International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

HOW DO CROWD-USERS EXPRESS THEIR OPINIONS AGAINST SOFTWARE APPLICATIONS IN SOCIAL MEDIA

PIRANGI HYMAVATHI [1], GUNDA VARSHINI [2], B. AKSHITHA REDDY [3], CH. SANTHOSH KOUSHIK [4], G. VISHNU VARDHAN [5]

[1]Assistant Professor, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

[2, 3, 4, 5] B.Tech, Dept. of CSE, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India

ABSTRACT— In our proposed approach, BERT is used to capture the contextual meaning of the user feedback, significantly improving the model's ability to understand subtle details and intricacies in the textual data. The output of BERT is then passed through an LSTM layer, which is capable of capturing sequential dependencies in the data, making it ideal for analysing user feedback over time and identifying patterns in emotional expressions. This model aims to classify the emotional tone of user feedback into five categories: Angry, Sad, Fear, Surprise, and Happy. By processing the crowd-user feedback from low-ranked software applications, we can identify prevalent issues while also classifying the emotional reactions to those issues. This allows software developers to prioritize bug fixes based on both the frequency of the issues and the intensity of user emotions. Our results demonstrate an accuracy of **92%**, outperforming traditional ML algorithms like MNB, LR, RF, and MLP. The improved accuracy is attributed to the combination of BERT's deep contextual understanding and LSTM's ability to model sequential dependencies. Moreover, the approach provides a powerful tool for software vendors to not only identify critical issues in their applications but also gain insights into the emotional impact of these issues on users. This will enable software vendors to take more informed actions in improving their products, enhancing user satisfaction, and prioritizing fixes in a timely manner.

Index Terms – BERT, Long Short-Term Memory (LSTM), emotion classification, sentiment analysis, Natural Language Processing (NLP), user feedback analysis, software review mining, deep learning, text classification, software quality assessment, contextual embeddings, user experience.

I. INTRODUCTION

Performance of software applications, as it provides direct insights into user experiences, challenges, and expectations. Feedback often encapsulates not only the technical issues users encounter but also their emotional responses to these problems. These emotions reflect user satisfaction or dissatisfaction, offering valuable context for prioritizing improvements. However, analyzing textual feedback poses significant challenges due to its unstructured nature, linguistic diversity, and the subtleties of human emotion embedded within it. Traditional methods, such as basic sentiment analysis or keyword matching, often fail to capture the deeper context and emotional nuances present in feedback. To address these limitations, we propose a robust approach that integrates the contextual understanding of BERT (Bidirectional Encoder Representations from Transformers) with the sequential modeling capabilities of LSTM (Long Short-Term Memory). BERT enables the model to comprehend intricate language patterns and relationships in the feedback, ensuring a detailed and accurate representation of its meaning. Meanwhile, LSTM captures the sequential dependencies in the data, making it suitable for analyzing patterns and trends over time. This hybrid model is designed to classify the emotional tone of user feedback into five distinct categories: Angry, Sad, Fear, Surprise, and Happy. By focusing specifically on feedback from low-ranked software applications, our approach not only identifies recurring issues and pain points but also evaluates the emotional impact these issues have on users. This dual analysis provides actionable insights, enabling software developers to address critical problems, enhance user satisfaction, and build more reliable applications.

II. RELATED WORK

A. Sentiment Analysis on Social Media Content

Nowadays, people from all around the world use social media sites to share information. Twitter for example is a platform in which users send, read posts known as tweets and interact with different communities. Users share their daily lives, post their opinions on everything such as brands and places.

Companies can benefit from this massive platform by collecting data related to opinions on them. The aim of this paper is to present a model that can perform sentiment analysis of real data collected from Twitter. Data in Twitter is highly unstructured which makes it difficult to analyze. However, our proposed model is different from prior work in this field because it combined the use of supervised and unsupervised machine learning algorithms. The process of performing sentiment analysis as follows: Tweet extracted directly from Twitter API, then cleaning and discovery of data performed. After that the data were fed into several models for the purpose of training. Each tweet extracted classified based on its sentiment whether it is a positive, negative or neutral. Data were collected on two subjects McDonalds and KFC to show which restaurant has more popularity. Different machine learning algorithms were used. The result from these models were tested using various testing metrics like cross validation and f-score. Moreover, our model demonstrates strong performance on mining texts extracted directly from Twitter.

