A HYBRID DEEP LEARNING FRAMEWORK INTEGRATING LSTM AND LIGHTGBM FOR SENTIMENT ANALYSIS OF ROMAN URDU TEXT

Kanwal Mehmood[*1], Muhammad Ahsan Naeem[2], Muhammad Imran[3]

[*1,2]*Department of Mechatronics and Control Engineering, University of Engineering and Technology (UET), Lahore, Pakistan*
[3]*University of the Punjab, Lahore, Pakistan*

[*1]kanwal.mehmood123@gmail.com, [2]mahsan.naeem@uet.edu.pk, [3]mimranag4u@gmail.com

[1]0009-0002-9890-0412, [2]0000-0003-1754-4357

DOI:https://doi.org/10.5281/zenodo.20700421

Keywords
Sentiment analysis; Roman Urdu; low-resource languages; hybrid deep learning; LSTM; LightGBM; natural language processing; text classification

Article History
Received: 16 April 2026
Accepted: 27 May 2026
Published: 15 June 2026

Corresponding Author: *
Kanwal Mehmood

Abstract
Sentiment analysis is central to extracting opinions and emotional context from user-generated text, yet its application to Roman Urdu remains constrained by the language's informal usage, non-standardised orthography, and scarcity of annotated resources. This study proposes a hybrid classification framework that couples a Long Short-Term Memory (LSTM) network with a Light Gradient Boosting Machine (LightGBM) classifier to improve sentiment prediction for Roman Urdu. The LSTM branch models sequential and contextual dependencies in the text, while the LightGBM branch captures non-linear interactions among engineered features; the two branches are combined through a weighted Softmax fusion layer. A publicly available Roman Urdu corpus of 98,984 samples obtained from Kaggle was preprocessed using a custom tokenizer, transliteration-aware normalisation, and language-specific stop-word removal. The framework was trained and evaluated using stratified ten-fold cross-validation. The hybrid model achieved a classification accuracy of 97.74%, exceeding the standalone LSTM (93.72%) and standalone LightGBM (69.51%) models, and also outperforming conventional classifiers including Random Forest, Support Vector Machine, and k-Nearest Neighbour. The results indicate that integrating sequential representation learning with gradient-boosted feature modelling is an effective strategy for sentiment analysis in low-resource, non-standardised languages, and provide a basis for future work on multilingual and code-mixed sentiment systems.

1. INTRODUCTION

The rapid expansion of online communication platforms has produced large volumes of user-generated text, making automated sentiment analysis an important tool for interpreting public opinion. Sentiment analysis is now applied across diverse domains, including social media monitoring, education, business intelligence, entertainment, and sports analytics, where user comments inform strategic decision-making [1]. Reported applications extend to the prediction of stock-market movements from social-media sentiment [2], the detection of fraudulent activity in financial data [3], and the assessment of patient feedback and public opinion on policy in healthcare and political science [2]. Contemporary sentiment-analysis methods based on deep learning are able to detect slang, sarcasm, and subtle contextual cues that keyword-based approaches typically miss [4]. Architectures such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs) capture long-range dependencies in text and have advanced performance on complex language-processing tasks [5, 6]. Despite this progress, sentiment analysis for low-resource languages with rich morphological structure remains difficult, largely because of limited annotated data and a lack of standardised linguistic tools [7]. These constraints are especially pronounced for Roman Urdu, an informal, Latin-script representation of Urdu that lacks codified spelling conventions [8, 9].

Roman Urdu is widely used in social-media and online communication across Pakistan and India, yet its non-standardised orthography, frequent code-mixing with English, and absence of formal linguistic resources limit the effectiveness of conventional sentiment-analysis models. Recurrent and transformer-based architectures have improved sentiment classification for morphologically complex languages by modelling long-range dependencies and contextual variation [10-13], and recent work indicates that hybrid models combining deep learning with gradient-boosting methods can further improve classification performance [14, 15]. Motivated by these observations, the present study proposes a hybrid framework that integrates an LSTM network with a LightGBM classifier for sentiment analysis of Roman Urdu text. The LSTM branch learns sequential and contextual representations, while the LightGBM branch models non-linear interactions among engineered features; combining the two is intended to yield more accurate sentiment predictions than either component alone. The main contributions of this work are: (i) a custom preprocessing pipeline, comprising transliteration-aware normalisation and a Roman Urdu-specific tokenizer, designed for the linguistic characteristics of the language; (ii) a hybrid LSTM-LightGBM architecture with a weighted Softmax fusion layer; and (iii) a comprehensive evaluation against standalone deep-learning and conventional machine-learning baselines using stratified cross-validation. The conceptual framework of the proposed approach is shown in Figure 1.

