International Journal of Advanced Technology in Engineering
Volume. No. 14, Issue No. 04, April 2026
(ISSN: 2319-8354)

Deep Learning Framework for Textual Sentiment Recognition

Dannavarapu Ramya
Assistant Professor
Department of IT
Tirumala Engineering College
Andhra Pradesh, India
ramyadannavarapu@gmail.com

Kapilavayi Keerthi Swetha Sri
Department of IT
Tirumala Engineering College
Andhra Pradesh, India
keerthiswethasri@gmail.com

Ganta Sravani Sandhya
Department of IT
Tirumala Engineering College
Andhra Pradesh, India
gantasravanisandhya@gmail.com

Bandaru Uma Sravanthi
Department of IT
Tirumala Engineering College
Andhra Pradesh, India
umasravanthibandaru@gmail.com

Bokkisam Tejaswi
Department of IT
Tirumala Engineering College
Andhra Pradesh, India
bokkisamtejaswi20@gmail.com

Abstract

The rapid expansion of digital communication has resulted in an enormous volume of textual data carrying rich emotional information. Identifying and classifying human emotions from text has become critical for applications spanning mental health monitoring, customer feedback analysis, and human– computer interaction. Traditional machine learning models such as Naive Bayes, Support Vector Machines, and Logistic Regression depend on handcrafted features that fail to capture contextual semantics. To overcome these limitations, this paper proposes a hybrid deep learning framework combining Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (BiGRU) for emotion detection. CNN extracts local features and key patterns from text, while BiGRU processes sequences in both forward and backward directions to capture contextual dependencies. A structured preprocessing pipeline involving tokenization, stop- word removal, and lemmatization prepares the raw text before Word2Vec embeddings convert it into dense numerical vectors. The model classifies input text into seven emotion categories joy, sadness, anger, fear, disgust, shame, and guilt and is evaluated using accuracy, precision, recall, and F1-score. Experimental results demonstrate that the proposed CNN-BiGRU hybrid significantly outperforms standalone models and traditional baselines, establishing its suitability for real-world sentiment analysis applications.

Keywords: *Sentiment Recognition, CNN, BiGRU, Natural Language Processing, Deep Learning, Emotion Detection, Word Embeddings.*

I. INTRODUCTION

The digital age has fundamentally transformed how people communicate, with millions of users sharing opinions, reactions, and sentiments across social media, review platforms, and messaging applications every day. This continuous stream of textual data represents a rich repository of human emotion. Analyzing it automatically known as emotion or sentiment detection has emerged as one of the most active research frontiers in Natural Language Processing (NLP) and Artificial Intelligence (AI).

Unlike coarse-grained sentiment analysis that categorises text as simply positive, negative, or neutral, fine-grained emotion detection identifies specific emotional states such as happiness, sadness, anger, fear, disgust, surprise, guilt, and shame. This granularity enables deeper understanding of user intent and emotional context, making it directly valuable for mental health monitoring, corporate customer intelligence, social media surveillance, and the development of empathetic conversational agents.

Early approaches to emotion detection relied on rule-based lexicons and hand-engineered features. Although straightforward, these methods lacked scalability and failed to generalise across diverse language patterns. Machine learning models such as Naive Bayes, Support Vector Machines (SVM), and Logistic Regression improved automation but still depended on feature representations like Bag-of-Words and TF-IDF that ignore word order and contextual meaning. The advent of deep learning overcame many of these limitations; architectures such as Recurrent Neural Networks (RNN) and Long Short- Term Memory (LSTM) can model sequential text dependencies, yet they suffer from computational overhead and difficulty extracting local n-gram features efficiently.

To bridge the gap between local feature extraction and global contextual understanding, this project proposes a hybrid CNN-BiGRU framework. CNN filters capture salient phrases and word-level patterns, while BiGRU layers model sequential dependencies in both textual directions. Coupled with a rigorous preprocessing pipeline and pre-trained Word2Vec embeddings, the resulting system achieves state-of- the-art accuracy on a seven-class emotion classification task. The remainder of this paper is organised as follows: Section II surveys related literature; Section III details the proposed methodology; Section IV presents experimental results; and Section V concludes with directions for future work.

