Deep Learning Approaches for Sentiment Analysis: Comparative results

Abstract—**Sentiment analysis is an indispensable tool with which to extract valuable knowledge from user-generated textual content, especially in the rapidly changing world of e-commerce. Sentiment analysis applied to Amazon reviews This research explored the use of several deep learning models, including Convolutional Neural Networks , Long Short-Term Memory networks and Bidirectional LSTMs . Focusing on rigorous evaluation metrics-accuracy, precision, recall and F1 score - the study attempts to find out the best model for distinguishing sentiments found in Amazon reviews. This research pushes the methodology beyond traditional lecture format, adding data augmentation and ensemble methods as well multiple types of optimizers. The results of this study not only enhance the methods used in sentiment analysis, but also indicate practical things that are important to e-commerce.**

Index Terms—**Sentiment analysis, User-generated textual content, E-commerce, Amazon reviews, Deep learning models,**

I. Introduction

Sentiment analysis plays a crucial role in the field of natural language processing, witnessing substantial advancements through the integration of deep learning techniques. Its significance lies in its ability to discern emotions and opinions embedded in textual content, providing valuable insights into user experiences for informed decision-making in the e-commerce domain. This research aims to assess the effectiveness of three well-established deep learning models in sentiment analysis, specifically focusing on Amazon reviews. The models under investigation include 1D CNN , LSTM, and Bi-LSTMs [1].

Within the expansive realm of sentiment analysis research, the careful selection and application of robust evaluation metrics are imperative to ensure the reliability and interpretability of results. Metrics such as accuracy, precision, recall, and F1 score serve as essential benchmarks in this evaluative process. As the e-commerce landscape continues to evolve, businesses are confronted with the increasing need to adapt, highlighting the exploration and implementation of state-of-the-art methodologies for extracting meaningful sentiments from diverse and voluminous textual data sources. This research transcends conventional boundaries by incorporating advanced techniques, including data visualization and machine learning algorithms, to provide a comprehensive understanding of sentiment analysis.

To this end, the main contribution of this paper is to provide a comparative analysis thorough evaluation of several deep learning models applied to sentiment analysis. The paper shows the impact of data preprocessing and data augmentation to handle unbalanced classes. The proposed models will be evaluated on the Amazon reviews dataset.

The rest of this paper is organized as the following: Section II shows several related works. Section III presents data analysis and preprocessing. Section IV introduces the proposed methodology. Section V shows the experimental results and the comparative analysis of the proposed models. Finally, Section VI concludes the paper.

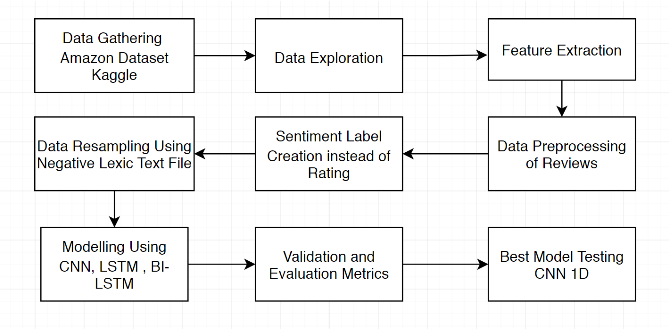


Fig. 1. Work Process Followed

II. Related Work

The area of sentiment analysis has experienced a boom in research activities in the recent past, indicating the increasing significance of deducing complex sentiments from written text [2] . There are many research papers that have looked into different techniques for figuring out what people feel about things they post online. There have been different ways of conducting this, like looking at the words people use or analyzing the patterns in the data. Despite the vast field of deep learning, there is a persistent shortage in investigating certain architectures, specifically the application of 1D CNNs and LSTM networks. Although these two models have proven their success in other areas, not enough studies examine their performance in sentiment analysis and how they handle the specific problems that Amazon reviews present. This research paper aims to examine and analyze the existing literature on the gap between the use of 1D CNN and LSTM models in sentiment analysis. This will help fill the limited or overlooked application areas at present.

