. The proposed ensemble made use of numerous fine-tuned CNNs as optimised feature extractors to learn picture features that caught the variety of information present in medical images from various modalities. A more effective image categorization system than the individual CNNs was produced by the ensemble, which combined the optimised CNN models. According to experimental findings, ensemble outperformed other CNN baselines and other approaches using the same benchmark training dataset in terms of classification accuracy, properly classifying the majority of the images in a public benchmark dataset.

In this research, we introduced an ensemble learning method that uses deep learning models as base classifiers and combines the outputs of these models with the stacking ensemble approach, taking into account the strengths and weaknesses of each deep learning classifier. In the proposed approach, five deep learning models, including CNN, LSTM, and various combinations of BiLSTM, are initially created and trained using varied K-fold cross validation data, where k = 3, 5, and 10. Increasing the ambiguity (while not increasing individual generalization errors) of the models will improve the overall generalization of the model [12]. So, using a variety of approximators, such as a mix of neural networks with various topologies or a variety of entirely distinct types of approximators is one technique to raise the ambiguity of the ensemble. Another approach is to train the networks on different training generated using a cross-validation technique.

The outputs from the base classifiers are then combined using a neural network model acting as a meta-learner. To the best of our knowledge, this is the first instance in which a meta-learner has been used to combine various deep learning models in a task involving negation-based sentiment analysis.

III. BASELINE MODEL

The authors [14] introduce a new model, the Recursive Neural Tensor Network (RNTN), which is recursively using tensor-based neural processes to capture the compositional structure of spoken language phrases. The model first creates a parse tree from the input sentence, and then it calculates the sentiment of each phrase in the sentence. The sentiment score for the sentence is calculated by adding the sentiment scores of each of its component phrases.

The Stanford Sentiment Treebank, a dataset of movie reviews with fine-grained sentiment labels is used by the authors to test the performance of their algorithm. They demonstrate that their model outperforms other cutting-edge methods when compared to their model, particularly for longer and more complicated words. It makes a significant addition to the fields of sentiment analysis and Natural Language Processing, showing the potential of Deep Learning models for capturing the compositional structure of language.

IV. ENSEMBLE LEARNING METHODS FOR DEEP LEARNING NEURAL NETWORKS

Neural networks for Deep Learning are nonlinear techniques. They provide more flexibility and can scale in line with the quantity of training data that is available. They learn using a Stochastic training technique, which means they are sensitive to the environment, which is a drawback of this flexibility. Each time they are trained, they may discover a different set of weights based on the peculiarities of the training data, which results in different predictions.

When trying to create a final model to utilize for making predictions, this is commonly referred to as neural networks having a high variance and it can be irritating. When trying to create a final model to utilise for making predictions, this is commonly referred to as neural networks having a high variance and it can be irritating. By training numerous models instead of just one, and combining their predictions, neural network models can successfully reduce their variance. This process known as ensemble learning not only lowers the variance of predictions but also has the potential to produce predictions that are superior to those produced by a single model. [14] provides a thorough overview of ensemble learning.

Each of the three major elements of the ensemble method can be varied, for example:

- **Training Data:** To train each model in the ensemble, utilise a different set of training data.
- **Ensemble Models:** Choose variety of models to be used in the ensemble.
- **Combinations:** Choose variety of methods to integrate the results from ensemble members.

***A. Training Data*:** To calculate the generalisation error of the selected model configuration, the simplest method would be to utilise k-fold cross-validation. This process involves training k distinct models on k distinct subsets of the training data. Then, these k models can be used as ensemble members by being saved.

The training dataset is resampled using replacement in a method known as bootstrap aggregation, or bagging for short. Next, a network is trained using the resampled dataset. Due to the resampling process, each training dataset has a unique composition and may contain duplicate cases. As a result, the model trained on the dataset may have somewhat different expectations for the density of the samples and, consequently, may have varying generalisation error.

Other methods may entail randomly allocating a portion of the input space, such as a portion of the hyper-volume or a portion of the input features to each model.

Since each dataset example is only used once in the test dataset to estimate model performance, the K-fold cross-validation method is employed in the proposed model since it is less biased. The number of groups into which a specific data sample is to be divided is indicated by the procedure's sole parameter, k. A typical value for k is 10. Since we have a small dataset, so along with value of k=10, we are also considering $k = 1$ and $k = 3$.

B. Ensemble Models: Given the complexity of the issue and the stochastic nature of the learning procedure, training the same under-constrained model on the same data with various initial circumstances will produce distinct models. This is because there are numerous "good" and "different" ways to map inputs to outputs because the optimization problem the network is trying to solve is so difficult. Although generalisation error may not be significantly improved, the variance may be minimized as a result. Because all of the models have learnt comparable mapping functions, the mistakes that the models produce may still be too closely associated.