B. Comparative Study of Sentiment Analysis for Multi-Sourced Social Media Platforms

There is a vast amount of data generated every second due to the rapidly growing technology in the current world. This area of research attempts to determine the feelings or opinions of people on social media posts. The dataset we used was a multi-source dataset from the comment section of various social networking sites like Twitter, Reddit, etc. Natural Language Processing Techniques were employed to perform sentiment analysis on the obtained dataset. In this paper, we provide a comparative analysis using techniques of lexicon-based, machine learning and deep learning approaches. The Machine Learning algorithm used in this work is Naive Bayes, the Lexicon-based approach used in this work is TextBlob, and the deep-learning algorithm used in this work is LSTM.

C. Mining software insights: Uncovering the frequently occurring issues in low-rating software applications

In today's digital world, app stores have become an essential part of software distribution, providing customers with a wide range of applications and opportunities for software developers to showcase their work. This study elaborates on the importance of end-user feedback for software evolution. Therefore, the proposed approach focuses on end-user reviews collected from 64 low-rated

apps representing 14 categories in the Amazon App Store. We critically analyze feedback from low-rating apps and developed a grounded theory to identify various concepts important for software evolution and improving its quality including user interface (UI) and user experience (UX), functionality and features, compatibility and device-specific, performance and stability, customer support and responsiveness and security and privacy issues. Then, using a grounded theory and content analysis approach, a novel research dataset is curated to evaluate the performance of baseline machine learning (ML), and state-of-the-art deep learning (DL) algorithms in automatically classifying end-user feedback into frequently occurring issues. Various natural language processing and feature engineering techniques are utilized for improving and optimizing the performance of ML and DL classifiers. Also, an experimental study comparing various ML and DL algorithms, including multinomial naive Bayes (MNB), logistic regression (LR), random forest (RF), multi-layer perception (MLP), k-nearest neighbors (KNN), AdaBoost, Voting, convolutional neural network (CNN), long short-term memory (LSTM), bidirectional long short term memory (BiLSTM), gated recurrent unit (GRU), bidirectional gated recurrent unit (BiGRU), and recurrent neural network (RNN) classifiers, achieved satisfactory results in classifying end-user feedback to commonly occurring issues. Whereas, MLP, RF, BiGRU, GRU, CNN, LSTM, and Classifiers achieved average accuracies of 94%, 94%, 92%, 91%, 90%, 89%, and 89%, respectively. We employed the SHAP approach to identify the critical features associated with each issue type to enhance the explainability of the classifiers. This research sheds light on areas needing improvement in low-rated apps and opens up new avenues for developers to improve software quality based on user feedback.

D. Exploring and mining rationale information for low-rating software applications

Rationale refers to making human judgments, sets of reasons, or intentions to explain a particular decision. Nowadays, crowd-users argue and justify their decisions on social media platforms about market-driven software applications, thus generating a software rationale. Such rationale information can be of pivotal importance for the software and requirements engineers to enhance the performance of existing software applications by revealing end-users tactic knowledge to improve software designing