The study by Elli, Maria, and Yi-Fan [3] began with sentiment extraction from reviews which helped them to build a strong business model. The research was conducted on different topics such as sentiment analysis, recognition of emotional expressions, determining gender from names, and determining fake reviews. The two programming languages that came out on top in their toolkit were Python and R. Meanwhile, the primary tools used for classification were Multinomial Naïve Bayesian (MNB) and SVM.

Shaikh, Tahura, and Deepaeshpande introduce distinct sentiment analysis feature extraction and selection techniques [4]. Initially, an Amazon dataset was gathered, and a meticulous data cleaning process ensued, involving the removal of stop words and special characters. The research then delved into an examination of various feature selection and extraction methods, strategically applied at different grammatical levels, namely phrases, single words, and multi-word phrases. The Naive Bayes classifier was chosen for evaluation in this task.

According to the study, it was discerned that employing the Naive Bayes classifier at the word phrase level yielded more effective outcomes compared to its application with single words or multiple words. Nevertheless, it is imperative to acknowledge a limitation inherent in the research—sole reliance on the Naive Bayes classifier, which may impede the generation of comprehensive outcomes.

Nasr, Mona Mohamed, Essam Mohamed Shaaban and Ahmed Mostafa Hafez made a purposive decision to use simple, easy-to-understand algorithms [5]. The authors opted for straightforward algorithms to enhance the interpretability. The results were amazing, especially when achieving high accuracy with the (SVM) model. However, it is important to remember that there was an issue with the system's effectiveness when used on large data sets during experimentation. Srinivas, Akana , Satyanarayana [6] used LSTM as their main model, and it achieved reasonable outcomes.

Haque, Tanjim Saber, Nudrat Shah, Faisal described a method that utilized the TF-IDF technique to rank words and make them more significant in sentiment analysis [7]. Classification techniques were also used such as logistic regression. the main emphasis of their study is on Deep Neural Networks (DNN), specifically 1D CNN, LSTM, and Bi-LSTM models. Although the research methods used are not CNN, LSTM or Bi-LSTM, t the authors managed to exceed the evaluation threshold by more than 90%. It demonstrates the variance in techniques among researchers to optimize sentiment analysis and achieve the desired outcomes.

Ammar Rashed Hamdallah encountered an instance of class imbalance even though data they used was diverse [8]. Utilizing the confusion matrix as a primary evaluation metric, the model's performance was degraded. This includes accuracy, kappa (inter-rater agreement), specificity, and sensitivity; where appropriate. The model, according to the results, has shown 79% accuracy when measured. Although the dataset is distinct, it showed that the research could maintain high accuracy at 97% among varied datasets.

While the sentiment analysis research landscape has seen considerable growth, the investigation into existing literature has not yielded specific documentation on applying 1D CNNs and LSTM networks to the same dataset utilized in this study. Although Kaggle code projects have explored sentiment analysis on similar datasets, a notable observation is the recurring issue of class imbalance that remains unaddressed in many models. Furthermore, these Kaggle projects often lack comprehensive testing on comments resembling the intricacies of the Amazon review dataset this study focused on. Despite the plethora of code implementations, the gap in research documentation persists, emphasizing the need to thoroughly examine the effectiveness of 1D CNNs and LSTM networks in handling the specific challenges posed by Amazon reviews. This paper aims to fill this void by critically analyzing the performance of these models in the context of sentiment analysis, addressing the existing limitations in the current body of literature.

III. Data Analysis and Preprocessing

The selection of the Amazon dataset was motivated by its vast repository of reviews, offering a substantial and diverse collection of user sentiments. However, this paper's exploration faced certain challenges, which were addressed in this section, including considerations related to data imbalance and the subsequent preprocessing steps undertaken to address these issues.

A. Data Preprocessing

In the initial stages of the preprocessing pipeline, comprehensive examination pf the dataset's structure was conducted. This involved scrutinizing its shape, inspecting the first few entries using the head method, and reviewing the available columns. Subsequently, it was discerned that only two columns were pertinent to the analysis: ”overall,” denoting the user ratings on a scale of 1 to 5, and ”reviewText,” encapsulating the textual content of the reviews. This focused selection laid the foundation for the subsequent preprocessing techniques.

