Sentiment Analysis With Lipschitz Recurrent Neural Networks Based Generative Adversarial Networks

Mahmudul Hasan
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, Virginia, USA
mhasa005@odu.edu

Sachin Shetty
Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, Virginia, USA
sshetty@odu.edu

***Abstract*—This paper applies a novel sentiment analysis method to the IMDB movie review dataset by integrating Lipschitz Recurrent Neural Networks (LRNN) into a Generative Adversarial Network (GAN) architecture. Traditional sentiment analysis techniques frequently struggle with instability and disappearing gradients during model training. Our method reduces these problems by using LRNN's stability and effectiveness, improving sentiment analysis's precision and resilience. The GAN framework efficiently synthesizes and discriminates between real and generated sentiment-laden textual data. It consists of a generator and discriminator that are both equipped with LRNN. The discriminator distinguishes between real and fake data and rates the sentiment correctness of the generated text, while the generator concentrates on producing realistic, sentiment-laden prose. A typical issue in traditional GAN models, the gradient vanishing problem is addressed by integrating LRNN in both the generator and discriminator, stabilizing the training process and enhancing performance. With an accuracy of 91.30 % on the IMDB dataset, our results significantly improve sentiment analysis accuracy, surpassing some well-known models. This indicates that LRNN-based GAN models can effectively handle challenging sentiment analysis tasks. Subsequent investigations will delve into the utilization of this paradigm in industry-specific settings like healthcare, banking, and education, as well as its incorporation into chatbot platforms. This work advances sentiment analysis techniques by providing a reliable and effective method for analyzing sentiments in big datasets.**

***Index Terms*—Lipschitz Recurrent Neural Networks (LRNN), Vanishing Gradient, Natural Language Processing (NLP), Generative Adversarial Networks (GAN)**

I. Introduction

The demand for online social media is increasing day by day, where people give their feedback easily [1]. These online social media are widely used by consumers, where they can express their sentiments or feedback about any goods and services after purchasing. The ultimate goal of sentiment analysis is to analyze the public reviews they leave online, especially on social media. Sentiment analysis is one kind of data mining process [2]. For natural language processing, the data mining process is essential. Because to summarise the text, to classify the text, for clustering the text, data mining plays a vital role. In addition, a data mining or text mining process is required to determine the positive and negative sentiment.

The sentiment analysis process not only helps the consumers but also helps the sellers or owners. Because every reputed company has a customer support management team and this team, and this team analyzes the customer feedback from the online [3]. Based on the feedback of customers, the companies can improve the quality of the goods.

In addition, movie lovers give feedback after watching any movie, which helps producers and directors improve the quality of their future movies and helps promote the movie. Moreover, movie reviews help the view to classify good movies and bad movies, especially the parents.

Similarly, people give feedback about healthcare, educational institutions, and other social services, and this helps them choose the right things when needed.

From the above discussion, it is easy to realize that public opinion is very important for business, education, health services, etc. The main question is how accurately the text/sentiment can be analyzed from the online platform. WordNet lexical database [4] is one of the methods to analyze the sentiment from the text.

The machine learning method is also famous for analyzing sentiments. Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) are widely used in sentiment analysis. Generative Adversarial Networks (GAN) are used to analyze the sentiment [1].

In this paper, we will discuss the Generative Adversarial Networks (GAN) [5] based sentiment analysis model where the famous Lipschitz Recurrent Neural Networks (LRNN) [6] is used as both generator and discriminator. In LRNN [6], the hidden-to-hidden matrices are used, which helps to decrease the vanishing gradient problem, which is very common in GAN, and helps to increase the stability of GAN, which helps to get good accuracy. In addition, for the LRNN, we make the discriminator serve a dual role: one is to distinguish between the real and fake text, and another is to analyze the sentiment, in other words, to analyze whether the sentiment is positive or negative.

II. Related works

Researchers are using different methods for sentiment analysis. Kurniasari and Setyanto [7] showed good accuracy by using their word2voc method. A Long Short-Term Memory (LSTM) method was proposed by Wang et al. [8]. This method is also attention-based. Can et al. [9] proposed an RNN-based model sentiment analysis model. They used a machine translation approach.

