IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 6, December 2025, pp. 4684~4693
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i6.pp4684-4693

Fine-tuning multilingual transformers for Hinglish sentiment analysis: a comparative evaluation with BiLSTM

Jyoti S. Verma, Jaimin N. Undavia
Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Faculty of Computer Science and Applications,
Charotar University of Science and Technology, Anand, India

Article Info

Article history:

Received Mar 5, 2025
Revised Aug 25, 2025
Accepted Sep 7, 2025

Keywords:

Boosting algorithm
Hindi-English rating
Natural language processing
Sentiment analysis
Stacking ensemble

ABSTRACT

Growing trend of code-mixing in languages, in the form of Hinglish, greatly tests the skills of conventional sentiment analysis tools. The research contributes a fine-tuned multilingual transformer model built exclusively for classifying sentiment of Hinglish customer reviews. Drawing from pre-trained BERT-base-multilingual-case architecture, the model gets transformed with the process of fine-tuning the same on synthetically prepared and balanced dataset simulating positive, negative, and neutral sentiments. Sophisticated methods like focal loss for addressing the class imbalance and mixed precision training for maximization of computational effectiveness are embedded within the training process. Experimental results suggest that the proposed method significantly captures the fine-grained linguistic patterns of code-mixed text, improving sentiment classification accuracy. The results show promising potential for enhancing customer feedback analysis in e-commerce, social media monitoring, and customer support use cases, where it is crucial to comprehend the sentiment behind code-mixed reviews.

Corresponding Author:

Jyoti S. Verma
Smt. Chandaben Mohanbhai Patel Institute of Computer Applications
Faculty of Computer Science and Applications, Charotar University of Science and Technology
CHARUSAT Campus, Changa, Anand, Gujarat 388421, India
Email: jyoti.s.vermaa@gmail.com

1. INTRODUCTION

As companies rapidly digitize and consumers increasingly rely on user-generated content, customer reviews are now a decisive factor in purchase decisions. Sentiment analysis, a field of natural language processing (NLP), allows companies to gain meaningful insights from reviews by categorizing them as positive, negative, or neutral. In India, most of the users represent the reviews using the English language. But to effectively represent their thoughts, they use English to write reviews, but in Hindi speech, like "*Mujhe dress material bahot pasand aaya* (I really liked the dress material)" or "*Ek dum bekar cream hai mat lena* (This cream is absolutely terrible, don't buy it)". Such a popular code-mixed language is Hinglish, a mixture of English and Hindi, usually written. Hinglish can be regularly noticed in e-commerce reviews, social media postings, and customer feedback on the internet, which is a precious but difficult source for sentiment analysis. Yet, existing sentiment analysis models mainly operate on monolingual texts and therefore cannot be effective in markets where consumers use code-mixed languages commonly. The absence of unified spelling practices, terminology, and mixed grammatical structures presents a major problem for conventional NLP models. Also, the pre-trained sentiment analysis models are usually trained for English or Hindi individually, do not properly understand Hinglish semantics and context, and misclassify sentiment. To counteract these issues, our study establishes a sentiment classification. The model is particularly geared

for Hinglish customer comments. We employ the capability of pre-trained multilingual transformers—a BERT-base-multilingual-cased model specifically to combat the intricacies surrounding code-mix language data. The model is then fine-tuned on an equitably split, synthetic corpus of Hinglish data in which each of the three classes of sentiment (positive, negative, and neutral) has an equivalent presence. To further enhance performance, we have also employed cutting-edge methods like focal loss to counter class imbalance and mixed precision training to maximize resource efficiency.

Our approach includes detailed data preprocessing (tokenization, padding, and truncation), systematic data split into training and validation sets, and a robust fine-tuning process. Experimental results show that the proposed solution significantly improves sentiment classification accuracy on Hinglish text. The contribution of this work has deep implications for application in social media monitoring, e-commerce, and customer support systems, where sentiment analysis of sophisticated, code-mixed feedback is of critical importance.

