

# Deep Learning Approach for Sentiment-Driven Rating Prediction

Mohammad Asif<sup>1</sup>, Mujawar Mohammad Omar<sup>2</sup>, Mohamad Rafi<sup>3</sup>, Syed Aleemuddin<sup>4</sup>,  
Mohammed Abdul Muqueet<sup>5</sup>

<sup>1</sup>Assistant Professor, Department of CSE (Data Science), Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

<sup>2,3,4,5</sup> UG Students, Department of CSE (Data Science), Lords Institute of Engineering and Technology, Hyderabad, Telangana, India.

**Abstract**— This work presents a web-based application developed to analyze user reviews and predict both sentiment and corresponding star ratings using machine learning and deep learning techniques. The system provides an interactive interface where users can register, log in, upload datasets, and process textual data with ease. After loading the dataset, models such as BERT, LSTM, and a GRU-based approach are trained and evaluated. The application supports both bulk review analysis and single review prediction, making it suitable for different user needs. The output is presented in a clear format, including sentiment labels, predicted ratings, and visual summaries through charts. The results indicate that the system can effectively process textual data and extract meaningful insights from user feedback. Such a system can be applied in areas like e-commerce and customer experience analysis, where understanding user opinion is essential.

**Keywords**—Sentiment analysis, star rating prediction, machine learning, deep learning, BERT, LSTM, GRU model, text classification, customer feedback analysis, web-based application.

## I. INTRODUCTION

In recent years, the expansion of online platforms has resulted in a massive surge in user-generated content, especially in the form of reviews and feedback. These reviews often contain useful insights that can help organizations better understand customer preferences, satisfaction levels, and areas that require improvement. However, manually examining such large volumes of textual data is both time-consuming and inefficient. Because of this, there is an increasing demand for automated solutions that can analyze and interpret text accurately. Sentiment analysis addresses this need by determining whether a given

review expresses a positive, negative, or neutral opinion [11][12][20]. When combined with rating prediction, it becomes even more valuable, as it helps in identifying not just the sentiment but also the intensity of user opinions [13][18]. The aim of this project is to develop a system that simplifies this process by applying modern computational techniques to extract meaningful insights from large datasets.

Over time, advancements in machine learning and deep learning have significantly changed the way textual data is analyzed. Earlier methods relied heavily on manual feature engineering and simple algorithms, which often struggled to capture the true meaning of language [20]. In contrast, recent models are better equipped to understand context, relationships between words, and subtle linguistic patterns [5][8]. In this work, different types of models are brought together, including transformer-based approaches and recurrent neural networks, to improve prediction accuracy [2][6][3]. Each model contributes differently, making it possible to compare their strengths and limitations. By training these models on real-world review data, the system learns underlying patterns and produces reliable predictions [15][16]. This combination of techniques not only improves overall performance but also provides deeper insights into how various approaches handle sentiment-related tasks.

The application is designed in such a way that users can interact with it easily, even without technical expertise. Features such as user registration, login, dataset upload, and result visualization are integrated into a simple interface. Users can analyze large sets of reviews or test individual inputs depending on their requirements. Once the dataset is uploaded, it is preprocessed and transformed before being used for training [4][14]. After model training, performance metrics are displayed to help users understand how well the models are performing. The results are also presented visually, which makes interpretation

easier. This design ensures that the system is not only functional but also accessible to a wider range of users.

Another useful feature of the system is its ability to present results through visual representations. Graphical outputs make it easier to observe trends and patterns in the data. For instance, predicted ratings and sentiment distributions are displayed using charts, allowing users to quickly grasp the overall outcome. This is especially helpful in real-world scenarios where continuous monitoring of customer feedback is required [17]. The system also allows users to input custom reviews, making it flexible and adaptable to different use cases. By combining accurate predictions with clear visual outputs, the project offers a practical solution for analyzing textual data effectively [9][10][19].

## II. RELATED WORK

**Jacob Devlin et al., [2019] [1]** Devlin and his team introduced BERT, a revolutionary model that applies bidirectional training of Transformers for language understanding tasks. Unlike earlier models, BERT reads text in both directions, enabling deeper contextual awareness. This approach significantly improved performance across tasks such as question answering and sentiment analysis. The authors demonstrated that pre-training on large corpora followed by fine-tuning yields strong results. Their work reduced the need for task-specific architectures. BERT quickly became a foundational model in NLP research. It also influenced the development of many derivative models. Overall, the study marked a major shift toward pre-trained language models.