Alternate strategies include changing the configuration of each ensemble model by utilising networks with various capacities (e.g. different numbers of layers or nodes) or models that were trained under various circumstances (e.g. learning rate or regularization).

The end result might be an ensemble of models with a less correlated set of predictions and prediction error because they have trained a more diversified set of mapping functions.

In the proposed model, we have considered a heterogenous collection of 5 Deep Learning models including LSTM, BiLSTM, and CNN with varying layers and nodes.

C. Combinations: Calculating the average of the forecasts from the ensemble members is the simplest technique to combine the predictions. The predictions from each model can be weighted, and the weights can be tuned using a hold-out validation dataset. The result is a weighted average ensemble, often known as model blending.

Using a new model to figure out how to most effectively integrate the predictions from each ensemble member represents a further step in complexity. The model could be straightforward linear or sophisticated nonlinear, taking into account both the unique input sample, and the predictions made by each member. Model stacking, also known as layered generalisation, is a general method of learning a new model.

The neural networks are being used as sub-models. Therefore, using a neural network as a meta-learner is preferred. The sub-networks are incorporated into a sizable multi-headed neural network that learns how to combine the predictions from each input sub-model in the most effective way. This makes it possible to think of the stacking ensemble as a single big model. This method has the advantage of immediately giving the meta-learner the outputs of the sub-models. We did this by creating models using the Keras functional interface.

Each of the loaded models is used as a distinct input-head to the defined large stacking ensemble model after the models have been loaded as a list. To prevent weight updates when the new, larger model is being trained, it is necessary to identify all of the layers in each of the loaded models as not trainable. The names of each layer in each of the loaded models will need to be modified to reflect to which ensemble member they belong because Keras also mandates that each layer have a unique name.

This model will employ a distinct input head for each input layer from the sub-models. This means that the model will require k copies of any input data, where k is the total number of input models. We used K-fold cross validation to train the level 0 classifiers in the proposed model (5). Therefore, for $k = 3$, 5, and 10 accordingly, we had 15, 25, and 50 trained models. The results from all the models can then be combined. In order to build a single vector from the five class-probabilities predicted by each of the five models, we have performed a straightforward concatenation merging.

V. PROPOSED METHOD

In this study, we presented a novel stacked ensemble learning-based method for the classification of sentiments and identification of negations. The suggested approach uses a meta classifier to combine the output from five deep learning models, including CNN, LSTM, and BiLSTM as basis classifiers. Combining different deep learning models allows us to benefit from their structural and functional benefits, which improves overall performance. In the sections that follow, we go into further detail into the base learners and the meta classifier.

The initial basic classifier we used in this paper was a BiLSTM network. The BiLSTM model travels the text in two directions, accounting for both the previous and following words in a sentence, in order to extract semantics. The five layers of the BiLSTM model in this paper are the embedding, BiLSTM, Dense, Dropout, and Dense Layer layers.

Because the embedding layer transforms the input words into a vector of integers, words that have similar meanings in the context are embedded adjacent to one another. Prior to training, we removed punctuation and unusual terms. The sample data we used, totalling 17,611, completely determines the size of our vocabulary. We can fold each word in as many different ways as we like. We assumed that the embedding vector's size was 32.

BiLSTM, the layer below, has 128 neurons, 512 neurons, a dropout with a weight of 0.50 and the activation function "Relu," in addition to 512 neurons. Our BiLSTM classifier divides comments and reviews into five categories: very negative, negative, neutral, positive, and very positive. This dense layer has five neurons. Softmax is the activation function of the final layer. The softmax function generates a number between 0 and 1 that represents the likelihood of the target classes for each class. We used the categorical-cross entropy loss function and the Adam optimization approach to update the weights of the network.

The following basic classifier is CNN. CNN can automatically glean crucial information from the text. The top layer is our CNN network's embedding layer, which is in charge of vectorizing words. The next layer is a convolutional layer. The result of convolutioning the inputs and kernels is fed into a ReLU function. The dimensions are then reduced using MaxPooling1D. The next layer is substantial and contains five softtmax activated neurons. We optimise using the Adam method with categorical cross-entropy as the loss function.

With the third classifier having three BiLSTM layers and the fourth having two layers with varying numbers of neurons from 128 to 256, the third and fourth basic classifiers in this method are distinct combinations of BiLSTM. The fifth neuron in the dense layer of the final layer has both ReLU activation function and soft max activation function.

