International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 1, February 2025, pp. 949~957
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i1.pp949-957

Enhancing sentiment analysis through deep layer integration with long short-term memory networks

Parul Dubey[1], Pushkar Dubey[2], Hitesh Gehani[3]
[1]Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed to be University), Pune, India
[2]Department of Management, Pandit Sundarlal Sharma (Open) University Chhattisgarh, Bilaspur, India
[3]School of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Ramdeobaba University, Nagpur, India

Article Info

Article history:

Received Jul 12, 2024
Revised Aug 27, 2024
Accepted Sep 3, 2024

Keywords:

Deep learning
Long short-term memory
Natural language processing
Sentiment analysis
Text classification

ABSTRACT

This involves studying one of the most important parts of natural language processing (NLP): sentiment, or whether a thing that makes a sentence is neutral, positive, or negative. This paper presents an enhanced long short-term memory (LSTM) network for the sentiment analysis task using an additional deep layer to capture sublevel patterns from the word input. So, the process that we followed in our approach is that we cleaned the data, preprocessed it, built the model, trained the model, and finally tested it. The novelty here lies in the additional layer in the architecture of LSTM model, which improves the model performance. We added a deep layer with the intention of improving accuracy and generalizing the model. The results of the experiment are analyzed using recall, F1-score, and accuracy, which in turn show that the deep-layered LSTM model gives us a better prediction. The LSTM model outperformed the baseline in terms of accuracy, recall, and f1-score. The deep layer's forecast accuracy increased dramatically once it was trained to capture intricate sequences. However, the improved model overfitted, necessitating additional regularization and hyperparameter adjustment. In this paper, we have discussed the advantages and disadvantages of using deep layers in LSTM networks and their application to developing models for deep learning with better-performing sentiment analysis.

Corresponding Author:

Parul Dubey
Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed to be University)
Pune, India
Email: dubeyparul29@gmail.com

1. INTRODUCTION

Sentiment analysis (also known as opinion mining or emotion artificial intelligence (AI)) is a part of natural language processing (NLP) that focuses on analyzing and determining the sentiments of text data. Because of the increase in user-generated information in the digital space, and social networking sites in particular, sentiment analysis has been a valuable tool for both researchers and businesses. In this investigation, LSTM networks and other recurrent neural networks (RNNs) will be analyzed in the case of sentiment analysis. Sentiment analysis is arguably one of the most exciting areas of research in NLP in that it allows us to automatically determine whether the text is neutral, positive, or negative. While traditional methods used different machine learning algorithms, deep learning has improved sentiment analysis models in such a way that they are now much more accurate and robust. The long short-term memory (LSTM) network, with its ability to handle long-term dependencies in textual data, has made significant improvements in language modeling and has become one of the better deep learning network architectures for processing sequential data instead of traditional recurrent neural networks (RNNs).

One potential way to increase performance further with LSTM networks is to have a deeper layer. In many cases, deeper LSTM architectures are used to improve the model's ability to generalize to unseen data and learn more complex dimensional patterns and interactivities in the data. The additional layer could help increase key performance measures such as F1-score, recall, and accuracy, as it will make the model better equipped to understand complex connections between content in the text. We are going to merge LSTM and deep layers to introduce more reliable predictions about the sentiment of our model. Table 1 shows the list of previous work that has been done.

Table 1. Comparison of literature reviewed

Study	Dataset Source	Techniques	Key Findings	Comparison with proposed model
Srinivas *et al.* [1]	Twitter (Kaggle)	Convolutional neural network (CNN), LSTM, simple neural network	LSTM had the highest accuracy with 87% accuracy	Training accuracy better than previous models-98.34%
Muhammad *et al.* [2]	Indonesian hotel reviews	Word2Vec, LSTM	Highest accuracy with Word2Vec+LSTM-85.96%	
Mahadevaswamy and Swathi [3]	Amazon product reviews	Bidirectional LSTM	Better predictions with bidirectional LSTM	
Gandhi *et al.* [4]	Twitter (IMDB)	CNN, LSTM	Improved detection of tweet sentiment and reviews-87.74% (tweets), 88.02% (reviews)	
Behera *et al.* [5]	Social media reviews	Convolutional LSTM (Co-LSTM)	Better outcomes in social big data sentiment	
Jin *et al.* [6]	Stock market data	Sentiment analysis, LSTM, Attention mechanism	Enhanced stock price prediction accuracy	

An architectural depiction is used to further discuss the implemented procedure. The following section explains dataset description and data gathering. A detailed explanation of the model's construction, training, and hyperparameter tweaking is provided after the findings. Ultimately, we wrap up the paper with improved performance when compared to earlier models.