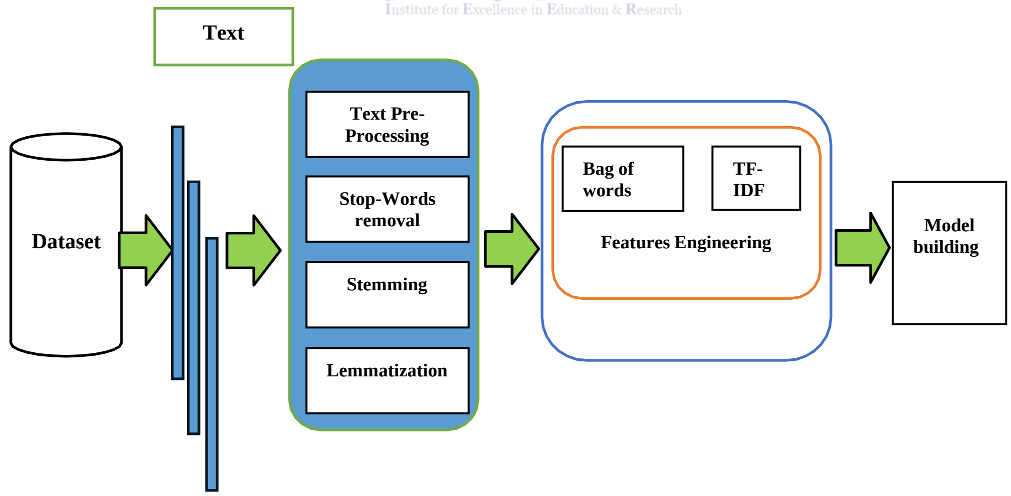

Figure 1:Proposed Solution

Research on sentiment analysis has evolved through three broad stages: lexicon-based methods, conventional machine learning, and deep learning. Early systems relied on fixed sentiment dictionaries and hand-crafted linguistic rules [16]. Although effective for simple cases, these approaches struggled with sarcasm, negation, and context-dependent sentiment, which motivated the adoption of machine-learning techniques. Classifiers such as Support Vector

Machines (SVM), Naive Bayes, and Random Forest improved performance by learning patterns from annotated corpora, although they depended heavily on manual feature engineering [5, 10].

Deep-learning models subsequently advanced the field by learning hierarchical representations and temporal dependencies directly from text. CNNs are effective at extracting local features, while recurrent networks, particularly LSTM networks, capture long-range contextual information; combined CNN-BiLSTM models have been shown to improve sentiment classification by exploiting both properties [10]. Transformer-based models such as BERT have further improved performance, and fine-tuning pretrained transformers has been reported to outperform conventional embedding methods even when annotated data are limited [5]. For morphologically rich languages such as Hindi, deep-learning models have consistently outperformed Naive Bayes, SVM, and Logistic Regression [10].

Hybrid architectures that combine complementary models have attracted increasing attention. Integrating the representation-learning capability of deep networks with the explicit feature modelling of tree-based or boosting methods can improve both predictive accuracy and interpretability [17, 18]. Reported examples include CNN-BiLSTM models with Doc2Vec embeddings achieving approximately 90.7% accuracy, and BiLSTM models reaching about 94% accuracy on Saudi-dialect text [1]. For Arabic sentiment analysis, LSTM-based models combined with word embeddings have repeatedly outperformed conventional methods by capturing semantic dependencies [18, 19]. Attention-augmented architectures, such as Cov-Att-BiLSTM, and transformer-based hybrids combining BERT, BiLSTM, and TextCNN have also improved classification performance, although often at higher computational cost [20, 21]. Collectively, these studies indicate that hybrid designs can balance accuracy and efficiency, which is particularly valuable for low-resource languages [22-25].

Although sentiment analysis has advanced considerably for high-resource languages, three gaps remain for Roman Urdu. First, most existing studies target languages with standardised orthography and large annotated corpora, whereas Roman Urdu lacks both, limiting the direct applicability of conventional models. Second, standalone deep-learning models capture sequential structure but do not explicitly model non-linear interactions among engineered features, while tree-based and boosting models capture such interactions but ignore word order; few studies have systematically combined the two for Roman Urdu. Third, prior Roman Urdu sentiment work has often relied on relatively small datasets and limited cross-validation, leaving the generalisation of reported results uncertain. The present study addresses these gaps by proposing a hybrid LSTM-LightGBM framework, evaluated on a large corpus of 98,984 samples using stratified ten-fold cross-validation.

3. Materials and Methods

Our methodical and iterative approach to creating a hybrid sentiment analysis model for Roman Urdu text combines the advantages of deep learning and machine learning approaches. To ensure the quality and relevance of the data, we first gathered a comprehensive labeled dataset of Roman Urdu text samples and implemented thorough preprocessing procedures. Then, using both conventional and contextual embedding techniques, we executed feature engineering steps to extract informative representations from the preprocessed text. The LIGHTGBM and LSTM components were integrated into the hybrid model architecture, each with a specific function in capturing high-level patterns and contextual relationships. The methodology scheme is shown in Figure 2.

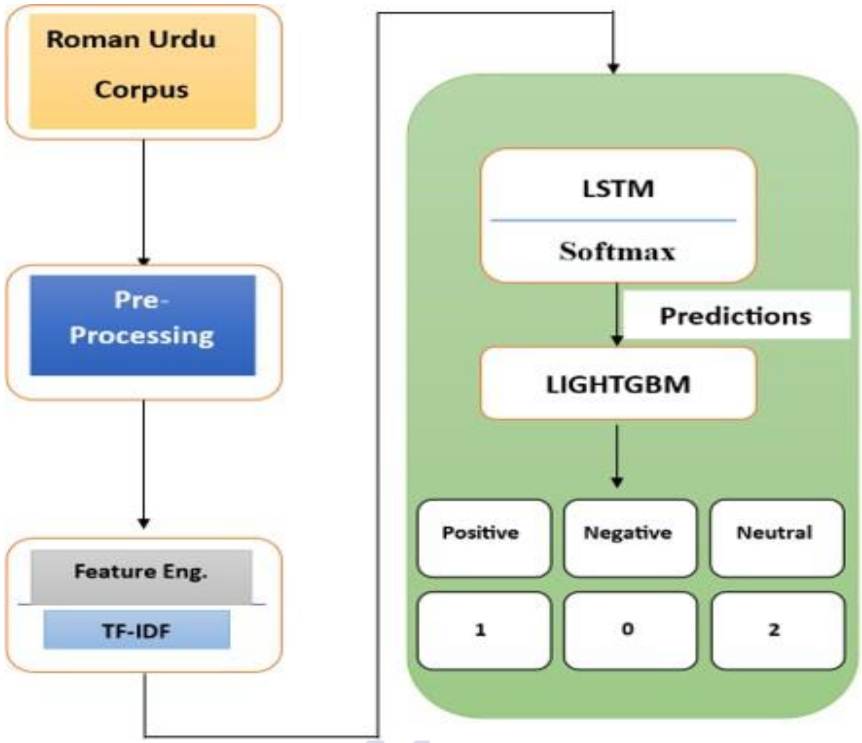

Figure 2:Methodology

All experiments were conducted in a controlled computing environment managed with the Anaconda distribution. A dedicated Python 3.8 environment was created with the required libraries isolated from system-level packages, ensuring reproducibility and avoiding dependency conflicts across experimental runs.