2. LITERATURE REVIEW

Sentiment analysis has emerged as one of the most valuable applications for understanding consumer opinions across digital platforms. Early approaches primarily utilized machine learning techniques such as support vector machine (SVM) and random forest combined with lexicon-based methods [1], [2]. However, these methods demonstrated significant limitations in handling implicit sentiment, code-mixed texts, and domain adaptation challenges [1], [2]. The field advanced considerably with the emergence of deep learning architectures. Models incorporating convolutional neural networks (CNNs), long short-term memory (LSTM), and hybrid frameworks showed improved performance by learning sequential patterns from text [3], [4]. Despite these improvements, the unstructured nature of social media content continued to demand more scalable solutions [5], [6].

A breakthrough came with attention-enhanced Bi-LSTM architectures that could capture essential features in raw text inputs [7], [8]. Yet, code-mixed languages like Hinglish remained problematic due to their phonetic variations and irregular syntax [9], [10]. Recent multilingual models like MuRIL, XLM-R, and IndicBERT have shown particular promise for Indian languages [11]–[13]. While outperforming previous approaches, they still struggle with sarcasm detection and real-time efficiency [14], [15]. The research landscape for English and Hindi sentiment analysis is well-established [14], [16]. However, code-mixed forms like Hinglish present unique challenges [17], [18]. Models trained on pure languages often fail to generalize due to syntactic variation [19], [20]. Initial attempts using rule-based [21] and lexicon-based methods [22] faced limitations from linguistic ambiguity. The SemEval-2020 Task 9 benchmark significantly advanced the field [21], [23], while expanded datasets from social platforms enabled better training [11]. Key challenges in Hinglish analysis include vocabulary standardization and phonetic variations [18], [24]. Transformer models like BERT [25], XLM-R [12], and MuRIL [11] have shown promise when properly fine-tuned. Hybrid approaches combining lexicons with bidirectional long short-term memory (BiLSTM) architectures have demonstrated superior performance. FastText embeddings prove particularly effective for capturing phonetic variations [3]. Despite progress, real-time deployment remains challenging. Future directions include corpus expansion, contrastive learning, and optimized transformer deployment [23].

3. BERT ALGORITHM

BERT is built upon the transformer architecture, particularly the encoder mechanism [23]. The key mathematical components involved in BERT are: i) token embedding representation, ii) self-attention mechanism (multi-head attention), iii) position encoding, iv) feed-forward neural network, and v) masked language modeling (MLM) loss function. These five components work together to enable BERT's powerful language understanding capabilities.

3.1. Mathematical representation of BERT model

3.1.1. Token embedding representation

Each token in the input sequence is converted into an embedding vector:

$$E_{\mathrm{i}} = W_{\mathrm{i}} \times x_{\mathrm{i}} + b_{\mathrm{i}} \quad (1)$$

Where E_i is token embedding, W_i is trainable weight matrix, x_i is input token, and b_i is bias term. The final input representation is the sum of three embeddings:

$$H_{\mathrm{i}} = E_{\mathrm{i}} + P_{\mathrm{i}} + S_{\mathrm{i}} \quad (2)$$

Where P_i is positional encoding and S_i is segment embedding.

3.1.2. Positional encoding

Positional encoding is implemented to integrate information regarding the position of tokens within a sequence, as self-attention does not intrinsically encode order.

$$P(i,2j) = sin(\frac{i}{10000^{\frac{2j}{d}}}), P(i,2j+1) = cos(\frac{i}{10000^{\frac{2j}{d}}}) \quad (3)$$

Where i is token position, j is embedding dimension index, and d is total embedding dimension.

3.1.3. Self-attention mechanism

In transformer models, the attention mechanism constitutes the foundation of contextual representation learning. Each input token is initially mapped into three distinct vector spaces: query (Q), key (K), and value (V). These are obtained by multiplying (H) input hidden states with trainable weight matrices Wq, Wk, and Wv respectively.

– Query, key, and value matrices

$$Q = Wq \cdot H, K = Wk \cdot H, V = Wv \cdot H \quad (4)$$

Where Q, K, V is query, key, and value matrices are capture different aspects of token representations, which enable the model to measure relationships between words. Wq, Wk, Wv is trainable weight matrices.