**Sepp Hochreiter et al., [1997] [2]** Hochreiter and Schmidhuber proposed the Long Short-Term Memory (LSTM) network to overcome limitations of traditional recurrent neural networks. Their model introduced memory cells and gating mechanisms to preserve long-term dependencies. This innovation addressed the vanishing gradient problem effectively. LSTM networks became highly suitable for sequential data processing tasks. The architecture enabled better learning in applications like speech recognition and sentiment analysis. Their work laid the foundation for many later deep learning models. It remains one of the most influential contributions in sequence modeling. Even today, LSTM is widely used in NLP systems.

**Ashish Vaswani et al., [2017] [3]** Vaswani and colleagues introduced the Transformer architecture, which relies entirely on attention mechanisms instead of recurrence. This design allowed parallel processing of sequences, greatly improving

efficiency. The model captures relationships between words regardless of their position in the sequence. Their approach outperformed traditional RNN-based models in translation tasks. The concept of self-attention became central to modern NLP systems. Transformers also reduced training time significantly. This work directly influenced models like BERT and GPT.

**Tomas Mikolov et al., [2013] [4]** Mikolov and his team proposed efficient methods for learning word embeddings, commonly known as Word2Vec. Their approach used neural networks to represent words in continuous vector space. These embeddings capture semantic and syntactic relationships between words. The model demonstrated that similar words appear closer in vector space. It significantly improved computational efficiency compared to earlier techniques. Word2Vec became widely adopted in various NLP tasks. The research also inspired further advancements in embedding techniques. Their contribution is essential in modern language representation learning.

**Yoon Kim et al., [2014] [5]** Kim explored the use of convolutional neural networks (CNNs) for sentence classification tasks. The study showed that CNNs can effectively capture local features in text data. By applying convolution and pooling operations, the model extracts meaningful patterns. The approach achieved strong performance on multiple benchmark datasets. It demonstrated that simple architectures can compete with more complex models. The research highlighted the usefulness of pre-trained word embeddings. It also influenced further work in text classification using deep learning. This study remains a key reference in NLP applications of CNNs.

## III. DATASET DETAILS

The dataset used in this study consists of textual data collected from multiple publicly available sources, including online reviews, social media platforms, and benchmark corpora widely used in natural language processing research. These datasets typically include labeled text samples categorized into sentiment classes such as positive, negative, and neutral. In some cases, finer-grained annotations like aspect-based sentiment labels or emotion categories are also included. The data undergoes preprocessing steps such as tokenization, removal of stop words, normalization, and padding to ensure consistency across inputs. Additionally, pre-trained word embeddings or token representations are applied to convert text into numerical form suitable for deep learning models. The dataset is usually divided into training, validation, and testing subsets to support model

development and unbiased performance evaluation. Care is taken to maintain class balance wherever possible to prevent biased predictions and ensure robustness across different text scenarios.

Furthermore, the dataset varies in size depending on the specific model requirements, ranging from a few thousand to millions of text samples. Large-scale datasets are particularly useful for training transformer-based models like BERT, while smaller datasets are often sufficient for models such as CNNs or LSTMs when combined with pre-trained embeddings. Data quality is an important consideration, and noisy or irrelevant entries are filtered out during preprocessing. In some cases, augmentation techniques such as synonym replacement or paraphrasing are applied to enhance dataset diversity. The dataset may also include metadata such as timestamps, user information, or domain-specific tags, which can be leveraged for advanced analysis. Proper evaluation metrics such as accuracy, precision, recall, and F1-score are used to measure model performance on the dataset, ensuring that the results are reliable and generalizable across different real-world applications.

#### IV. PROPOSED METHODOLOGY

The methodology followed in this work begins with collecting relevant textual data and preparing it for analysis. Before feeding the data into any model, several preprocessing steps are applied to improve its quality. These steps include removing unnecessary characters, converting all text into a uniform format, eliminating commonly used words that do not add much meaning, and splitting the text into smaller components through tokenization. Once the data is cleaned, it is converted into numerical form using embedding techniques so that machine learning models can process it. The dataset is then divided into separate portions for training and testing. Based on the problem requirements, suitable models such as CNN, LSTM, or transformer-based architectures are selected.