3.1. Data Collection and Preprocessing

The dataset used in this study was obtained from publicly available Roman Urdu repositories on Kaggle, comprising user-generated content from social-media platforms, community forums, and online discussion boards. This selection ensured that the corpus reflected the informal usage typical of Roman Urdu in everyday communication. The final corpus contained 98,984 samples annotated into three sentiment classes: 27,487 positive samples (27.8%), 23,555 negative samples (23.8%), and 47,942 neutral samples (48.4%). The raw text was preprocessed through a sequence of stages to improve quality and consistency. Text normalisation was applied to correct spelling variation arising from non-standardised transliteration; a custom tokenizer was used to segment text into words and subwords, addressing the absence of consistent word boundaries in Roman Urdu; and language-specific stop words, identified through exploratory analysis, were removed together with other uninformative tokens. This pipeline produced a clean and consistent dataset suitable for feature extraction and model training.

3.2. Feature Engineering and Model Architecture

Feature engineering was performed to convert preprocessed text into informative numerical representations. Term Frequency-Inverse Document Frequency (TF-IDF) vectors were used to quantify the relative importance of terms, and distributed word embeddings (Word2Vec and GloVe) were used to preserve semantic relationships between tokens.

These representations convert tokenized text into numerical vectors while retaining lexical meaning and contextual association. The most informative features were selected using chi-square tests, mutual-information scores, and feature-importance values derived from ensemble models, reducing dimensionality and limiting noise prior to model training.

The proposed hybrid architecture integrates two complementary components. The first is an LSTM network, a recurrent architecture designed for sequence modelling, which processes the token sequence to extract contextual and temporal patterns relevant to sentiment [26]. The second component is LightGBM, a gradient-boosting framework based on decision trees that is well suited to modelling categorical features and complex non-linear feature interactions [27].

In the hybrid design, the LSTM branch identifies contextual and temporal dependencies in the Roman Urdu text, while the LightGBM branch models non-linear interactions among the engineered features. The outputs of the two branches are subsequently combined to produce the final sentiment prediction.

$$X_{\text{seq}} = \{x_1, x_2, \ldots x_T\} \qquad (1)$$

For a given textual instance, the token sequence is provided as input to the LSTM branch, and the engineered feature vector X_feat (for example, TF-IDF features) is provided as input to the LightGBM branch. The LSTM component computes its hidden states using the standard gating operations.

The input gate and candidate memory-cell states at time t are given by Equations (2) and (3).

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \qquad (2)$$

$$\tilde{c}_t = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \qquad (3)$$

The forget gate is computed using Equation (4).

$$f_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_f\right) \qquad (4)$$

The updated memory-cell state at time t is obtained from Equation (5).

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \qquad (5)$$

The output gate and hidden state are computed using Equations (6) and (7).

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \qquad (6)$$

$$h_t = o_t \odot \tanh\left(c_t\right) \qquad (7)$$

Here, the sigmoid function is denoted by sigma and element-wise multiplication by the Hadamard operator; h_t and c_t denote the hidden and cell states at time t, and the weight matrices and bias vectors are learned during training. The final sequence representation produced by the LSTM is given by Equation (8).

The prediction logits produced by the LSTM branch are given by Equation (9).

$$^{Z}\mathrm{LSTM} = h_T \qquad (8)$$

The prediction logits produced by the LSTM branch are given by Equation (9).

$$^{s}\mathrm{LSTM} = W_\ell\, {}^{Z}\mathrm{LSTM} + b_\ell. \qquad (9)$$

LightGBM models the engineered feature vector X_feat as an additive ensemble of decision trees, as expressed in Equation (10).

$$F_{LightGBM}\left(X_{\text{feat}}\right) = \sum_{k=1}^{K} f_k\left(X_{\text{feat}}\right), \qquad (10)$$

where f_k denotes the k-th regression tree and K is the total number of trees. The output of the LightGBM branch is a vector of logits, given by Equation (11).

$$^{S}\mathrm{LGB} = F_{\mathrm{LightGBM}}\left(X_{\mathrm{feat}}\right) \tag{11}$$

The hybrid prediction logits are obtained by combining the two branches through a weighted sum.

$$^{S}\mathrm{hyb} = \alpha\, ^{S}\mathrm{LSTM} + (1-\alpha)\, ^{S}\mathrm{LGB} \tag{12}$$

where alpha is a mixing weight. In this study, alpha is selected on the validation set by grid search; alternatively, it may be learned as a scalar parameter or through a lightweight meta-learner. The final class probabilities are obtained by applying the Softmax function.

$$\hat{p} = \mathrm{softmax}\left(^{S}\mathrm{hyb}\right), \qquad \hat{y} = \arg\max \hat{p}. \tag{13}$$

During training, the categorical cross-entropy loss is minimised over the training set.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log \hat{p}_{i,c} \tag{14}$$

where $y_{i,c}$ represents the one-hot label for sample i and class c. The LSTM is trained via Adam optimization with a learning rate adjusted by validation, while LightGBM employs its inherent gradient boosting method; early stopping based on validation F1/accuracy is used for both components. Model performance is evaluated using Accuracy, Precision, Recall, and macro-F1 score. This hybrid architecture adheres to the conventional hybrid pipeline employed in recent applications that integrate deep sequence models with gradient boosting for final predictions [28].