We are using three BiLSTM classifiers since we need to re-evaluate the data for the detection and classification of negated words as well as its overall impact.

The final classifier that we use combines the CNN and LST models. The LSTM model can maintain the phrase-by-phrase sequentiality, making it very effective at modelling large texts and extracting information [15].

There are four layers in the model. The first step is embedding, followed by the second step, spatialdropout with 0.4, the third step, LSTM with 256 neurons, and the last step, dense with 5 neurons and softmax activation.

After creating the fundamental classifiers, we employ the previously described advantages of various models using ensemble learning. We provide a more detailed description of our models' structure next.

Our suggested model is based on a stacking ensemble technique in which we first train our basic classifiers before combining their outputs through the stacking method. In this study, we combine the output from the classifiers indicated earlier using neural networks. In more detail, the sub-networks can be incorporated into a bigger multi-headed neural network, which will then learn how to integrate the predictions from each input sub-model in the most effective way. It makes it possible to think of the stacking ensemble as a single, enormous model.

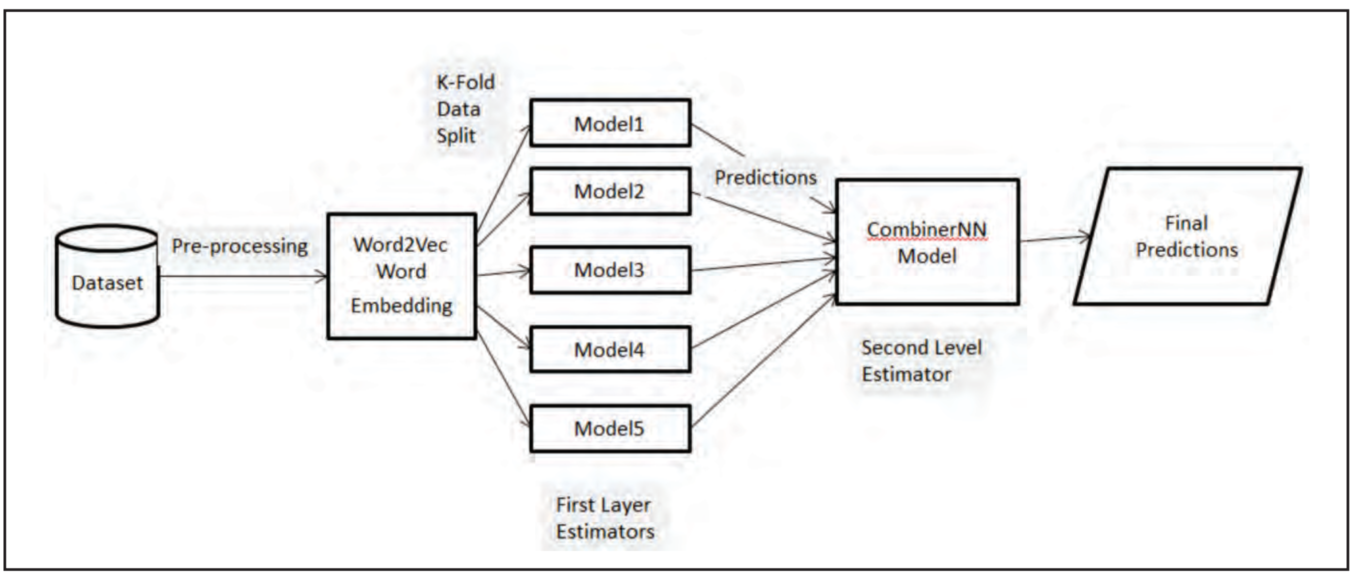

Fig. 1. Proposed Model

This method has the advantage of immediately giving the meta-learner the outputs of the sub-models. If necessary, it is also possible to update the sub-model weights in conjunction with the meta-learner model. The Keras functional interface for developing models can be used to accomplish this.

Each of the loaded models can be utilised as a distinct input-head to the bigger stacking ensemble model once they have been loaded as a list. To prevent weight updates when the new, larger model is being trained, it is necessary to identify all of the layers in each of the loaded models as not trainable. Additionally, Keras demands that each layer have an own name. Therefore, the names of each layer in each of the loaded models will have to be updated to indicate to which ensemble member they belong. The outline of the proposed model is displayed in Fig. 1.

VI. RESULTS AND DISCUSSION

In this section, we describe the datasets used in this work and discuss the experimental results.

