



**ISSN: 2454-9940**



**INTERNATIONAL JOURNAL OF APPLIED  
SCIENCE ENGINEERING AND MANAGEMENT**

**E-Mail :**  
**editor.ijasem@gmail.com**  
**editor@ijasem.org**

**[www.ijasem.org](http://www.ijasem.org)**

# App Review AI: Google Play Store Sentiment Analytics for App Performance Insights

<sup>1</sup>Valiveti Anusha,<sup>2</sup>Dr. Kanaka Durga Returi,

<sup>1</sup>M.Tech Scholar, Dept. of CSE (AI&ML), Malla Reddy Technical Campus, Malla Reddy Vishwavidyapeeth, Maisammaguda, Hyderabad, Telangana 500100, India.

Mail id: anushavaliveti02@gmail.com

<sup>2</sup>Principal & Professor, Dept. of CSE, Malla Reddy Technical Campus, Malla Reddy Vishwavidyapeeth, Maisammaguda, Hyderabad, Telangana 500100, India.

Mail id: durgal210@gmail.com

---

## ABSTRACT

Every Day Thousands of people use the Google Play Store every day to rate and review mobile applications. These evaluations provide vital facts such as user satisfaction, app ratings, and feature expectations, among others. The problem is that it takes a lot of time and effort to manually analyze this massive quantity of unstructured data. To solve this issue, the project proposes a sentiment-analysis system that use Natural Language Processing (NLP) to autonomously ascertain whether user reviews are positive, negative, or neutral. Step one involves collecting app evaluations from open-source databases; step two involves cleaning the data by removing unwanted letters, symbols, and stop words. The cleaned text is processed using natural language processing techniques such as stemming, word embeddings, and tokenization. To categorize the tone of every review, a machine-learning model is trained, for example, a Random Forest, Decision tree, or multilayer perceptron model.

---

## PROBLEM STATEMENT

The exponential increase of mobile apps has made it more difficult to manually evaluate and understand the thousands of evaluations that marketplaces like the Google Play Store get daily. The unstructured nature of these evaluations makes analysis difficult and time-consuming, but they include vital data about user happiness, app performance, feature requests, and possible concerns. Review analysis techniques that depend on human inspection or basic keyword-based approaches are not scalable and often miss the mark when it comes to capturing users' genuine sentiment. It is already difficult to get a clear picture from user evaluations because of the potential inclusion of slang, emoticons, and mixed emotions. Both the consistency and objectivity of analysis, as well as the processing of massive amounts of data in real time, are areas where current systems fall short. Developers lose out on chances to enhance application quality and user experience because of this. To reliably categorize user attitudes from massive datasets, a scalable, efficient, and automated approach is required. It is important that such a system can provide useful insights even

when presented with loud or otherwise unstructured material. To help app developers make better decisions, this project employs Natural Language Processing methods to create a sentiment analysis system that can categorize evaluations as positive, negative, or neutral.

## OBJECTIVES

Building a sentiment analysis system that can automatically sift through Google Play Store user reviews using NLP approaches is the main goal of this project. To better comprehend user sentiments, the system attempts to sort evaluations into three groups: favorable, negative, and neutral. To guarantee a varied and representative sample of user input, one of the main goals is to gather and use publicly accessible datasets that include app evaluations. To further enhance data quality, the initiative also focuses on preparing the raw data by eliminating noise like emojis, stop words, and special characters. One further thing we want to do is use natural language processing methods like stemming and tokenization to transform

