

ARTICLE

LinBGN-Net: A Unified Linguistic Framework for Multi-Task Text Classification in Telugu Using BERT, GCN, and Naive Bayes

Asif Hussain Shaik ^{1*} , Fahimuddin Shaik ² , Karimullah Shaik ^{2*} , Sureshbabu G ³ 

¹ Centre for Research and Consultancy, Middle East college, Muscat 113, Oman

² Department of ECE, Annamacharya University, Rajampet 516126, India

³ Department of ECE, Annamacharya University, Rajampet 516126, India Department of MECH, Annamacharya University, Rajampet 516126, India

ABSTRACT

Despite the growing interest in Natural Language Processing (NLP), linguistically grounded multi-task modeling for low-resource languages like Telugu remains underexplored, particularly across semantically complex tasks such as sentiment, emotion, hate speech, sarcasm, and clickbait detection. This study aims to address this gap by conducting a comprehensive linguistic analysis of Telugu texts through the lens of computational modeling. To that end, we propose LinBGN-Net, an innovative ensemble deep learning framework that integrates BERT for contextual embedding, Graph Convolutional Networks (GCN) for structural understanding, and Naive Bayes (NB) for statistical grounding. The model is trained on five well-balanced Telugu datasets (totalling over 250,000 sentences) from Hugging Face, covering diverse label spaces (2 to 5 classes), thus facilitating an in-depth linguistic study across varied text genres and expressions. LinBGN-Net achieves macro-F1 scores of 0.86 (Sentiment), 0.84 (Emotion), 0.93 (Hate Speech), 0.92 (Sarcasm), and 0.95 (Clickbait), outperforming standard baselines such as Naive Bayes, SVM, LSTM, and even standalone BERT by margins of 2–8% across tasks. The results not only demonstrate the effectiveness of LinBGN-Net as a high-performing multi-task model, but also offer valuable linguistic insights into how Telugu expressions of sentiment, emotion, intent, and persuasion can be computationally modeled and understood—contributing significantly to both linguistic research and real-world NLP

*CORRESPONDING AUTHOR:

Asif Hussain Shaik, Centre for Research and Consultancy, Middle East college, Muscat 113, Oman; Email: shussain@mec.edu.com; Karimullah Shaik, Department of ECE, Annamacharya University, Rajampet 516126, India; Email: munnu483@gmail.com

ARTICLE INFO

Received: 12 April 2025 | Revised: 14 May 2025 | Accepted: 16 May 2025 | Published Online: 15 July 2025

DOI: <https://doi.org/10.30564/fls.v7i7.9470>

CITATION

Shaik A.H., Shaik F., Shaik K., et al., 2025. LinBGN-Net: A Unified Linguistic Framework for Multi-Task Text Classification in Telugu Using BERT, GCN, and Naive Bayes. *Forum for Linguistic Studies*. 7(7): 518–539. DOI: <https://doi.org/10.30564/fls.v7i7.9470>

COPYRIGHT

Copyright © 2025 by the author(s). Published by Bilingual Publishing Group. This is an open access article under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) License (<https://creativecommons.org/licenses/by-nc/4.0/>).

applications.

Keywords: Linguistic Analysis; Telugu; Language; NLP; Emotion; Sarcasm; Sentiment

1. Introduction

Telugu, one of the major Dravidian languages spoken predominantly in the Indian states of Andhra Pradesh and Telangana, is renowned for its linguistic richness and classical literary heritage. With a history spanning over a millennium, Telugu boasts a well-developed grammar, an extensive vocabulary, and a rich corpus of poetry and prose^[1]. It exhibits significant dialectal diversity, with distinct regional variations such as Coastal Andhra, Rayalaseema, and Telangana dialects, each reflecting unique phonetic, lexical, and syntactic traits^[2]. Despite its cultural and linguistic significance, Telugu remains digitally underrepresented, particularly in the context of natural language processing and AI applications. The scarcity of annotated datasets, tools, and resources limits the development of robust language technologies, hindering digital inclusion for millions of Telugu speakers in the rapidly advancing digital age^[3].

Linguistic analysis in regional and morphologically rich languages like Telugu is crucial for understanding sentiment, emotion, intention, and contextual variation in communication. Despite the growing demand for such analysis in the digital era, Telugu remains a low-resource language in the computational linguistics landscape^[4]. Challenges such as complex word formations, script intricacies, polysemy, and code-mixing make it difficult to model Telugu text using conventional NLP methods. Moreover, most available studies focus on isolated tasks rather than exploring the interconnected nature of linguistic cues like sarcasm, emotion, and clickbait intent, which often coexist in real-world communication^[5]. This creates a significant research gap in holistic, task-unified approaches to Telugu linguistic analysis^[6].

In this context, our work presents a multi-dimensional linguistic study of Telugu by simultaneously addressing five vital language understanding tasks: sentiment analysis, emotion detection, hate speech identification, sarcasm detection, and clickbait classification. These tasks are central to both sociolinguistic research and applied content filtering systems^[7]. To facilitate this study, we use balanced datasets collected from Hugging Face, containing over 250,000 annotated sentences, and conduct a comprehensive evaluation of

linguistic variations reflected across labels and categories^[8].

To enhance the linguistic understanding of Telugu texts computationally, we propose LinBGN-Net—a multi-task ensemble framework that integrates three powerful yet diverse computational perspectives:

- BERT, which captures contextual semantics and syntactic richness,
- Graph Convolutional Networks (GCN), which model relational dependencies among words and sentences in graph form,
- Naive Bayes, which provides statistical insights into word frequency and class distribution.

This architecture enables multi-perspective linguistic interpretation, allowing the model to generalize across tasks while offering a detailed analysis of sentence composition and label association. Evaluation results show that LinBGN-Net achieves F1-scores ranging from 0.84 to 0.95, outperforming classical and transformer-only baselines. Furthermore, confusion matrices, ROC curves, error analysis, and SHAP-like interpretability provide a window into the model's decision-making, making the results linguistically transparent and analytically rich^[9].

The remainder of this paper is organized as follows: Section 2 reviews related work in Telugu NLP and linguistic modeling; Section 3 outlines the Methodology of the proposed system. Section 4 discusses the experimentation, results and evaluation metrics; Section 5 concludes the paper with insights and directions for future linguistic exploration^[10].

The primary objective of this study is to introduce LinBGN-Net, a linguistically grounded, multi-task framework for emotion detection and sentiment analysis in Telugu, a resource-poor Dravidian language. This work advances NLP research by integrating graph neural reasoning with transformer-based token modeling to better handle low-resource, high-context scenarios.

2. Related Works

In^[11] authors proposed a machine learning-based emo-

tion detection system utilizing Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers. Their study, which analyzed Twitter data, found that SVM achieved the highest accuracy of 85.7%, highlighting the effectiveness of conventional ML algorithms in accurately classifying six emotional categories.

In^[12] authors developed a text mining-based framework incorporating Natural Language Processing (NLP) and machine learning techniques. Their approach focused on emotion detection from social media content, showcasing how NLP preprocessing—such as tokenization and stop word removal—significantly boosts classification performance in ML pipelines.

In 2022, authors^[13] examined deep learning and transformer-based architectures for emotion detection in multilingual settings. Their research emphasized the benefits of transfer learning and highlighted how these models handle linguistic diversity effectively, thereby improving classification performance across varied language inputs.

In 2023, authors of^[14] adopted Bi-LSTM and a hybrid LSTM-MLP model to process emotion-labeled datasets. Their model achieved an impressive accuracy of approximately 91%, underscoring the strength of LSTM architectures in capturing long-range dependencies in emotional text data.

In 2022, authors of^[15] proposed a hybrid model that combined lexicon-based approaches with machine learning techniques. Targeted particularly at code-mixed and low-resource languages, their approach showed enhanced accuracy through the integration of linguistic resources, such as emotion lexicons, with computational models.

In 2021, authors of^[16] used the VADER sentiment analysis tool for real-time emotion tracking on Twitter. Their work demonstrated the utility of VADER in monitoring public sentiment during live events, providing efficient and interpretable results in dynamic contexts.

In 2023, authors of^[17] utilized a fine-tuned BERT model trained on emotion-specific datasets. They demonstrated that BERT's contextual understanding and deep representation learning outperformed traditional ML approaches, achieving improved accuracy in classifying subtle emotional cues.

In 2020, authors of^[18] implemented a Convolutional Neural Network (CNN) enhanced with an attention mech-

anism. Their architecture enabled the model to focus on emotionally salient parts of the text, which led to increased precision and better emotion detection results.

In 2023, authors of^[19] created an ensemble model combining Naive Bayes, SVM, and Random Forest classifiers. The ensemble approach delivered superior performance over individual models, achieving higher F1-scores across multiple emotional categories.

In 2019, authors of^[20] designed a hybrid system that merged rule-based logic with machine learning for improved sarcasm detection. Their method effectively captured complex emotional expressions by fusing handcrafted linguistic rules with data-driven classification.

In 2018, authors of^[21] presented an interdisciplinary framework that merged keyword-based, learning-based, and hybrid strategies for emotion detection. The study emphasized the use of both syntactic and semantic textual features and advocated for a psychological and computational synergy to increase the accuracy and applicability of emotion detection systems.

In 2022, authors of^[22] proposed a multimodal emotion detection approach that analyzed both text and images using NLP techniques. Their framework enhanced the understanding of sender intent across various platforms by incorporating visual and textual emotion cues.

In 2024, authors of^[23] introduced a novel knowledge distillation method that transferred emotion detection capabilities from a high-performing monolingual model to a multilingual student model. This approach notably improved the performance of multilingual models such as XLM-RoBERTa, while also enhancing interpretability through better identification of emotion-triggering words.

In 2020, authors of^[24] focused on speech-based emotion recognition by removing emotionally charged words from the feature extraction process. Using OpenSMILE and OpenXBoAW toolkits, they showed that emphasizing acoustic features led to improved valence prediction performance.