This module processes the vectorized text features and intermediate representations obtained from the LSTM predictions to develop complementary feature sets and representations. The fusion layer integrates outputs from both the LSTM and LIGHTGBM components to create a comprehensive representation of the text data. This integration method represents text's global and local semantics with LSTM sequence context and LIGHTGBM high-level features. The hybrid model improves Roman Urdu sentiment analysis using LSTM and LIGHTGBM using this architectural approach.

3.3. Model Training and Evaluation

The hybrid model was trained and evaluated using supervised learning with stratified k-fold cross-validation. Stratified partitioning was selected because the sentiment dataset is class-imbalanced, and random splitting can over-represent individual classes within folds [29].

The LSTM branch was trained to capture temporal dependencies in the Roman Urdu text using Backpropagation Through Time with adaptive gradient descent, which mitigated vanishing-gradient effects. The LightGBM branch was trained using histogram-based split finding, which provides efficient handling of the categorical features common in Roman Urdu text. The fusion layer was trained on the combined outputs of both branches, with the integration (fusion) weights optimised by minimising the categorical cross-entropy loss using the Adam optimiser [30].

The Softmax layer normalises the contributions of the LSTM and LightGBM branches into a probability distribution, while the Adam optimiser adapts the learning rate to support stable convergence. This configuration was adopted because Softmax-based fusion balances the contributions of the two branches probabilistically, preventing either component from dominating, and because Adam provides effective optimisation in high-dimensional feature spaces, which is well suited to hybrid deep-learning frameworks.

Model performance was assessed using accuracy, precision, recall, and F1-score, computed across cross-validation folds. Hyperparameter tuning covered the LightGBM parameters, the LSTM architecture, and the configuration of the fusion layer, with systematic search used to identify settings that maximised validation performance. Iterative refinement following validation further improved classification accuracy on Roman Urdu text, allowing the framework to detect subtle sentiment cues across a range of informal usage.

4. Results

The performance of a deep-learning model depends on its network architecture, training procedure, and optimisation strategy. In this study, model performance was evaluated using accuracy, precision, recall, F1-score, and confusion-matrix analysis. These metrics quantify how accurately the model predicts positive, negative, and neutral sentiment while limiting false positives and false negatives.

4.1. Neural Network Architecture

The architecture of the LSTM branch, comprising the embedding, LSTM, and dense layers, is summarised in Figure 3. This configuration supports efficient analysis of sequential data and is widely used in natural language processing and time-series tasks.

4.1.1. Input and Embedding Layers

The embedding layer receives input sequences of fixed length 100. Incoming tokens are mapped to dense vector representations using an embedding matrix of dimension 40,647 x 50, allowing the model to represent a vocabulary of 40,647 tokens, each as a 50-dimensional vector. This representation enables the model to capture semantic relationships between tokens prior to sequential processing.

4.1.2. LSTM and Dense Layers

The LSTM layer processes the embedded sequences to extract temporal dependencies. Its main parameters are as follows:

1- the input kernel is a 50 x 400 weight matrix that processes incoming connections;
2- the recurrent kernel is a 100 x 400 weight matrix that governs connections from previous time steps;
3- and a bias vector of length 400 adjusts the output of the LSTM units. A non-linear activation function is applied after the LSTM layer to improve the model's capacity to represent complex patterns.

The final dense layer produces the model output. It comprises a 100 x 3 weight matrix connecting the LSTM layer to the output layer and a bias vector of length 3. A Softmax activation is applied to generate class predictions, and the dense layer produces the final three-class sentiment output. This structured design supports sequential-data modelling and can be adapted to related classification tasks.

4.2. Accuracy and Loss Evaluation

Figure 4 presents the accuracy and loss curves recorded for the LSTM model over 25 training epochs. The model reached stable accuracy after approximately ten epochs, with accuracy remaining consistent thereafter, indicating effective learning on the training data. The loss decreased sharply during the initial epochs before stabilising, indicating convergence. Together, these curves indicate stable and effective training