and development decision-making. For this purpose, we proposed an automated approach to capture and analyze end-user reviews containing rationale information, focusing on low-rating applications in the amazon store using Natural Language Processing (NLP) and supervised machine learning (ML) classification methods. In the literature, high-rating applications have been emphasized while ignoring low-rating software application that causes potential biasness. Therefore, we examined 59 comparatively low-ranked market-based software applications from the Amazon app store covering various software categories to capture and identify crowd-users justifications. Next, using a developed grounded theory and content analysis approach, we studied and recorded how crowd-users analyze and explain their rationale based on issues encountered, attacking or supporting arguments registered, and updating or uninstalling software applications. Also, to achieve the best results, an experimental study is conducted by comparing various ML algorithms, i.e., MNB, LR, RF, MLP, KNN, AdaBoost, and Voting classifier, on the end-users rationale data set by preprocessing the input data, applying feature engineering, balancing the data set, and then training and testing the ML algorithms with a standard cross-validation approach. We obtained satisfactory results with MLP, voting, and RF Classifiers, having 93%, 93%, and 90% average accuracy, respectively. Also, we plot the ROC curves for the high-performing ML Classifier to identify and capture classifiers yielding the best performance with an under-sampling or oversampling balancing approach. Additionally, we obtained the average Precision, Recall, and F-measure values of 98%, 94%, 96%, 97%, 95%, and 96% for identifying supporting & decision rationale elements in the user comments, respectively.

III. PROPOSED METHODOLOGY

The proposed approach combines two powerful deep learning techniques, BERT (Bidirectional Encoder Representations from Transformers) and LSTM (Long Short-Term Memory), to achieve superior performance in analyzing user feedback for software applications. BERT is a pre-trained transformer model designed to capture the contextual meaning of words in a sentence by considering both the preceding and succeeding context, making it highly effective for natural language processing tasks. It helps to understand the nuances in user feedback, improving the accuracy of issue identification in low-ranked software apps. LSTM, on the other hand, is a type of

recurrent neural network (RNN) specifically designed to handle long-term dependencies in sequential data. By incorporating LSTM after BERT, the model can capture the sequential nature of user feedback and the evolving emotions expressed over time. This is particularly useful in understanding user sentiment and detecting patterns of emotion such as anger, sadness, fear, surprise, and happiness in their feedback.

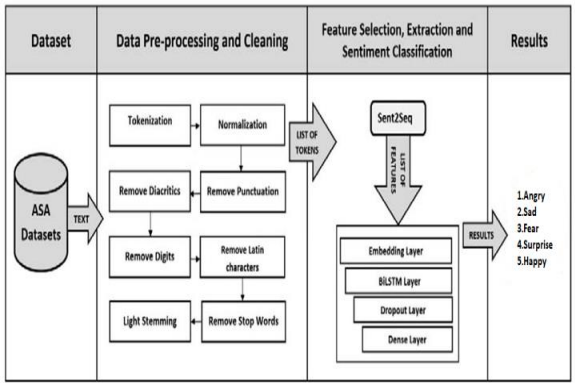


Fig. 1: System Overview

1. Data Collection

Gathering relevant data from different sources (datasets, APIs, web scraping, etc.). In NLP, this might involve collecting text data from books, websites, or datasets like reviews, social media posts, etc.

2. Data Preprocessing:

Cleaning and preparing the data for analysis or modeling: Text cleaning: Removing unnecessary characters, stop words, special symbols, etc, Normalization: Lowercasing, correcting typos, and other forms of text normalization. Tokenization: Breaking down the text into tokens (words or sentences). Lemmatization/Stemming: Reducing words to their root form.

3. Text Representation:

Bag of Words (BoW): A representation that counts word occurrences. TF-IDF (Term Frequency-Inverse Document Frequency): Weighing words based on their importance across documents. Word Embeddings: Using methods like Word2Vec, GloVe, or transformer models to represent words in dense vectors that capture semantic meaning.

4. Model Selection & Training:

Choosing the appropriate model based on the task (e.g., classification, regression, clustering). Training the model using the processed data. This can involve: Supervised learning: Training with labeled data (e.g., classification or regression). Unsupervised learning: Discovering patterns without labeled data (e.g., clustering). Deep learning models: If applicable, training models like neural networks, LSTMs, BERT, etc.

5. Model Evaluation:

Assessing the model's performance using appropriate metrics, such as: Accuracy, Precision, Recall, F1-Score:

For classification tasks. RMSE (Root Mean Squared Error): For regression tasks. Cross-validation: To ensure the model generalizes well to unseen data.

