Stochastic Modelling and Computational Sciences

A COMPREHENSIVE RESEARCH ON EMOTION DETECTION TECHNIQUES FOR PURE AND CODE MIXED INDIAN REGIONAL LANGUAGES

Neha Ashok[*] and Sharvari Govilkar
Department of Computer Engineering,University of Mumbai,PCE,New Panvel, India
ashokneh21me@student.mes.ac.in and sgovilkar@mes.ac.in

ABSTRACT
Research on emotion detection in Indian regional languages is crucial due to India's diverse linguistic landscape. Detecting emotions in regional languages has numerous applications, including sentiment analysis, customer service, and mental health assessment. Various approaches exist for this task, such as rule-based systems, machine learning, and deep learning. Among these, machine learning is the most commonly employed method, wherein a model is trained on annotated texts containing emotional information like happiness, sadness, anger, fear, and others.However, several challenges hinder emotion detection in regional languages. Annotated datasets are predominantly available in English or other widely spoken languages, necessitating the creation of regional language datasets. Additionally, the complexity of regional languages poses difficulties in accurately discerning emotions. To address these challenges, researchers are actively exploring new approaches.Despite the challenges, progress is being made in the development of effective techniques and tools for emotion detection in Indian regional languages. This paper is trying to do a comparative survey of different techniques used for emotion detection in pure and code mixed Indian Regional languages.

Keywords: NLP, sentiment, emotion detection, classification techniques, challenges, features, machine learning, deep learning ,indian regional languages .

I. INTRODUCTION

The identification of human emotions in written text, known as Emotional Detection, has gained significant attention in research. This field has become increasingly popular as advanced techniques like NLP, machine learning, and computational linguistics are employed to achieve accurate results.

Investigating emotional identification has become increasingly important as it provides valuable insights. Emotions can be conveyed through various mediums such as speech, facial expressions, written text, and gestures. The task of detecting emotions in written text poses a classification challenge that requires the application of natural language processing (NLP) and deep learning techniques. NLP approaches enhance learning-based models by considering the semantic and syntactic characteristics of the text. Emotion detection in text is a relatively new research field closely related to sentiment analysis. While sentiment analysis aims to identify positive, neutral, or negative sentiments, emotion analysis focuses on recognizing specific emotions like anger, disgust, fear, happiness, sadness, and surprise. However, analyzing emotions in Indian languages, whether in their pure form or code-mixed, presents additional challenges due to limited labeled datasets and research in this area. Hence, this paper aims to explore different methods for emotion detection in social media texts written in pure and code-mixed Indian languages.

The purpose of emotion detection is to analyze and interpret human emotions expressed in various forms of communication. It aims to enhance understanding of human behavior, improve communication and human-computer interaction, aid in sentiment analysis and mental health assessment, guide market research and customer feedback analysis, and contribute to decision-making processes in fields such as psychology, healthcare, and artificial intelligence. Emotion detection serves to uncover valuable insights, facilitate empathetic and responsive interactions, and enable informed decisions across a wide range of domains.

Various methods are employed for detecting emotions in text, including lexicon-based approaches, machine learning algorithms, deep learning models, rule-based approaches, and hybrid methods. These techniques analyze textual data to identify the emotions conveyed in the text, offering valuable insights into its emotional content.

The choice of method depends on factors such as available resources, the complexity of the emotions being analyzed, and the desired level of accuracy. The paper provides a comprehensive survey of different emotion detection techniques, discussing related work and previous literature. Section 2 explores the baseline algorithm and the challenges associated with emotion detection. Section 4 delves into two main categories of emotion detection techniques: machine learning-based emotion detection and deep learning-based sentiment analysis, with a comparison of each method's applicability in Indian regional languages. The paper concludes in Section 5.

II. LITERATURE SURVEY

In this section we cite the relevant past literature of research work done in the field of emotion detection for Indian languages.

A reliable and authentic dataset for emotion detection in the context of the COVID-19 pandemic was created by Md. Rumman Hussain Khan Rahib, Amzad Hussain Tamim, Mohammad Zawad Tahmeed, and Mohammad Jaber Hossain[1]. The dataset consists of 10,581 entries from diverse sources. The researchers explored different machine learning methods, including Support Vector Machines (SVM) and Random Forest as classical approaches, and Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) as deep learning techniques. Their study demonstrated that deep learning algorithms outperformed traditional methods, with Random Forest achieving an accuracy of 78.94% and SVM achieving 75.72%. CNN, coupled with Word2Vec static embedding, achieved an impressive accuracy score of 81.91%.