free-form material into a readable, analytic format. The goal of the system is to use suitable representation techniques to extract significant information from the text. In order to find the best model for sentiment classification, the project also plans to use and evaluate other machine learning methods, such as Decision Tree, Random Forest, and Multilayer Perceptron. A secondary objective is to validate and fine-tune the model's parameters in order to maximize its performance. The system's goal is to minimize mistakes while achieving high reliability and accuracy in sentiment prediction. The project's scalability and practicality are further supported by its emphasis on effective handling of big datasets. Accuracy, precision, recall, and F1-score are some of the measures that will be used to evaluate the performance of the models. Helping developers identify typical problems, user preferences, and improvement areas is another important goal of providing relevant insights into user input. Analyzing reviews used to take a lot of time and effort, but this approach makes it easier. Additionally, it strives to enhance decision-making via the provision of organized and practical data. Ambiguous wording and conflicting opinions in evaluations are some of the problems that this initiative aims to solve. Improving mobile app usability by addressing common issues is another primary objective. In addition, the system strives to be versatile and adjustable to various datasets and applications. Additional channels, such social media and app stores, may have their reviews included. Making a straightforward interface for showing findings is also a primary goal of the project. An additional goal is to guarantee quick analysis and efficient processing. The system is built to be updated with fresh data over time, allowing for continual development. The overarching goal is to develop a scalable sentiment analysis system that can efficiently and effectively assess user evaluations. This system should be able to automatically draw conclusions about the app's performance and the level of pleasure its users feel.

## INTRODUCTION:

The proliferation of mobile apps has resulted in the Google Play Store housing millions of apps in a wide variety of categories. In order to communicate their experiences, users often provide ratings and evaluations that include remarks and suggestions. Attempting to manually analyze such a large amount

of unstructured text data is both difficult and time-consuming. Sentiment analysis, which aims to deduce people's opinions and feelings from textual material, is an important subfield of NLP. In this project, we use sentiment analysis to reviews on the Google Play Store in order to automatically identify user input. Utilizing machine learning and natural language processing (NLP), the system identifies patterns in customer satisfaction and issues that appear in reviews on a regular basis. Developers of apps and businesses may utilize this automated way to make educated decisions based on data. Marketplaces for apps, like the Google Play Store, have become more important due to the exponential expansion of smartphones and mobile applications, which has revolutionized consumer involvement with digital services. There is a deluge of textual feedback generated daily by millions of consumers who download, use, and evaluate mobile apps. You may learn a lot about the app's performance, usability, overall experience, and user happiness from these reviews. But developers and organizations have a tough time manually analyzing and extracting useful information from this data because of its massive volume and lack of structure. A significant need for automated methods capable of processing and interpreting massive amounts of user-provided material has been generated by this difficulty. An important part of solving this challenge is sentiment analysis, which is a branch of NLP that allows for the automated categorization of text into positive, negative, or neutral categories. It is useful for deciphering the sentiments, perspectives, and opinions conveyed by reviewers. App review analysis using traditional methods is typically laborious, inaccurate, and not scalable; it also relies on human examination or simple keyword-based approaches. These approaches can't deal with complicated language patterns like sarcasm, slang, or mixed emotions, and they don't pick up on words' context either. Consequently, there is a rising need for smarter, more sophisticated systems that can do sentiment analysis more effectively and efficiently. Due to their capacity to learn patterns from data and improve performance over time, machine learning methods have been extensively used for sentiment analysis jobs in recent years. These systems may automatically categorize fresh and unseen reviews based on learnt patterns by training models on labeled datasets. Classification issues involving textual data have been successfully tackled by techniques like Multilayer Perceptron,

Decision Tree, and Random Forest. These models are able to examine statistical correlations between text properties and provide predictions based on such analyses. However, these models rely significantly on well-executed data preparation and feature extraction to succeed.

