A Hybrid RNN based Deep Learning Approach for Text Classification

Pramod Sunagar, Anita Kanavalli
Department of Computer Science and Engineering
M S Ramaiah Institute of Technology (Affiliated to VTU)
Bangalore, India

***Abstract*—Despite the fact that text classification has grown in relevance over the last decade, there are a plethora of approaches that have been created to meet the difficulties related with text classification. To handle the complexities involved in the text classification process, the focus has shifted away from traditional machine learning methods and toward neural networks. In this work the traditional RNN model is embedded with different layers to test the accuracy of the text classification. The work involves the implementation of RNN+LSTM+GRU model. This model is compared with RCNN+LSTM and RNN+GRU. The model is trained by using the GloVe dataset. The accuracy and recall are obtained from the models is assessed. The F1 score is used to compare the performance of both models. The hybrid RNN model has three LSTM layers and two GRU layers, whereas the RCNN model contains four convolution layers and four LSTM levels, and the RNN model contains four GRU layers. The weighted average for the hybrid RNN model is found to be 0.74, RCNN+LSTM is 0.69 and RNN+GRU is 0.77. RNN+LSTM+GRU model shows moderate accuracy in the initial epochs but slowly the accuracy increases as and when the epochs are increased.**

Keywords—F1 score; gated recurrent unit; GloVe; long - short term memory; precision; recall; recurrent neural network; region-based convolutional neural network; text classification

I. Introduction

Text classification has posed a necessity in the current generation, which is precisely due to the fact that the data being handled is increasing in volume at an alarming rate [1]. This can be attributed to the increase in the number of end-users, thereby necessitating effective data handling. Effective handling also involves uploading and retrieval of data at least possible time. The data being uploaded and retrieved may be in the context of many real-time applications like web applications, banking servers, scientific literature, or digital libraries of documents. Some of the applications also involve data filtering [2] and organization, where voluminous data is sorted and categorized as per the relevance [3]. Also, apart from data organization, opinion mining is an application of utmost importance. Therefore, efforts have been put-forth to extend the classified data for opinion mining [4]. Lastly, e-mail classification is also an application of great significance, where text classification is used to identify spam e-mails [5-7]. The applications, as mentioned earlier, also include challenges that are to be addressed critically and with utmost precision.

Researchers have made few significant efforts toward addressing real-world problems in the recent past [8-14]. Most of the applications concentrate on Natural Language Processing (NLP) and text analytics, with enormous efforts to efficiently handle the data. The extension of efforts also aims at text classification more effectively and intricately. In general, post receipt of the raw data, text classification can be executed stage-wise, viz. feature extraction, reduction of dimensions of the data, selection of the classifier, and finally, the metrics that facilitate quantifying the accuracy of classification. Although many models have been implemented to improve the text classifications, still there are lot of challenges persists. The proposed model tries to improve the text classification by creating a hybrid models and utilizes the advantages of RNN, LSTM and GRU models. Transfer learning is a strategy that involves developing a model for one problem and then utilizing it to train another related problem [37]. Using transfer learning the accuracy of the text classification tasks can be improved [38].

II. Related Works

The authors have used a BBC news text categorization structure in this work and implemented ML algorithms like logistic regression, KNN algorithms [15] and random forest. These methods are evaluated using measures such as accuracy, precision, F1-score, confusion matrix, and support. The logistic regression algorithm has better accuracy than other algorithms in classifying the text in the given dataset. In this work, the authors have performed a comparative analysis of deep learning models on Arabic text for the single and multi-label text classification [16]. Authors have demonstrated that the pre-processing stage is not required when using the suggested models. The word2vec embedding method is included to enhance the accuracy of the deep learning models. In this work, the authors have projected a model which utilizes a convolutional layer, Bi-LSTM, and attention mechanism to understand the semantics and improve the text classification accuracy [17]. The authors have analyzed the traditional deep learning models and proved that the proposed model has the highest accuracy over others. The feature extraction methods and developing classifiers are significant for the text classification techniques. In this work, the authors have highlighted the improved word embeddings with machine learning models for automatic document classification jobs [18]. The authors have used the word embedding techniques such as word2vec, Glove, and fastText. The authors have used the freely available dataset and implemented algorithms such as SVM, XGBoost, and CNN to use hierarchical and flat measures. The fastText embedding technique has proved to improve the classification accuracy of the text. The authors have investigated the hierarchical multi-label text classification task [19]. Since the documents are stored in a hierarchical structure, the classification task becomes tough. The authors have proposed a new framework called Hierarchical Attention-based Recurrent Neural Network (HARNN) for categorizing documents into the appropriate labels by integrating texts and the hierarchical category structure.