A BiLSTM-based sentiment analysis model was proposed by Xu et al. [10]. [10] proposed a Convolutional Neural Network (CNN) based sentiment analysis method, which achieved good accuracy. In addition, Usman et al. [11] proposed a Convolutional Neural Network (CNN) based sentiment analysis model.

Moreover, Pal et al. [12], Baliyan et al. [13], Nistor et al. [14], Ni et al. [15] invented the LSTM model to analyze the sentiment. And they have good accuracy.

Agarwal et al. [16] proposed a multimodal sentiment analysis via RNN variants, which can classify sentiment using text, audio, and video. Basiri et al. [17] present An Attention-based Bidirectional CNN-RNN Deep Model for sentiment analysis. They showed the most common and essential task of sentiment analysis. Their experiments were conducted on five reviews and three Twitter datasets. They built a new deep architecture for sentiment analysis.

A Hybrid CNN-LSTM-based model was proposed by Rehman et al. [18] to improve the accuracy of Movie Reviews Sentiment Analysis. Their approach achieved competitive results using state-of-the-art techniques on the IMDB movie review dataset and Amazon movie reviews dataset. Naseem et al. [19] proposed a transformer-based Deep intelligent contextual embedding for Twitter sentiment analysis. Takamura et al. [20] proposed latent variable models for semantic orientations of phrases, which achieved 82%

On the other hand, the BiLSTM-based sentiment analysis models have some drawbacks, and one of the problems is that long sequences are challenging for them to follow, which could result in information loss. Longer training times and increased resource needs result from their computational complexity. Even with advancements, they could still run into problems like overfitting and the vanishing gradient problem, mainly when insufficient training data exists.

In addition, the vanishing gradient problem is a significant barrier to text-based sentiment analysis employing RNN variations, affecting the model's capacity to learn from longer text sequences and context. RNNs can also be inefficient and scalable in real-world applications due to overfitting and computational demands, particularly when handling massive text datasets.

III. Materials and Methods

A. Dataset

We used the IMDB movie review dataset for this project, which contains 100,000 reviews, whereas the test data set contains 25,000 reviews, where 12500 are positive reviews and 12500 are negative reviews. In addition, the training dataset contains 75,000 reviews, whereas 12,500 are positive reviews, 12,500 are negative reviews, and 50,000 reviews are unsupervised.

B. Generative Adversarial Networks (GAN)

There are many algorithms of Machine Learning, and Generative Adversarial Networks (GAN) is one of them. A generative adversarial network (GAN) developed by Ian Goodfellow et al. [5]. The GAN works with two kinds of neural networks: one is a generator, and another is a discriminator. The generator works as an input and generates fake samples, and the discriminator gives binary results; it tells whether the sample is real or fake. The generator intends to take the wrong one, whereas the discriminator's objective is to make a difference between the real and fake samples. The below equation gives a mathematical overview of GAN.

$$\min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

where, the z is random noise, x is real sample, $G(z)$ is generated sample.

On the other hand, the generator G wants to fool the discriminator, D, and compel the discriminator to believe that the generated samples are real.

C. Lipschitz Recurrent Neural Networks (LRNN)

The Lipschitz Recurrent Neural Networks was first introduced by Erichson et al. [6].In their paper [6], they expressed the continuous-time Lipschitz recurrent neural network with the functional by the following equation:

$$\dot{h} = A_{\beta_A,\gamma_A} h + tanh(W_{\beta_w,\gamma_w} h + Ux + b) \tag{2}$$

$$y = Dh \tag{3}$$

where the hidden-to-hidden matrices $A_{\beta,\gamma} \epsilon R^{N*N}$ and $W_{\beta,\gamma} \epsilon R^{N*N}$ and Erichson et al. [6] defined the hidden-to-hidden by the following equation:

$$A_{\beta_A,\gamma_A} = (1-\beta_A)(M_A + M_A^T) + \beta_A(M_A - M_A^T) - \gamma_A I \tag{4}$$

$$W_{\beta_W,\gamma_W} = (1-\beta_W)(M_W + M_W^T) + \beta_W(M_W - M_W^T) - \gamma_W I \tag{5}$$

where $\beta_A, \beta_W, \gamma_A, \gamma_W$ are tunable parameters and M_A, M_W are trainable matrics [6].