2. METHOD

Our sentiment analysis method incorporates a number of critical steps, including data prediction and model evaluation. It is an improved version of LSTM networks with an extra deep layer. As such, our initial task will be to amass a gigantic database of reviews made by users and annotated with their sentiment. The text data in this dataset has been pre-processed by cleaning the initial text, using stemming or lemmatization, and removing stop words; hence, it is in a tokenized and standardized form [7]–[9]. After this phase, the text is ready for analysis [10]. The next step is to convert this pre-processed text into numerical representations using text vectorization methods, e.g., word embeddings (e.g., GloVe). We had to do this to ensure that our model works well with text data. We can see that some of the structure, when we do our own model, is a multi-layer architecture. The embedding layer is the first layer, which converts the input text into word embeddings. The next two layers, named “LSTM,” which here stands for long short-term memory [11], [12], are specialized to detect sequential dependencies present in text data [13], [14]. Here, “deep,” which sits above dense, recognizes more complex patterns and interactions. A binary classification is done in the output layer via a single neuron with sigmoid activation, and each layer is a dense layer.

The model is trained on the training dataset. For this, to keep from overfitting, we tune the weights in this step and monitor the loss in training and validation, all this with the aid of our friend Adam optimizer. In the end, we will evaluate the efficiency of the model by deriving the F1-score, accuracy, precision, and recall. The model is then validated on a separate validation dataset to ensure it is functional and generalizable. The architecture of the proposed model can be seen in Figure 1. To the best of our knowledge, our model is the first to incorporate a deep layer at the LSTM level, and it offers an efficient solution to sentiment analysis tasks with the ability to generalize and predict more accurately. From data preparation through model validation, this extensive approach ensures each step is taken with the utmost rigor to obtain the best results.

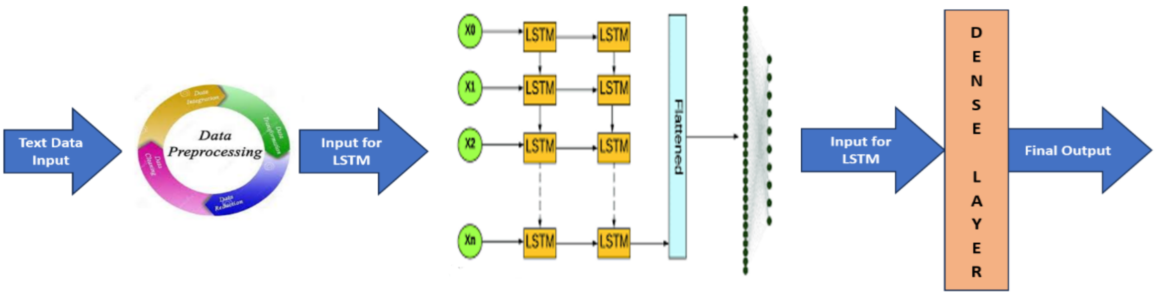

Figure 1. Architecture of proposed model

As user reviews with associated sentiment labels make up the dataset used for this sentiment analysis project, which is derived from Kaggle. The dataset is presented here in great detail:

a. Structure of the dataset: Most often, the dataset requires the following columns:
   - Review text: The review text itself is in this column. This column contains strings, i.e., user text, expressing an opinion or comment about products, services, or experiences.
   - Sentiment Label - Every review has its sentiment labels in this column. Basically, the mood is just a yes/no with tags. The labels can also be mathematical. In pneumatical terms, 1 is to be rewarded as positive and 0 is to be negative.

b. Sample data:
   Here's a small sample to illustrate the structure of the dataset in Table 1.