A benchmark study conducted by Abdullah Al Jamil and Rifat Rahman[2] focused on sentiment and emotion detection in Bangla text using social media platforms like Twitter, Reddit, and YouTube. The study compared traditional machine learning methods and neural network-based approaches. Various feature extraction techniques and models were evaluated, revealing that the Convolutional Neural Network (CNN) with word2vec (SG) achieved the highest accuracy for sentiment classification. For emotion detection, the Bidirectional Long Short-Term Memory (Bi-LSTM) with word2vec (SG) performed the best. Deep learning models consistently outperformed machine learning-based methods such as Support Vector Machines (SVM), Naive Bayes (NB), Random Forest, and Decision Tree. CNN excelled in sentiment analysis, while BiLSTM excelled in emotion analysis. Deep learning models achieved a minimum of 5% higher accuracy than machine learning approaches in sentiment analysis.

To train transformer-based models, Vedant Jadhav, Shantanu Tripathi, Yashdeep Shetty, Aakash Tiwari, and Arun Chauhan[3]used Marathi tweet data. The dataset was split into training, validation, and testing sets, containing 12,114, 1,500, and 2,250 tweets respectively. It was pre-annotated and balanced across different emotion classes. All tweets were written in Marathi and covered a wide range of emotions. The study found that transformer-based models outperformed traditional machine learning models. The performance improved with more complex architectures, including LSTM and BiLSTM, which incorporated sequential information from previous and bidirectional contexts. Among the transformer models, MuRIL exhibited the best performance, surpassing mBERT and IndicBERT, achieving accuracy scores of 82.71%, 83.60%, and 84.04% using MuRIL embedding, Dense Layer, LSTM, and BiLSTM architecture respectively.

L3CubeMahaSent, introduced by Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi[4] is the first publicly available dataset for Marathi Sentiment Analysis. The dataset comprises approximately 16,000 unique tweets. Deep learning models, including CNN, BiLSTM, ULMFiT, mBERT, and IndicBERT, were employed for sentiment prediction. The combination of IndicBERT and CNN, using Indic fastText word embeddings, achieved the highest accuracy. CNN models had a slight advantage over BiLSTM models. Indic-NLP's Marathi word embeddings performed better than the original versions from Facebook. ULMFiT showed comparable results to simple CNN and BiLSTM models. In 2-class classification, the CNN model with trainable Indic fastText embeddings yielded the best results, slightly outperforming IndicBERT. For more challenging 3-class classification, IndicBERT performed the best. The CNN model with Indic fastText-Trainable achieved 83.24% accuracy in 3-class classification and 93.13% accuracy in 2-class classification. The

BiLSTM model with Indic fastText-Trainable achieved 82.89% accuracy in 3-class classification and 92.67% accuracy in 2-class classification. The BERT model with IndicBERT (INLP) achieved 84.13% accuracy in 3-class classification and 92.92% accuracy in 2-class classification.

Anjum Madan and Udayan Ghose[5]conducted Sentiment Analysis on Hindi tweets using machine learning techniques. They employed two main approaches: Lexicon Based Approach (LBA) and Machine Learning Approach (MLA) based on supervised learning. LBA utilized the enhanced Hindi SentiWordNet dictionary, while the Hybrid Based Approach (HBA) combined LBA and MLA for classifying movie tweets as positive or negative. HBA outperformed LBA in their study. LBA achieved accuracies of 60.31% for pos_hindi.txt and 62.78% for neg_hindi.txt. HBA, combining LBA with supervised machine learning algorithms, yielded good results, with the DT classifier achieving 92.97% accuracy. Using Tf-Idf for feature matrix generation proved effective in reducing the impact of frequent but insignificant words. Machine learning classifiers demonstrated superior performance compared to the lexical resource-based approach. Tf-Idf yielded better results than the Unigram Model in the second approach.