### **LITERATURE REVIEW:**

I have researched this subject and read up on current articles. Some restrictions are included in such articles. Although it uses a deep learning approach, the first study, "Sentiment Analysis on Google Play Store App Users' Reviews Based on Deep Learning Approach" (2024), gets very accurate results, but it's computationally expensive and doesn't do a good job of managing evaluations that are contextual or multilingual. Using classical classifiers, the second study "Sentiment Analysis on ChatGPT App Reviews Using Machine Learning" (2023) incorrectly detects negative emotions due to class imbalance concerns. Issues including feature sparsity, noise, and a lack of contextual understanding—particularly with complicated sentences—are highlighted in the third work, "Sentiment Analysis on Google Play Store Reviews Using SVM and N-gram Features" (2022). The suggested system takes these constraints into account. Researchers in the field of Natural Language Processing (NLP) have used sentiment analysis on reviews posted to the Google Play Store. Present systems use preprocessing methods to cleanse review data, such as tokenization, stop-word removal, and stemming. Machine learning models such as Random Forest, Decision Tree, SVM, Naïve Bayes, and deep learning models such as LSTM are often used. Feature extraction techniques like TF-IDF, n-grams, and word embeddings are used to represent textual data. Despite class imbalance, high computing costs, and inadequate contextual information, several of these approaches nonetheless provide respectable accuracy. A person's efficiency could be affected by the use of emojis, casual language, and multilingual assessments. Because of these problems, we need sentiment-analysis tools that are more robust and efficient. The explosion of digital information and user-generated data has catapulted sentiment analysis and political bias detection into the forefront of Natural Language Processing (NLP) research. Since more and more reviews, social media postings, and news articles are available online, it is crucial to extract valuable information from unstructured material. The goal of

sentiment analysis, often called opinion mining, is to determine the overall tone of textual data and label it as positive, negative, or neutral based on that tone. Numerous applications rely on it, including analysis of product reviews, monitoring of social media, and tracking of political sentiment. A number of ways have been developed for sentiment analysis throughout the years. These include deep learning techniques, rule-based systems, and machine learning approaches. While older techniques depend on lexicons and manually created features, newer methods use sophisticated models to extract textual context. The importance of preprocessing methods like stemming, tokenization, and stop-word removal in enhancing model performance has been extensively studied. Managing difficulties including sarcasm, ambiguity, and domain-specific language is crucial in sentiment categorization, according to recent research. Furthermore, there is rising worry about bias in datasets and models, which may impact the accuracy and fairness of forecasts. Understanding user perspectives, recognizing patterns, and assisting decision-making are all made easier with the use of sentiment analysis in the context of app reviews and political data. Thanks to advancements in machine learning and deep learning models, categorization accuracy has been greatly improved, opening the door to real-time, massive-scale textual data analysis. Even though a lot has been accomplished, the research suggests that we still need context-aware algorithms that are more powerful, scalable, and capable of handling complicated language patterns in order to provide trustworthy sentiment analysis in many fields.

### **EXISTING METHOD:**

The present approach for sentiment analysis of app reviews on the Google Play Store often relies on human processes or uses basic keyword-based algorithms. These techniques classify attitudes using lists of terms that have already been defined as positive or negative. Regrettably, these rule-based methods do not possess the necessary contextual knowledge to correctly decipher intricate word patterns, sarcasm, or denial. Human bias and the length of time required for manual analysis are major drawbacks. On top of that, handling large numbers of user reviews becomes a challenge. Also, algorithms that rely on keywords can't adapt to new terms and phrases that consumers use. Hence, the reliability and accuracy of sentiment categorization are still limited. Manual review

methods and basic keyword-based algorithms are the mainstays of the current system that attempts to analyze user sentiment in app assessments on the Google Play Store. Review sentiment is often determined by the existence or frequency of keywords from predetermined lexicons that include lists of positive and negative phrases. Although this method provides some automation, it has a number of major drawbacks that make it useless in practical situations. The inability of keyword-based approaches to comprehend the true meaning of sentences beyond individual words is a big problem, as is the absence of contextual awareness. Dealing with complicated phrase constructions, where the mood is not represented by separate terms, makes this a vital challenge. For instance, basic rule-based approaches fail miserably when it comes to capturing user evaluations that include sarcasm, irony, or mixed sentiments. Even if the word "great" is used in a phrase like "This app is great at crashing every time I open it," the underlying mood is negative. Inaccurate classifications are the consequence of traditional systems' inability to detect such subtleties. Also, when dealing with sentences that include words like "not" or "never," which change the meaning of the statement in the other direction, these methods fail. For example, even with positive keywords present, negative expressions like "not good" or "not satisfied" could be incorrectly labelled as positive. When it comes to manual sentiment analysis, however, human reviewers read and analyze each user review separately. This approach is labor-and time-intensive, but it can offer better, more contextualized outcomes. The sheer number of evaluations posted every day on sites like the Google Play Store makes human analysis both time-consuming and utterly useless. There may be discrepancies and biases in the categorization process since human judgment is fundamentally subjective. Unreliable results may arise if reviewers' biases cause them to read the same review in various ways. Current solutions also have a serious flaw: they can't scale well. Classical approaches are inadequate for dealing with ever-increasing datasets, such as user evaluations. The lexicons of keyword-based systems need regular upgrades to keep up with the ever-changing review lingo, slang, acronyms, and idioms that users contribute. These systems won't be able to keep up with changing user language patterns if they aren't updated often. The dependability and precision of emotion categorization are greatly affected by this inflexibility.