Table 1. Structure of dataset

Review text	Sentiment label
"I love this product! It works great and is very easy to use."	1
"Terrible experience. The item arrived broken and the customer service was unhelpful."	0
"Just okay, nothing special but not bad either."	0
"Fantastic! Exceeded my expectations in every way."	1
"Not worth the money. Poor quality and bad performance."	0

c. Data characteristics
   - Diversity of reviews: between the reviews of a broad range of items and services, there is a diverse dataset that may allow the model to generalize across settings.
   - Length variability: there is a lot of variability in how long the reviews are, from short comments to in-depth analysis. This uncertainty is partly addressed during preprocessing by using padding.
   - Sentiment distribution: the user needs to verify the sentiment label distribution to make the dataset balanced. If there are indeed no balanced data, over-sampling, under-sampling, or balancing class weights in the model training will be required.

d. Importance of preprocessing

Text cleaning involves removing redundant punctuation, HTML tags, and special characters to retain the core content [15], [16]. Tokenization splits the text into individual words or tokens, followed by stop words removal to eliminate common words like "and" and "the." Stemming or lemmatization then simplifies words to their root forms, while padding ensures uniform sequence lengths by adding zeros where necessary [17]–[19].

e. Use in model training

The pre-processed data set is separated into a training set and a testing set. For training the LSTM model, we use the training set and evaluate it on the testing set [20]. These labels are the sentiment labels, which are the target variables that the model learns to use to categorize reviews into sentiment classes.

3. MODEL DEVELOPMENT

The centerpiece of our method is designing an LSTM-based neural network model for sentiment analysis. To transform raw text input into valuable predictions as defined in the tasks, this model architecture depends on many key parts. The following section further explains the architecture of the method in detail. They give us a clear picture of the model.

3.1. Embedding layer

Firstly, the embedding layer of the model is its input layer [21]. Its job is to generate dense word embeddings for the text input. This creates a dense vector of a fixed size for each word in the input sequence, allowing the model to encode semantic information about the words as well as their relationships to one another. Thus, instead of [1, 2, 3, 4], for example, we get [0.1, 0.2], [-0.25, ...], [0.05, 0.1, ...], [0.3, 0.4, ...] instead of word indexes. Moreover, given that we use pre-trained embeddings like GloVe, we can increase text understanding in the model by leveraging pre-existing knowledge about word connections.

3.2. LSTM layer

The core of the model are the LSTM layers, which are designed to learn the temporal dependencies in the text input. LSTM layers were designed specifically to handle the vanishing gradient issue that impacts conventional RNNs (as well as model long-range dependencies) by enabling the network to remember information for long periods of time and to moderate the movement of data through the network [22]. There are sub-functions within them that let them store data, called memory cells, and circulate data, called gates [23]. By stacking LSTM layers, it is possible to greatly enhance the model's capability to learn intricate patterns [24], [25].

3.3. Deep layer

MADE-LSTM with deep layer: we add a thick deep layer, which enables learning more complex patterns and interactions. Its primary function is to help the model learn difficult patterns and correlations that were unnoticed by the LSTM layers operating alone. We can have the intuition that with this extra layer, the model is more able to learn the structure of the input data and to generalize. This provides the model with an unsurpassed ability to shape the incoming data in more complicated ways, allowing the model to mingle and interpret information in a deeper sense. The DHL model is a non-linear hidden layer model because of the Rectified Linear Unit activation function, which helps the DHL model accommodate the non-linearity. The model is able to learn non-linear relationships amongst the data, which helps in the accurate representation of high-degree relationships existing in the data. Now we can do this using the rectified linear unit (ReLU) activation function because this enables our model to train without running into problems such as the vanishing gradient problem. This adds a level of depth, and now the model based on LSTM learns from data in a better way, which is why it performs better in tasks like sentiment analysis. Besides LSTM layers, the additional layer supplements the LSTM layers by helping to discover the correlations and patterns that are generally left unnoticed and, as a result, are capable of generating a more stable as well as accurate table-based representation.

3.4. Dense layers

This is followed by dense layers applied to the output after the LSTM layers. One of the main tasks for these fully connected layers is to take the output of the LSTM and put it into a format that matches up with the final classification. Since all neurons in a given layer are connected to all the neurons in the previous layer, dense layers are able to process very complex data. ReLU activation functions are frequently used to introduce non-linearity in these middle layers, as this helps the model learn more complex correlations in the data.

3.5. Output layer

Finally, the model includes an output layer, the heart of the model, in which with binary classification the model can say if those feelings are positive or negative. This layer comprises a neuron with the sigmoid activation function outputting a value between 0 and 1. This number represents the chance of the positive class, using a common threshold of 0.5 to predict the class label. Anything greater than half means that we are feeling positive emotions and anything less than half means that we are feeling negative emotions.