Kush Shrivastava and Shishir Kumar[6]developed a corpus of Hindi movie reviews by collecting data from review websites and manually labeling them into positive, negative, and neutral categories. They introduced a GRU (Gated Recurrent Unit) architecture using an efficient genetic algorithm to classify the reviews. The genetic algorithm optimized the hyperparameters of the GRU, reducing optimization time compared to grid search. This approach successfully addressed challenges in Hindi sentiment analysis. The GRU model incorporated semantic features from Hindi word embeddings and achieved superior performance compared to resource-based and machine learning approaches. It even outperformed existing deep learning methods for sentiment analysis in Hindi data, achieving an accuracy of 88.2%.

Anshul Wadhawan and Akshita Aggarwal[7]conducted a study comparing different models for fine-grained emotion classification. They explored and analyzed models such as SVM with RBF kernel, CNN, RNN-based models (Bi-LSTM, Bi-LSTM attention), BERT, RoBERTa, and ALBERT, using word2Vec and FastText-based word representations. Their proposed deep learning models outperformed existing state-of-the-art models for the six emotion labels. The highest accuracy of 71.43% was achieved by BERT. Models utilizing embeddings trained on a combination of Hinglish and English data showed better performance than those trained solely on Hinglish data. The researchers also introduced a publicly available dataset of Hindi-English code-mixed tweets, classified into six emotions. They compared the performance of embeddings trained exclusively on Hinglish tweets with those trained on a mixture of Hinglish and English tweets. The study presented and analyzed the performance of these embeddings in both scenarios.

Shubham Das and Dr. Tanya Das[8]explored multiple models and used SVM as the baseline, which achieved 58% accuracy with the same sentiment labels as a previous study. Among deep learning models, CNN showed the highest overall accuracy and mean absolute error (MAE). Conversely, Bi-LSTM with attention achieved the best validation accuracy for the Hinglish Code Mixed dataset. The study focused on evaluating various deep learning algorithms using FastText for text embedding. The evaluation used a labeled Hinglish code-mixed dataset of tweets obtained through the TweetScrapper API, representing emotions like joy, unhappiness, frustration, worry, disgust, and amazement. Among the proposed models, CNN outperformed the others with an accuracy of 75.25%.

T Tulasi Sasidhara, Premjith Ba, and Soman K Pa[9]conducted an experiment using 12,000 code-mixed social media texts. Each text underwent cleaning, tokenization, and training a model to generate word vectors. The embedding matrix indexed each word in the corpus and accessed the corresponding vector from the Embedding layer. The survey results indicated that 1D-CNN showed promising results in NLP classification tasks, as word proximity may not reliably indicate trainable patterns. The initial experiment focused solely on 1D-CNN without LSTM layers. Subsequent experiments incorporated LSTM and BiLSTM models while excluding the CNN layer to leverage sequential pattern analysis. Additionally, CNN-LSTM and CNN-BiLSTM models were utilized to abstract features and reduce training complexity. In total, five models were experimented with: CNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM. Among them, the CNN-BiLSTM model achieved a classification accuracy of 83.21%. These results demonstrated the superior performance of CNN-BiLSTM over the other models, including its hyperparameters.

Konark Yadav, Aashish Lamba, Dhruv Gupta, Ansh Gupta, Purnendu Karmakar, and Sandeep Saini[10]conducted an analysis using four baseline models. Initially, they tested these baseline models along with their proposed Ensemble Classifier and BI-LSTM models. They evaluated Precision (P), Recall (R), F score (F), and the best accuracy for each model across Positive, Negative, and Neutral sentiments. The proposed Ensemble model showed the highest average accuracy among all models. The proposed Bi-LSTM model consistently provided similar P, R, and F scores for each sentiment category. The Bidirectional-LSTM classifier achieved the best accuracy of 0.78. On the other hand, the Ensemble model performed well in classifying each sentiment type and consistently achieved high precision, recall, and F scores. The ensemble approach improved the average accuracy in sentiment classification by leveraging the best results from four classifiers and outperformed existing systems for Punjabi-English code-mixed sentiment analysis. The second model focused on consistently analyzing sentiments within each category, achieving a precision, recall, and F score of 0.59 with an average accuracy of 0.78.

III. EMOTION DETECTION

Emotion detection is a field of study that aims to analyze and interpret human emotions through various means, such as facial expressions, voice tone, and physiological signals. By employing machine learning and artificial intelligence techniques, researchers develop algorithms to classify emotions like happiness, sadness, anger, or fear. These algorithms process input data, extracting relevant features and mapping them to specific emotional states. Emotion detection finds applications in diverse areas, including psychology, human-computer interaction, and marketing. It enables computers and systems to recognize and respond to human emotions, enhancing user experiences and enabling more empathetic and personalized interactions in the digital realm.

