Optimizing Sentiment Analysis in Hindi Poetry: A Hybrid Model Unifying Deep Learning, Machine Learning, and Metaheuristic Techniques

VINOD KUMAR[1], ARCHISMITA GHOSH[2], KANDIKATTU SAI RACHANA[3], TEETAS BHUTIYA[4], SUKANYA WATTAL[5]

[1] *Undergraduate Student, (Computer Science Engineering), ADGIPS, Delhi, India*

[2] *Software Engineer, LTI Mindtree, Kolkata, India*

[3] *Software Engineer, Samsung Research and Development, Delhi, India*

[4] *Software Development Engineer, Zigram, Gurugram, India*

[5] *Data Scientist, Hero MotoCorp, Gurugram, India*

Abstract- ***Sentiment analysis, an automated computational methodology employed for the investigation and assessment of sentiments, emotions, and feelings conveyed in comments, feedback, or critiques, utilizes machine learning techniques to discern text patterns proficiently. This research leverages supervised machine learning, specifically exploring its application in the sentiment analysis of Hindi poetry-based text through the validation of model feasibility and accuracy using the Hindi Poetry Sentiment Corpus. The study delves into the examination of prevalent supervised machine learning techniques, including Multinomial Naive Bayes, Logistic Regression, and Random Forest, alongside deep learning methodologies such as Long Short-Term Memory and Convolutional Neural Networks. To evaluate classifier performance comprehensively, standard datasets are utilized, and metrics such as precision, recall, F1-score, RoC curve, accuracy, running time, and k-fold cross-validation are employed. This analytical approach yields valuable insights into the efficacy of diverse deep learning techniques, aiding practitioners in selecting suitable methods tailored to their specific applications. Furthermore, the investigation incorporates the application of the metaheuristic-based Grey Wolf Optimization technique to discern optimal features from pre-processed data. The genesis of "deep learning" (DL) in artificial neural network research is acknowledged, wherein word vectors trained by Word2Vec are utilized for the input layer (IL) and input into the CNN-LSTM joint model. Subsequently, the output of the joint model undergoes weighting and summation through self-attention before entering the SoftMax classifier, facilitating the emotion classification of the text. Rigorous comparative experiments validate the utility of the proposed model, demonstrating its superior performance over three comparison models [CNN, LSTM, CNN-LSTM] across various evaluation indices. Comparisons with other machine learning techniques, including Random Forest, Logistic Regression, Naive Bayes, CNN, and LSTM, reveal notable accuracies. Specifically, Random Forest, Naive Bayes, CNN, and LSTM achieve accuracies of 87.75%, 85.54%, 91.46%, and 88.72%, respectively. Notably, the proposed ensemble hybrid model attains the highest classification accuracy of 95.54%, precision of 91.44%, recall of 89.63%, and F-score of 90.87%, showcasing its efficacy in sentiment analysis applications.***

Indexed Terms- ***Hindi poetry-based text sentiment analysis, Machine Learning, Deep Learning, Grey Wolf Optimization, natural language processing, CNN-LSTM multi-feature fusion.***

I. INTRODUCTION

The rapid evolution of computing technology has facilitated substantial advancements in the analysis and processing of monolingual text corpora through the application of various Natural Language Processing (NLP) techniques. As a subfield of Artificial Intelligence (AI), NLP involves the

manipulation of sentences in adherence to grammatical principles. While the terms "code-switching" and "code-mixing" are commonly used interchangeably, distinctions arise in their application within natural language data and the utilization of computational methods by both humans and computers [1].

NLP, as a discipline, encompasses the implementation of computational methods for handling natural language data, ranging from elementary tasks such as word frequency analysis to more intricate endeavors, including the comprehension of comprehensive human expressions [2]. Code-switching involves the transition between languages, whereas code-mixing integrates various phonetic units—such as words, phrases, morphemes, clauses, affixes, and modifiers—from one language into the discourse of another. In the domain of language and communication, the term "code" denotes a set of rules dictating the transformation of information from one form of representation to another. Multilingual communities frequently exhibit the practices of code-mixing and code-switching, wherein individuals employ their native language alongside a secondary language across different communicative domains. This linguistic phenomenon manifests as code-switching between sentences and code-mixing within focused language expression. These practices are integral to training machines to process and comprehend human language, facilitating natural language-based interactions [2].