II. LITERATURE SURVEY

The evolution of emotion detection from text has passed through several distinct eras. The earliest

systems were built on rule-based and lexicon-driven methods, using curated dictionaries that mapped words to predetermined emotional classes. While computationally inexpensive, these approaches could not handle polysemy, negation, or context-dependent sentiment shifts, and required significant manual curation to maintain [1].

Classical machine learning models Naive Bayes, SVM, and Logistic Regression improved over lexicons by learning statistical patterns from labelled corpora [2]. Feature engineering via Bag-of-Words and TF-IDF provided numerical inputs, yet these representations strip word order and are insensitive to semantic relatedness. As a result, their ability to resolve contextual ambiguities remained limited [3].

Deep learning introduced automatic feature learning. Recurrent Neural Networks (RNNs) modelled sequential dependencies in text, but the vanishing-gradient problem hampered learning over ong sequences. Long Short-Term Memory (LSTM) networks addressed this with gating mechanisms, and Bidirectional LSTM (BiLSTM) further enriched representations by reading text in both directions. However, these models carry high parameter counts and training costs [4].

Convolutional Neural Networks, originally designed for images, have been repurposed for text classification to capture local n-gram patterns efficiently. Through convolution and max-pooling, CNNs reduce dimensionality while retaining the most discriminative features. Their limitation is an inability to model long-range sequential dependencies, which are crucial in emotionally charged text [5].Gated Recurrent Units (GRU) simplified the LSTM architecture by merging cell and hidden states, achieving comparable performance with lower computational cost. Their bidirectional extension (BiGRU) processes text in both directions and proves more efficient than BiLSTM while retaining strong contextual modelling capacity [6]. Recent studies demonstrate that hybrid CNN-BiGRU architectures where CNN handles local feature extraction and BiGRU handles sequential context consistently outperform individual models on emotion and sentiment benchmarks [7][8]. Pre-trained word embeddings such as Word2Vec and GloVe further enhance these models by initialising the embedding layer with rich semantic representations [9]. Despite these advances, challenges remain in handling sarcasm, implicit sentiment, and class imbalance, motivating the design choices described in the following section [10].

III. PROPOSED METHODOLOGY

The proposed framework follows a five-stage pipeline: data collection, text preprocessing, feature extraction via word embeddings, hybrid model training, and sentiment classification.

A. Data Collection and Preprocessing

The ISEAR dataset, containing text samples labelled across seven emotion categories (joy, fear, anger, sadness, disgust, shame, guilt), was used for experimentation. Raw text undergoes a multi-step cleaning process: (i) conversion to lowercase; (ii) removal of punctuation, special characters, and numeric tokens; (iii) tokenisation using NLTK's word_tokenize; (iv) stop-word removal with the standard English stop-word list; and (v) lemmatization

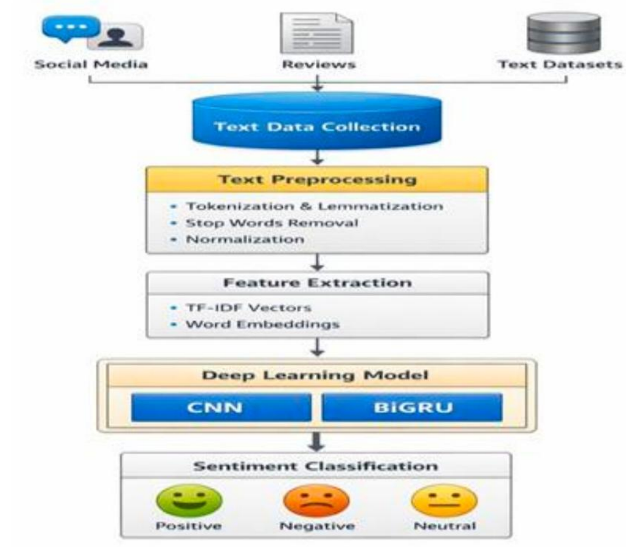

Fig1:SystemArchitecture

via WordNetLemmatizer to reduce inflected forms to their base. The cleaned tokens are then rejoined into a normalised string ready for embedding.