After this initial selection, text normalization techniques were applied:

1) Lowercasing: All text was converted to lowercase to ensure uniformity, preventing variations in letter cases from affecting subsequent analyses [2].
2) Tokenization: Tokenization involved breaking down the text into individual words or tokens. This process facilitates the analysis of the textual content at a granular level [2].
3) Stopword Removal: Stopwords, commonly occurring words that may not have significant meaning, were removed. This step helps focus on the essential content of the reviews [2].

By applying these preprocessing techniques, the aim was to create a more standardized and informative representation of the Amazon reviews dataset, preparing it for subsequent analysis using deep learning models.

Following the initial preprocessing steps, refinement of the dataset was conducted by transforming the rating column (”overall”). To simplify the sentiment analysis task, categorization of the reviews into three sentiment classes was performed:

1) Negative: Reviews with ratings of 1 or 2 were labeled as negative sentiments.

2) Neutral: Reviews with a rating of 3 were considered neutral. Placing a rating of 3 in the neutral category aims to avoid biasing the model toward specific sentiments and maintain a balanced perspective.
3) Positive: Reviews with ratings of 4 or 5 were classified as positive sentiments.

This transformation provided a more straightforward and interpretable process for sentiment analysis. By introducing a neutral category for reviews with a moderate rating, the aim was to prevent the model from associating specific words with either positive or negative sentiments too strongly. The objective was to achieve a more nuanced understanding of sentiment in Amazon reviews, enhancing the robustness of the subsequent deep-learning models.

B. Imbalance of Data

Upon visualizing the distribution of sentiments in the dataset, A notable imbalance was encountered, with the positive sentiment class dominating. This observation prompted us to address the data imbalance issue, as imbalanced datasets can lead to biased models, particularly favoring the majority class. In the preliminary attempts to build a model without addressing the imbalance, the performance metrics, especially during validation and testing, were suboptimal. The model tended to predict most instances as positive sentiments, effectively ignoring the nuances of negative sentiments. This imbalance hindered the learning process, as the model was not effectively exposed to a diverse set of negative sentiments for robust sentiment analysis.

Recognizing the significance of mitigating data imbalance, the employment of specific strategies in subsequent iterations to rectify the class distribution and enhance the model's capability to discern sentiments effectively was executed. These strategies were crucial for achieving a balanced and reliable sentiment analysis model

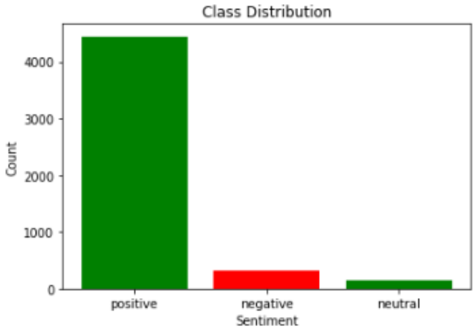


Fig. 2. Class Distribution: Imbalance of Data

C. Approach for Imbalance of Data

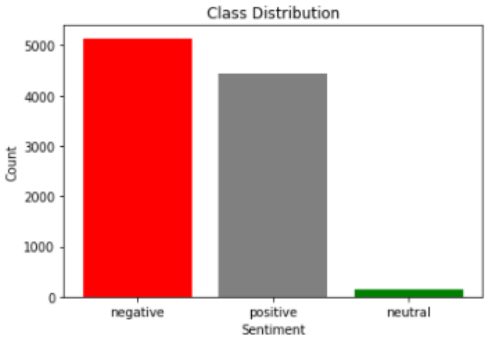
The preliminary attempts to construct a sentiment analysis model without addressing the imbalance unveiled noteworthy shortcomings. During validation and testing, the model displayed suboptimal performance metrics. The prevalence of positive sentiment predictions indicated a lack of nuance in capturing negative sentiments, hindering the model's ability to learn from a diverse set of negative instances.