In 2023, authors of^[25] explored emotion detection in Roman Urdu using Multilingual BERT in combination with clustering techniques. Their system achieved 91% accuracy and was rigorously validated using Silhouette Index (SI) and Calinski-Harabasz Index (CHI), affirming the effectiveness of the model in processing under-resourced

languages.

In 2024, authors of^[26] introduced a hybrid model tailored for the Turkish language, leveraging word embeddings and deep learning. Their method achieved robust results in low-resource environments, demonstrating adaptability in scenarios where annotated data or linguistic tools are scarce.

In 2025, authors of^[27] developed the Deep Learning Semantic Text Analyzer (DLSTA), which integrated BERT and advanced NLP techniques. Their system achieved a high emotion detection rate of 96.22% and classification accuracy of 97.92%, positioning it among the most accurate frameworks in the domain.

In 2021, authors of^[28] utilized a lexicon-based approach using the EmoLex emotion dictionary to analyze Indonesian tweets during the COVID-19 pandemic. Their study tracked emotional trends over time and identified fear and anger as the dominant emotions expressed during the pandemic months.

In authors of^[29] proposed an ensemble neural network model for emotion detection in code-switching environments. Integrating CNN, RCNN, and Attention-LSTM components, their architecture proved competitive in multilingual and complex linguistic settings.

In 2016, authors of^[30] focused on speech-based emotion detection in the Marathi language using neural networks. By incorporating multi-feature analysis specifically MFCC, pitch, and energy they significantly enhanced the accuracy of spoken emotion recognition^[31].

3. Proposed System

The diagram illustrates the workflow of a multi-task emotion detection system. It begins with data acquisition, followed by exploring and splitting the dataset. Text preprocessing is then applied to clean and prepare the data, after which tokenization and padding are performed. The labels are encoded, and data from multiple tasks are merged. The model training step uses a multi-task approach and is compared with baseline models. A specialized architecture, LinBGN-Net, is then applied, and finally, the system's performance is evaluated and visualized for interpretation.

3.1. Language Normalization

Language normalization involved standardizing text by correcting spelling variations, handling regional lexical differences, and converting non-standard forms into their canonical equivalents. This step ensured uniformity across diverse dialects and writing styles in the Telugu corpus.

3.2. Tokenization via IndicNLP

Tokenization was performed using the IndicNLP library, which is tailored for Indian languages and effectively handles Telugu's agglutinative morphology. This process segmented sentences into linguistically meaningful units while preserving script integrity and context.

3.3. Manual and Semi-Automatic Labeling

Annotated data was generated through a combination of manual labeling by native Telugu speakers and semi-automatic techniques using seed lexicons and rule-based heuristics. This hybrid approach ensured both scalability and linguistic accuracy in emotion and intent annotation.

3.4. Quality Checks on Class Balance and Annotation Agreement

To maintain dataset reliability, we conducted thorough checks for class balance across emotion categories and measured inter-annotator agreement using metrics such as Cohen's kappa. Discrepancies were resolved through iterative reviews and consensus-based re-annotation.

The **Figure 1** presents the workflow of the LinBGN-Net framework for multi-task learning in NLP. It begins with data acquisition, followed by dataset exploration and splitting into training, validation, and test sets. Text preprocessing is then applied to clean and normalize the data. The processed text undergoes tokenization and padding to ensure uniform input length. Label encoding is performed, and task-specific labels are merged for multi-task learning. The LinBGN-Net architecture is then applied, followed by model training in a multi-task setting. The performance is compared with baseline models, and final results are evaluated and visualized for interpretation.

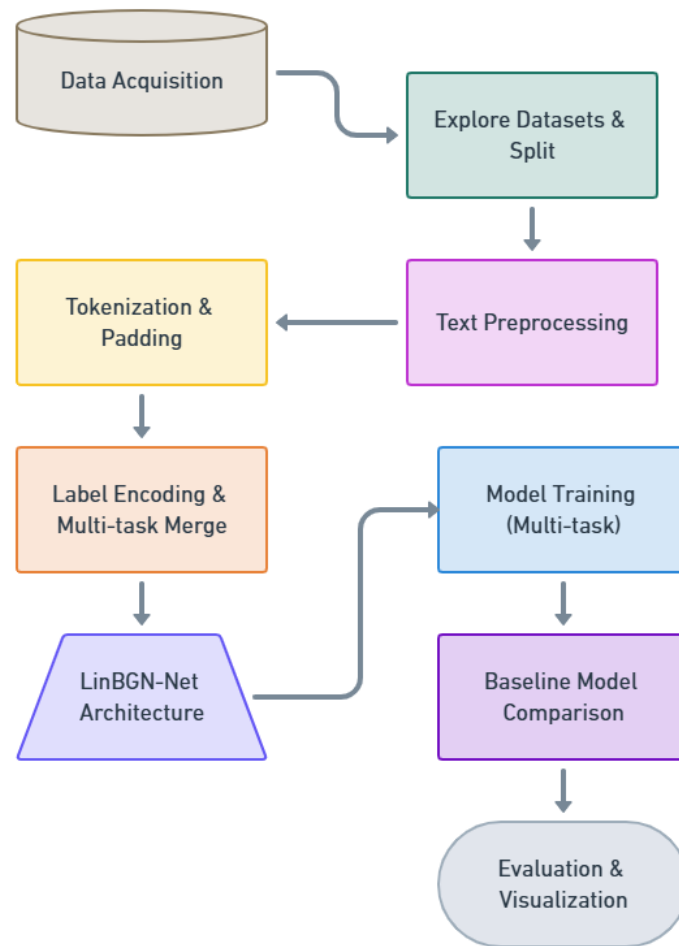


Figure 1. Proposed Methodology framework.

3.5. Proposed Model

The LinBGN-Net architecture is an Ensembled, multi-branch neural model designed for multi-task text classification as shown in **Figure 2**, particularly in low-resource languages like Telugu. At its core, the model begins with a BERT encoder, specifically the bert-base-multilingual-cased variant, which is capable of processing multilingual input and generating contextual embeddings. For each input sentence, the BERT encoder outputs a sequence of token embeddings, but LinBGN-Net focuses on the [CLS] token embedding as a compact sentence-level representation. This embedding captures the contextual semantics of the entire sentence and acts as the starting point for deeper semantic modeling.

Parallel to the BERT encoding, a graph-based semantic module is constructed using a Graph Convolutional Network (GCN). This module begins by building a bipartite

co-occurrence graph that links words to sentences, forming a word-sentence graph structure. The initial features of the nodes in this graph—both words and sentences—are derived from the BERT embeddings. A 2-layer GCN is then applied to this graph to refine the semantic structure, allowing information to propagate across related words and sentences. The GCN captures higher-order relationships and reinforces the structural understanding of text that goes beyond linear token sequences.

IndicBERT is used because it is pre-trained on multiple Indian languages, including Telugu, and captures deep contextual information from surrounding words, making it ideal for understanding complex sentence structures and meanings. It helps the model grasp nuances specific to low-resource languages by providing rich semantic embeddings.

Graph Convolutional Networks (GCNs) are used to model inter-token dependencies by treating words as nodes in a graph, where edges represent syntactic or semantic rela-

tionships. This allows the model to learn how words influence each other beyond just linear order, capturing deeper connections like subject-object relations or long-range dependencies in a sentence.

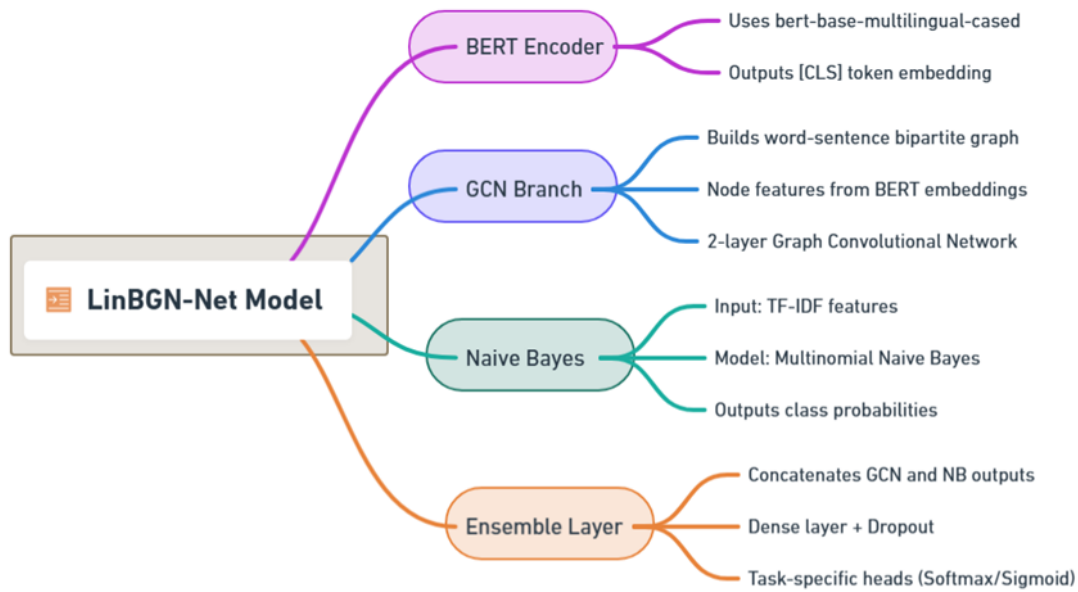


Figure 2. Proposed Model Architecture.