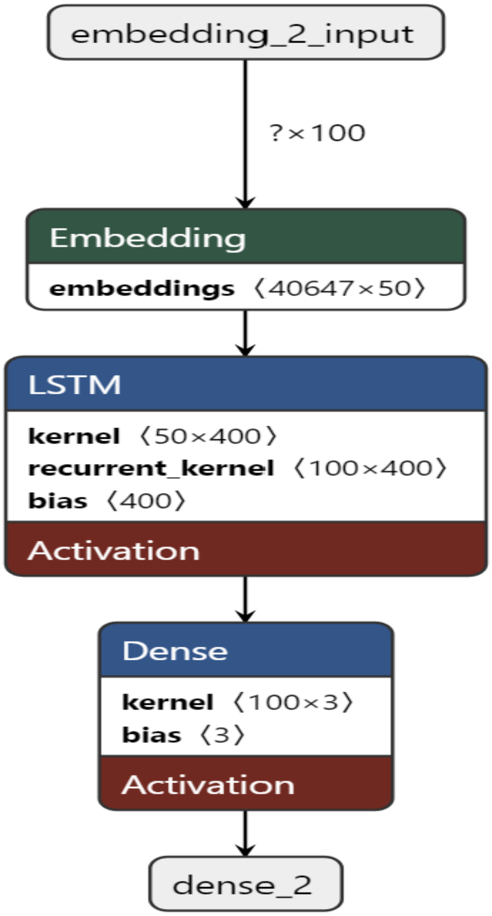

Figure 3 LSTM Model Overview

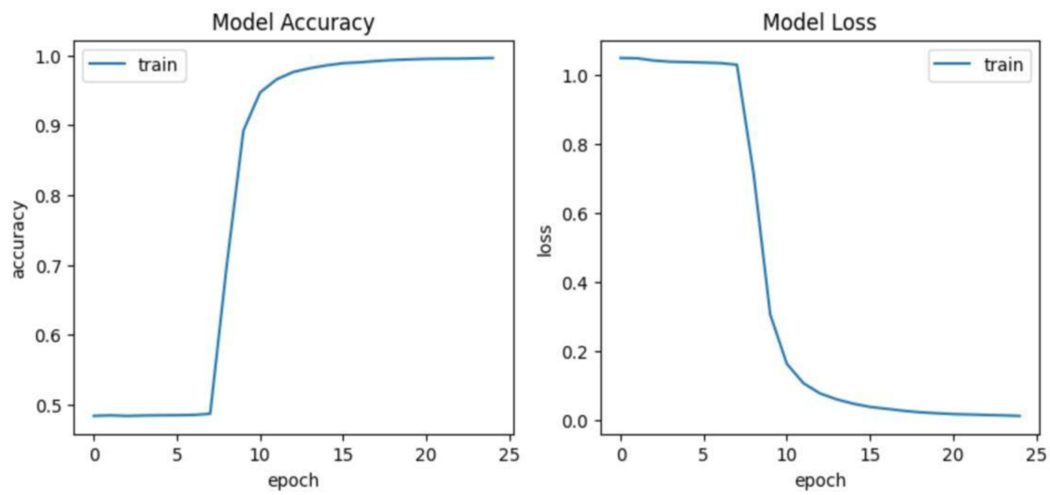

Figure 4 Accuracy and Loss Graph

The LightGBM model was trained independently on the Roman Urdu sentiment dataset. When evaluated as a standalone classifier, it achieved an accuracy of 0.69 (Figure 5), correctly predicting the sentiment of Roman Urdu text in 69% of cases.

```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of testing was 0.116663 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 69179
[LightGBM] [Info] Number of data points in the train set: 89085, number of used features: 500
[LightGBM] [Info] Start training from score -1.435731
[LightGBM] [Info] Start training from score -1.279958
[LightGBM] [Info] Start training from score -0.725650
Accuracy: 0.6951207192645722
```

Figure 5 LightGBM Accuracy

4.3. Hybrid Model Performance

The hybrid model, combining LightGBM and LSTM, classified Roman Urdu sentiment with high accuracy. By integrating sequential representation learning with gradient-boosted feature modelling, the hybrid framework was able to handle the spelling and grammatical variation characteristic of Roman Urdu, a code-mixed form combining Urdu and English. The per-class precision, recall, and F1-score of the hybrid model are reported in Figure 6. The combined architecture processed large volumes of text efficiently while capturing both contextual dependencies and non-linear feature interactions, improving overall accuracy and the handling of inherent ambiguity in Roman Urdu text.

```
2784/2784 [==============================] - 67s 24ms/step
310/310 [==============================] - 8s 25ms/step
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.028191 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 24870
[LightGBM] [Info] Number of data points in the train set: 89085, number of used features: 100
[LightGBM] [Info] Start training from score -1.435731
[LightGBM] [Info] Start training from score -1.279958
[LightGBM] [Info] Start training from score -0.725650
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.024694 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 25634
[LightGBM] [Info] Number of data points in the train set: 89085, number of used features: 103
[LightGBM] [Info] Start training from score -1.435731
[LightGBM] [Info] Start training from score -1.279958
[LightGBM] [Info] Start training from score -0.725650
Accuracy: 0.9774724719668654
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97      2358
           1       0.98      0.98      0.98      2717
           2       0.98      0.98      0.98      4824

    accuracy                           0.98      9899
   macro avg       0.98      0.98      0.98      9899
weighted avg       0.98      0.98      0.98      9899
```

Figure 6:Classification Report for Hybrid Model

4.4. Confusion Matrix and Comparative Analysis

Figure 7 presents the confusion matrices for the three classification models. In each matrix, rows correspond to the true sentiment classes and columns to the predicted classes, allowing class-level prediction performance to be examined. The matrices quantify true positives, false positives, true negatives, and false negatives, providing information beyond overall accuracy. Three models trained on the same Roman Urdu dataset were compared: a standalone LSTM model, a standalone LightGBM classifier, and the proposed hybrid architecture. This comparison isolates the contribution of each component and shows the effect of architectural integration on classification performance.