After preparing the data, the next step involves training the chosen models and adjusting their parameters to achieve better performance. Techniques like dropout and regularization are used to avoid overfitting and ensure that the models generalize well to new data. The trained models are evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Further improvements are made by tuning hyperparameters and testing different configurations. Finally, the best-performing model is applied to unseen data to verify its reliability. This step confirms whether the

system can consistently produce accurate sentiment classifications and rating predictions in real-world scenarios.



Figure [1] : Sentiment Analysis System Architecture

Figure[1] This diagram illustrates the overall workflow of the sentiment-based rating prediction system. It begins with the user interacting through a web interface, followed by dataset upload and preprocessing. The processed data is then passed through multiple models such as BERT, LSTM, and an enhanced GRU for training and evaluation. After model processing, the system performs sentiment analysis and predicts ratings. Finally, the results are presented through batch processing, single review analysis, and visualization, producing the final predicted sentiments and ratings.

#### V. RESULT AND DISCUSSION

The system was implemented successfully and evaluated using both training and testing datasets to measure its effectiveness in sentiment classification and rating prediction. After preprocessing and loading the dataset, different deep learning models, including BERT, LSTM, and a GRU-based model, were trained and tested. The performance of these models varied noticeably. Among them, the GRU-based model achieved the highest accuracy, indicating its strong ability to capture patterns in sequential text data. BERT showed moderate performance, while LSTM produced comparatively lower accuracy. The system also supports batch processing, where multiple reviews are analyzed together. Each review is assigned a sentiment label along with a corresponding star rating. To make the results easier to understand, a pie chart is generated to show the distribution of predicted ratings. In addition, the single review feature allows users to enter custom text and instantly receive predictions. This makes the system practical for both large-scale analysis and quick individual evaluations.



individual feedback and understanding its sentiment polarity.



**Figure [6] : Single Review Sentiment Analysis Result**

Figure [6] shows the output of sentiment analysis for a single user-entered review. The user inputs a comment, and the system processes it using the trained model (BERT & LSTM) to determine the sentiment. The result is displayed in a table format, showing the original review text, the predicted sentiment (e.g., Positive), and a corresponding rating. This feature helps users quickly understand the emotional tone and quality of individual feedback.

### DISCUSSION

The findings suggest that combining different deep learning approaches can improve the overall effectiveness of sentiment analysis systems. The better performance of the GRU-based model may be due to its ability to retain contextual information more efficiently in this particular dataset. Although BERT is generally known for strong contextual understanding, its performance here might be influenced by factors such as dataset size or parameter settings. LSTM, while useful for sequence modeling, appears to require further tuning to achieve better results. Another important aspect of the system is the way results are presented. Visual representations, such as charts and structured tables, make it easier to interpret the output. This improves usability, especially for non-technical users. The system can be further enhanced by using larger datasets, experimenting with hybrid models, and refining training parameters. Overall, the approach demonstrates how deep learning can be applied effectively to extract insights from textual data in real-world scenarios.

### VI. CONCLUSION

This work presents the development of a sentiment analysis system capable of predicting both the polarity of user reviews and their corresponding ratings. By integrating multiple deep learning models into a single platform, the system offers a practical way to analyze textual data efficiently.

The application enables users to upload datasets, train models, and view predictions through a simple and user-friendly interface. The experimental results show that model performance can vary depending on the approach used, highlighting the importance of proper model selection and tuning. The inclusion of visualization features makes it easier to understand the results and identify patterns in user feedback. In addition, the ability to process both multiple reviews and single inputs adds flexibility to the system. With further improvements, such as better parameter tuning and larger datasets, the system can be extended for broader real-world applications.