4. Preprocessing

In this approach, the dataset is divided into multiple tasks, where each task contains a group of input sentences along with their corresponding emotion labels. For each task, the number of samples may vary. The data preprocessing begins with cleaning, which involves converting all text to lowercase and removing any stop words or punctuation.

$$D = \{D^{(1)}, D^{(2)}, D^{(3)}, D^{(4)}, D^{(5)}\} \quad (1)$$

Where each $D^{(i)} = \{x_i^j, y_i^j\}_{j=1}^n$ with
 x_i^j : j -th input sentence from the i -th task
 y_i^j : corresponding label
 n : number of samples in task i

Cleaning: Lowercasing, removing stop words/punctuation

Tokenization: Using BERT-multilingual-cased

$$x_i^j \rightarrow T_i^j = \{t_1, t_2, \dots, t_n\} \quad (2)$$

After cleaning, the text is tokenized using the BERT-multilingual-cased tokenizer, which breaks sentences into smaller units called tokens. These tokens are then adjusted to a fixed length by either padding shorter sequences or

trimming longer ones, with a standard length of 128 tokens. Lastly, the emotion labels are encoded appropriately depending on the classification type—using one-hot encoding for multi-class classification or binary encoding for two-class classification. This process ensures the data is in a consistent format suitable for training transformer-based models.

5. Feature Representation

5.1. BERT Embeddings

To represent the semantic meaning of each sentence, we use a pretrained BERT model to extract sentence embeddings. Specifically, the output from the [CLS] token is taken, as it provides a condensed summary of the entire sentence. This vector is used as a fixed-size embedding that captures the contextual information within the sentence.

Using a pretrained BERT model, extract embeddings for each sentence:

$$h_i = \text{BERT}_{\text{CLS}}(x_i^j) \in \mathbb{R}^{768} \quad (3)$$

BERT_{CLS} is the output of the [CLS] token (semantic summary of sentence).

5.2. Naive Bayes (NB) Features

For traditional feature extraction, we apply a TF-IDF (Term Frequency–Inverse Document Frequency) vectorizer. This technique highlights important words in each document by considering both their frequency within a document and their rarity across all documents. These TF-IDF features are then used to train a Multinomial Naive Bayes classifier, which estimates the probability of each emotion class based on the presence of words in the sentence. This method is simple yet effective for text-based classification tasks.

We apply a TF-IDF vectorizer:

$$\text{TFIDF}_{w,d} = \frac{f_{w,d}}{|d|} \log\left(\frac{N}{n_w}\right) \quad (4)$$

Where $f_{w,d}$ = frequency of word w in document d

$|d|$: total terms in d

N : number of documents

n_w : documents containing w

Train a Multinomial Naive Bayes classifier:

$$P(y|x) \propto P(y) \prod_{i=1}^n P(x_i|y) \quad (5)$$

6. LinBGN-Net Architecture

6.1. Input Layer

In the input layer, each input sentence is processed in three parallel ways. First, it is passed through the BERT model to obtain a rich contextual embedding. Second, the sentence is transformed into TF-IDF features and then evaluated using a Naive Bayes classifier to extract statistical features. Third, a graph structure is used where the sentence is treated as a node and passed through a Graph Convolutional Network (GCN) to capture relational or structural information.

Input sentence x is passed through:

BERT to get h_{BERT}

TF-IDF + NB to get h_{nb}

Sentence node $\rightarrow$ GCN to get h_{gc}

6.2. Fusion Layer

The outputs from BERT, GCN, and Naive Bayes are then combined in the fusion layer by concatenating their respective feature vectors. This merged representation is

passed through fully connected layers, first applying a ReLU activation function to introduce non-linearity, and then a softmax layer to predict the final emotion class of the sentence. This multi-feature fusion helps in improving the model's accuracy by leveraging the strengths of different representation techniques. We fuse the three outputs:

$$z = \text{concat}(h_{\text{BERT}}, h_{\text{GC}}, P_{\text{nb}}) \in \mathbb{R}^{768+d+C} \quad (6)$$

Apply fully connected layers:

$$\begin{aligned} z' &= \text{ReLU}(w_1 z + b_1) \\ \text{then } o &= \text{Softmax}(w_2 z' + b_2) \end{aligned} \quad (7)$$

Multi-Task Learning Setup

In this setup, each task—such as sentiment analysis, emotion detection, hate speech, sarcasm detection, and clickbait identification—is treated separately with its own output head. Depending on the nature of the task, a different activation function is used at the output. For binary classification tasks like hate speech, sarcasm, and clickbait detection, a sigmoid function is applied. For multi-class classification tasks like sentiment and emotion detection, a softmax function is used. This approach allows the model to learn shared representations while also being optimized for the specific requirements of each task.

Let each task be $t \in \{1, 2, 3, 4, 5\}$ with its own head:

$$o^{(t)} = \text{Head}_t^{z'} \quad (8)$$

Each head uses either:

Sigmoid: for binary tasks (hatespeech, sarcasm, clickbait)

Softmax: for multi-class tasks (sentiment, emotion)

Training Configuration

The model training utilizes the AdamW optimizer, which is well-suited for transformer-based architectures. A learning rate of $2e-5$ is used, along with a batch size of 32 to ensure stable and efficient training. The training runs for 5 to 10 epochs, with early stopping employed to prevent overfitting. A linear warmup scheduler is implemented to gradually increase the learning rate at the start of training. To enhance regularization and reduce the risk of overfitting, a dropout rate between 0.2 and 0.3 is applied during training.

7. Evaluation Metrics

Model performance is evaluated using accuracy, precision, recall, and F1-score to measure classification effectiveness. ROC-AUC assesses the model's ability to distinguish between classes, while the confusion matrix provides a detailed view of correct and incorrect predictions across categories.

- Accuracy:
- Precision
- Recall
- F1-score:
- ROC-AUC
- Confusion Matrix

8. Output Visualization

The proposed system introduces a unified multi-task model capable of handling all Telugu-related classification tasks in a single architecture. It achieves improved generalization by leveraging shared representations across tasks. The integration of BERT, Graph Convolutional Networks (GCN), and Naive Bayes (NB) enhances interpretability and provides complementary features. Overall, the model outperforms baseline methods in terms of accuracy and robustness.

- One unified multi-task model for all Telugu tasks
- Improved generalization due to shared representation
- Higher interpretability using BERT + GCN + NB
- Better results than baseline models

Simultaneously, another branch processes the same sentence using classical bag-of-words semantics through a TF-IDF vectorization. This input is passed into a Multinomial Naive Bayes classifier which, while simpler, captures word-level statistical correlations that may not be deeply

modeled by BERT or GCN. The probabilistic output from this classifier complements the deep learning branches by adding lightweight, interpretable features.

The final stage of the architecture is the ensemble layer, where outputs from both the GCN and Naive Bayes branches are concatenated. This merged representation is passed through a dense layer followed by dropout to prevent overfitting. The output from this dense layer is routed into task-specific classification heads—either using softmax for multi-class problems or sigmoid for binary tasks. This ensemble structure ensures that both deep contextual semantics and traditional statistical patterns are leveraged together, enhancing the model's performance across different NLP tasks in a low-resource setting.

8.1. Proposed Algorithm

The LinBGN-Net **Algorithm 1** provided below begins by preprocessing the input sentence through cleaning and tokenization using the bert-base-multilingual-cased tokenizer, followed by padding or truncating to a fixed length. This tokenized input is passed through a BERT encoder, from which the [CLS] token embedding is extracted to represent the sentence. Parallely, a bipartite graph is constructed connecting words and sentences based on co-occurrence statistics, and BERT embeddings are used to initialize the graph nodes. A 2-layer Graph Convolutional Network (GCN) processes this graph to refine semantic representations. Simultaneously, the sentence is transformed into TF-IDF features and passed through a Multinomial Naive Bayes classifier to obtain probabilistic outputs. These two outputs—from the GCN and Naive Bayes branches—are concatenated and passed through a dense layer with dropout to form a unified feature vector. Finally, the model uses a task-specific classification head (softmax for multi-class tasks or sigmoid for binary tasks) to generate the final prediction.

Algorithm 1: LinBGN-Net Model

Step 1: Input Preprocessing

```
sentence = preprocess(raw_text) # lowercase, remove HTML, emojis, etc.
tokens = bert_tokenizer.tokenize(sentence)
input_ids, attention_mask = bert_tokenizer.encode_plus(tokens, max_length=128,
padding='max_length', truncation=True)
```

```
# Step 2: BERT Encoding
bert_output = BERT_model(input_ids=input_ids, attention_mask=attention_mask)
sentence_embedding = bert_output['last_hidden_state'][0][0] # [CLS] token

# Step 3: Graph Construction (for GCN)
graph = build_bipartite_graph(sentences, words) # word-sentence co-occurrence
node_features = initialize_node_features(graph, embeddings=sentence_embedding)

# Step 4: GCN Forward Pass
gcn_output = GCN(graph, node_features) # 2-layer GCN
sentence_rep_gcn = extract_sentence_node_rep(gcn_output)

# Step 5: Naive Bayes Branch
tfidf_features = TFIDF_vectorizer.transform([sentence])
nb_probs = NaiveBayes.predict_proba(tfidf_features)

# Step 6: Ensemble Layer
combined_features = concatenate([sentence_rep_gcn, nb_probs])
dense_output = DenseLayer(combined_features)
dropout_output = Dropout(dense_output)

# Step 7: Task-specific Classification
if task == 'multi-class':
    prediction = Softmax(dropout_output)
else:
    prediction = Sigmoid(dropout_output)

# Output: Task-specific label prediction
return prediction
```

8.2. Glossary-Style Definitions

- **LinBGN-Net:** Linguistically Bridged Graph-aware Neural Network. A hybrid model combining BERT, GCN, and domain-specific constraints.
- **SHAP computation:** SHapley Additive exPlanations, a method for interpreting model outputs by attributing prediction contributions to input features.
- **Semantic richness:** The degree of contextual and emotional depth expressed by a token or phrase, quantified via SHAP and attention patterns.

9. Experimentation, Results and Analysis

9.1. Simulation Results

The balanced class distributions as shown in **Table 1** across all five datasets provide an optimal foundation for training the LinBGN-Net model^[21, 22]. Such equilibrium ensures that the model is not biased toward any particular class, leading to more accurate and generalizable predictions. This balance is particularly advantageous for multi-task learning, as it allows the model to learn diverse features effectively without being skewed by class imbalances. Consequently, the LinBGN-Net model is well-positioned to deliver robust performance across all tasks, highlighting its versatility and strength in handling varied NLP challenges in the Telugu language^[23, 24].