4. RESULT AND COMPARISON

Precision level, the proportion of positive predictions made by the model that were actually correct. The precision of the basic LSTM model is 0.8, and the LSTM with deep layer has a higher precision of 0.85 in this comparison aspect. This means that the improved model is capable of more accurately detecting the positives, thus reducing the number of false positives.

The base LSTM model has a recall of 0.85; with deep layered LSTM, it reaches 0.91. This is an indication that the deep layer helps the model better discriminate positive cases, which reduces false negatives. The F1-score, which balances precision and recall into a single metric, gives an averaged representation of how well the model predicts when classifying non-main material. The F1-score for the

Basic LSTM Model is 0.80, and the F1-score for the LSTM with deep layer is 0.86. This means the improved model keeps a very good balance between precision and recall, which results in enhanced general performance. With a deep layer on the LSTM, adding it to the model brings a substantial increase in performance for all metrics. Same for basic LSTM only; its deep LSTM layer has better precision, recall, and F1-score. This in turn means that the capable model is performing well; it will report fewer true positives and higher predictions for non-accepted students. So, for applications that need high precision, recall, and balanced performance, LSTM with deep layer is a good option. Figure 2 shows the comparison result based on precision recall and the F1-score.

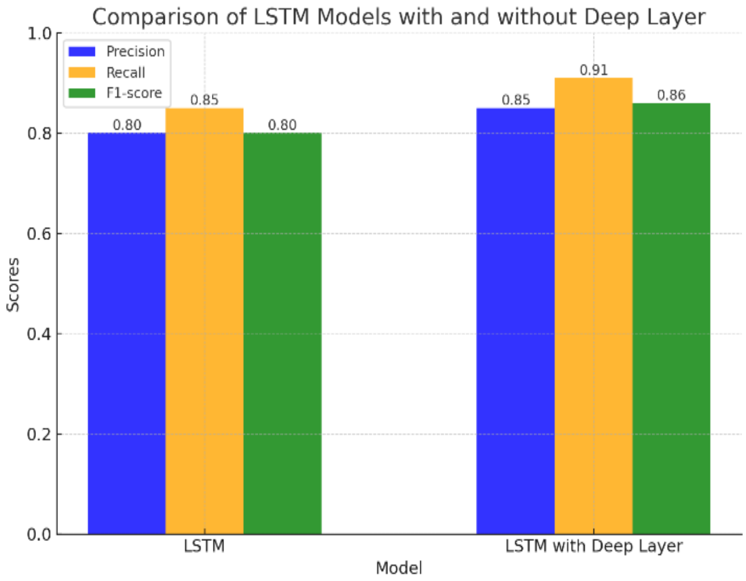

Figure 2. Comparison for precision, recall, and F1-score

4.1. Training and validation loss

In the LSTM model, the train loss is decreasing over the epochs, which indicates that the model is learning from training data very effectively. This consistent decline shows that the model is able to reduce the training error. But the validation loss decreases and then stays put in the early middle; it even starts increasing towards the end, but very slightly. This can be an indication that the model is beginning to overfit the training data during training. In simple words, overfitting arises when the model determines the training data so well that it catches the noise and patterns that are peculiar to the training data and will not track the new, unseen data.

On the other hand, the training loss for the LSTM with deep layer is way shorter than that of the simplistic LSTM model. Such a large decrease in the training loss indicates that the deep model in fact learns the training data extremely well, allowing it to generalize further and pick up complex patterns and relationships. Secondly, we notice that the validation loss is relatively high and does not show a large decrease. The model with the more complicated deep layer is learning patterns that are found within the training data, but these patterns are not general (they're only present in the training data). So, the model is overfitting to the training data and therefore not generalizing as well to the validation data this time. The high validation loss, even though it gives a really good performance on the training set, indicates bad generalization.

4.2. Train vs validation accuracy

Training accuracy for the LSTM model increases at a steady rate across the epochs, which suggests that it is learning consistently. The accuracy increasing slowly shows that the model is predicting the correct sentiment from the training data. The validation accuracy is a touch lower than our training accuracy but fluctuates much less over the course of the epochs. This consistency means that the model is generalizing well and continuing to perform reliably on unseen data.