D. A process for the prediction of sentiment

This text analytics approach involves several key steps in the sentiment prediction process. The model first creates token vectors by converting textual reviews. The complex Lipschitz Recurrent Neural Network (LRNN) architecture systematically processes these token vectors.

The main components of the LRNN operation are the modification and Improvement of the hidden state, represented by $h(t)$. This improvement occurs by combining the prior state with the present word vector. As it moves through each word in the input sequence, this crucial process makes use of matrices $W_{\beta,\gamma}$ and $A_{\beta,\gamma} \in R^{N*N}$ to enable the continuous updating of the hidden state. The dynamic updating process plays a crucial role in allowing the model to identify and capture complex sequential dependencies and subtle patterns present in the text input.

The final goal of this extensive exercise is to give the model the capacity to provide precise binary sentiment forecasts. Specifically, the model classifies the sentiment of the text under investigation by allocating a value of 1 for positive view and 0 for negative sentiment. During this challenging journey, the hidden state h is a critical channel for maintaining and distributing essential data throughout the network. Its significance in the procedure cannot be emphasized, as it provides the foundation for the model's ability to forecast sentiment accurately. Notably, the transformation of the hidden state may be formally represented by the following equation, which captures its evolution with the symbol $\hbar$:

$$\hbar = h_t + \epsilon(\alpha A_{\beta_A,\gamma_A} h_t + tanh(W_{\beta_w,\gamma_w} h_t + Ux + b)) \\ .\sigma(W_{\beta_w,\gamma_w} h_t + U_{gate} x + b)$$

Where,

- ϵ, α, U, U_{gate}: Scalar, hyperparameter, and weight matrices
- σ and $tanh$: Activation functions

E. Lipschitz Recurrent Neural Networks (LRNN) as generator and discriminator

One of the goals of LRNN is to show how the generator produces text with desired sentiment labels as input. The model can predict two types of sentiment: one is a positive sentiment, in which the label is 1, and another is a negative sentiment, in which the label is 0. Since the generator generates the fake sentiments from the random noise, the LRNN model can check the label of the generated sentiment label to determine whether it is positive or negative. If the generated sentiment will mismatch, then the hidden state of LRNN shows how to get the desired sentiment. In addition, another role of the hidden state is to capture the sequential dependencies in the generated text. In addition, the LRNN model gives the sentiment score for the generated sentiment, which can also measure the generator's performance. The following equation indicates how the hidden state works with the generator. Now, by using the Equation 1 the loss function for the Generator (G) can be written:

$$L_G = -\log(D(G(z, \hbar))) \tag{6}$$

In the above equation, z represents random noise, z and $\hbar$ collaborate to generate fake reviews that mislead the Discriminator (D). This z and $\hbar$ combination creates realistic, sentiment-aware data.

In our model, a set of equations that incorporate linear and non-linear transformations with trainable matrices (M_A, M_W) and adjustable parameters ($\beta_A, \beta_W, \gamma_A, \gamma_W$) control the hidden state dynamics of the generator. The hidden state $\hbar$ is updated according to these equations (2-5), which affects the generator's capacity to process inputs and produce data.

The above loss functions help ensure that the Generator produces text with the desired sentiment characteristics.

On the other hand, the discriminator in this model is whether the sentiment generated by the generator is real or fake. In this model, the discriminator gets two inputs: one is from the real data, and another is from the generated data. This dual input mechanism helps the discriminator determine whether the generated sentiment is real or fake. In addition, due to its good detection ability, it can detect the sentiment label from the real data with the help of the hidden state of LRNN. The following equation indicates the interaction between the hidden state and the discriminator. So by using the Equation 1 the loss function for the Discriminator (D) can be written:

$$L_D = -\log(D(\text{real})) - \log(1 - D(G(z, \hbar))) \tag{7}$$

From the above equation, the combination of z and $\hbar$ helps the Discriminator (D)differentiate between real and fake data. In addition, the combination of z and $\hbar$ improves the Discriminator's ability to accurately identify sentiment (e.g., positive or negative) from the real data.