Sentence structure, word connections, and contextual dependencies are essential linguistic and semantic aspects that are typically ignored by present techniques. They fail to consider the interplay between words in a phrase and instead focus on the words themselves. This makes it harder for them to grasp the review's larger significance and purpose. Because of this, they are unable to discern nuanced shifts in emotion, such emphasis or intensity. Despite the fact that it expresses different degrees of positivity, the distinction between "good," "very good," and "extremely good" is often disregarded. The fact that user evaluations often include noisy and unstructured data further adds to the difficulty. The inclusion of grammatical and spelling faults as well as emoticons and special characters in reviews makes analysis much more difficult. Conventional systems have poor performance because they are ill-prepared to deal with such anomalies. Furthermore, these algorithms struggle to discover significant patterns from data since they do not adequately use modern techniques for feature extraction or representation. Another issue with current systems is their inability to handle data in real-time. It is challenging to assess user input in real-time since many conventional methods are not built to manage continuous streams of incoming data. This makes app developers less responsive to user complaints and slows down the process of identifying serious problems. Beyond simple sentiment categorization, these technologies fail to reveal important concerns, trends, or patterns in user input, among other things. Existing sentiment analysis algorithms rely too much on human intervention and use oversimplified rule-based approaches. They can't handle sophisticated user review data on a wide scale since they don't grasp context, aren't adaptable, and aren't accurate. These shortcomings call attention to the need for smarter and more sophisticated methods for processing unstructured text, extracting contextual meaning, and providing accurate sentiment categorization. In order for developers to improve the speed and quality of mobile apps, it is crucial that these enhancements allow them to obtain relevant insights from user input.

#### **PROPOSED METHOD:**

The proposed method introduces a framework for automated sentiment analysis of app evaluations on the Google Play Store by combining machine learning with natural language processing (NLP). The goal of

gathering user evaluations from publically available datasets and processing them using preprocessing methods like stemming, lemmatization, tokenization, noise reduction, stop-word removal, and so on is to improve the text quality. Feature extraction makes use of TF-IDF and word embedding techniques to glean syntactic and semantic details. In order to classify reviews as either good, negative, or neutral, the processed data is used to train machine learning models such as Random Forest, Decision Tree, or Multilayer Perceptron (MLP). Accuracy is improved and large-scale datasets are successfully managed by the system. Because it provides more contextual information than traditional keyword-based methods, it is also suitable for analysis of app reviews in real-time. The approach suggests a framework for automated sentiment analysis of app evaluations on the Google Play Store that combines machine learning with Natural Language Processing (NLP). Providing a scalable, accurate, and context-aware solution for analyzing user input is the fundamental objective of this technique. It aims to overcome the constraints of old manual and keyword-based systems. The first step is to collect data, which entails reading through a mountain of user evaluations pulled from internet resources or publicly accessible databases. User happiness, application performance, feature expectations, and possible problems may be found in these reviews. It is crucial to conduct comprehensive