The LSTM with deep layer quickly gets perfect training accuracy and never moves. This high score suggests the model is overfitting, and it is very good at learning the training data and inferring complex patterns. But it does not see major improvement in the validation accuracy, and it is a bit worse than the basic model, as shown below. The fact that training and validation accuracy are quite distant reaffirms our suspicion of overfitting. Such a model would fit the training data very well but would fail to reach the same level of prediction on the validation set, which means it has poor generalization capability.

The LSTM model with an extra-deep layer gives a quantitatively stronger performance on training but lacks generalizing ability. This is evidenced by the large validation loss and the unusual stability of the validation accuracy. This overfitting seen in the improved model reveals that the model is able to capture detailed information from the training data, but this cannot be generalized to unseen data. Regularization: Additional measures to improve the generalization performance of the enhanced model could be needed, may it be dropout, early stopping, or a more aggressive data augmentation. These techniques add constraints that help a model by training to prevent overfitting, which may in turn make a model more generalizable to new data. Figure 3. depicts the training and validation loss/accuracy of both models.

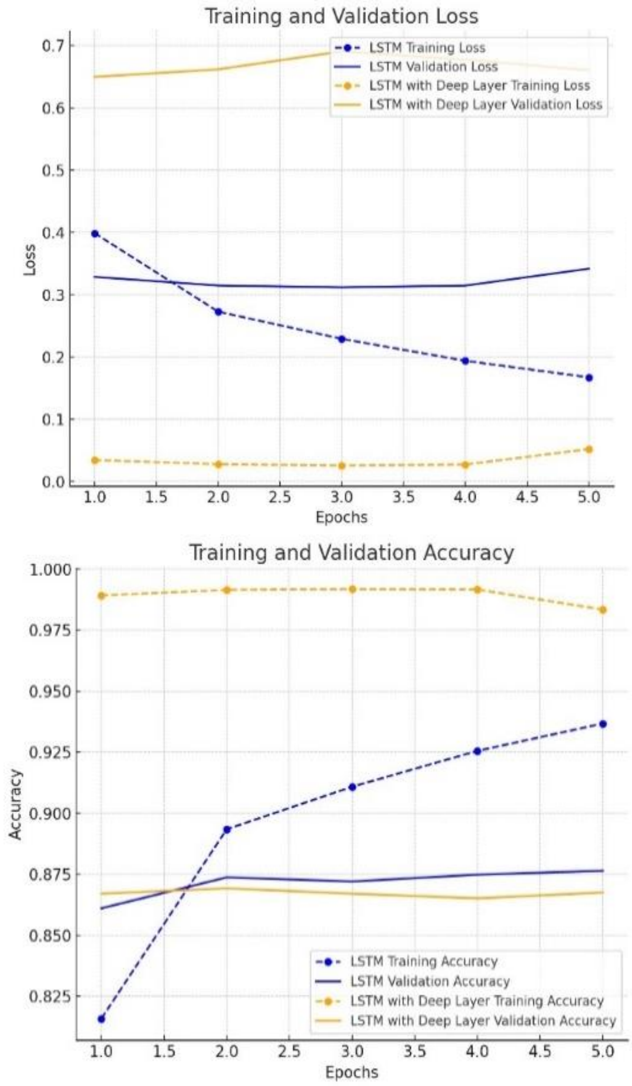

Figure 3. Training and validation loss and accuracy

4.3. Correlation plot analysis

In neural network training, it is essential to analyze the relationship between training and validation metrics to assess model performance and generalization capabilities. Correlation plots help visualize how training loss, validation loss, and accuracy metrics evolve over time, providing insights into model behavior. This comparison is particularly important when evaluating different architectures, such as a standard LSTM model versus an LSTM with an additional deep layer, to determine which model achieves better overall performance and generalizes more effectively on unseen data.

a. LSTM model correlation matrix
   – Training Loss vs. Validation Loss: There is a positive correlation between training and validation loss, which means that as the training loss decreases, the validation loss decreases as well. It means the model is generalizing well.
   – Training and validation accuracy: Training accuracy is in direct proportion to validation accuracy. Remind yourself that whatever is done to improve training accuracy (the horizontal axis) also tends to improve validation accuracy (the vertical axis), which is a sign of good generalization.
b. LSTM with deep layer: correlation matrix
   – Training loss vs. validation loss: Still indicates a high positive relationship, but possibly lower than the basic LSTM model. While the training loss diminishes considerably, the validation loss gets low (very high compared to the training loss), which may indicate that the training loss is overfitting, but the validation loss is not.
   – Training accuracy vs. validation accuracy: the correlation between training and validation accuracy is strong and has a slightly different behavior compared to the basic LSTM model. When the training accuracy is high and we have differences more like this, it suggests that the model is overfitting.