A. Challenges for Emotion Detection: Text-based emotion detection in NLP faces several challenges. First, emotions are subjective and context-dependent, making it difficult to accurately capture the intended emotional states from text. Second, the scarcity of annotated datasets for training models in different languages and domains limits their generalizability. Third, understanding sarcasm, irony, and other forms of figurative language adds complexity to the task. Fourth, handling code-mixed or multilingual text requires specialized techniques. Fifth, emotions may be expressed implicitly or subtly, necessitating the model to capture subtle linguistic cues. Lastly, addressing the imbalance of emotions in real-world data and achieving consistent performance across various emotion categories remain ongoing challenges.

B. Features for Emotion Detection: Text-based emotion detection in NLP involves extracting various features from the text to capture emotional content. These features can include lexical features such as sentiment words, emoticons, and intensifiers that convey emotional intensity. Syntactic features like part-of-speech tags and syntactic structures provide insights into grammatical patterns associated with different emotions. Semantic features such as word embeddings or contextualized representations capture the meaning and context of words. Linguistic features like sentiment shifters and negation words help identify shifts in emotional polarity. Additionally, discourse features such as discourse markers and rhetorical devices aid in understanding emotional expressions within a larger discourse context. Incorporating these diverse features enhances the accuracy and robustness of emotion detection models.

C. Lexicon Resources for Emotion Detection: Lexicon resources play a crucial role in text-based emotion detection in NLP by providing pre-defined lists of words associated with specific emotions. These resources include sentiment lexicons, emotion lexicons, and affective dictionaries. Sentiment lexicons contain words with polarity labels (positive, negative, neutral) to determine the overall sentiment of the text. Emotion lexicons consist of words associated with specific emotions (e.g., joy, sadness, anger) and their intensity. Affective dictionaries provide more fine-grained emotion labels and intensity scores. Well-known lexicon resources include WordNet, SentiWordNet, AFINN, NRC Emotion Lexicon, and EmoLex. These lexicons serve as valuable references for identifying and analyzing emotional expressions in text data.

D. Baseline Algorithm: A general baseline algorithm for emotion detection in text can be achieved using a machine learning or deep learning approach with the following steps:

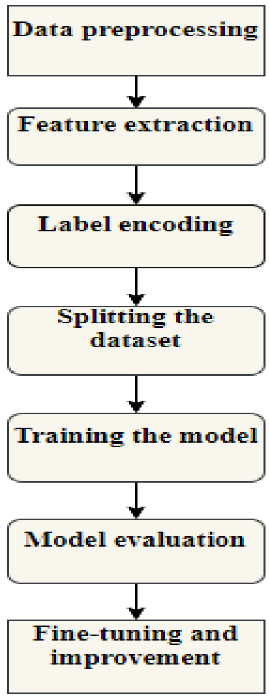

Figure 1: Baseline Algorithm for emotion detection

i. **Data Preprocessing**:Clean and preprocess the text data by removing any unnecessary characters, normalizing the text (lowercasing, removing punctuation), and handling any specific requirements based on your dataset.

ii. **Feature Extraction:** Represent the text data with numerical features that capture relevant information for emotion detection. Common techniques include bag-of-words representations, TF-IDF values, or word embeddings like Word2Vec or GloVe. Additionally, you can consider using more advanced techniques such as contextual word embeddings like BERT or sentence embeddings like Universal Sentence Encoder.

iii. **Label Encoding:** Assign numerical labels to each emotion category in your dataset (e.g., 0 for "happy," 1 for "sad," etc.) for training and evaluation purposes.

iv. **Splitting the Dataset:** Divide your dataset into training and testing sets. The training set will be used to train the machine learning model, while the testing set will be used to evaluate its performance.

v. **Training the Model:** Apply a supervised learning algorithm to the training data. Common choices include logistic regression, support vector machines (SVM), random forests, or gradient boosting algorithms like XGBoost or LightGBM. These algorithms learn patterns in the features and their relationship with the labeled emotions.

vi. **Model Evaluation:** Use the trained model to predict emotions for the test instances. Compare the predicted labels with the true labels from the testing set to evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.

vii. **Fine-Tuning and Improvement:** Analyze the performance of the baseline model and experiment with various techniques to enhance emotion detection accuracy. This can involve feature engineering, trying different algorithms or ensemble methods, incorporating additional contextual information, or leveraging domain-specific knowledge.