1) Mitigation Strategies: In response to the identified imbalance, targeted strategies were adopted to augment the dataset using a negative words text file:

2) Data Augmentation with Negative Words: Negative words from an external text file were integrated to augment the existing positive sentiment class. This involved the introduction of variations, paraphrasing, and incorporation of negative lexicons, enriching the dataset with nuanced instances of negative sentiments.

3) Resampling Techniques: Complementing the augmentation process, a combination of oversampling and undersampling techniques was applied. The negative words text file facilitated the oversampling of the minority class, ensuring a more balanced representation of sentiments.

The integration of negative words from an external source successfully contributed to achieving a balanced sentiment analysis model. Evaluation metrics demonstrated significant enhancements, affirming the effectiveness of leveraging external negative lexicons to counteract data imbalance.



/

Fig. 3. Class Distribution After Augmentation

IV. The Proposed Methodology

After achieving a clean and evenly divided dataset, the next task was to figure out how to categorize people's opinions accurately. Since it has been found effective in many research studies, the deep neural network (DNN) method was chosen for its ability to extract complex patterns from text data. This selection was due to DNNs' impressive track record in many NLP jobs, specifically in determining consumer sentiment.

The first step in the text preprocessing pipeline was tokenization. This process converts the text into a format that can be fed into a neural network. Breaking down a text into smaller units, such as words or sentences, is referred to as tokenization. In the next step, padding was added to the sequences so that all of them can have the same length, which is necessary for neural network models. Then, label encoding is applied to padded sequences, used to convert categorical labels into numerical values for feeding into neural networks [10] .

So, these preprocessing steps were kind of important for creating deep learning models that could totally understand what people felt about stuff. Applying tokenization, padding, and label encoding as the initial preparations before starting the training, validating, and testing the deep neural network models on sentiment analysis.

A. *1D Convolutional Neural Network (1D CNN)*

The initialization of the model was conducted with the selection of a 1D Convolutional Neural Network (1D CNN), recognizing its effectiveness in capturing local patterns and dependencies within sequential data. The convolutional layers in a 1D CNN act as filters, enabling the model to automatically learn relevant features from the input text.This architecture is particularly adept at identifying key sentiment-related patterns in sequential data. [9]

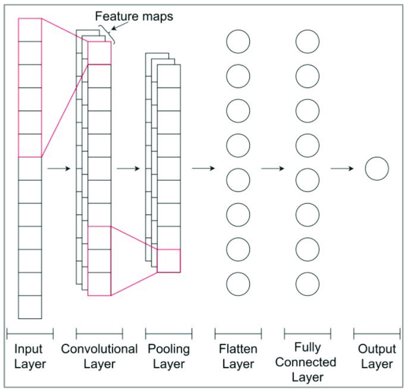


Fig. 4. 1D CNN Model Architecture [10]

Input Representation: The input to the 1D CNN is the sequence of words in the statement, and each word is represented as an embedding vector. This sequence of word embeddings serves as the model's input. Convolutional Operation: The 1D CNN applies filters over the input sequence, sliding them to capture local patterns. For the statement ”This product is bad,” a filter might focus on the combination of words like ”is bad.” The convolution operation computes the dot product at different positions, capturing the significance of the chosen filter. Following the convolution, a Rectified Linear Unit (ReLU) activation function is applied to introduce non-linearity. Max pooling is used to down-sample the output, selecting the most relevant information. It may emphasize the critical words in the phrase that contribute to the sentiment, such as bad. The 2D matrix resulting from pooling is flattened into a 1D vector. This step retains the learned information in a format suitable for fully connected layers. The flattened vector is connected to fully connected layers, enabling the model to capture high-level abstractions. These layers aggregate information from different parts of the input sequence. The output layer produces a probability score indicating the likelihood of the sentiment being negative. In this case, the model, having learned patterns from training data, might assign a high probability to negativity based on the phrase ”is bad.”