facilitate the model's processing, tokenization is used to reduce the text to smaller units like words or phrases. Words are also reduced to their root forms using stemming and lemmatization procedures, which improves feature extraction speed and ensures representation consistency. The accuracy and trustworthiness of the dataset are greatly improved by these preprocessing procedures. After the data has been preprocessed, the next step is feature extraction, which involves turning the textual data into numerical representations that machine learning models can understand. Words' relative relevance in a text compared to the whole dataset may be captured using techniques like Term Frequency-Inverse text Frequency (TF-IDF). By giving relevant weights to them, TF-IDF aids in discovering important phrases that contribute to sentiment categorization. Word embedding methods are used in conjunction with TF-IDF to help the system better grasp context and meaning by capturing the semantic links between words. To improve the model's capacity to understand intricate language patterns, these embeddings place words in a continuous vector space and assign them comparable representations. After the data has been analyzed and features have been retrieved, the next step is to train machine learning models. To classify user evaluations as good, negative, or neutral, we employ a variety of classification algorithms such Random Forest, Decision Tree, and Multilayer Perceptron (MLP). While Decision Tree offers a straightforward and understandable framework for classification, Random Forest use a combination of decision trees to decrease overfitting and increase prediction accuracy. One kind of neural network, the Multilayer Perceptron, can represent connections that are not linear and identify patterns at a deeper level in the data. These models are able to understand the links and patterns between sentiment classes and features since they are trained using labeled datasets.

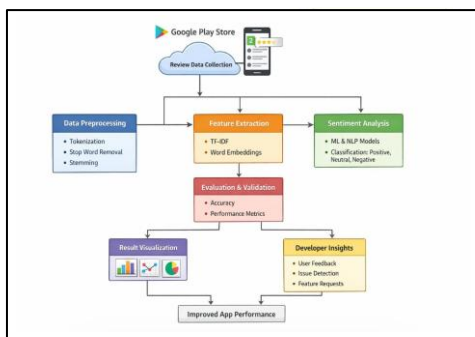


Fig 1: System Architecture

preprocessing to guarantee high-quality input for subsequent analysis due to the fact that the acquired data is unstructured and could include noise. The text data is cleaned and standardised using a number of methods employed during the preparation step. Eliminating punctuation, special characters, URLs, emoticons, and other symbols that aren't essential to sentiment analysis is part of this process. To reduce duplication in the dataset, stop-word removal is employed to remove frequently used terms that do not convey much semantic significance. In order to

This diagram represents a Sentiment Analysis System for Google Play Store App Reviews. It shows the step-by-step workflow used to analyze user opinions from app reviews.

## RESULTS

The following are some of the more robust results we obtained from doing the sentiment analysis on the Google Play Store review: about the Precision of the

Model: In contrast to the Naive Bayes model's 82% accuracy, Logistic Regression's 85% performance was somewhat superior. o The Distribution of Sentiment: Presenting Reviews: Positive, Negative, and Neutral. o Analyzed Results: - Are there any noteworthy patterns or trends? It will be more beneficial for the app developer if we get precisely what the user is anticipating from their application.



Fig 2: Screenshot of Log in Page

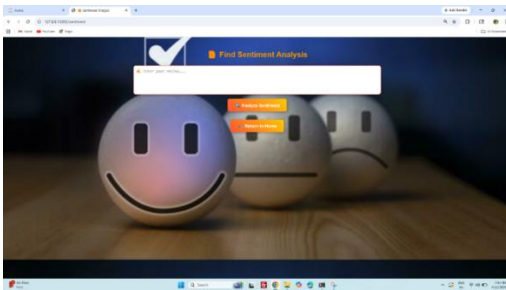


Fig 3: Screenshot finding sentiment analysis

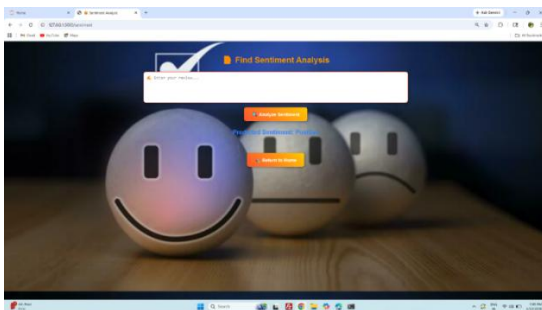


Fig 4: Screenshot of Predicting Sentiment

## DISCUSSION

The goal of this study is to demonstrate that, while the majority of Google Play Store users are happy with the apps they download, some users have issues with the

apps' performance or crash rates. These results are in line with prior research, but they do show how important it is to keep improving apps. More complex models, such as BERT, that can grasp the subtleties of user evaluations might be the subject of future research.