From the correlation plots, the variation in how much the models generalize from the training data to the unseen validation data can be seen. The simple LSTM model seems to have reasonable generalization, with stable behavior across the training and validation metrics. The LSTM model with a deep layer suffers from overfitting, which is when the model performance is amazing on train data, but when we validate this model, it is so mystique because the model just learns existing data without learning the pattern correlation in the data. With these lessons, more work can be done to improve the design of models and the regularization techniques to further improve generalization performance. Figure 4 shows the correlation plot analysis for both models. Table 3 compares the performances of LSTM with the deep layer model.

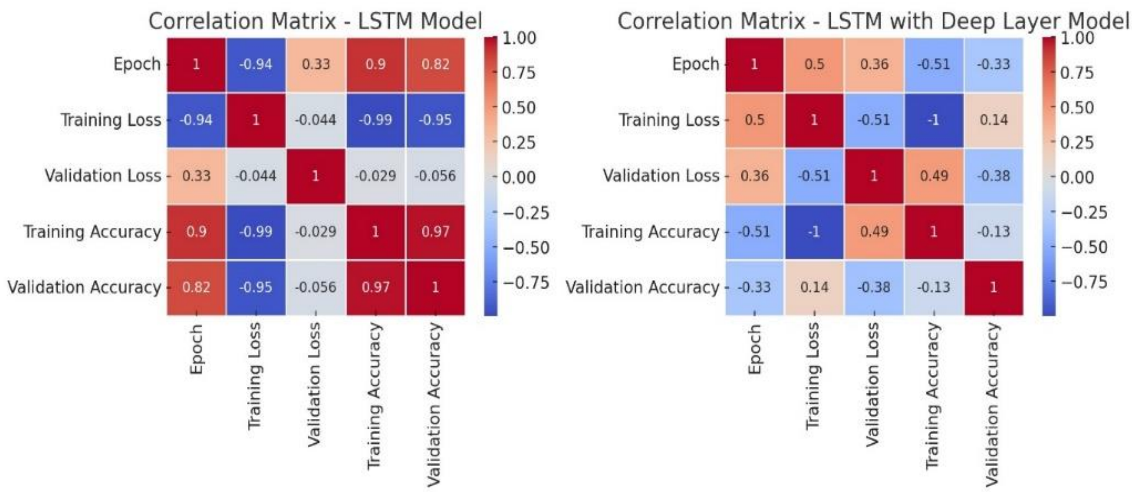

Figure 4. Correlation plot analysis-comparison

Table 3. Comparison of LSTM and LSTM with deep layer model

Metric	LSTM model	LSTM with deep layer
Training loss	0.1676	0.0524
Validation loss	0.3419	0.6604
Training accuracy	0.9367	0.9834
Validation accuracy	0.8764	0.8675
Precision	0.8	0.85
Recall	0.85	0.91
F1-score	0.8	0.86

5. CONCLUSION

This article looked at the sentiment analysis of adding a deep layer to an LSTM network. Our extensive evaluation, which ranged from data preprocessing to model evaluation, proved that the deep-layer LSTM model had better accuracy, recall, and F1-score compared to the basic model. The deep layer was trained to record complex sequences and significantly improved predictive accuracy. Nevertheless, the upgraded model overfitted, which called for even more regularization and hyperparameter tuning. However, a core level of enhanced sentiment analysis. In summary, the main contribution of this work is using a deep-layered LSTM network to increase the performance of the model. In future work, these issues, overfitting, and some effective model optimization will be applied to the high-score game for better generalization and accuracy. This study benefits deep learning models for natural language processing.

REFERENCES

[1] A. C. M. V. Srinivas, C. Satyanarayana, C. Divakar, and K. P. Sirisha, “Sentiment analysis using neural network and LSTM,” *IOP Conference Series: Materials Science and Engineering*, vol. 1074, no. 1, 2021, doi: 10.1088/1757-899x/1074/1/012007.

[2] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, “Sentiment analysis using Word2Vec and long short-term memory (LSTM) for Indonesian hotel reviews,” *Procedia Computer Science*, vol. 179, pp. 728–735, 2021, doi: 10.1016/j.procs.2021.01.061.