The training progress of the GAN is tracked by logging these loss values. As training advances, decreasing L_D and L_G values indicates an improvement in the discriminator's capability to distinguish between real and fake reviews and the generator's proficiency in generating more authentic reviews.

In this paper, Fig. 1 refers to the overall sentiment analysis process, and Fig. 2 shows the training process of GAN can find this figure on page 4.

F. Hyperparameters

Different kinds of hyperparameters are used in this experiment. Parameters are outlined in Table I

TABLE I
HYPERPARAMETERS

Hyperparameter	Value
Batch Size	32
Epochs	100
Noise Dimension	100
Max Sequence Length	100
Generator Hidden Size	256
Generator Hidden Size	10
Sentiment Embedding Dim	2
Number of Sentiments	1
Learning Rate (Generator)	0.001
Learning Rate (Discriminator)	0.001

IV. Results

A. Calculations

The following equations were used to determine the evaluation matrices Precision, Recall, F1 score, and accuracy, and table II shows the value of Precision, Recall, F1 score, and accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

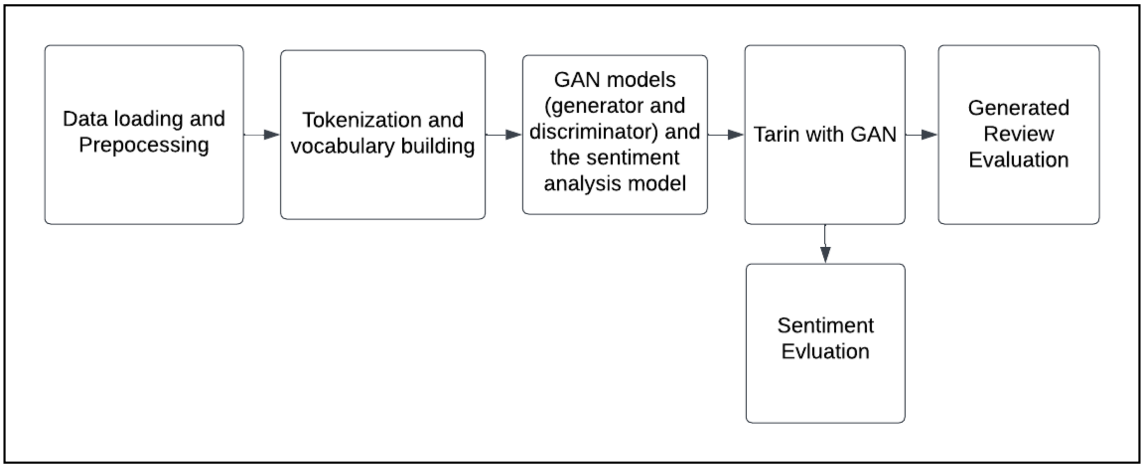

Fig. 1. Overall Sentiment Analysis Model Framework

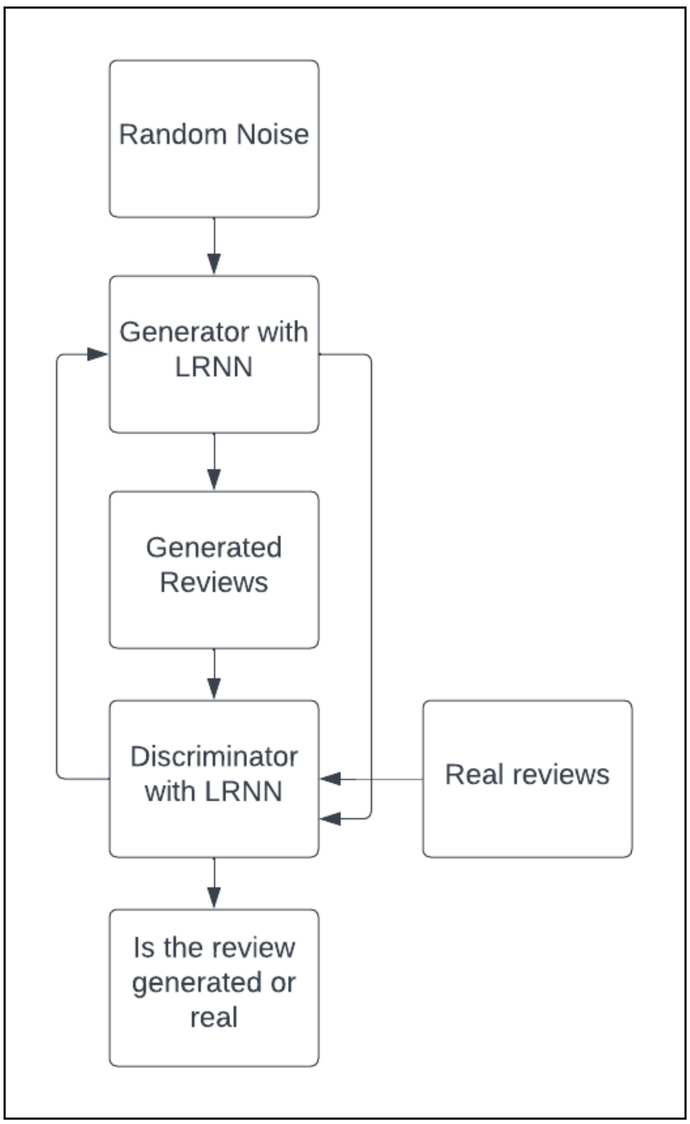

Fig. 2. GAN Training process

Where, TP= True Positive and FP = False Positive

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

Where, TP= True Positive and FN = False Negative

$$F1 = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \tag{11}$$

Where, TP= True Positive, FN = False Negative, FP = False Positive and TN= True Negative

TABLE II
EVALUATION MATRICES PERFORMANCE AND ACCURACY

F1 score	Recall	Precision	Accuracy (%)
0.94	1.00	0.99	91.30

B. *Compare with other models*

Our model has been evaluated in comparison to the following models:

- S-LSTM [21]
- CNN+LSTM [22]
- CoRNN [23]
- UnICORNN [24]