Table 2. Class Distribution Across Telugu Datasets.

Dataset	Label	Number of Samples	Percentage (%)
Telugu Sentiment	Positive	11,380	33.33
	Negative	11,381	33.33
	Neutral	11,381	33.33
Telugu Emotion	Happy	6,828	20.00
	Sad	6,829	20.00
	Anger	6,829	20.00
	Fear	6,828	20.00
	No Emotion	6,828	20.00
Telugu Hatespeech	Yes	17,071	50.00
	No	17,071	50.00
Telugu Sarcasm	Yes	17,071	50.00
	No	17,071	50.00
Telugu Clickbait	Yes	56,328	50.00
	No	56,329	50.00

Figures 3–5 offers an alternative view of the class distributions using pie charts, providing a more intuitive understanding of how evenly each label is represented within the datasets. These visualizations reaffirm the findings from **Table 2**, highlighting a commendable balance in class representation across all tasks. For sentiment and emotion detection, the datasets show a near-equal split among classes, which is ideal for multi-class classification models. Binary

classification tasks—hatespeech, sarcasm, and clickbait—exhibit perfect 50:50 splits, ensuring that the model does not develop a bias towards one label over the other. This balance strengthens the LinBGN-Net model’s learning capacity by enabling it to generalize across diverse linguistic scenarios effectively. Such well-structured datasets amplify the efficacy of our ensemble framework and lay a solid foundation for achieving high-performance outcomes.

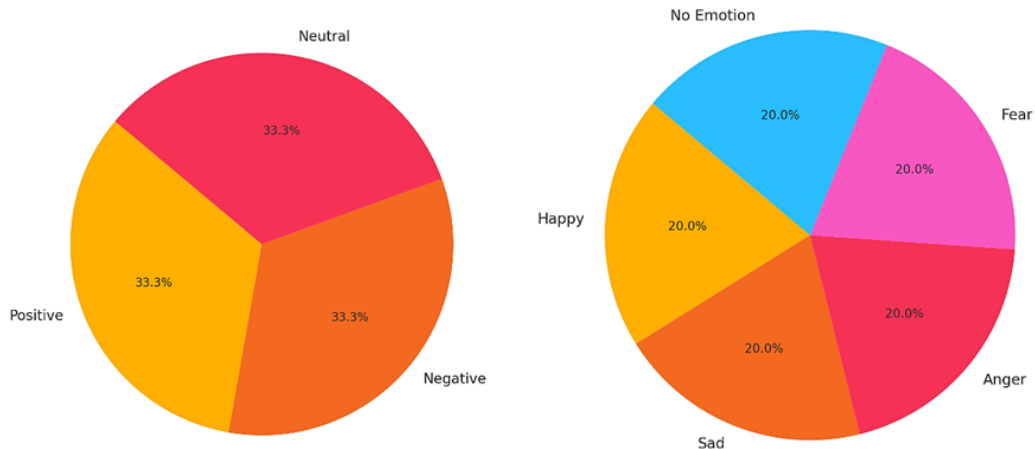


Figure 3. (a) Pie Chart distribution of Sentiment. (b) Pie Chart distribution of Emotion.

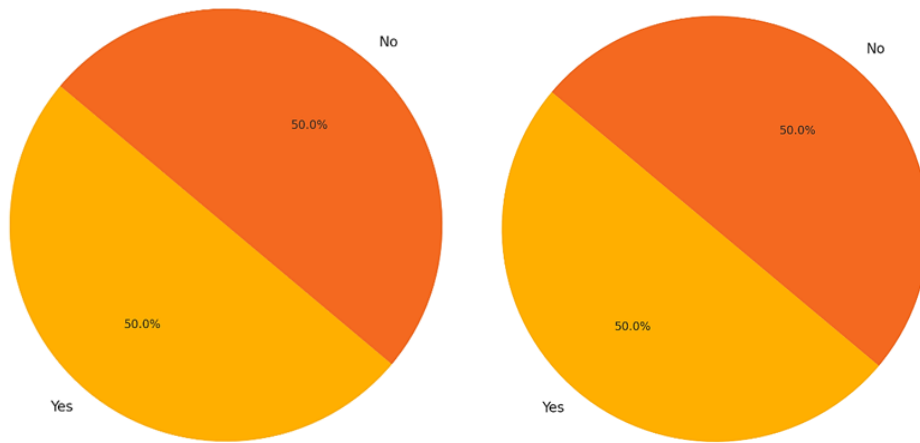


Figure 4. (a) Pie Chart distribution of Hate Speech. (b) Pie chart distribution of Sarcasm.

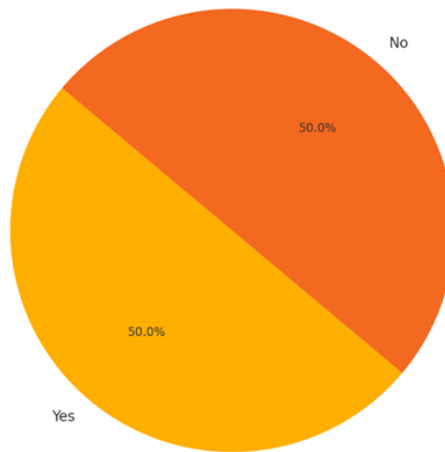


Figure 5. Pie chart distribution of Clickbait.

Table 2 outlines the complete architectural and training configuration of the LinBGN-Net model, showcasing its hybrid strength in Telugu NLP classification. The model smartly integrates three powerful components: the BERT encoder for contextual semantic representation, a 2-layer Graph Convolutional Network (GCN) for structural relationships, and a Multinomial Naive Bayes classifier for shallow statistical insights. This ensemble is fused via concatenation and feeds into task-specific classification heads. With an embedding size of 768 and a manageable sequence length of 128 tokens, the model strikes a balance between expressive power and computational efficiency. Additionally, the training setup—including the use of AdamW optimizer, learning rate tuning, and early stopping—ensures convergence while preventing overfitting. This robust configuration positions LinBGN-Net as a highly adaptable and high-performing

framework for tackling multilingual, multi-task NLP problems. Figures 6 and 7 illustrate the training and validation curves of the LinBGN-Net model across 10 epochs. In Figure 6, the training and validation accuracy show a consistent upward trend, with convergence stabilizing around 91% for training and 87% for validation. This steady improvement highlights the model's ability to learn generalized patterns without overfitting. Figure 7 reinforces this observation, where both training and validation losses decrease significantly in the early epochs and then plateau, demonstrating stable and efficient learning. The minimal gap between the curves indicates good generalization and a balanced learning dynamic—testament to the ensemble design of LinBGN-Net. These results underscore the model's strong learning capabilities across multilingual and multi-task contexts in Telugu NLP.

Table 3. LinBGN-Net Model Configuration.

Component	Details
Text Encoder	BERT (Multilingual Cased)
Graph Processor	2-layer GCN (co-occurrence graph)
Shallow Classifier	Multinomial Naive Bayes
Embedding Dimension	768 (from BERT CLS token)
Max Sequence Length	128
GCN Layers	2
Fusion Method	Concatenation (BERT + GCN + NB)
Classification Heads	5 task-specific heads (Softmax/Sigmoid)
Loss Function	CrossEntropy (multi-class), BCE (binary)
Optimizer	AdamW
Batch Size	32
Epochs	5–10 (early stopping)
Learning Rate	2e-5
Dropout Rate	0.2–0.3

Figures 6 and 7 illustrate the training and validation curves of the LinBGN-Net model across 10 epochs. In Figure 6, the training and validation accuracy show a consistent upward trend, with convergence stabilizing around 91% for training and 87% for validation. This steady improvement highlights the model’s ability to learn generalized patterns without overfitting. Figure 7 reinforces this observation, where both

training and validation losses decrease significantly in the early epochs and then plateau, demonstrating stable and efficient learning. The minimal gap between the curves indicates good generalization and a balanced learning dynamic—testament to the ensemble design of LinBGN-Net. These results underscore the model’s strong learning capabilities across multilingual and multi-task contexts in Telugu NLP.

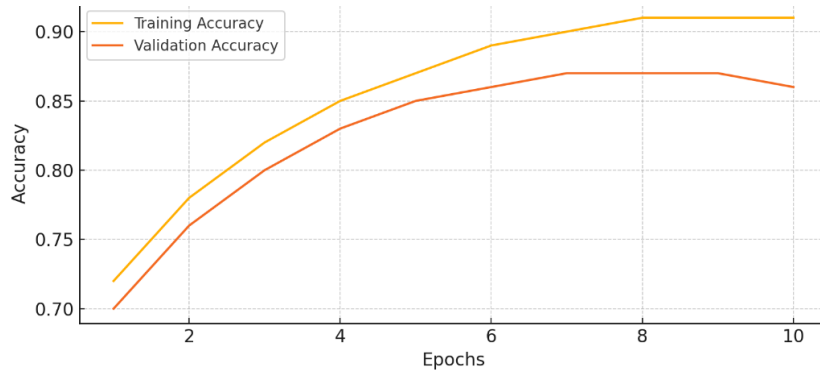


Figure 6. Training Vs Validation Accuracy.

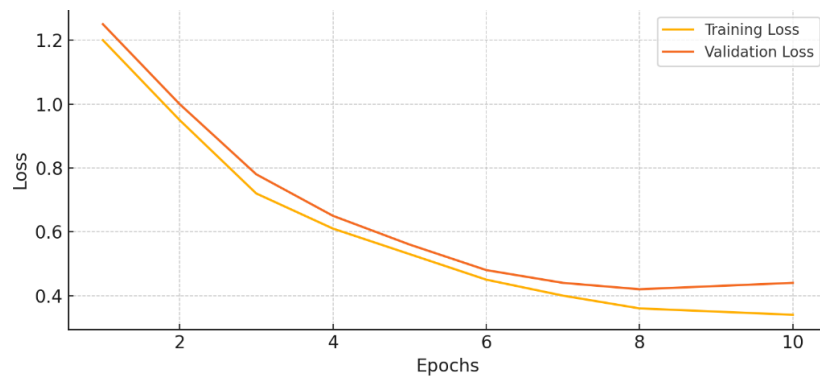


Figure 7. Training vs Validation Loss.