The authors examined text categorization techniques based on machine learning [20]. AG's News Topic Classification Dataset, which comprises 120000 training and 7600 testing samples, was used in this case. Support vector machine, Rocchio, bagging, boosting, naive bayes, and KNN are examples of commonly used machine learning algorithms that are implemented and assessed using accuracy, precision, F1 score, and recall. According to the authors, the SVM technique outperformed all other algorithms tested for this job. The authors classified tweets regarding Covid-19 pandemic using DL techniques [21]. They have investigated three traditional deep learning algorithms, CNN, RNN, and RCNN, and two hybrid algorithms, RNN+LSTM and RNN+Bi-LSTM with Attention. When coupled with GloVe and Word2Vec, the RNN+Bi-LSTM with Attention mechanism correctly classified tweets. The time taken to train and predict the accurate labels is more for RNN+LSTM when compared to other deep learning models.

The authors provide a different feature selection approach called Multivariate Relative Discrimination Criterion (MRDC) in this study to minimize dimensionality and feature space in order to enhance text classification performance [22]. The suggested technique focuses on reducing duplicate features by employing the notions of minimal redundancy and maximal relevancy. The suggested approach considers document frequency for each word while assessing its usefulness for that purpose. The suggested technique picks the characteristics with the highest relevance and considers the redundancy between them using a correlation metric. The authors discussed a deep learning strategy called HDLTex, which integrates various deep learning algorithms to create hierarchical classifications [23].

The authors' proposed architecture uses a mix of RNN at the top level and DNN or CNN at the bottom level to classify articles more accurately than standard SVM or Naive Bayes. The deep learning models were optimized using RMSProp and Adam to improve accuracy. The DNN has eight hidden layers. RNN was built using LSTM and GRU, and CNN with eight hidden layers. The authors have presented a comprehensive overview of the 150 classifiers introduced in recent years to accomplish this study's text categorization problems [24]. The work comprises 40 widely used datasets for text categorization. To improve the precision of text categorization, the authors proposed combining transformers and pre-trained language models with deep learning techniques. The work also establishes the grouping of 150 DL models into ten broad categories: feed-forward networks, CNN-based models, RNN-based models, Graph neural networks, hybrid models, etc.

III. METHODOLOGY

Based on earlier efforts, it is clear that every neural network architecture has shortcomings, which can be addressed by combining other architectures to compensate for the deficiencies. Therefore, embedding layers of other architectures to the existing neural network model and their contribution to the model's performance has been studied in the present work. For example, the Long Short Term Memory (LSTM) layer [25-27] (Fig. 1) has the ability to remember long-distance relations when compared to Gated Recurrent Unit (GRU) [28-29] (Fig. 2).

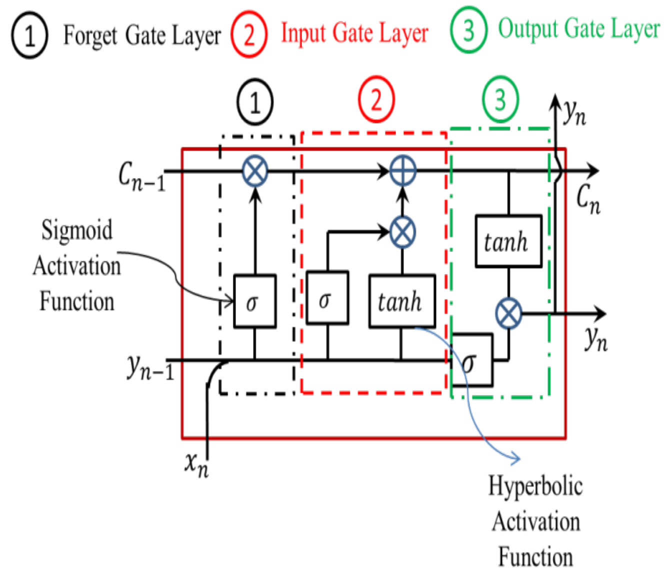

Fig. 1. Long Short Term Memory Units.