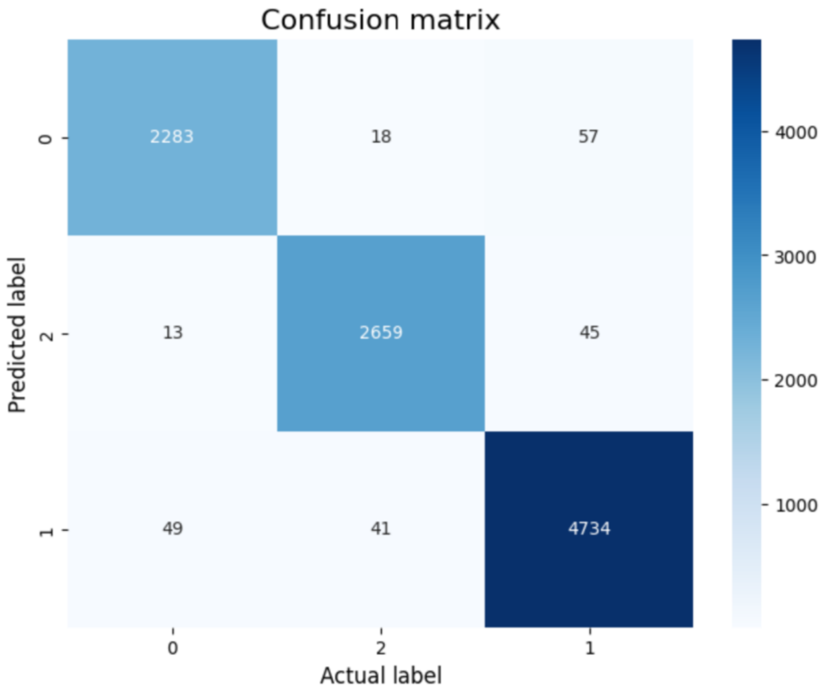

Figure 7 Confusion Matrix

4.5. Cross-Validation and Generalisation

The generalisation capability of the hybrid LightGBM-LSTM model was assessed using ten-fold cross-validation. The dataset was partitioned into ten folds, and training and testing were repeated across all folds. As shown in Figure 8, the model produced consistent performance across folds. The loss curves exhibited rapid initial convergence followed by continued optimisation at low error levels, indicating stable learning dynamics, and the mean accuracy across folds confirmed the model's tolerance to variation in the data.

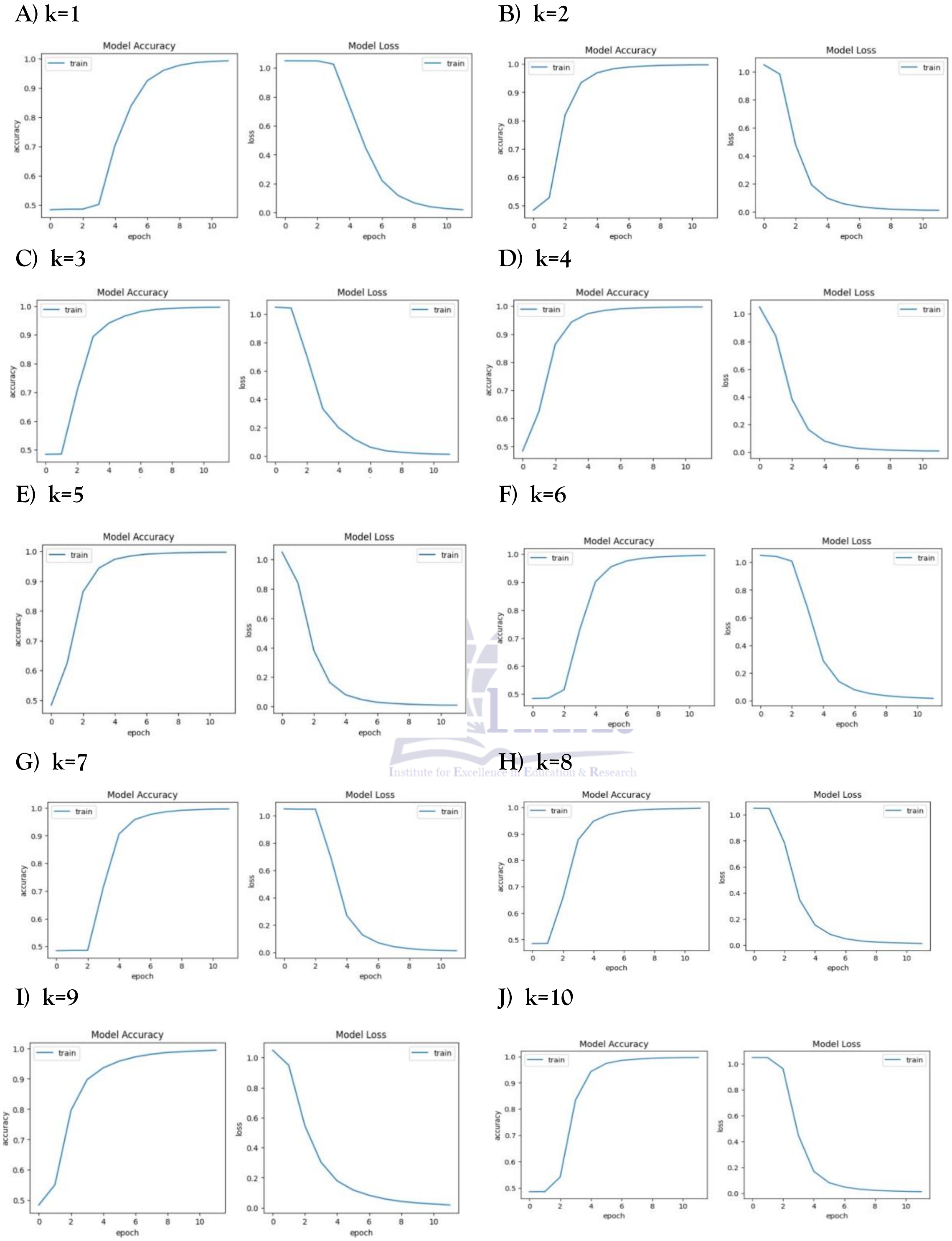

Figure 8: K-Fold Cross-Validation (Results)

These results indicate that the hybrid LightGBM-LSTM architecture achieves strong classification performance during training while maintaining robust generalisation to previously unseen Roman Urdu text. The balance between specialised representation learning and general applicability makes the framework well suited to Roman Urdu sentiment analysis.

4.6. Comparison with Baseline Models

Three architectures were compared for Roman Urdu sentiment analysis: a standalone LSTM model, a standalone LightGBM model, and the proposed hybrid model.

The LSTM model captured contextual dependencies and sentiment-bearing sequential features through its recurrent memory structure, whereas the gradient-boosted LightGBM model identified sentiment patterns through hierarchical feature interactions but did not model sequential structure directly. Combining the two architectures in the hybrid design produced higher performance than either component alone, as summarised in Table 1.