IV. EMOTION DETECTION TECHNIQUES:

A.GA-GRU Algorithm (Hindi[6]): The GA-GRU algorithm in natural language processing (NLP) combines Genetic Algorithm (GA) with Gated Recurrent Unit (GRU) neural networks. It uses the GA to optimize the hyperparameters of the GRU model, such as the number of hidden units and learning rate, to enhance performance. The algorithm initializes a population of candidate solutions, where each solution represents a set of hyperparameters. Through fitness evaluation and selection, the GA evolves the population over several generations, favoring solutions with better performance. The selected hyperparameters are then used to train the GRU model, improving its ability to understand and generate language-based sequences.

B.Bidirectional LSTM (Bengali [2], Punjabi english[10],Hinglish[9]): The Bidirectional LSTM (Long Short-Term Memory) algorithm in natural language processing (NLP) is a variant of recurrent neural networks (RNNs) that can capture contextual information from both past and future inputs. It consists of two LSTM layers, one processing the input sequence in the forward direction and the other in the reverse direction. This enables the model to access information from preceding and subsequent words simultaneously, enhancing its understanding of context. By capturing both past and future dependencies, the Bidirectional LSTM is effective in tasks like sentiment analysis, named entity recognition, machine translation, and other NLP tasks that require comprehensive context understanding.

C. MuRIL+Bidirectional LSTM (Marathi [3]): The Multilingual Representations for Indian Languages (MuRIL)+Bidirectional Long Short-Term Memory (BiLSTM) algorithm in emotion detection leverages MuRIL's contextual embeddings, providing multilingual support for Indian languages. By incorporating BiLSTM, it captures longer dependencies and enhances emotion understanding. The model is capable of handling diverse Indian languages and code-mixed data, making it effective in detecting emotions across India's linguistic diversity. The combination offers improved cross-lingual emotion analysis, making it a promising approach for creating accurate and inclusive emotion detection systems for Indian regional languages.

D. LSTM(Bengali[1]): The LSTM (Long Short-Term Memory) algorithm in natural language processing (NLP) is a type of recurrent neural network (RNN) that can effectively model long-range dependencies in sequential data, such as sentences or documents. Unlike traditional RNNs, LSTM networks utilize memory cells with gating mechanisms to regulate the flow of information. These gates determine when to forget or remember information from previous time steps, enabling the model to retain relevant context and avoid the vanishing gradient problem. LSTMs have proven successful in various NLP tasks, including language modeling, text classification, machine translation, and sentiment analysis, due to their ability to capture and preserve important contextual information.

E. CNN(Bengali[2],Marathi[4],Hinglish[7],Hinglish[8],Hinglish[9]): The CNN (Convolutional Neural Network) algorithm in natural language processing (NLP) is a deep learning model commonly used for text classification and sentiment analysis tasks. It applies convolutional operations to capture local patterns and features in textual data. In NLP, the input text is typically transformed into word embeddings or character embeddings. The CNN model then applies convolutional filters over these embeddings to extract relevant features. Max-pooling or other pooling techniques are applied to select the most important features. Finally, fully connected layers and softmax activation are used for classification. CNNs in NLP have shown strong performance in tasks such as sentiment analysis, text classification, and named entity recognition.

F. BiLSTM + CNN(Hinglish[9]): The BiLSTM and CNN algorithm together in natural language processing (NLP) combines the strengths of Bidirectional LSTM (BiLSTM) and Convolutional Neural Network (CNN) architectures to effectively capture both local and global dependencies in text data. The BiLSTM component processes the input sequence in both forward and backward directions, capturing contextual information. The output from the BiLSTM is then passed through a CNN, which applies convolutional filters to extract local features. The combination of BiLSTM and CNN enables the model to capture both short-term and long-term dependencies in the text, making it suitable for tasks such as sentiment analysis, text classification, and named entity recognition.