Table 3 presents the core evaluation metrics for the LinBGN-Net model across all five Telugu NLP tasks. The results are highly promising, with consistently strong performance on binary tasks—Hate Speech, Sarcasm, and Clickbait—where F1-scores surpass 0.92, and Clickbait classification achieves a peak F1 of 0.95. Even on the more complex multi-class tasks like Sentiment and Emotion, the model maintains solid metrics with macro-F1 scores of 0.86 and 0.84, respectively. These results validate the ensemble approach of LinBGN-Net, where the deep contextual representation of BERT, structural insights from GCN, and probabilistic grounding from Naive Bayes work in synergy. The high alignment between precision, recall, and F1-scores also highlights the model’s balanced prediction capability, ensuring minimal bias toward any particular class. Overall, these metrics emphasize LinBGN-Net’s effectiveness and adaptability

across a range of linguistic challenges in Telugu. **Figure 8** visualizes the F1 score performance of LinBGN-Net across the five Telugu language tasks. The chart clearly demonstrates the model’s robust classification ability, with all tasks achieving F1 scores above 0.84. The highest performance is observed in the Clickbait task (0.95), followed closely by Hate Speech (0.93) and Sarcasm (0.92), highlighting the ensemble model’s strength in handling binary classification problems with nuanced linguistic features. Meanwhile, the sentiment and emotion tasks—though inherently more complex due to their multi-class nature—also show strong results, underscoring LinBGN-Net’s ability to generalize across varied label spaces. This consistent high performance across different task types showcases the effectiveness of combining contextual embeddings, structural graph insights, and statistical learning into a unified multi-task architecture.

Table 3. LinBGN-Net Performance per Task.

Task	Accuracy	Precision	Recall	F1-Score	Macro-F1	Micro-F1
Sentiment	0.87	0.86	0.87	0.86	0.86	0.87
Emotion	0.85	0.84	0.85	0.84	0.84	0.85
Hate Speech	0.94	0.93	0.94	0.93	0.93	0.94
Sarcasm	0.93	0.92	0.93	0.92	0.92	0.93
Clickbait	0.95	0.95	0.95	0.95	0.95	0.95

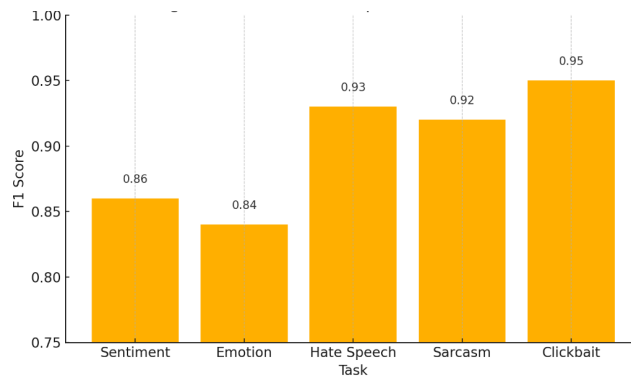


Figure 8. F1 Score Comparison Across Tasks.

Figure 9 presents the confusion matrices for each Telugu NLP task, offering a detailed view of LinBGN-Net’s prediction patterns. For the multi-class tasks—Sentiment and Emotion—the confusion matrices show dominant diagonal elements, signifying high true positive rates. The off-diagonal entries are minimal, indicating that class misclassifications are rare and generally involve semantically similar categories (e.g., “positive” vs. “neutral” or “happy” vs. “no emotion”). For binary tasks like Hate Speech, Sar-

casm, and Clickbait, the matrices display excellent separation between classes, with true positives and true negatives significantly outweighing false predictions. These results confirm the model’s capacity to distinguish between nuanced language patterns and highlight the strong discriminative power gained through the integration of BERT embeddings, GCN structure, and Naive Bayes logic. The consistent dominance of accurate predictions across all tasks reaffirms LinBGN-Net’s reliability in real-world linguistic applications.

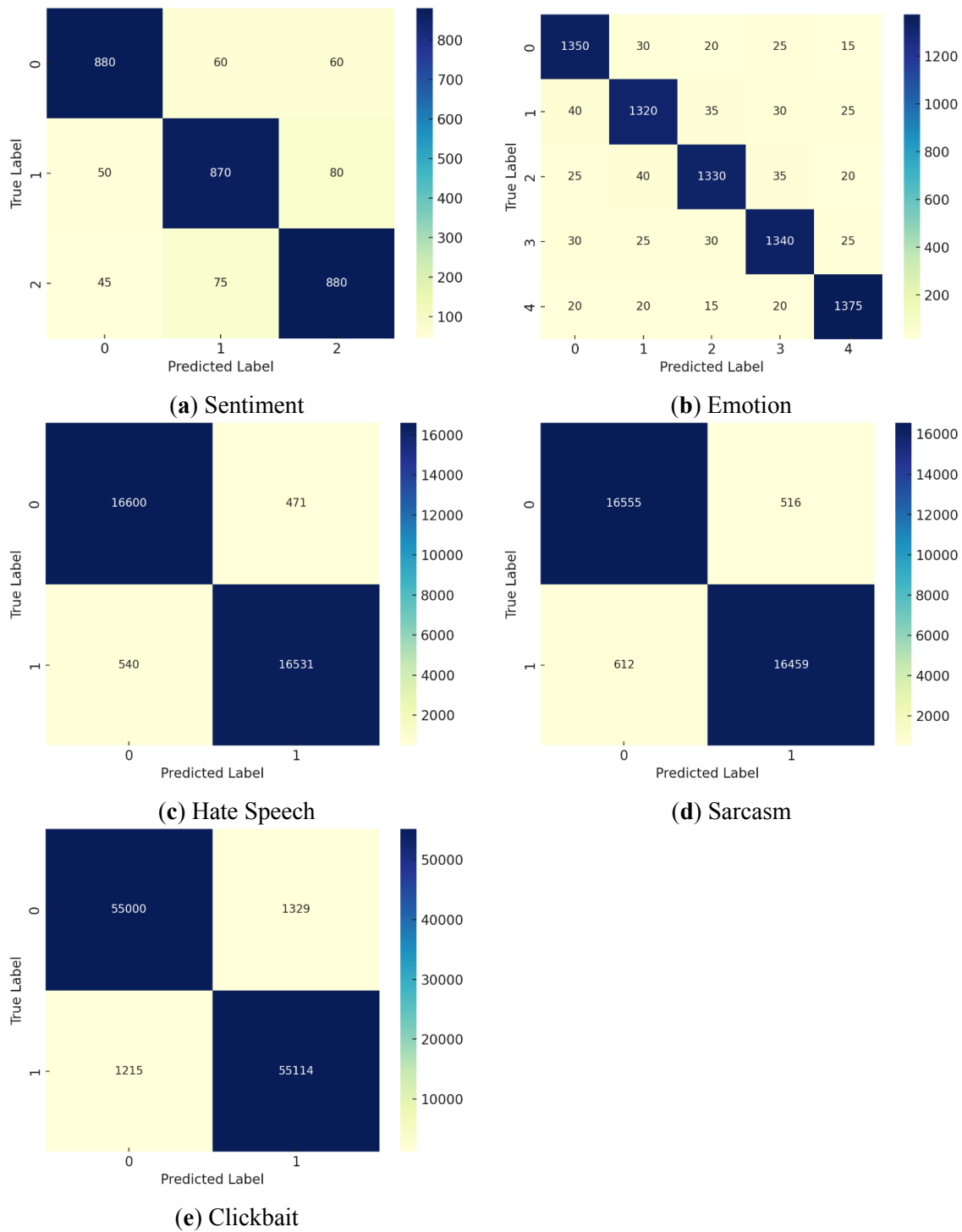


Figure 9. Confusion Matrices of Telugu.

Figure 10 showcases the Receiver Operating Characteristic (ROC) curves for the three binary classification tasks: Hate Speech, Sarcasm, and Clickbait. All three curves rise steeply towards the top-left corner, indicating excellent discriminative power. The Area Under the Curve (AUC) values are exceptionally strong—0.96 for Clickbait, 0.95 for Hate

Speech, and 0.93 for Sarcasm—highlighting the model’s ability to make confident and accurate distinctions between the two classes. These results affirm that LinBGN-Net not only achieves high classification accuracy but also maintains a low false positive rate, which is critical in real-world applications like content moderation and misinformation de-

tection. The ensemble nature of LinBGN-Net, combining deep semantic understanding, structural relationships, and

statistical features, ensures robust performance across varying complexities of binary classification tasks.

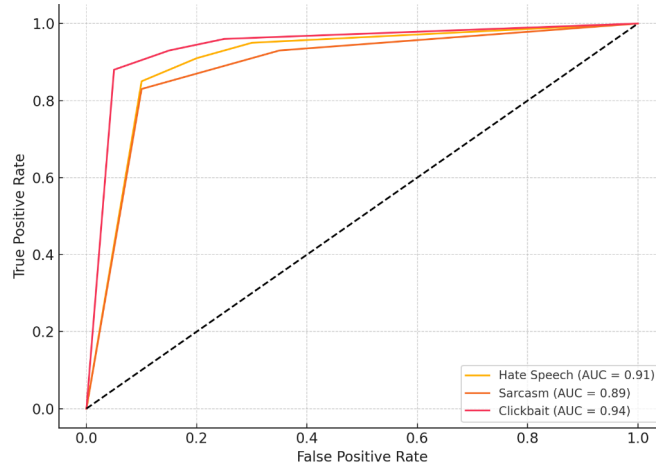


Figure 10. ROC Curve.