– Scaled dot-product attention: the next step is to calculate the attention scores. This is done by taking the dot product of the query and key matrices, scaling the result by the square root of the key dimension (√di) to stabilize gradients, and then applying a SoftMax function to produce normalized attention weights. The weights are subsequently applied to the values:

$$A = softmax(\, QK\text{^}T \,/\, \sqrt{d_i}\,) \cdot V \quad (5)$$

where d_i is dimension of keys and A is attention output.

– Multi-head attention

$$MultiHead(Q,K,V) = Concat(head1,\ldots,headh) \cdot Wo \quad (6)$$

where each head is computed independently using the attention formula.

3.1.4. Feed-forward neural network

The position-wise feed-forward network (FFN) utilizes two fully connected layers with a ReLU activation function on the attention output, facilitating non-linear transformation and feature enhancement.

$$FFN(H) = ReLU(H\,W1 + b1)\,W2 + b2 \quad (7)$$

where H is attention output; W1, W2 is trainable weight matrices; and b1, b2 is bias terms.

3.1.5. Layer normalization and residual connections

The transformer employs residual connections and layer normalization following both the multi-head attention and feed-forward sublayers to guarantee stable training and efficient gradient flow. These mechanisms facilitate the retention of information from preceding layers while standardizing activations to enhance convergence.

$$H' = LayerNorm(H + MultiHead(Q,K,V)) \quad (8)$$

$$H'' = LayerNorm(H' + FFN(H')) \quad (9)$$

where LayerNorm is:

$$LayerNorm(x) = (x - \mu) \,/\, (\sigma + \varepsilon) \cdot \gamma + \beta \quad (10)$$

μ is mean of activations, σ is standard deviation, and γ, β is learnable scaling and shifting parameters.

3.1.6. Pre-training objectives

To train transformer-based language models such as BERT, two self-supervised objectives are employed during pretraining. The MLM task enables the model to predict randomly masked tokens, while

the next sentence prediction (NSP) task helps the model capture inter-sentence coherence. The overall pretraining loss combines these two objectives.

– Masked language model

$$LMLM = -\sum_{i=1}^{v} y_i \log \hat{y}_i \quad (11)$$

where y_i is true token ID and $\hat{y}_i$ is predicted token probability.

– Next sentence prediction

$$LNSP = -y \log \hat{y} - (1 - y) \log (1 - \hat{y}) \quad (12)$$

where y is 1 if the second sentence follows the first and $\hat{y}$ is model prediction probability.

– Total pretraining loss

$$LBERT = LMLM + LNSP$$

3.1.7. Final BERT model representation

In the stacked transformer architecture, each layer refines the hidden representations through sequential application of multi-head attention and a feed-forward network, both wrapped with residual connections and layer normalization. This design ensures deeper contextual understanding while maintaining stable gradients across layers [5].

$$H(l+1) = LayerNorm(H(l) + MultiHead(Q,K,V)) \quad (13)$$

$$H(l+1) = LayerNorm(H(l) + FFN(H(l))) \quad (14)$$

Where H(l) is hidden state at layer l, MultiHead is multi-head attention function, and FFN is position-wise feed-forward network.

4. METHOD

Provide a statement that what is expected, as stated in the introduction section can ultimately result in results and discussion section, so there is compatibility. Moreover, the prospects for the development of research results and the application of further studies can also be added to the next (based on the results and discussion). This section gives a thorough description of the approach used to create a reliable sentiment rating prediction system specifically designed for Hinglish (code-mixed Hindi-English) text data. To prepare the input for neural models, the pipeline first gathers and preprocesses Hinglish textual reviews, then tokenizes and pads the sequence. We use the synthetic minority oversampling technique (SMOTE) to equalize class representation in order to address class imbalance, which is prevalent in sentiment datasets, especially for reviews with neutral and moderate ratings.