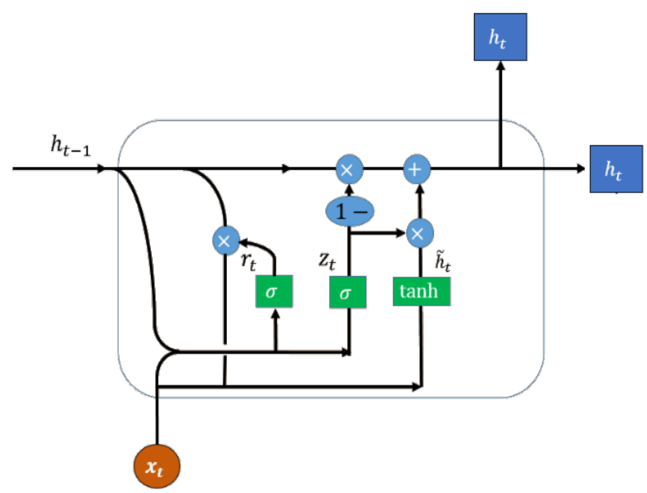

Fig. 2. Gated Recurrent Units.

The sigmoid function is used in both LSTM and GRU units, whereas the hyperbolic activation function in the output layer facilitates data retrieval even after a considerable amount of time. On the other hand, GRU is easy to train compared to LSTM, with a lesser number of data and a better performance than LSTMs. This motivates researchers to embed GRUs against LSTM when retrieval is not of great significance.

Three models are considered for the present study namely Recurrent Neural Network (RNN) [30-31] with GRU unit, RNN with LSTM as well as GRU units and Region-Based Convolutional Neural Network (RCNN) with LSTM layer. A conventional RCNN [32] model is modified by adding four LSTM layers along with the convolutional layers as indicated in Fig. 3, the performance of which is compared with two models comprising a typical RNN model with four GRU layers (Fig. 4) and RNN model with 3 LSTM layers and 2 GRU layers (Fig. 5). The LSTM layer used in RCNN and RNN architectures is a forgotten gate type and an input and output gate, as shown in Fig. 1.

The output gate layer is the most significant layer that enables long-term dependencies handling. The forget gate layer assigns a value based on the input vector of the current cell, the output vector of the previous and the previous cell state. This is also indicative that the forget layer carries out the decision of allowing the value to the input layer. The sigmoid neural function with a point-wise multiplication operator is used to generate values in the LSTM's forget gate layer. The Input gate layer is responsible for two tasks. The entering vector data is first updated using a sigmoid activation function, and the value created by the sigmoid activation function is then compared to the hyperbolic activation function. The new values obtained as a result of the comparison are combined with the prior cell state. The Output Gate Layer compares the input vector generated by the sigmoid activation function to the updated cell state generated by the hyperbolic activation function.

Whereas, irrespective of LSTM, a fully gated GRU comprises only two gates, viz. forget gate layer and an input gate. Though GRU is around half a decade old, it is preferred in specific, precise circumstances because they need a considerably smaller dataset and time for training the model. It can be observed that LSTM has a separate update gate and forget gate, rendering it more sophisticated. Therefore, the complexity of the LSTM paves the way for the usage of GRU, wherein the control on the model embedded with GRU units is better. Based on the earlier observations, three models are considered with LSTM, GRU, and LSTM-GRU units embedded, respectively, and their performances are evaluated. The training of the models is facilitated by the GloVE dataset [14], which can effectively capture syntactic and semantic representations of the words. The shortcoming of GloVE [33] is its inability to capture out-of-vocabulary words, which demands a considerable corpus to train the model, thereby eventually increasing the memory requirement. Though GloVE is similar to Word2Vec [34-35] in its operation, the weights associated with frequent word pairs will not pre-occupy the training process. The merits mentioned above support the usage of the GloVE dataset for training the models considered for the study.

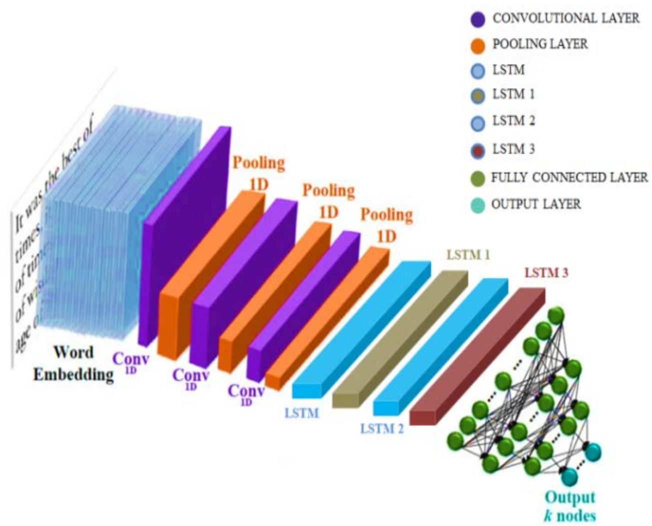

Fig. 3. RCNN Model with LSTM Layers.