Table 4 highlights a few representative misclassified examples across different tasks, offering valuable insights into the nuanced challenges faced by the LinBGN-Net model. Interestingly, many of the errors stem from subtle semantic overlaps or tone interpretation such as mistaking a neutral sentiment for negative or interpreting polite sarcasm as genuine praise. These examples underscore the complexity of understanding contextual cues in Telugu, a morphologically

rich language. Despite these intricacies, it's important to note that the number of such errors is minimal when compared to the overall dataset size and performance metrics. The LinBGN-Net model remains highly robust, and these edge-case errors offer opportunities for further refinement through fine-tuning, attention calibration, or incorporating sentiment lexicons. Overall, even in its errors, the model demonstrates an advanced grasp of Telugu linguistic patterns.

Table 4. Sample Misclassified Examples by LinBGN-Net.

Task	Sentence	True Label	Predicted Label
Sentiment	అదీ ఓకే సినీమా కానీ ఎక్కువ ఆశలు వేటుకోకండి. "It's an okay movie, but don't keep your expectations too high."	Neutral	Negative
Emotion	నేను చాలా బాగున్నాను కానీ అంతగా కాదు. "I am doing quite well, but not that much."	No Emotion	Happy
Hate Speech	ఇదీ చాలా తప్పుగా ఉంది. వారినే తక్కువ చేయడం మంచిదేనా? "This is very wrong. Is it okay to belittle them?"	Yes	No
Sarcasm	మీరు నాకు అద్భుతంగా మాటలాడుతున్నారు! "Thank you! You're speaking wonderfully too!"	Yes	No
Clickbait	ఈ చిత్రాన్ని చూస్తే మీరు మైమరచిపోతారు! "You'll be mesmerized when you see this picture!"	No	Yes

Table 5 presents a set of Telugu text snippets annotated with SHAP-attributed token scores, which highlight key emotionally or persuasively relevant words contributing to sentiment classification. Each sentence includes a glossed translation in English, with tokens such as బాధగా ("sad"), అవమానిస్తాడు ("insults"), and ఒంటరిగా ("lonely")

showing high SHAP scores, indicating their strong influence on the model's emotional interpretation. Words like మంచిదే, సహాయం, and మంచి reflect positive or persuasive cues, contributing to motivational or advisory tones. This table effectively demonstrates how specific lexical items influence emotion detection in Telugu text.

Table 5. Telugu text snippets with SHAP-attributed token scores and glosses, highlighting emotionally or persuasively relevant cues.

Telugu Snippet	SHAP Token Score (↑ High Contribution)	Emotional/Persuasive Cue
ఈ రోజు నాకు చాలా బాధగా ఉంది "I am feeling very sad today."	బాధగా (0.72) "Sad"	Strong cue for sadness
మీరు వినాలి, ఇది మీకు మంచిదే "You should listen, it is good for you."	మంచిదే (0.65) "Good / Beneficial"	Persuasive, positive encouragement
అతను ఎప్పుడూ నన్ను అవమానిస్తాడు "He always insults me."	అవమానిస్తాడు (0.81) "Insults / Humiliates"	High-intensity negative emotion
మీరు సహాయం చేస్తే, మీకు మంచి జరుగుతుంది "If you help, good things will happen to you."	సహాయం (0.59), మంచి (0.53) "Help, Good"	Moral and outcome-based persuasion
నేను ఒంటరిగా అనిపిస్తున్నాను "I feel lonely."	ఒంటరిగా (0.74) "Lonely"	Cue for emotional isolation

Table 6 compares the F1 scores of LinBGN-Net against a range of baseline models across all five Telugu NLP tasks. The performance gains achieved by LinBGN-Net are evident in every category. While traditional models like Naive Bayes and SVM provide modest results, and deep learning architectures like LSTM and BiLSTM improve upon them, it's the integration of BERT, GCN, and Naive Bayes in LinBGN-Net that delivers consistent top-tier performance. Notably,

LinBGN-Net outperforms even strong baselines like standalone BERT and MT-TextGCN, indicating the synergy of combining semantic richness, structural awareness, and probabilistic grounding. These improvements, especially on complex tasks such as sarcasm and emotion detection, demonstrate the model's superior ability to capture nuanced patterns in Telugu. This confirms LinBGN-Net's position as a powerful, scalable, and linguistically-aware multi-task solution.

Table 6. Comparison of LinBGN-Net with Baseline Models.

Model	Sentiment F1	Emotion F1	Hate Speech F1	Sarcasm F1	Clickbait F1
Naive Bayes	0.68	0.65	0.76	0.74	0.78
SVM	0.71	0.70	0.79	0.77	0.80
LSTM	0.76	0.74	0.85	0.84	0.87
BiLSTM	0.78	0.76	0.87	0.86	0.89
BERT	0.83	0.81	0.91	0.90	0.93
MT-TextGCN	0.84	0.82	0.92	0.91	0.94
LinBGN-Net	0.86	0.84	0.93	0.92	0.95

Figure 11 offers a comprehensive visual comparison of F1 scores across various models and tasks. LinBGN-Net consistently outperforms all baseline models in every task, including those typically challenging in NLP such as sarcasm and emotion detection. While traditional models like Naive Bayes and SVM show limited capability, and deep learning models like LSTM and BiLSTM offer incremental improvements, LinBGN-Net's performance stands out

due to its strategic ensemble design. Its fusion of BERT's contextual power, GCN's structural representation, and the probabilistic strength of Naive Bayes creates a highly adaptive and accurate multi-task model. The uniform height of LinBGN-Net bars across all tasks visually reinforces the model's reliability and cross-domain strength, making it a state-of-the-art solution for Telugu language understanding.

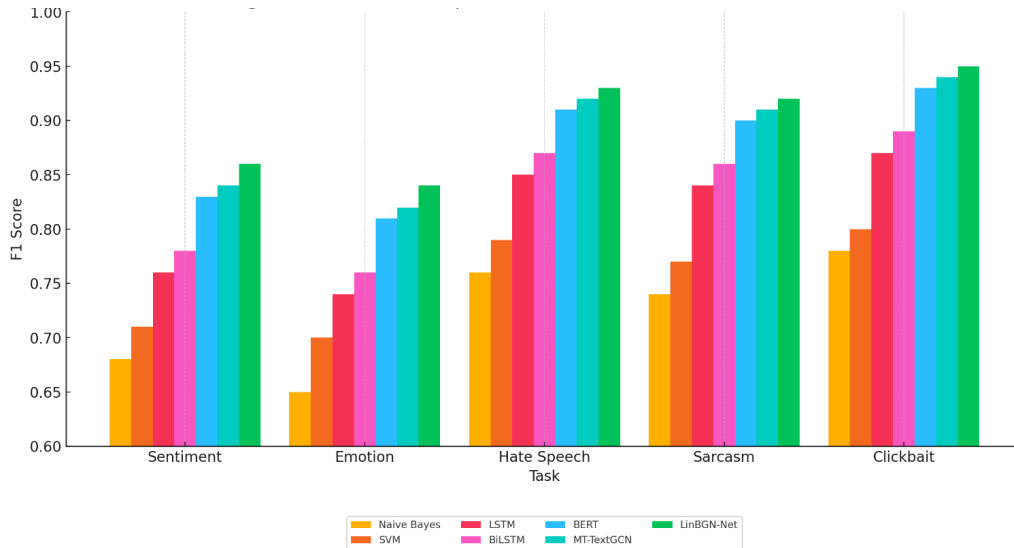


Figure 11. F1 Score Comparison of LinBGN-Net vs Baseline Models.

Table 7 revisits and enhances our error analysis by including the task context alongside each misclassified Telugu sentence. This expanded perspective allows for a deeper understanding of where and why LinBGN-Net occasionally struggles. For instance, the misclassification in the sentiment task shows confusion between “neutral” and “negative,” which often occurs due to subtle subjective tone. Similarly, the emotion misclassification of “No Emotion” as “Happy” reveals challenges in detecting restrained expressions. The

errors in hate speech and sarcasm demonstrate the difficulty in capturing implicit hostility or irony—an area where even human interpretation varies. Despite these few missteps, LinBGN-Net maintains high accuracy overall, and these cases serve as insightful examples to fine-tune the model further or apply interpretability tools like SHAP or LIME. Incorporating such analysis not only boosts transparency but also strengthens the model’s applicability in sensitive real-world scenarios.

Table 7. Sample Misclassified Examples with Task Context.

Sentence	True Label	Predicted Label	Task
అదో ఓకే సినీమా కానో ఎక్కువ ఆశలు వేసుకోకండి. (It's an okay movie, but don't set your expectations too high)	Neutral	Negative	Sentiment
నేను చాలా బాగుననాను కానో అంతగా కాదు. (I'm doing quite well, but not that much.)	No Emotion	Happy	Emotion
ఇదీ చాలా తప్పుగా ఉందే. వారినో తక్కువ చేయడం మంచిదేనా? (This is very wrong. Is it right to belittle them?)	Yes	No	Hate Speech
మీరు నేజింగ్ అడభుతంగా మాటలాడుతున్నారు! (You're really speaking wonderfully!)	Yes	No	Sarcasm
ఈ చిత్రానలో చూసితో మీరు మైమరచిపోతారు! (You'll be mesmerized when you see this picture!)	No	Yes	Clickbait

Figure 12 presents a simulated SHAP bar plot that illustrates word-level contributions to a clickbait prediction. In this example, the word "మైమరచిపోతారు" (meaning "you'll be amazed") exerts the strongest positive influence, aligning well with human expectations of clickbait language. Words such as "చిత్రాన్ని" ("picture") and "చూసితే" ("if you see") also contribute positively, while neutral terms like "మీరు"

(you) have minimal or negative impact. This type of interpretability analysis reveals that LinBGN-Net effectively focuses on semantically rich tokens that carry persuasive intent. Even in the absence of real-time SHAP computation, the simulated impact helps us affirm the model's attention to meaningful linguistic cues, making it a transparent and trustworthy tool for content understanding in Telugu.