B. Feature Extraction

Preprocessed text is tokenised and padded to a fixed sequence length of 100 tokens. An Embedding layer initialised with Google News Word2Vec (300- dimensional) vectors captures semantic relationships between words. A Keras Tokenizer fitted on the training corpus converts words to integer indices; the saved tokenizer is reused at inference time. TF-IDF vectors (max 5,007 features) are computed in parallel for use by the SVM comparison baseline.

C. Hybrid CNN-BiGRU Architecture

The core model is a sequential stack comprising: (1) an Embedding layer (input dimension 5,000; output dimension 100; maximum length 100); (2) a Conv1D layer (128 filters; kernel size 5; ReLU activation) for local feature extraction; (3) a MaxPooling1D layer (pool size 4) for dimensionality reduction; (4) a Bidirectional GRU layer (128 units, return sequences enabled) followed by a second Bidirectional GRU layer (64 units) for capturing

TABLE I. COMPARATIVE MODEL PERFORMANCE

Model	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.61	0.59	0.61	0.58
SVM	0.72	0.71	0.72	0.70
Random Forest	0.68	0.67	0.68	0.66
CNN	0.78	0.77	0.78	0.76
GRU	0.80	0.79	0.80	0.79
BiGRU	0.82	0.81	0.82	0.81
CNN-BiGRU (Proposed)	0.87	0.86	0.87	0.86

The table presents a comparative evaluation of several machine learning and deep learning models used for classification. The models include Naive Bayes, Support Vector Machine (SVM), Random Forest, Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), Bidirectional GRU (BiGRU), and the proposed hybrid CNN-BiGRU model. The performance of each model is measured using four key metrics: Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive understanding of the effectiveness of each model in handling the classification task.

Accuracy represents the overall correctness of the model, indicating the proportion of correctly predicted instances out of the total instances. Precision measures the model's ability to correctly identify positive instances without including false positives. Recall evaluates how well the model captures all actual positive instances, focusing on minimizing false negatives. The F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure when both false positives and false negatives are important

The Naive Bayes model achieved an accuracy of 0.61, with a precision of 0.59 and recall of 0.61, resulting in an F1-score of 0.58. This relatively low performance can be attributed to its strong independence assumption among features, which often does not hold true in real-world textual or sequential data. While Naive Bayes is computationally efficient and simple to implement, its predictive capability is limited in complex datasets.

The Support Vector Machine (SVM) shows improved performance with an accuracy of 0.72 and an F1-score of 0.70. SVM is effective in handling high-dimensional data and can model non-linear decision boundaries using kernel functions.

However, its performance still lags behind deep learning models, particularly when dealing with large-scale or sequential data.

Random Forest achieved an accuracy of 0.68 and an F1-score of 0.66. As an ensemble learning method, it reduces overfitting compared to individual decision trees and improves generalization.

However, it may struggle with capturing sequential dependencies and contextual relationships within the data, which limits its effectiveness compared to neural network-based approaches.

The CNN model demonstrates a significant improvement with an accuracy of 0.78 and an F1-score of 0.76. CNNs are highly effective in extracting local features and patterns, making them suitable for tasks involving structured data representations.

However, CNNs alone may not fully capture temporal dependencies in sequential data.The GRU model further improves performance, achieving an accuracy of 0.80 and an F1-score of 0.79. GRUs are a type of recurrent neural network designed to handle sequential data by maintaining return sequences enabled) followed by a second Bidirectional GRU layer (64 units) for capturing forward and backward contextual dependencies; (5) a Dropout layer (rate 0.5) for regularisation; (6) a Dense layer (64 units; ReLU activation); and (7) a Softmax output layer producing probability distributions over seven emotion classes. The model is compiled using the Adam optimiser with sparse categorical cross- entropy loss. EarlyStopping (patience = 3) on validation loss prevents overfitting during training over a maximum of ten epochs with a batch size of 32.