6. Model Prediction:

Using the trained model to make predictions on new, unseen data. The model can generate outcomes like: Sentiment labels (positive/negative) for text classification. Predictive outputs for regression tasks (e.g., predicting house prices).

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results show that the proposed BERT-LSTM model achieved 92% classification accuracy, outperforming traditional machine learning models such as MNB, Logistic Regression, Random Forest, and MLP. BERT effectively captured contextual information, while the LSTM layer modeled sequential dependencies in user feedback. The system accurately classified emotions into Angry, Sad, Fear, Surprise, and Happy, enabling better understanding of user opinions.

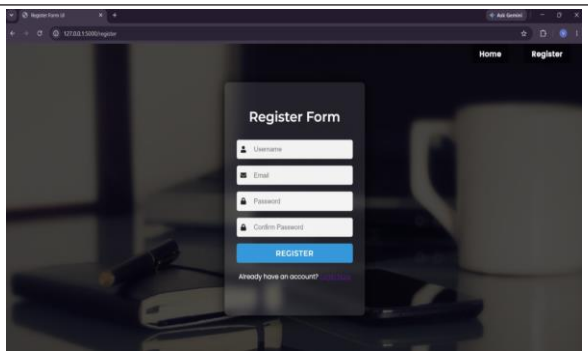


Fig. 2: Registration

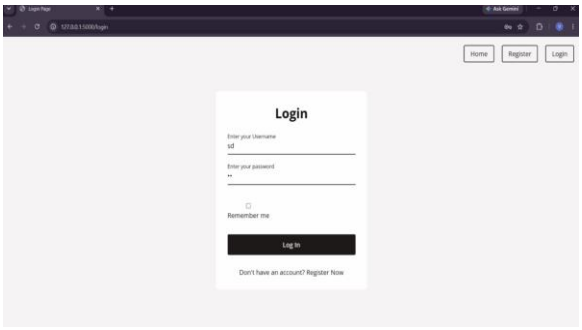


Fig. 3: Login

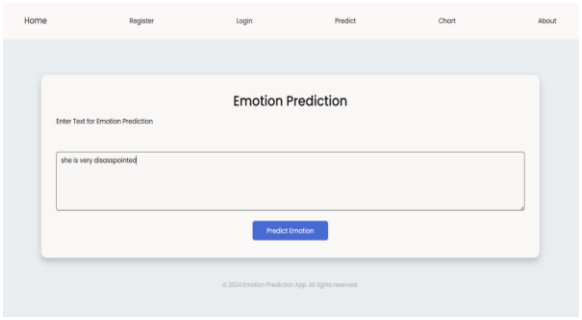


Fig. 4: Emotion Prediction

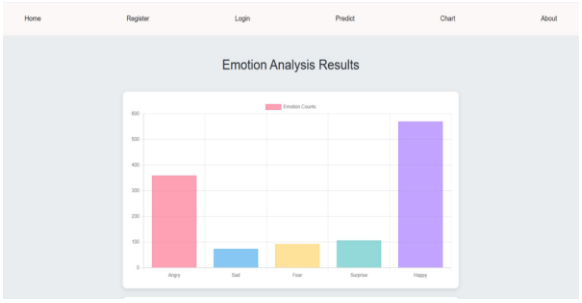


Fig. 5: Analysis

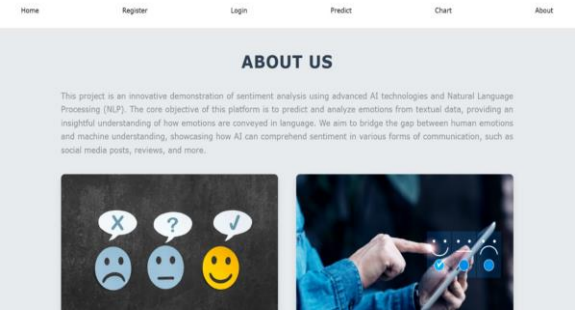


Fig. 6: About Us

V. CONCLUSION

In conclusion, the proposed approach combining BERT and LSTM offers a significant advancement in analyzing user feedback for software applications, not only identifying prevalent issues but also capturing the emotional context of those issues. This dual approach of emotional tone classification alongside bug identification allows developers to prioritize issues that most impact users. With a 92% accuracy, this model outperforms traditional machine learning algorithms, demonstrating the power of deep learning techniques in the realm of sentiment analysis. The ability to provide actionable insights for software vendors enables them to address critical issues swiftly, improve user satisfaction, and enhance their product's overall quality by integrating feedback in a more emotionally-aware manner.

REFERENCES

[1] Apple Store Downloads 2016 | Statista—Statista.com. Accessed:Mar. 6, 2024. [Online]. Available: https://www.statista.com/statistics/263794/number-of-downloads-from-the-apple-app-store/

[2] M. Pandey, R. Litoriya, and P. Pandey, ‘‘An ISM approach for modeling the issues and factors of mobile app development,’’ Int. J. Softw. Eng. Knowl. Eng., vol. 28, no. 7, pp. 937–953, Jul. 2018.