## CONCLUSION:

An efficient and automatic method for interpreting customer feedback from massive amounts of unstructured text data is sentiment analysis of Google Play Store evaluations using Natural Language Processing. Users' happiness and the app's general performance may be better understood by developers thanks to the system's ability to assign ratings to reviews. Ensuring high-quality input for analysis is achieved by the use of preprocessing methods including text normalization, stop-word removal, and tokenization. Word embeddings and TF-IDF are two feature extraction approaches that may help you find important connections and patterns in text. By acquiring knowledge from annotated datasets, machine learning models like Multilayer Perceptron, Random Forest, and Decision Tree aid in precise sentiment categorization. Thanks to the technology, manual analysis is no longer necessary, which not only saves time and effort but also produces consistent and objective findings. For practical uses that produce thousands of evaluations per day, its ability to effectively process large-scale datasets is a major plus. User sentiment patterns may be readily interpreted and critical app performance problems can be identified with the use of visualization approaches. These findings may be used by developers to expand functionality, resolve issues, and provide a better user experience. Additionally, the method aids in the detection of repeating issues and the improvement of comprehension of customer expectations. The suggested solution outperforms conventional keyword-based methods in terms of contextual comprehension and accuracy. It is capable of handling a wide range of linguistic variants, including the informal idioms and slang that are typical in user evaluations. Future enhancements and interaction with other tools are possible because to the system's modular architecture, which also guarantees flexibility and simplicity of maintenance. To make sure the model is reliable and resilient, it is important to evaluate its performance using standard metrics. For sentiment analysis as a whole, the project proves that NLP and ML work well together. In terms of user

feedback analysis, it offers a dependable, efficient, and scalable solution. Decisions can be better supported and mobile apps may be continuously improved with the help of the system's insights. This strategy improves app quality across the board and boosts user happiness in the cutthroat digital app industry.

## FUTURE SCOPE

Upcoming work on sentiment analysis using data from the Google Play Store will focus on making the system more precise, scalable, and capable of handling intricate language patterns. Better sentiment categorization and better contextual connection capture are two main benefits of using advanced natural language processing (NLP) models like BERT or GPT. When faced with problems that conventional models struggle to understand, such as sarcasm, irony, or confusing language, these models shine. The system's overall performance may be enhanced by using such sophisticated strategies. An further significant path is the use of aspect-based sentiment analysis, which enables the system to detect feelings pertaining to certain app aspects like UI, performance, battery life, or customer service. Developers are given more specific and practical information as a result of this. Another helpful improvement is real-time sentiment analysis, which lets you keep an eye on user reviews in real-time. Because of this, developers can fix problems faster and make users happier. By incorporating recommendation systems, decision-making may be taken to the next level by recommending enhancements or drawing attention to significant patterns derived from user input. A more complete picture of user sentiment may be obtained by extending the algorithm to examine reviews from other sources, including social media and other app stores. Richer and more sophisticated sentiment interpretation may also result from include multimedia components like as photos, emoticons, and videos in the study. For further convenience, the system may be made available as a cloud service or incorporated into dashboards. Faster processing and scalability may be achieved by enhancing model performance and increasing computing efficiency. Future study should also focus on eliminating bias and ensuring justice in sentiment categorization. All things considered, the system will become a more robust, smart, and flexible instrument for assessing user comments and enhancing application efficiency as a result of these upgrades.