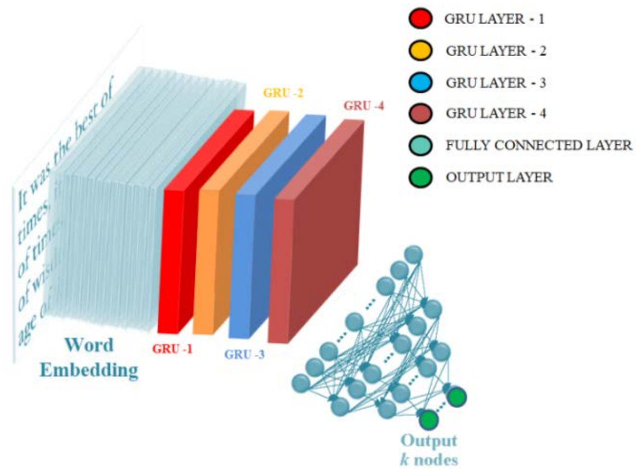

Fig. 4. RNN Model with GRU Layers.

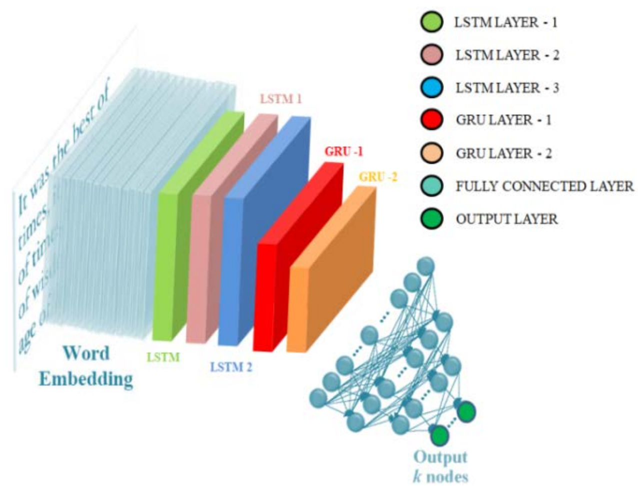

Fig. 5. RNN Model with LSTM and GRU Layers.

IV. Evaluation

The evaluation of the models is carried out with the help of two parameters, viz. precision, and recall. Precision is the ratio of true prediction to a total number of predictions (Equation 1), while recall is the ratio of true positive to the sum of true positive and false negative. Though precision highlights the accuracy of the model, recall indicates sensitivity. An increase in precision reduces the number of false positives, increasing false negatives. Therefore increase in false negative decreases the recall value. It is undoubtedly a balance between precision and recall, which renders a valuable model for a given application as they demonstrate an inverse behavior. The cumulative effect of precision and recall are captured using the F1 score, which can be obtained by evaluating the area under the precision and recall curve plotted for both models. F1 score is the harmonic mean calculated from precision and recall values (eq. 3), which forms a significant metric to evaluate the performance of the models.

$$Precision\ (P) = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall\ (R) = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F_1 = 2\left[\frac{P\ X\ R}{P + R}\right] \quad (3)$$

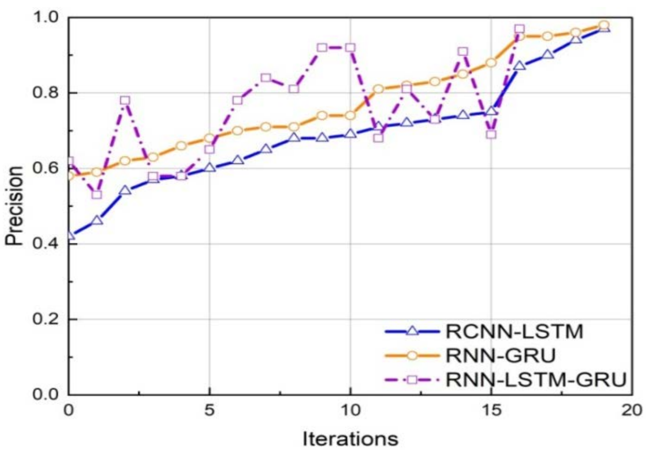

Fig. 6. Comparison of Precision for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

The results are plotted for the precision and recall as shown in Fig. 6 and Fig. 7, respectively. It is observed that the RNN model consistently outperforms the RCNN model. After the 18th iteration, hybrid RNN and RCNN provide almost the same precision and recall values. Hybrid RNN is embedded with GRU layers that can be trained for a lesser number of data and for a lesser time to generate a greater accuracy, as indicated in Fig. 6 and Fig. 7.