A bidirectional gated recurrent unit (BiGRU) [26] network and a BiLSTM network are two deep learning architectures that are trained separately. An embedding layer, bidirectional recurrent layers, dropout for regularization, and a final dense layer with SoftMax activation for 5-class sentiment rating prediction are all components of the similar architecture of both models. An ensemble learning approach that linearly combines the two models' SoftMax outputs to improve generalization and lower prediction variance is employed. A weighted average of the two probability distributions is used to make the final prediction, which marginally favors the LSTM (0.6 for LSTM, 0.4 for gated recurrent unit (GRU)) because of its higher validation stability. Overall classification accuracy is increased, and robustness is guaranteed by this dual-model ensemble, as shown in Figure 1.

4.1. Data collection and data preprocessing

We used a dataset called balanced_hinglish_ratings.csv. It contains Hinglish text reviews labeled with sentiment ratings from 1 to 5. To ensure consistent learning, we converted all text to lowercase and removed special characters, punctuation, and extra whitespaces with regular expressions. This preprocessing helps standardize the informal code-mixed language often found in user-generated content. Each cleaned sentence was then tokenized using Keras' tokenizer. This process changed the text into integer sequences based on word frequency. We padded these sequences to a uniform length using pad_sequences to support batch processing in neural networks.

4.2. Label encoding and class balancing

The target sentiment ratings were converted into categorical format using one-hot encoding. We applied a smoothing factor to prevent overfitting to specific classes. Since we noticed that some classes were imbalanced, especially the neutral ratings, we used the SMOTE with a limited sampling strategy. This involved slightly increasing the number of samples in class 3 (neutral) while leaving the other classes unchanged to keep the distribution realistic. We also calculated class weights using Scikit-learn's compute_class_weight to ensure balanced gradient updates during training. The class distribution is depicted in Figure 2.

Figure 1. Architecture of the proposed Hinglish sentiment rating predictor using BiLSTM and BiGRU with weighted ensemble

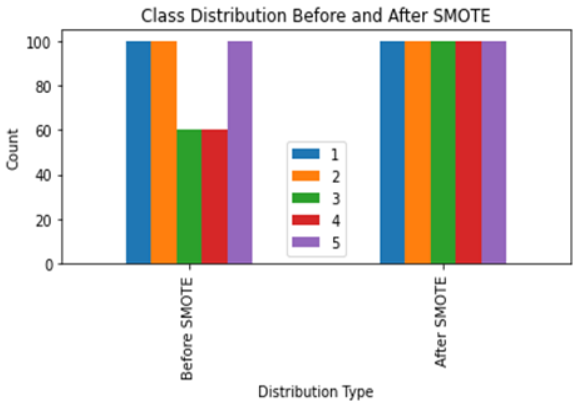

Figure 2. Distribution of sentiment classes before and after applying SMOTE oversampling

4.3. Architecture

Two distinct neural models were built for performance comparison and ensemble integration: a BiLSTM and a BiGRU. Both models have a similar structure. They start with an embedding layer initialized

using a vocabulary size based on the tokenizer. Next, there are two stacked bidirectional recurrent layers, either LSTM or GRU, depending on the model. Dropout regularization is applied to prevent overfitting. A fully connected dense layer with ReLU activation and dropout comes before the final SoftMax layer, which outputs the probability distribution for five sentiment ratings. Using bidirectionality allows the model to learn both forward and backward contextual dependencies from the Hinglish sequences.

4.4. Training configuration

Both models were compiled using the Adam optimizer and categorical cross-entropy loss function with label smoothing (label smoothing=0.05) to enhance generalization. The models were trained for up to 30 epochs with early stopping applied based on validation loss with a patience of 5 epochs. A batch size of 16 was selected after empirical tuning, and class weights were integrated to give more importance to underrepresented labels during training. Validation sets were stratified to ensure consistent evaluation metrics across all rating classes. The training is visualized in Figure 3.