A. Dataset

The model is applied to the Stanford Sentiment Treebank Dataset in order to assess the effectiveness of our suggested strategy. This dataset consists of 11,855 labelled Rotten Tomatoes movie reviews. Reviews were rated out of five (from zero to four) on a scale of extremely favourable, neutral, negative, and very bad. The considered dataset is the dataset that has been labelled after taking into account all of the negative ratings.

Due to the short size of our dataset, we experimented with several resampling approaches and found that the K-Fold cross validation method typically produces the best results. Unlike random train-test splits, where a given example may be utilised many times to evaluate model performance, the method is less biased because each example in the dataset is only used once in the test dataset to estimate model performance.

The process contains a single parameter k that designates how many groups should be created from a given data sample. An estimate of model performance that is less skewed is produced by averaging the scores of each model. For the three values of k = 3, 5, and 10, the model was evaluated.

For training at the ensemble's first level, we employ each of the five base classifiers described in the previous section. A cross-validation ensemble can be created by combining the models that are produced throughout the cross-validation process; this ensemble is likely to perform better overall than any one particular model.

The data is pre-processed using the common nltk and nlp-preprocessor packages before being input to the models. The pre-trained word embedding is used to create word embedding matrices. [16] demonstrates that word2vec word embedding produced superior outcomes for our dataset and challenge. Therefore, we also utilised Word2Vec's pre-trained word embedding here.

B. Experimental Results

For evaluating our proposed method, we applied it on Stanford Sentiment Treebank. We used 20 epochs for training each classifier and considered the batch size equal to 32. To prevent overfitting, we used Early stopping technique in all models. Other model-related parameters are specified in Proposed Model. Network graph is created with plot_model() function to have an idea of how the ensemble model fits together. For three values of k = 3, 5, 10 first level of ensemble will have 15, 25, and 50 models respectively. First level of models will combine into single layer with two additional layers to form the second level of ensemble. It is difficult to show the network graph here as it is too huge to be displayed in a single page.

For evaluation, we used accuracy as performance measure which is defined as:

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN)$$

Where,

TP = True Positives ;

TN = True Negatives ;

FP = False Positives ;

FN = False Negatives.

Table I lists the accuracy for both level 1 and level 2. Results clearly indicate the increased accuracy in correct identification of negation cues in the reviews.

TABLE I.
ACCURACY LEVELS OF THE PROPOSED MODEL.

Value of *K*	Accuracy of the first level base classifiers	Accuracy of the Proposed Model
3	0.6782	0.910
5	0.7111	0.938
10	0.7633	0.956

C. Compared Model

The Recursive Neural Tensor Network (RNTN) baseline model achieved state-of-the-art performance on the Stanford Sentiment Treebank dataset, but our Super Ensemble Model give us 95.6% accuracy on the fine grained prediction across all sentences and captures negations of different sentiments and scope more accurately than a test set accuracy of 85.4% of RNTN.

Moreover, the RNTN model outperformed several other models, including the Recursive Neural Network (RNN), the Sentiment-specific RNN (S-RNN), and the Constituent Sentiment Tree (CST) model, which were also evaluated on the same dataset.

The authors also conducted experiments to analyze the performance of the RNTN model on different types of sentences, such as long sentences and sentences with negations. They found that the RNTN model outperformed the other models on these types of sentences as well.

Overall, the results demonstrate the effectiveness of the RNTN model for capturing the compositional structure of language and for improving the accuracy of sentiment analysis on complex and nuanced sentences.

VII. CONCLUSION

Ensemble Methods are very powerful tools to increase the efficiency of the individual deep learning models. Deep Learning Models tend to overfit the training data if parameters are not correctly defined. Also, different models have different characteristics. Ensembling techniques tend to utilize all these characteristics of different models by combining them. We also tried to use this efficiency of technique and proposed a model which is effective in correctly identification of sentiments.

AUTHORS' CONTRIBUTION

Prof. Subrahmanyam guided and supervised the research and Sarita Bansal Garg worked on the model and conducted the experiment. Both the authors collectively finalized the paper.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in the manuscript.

FUNDING ACKNOWLEDGEMENT

The authors have not received any financial support for the research, authorship, and/or for the publication of the article.

REFERENCES

[1] A. Al-Thubaity, Q. Alqahtani, and A. Aljandal, "Sentiment lexicon for sentiment analysis of Saudi dialect tweets," *Procedia Comp. Sci.*, vol. 142, pp. 301–307, Nov. 2018, doi: 10.1016/j.procs.2018.10.494.

[2] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, "Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment," *Expert Syst. Appl.*, vol. 109, pp. 66–85, Nov. 2018, doi: 10.1016/j.eswa.2018.05.010.