Upon conducting this comparison, it becomes evident that our model achieved the highest accuracy, reaching 91.30%. For a detailed comparison of the various models, see the table III.

V. DISCUSSION

This experiment combines a GAN (Generative Adversarial Network) with a Lipschitz Recurrent Neural Networks (LRNN), a type of neural network. Our model has demonstrated strong performance with an accuracy of 91.30%. Furthermore, our F1 score, Recall, and Precision metrics stand at 0.91, 0.99, and 0.95, respectively.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SENTIMENT ANALYSIS MODEL

Model	Accuracy (%)
S-LSTM	87.15
CNN+LSTM	88.9
CoRNN	87.40
UnICORNN	88.40
Our Model	**91.30**

The primary objective of this experiment is to implement the LRNN-based GAN model into a specialized chatbot, particularly in the medical or domain-specific context. Text generation, particularly for responding to questions, is a critical task in this scenario, with the LRNN playing a vital role in the generation process. However, due to the absence of medical data, we opted to use IMBD movie reviews, which often contain unclear pieces of reviews. We anticipate achieving even higher accuracy if we train this model with domain-specific data. Each domain possesses distinct terminology and context, crucial for predicting and generating dialogue.

VI. Conclusion and Future Work

Sentiment analysis using the Lipschitz Recurrent Neural Network (LRNN) in combination with Generative Adversarial Networks (GAN) has demonstrated promising outcomes when applied to the IMDB dataset. This approach addresses common GAN training challenges, such as instability and the vanishing gradient problem, leading to enhanced model accuracy.

Furthermore, we plan to deploy this model in various domains, including healthcare, the stock market, and educational institutions. Additionally, we aim to integrate this model into chatbot applications in the future.