### REFERENCES

1. Jacob Devlin and colleagues (2019) introduced the BERT model for deep bidirectional language understanding at NAACL-HLT.
2. Sepp Hochreiter and Jürgen Schmidhuber (1997) proposed the Long Short-Term Memory (LSTM) architecture in *Neural Computation*.
3. Kyunghyun Cho et al. (2014) developed RNN encoder-decoder models for machine translation, presented at EMNLP.
4. Yinhan Liu (2019) discussed methods for fine-tuning BERT in a text classification context (arXiv).
5. Tomas Mikolov and team (2013) proposed efficient techniques for learning word embeddings (arXiv).
6. Ashish Vaswani et al. (2017) introduced the Transformer architecture in "Attention is All You Need" at NIPS.
7. Yoon Kim (2014) applied convolutional neural networks to sentence classification tasks at EMNLP.
8. Jeffrey Pennington, Richard Socher, and Christopher Manning (2014) proposed GloVe word embeddings at EMNLP.
9. Zichao Yang et al. (2016) presented hierarchical attention networks for document classification at NAACL.
10. Duyu Tang and collaborators (2015) explored document modeling using gated recurrent networks for sentiment analysis at EMNLP.
11. Huimin Xu et al. (2019) provided a survey on deep learning methods for sentiment analysis in *Wiley Interdisciplinary Reviews*.
12. Lei Zhang and team (2018) conducted a comparative review of deep learning techniques in *Information Fusion*.

13. Yuan Ma et al. (2018) studied aspect-based sentiment analysis using commonsense embeddings at AAAI.
14. Chi Sun and colleagues (2019) examined strategies for fine-tuning BERT models at CCL.
15. Bo Wang and Wei Lu (2020) combined CNN and Bi-GRU models for sentiment classification in *Neurocomputing*.
16. Fei Jiang and Shuai Yu (2021) applied deep learning methods for analyzing online reviews in *Expert Systems with Applications*.
17. Soujanya Poria et al. (2017) studied context-aware sentiment analysis in user-generated videos at ACL.
18. Ali Hassan and Arif Mahmood (2017) explored deep learning approaches for short text sentiment analysis at COLING.
19. Qian Qian et al. (2017) introduced a co-attention mechanism for sentiment classification at ACL.
20. Richa Sharma and Lipika Dey (2012) compared feature selection and machine learning methods for sentiment analysis in ACM publications.
21. Babburi, S. Lightweight Distributed Provenance Framework for Edge and IoT Data Systems.
22. Gaddam, S. From Fixed Specifications to Self-Adapting Systems: A Machine Learning Perspective on Software Engineering.
23. Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. *Applied Research for Growth, Innovation and Sustainable Impact*, 377–384. <https://doi.org/10.1201/9781003684657-63>
24. Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
25. Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.
26. Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.
27. Viswanathan, V. Generative AI for Smarter Workforce Planning and Enterprise Resource Decisions.
28. Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
29. Maturi, S. Y. (2023). Crowdsourced frontier: Unveiling autonomous adversarial cybercapabilities via open AI competition. *International Journal of Intelligent Systems and Applications in Engineering*, 11(1s), 275–284.
30. Venkata Ramana, P. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. *International Journal of Research in Information Technology and Computing*, 8(4).
31. Srikanth Kavuri. (2025). AI-DRIVEN TEST AUTOMATION FRAMEWORKS: ENHANCING EFFICIENCY AND ACCURACY IN SOFTWARE QUALITY ASSURANCE. *International Journal of Applied Mathematics*, 38(10s), 699–710. <https://doi.org/10.12732/ijam.v38i10s.990>
32. Gajula, S. (2024). Cybersecurity risk prediction using graph neural networks. *Journal of Information Systems Engineering and Management*.
33. Shashank, A. (2025). AI-Enhanced ETL Processes: Leveraging Artificial Intelligence for Optimized Data Integration Systems. *Journal Of Multidisciplinary*, 5(8), 219-225.
34. Kandula, S. T. R., Susarla, R. S., & Boyapati, P. K. (2025, July). Enhanced Cyber Security Using Global Local Artificial Neural Network Based Intrusion Detection in Big Data Environment. In 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC) (pp. 426-431). IEEE.
35. Boyapati, P. K. Building a centralized data operations hub for healthcare enterprise integration. *IJSAT-Int. J. Sci. Technol.* 16 (2). <https://doi.org/10.71097/IJSAT.v16.i2.3219>