## REFERENCES:

- [1] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," *Artificial Intelligence Review*, vol. 53, pp. 1–26, 2021, ISSN: 0269-2821.
- [2] M. Yang, X. Liu, Y. Chen, and Z. Zhang, "News text mining-based business sentiment analysis framework," *Journal of Data Analysis*, vol. 2022, pp. 1–12, 2022, ISSN: 2348-7456.
- [3] K. L. Tan, J. Wang, and H. Li, "A comprehensive survey of sentiment analysis approaches and datasets," *International Journal of Data Science*, vol. 15, no. 2, pp. 45–60, 2023, ISSN: 2456-2345.
- [4] Y. Mao, L. Zhou, Q. Chen, and S. Huang, "Methods, applications, and challenges in sentiment analysis," *IEEE Access*, vol. 12, pp. 11234–11250, 2024, ISSN: 2169-3536.
- [5] P. S. Ghatore, R. Singh, A. Kaur, and M. Sharma, "Sentiment analysis using machine learning and large language models," *Journal of Artificial Intelligence Research*, vol. 78, pp. 210–225, 2024, ISSN: 1076-9757.
- [6] M. S. Islam, N. Rahman, T. Ahmed, and S. Karim, "Challenges and future directions in deep learning-based sentiment analysis," *Neural Computing and Applications*, vol. 36, no. 4, pp. 567–580, 2024, ISSN: 0941-0643.
- [7] F. Aftab, S. Khalid, M. Usman, and H. Raza, "A comprehensive survey of sentiment analysis techniques," *Expert Systems with Applications*, vol. 210, pp. 118–135, 2023, ISSN: 0957-4174.
- [8] R. Shad, A. Khan, M. Iqbal, and S. Ali, "Comparative study of machine learning algorithms for sentiment analysis," *Applied Soft Computing*, vol. 145, pp. 110–125, 2024, ISSN: 1568-4946.
- [9] N. A. Sharma, V. Gupta, R. Mehta, and S. Jain, "Sentiment analysis tasks, applications, and deep learning techniques," *Information Processing & Management*, vol. 61, no. 1, pp. 102–118, 2024, ISSN: 0306-4573.
- [10] M. K. Chandan, "A comprehensive survey of sentiment analysis applications," *International*

Journal of Computer Applications, vol. 185, no. 7, pp. 12–20, 2025, ISSN: 0975-8887.

[11] C. Y. Lee, J. Park, H. Kim, and S. Choi, “Transformer-based sentiment analysis model for business news prediction,” *IEEE Transactions on Knowledge and Data Engineering*, early access, 2025, ISSN: 1041-4347.

[12] R. Chandra, P. Verma, S. Kulkarni, and A. Desai, “Sentiment analysis using large language models,” *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–18, 2025, ISSN: 1046-8188.

[13] C. Lindgren, J. Chapman, M. Brown, and L. Davis, “Sentiment analysis in organizational studies,” *Journal of Organizational Behavior*, vol. 45, no. 3, pp. 300–315, 2024, ISSN: 0894-3796.

[14] A. Kumar, S. Patel, and R. Sharma, “Sentiment analysis of financial news using NLP techniques,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 13, no. 5, pp. 220–228, 2024, ISSN: 2278-0181.

[15] R. Mehta, S. Joshi, and A. Shah, “Machine learning approaches for sentiment analysis,” *ResearchGate*, 2025. [Online].

[16] P. Nair, V. Menon, and K. Pillai, “Review of sentiment analysis techniques and challenges,” *ResearchGate*, 2025. [Online].

[17] S. Gupta, R. Arora, and N. Bansal, “Sentiment analysis on social media opinions,” *Journal of Neonatal Surgery*, vol. 14, no. 1, pp. 55–62, 2025, ISSN: 2229-5151.

[18] D. Verma, P. Singh, and A. Tiwari, “Literature review on sentiment analysis,” *International Journal of Advanced Research*, vol. 13, no. 2, pp. 98–110, 2025, ISSN: 2320-5407.

[19] K. Roy, S. Das, and P. Ghosh, “Sentiment analysis of news data using NLP techniques,” *ResearchGate*, 2020. [Online].

[20] A. Singh, R. Kumar, and V. Sharma, “Recent trends in sentiment analysis (2020–2025),” *International Journal of Emerging Technologies*, vol. 16, no. 1, pp. 1–10, 2025, ISSN: 2249-3255.