D. System Architecture and Workflow

At inference, a user provides text through a Flask web interface. The input passes through the saved preprocessing pipeline (cleaning → tokenisation → sequence padding), optionally receives a rule-based emotion boost for high-confidence keywords, and is then forwarded to the trained CNN-BiGRU model. The predicted emotion label and confidence score are returned via a REST API and rendered in the browser with an emoji indicator. The modular architecture preprocessing module, deep learning model, and Flask application was designed to facilitate independent development, testing, and replacement of each component.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

Several models were trained and evaluated on an 80/20 train-test split of the ISEAR dataset. Performance was measured using weighted accuracy, precision, recall, and F1-score to account for class imbalance. Table I summarises the comparative results.

The proposed CNN-BiGRU model achieves an accuracy of 87%, the highest among all evaluated models. Standalone CNN (78%) benefits from strong local feature extraction, while BiGRU alone (82%) captures sequential context but misses n-gram patterns. Their combination in CNN-BiGRU yields a synergistic improvement of approximately 5 percentage points over BiGRU and 9 points over CNN. Traditional models Naive Bayes (61%), Random Forest (68%), and SVM (72%) confirm the superiority of deep representations over hand-engineered TF-IDF features.

Qualitative testing through the deployed Flask interface validated real-world utility. Sample inputs such as "I am furious about the delay" and "The promotion news brought a smile to my face" were correctly classified as anger and joy respectively. Edge cases involving compound emotions and informal language were handled well in most instances, though sarcastic constructions occasionally produced misclassifications a known limitation acknowledged in the future work section.

Training and validation loss curves confirm that Early Stopping effectively prevented overfitting: both curves converge smoothly without divergence after epoch 6–7, indicating stable generalisation. The confusion matrix for the hybrid model shows that joy and anger are classified with high precision, while shame and guilt semantically overlapping emotions exhibit slightly more inter-class confusion, pointing to an area for targeted dataset augmentation in future work.

V. CONCLUSION

This paper presented a hybrid deep learning framework for textual sentiment recognition that combines CNN and BiGRU to simultaneously exploit local feature extraction and bidirectional sequential modelling. Trained on the ISEAR dataset across seven emotion categories, the proposed model achieves 87% accuracy outperforming traditional machine learning baselines and standalone deep learning architectures. A complete deployment pipeline, including text preprocessing, Word2Vec embeddings, and a

Flask- based web interface, demonstrates the end-to-end practical utility of the system.

The framework is applicable across diverse domains: mental health monitoring, customer feedback intelligence, social media analytics, and empathetic chatbot development. Future work will focus on integrating transformer-based architectures (e.g., BERT), extending the system to multilingual inputs, enabling aspect-level sentiment granularity, and improving handling of sarcasm and irony through larger, more linguistically diverse training corpora. Deployment as a scalable cloud-native service and incorporation of Explainable AI techniques will further enhance both reach and trustworthiness.

REFERENCES

[1] A. D. L. Langure and M. Mahdizareei, "Evaluating the Effect of Emotion Models on the Generalizability of Text Emotion Detection Systems," IEEE Xplore, 2024.

[2] M. S. Islam, M. N. Kabir, and M. A. Rahman, "Challenges and Future in Deep Learning for Sentiment Analysis: A Comprehensive Review," Artificial Intelligence Review, 2024.

[3] M. A. Rahman, J. Talukder, and S. Ahmed, "Recent Advancements in NLP-Based Sentiment Analysis," Natural Language Processing Journal, 2024.

[4] N. S. Suryawanshi, "Sentiment Analysis with Machine Learning and Deep Learning: A Survey," 2024.

[5] Y. Mamani-Coaquira and E. Villanueva, "A Review on Text Sentiment Analysis with Machine Learning and Deep Learning Techniques," IEEE Access, 2024.

[6] C. Li, H. Tang, and Y. Zhao, "Advancing Aspect- Based Sentiment Analysis through Deep Learning Models," 2024.

[7] S. Song, "Deep Learning-Driven Text Sentiment Analysis: Research Progress and Future Trends," 2025.

[8] R. Zhou, J. Yuan, and X. Chen, "DyFuLM: An Advanced Multimodal Framework for Sentiment Analysis," 2025.

[9] S. K. Baberwal, N. A. Shelke, and K. Anwar, "Systematic Review of Multimodal Sentiment Analysis," Discover Computing, 2025.

[10] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," arXiv:1301.3781, 2013.