[3] C. Shang, M. Li, S. Feng, Q. Jiang, and J. Fan, "Feature selection via maximizing global information gain for text classification," *Knowl. Based Syst.*, vol. 54, pp. 298–309, Dec. 2013, doi: 10.1016/j.knosys.2013.09.019.

[4] H. Xu, B. Liu, L. Shu, and P. Yu, “Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction.” Accessed: Dec. 06, 2022. [Online]. Available: https://arxiv.org/pdf/1805.04601.pdf

[5] Y. Wang, M. Huang, L. Zhao, and X. Zhu, “Attention-based LSTM for aspect-level sentiment classification,” in *Proc. 2016 Conf. Empirical Methods Natural Lang. Process.*, pp. 606–615, Austin, Texas, November 1–5, 2016. [Online]. Available: https://aclanthology.org/D16-1058.pdf

[6] Y. Xing and C. Xiao, “A GRU Model for Aspect Level Sentiment Analysis,” *J. Physics: Conf. Series*, vol. 1302, p. 3, 032042, Aug. 2019, doi: 10.1088/1742-6596/1302/3/032042.

[7] D. Gautam, N. Maharjan, R. Banjade, L. J. Tamang, and V. Rus. "Long Short Term Memory Based Models for negation handling in tutorial dialogues." in *FLAIRS Conf.*, pp. 14–19. 2018, doi: 10.13140/RG.2.2.26250.36804.

[8] W. Liu, M. Zhang, Z. Luo, and Y. Cai, “An Ensemble Deep Learning Method for vehicle type classification on Visual Traffic Surveillance sensors,” *IEEE Access*, vol. 5, pp. 24417–24425, 2017, doi: 10.1109/ACCESS.2017.2766203.

[9] F. Yang, C. Du, and L. Huang, “Ensemble sentiment analysis method based on R-CNN and C-RNN with Fusion Gate,” *Int. J. Comput. Commun. Control*, vol. 14, no. 2, pp. 272–285, Apr. 2019. [Online]. Available: https://univagora.ro/jour/index.php/ijccc/article/view/3375. Accessed: May 9, 2023.

[10] D. Nozza, E. Fersini, and E. Messina, “Deep Learning and Ensemble Methods for Domain Adaptation,” *IEEE Xplore*, Nov. 1, 2016 in *2016 IEEE 28th Int. Conf. Tools Artif. Intell.*, San Jose, CA, USA, 2016, pp. 184–189, Accessed: May 9, 2023. doi: 10.1109/ICTAI.2016.0037.

[11] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, “An ensemble of fine-tuned Convolutional Neural Networks for Medical Image Classification,” *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 31–40, Jan. 2017, doi: 10.1109/jbhi.2016.2635663.

[12] A. Krogh and J. Vedelsby, “Neural Network Ensembles, Cross Validation, and Active Learning,” *Neural Inf. Process. Syst.*, pp. 231–238, 1994. Accessed: May 9, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1994/file/b8c37e33defde51cf91e1e03e51657da-Paper.pdf

[13] Z.-H Zhou. “Ensemble Learning.” In: Li, S.Z., Jain, A. (eds) Encyclopedia Biometrics. Springer, Boston, MA, 2009. doi: 10.1007/978-0-387-73003-5_293.

[14] K. Smagulova, A. P. James, “A survey on LSTM memristive neural network architectures and applications,” *Eur. Phys. J. Spec. Top*. vol. 228, pp. 2313–2324, 2019, doi: 10.1140/epjst/e2019-900046-x.

[15] S. B. Garg and V. V. Subrahmanyam, “Sentiment analysis: Choosing the Right Word Embedding for Deep Learning Model,” *Adv. Comput. Intell.Technologies*, pp. 417–428, Jul. 2021, doi: 10.1007/978-981-16-2164-2_33.

[16] S. B. Garg and V. V. Subrahmanyam, “Sentiment analysis: Choosing the right word embedding for deep learning model,” in M. Bianchini, V. Piuri, S. Das, R. N. Shaw (eds) *Adv. Comput. Intell. Technologies*. Lecture Notes in Networks and Systems, vol. 218, 2022. Singapore : Springer, doi: 10.1007/978-981-16-2164-2_33.

About the Authors

Sarita Bansal Garg has been associated with Maharaja Agrasen Institute of Management Studies since 2003. She is a Research Scholar at IGNOU and was a Gold Medallist in M.C.A at Guru Jambheshwar University, Hisar.

Prof. V. V. Subrahmanyam is Faculty with School of Computer and Information Sciences (SOCIS), IGNOU since 1998 and has over 22+ years of research and teaching experience. He has 2 patents and more than 50 publications to his credit.