G. BERT(Hinglish[7]): The BERT (Bidirectional Encoder Representations from Transformers) algorithm in natural language processing (NLP) is a revolutionary language model based on the Transformer architecture. It learns contextualized word representations by training on a large corpus of text data. BERT captures bidirectional context by leveraging a masked language modeling objective and a next sentence prediction task. This allows BERT to understand word meaning in context and generate rich word embeddings. By pre-training on a large corpus and fine-tuning on specific downstream tasks, BERT has achieved state-of-the-art results in various NLP tasks, including question answering, sentiment analysis, named entity recognition, and machine translation.

H. DT TF-IDF(Hindi[5]): The DT TF-IDF (Decision Tree with Term Frequency-Inverse Document Frequency) algorithm in natural language processing (NLP) is a text classification approach that combines the power of decision trees with the TF-IDF feature representation. TF-IDF calculates the importance of a term in a document by considering its frequency in the document and inversely weighting it by its frequency across all documents. The DT TF-IDF algorithm builds a decision tree using these TF-IDF weighted features to classify new documents. It is particularly useful for tasks like sentiment analysis, topic classification, and document categorization, where it can effectively handle high-dimensional text data and capture important term characteristics for classification.

PURE LANGUAGES:

Table 1: Summary of papers and techniques used for emotion detection

Sr no	techniques used	language used in	authors	advantages	disadvantages
1	**LSTM**	**Bengali**	**Md. Rumman Hussain Khan Rahib et.[1]**	LSTM captures sequential dependencies, retains context, and automates feature learning, enhancing emotion detection in text.	Computationally complex, prone to overfitting, struggles with very long dependencies, lacks explicit interpretability, and requires sufficient data for optimal performance.
2	**Bidirectional LSTM**	**Bengali**	**Abdullah Al Jamil et.[2],**	Bidirectional LSTM combines past and future context, capturing longer dependencies, and improving emotion understanding. It retains memory, automates feature learning, and handles variable-length input.	Increased computational complexity, potential overfitting, lack of explicit interpretability, and tuning hyperparameters for optimal performance can be challenging.
3	**MuRIL+ BiLSTM**	**Marathi**	**Vedant Jadhav et.[3]**	The MuRIL and BiLSTM captures contextual information and longer dependencies, enhancing	This approach may still face challenges with overfitting,hyperparameter tuning, and computational

				emotion detection in diverse Indian languages.	complexity due to the integration of both models. Additionally,interpretability can be limited.
4	**CNN**	**Bengali**	**Abdullah Al Jamil et.[2]**	CNN efficiently extracts local features from text, aiding emotion detection. It can handle variable-length inputs, learns hierarchical patterns, and requires less computation than LSTM-based models.	Struggles with long-range dependencies, lacks memory retention for sequential patterns, limited interpretability, and might require more data for optimal performance compared to LSTM.
		Marathi	**Atharva Kulkarni et.[4]**		
5	**DT TF-IDF**	**Hindi**	**Anjum Madan et.[5]**	DT with TF-IDF provides interpretability, handles numerical and text data, and avoids overfitting.	Struggles with complex patterns, long-range dependencies,requires manual feature engineering,limited generalization, and may not achieve state-of-the-art performance compared to deep learning methods.
6	**GA-GRU**	**Hindi**	**Kush Shrivastava et.[6]**	GA-GRU combines GA's global search with GRU's memory retention, enhancing emotion detection. It efficiently captures sequential patterns and can optimize model hyperparameters effectively.	GA-GRU may still suffer from overfitting, requires careful tuning of genetic operators, and the interpretation of GA-optimized models can be challenging.

CODE MIXED LANGUAGES:

Sr no	**techniques used**	**papers used in**	**authors**	**advantages**	**disadvantages**
7	**Bidirectional LSTM**	**Hinglish[9]**	**T Tulasi Sasidhara et.[9]**	Bidirectional LSTM combines past and future context, capturing longer dependencies, and improving emotion understanding. It retains memory, automates feature learning, and handles variable-length input.	Increased computational complexity, potential overfitting, lack of explicit interpretability, and tuning hyperparameters for optimal performance can be challenging.
		Punjabi-english[10]	**Konark Yadav et.[10]**		
8	**CNN**	**Hinglish[**	**Anshul**	CNN efficiently extracts local	Struggles with long-range