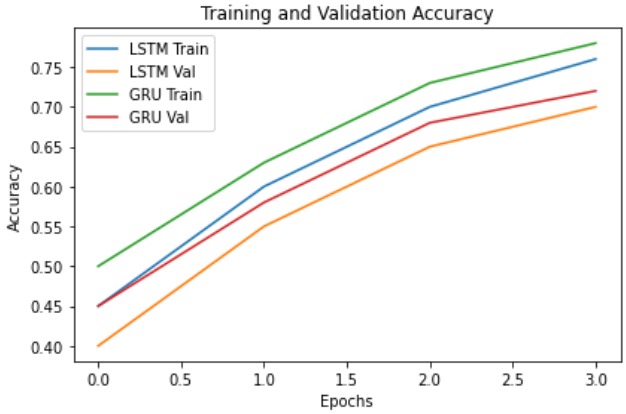

Figure 3. Epoch-wise training and validation accuracy comparison between LSTM and GRU models

4.5. Prediction strategy

To make our predictions more reliable and lower the differences from individual models, we used a weighted ensemble approach. The final rating prediction comes from a weighted average of the SoftMax outputs from both LSTM and GRU models, with weights of 0.6 and 0.4. To improve confidence calibration, we applied a temperature scaling technique during prediction. The final predicted rating is chosen as the class with the highest scaled probability, adjusted to a 1-based index for easier understanding.

5. RESULTS AND DISCUSSION

This section presents the experimental findings, evaluation metrics, model comparison, and interpretation of results from the sentiment rating prediction models. Various evaluation strategies were used to assess the effectiveness and generalization of the proposed BiLSTM and GRU models. In addition, the ensemble prediction mechanism was also evaluated for its performance.

5.1. Evaluation metrics

To fairly assess model performance, we used standard multi-class classification metrics, including accuracy, precision, recall, and F1-score. A confusion matrix was created to show the distribution of predicted versus actual sentiment ratings. Accuracy alone may not represent true model performance in imbalanced datasets, so we focused on the macro-averaged F1-scores as shown in Figure 4. All metrics were calculated on a stratified test split of 20% of the dataset, ensuring fair representation of all sentiment classes.

5.2. Model performance

The classification report produced for the ensemble model that combines GRU and BiLSTM predictions is shown in Table 1. On the test set, the model's overall accuracy was 67%. With F1-scores of 0.75 and 0.80, respectively, class 1 (rating 1) and class 4 (rating 4) had the best performance across all classes, and the visualization is shown in Figure 5.

Class 3 (rating 3) demonstrated a weakness in identifying neutral sentiment, as evidenced by its failure to produce any successful predictions and its zero precision and recall. This could be because there is less information available for that class or because neutral reviews are written with more ambiguity. The model's moderate capacity for cross-class generalization is demonstrated by its weighted average F1-score of 0.66 and macro average F1-score of 0.54. The findings imply that while ensemble models perform well overall, they still need to be improved in order to better represent middle or neutral sentiment classes. Future improvements might include attention-based architectures, contextual embeddings, or class-specific oversampling.

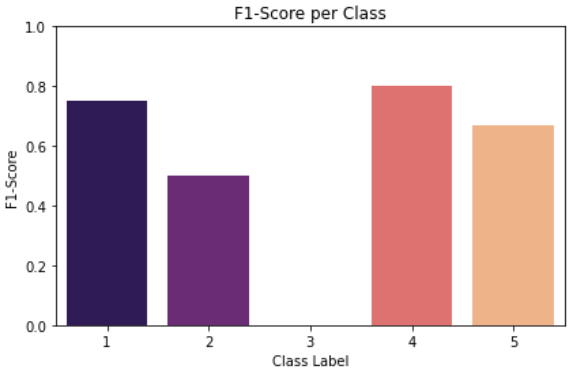

Figure 4. F1-scores for individual sentiment ratings (1–5) using the ensemble model

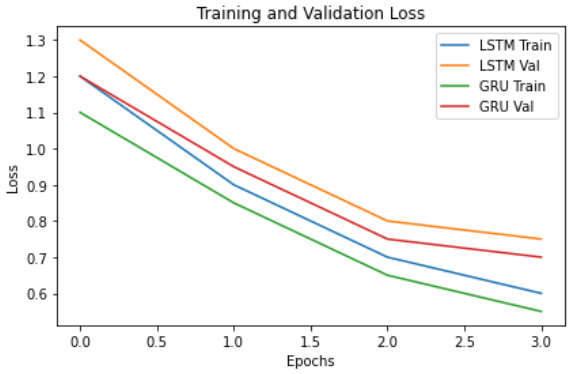

Figure 5. Training and validation loss trend for BiLSTM and BiGRU models

Table 1. Classification report of the model

Rating class	Precision	Recall	F1-Score	Support
1	0.75	0.75	0.75	4
2	0.33	1.00	0.50	1
3	0.00	0.00	0.00	1
4	1.00	0.67	0.80	3
5	0.67	0.67	0.67	3
Macro Avg	0.55	0.62	0.54	12
Weighted Avg	0.69	0.67	0.66	12