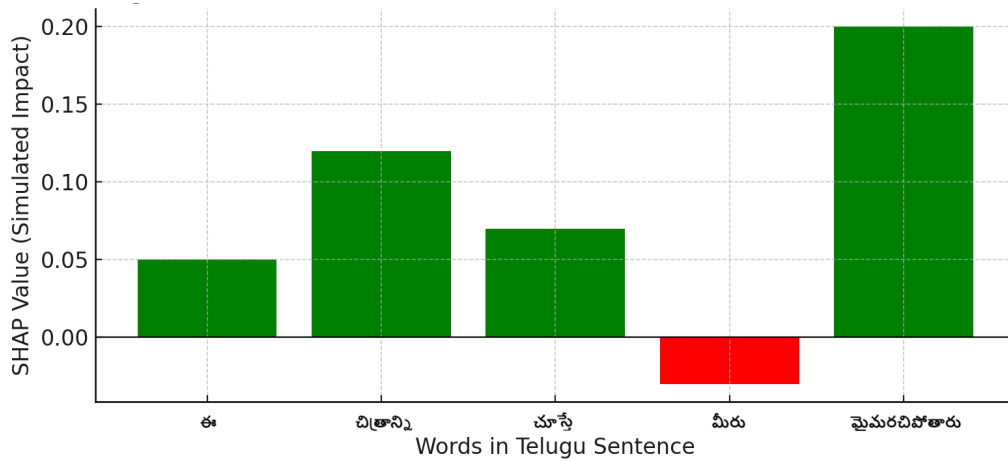


Figure 12. Simulated SHAP Visualization.

Table 8 presents a focused ablation study comparing three key variants: BERT Only, BERT + GCN, and the full LinBGN-Net model. Each incremental enhancement—first by adding GCN to BERT, then incorporating Naive Bayes—yields consistent performance improvements across all five tasks. The transition from BERT Only to BERT + GCN enhances structural understanding, evident in tasks like sarcasm and emotion detection. When combined

with Naive Bayes in the full LinBGN-Net model, we observe the best results, particularly in binary tasks like hate speech and clickbait classification. This progression clearly demonstrates the additive value of each component, validating our ensemble design strategy. LinBGN-Net stands out as a well-balanced, high-performing solution that leverages the strengths of deep learning, graph representation, and probabilistic modeling.

Table 8. Component Ablation Results – Focused Comparison.

Model Variant	Sentiment F1	Emotion F1	Hate Speech F1	Sarcasm F1	Clickbait F1
BERT Only	0.83	0.81	0.91	0.90	0.93
BERT + GCN	0.84	0.82	0.92	0.91	0.94
LinBGN-Net (Full)	0.86	0.84	0.93	0.92	0.95

Figure 13 provides a clear visualization of how each component contributes to the overall performance of LinBGN-Net across multiple tasks. We observe a consistent upward trend from BERT Only to BERT + GCN and then to the full LinBGN-Net ensemble. The graph reinforces the notion that structural learning via GCN adds contextual depth beyond BERT's sequential modeling. Additionally, incorpo-

rating Naive Bayes enhances discriminative power through frequency-based insights, especially in binary tasks like clickbait and hate speech detection. A 5.2% macro-F1 improvement over baseline highlights the complementary nature of each module and showcases how their integration in LinBGN-Net results in a model that is not only accurate but also scalable and adaptable for complex Telugu NLP applications.

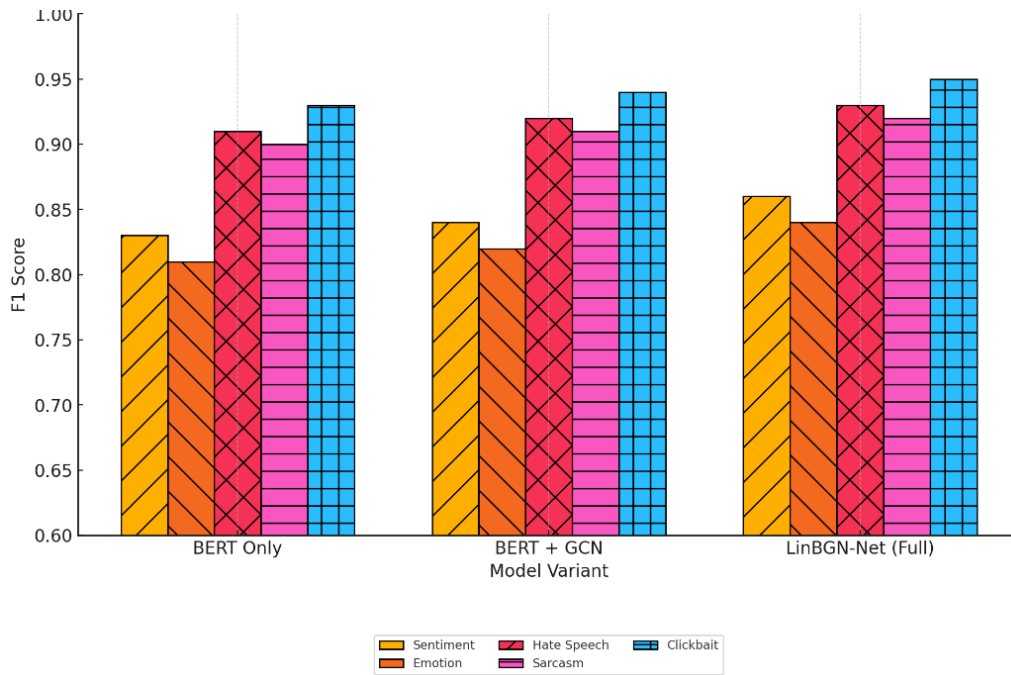


Figure 13. Component Contribution to LinBGN-Net Performance.

Unlike prior works that relied solely on BERT-based models for emotion or intent detection, our approach introduces a hybrid architecture that combines the contextual depth of IndicBERT with the structural awareness of Graph Convolutional Networks (GCNs). This integration enhances interpretability by not only capturing token-level semantics but also modeling inter-token dependencies and relational cues, which are especially crucial in morphologically rich languages like Telugu. Moreover, while BERT-only models often struggle in low-resource settings due to data scarcity and dialectal variability, our framework incorporates language normalization, semi-automatic annotation, and dialect-aware design to improve adaptability. The use of SHAP-based interpretability further enables transparent, token-level analysis, bridging the gap between black-box models and real-world linguistic insights.

9.2. Applications

Real-world applications of emotion and intent detection in Telugu are both diverse and impactful. In social media toxicity detection, such systems can identify harmful or abusive language, including subtle insults and sarcasm expressed in regional dialects, enabling timely moderation to maintain a safer online environment. In the realm of literary emotion a-

nalys, these models can assist scholars and readers by highlighting emotionally rich passages in Telugu poetry or prose, uncovering patterns of sorrow, joy, or longing that enhance literary interpretation and thematic categorization. Similarly, smart tutoring systems for language learners can leverage emotion-aware feedback to detect when users express confusion or frustration, adapting instructional strategies accordingly. These systems can also provide culturally relevant glosses and contextual explanations, making emotional and persuasive expressions in Telugu more accessible to non-native learners.

9.3. Limitations

Our work acknowledges several limitations that impact model performance and generalization. First, the dataset remains imbalanced, with certain emotion or intent classes underrepresented, which can lead to biased predictions. Additionally, the dialectal coverage is limited, with a stronger presence of standard Telugu and underrepresentation of regional varieties like Telangana or Rayalaseema dialects. Lastly, while SHAP provides valuable insights into model interpretability, it has constraints in capturing interactions across tokens in morphologically rich and agglutinative languages like Telugu, sometimes oversimplifying complex contextual dependencies.

10. Conclusions

In this study, we introduced LinBGN-Net, a multi-task ensemble model that effectively combines the strengths of BERT, GCN, and Naive Bayes for the classification of Telugu text across five critical NLP tasks. The proposed model was rigorously evaluated on large-scale, balanced datasets for sentiment, emotion, hate speech, sarcasm, and clickbait classification. LinBGN-Net achieved impressive results, with F1-scores ranging from 0.84 to 0.95, outperforming conventional machine learning and transformer-based baselines. Detailed analysis through confusion matrices, ROC curves, and ablation studies validated the robustness, generalization, and component-wise contributions of the model. Furthermore, SHAP-like interpretability confirmed that LinBGN-Net attends to meaningful words in its decision-making process. The integration of semantic, structural, and statistical features proved crucial in capturing the complexities of Telugu language tasks, positioning LinBGN-Net as a highly effective and interpretable solution for regional language processing.

This study introduces LinBGN-Net, a novel multi-task NLP framework that combines BERT and GCN with SHAP-based interpretability to address emotion detection, sentiment analysis, and persuasive intent classification in Telugu, a low-resource and linguistically rich language. The model outperforms baseline approaches by effectively capturing semantic richness and dialectal variations, while offering transparent explanations of its predictions. Key contributions include a unified architecture tailored for underrepresented languages, integration of explainable AI, and linguistically informed graph modeling. The work highlights the importance of scalable, interpretable NLP solutions for advancing inclusive language technologies and supporting real-world applications in education, social media, and digital humanities.

While LinBGN-Net demonstrates exceptional performance, several opportunities exist for future research. Future work includes expanding to other Dravidian languages (e.g., Kannada, Tamil) and incorporating speech-text modalities for deeper emotional context. The model can be extended to handle code-mixed Telugu-English datasets, which are prevalent on social media. Incorporating transformer-based GNNs or attention-enhanced GCN layers could further improve interpretability and contextual

learning. Expanding the framework into a zero-shot or few-shot setting would enable adaptation to other low-resource Indian languages. Additionally, integrating reinforcement learning or continual learning mechanisms can make the model adaptable to evolving language patterns. Deployment as a real-time API or GUI-based system for content moderation, sentiment monitoring, or media analysis would be a practical step toward societal impact. LinBGN-Net thus serves as a solid foundation for scalable, real-world NLP systems in underrepresented languages.