[3] P. Suresh and K. Gurumoorthy, ‘‘Mining of customer review feedback using sentiment analysis for smart phone product,’’ in Proc. Int. Conf. Comput., Commun., Electr. Biomed. Syst. Cham, Switzerland: Springer, 2022, pp. 247–259.

[4] A. Di Sorbo, G. Grano, C. Aaron Visaggio, and S. Panichella, ‘‘Investigating the criticality of user-reported issues through their relations with app rating,’’ J. Softw., Evol. Process, vol. 33, no. 3, p. e2316, Mar. 2021.

[5] Y. Liu, L. Liu, H. Liu, and X. Wang, ‘‘Analyzing reviews guided by app descriptions for the software development

and evolution,'' J. Softw., Evol. Process, vol. 30, no. 12, p. e2112, Dec. 2018.

[6] Gummadi, V. P. K. (2025). MuleSoft Architectural Paradigms and Sustainability: A Comprehensive Technical Analysis. Journal of Computer Science and Technology Studies, 7(12), 534-540. https://doi.org/10.32996/jcsts.2025.7.12.59.

[7] Bhagwat, V. B. (2025). Simplifying Payroll Balance Conversions in Payroll Systems Implementation through the Use of Generative AI.

[8] J. A. Khan, L. Liu, L. Wen, and R. Ali, ''Crowd intelligence in requirements \engineering: Current status and future directions,'' in Proc. Int. Working Conf. Requirement Eng., Found. Softw. Quality, 2019, pp. 245–261.

[9] Pavan Kumar Adabala. (2026). Best Practices for Enterprise System Integration in Modern Organizations. Journal of Information Systems Engineering and Management, 11(2s), 1137–1146. https://doi.org/10.52783/jisem.v11i2s.14558.

[10] Akinapalli, S. (2026). AN AI-POWERED DATA TRUST AND QUALITY SCORING FRAMEWORK FOR ENTERPRISE DECISION INTELLIGENCE SYSTEMS. International Journal of Data Science and IoT Management System, 5(1), 946-950.

[11] 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

AUTHORS Profile



Mrs. PIRANGI HYMAVATHI has received her B.Tech from JNTUH, Hyderabad in 2016, M.Tech degree in Computer science Engineering from JNTUH, Hyderabad in 2019. At present She is working as Assistant Professor in CSE Department at Guru Nanak Institutions Technical Campus, Ibrahimpatnam, Telangana, India.



Ms. GUNDA VARSHINI pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Ms. B. AKSHITHA REDDY pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. CH. SANTHOSH KOUSHIK pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.



Mr. G. VISHNU VARDHAN pursuing B. Tech degree in Computer Science and Engineering at Guru Nanak Institutions Technical Campus affiliated to JNTUH in 2022-2026.