The recall precision curve is of great significance for assessing the performance of the RCNN-LSTM, RNN-GRU and RNN-LSTM-GRU model as shown in the Fig. 8, 9 and 10. The results obtained are plotted, for which the polynomial curve is fitted by using the least-squares method. The area under the curve (AUC) [36] indicates the F1 score for each model, thereby depicting the optimum blend of precision and recall of the model. The RCNN-LSTM model occupies a larger area when compared to RNN-GRU and RNN-LSTM-GRU models, indicating a more extensive range of recall-precision values.

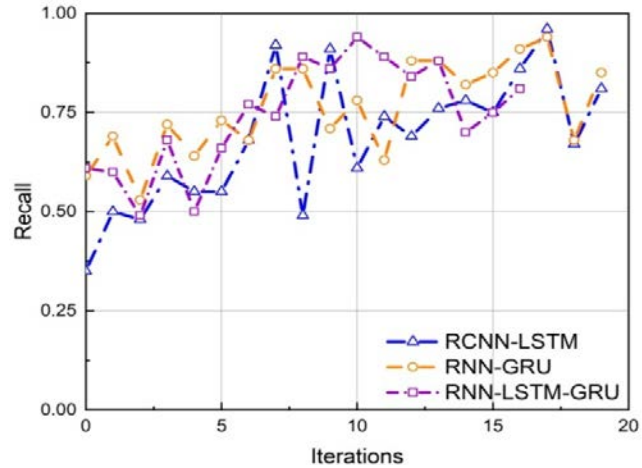

Fig. 7. Comparison of Recall for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

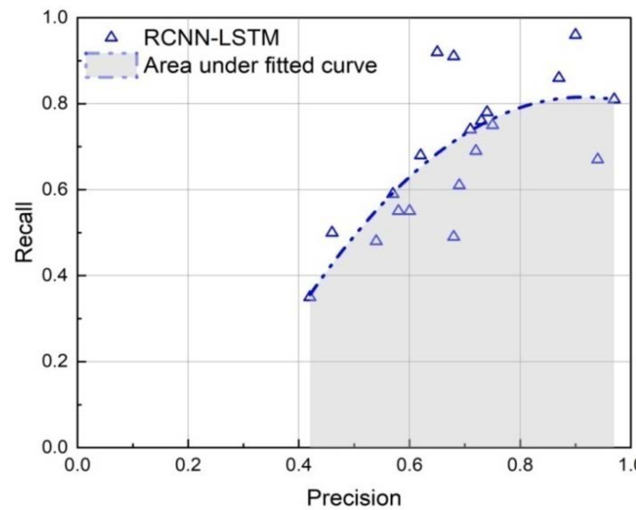

Fig. 8. Variation of Precision with Recall for RCNN-LSTM Model.

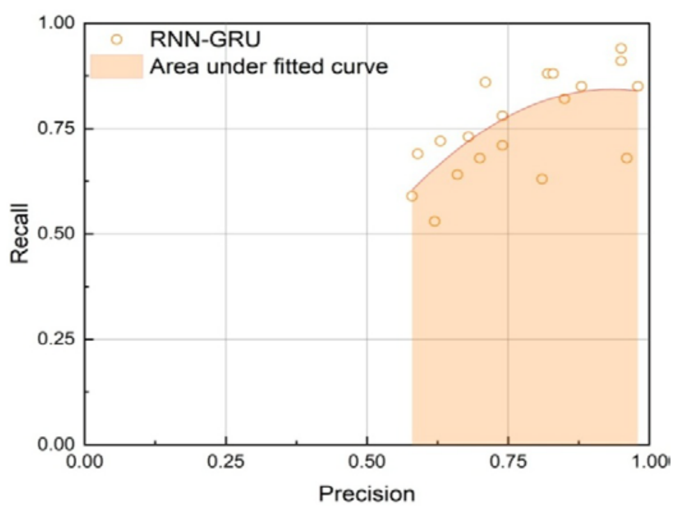

Fig. 9. Variation of Precision with Recall for RNN-GRU Model.