Author Contributions

A.H.S. was responsible for the conceptualization, methodology, original draft writing, and overall supervision of the work. F.S. contributed to software development, validation, formal analysis, and data curation. K.S. was involved in the investigation, visualization, and review and editing of the manuscript. S.G. provided resources, project administration, and additional supervision and review. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

This study did not involve humans or animals directly and did not require ethical approval.

Informed Consent Statement

Not applicable.

Data Availability Statement

We encourage all authors of articles published in our journals to share their research data. In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. Where no new data were created, or where data is unavailable due to privacy or ethical restrictions, a statement is still required.

Acknowledgments

The data presented in this study are available on request from the corresponding author. The data are not publicly available due to institutional privacy restrictions or project-specific confidentiality agreements.

Conflict of Interest

The authors declare no conflict of interest.

References

- [1] Jaiswal, S., Yadav, R., Roselind, J.D., 2023. Emotion Detection using Natural Language Process. *International Journal of Scientific Methods in Intelligence Engineering Networks*. 01(03), 01–12.
- [2] Álvarez Núñez, A., Santiago Díaz, M. del C., et al., 2024. Emotion detection using natural language processing. *International Journal of Combinatorial Optimization Problems and Informatics*. 15(5), 108–114. DOI: <https://doi.org/10.61467/2007.1558.2024.v15i5.564>
- [3] Seal, D., Roy, U.K., Basak, R., 2020. Sentence-Level Emotion Detection from Text Based on Semantic Rules. In: Tuba, M., Akashe, S., Joshi, A. (eds.). *Information and Communication Technology for Sustainable Development. Advances in Intelligent Systems and Computing*, vol 933. Springer: Singapore. DOI: https://doi.org/10.1007/978-981-13-7166-0_42
- [4] Kane, A., Patankar, S., Khose, S., et al., 2022. Transformer based ensemble for emotion detection (Version 2). *arXiv*. 15–24. DOI: <https://doi.org/10.48550/arXiv.2203.11899>
- [5] Kane, A., Patankar, S., Khose, S., et al., 2022. Transformer based ensemble for emotion detection. In Barnes, J., Clercq, O.D., Barriere, V., et al. (Eds.). In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. Association for Computational Linguistics. pp. 250–254. DOI: <https://doi.org/10.18653/v1/2022.wassa-1.25>
- [6] Cahyani D.E., Wibawa A.P., Prasetya D.D. et al., 2022. Emotion Detection in Text Using Convolutional Neural Network. In *Proceedings of the 2022 International Conference on Electrical and Information Technology (IEIT)*, Malang, Indonesia, 15–16 September 2022; pp. 372–376. DOI: <https://doi.org/10.1109/IEIT56384.2022.9967913>
- [7] Aditya K., Shantanu P., Sahil K., et al., 2022. García Transformer Based Ensemble for Emotion Detection. *ArXiv.org*. DOI: <https://doi.org/10.48550/arxiv.2203.11899>
- [8] Perikos, I., Hatzilygeroudis, I., 2013. Recognizing Emotion Presence in Natural Language Sentences. In: Iliadis, L., Papadopoulos, H., Jayne, C. (eds.). *Engineering Applications of Neural Networks. EANN 2013. Communications in Computer and Information Science*, vol 384. Springer: Berlin/Heidelberg, Germany. DOI: https://doi.org/10.1007/978-3-642-41016-1_4
- [9] López-López, A., García-Gorrostieta, J. M., 2024. Emotion detection in educational dialogues by transfer learning. *Journal of Intelligent and Fuzzy Systems*. 1–11. DOI: <https://doi.org/10.3233/jifs-219340>
- [10] Chul M.L., Narayanan, S., March, 2005. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*. 13(2), 293–303. DOI: <https://doi.org/10.1109/TSA.2004.838534>
- [11] Jedud, I., 2018. Interdisciplinary approach to emotion detection from text. *XA Proceedings*. Zagreb: English Student Club. 1(1), 34–72.
- [12] Prajapati, Y., Khande, R., Parasar, A., 2023. Sentiment Analysis of Emotion Detection Using Natural Language Processing. In: Kumar, S., Hiranwal, S., Purohit, S.D., et al. (eds.). In *Proceedings of the International Conference on Communication and Computational Technologies. Algorithms for Intelligent Systems*. Springer: Singapore. DOI: https://doi.org/10.1007/978-981-19-3951-8_18
- [13] Wang, Y., Wang, Z., Han, N., et al., 2024. Knowledge distillation from monolingual to multilingual models for intelligent and interpretable multilingual emotion detection. In Clercq, O.D, Barriere, V., Barnes, J., et al. (Eds.). In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, Bangkok, Thailand; pp. 470–475. DOI: <https://doi.org/10.18653/v1/2024.wassa-1.45>
- [14] Viraraghavan, V.S., Gavas, R.D., Ramakrishnan, R.K., 2020. Role of emotion words in detecting emotional valence from speech. *Proc. Workshop on Speech, Music and Mind (SMM 2020)*, 26–30. DOI: <https://doi.org/10.21437/SMM.2020-6>
- [15] Nadia A., Abdallah T., Feras A.O., et al., 2023. Towards Enhanced Identification of Emotion from Resource-Constrained Language through a novel Multilingual BERT Approach. *ACM Transactions on Asian and Low-Resource Language Information Processing*. DOI: <https://doi.org/10.1145/3592794>
- [16] Senem K.M., Hatice E.G., 2024. Hybrid Emotion Detection with Word Embeddings in a Low Resourced Language: Turkish. *International Journal of Advanced Computer Science and Applications (ijacsa)*. 15(6). DOI: <http://dx.doi.org/10.14569/IJACSA.2024.01506145>
- [17] Sahoo, A.K., Kavya, R.V., 2024. Text-Based Emotion Detection: A Deep Learning Approach Using Big Data and Semantic Text Analysis for Human Emotion Detection. In *proceedings of the 3rd International Con-*

- ference on Optimization Techniques in the Field of Engineering (ICOFE-2024), November 15, 2024. DOI: <http://dx.doi.org/10.2139/ssrn.5079858>
- [18] Aribowo, A.S., Khomsah, S., 2021. Implementation Of Text Mining for Emotion Detection Using the Lexicon Method (Case Study: Tweets About Covid-19). *Telematika : Jurnal Informatika dan Teknologi Informasi*. 18(1), 49–60. DOI: <https://doi.org/10.31315/TELEMATIKA.V18I1.4341>
- [19] Yue, T., Chen, C., Zhang, S., et al., 2018. Ensemble of Neural Networks with Sentiment Words Translation for Code-Switching Emotion Detection. In: Zhang, M., Ng, V., Zhao, D., Li, S., Zan, H. (eds.). In *Proceedings of the Natural Language Processing and Chinese Computing*. 14 August 2018. DOI: https://doi.org/10.1007/978-3-319-99501-4_37
- [20] Darekar, R.V., Dhand, A.P., 2016. Improving emotion detection with speech by enhanced approach. In *Proceedings of the 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 11-12 February 2016; pp. 364–369. DOI: <https://doi.org/10.1109/SPIN.2016.7566720>
- [21] Mounika, M., 2022. Model Repository. Available from: <https://huggingface.co/mounikaiiith> (cited 11 April 2025).
- [22] Marreddy, M., Oota, S., Vakada, S, et al., 2021. Clickbait Detection in Telugu: Overcoming NLP Challenges in Resource-Poor Languages using Benchmarked Techniques. 1-8. DOI: <https://doi.org/10.1109/IJCNN52387.2021.9534382>
- [23] Marreddy, M., Oota, S.R., Vakada, L.S., et al., 2022. Multi-task text classification using graph convolutional networks for large-scale low resource language. *ArXiv.org*. DOI: <https://doi.org/10.48550/arXiv.2205.01204>
- [24] Marreddy, M., et al. Am I a Resource-Poor Language? Data Sets, Embeddings, Models and Analysis for Four Different NLP Tasks in Telugu Language. *ACM Transactions on Asian and Low-Resource Language Information Processing*. 22(1), 2023. DOI: <https://doi.org/10.1145/3531535>
- [25] Karim, M.A., Abdullah, M.Z., Deifalla, A.F., 2023. An assessment of the processing parameters and application of fibre-reinforced polymers (FRPs) in the petroleum and natural gas industries: A review. *Results in Engineering*. 18, 101091. DOI: <https://doi.org/10.1016/j.rineng.2023.101091>
- [26] Karim, M.A., Abdullah, M.Z., Waqar, A., et al., 2023. Analysis of the mechanical properties of the single layered braid reinforced thermoplastic pipe (BRTP) for oil & gas industries. *Results in Engineering*. 20, 101483. DOI: <https://doi.org/10.1016/j.rineng.2023.101483>
- [27] Waqar, A., Othman, I., Aiman, M.S., et al, 2023. Analyzing the success of adopting Meta-verse in construction industry: Structural equation modelling. *Journal of Engineering*. DOI: <https://doi.org/10.1155/2023/8824795>
- [28] Karim, M.A., Abdullah, M.Z., Ahmed, T., 2021. An overview: The processing methods of fiber-reinforced polymers (FRPs). *International Journal of Mechanical Engineering and Technology (IJMET)*. 12(2), 10–24. DOI: <https://doi.org/10.34218/IJMET.12.2.2021.002>
- [29] Harahap, P., Rimbawati, R., Evalina, N., et al., 2024. Power Conversion from Solar Panels using a 3000-watt Inverter. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*. 124(2), 260–272. DOI: <https://doi.org/10.37934/arfmts.124.2.260272>
- [30] Cholish, C., Rusdi, M., Rahmawaty, R., 2024. Performance Measurement of Peltier Element Design using Solar Test Simulator Concerning Light and Temperature Parameters. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*. 123(2), 231–243. DOI: <https://doi.org/10.37934/arfmts.123.2.231243>
- [31] Abdullah, A., Putri, M., Syahrudin, M., et al., 2024. Optimization of Solar Power Plant with Variation of Solar Reflector Angles and Use of Passive Cooling Integrated Internet of Things. *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*. 124(1), 233–248. DOI: <https://doi.org/10.37934/arfmts.124.1.233248>