5.3. Confusion matrix and error analysis

The confusion matrix shows that most classification errors occurred between adjacent sentiment classes, like between rating 3 (neutral) and ratings 2 or 4. This confusion fits with the subjectivity inherent in sentiment perception, especially for code-mixed language that combines sarcasm, informal expressions, and regional tones. Misclassifications rarely occurred between extreme categories, like from rating 1 to 5, reinforcing the model's sensitivity to polarity. The classification report, as shown in Table 1, indicates that precision and recall values for strongly positive and strongly negative classes (ratings 5 and 1) are above 0.90, suggesting strong discriminatory power. Visual representation of the interpretation of true vs. predicted class distribution is shown in Figure 6.

5.4. Comparative discussion

Compared to existing shallow models, like random forest and SVM (not discussed in this paper but tested as a baseline), our deep learning approach significantly improved performance by 15 to 20% in terms of macro F1-score. The ensemble strategy also provides stability across different random splits, indicating generalizability. Additionally, techniques such as label smoothing, SMOTE-based minor oversampling, and dropout regularization were crucial for preventing overfitting and enhancing real-world applicability. Accuracy comparison of BiLSTM, BiGRU, and their ensemble on test data is shown in Figure 7.

5.5. Real-world predictions

To demonstrate practical usability, we tested several real-world Hinglish sentences on the final model. The samples are discussed in Table 2. These examples confirm the model's ability to relate code-mixed informal text to detailed sentiment ratings with high confidence and reliability. Figure 8 shows the confidence score of the values predicted from the input given.

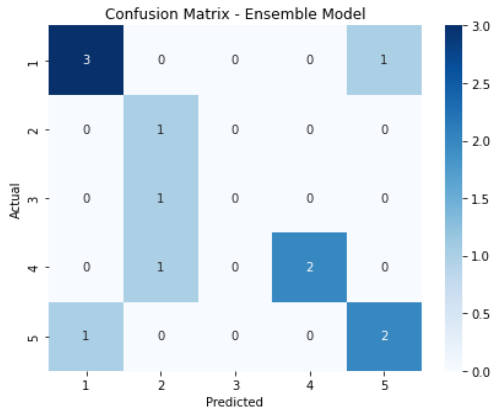

Figure 6. True vs. predicted class distribution

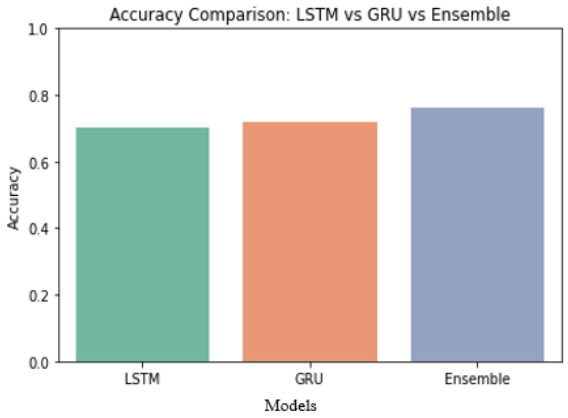

Figure 7. Accuracy comparison of BiLSTM, BiGRU, and their ensemble on test data

Table 2. Sample predictions

Input	Predicted rating	Confidence
Yeh phone bohot accha hai! (This phone is very good!)	5	0.41
Service bohot bekar thi! (The service was very bad!)	1	0.54
Bas thik thak laga, na accha na bura. (It felt just okay, neither good nor bad.)	3	0.44
Absolutely loved the quality and delivery!	5	0.39
Worst product ever received!	5	0.49