B. *Long Short-Term Memory Networks (LSTM)*

Long Short-Term Memory Networks (LSTM) were the second model in the lineup, chosen for their ability to capture long-range dependencies in sequential data. LSTMs excel at mitigating the vanishing gradient problem, allowing them to retain and utilize context information over extended sequences. [11] This makes LSTMs well-suited for tasks where understanding sentiment requires grasping the nuanced relationships between distant words in a text. [11]

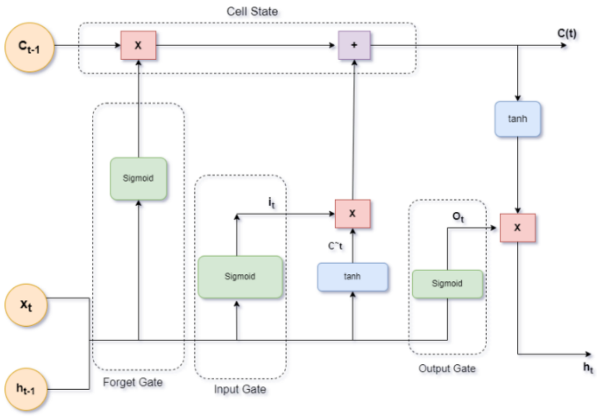


Fig. 5. LSTM Model Architecture [12]

Input Representation: The LSTM processes input sequences, where each word is represented as an embedding vector. These vectors capture the semantic meaning of words in the context of the sequence.

Time Steps: The LSTM operates sequentially, processing one word at a time. Each word corresponds to a specific time step in the sequence.

Forget Gate Operation: The forget gate determines what information from the previous time step should be retained or discarded. It considers the current input and the output from the previous time step.

Input Gate Operation: The input gate decides what new information should be stored in the memory cell. It evaluates the current input and the output from the previous time step, determining the relevance of incoming information.

Candidate Cell State: The candidate cell state represents new information that can be added to the memory cell. It reflects the LSTM's assessment of the current input's significance.

Cell State Update: The cell state is updated based on the forget gate, input gate, and the candidate cell state. This allows the LSTM to selectively incorporate valuable information while discarding less relevant details.

Output Gate Operation: The output gate decides the final output at the current time step. It considers the current input, the output from the previous time step, and the updated cell state, determining what information should be passed on as the output.

Final Output: The final output represents the LSTM's interpretation of the entire input sequence after processing all time steps. This output is often used for downstream tasks, such as sentiment analysis.

C. Bidirectional Long Short-Term Memory (Bidirectional LSTM)

The third model in the comparison was the Bidirectional Long Short-Term Memory , a variant of the traditional LSTM [14]. By processing input sequences in both forward and backward directions, Bidirectional LSTMs enhance the model's ability to capture contextual information from both ends of a sequence. This bidirectional processing enables a more comprehensive understanding of the sentiment expressed in a text, making Bidirectional LSTMs a valuable choice for sentiment analysis tasks [14].

Each of these models was chosen with a specific set of advantages in mind, and their comparative analysis would provide insights into their respective strengths and weaknesses in the context of sentiment analysis on Amazon reviews.

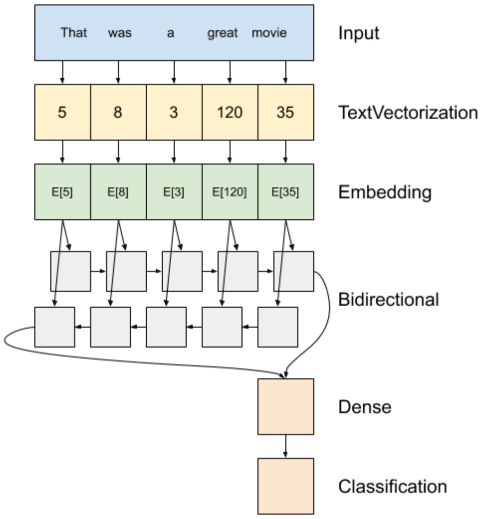


Fig. 6. Bidirectional LSTM Model Architecture [13]

Input Representation: Similar to the unidirectional LSTM, the input to the Bidirectional LSTM is the sequence of words in the statement, and each word is represented as an embedding vector. The sequence of word embeddings serves as the model's input.

Bidirectional LSTM Operation: In a Bidirectional LSTM, the input sequence is processed in both forward and backward directions. Two separate LSTM layers are employed—one processes the sequence from the beginning to the end, and the other processes it from the end to the beginning. The outputs from both directions are concatenated at each time step, providing a more comprehensive understanding of the context.