Table 1:Comparison of Model Accuracies

Model	Accuracy
LSTM	0.9372
LIGHTGBM	0.6951
LIGHTGBM-LSTM	0.9774

The proposed model was further compared with three conventional classifiers, Random Forest, SVM, and k-Nearest Neighbour (KNN), trained on the same Roman Urdu dataset. Random Forest achieved moderate accuracy but had difficulty capturing context and idiomatic expressions. SVM produced reasonable classification but did not model sequential relationships between words, which are important for sentiment interpretation. KNN performed least well, as its distance-based decision rule was sensitive to the informal and inconsistent nature of Roman Urdu. The corresponding results are reported in Table 2.

Table 2:Comparison of Proposed Model with Existing ML Classifiers

ML Classifier	Accuracy %
Random Forest	0.95 %
SVM	0.83 %
KNN	0.56 %
LIGHTGBM	0.71 %
Proposed LSTM-LIGHTGBM	0.97 %

5. Discussion

This study proposed a hybrid framework for Roman Urdu sentiment analysis that integrates LightGBM and LSTM architectures. The experimental results show that the hybrid model outperforms both standalone implementations and conventional baselines across accuracy, precision, recall, and F1-score. This improvement can be attributed to the complementary strengths of the two components: the LSTM branch models sequential and contextual structure, while the LightGBM branch efficiently captures non-linear interactions among engineered features. The study makes three main contributions. First, it addresses Roman Urdu, a widely used but under-resourced script for which many off-the-shelf models perform poorly because of the absence of standardised orthography. Second, it demonstrates that a hybrid deep-learning and gradient-boosting model handles informal and inconsistent text more effectively than either method individually. Third, it characterises linguistic features of Roman Urdu that are relevant to the design of NLP systems for low-resource and non-standardised languages. These findings are consistent with prior work reporting that systematic hyperparameter optimisation improves machine-learning performance on Roman Urdu tasks [31].

Considered in the context of the wider literature, these results show a clear pattern. Conventional classifiers such as SVM and Random Forest perform adequately for languages with standardised orthography and extensive linguistic resources, but are less effective for highly variable, non-standardised languages such as Roman Urdu. This observation is consistent with research on dialectal and code-mixed languages, where hybrid and deep-learning approaches have shown superior performance.

The hybrid model achieved an accuracy of 97.74%, compared with 93.72% for the standalone LSTM and 69.51% for the standalone LightGBM, indicating that combining sequential modelling with gradient-boosted decision trees improves feature extraction and classification. This performance gain supports the use of hybrid approaches for sentiment analysis in under-resourced languages, particularly where computational resources and standardised data are limited.

Several limitations should be acknowledged. The evaluation relied on a single Roman Urdu corpus, and validation across additional datasets would provide stronger evidence of generalisation. The present work also addressed three-class sentiment classification only; future studies could incorporate finer-grained sentiment categories or emotion detection [2]. Incorporating phonetic transcription or morphological analysis may help the model accommodate the linguistic variability of Roman Urdu, and attention mechanisms or cross-lingual transfer learning could further improve performance for low-resource settings [31].

In addition, although the dataset is large, it may not capture the full diversity and contextual complexity of Roman Urdu usage across social-media platforms. Because online language evolves continuously, periodic retraining may be required to maintain performance. Further investigation of feature engineering, text preprocessing, and hyperparameter optimisation may also yield additional improvements. While ten-fold cross-validation indicated robust generalisation, evaluation on independent external datasets would strengthen confidence in the model's reliability across domains. Future work could extend the framework to additional Roman Urdu text sources, explore alternative architectures, and assess the transferability of the hybrid approach to other low-resource languages and dialects; the use of Roman Urdu-specific contextual embeddings is a further promising direction.

The proposed framework has practical relevance for applications requiring sentiment analysis in low-resource and code-mixed language environments. It could support social-media monitoring systems that automatically assess Roman Urdu public sentiment, providing useful information for organisations, political analysts, and public-health initiatives [32]. Because conventional methods are limited by the non-standardised orthography of Roman Urdu, the framework may also assist in identifying hate speech and harmful content [33]. Its ability to process code-mixed text makes it suitable for chatbots and customer-support systems serving Roman Urdu speakers [34], and sentiment information of this kind can additionally inform cross-lingual market-analysis applications [35].

6. Conclusion

This study presented a hybrid LSTM-LightGBM framework for sentiment analysis of Roman Urdu text. By combining sequential representation learning with gradient-boosted feature modelling and a weighted Softmax fusion layer, the framework achieved a classification accuracy of 97.74%, outperforming standalone LSTM and LightGBM models as well as conventional classifiers. The results demonstrate that hybrid architectures are an effective strategy for sentiment analysis in non-standardised, low-resource languages. Beyond its academic contribution, the framework can support practical applications including social-media monitoring, content moderation, and customer-support systems for Roman Urdu speakers, and the overall approach may transfer to other low-resource and code-mixed languages. Future work should focus on validation across multiple datasets, finer-grained sentiment and emotion classification, the integration of attention mechanisms and contextual embeddings, and improvements in computational efficiency.

References

Rahman, M.M., et al., *RoBERTa-BiLSTM: A context-aware hybrid model for sentiment analysis.* IEEE Transactions on Emerging Topics in Computational Intelligence, 2025.

Akdogan, Y.E. and A. Anbar, *More than just sentiment: Using social, cognitive, and behavioral information of social media to predict stock markets with artificial intelligence and big data.* Borsa Istanbul Review, 2024. **24**: p. 61-82.

Liaras, E., M. Nerantzidis, and A. Alexandridis, *Machine learning in accounting and finance research: a literature review.* Review of Quantitative Finance and Accounting, 2024. **63**(4): p. 1431-1471.