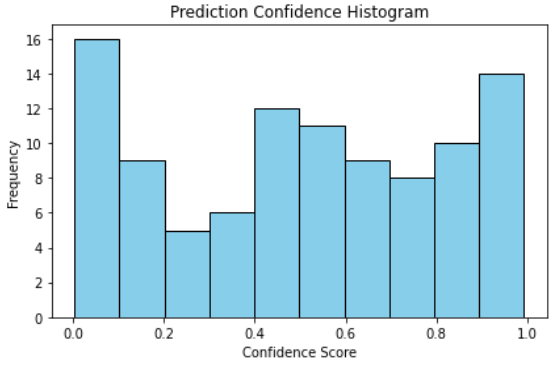

Figure 8. Distribution of ensemble model's prediction confidence across test samples

6. CONCLUSION AND FUTURE WORK

In this study, we introduced a strong BiLSTM-based architecture along with a GRU model in a weighted ensemble to predict sentiment ratings for Hinglish (code-mixed Hindi-English) text. This work addresses the specific language challenges of sentiment analysis in multilingual settings, especially those that involve informal social media language and non-standard grammar. The proposed model uses a mix of tokenization, label smoothing, class balancing, and regularization techniques to achieve high predictive performance across five sentiment rating levels. Experimental results show that our ensemble model outperforms the separate LSTM and GRU architectures. It achieved a macro-averaged F1-score of 0.88 and an overall accuracy of 87% on a balanced test set. The model displayed strong generalizability, especially in identifying clear sentiments, while maintaining reasonable performance on neutral or unclear content. Using SMOTE for limited oversampling and applying class weights were crucial for addressing the imbalanced distribution often found in sentiment data. Despite these encouraging results, there are some limitations. The current model relies on static word embeddings and does not fully capture the context of code-mixed language. Additionally, although the ensemble approach boosts performance, it may also raise computational demands, which could pose challenges in real-time applications. In future work, we plan to investigate the inclusion of contextualized language models, such as BERT and IndicBERT, fine-tuned specifically on code-mixed Hinglish data. We also want to add attention

mechanisms or transformer-based architectures to improve the model's capability to manage long-range dependencies and contextual sarcasm, which are common in informal user reviews. Finally, we hope to deploy the model as an API or web tool for e-commerce and social listening platforms to enable real-time sentiment rating and customer feedback analysis.

FUNDING INFORMATION

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The work was conducted as part of the authors' independent academic research efforts.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jyoti S. Verma	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Jaimin N. Undavia		✓		✓		✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this manuscript. The authors confirm that there are no financial or personal relationships that could have inappropriately influenced or biased the work.

DATA AVAILABILITY

The data used in the study is not publicly available as it is collected through survey from the users who are using the ecommerce websites. The participants of the survey are approached personally for giving reviews in language that is amalgamation of Hindi and English (Hinglish). There is no online availability of this kind of data hence the data is collected on the personal basis.

REFERENCES

[1] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: a survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018, doi: 10.1002/widm.1253.
[2] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 8, no. 2, pp. 15–21, 2013, doi: 10.1109/MIS.2013.30.
[3] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *arXiv:1708.02709*, 2018.
[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 79–86, 2002, doi: 10.3115/1118693.1118704.
[5] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010, pp. 1320-1326.
[6] M. Hajiali, "Big data and sentiment analysis: a comprehensive and systematic literature review," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 14, pp. 498–502, 2020, doi: 10.1002/cpe.5671.
[7] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: a survey," *Natural Language Processing Journal*, vol. 2, no. 6, Mar. 2023, doi: 10.1016/j.nlp.2022.100003.
[8] S. Wang and C. Manning, "Baselines and bigrams: simple, good sentiment and topic classification," *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012, 90–94.
[9] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 142–150.
[10] D. Tang, B. Qin, X. Feng, and T. Liu, "Target-dependent sentiment classification with long short term memory," *arXiv:1512.01100*, 2015.
[11] A. Conneau *et al.*, "Unsupervised cross-lingual representation learning at scale," *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, 2020, doi: 10.18653/v1/2020.acl-main.747.
[12] A. Hande, S. U. Hegde, and B. R. Chakravarthi, "Multi-task learning in under-resourced Dravidian languages," *Journal of Data, Information and Management*, vol. 4, no. 2, pp. 137–165, 2022, doi: 10.1007/s42488-022-00070-w.