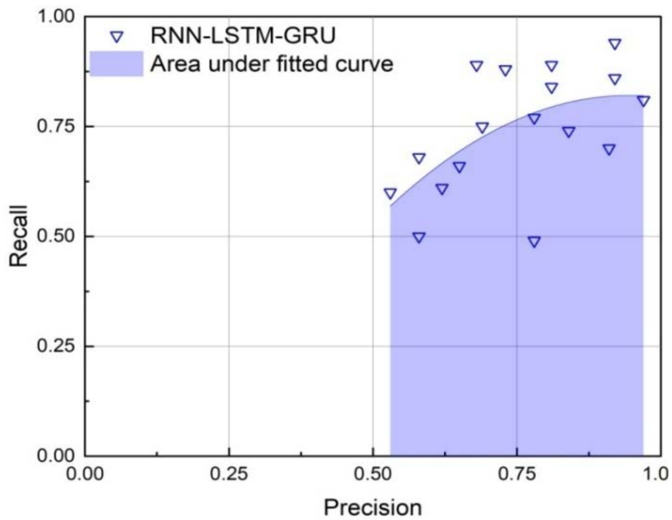

Fig. 10. Variation of Precision with Recall for RNN-LSTM-GRU Model.

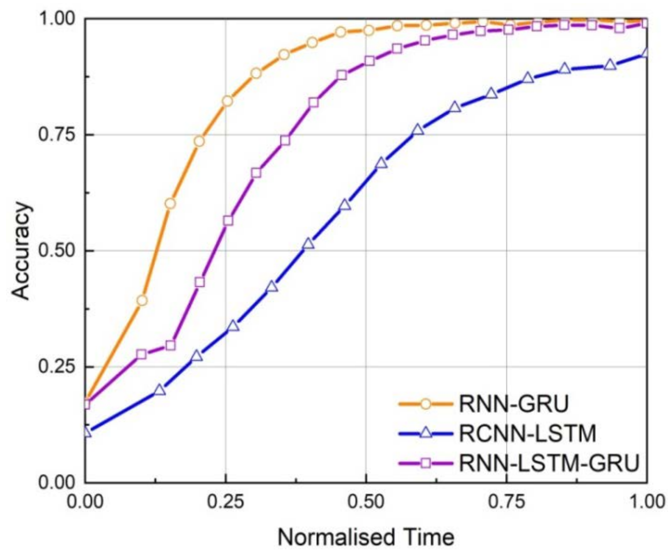

Fig. 11. Comparison of Accuracy for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

The comparison can be more clearly observed in Fig. 13, and the corresponding variations in the F1 score are indicated in Fig. 14. The average F1 score for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU models are indicated in Table I. It can be observed that the RNN-GRU model outperforms RCNN and RNN-LSTM-GRU model by a margin of 10% and 4%, respectively. Since the GRU layers embedded in the RNN model require lesser time and smaller datasets for training, the F1 curve of the RNN model consistently dominates over RCNN and RNN-LSTM-GRU models. RNN-GRU model is embedded with the GRU layer, which cannot handle long-term dependencies. When long-term dependencies are of great significance, as is the case in text classification, RNN-LSTM-GRU is preferred the most, which carries the merits of both LSTM and GRU layers. RCNN has LSTM layers, which are capable of handling long-term dependencies; they take more significant time and more extensive data to get trained, which renders RNN-LSTM-GRU more suitable for text classification.

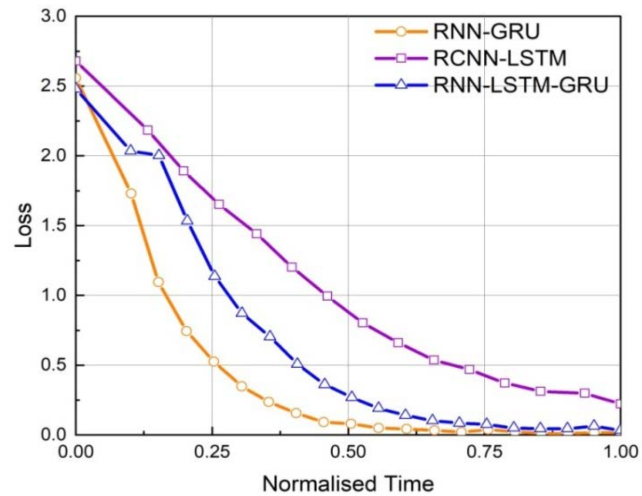

Fig. 12. Variation of Loss in values for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