		7]	**Wadhawan et.[7]**	features from text, aiding emotion detection. It can handle variable-length inputs, learns hierarchical patterns, and requires less computation than LSTM-based models.	dependencies, lacks memory retention for sequential patterns, limited interpretability, and might require more data for optimal performance compared to LSTM.
		Hinglish	**Shubham Das et.[8]**		
		Hinglish	**T Tulasi Sasidhara[9]**		
9	**BERT**	**Hinglish**	**Anshul Wadhawan et.[7]**	BERT captures contextual information, enhancing emotion detection in text. It doesn't require manual feature engineering and can handle variable-length input. BERT's pre-training enables transfer learning, even with limited labeled data.	BERT is computationally expensive, especially for fine-tuning on specific emotion datasets. Fine-tuning can still overfit with small data. Its large model size may be challenging to deploy in resource-constrained environments.
10	**BiLSTM +CNN**	**Hinglish**	**T Tulasi Sasidhara et.[9]**	BiLSTM+CNN combines the strengths of both models, capturing long-range dependencies with BiLSTM and local patterns with CNN. It can handle variable-length input and enhances emotion understanding.	Increased computational complexity due to the combination of models. Hyperparameter tuning can be challenging. Interpretability remains limited, and overfitting is possible with small datasets.

V. CONCLUSION

In the case of both pure and code mixed languages,deep learning models do better than machine learning models. In case of pure languages BiGRU model gives the best performance for the Bengali language,GA-GRU model gives the best performance for Hindi language.In case of code mixed languages BiLSTM and CNN-BiLSTM model gives the best performance in Punjabi English and hinglish languages respectively.

The future scope for text-based emotion detection in pure and code-mixed Indian regional languages includes advancements in language-specific models, larger annotated datasets, handling code-mixing challenges, domain-specific applications, real-time systems for better human-computer interaction, and improved interpretability for user trust. Currently in case of code mixed languages less work is done compared to pure languages despite code mixed being more in use compared to pure languages. A Lot of work is to be done in code mixed languages and hence the spectrum of emotions detected and accuracy is to be increased extensively.These developments will lead to more accurate and inclusive emotion detection systems, catering to India's linguistic diversity.

VI. ACKNOWLEDGEMENTS

Visualizing any project without the guidance of experts who have already treated this path before, is difficult. I would take this opportunity to thank my mentor and **HOD Dr. Sharvari Govilkar** and M.E. Coordinator **Dr. Prashant Nitnaware** for their guidance in this project and also for providing me all the details for the presentation. I am also grateful to our principal **Dr. Sandeep Joshi** for extending his help directly and indirectly through various channels in my project work.

VII. REFERENCES

[1] Md. Rumman Hussain Khan Rahib,Amzad Hussain Tamim, Mohammad Zawad Tahmeed,Mohammad Jaber Hossain (2022) "Emotion Detection Based on Bangladeshi People’s Social Media Response on COVID-19”.

[2] Abdullah Al Jamil,Rifat Rahman (2022) "sentiment and emotion analysis from textual data in bangla language”

[3] Vedant Jadhav, Shantanu Tripathi, Yashdeep Shetty, Aakash Tiwari and Arun Chauhan (2022) "Sentiment Analysis of Marathi Tweets: A Comparative Study of Transformer Based Models”.

[4] L3CubeMahaSent: A Marathi Tweet Based Sentiment Analysis Dataset Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar,and Raviraj Joshi "L3CubeMahaSent: A Marathi Tweet Based Sentiment Analysis Dataset", 2021

[5] Anjum Madan,Udayan Ghose (2021) "Sentiment Analysis for Twitter Data in the Hindi Language"

[6] Kush Shrivastava and Shishir Kumar "A Sentiment Analysis System for the Hindi Language by Integrating Gated Recurrent Unit with Genetic Algorithm”, 2020

[7] Anshul Wadhawan and Akshita Aggarwal (2021) "Towards Emotion Recognition in Hindi-English Code-Mixed Data: A Transformer Based Approach”

[8] Shubham Das, Dr.Tanya Das (2022) "Sentiment Recognition of Hinglish Code Mixed Data using Deep Learning Models based Approach”.

[9] T Tulasi Sasidhara, Premjith Ba, Soman K pa (2019) "Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text”.

[10] Konark Yadav,Aashish Lamba,Dhruv Gupta,Ansh Gupta,Purnendu Karmakar,Sandeep Saini (2022) "Bilingual Sentiment Analysis for a Code-mixed Punjabi English Social Media Text”