Fully Connected Layer: The concatenated outputs from the Bidirectional LSTM are connected to a fully connected layer, which captures higher-level features and prepares the data for the final sentiment prediction.

Output: The output layer produces a probability score indicating the likelihood of the sentiment being negative. For the input ”This product is bad,” the Bidirectional LSTM, having considered information from both directions, might assign a high probability to negativity based on the phrase ”is bad.”

V. Experimental Results

A. 1D Convolutional Neural Network (1D CNN)

Class	Precision	Recall	F1-Score	Support
0 Positive	0.99	0.97	0.98	991
1 Neutral	0.17	0.03	0.06	29
2 Negative	0.95	0.99	0.97	927
Accuracy		0.97		

TABLE I
Metrics for the 1D CNN Model

The 1D CNN model demonstrates robust performance in sentiment analysis on the Amazon reviews dataset. Precision scores for both positive (Class 0) and negative (Class 2) sentiments are notably high, showcasing the model's ability to correctly identify instances of each class. With a precision of 0.99 for positive sentiment, the model exhibits a minimal rate of false positives, while the precision of 0.95 for negative sentiment signifies a strong capability to correctly classify negative instances. The recall metrics further emphasize the model's effectiveness, with a recall of 0.97 for positive sentiment and an impressive 0.99 for negative sentiment. These high recall values indicate that the model excels in capturing the majority of relevant instances for both classes. The overall accuracy of 0.97 attests to the model's proficiency in accurately classifying sentiments, providing a comprehensive evaluation of its performance on the given Amazon reviews dataset.

B. Long Short-Term Memory Networks (LSTM)

Class	Precision	Recall	F1-Score	Support
0 Positive	0.99	0.95	0.97	991
1 Neutral	0.13	0.14	0.13	29
2 Negative	0.93	0.97	0.95	927
Accuracy		0.95		

TABLE II
Metrics for the LSTM Model

The LSTM model delivers a solid performance in sentiment analysis on the Amazon reviews dataset, showcasing distinctive strengths and nuances compared to the 1D CNN model. In terms of precision, the LSTM model achieves a precision of 0.99 for positive sentiment (Class 0), surpassing the 1D CNN model's precision of 0.99. However, for negative sentiment (Class 2), the LSTM model records a precision of 0.93, slightly lower than the 1D CNN model's precision of 0.95.

Moving on to recall metrics, the LSTM model demonstrates a recall of 0.95 for positive sentiment and an impressive 0.97 for negative sentiment. In comparison, the 1D CNN model shows a recall of 0.97 for positive sentiment and an equally impressive 0.99 for negative sentiment. These high recall values across both models indicate their effectiveness in capturing the majority of relevant instances for each sentiment class.

The overall accuracy of the LSTM model stands at 0.95, providing a comprehensive evaluation of its performance. While the LSTM model excels in certain aspects, it's crucial to consider the trade-offs and nuances when compared to the 1D CNN model, highlighting the diverse strengths of each approach in sentiment analysis on the given Amazon reviews dataset.

C. Bidirectional Long Short-Term Memory (Bidirectional LSTM)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0 Positive	0.99	0.94	0.96	991
1 Neutral	0.04	0.07	0.05	29
2 (Negative)	0.93	0.97	0.95	927
Accuracy		0.94		

TABLE III
METRICS FOR THE BIDIRECTIONAL LSTM MODEL

The Bidirectional LSTM model demonstrates competitive performance in sentiment analysis on the Amazon reviews dataset. Looking at precision, recall, and F1-score metrics, the model achieves notable results for both positive and negative sentiments. However, compared to the 1D CNN and LSTM models, the Bidirectional LSTM exhibits a lower precision and recall for the neutral class (Class 1).

When comparing the three models, the 1D CNN consistently outperforms the LSTM and Bidirectional LSTM models across all classes. The 1D CNN achieves higher precision, recall, and F1-score for positive and negative sentiments, showcasing its robustness in correctly classifying instances of each class. Additionally, the 1D CNN demonstrates a substantially higher precision for the neutral class compared to the Bidirectional LSTM model.