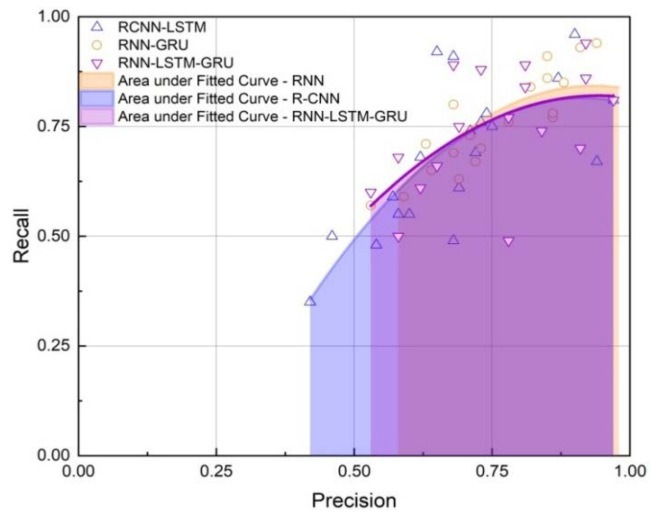

Fig. 13. Comparison of Precision-Recall for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

Also, Fig. 11 and Fig. 12 indicate the variation of accuracy with the normalized time. It can be observed that the RNN-GRU model and RNN-LSTM-GRU model perform almost the same after a normalized time of 0.75s. The accuracy of both the models is nearly the same, which is indicative that the RNN-LSTM-GRU can replace the RNN-GRU model to enable the retrieval of long-term dependencies. The difference in slopes over the initial normalized time from 0 to 0.6s is because of the presence of the LSTM layer in the RNN-LSTM-GRU model, which requires more time for training, while the GRU layer compensates for the initial delay by matching the slope when trained beyond a normalized time of 0.75s.

The area under the precision-recall curve gives the F1 score (Fig. 13). The variation of the F1 score for all three models is depicted in Fig. 14. Though the fluctuation of RNN-LSTM-GRU is considerably significant compared to RNN-GRU and RCNN-LSTM, the average F1 value is more significant than RCNN-LSTM and marginally less than RNN-GRU. The values are listed in Table I.

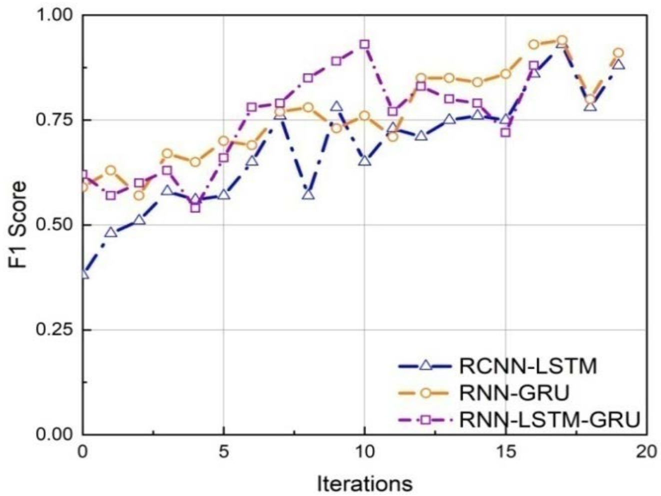

Fig. 14. Variation of F1 Score with Epoch for RCNN-LSTM, RNN-GRU, and RNN-LSTM-GRU Models.

TABLE I. COMPARISON OF PRECISION, RECALL, AND F1 SCORE

Model	Average Precision	Average Recall	F1 Score
RCNN-LSTM	0.691	0.6825	0.682
RNN-GRU	0.7695	0.7615	0.7615
RNN-LSTM-GRU	0.7395	0.7163	0.7226

V. CONCLUSION

The present comprehensive study intends to address three text classification strategies by modifying the models and evaluating the performances in connect with precision and sensitivity. It is observed that the model selection for a given application is a trade-off among the variables such as time, dataset, and handling of the long-term dependencies, which define a suitable model for the application. Hence, RCNN with more accurate when compared to RNN with an ability to remember the dependencies, whereas RNN embedded with GRU has a shorter learning time which a smaller volume of the dataset can train. The average F1 score of RNN (4 layer model) is 0.77, whereas, for RCNN (8 layer model), the average F1 score is 0.69, which is around 10% less than RNN.

ACKNOWLEDGMENT

This work was supported by M S Ramaiah Institute of Technology, Bangalore-560054, and Visvesvaraya Technological University, Jnana Sangama, Belagavi-590018.

REFERENCES

[1] Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. Information, 10(4), 150.

[2] Lang, K. (1995). Newsweeder: Learning to filter netnews. In Machine Learning Proceedings 1995 (pp. 331-339). Morgan Kaufmann.