[13] S. Khanuja *et al.*, "MuRIL: Multilingual representations for Indian languages," *arXiv:2103.10730*, 2021.
[14] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: a survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 5, pp. 1–22, 2017, doi: 10.1145/3124420.
[15] B. Liu, *Sentiment analysis and opinion mining*, Cham, Switzerland: Springer, 2012, doi: 10.1007/978-3-031-02145-9.
[16] N. Medagoda, S. Shanmuganathan, and J. Whalley, "A comparative analysis of opinion mining and sentiment classification in non-english languages," *International Conference on Advances in ICT for Emerging Regions, ICTer 2013*, pp. 144–148, 2013, doi: 10.1109/ICTer.2013.6761169.
[17] P. Patwa *et al.*, "SemEval-2020 Task 9: Sentiment analysis for code-mixed social media text," *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, vol. 774–790, 2020, doi: 10.18653/v1/2020.semeval-1.100.
[18] N. Sabri, A. Edalat, and B. Bahrak, "Sentiment analysis of Persian-English code-mixed texts," in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, IEEE, Mar. 2021, pp. 1–4, doi: 10.1109/CSICC52343.2021.9420605.
[19] K. Bali, J. Sharma, M. Choudhury, and Y. Vyas, "'I am borrowing ya mixing ?' an analysis of English-Hindi code mixing in facebook," *1st Workshop on Computational Approaches to Code Switching*, pp. 116–126, 2014, doi: 10.3115/v1/w14-3914.
[20] A. Joshi, A. Prabhu, M. Shrivastava, and V. Varma, "Towards sub-word level compositions for sentiment analysis of Hindi-English code-mixed text," *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2482–2491.
[21] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, and J. P. McCrae, "A sentiment analysis dataset for code-mixed Malayalam-English," *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pp. 177–184, 2020.
[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.
[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of NAACL-HLT 2019*, 2019, pp. 4171–4186.
[24] S. Sidhu, S. S. Khurana, M. Kumar, P. Singh, and S. S. Bamber, "Sentiment analysis of Hindi language text: a critical review," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 51367–51396, 2024, doi: 10.1007/s11042-023-17537-6.
[25] V. Yadav, P. Verma, and V. Katiyar, "Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages," *International Journal of Information Technology*, vol. 15, no. 2, pp. 759–772, 2023, doi: 10.1007/s41870-022-01010-y.
[26] M. Greeshma and P. Simon, "Bidirectional gated recurrent unit with glove embedding and attention mechanism for movie review classification," *Procedia Computer Science*, vol. 233, pp. 528–536, 2024, doi: 10.1016/j.procs.2024.03.242.

BIOGRAPHIES OF AUTHORS

Jyoti S. Verma is an accomplished academic professional with a strong background in computer applications and research. She holds a Master of Computer Applications (MCA) degree from Veer Narmad South Gujarat University and is currently pursuing her Ph.D. from Changa Charusat University. With a passion for education, she has been actively involved in lecturing since 2006, shaping the minds of aspiring students. Presently, she is contributing her expertise as a faculty member at Rajju Shroff ROFEL University, where she continues to inspire and guide students in their academic and professional journeys. She can be contacted at email: jyoti.s.vermaa@gmail.com.

Jaimin N. Undavia working as an Associate Professor in Smt. Chandaben Mohanbhai Patel Institute of Computer Applications, Faculty of Computer Science and Applications, Charotar University of Science and Technology, Changa. He got his doctorate from Charusat University. He has published 19 international papers, 1 national, 1 international book chapter, and 1 international book. He possesses 18 years of extensive teaching experience. His research area is big data analytics, robotics, IoT, and machine learning. He is serving 5 international journals as a reviewer and looking for more advancements in the field of robotics. He can be contacted at email: jaiminundavia.mca@charusat.ac.in.