[3] U. B. Mahadevaswamy and P. Swathi, “Sentiment analysis using bidirectional LSTM network,” *Procedia Computer Science*, vol. 218, pp. 45–56, 2022, doi: 10.1016/j.procs.2022.12.400.

[4] U. D. Gandhi, P. M. Kumar, G. C. Babu, and G. Karthick, “Sentiment analysis on twitter data by using convolutional neural network (CNN) and long short-term memory (LSTM),” *Wireless Personal Communications*, 2021, doi: 10.1007/s11277-021-08580-3.

[5] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, “Co-LSTM: convolutional LSTM model for sentiment analysis in social big data,” *Information Processing and Management*, vol. 58, no. 1, 2021, doi: 10.1016/j.ipm.2020.102435.

[6] Z. Jin, Y. Yang, and Y. Liu, “Stock closing price prediction based on sentiment analysis and LSTM,” *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713–9729, 2020, doi: 10.1007/s00521-019-04504-2.

[7] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, “A survey on long short-term memory networks for time series prediction,” *Procedia CIRP*, vol. 99, pp. 650–655, 2021, doi: 10.1016/j.procir.2021.03.088.

[8] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” *Physica D: Nonlinear Phenomena*, vol. 404, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[9] M. Alizamir *et al.*, “Improving the accuracy of daily solar radiation prediction by climatic data using an efficient hybrid deep learning model: Long short-term memory (LSTM) network coupled with wavelet transform,” *Engineering Applications of Artificial Intelligence*, vol. 123, 2023, doi: 10.1016/j.engappai.2023.106199.

[10] Z. Gao *et al.*, “A novel multivariate time series prediction of crucial water quality parameters with long short-term memory (LSTM) networks,” *Journal of Contaminant Hydrology*, vol. 259, 2023, doi: 10.1016/j.jconhyd.2023.104262.

[11] M. Wen, M. S. Ibrahim, A. H. Meda, G. Zhang, and J. Fan, “In-Situ early anomaly detection and remaining useful lifetime prediction for high-power white LEDs with distance and entropy-based long short-term memory recurrent neural networks,” *Expert Systems with Applications*, vol. 238, 2024, doi: 10.1016/j.eswa.2023.121832.

[12] S. Chae, W. N. Street, N. Ramaraju, and S. Gilbertson-White, “Prediction of cancer symptom trajectory using longitudinal electronic health record data and long short-term memory neural network,” *JCO Clinical Cancer Informatics*, no. 8, 2024, doi: 10.1200/cci.23.00039.

[13] M. Liu, W. Zhou, and Z. Xu, “Hesitant fuzzy long short-term memory network and its application in the intelligent building selection,” *IEEE Transactions on Fuzzy Systems*, vol. 32, no. 5, pp. 2590–2602, 2024, doi: 10.1109/TFUZZ.2024.3355000.

[14] J. Shan, X. Zhang, Y. Liu, C. Zhang, and J. Zhou, “Deformation prediction of large-scale civil structures using spatiotemporal clustering and empirical mode decomposition-based long short-term memory network,” *Automation in Construction*, vol. 158, 2024, doi: 10.1016/j.autcon.2023.105222.

[15] A. Gogineni, M. K. D. Rout, and K. Shubham, “Evaluating machine learning algorithms for predicting compressive strength of concrete with mineral admixture using long short-term memory (LSTM) Technique,” *Asian Journal of Civil Engineering*, vol. 25, no. 2, pp. 1921–1933, 2024, doi: 10.1007/s42107-023-00885-x.

[16] B. Ghojogh and A. Ghodsi, “Recurrent neural networks and long short-term memory networks: tutorial and survey,” *arXiv:2304.11461*, 2023.

[17] F. Liu, J. Li, and L. Wang, “PI-LSTM: physics-informed long short-term memory network for structural response modeling,” *Engineering Structures*, vol. 292, 2023, doi: 10.1016/j.engstruct.2023.116500.

[18] S. H. Park *et al.*, “Development of a soil moisture prediction model based on recurrent neural network long short-term memory (RNN-LSTM) in soybean cultivation,” *Sensors*, vol. 23, no. 4, 2023, doi: 10.3390/s23041976.

[19] C. Avci, B. Tekinerdogan, and C. Catal, “Analyzing the performance of long short-term memory architectures for malware detection models,” *Concurrency and Computation: Practice and Experience*, vol. 35, no. 6, 2023, doi: 10.1002/cpe.7581.