[3] Chakrabarti, S., Dom, B., Agrawal, R., & Raghavan, P. (1997, August). Using taxonomy, discriminants, and signatures for navigating in text databases. In VLDB (Vol. 97, pp. 446-455).

[4] Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In Mining text data (pp. 415-463). Springer, Boston, MA.

[5] Lewis, D. D., & Knowles, K. A. (1997). Threading electronic mail: A preliminary study. Information processing & management, 33(2), 209-217.

[6] Sharma, P., & Bhardwaj, U. Machine learning based spam e-mail detection. International Journal of Intelligent Engineering and Systems, 11(3), 1-10, 2018.

[7] Cohen, W. W. (1996, March). Learning rules that classify e-mail. In AAAI spring symposium on machine learning in information access (Vol. 18, p. 25).

[8] Carvalho, V. R., & Cohen, W. W. (2005, August). On the collective classification of email" speech acts". In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 345-352).

[9] Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., & Guan, R. (2018). Text classification based on deep belief network and softmax regression. Neural Computing and Applications, 29(1), 61-70.

[10] Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In Mining text data (pp. 163-222). Springer, Boston, MA.

[11] Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2018, April). Rmdl: Random multimodel deep learning for classification. In Proceedings of the 2nd International Conference on Information System and Data Mining (pp. 19-28).

[12] McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

[13] Heidarysafa, M., Kowsari, K., Brown, D. E., Meimandi, K. J., & Barnes, L. E. (2018). An improvement of data classification using random multimodel deep learning (rmdl). arXiv preprint arXiv:1808.08121.

[14] Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February). Recurrent convolutional neural networks for text classification. In Twenty-ninth AAAI conference on artificial intelligence.

[15] Altınel, B., & Ganiz, M. C. (2018). Semantic text classification: A survey of past and recent advances. Information Processing & Management, 54(6), 1129-1153.

[16] Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research, 5(1), 1-16.

[17] Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. Information Processing & Management, 57(1), 102121.

[18] Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing, 337, 325-338.

[19] Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. Information Sciences, 471, 216-232.

[20] Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., Liu, Y., ... & Wang, S. (2019, November). Hierarchical multi-label text classification: An attention-based recurrent network approach. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (pp. 1051-1060).

[21] Sunagar, P., Kanavalli, A., Nayak, S. S., Mahan, S. R., Prasad, S., & Prasad, S. (2021). News Topic Classification Using Machine Learning Techniques. In International Conference on Communication, Computing and Electronics Systems: Proceedings of ICCCES 2020 (Vol. 733, p. 461). Springer Nature.

[22] Sunagar, P., Kanavalli, A., Poornima, V., Hemanth, V. M., Sreeram, K., & Shivakumar, K. S. (2021). Classification of Covid-19 Tweets Using Deep Learning Techniques. In Inventive Systems and Control (pp. 123-136). Springer, Singapore.

[23] Labani, M., Moradi, P., Ahmadizar, F., & Jalili, M. (2018). A novel multivariate filter method for feature selection in text classification problems. Engineering Applications of Artificial Intelligence, 70, 25-37.

[24] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning--based Text Classification: A Comprehensive Review. ACM Computing Surveys (CSUR), 54(3), 1-40.

[25] Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017, December). Hdltex: Hierarchical deep learning for text classification. In 2017 16th IEEE international conference on machine learning and applications (ICMLA) (pp. 364-371). IEEE.

[26] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[27] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), 602-610.

[28] Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In International conference on machine learning (pp. 1310-1318). PMLR.

[29] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

[30] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[31] Sutskever, I., Martens, J., & Hinton, G. E. (2011, January). Generating text with recurrent neural networks. In ICML.

[32] Mandic, D., & Chambers, J. (2001). Recurrent neural networks for prediction: learning algorithms, architectures and stability. Wiley.

[33] Wang, B., Xu, J., Li, J., Hu, C., & Pan, J. S. (2017, June). Scene text recognition algorithm based on faster RCNN. In 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS) (pp. 1-4). IEEE.

[34] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

[35] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[36] Pencina, M. J., D'Agostino Sr, R. B., D'Agostino Jr, R. B., & Vasan, R. S. (2008). Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. Statistics in medicine, 27(2), 157-172.

[37] Shreyashree, S., Sunagar, P., Rajarajeswari, S., & Kanavalli, A. (2022). A Literature Review on Bidirectional Encoder Representations from Transformers. Inventive Computation and Information Technologies, 305-320.

[38] Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. Journal of healthcare engineering, 2022.