While the Bidirectional LSTM model provides a reasonable performance, the 1D CNN stands out as the superior model for sentiment analysis on the Amazon reviews dataset, offering a balanced and accurate classification across all sentiment classes. The choice of the 1D CNN model can be attributed to its effectiveness in capturing local patterns and dependencies in sequential data, making it particularly well-suited for text-based sentiment analysis tasks.

VI. CONCLUSION

In conclusion, this study looked at sentiment analysis on Amazon reviews in depth. Utilizing the newest deep learning models like 1D CNN, LSTM, and Bidirectional LSTM to achieve it. A detailed approach was used when dealing with data preprocessing which included addressing challenges such as datasets that were not balanced properly and different types of rating scripts that had to be handled with care. The decision to utilize the Kaggle dataset available on Amazon reviews was grounded in the dataset's large number of diverse reviews.

after analyzing the results of the three models, it appears that the 1D CNN model performed the best in sentiment classification. The 1D CNN model has shown the highest level of accuracy in sentiment analysis when compared to other models. As a result, the 1D CNN outperformed both LSTM and Bidirectional LSTM in textual data analysis, thus highlighting its effectiveness in recognizing intricate patterns.

The approach used in this paper for sentiment analysis is different than other methods due to the usage of deep neural networks which are more effective. The usage of 1D CNN helped overcoming the inaccuracies in traditional models of classification such as logistic regression, decision trees and SVM, and coming up with a highly detailed representation of sentiments on Amazon reviews. The application of deep neural networks, especially 1D CNN, sets a standard for sentiment analysis, this shows how valuable it is to use advanced techniques to extract meaningful information from large amounts of varied text data.

REFERENCES

[1] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural language processing: python and NLTK*. Packt Publishing Ltd, 2016.

[2] J. Sangeetha and U. Kumaran, "Sentiment analysis of amazon user reviews using a hybrid approach," *Measurement: Sensors*, vol. 27, p. 100790, 2023.

[3] M. S. Elli, Y.-F. Wang, *et al.*, "Amazon reviews, business analytics with sentiment analysis," *Elwalda, Abdulaziz, et al."Perceived Derived Attributes of Online Customer Reviews*, 2016.

[4] T. Shaikh and D. Deshpande, "Feature selection methods in sentiment analysis and sentiment classification of amazon product reviews," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 36, no. 4, pp. 225–230, 2016.

[5] M. M. Nasr, E. M. Shaaban, and A. M. Hafez, "Building sentiment analysis model using graphlab," *Int J Sci Eng Res*, vol. 8, no. 11551160, 2017.

[6] A. C. M. V. Srinivas, C. Satyanarayana, C. Divakar, and K. P. Sirisha, "Sentiment analysis using neural network and lstm," in *IOP conference series: materials science and engineering*, vol. 1074, p. 012007, IOP Publishing, 2021.

[7] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE international conference on innovative research and development (ICIRD)*, pp. 1–6, IEEE, 2018.

[8] A. R. Hamdallah, "Amazon reviews using sentiment analysis," 2021.

[9] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," *arXiv preprint arXiv:1809.08037*, 2018.

[10] S. Guessoum, S. Belda, J. Ferrándiz, S. Modiri, S. Raut, S. Dhar, R. Heinkelmann, and H. Schuh, "The short-term prediction of length of day using 1d convolutional neural networks (1d cnn)," *Sensors*, vol. 22, p. 9517, 12 2022.

[11] K. Borna and R. Ghanbari, "Hierarchical lstm network for text classification," *SN Applied Sciences*, vol. 1, pp. 1–4, 2019.

[12] E. B. Thomas, "Understanding lstm: An in-depth look at its architecture, functioning, and pros cons," *LinkedIn*, Jul 2023.

[13] V. Coumar and LakshmanaPandian.S, "Deep learning-based text segmentation in nlp using fast recurrent neural network with bi-lstm," *Advances in Parallel Computing*, vol. 38, p. 87, 10 2021.