[20] S. Baniasadi, R. Salehi, S. Soltani, D. Martín, P. Pourmand, and E. Ghafourian, “Optimizing long short-term memory network for air pollution prediction using a novel binary chimp optimization algorithm,” *Electronics (Switzerland)*, vol. 12, no. 18, 2023, doi: 10.3390/electronics12183985.

[21] L. W. Tsai and A. Alipour, “Physics-informed long short-term memory networks for response prediction of a wind-excited flexible structure,” *Engineering Structures*, vol. 275, 2023, doi: 10.1016/j.engstruct.2022.114968.

[22] R. Ghosh, S. Phadikar, N. Deb, N. Sinha, P. Das, and E. Ghaderpour, “Automatic eyeblink and muscular artifact detection and removal from EEG signals using k-nearest neighbor classifier and long short-term memory networks,” *IEEE Sensors Journal*, vol. 23, no. 5, pp. 5422–5436, 2023, doi: 10.1109/JSEN.2023.3237383.

[23] H. H. Zhang, H. M. Yao, L. Jiang, and M. Ng, “Deep long short-term memory networks-based solving method for the FDTD method: 2-D case,” *IEEE Microwave and Wireless Technology Letters*, vol. 33, no. 5, pp. 499–502, 2023, doi: 10.1109/lmwt.2022.3223959.

[24] T. Li and T. Wu, “Modeling nonlinear flutter behavior of long-span bridges using knowledge-enhanced long short-term memory network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 11, pp. 1504–1519, 2023, doi: 10.1111/mice.13000.

[25] A. I. Jony and A. K. B. Arnob, “A long short-term memory based approach for detecting cyber attacks in IoT using CIC-IoT2023 dataset,” *Journal of Edge Computing*, vol. 3, no. 1, pp. 28–42, 2024, doi: 10.55056/jec.648.

BIOGRAPHIES OF AUTHORS

Parul Dubey is currently working as an assistant professor in Department of Computer Science and Engineering, Symbiosis Institute of Technology, Nagpur Campus, Symbiosis International (Deemed University), Pune, India. Authors highest degree is the degree of master's in computer engineering. She is also a Ph.D. Research scholar in Computer Science and Engineering at C.V. Raman University, Bilaspur, Chhattisgarh, India. She has good academic and research experience in various areas of CSE and IT. She has teaching experience of five years. She is indulged in research activities which includes book chapters, books, conference papers and journal articles as well. She has 17 Indian published patents. She holds around 15 publications which are part of conferences, Scopus and other journals as well. She can be contacted on email: dubeyparul29@gmail.com.

Pushkar Dubey is currently working as assistant professor and head in Department of Management at Pandit Sundarlal Sharma (Open) University Chhattisgarh Bilaspur. He is a gold medalist in Master of Business Administration (MBA) and Ph.D. in human resource management. He has published more than 70 research papers in reputed journals such as Emerald, Taylor and Francis, Springer. His 12 articles are indexed in Scopus. He has also attended several international conferences including IIM level conferences. Having successfully guided 6 Ph.D. scholars, Dr. Dubey is also accredited with 07 published patents. He has also accomplished 05 research projects including 03 sponsored by Indian Council of Social Science Research (ICSSR) New Delhi. Having specialized in statistical softwares for data analysis, he has delivered several lectures on SPSS, AMOS and others. He is also pursuing the highest academic degree for research i.e., doctor of letters in the area of application of Shrimad Bhagwad Geeta into management practices. He can be contacted on email: drdubeypkag@gmail.com.

Hitesh Gehani (Ph.D., electronics engineering), assistant professor at School of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Ramdeobaba University, Nagpur, India is a renowned academic personality and has achieved continued success in his role as a scholastic intellectual leader in diversified areas. He holds a patentable profile with brilliant academic achievements. His tutorial excellence, especially the various innovative teaching/learning practices adopted by him marks him as a distinguished luminary in academic circles. Recipient of Best Teacher Award, NPTEL Star and Ph.D. Persuing. Hitesh Gehani has 2 patents (Published), 1 copyright, 3 book chapters, 6 research papers (published in international journals, conferences). He has worked as a reviewer on many international journals and conferences. He can be contacted at email: gehanihs@rknec.edu.