Kotecha, N., *Artificial Intelligence in the Stock Market: The Trends and Challenges Regarding AI-Driven Investments.* Open Journal of Business and Management, 2025. **13**(2): p. 709-734.

Sehar, U., et al., *A hybrid dependency-based approach for Urdu sentiment analysis.* Scientific Reports, 2023. **13**(1): p. 22075.

Parvin, S., M. Sumathi, and R. Barani, *A Novel Approach to Classify Sentiments on Different Datasets Using Hybrid Approaches of Sentiment Analysis.* Indian Journal of Science and Technology, 2023. **16**(44): p. 3962-3970.

Hasan, M.A., *Ensemble language models for multilingual sentiment analysis.* arXiv preprint arXiv:2403.06060, 2024.

Liaqat, M.I., et al., *Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study.* PeerJ Computer Science, 2022. **8**: p. e1032.

Ghatora, P.S., et al., *Sentiment analysis of product reviews using machine learning and pre-trained llm.* Big Data and Cognitive Computing, 2024. **8**(12): p. 199.

Sai Aparna, T., et al., *Aspect-based sentiment analysis in Hindi: Comparison of machine/deep learning algorithms*, in *Inventive computation and information technologies: Proceedings of ICICIT 2020*. 2021, Springer. p. 81-91.

Goje, S.P. and R.H. Patil, *EXPLORING WORD EMBEDDINGS FOR SENTIMENT ANALYSIS OF MARATHI POLITICAL TWEETS: A MACHINE LEARNING APPROACH.* ICTACT Journal on Soft Computing, 2025. **15**(3).

Chaudhary, L., et al., *A review of deep learning models for twitter sentiment analysis: Challenges and opportunities.* IEEE Transactions on Computational Social Systems, 2023. **11**(3): p. 3550-3579.

Krasitskii, M., et al., *Advancing Sentiment Analysis in Tamil-English Code-Mixed Texts: Challenges and Transformer-Based Solutions.* arXiv preprint arXiv:2503.23295, 2025.

Mahmood, Z., et al., *Deep sentiments in roman urdu text using recurrent convolutional neural network model.* Information Processing & Management, 2020. **57**(4): p. 102233.

Kumar, A., C. Esposito, and D.A. Karras, *Introduction to special issue on misinformation, fake news and rumor detection in low-resource languages*. 2021, ACM New York, NY. p. 1-3.

Guidolin, M., *Machine learning in portfolio decisions*, in *Artificial Intelligence and Beyond for Finance*. 2024, World Scientific. p. 1-72.

Gogineni, A.K., et al., *A Hybrid Deep Learning Framework for Efficient Sentiment Analysis.* International Journal of Advanced Computer Science & Applications, 2023. **14**(12).

Farha, I.A. and W. Magdy, *A comparative study of effective approaches for Arabic sentiment analysis.* Information Processing & Management, 2021. **58**(2): p. 102438.

Diwali, A., et al., *Knowledge Distillation with Applications to Interpretable Arabic Sentiment Analysis.* 2024.

Ahmad, W., et al., *Enhanced sentiment analysis regarding COVID-19 news from global channels.* Journal of Computational Social Science, 2023. **6**(1): p. 19-57.

Abu-Salih, B., et al., *Natural language inference model for customer advocacy detection in online customer engagement.* Machine Learning, 2024. **113**(4): p. 2249-2275.

Alemayehu, F., M. Meshesha, and J. Abate, *Amharic political sentiment analysis using deep learning approaches.* Scientific Reports, 2023. **13**(1): p. 17982.

Dioubi, F., et al., *Enhancing stock market predictions via hybrid external trend and internal components analysis and long short term memory model.* Journal of King Saud University-Computer and Information Sciences, 2024. **36**(10): p. 102252.

Oueslati, O., et al., *A review of sentiment analysis research in Arabic language.* Future Generation Computer Systems, 2020. **112**: p. 408-430.

Merayo, N., et al., *Social network sentiment analysis using hybrid deep learning models.* Applied Sciences, 2023. **13**(20): p. 11608.

Hochreiter, S. and J. Schmidhuber, *Long short-term memory.* Neural computation, 1997. **9**(8): p. 1735-1780.

Ke, G., et al., *Lightgbm: A highly efficient gradient boosting decision tree.* Advances in neural information processing systems, 2017. **30**.

Gao, Q., *Multi-temporal Scale Wind Power Forecasting Based on Lasso-CNN-LSTM-LightGBM.* EAI Endorsed Transactions on the Energy Web, 2024. **11**(1).

Kohavi, R. *A study of cross-validation and bootstrap for accuracy estimation and model selection*. in *Ijcai*. 1995. Montreal, Canada.

Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization.* arXiv preprint arXiv:1412.6980, 2014.

Ashiq, W., et al., *Roman urdu hate speech detection using hybrid machine learning models and hyperparameter optimization.* Scientific Reports, 2024. **14**(1): p. 28590

Kazbekova, G., et al., *Offensive Language Detection on Online Social Networks using Hybrid Deep Learning Architecture.* International Journal of Advanced Computer Science & Applications, 2023. **14**(11).

Khan, M.M., K. Shahzad, and M.K. Malik, *Hate speech detection in roman urdu.* ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP), 2021. **20**(1): p. 1-19.

Husnain, M., et al., *How Chatbot negative experiences damage consumer-brand relationships in hospitality and tourism? A mixed-method examination.* International Journal of Hospitality Management, 2025. **126**: p. 104076.

Wang, Z., S.-B. Ho, and E. Cambria, *A review of emotion sensing: categorization models and algorithms.* Multimedia Tools and Applications, 2020. **79**(47): p. 35553-35582.