Acknowledgment

This work is supported in part by DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory, National Science Foundation under Grant No. 2219742 and Grant No. 2131001.

References

[1] Y. Shang, X. Su, Z. Xiao, and Z. Chen, “Campus sentiment analysis with gan-based data augmentation,” in *2021 13th International Conference on Advanced Infocomm Technology (ICAIT)*. IEEE, 2021, pp. 209–214.

[2] X. Gong, “English text sentiment analysis based on generative adversarial network,” *Evolutionary Intelligence*, pp. 1–9, 2022.

[3] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[4] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] N. B. Erichson, O. Azencot, A. Queiruga, L. Hodgkinson, and M. W. Mahoney, “Lipschitz recurrent neural networks,” *arXiv preprint arXiv:2006.12070*, 2020.

[7] L. Kurniasari and A. Setyanto, “Sentiment analysis using recurrent neural network,” in *Journal of Physics: Conference Series*, vol. 1471, no. 1. IOP Publishing, 2020, p. 012018.

[8] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based lstm for aspect-level sentiment classification,” in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 606–615.

[9] E. F. Can, A. Ezen-Can, and F. Can, “Multilingual sentiment analysis: An rnn-based framework for limited data,” *arXiv preprint arXiv:1806.04511*, 2018.

[10] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, “Sentiment analysis of comment texts based on bilstm,” *Ieee Access*, vol. 7, pp. 51 522–51 532, 2019.

[11] M. Usama, B. Ahmad, E. Song, M. S. Hossain, M. Alrashoud, and G. Muhammad, “Attention-based sentiment analysis using convolutional and recurrent neural network,” *Future Generation Computer Systems*, vol. 113, pp. 571–578, 2020.

[12] S. Pal, S. Ghosh, and A. Nag, “Sentiment analysis in the light of lstm recurrent neural networks,” *International Journal of Synthetic Emotions (IJSE)*, vol. 9, no. 1, pp. 33–39, 2018.

[13] A. Baliyan, A. Batra, and S. P. Singh, “Multilingual sentiment analysis using rnn-lstm and neural machine translation,” in *2021 8th international conference on computing for sustainable global development (INDIACom)*. IEEE, 2021, pp. 710–713.

[14] S. C. Nistor, M. Moca, D. Moldovan, D. B. Oprean, and R. L. Nistor, “Building a twitter sentiment analysis system with recurrent neural networks,” *Sensors*, vol. 21, no. 7, p. 2266, 2021.

[15] R. Ni and H. Cao, “Sentiment analysis based on glove and lstm-gru,” in *2020 39th Chinese control conference (CCC)*. IEEE, 2020, pp. 7492–7497.

[16] A. Agarwal, A. Yadav, and D. K. Vishwakarma, “Multimodal sentiment analysis via rnn variants,” in *2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*. IEEE, 2019, pp. 19–23.

[17] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, “Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis,” *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[18] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, “A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis,” *Multimedia Tools and Applications*, vol. 78, pp. 26 597–26 613, 2019.

[19] U. Naseem, I. Razzak, K. Musial, and M. Imran, “Transformer based deep intelligent contextual embedding for twitter sentiment analysis,” *Future Generation Computer Systems*, vol. 113, pp. 58–69, 2020.

[20] H. Takamura, T. Inui, and M. Okumura, “Latent variable models for semantic orientations of phrases,” in *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006, pp. 201–208.

[21] Y. Zhang, Q. Liu, and L. Song, “Sentence-state lstm for text representation,” *arXiv preprint arXiv:1805.02474*, 2018.

[22] J. Camacho-Collados and M. T. Pilehvar, “On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis,” *arXiv preprint arXiv:1707.01780*, 2017.

[23] T. K. Rusch and S. Mishra, “Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies,” *arXiv preprint arXiv:2010.00951*, 2020.

[24] T. Konstantin Rusch and S. Mishra, “Unicornn: A recurrent model for learning very long time dependencies,” *arXiv e-prints*, pp. arXiv–2103, 2021.