Given the prevalence of multilingualism in online communication and social media platforms, code-mixing and code-switching emerge as common phenomena among multilingual speakers during informal interactions. Various factors contribute to code-mixing, including bilingualism, the linguistic backgrounds of conversational partners, social community norms, situational factors, vocabulary availability, and language prestige [3]. A notable motivator for code-mixing or switching is the unavailability of a specific word or phrase in a given language, compelling individuals to draw from their native language to enhance recipient comprehension [3]. This linguistic complexity is particularly pronounced on social media platforms within multilingual societies, where individuals often blend multiple languages to convey their sentiments. Notably, they frequently employ the Roman script instead of native language scripts when incorporating non-English words, posing a challenge for automatic language detection. Within this context, sentiment analysis, also referred to as opinion mining or emotion analysis, involves the identification, recognition, or categorization of individuals' perspectives and reviews on diverse subjects—such as services, products, social issues, events, or moments—into positive, negative, and neutral classes [4].

In the realm of sentiment analysis for bilingual or multilingual text, this research delves into methodologies and approaches rooted in both machine learning (ML) and deep learning (DL). The ensuing discussion encompasses an exploration of their respective applications and the outcomes derived from their implementation across diverse scenarios and datasets. The ensuing section encapsulates the principal discoveries emanating from the investigation:

- The initial phase of this study involves the application of diverse Natural Language Processing (NLP) techniques to perform data pre-processing, which precedes the construction of labeled feature vectors.
- Subsequently, the integration of machine learning (ML) models, specifically Random Forest, Logistic Regression, and Naïve Bayes is employed for comprehensive analysis.
- Conclusively, a sophisticated approach is undertaken through the incorporation of an ensemble deep learning model, combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures. This amalgamated model is specifically applied to the task of sentiment classification, discerning sentiments as positive, neutral, or negative.
- Furthermore, the optimization of feature selection is addressed through the utilization of the Grey Wolf Optimization algorithm, contributing to the enhancement of the overall model's performance and efficacy.

II. LITERATURE REVIEW

A pattern-based method for identifying aspects and analyzing sentiment was developed by Rodrigues et al. [18]. In this instance, explicit aspect syntactic pattern is extracted from product sentiments using pattern analysis. To determine the sentiment polarity of the sentence, Senti-Wordnet is used to extract the bigram characteristics. This investigation shows that the multi-node clustering strategy works better than the single-node clustering approach.

Proportional Rough Feature Selector (PRFS) is a filter-based feature selection technique used by Cekik et al. [16] to test feature selection with a variety of classifiers, including SVM, DT, KNN, and Naive Bayes. To assess whether documents belong to a particular class or not, PRFS employs crude set theory. A 95% confidence level classifier performance is improved.

Endsuy [14] performed an exploratory data study on the 2020 US Presidential Election using Twitter Datasets. They contrast the attitude expressed in location-based tweets with local public opinion. Using the Open Cage API and sciSpacy NER, they gather features like latitude, longitude, city, country, continent, and state code. They make use of two November 18, 2020-dated Kaggle datasets about Joe Biden and Donald Trump. They employ a valence model for lexicon-based feature extraction.

A Cooperative Binary-Clustering Framework was developed by Bibi et al. [15] for sentiment analysis on Indigenous data sets utilizing Twitter. They integrate single linkage, complete linkage, and average linkage procedures and divide the data using majority voting. They split the cluster into positive and negative groups based on the confusion matrix. They employ word polarity, TF-IDF, and unigram techniques for feature selection.

The feature set for sentiment analysis is minimized using the Apirori algorithm by Jain et al. [17], who also created a feature selection method based on association rule mining. They employed supervised classification techniques for the experiments, including logistic regression, support vector machines, random forests, and naive bayes.

Chandra et al. [21] applied the Long Short-Term Memory (LSTM) model to perform sentiment analysis. Their study focused on analyzing a dataset of 150,000 Indian COVID-19-related tweets collected from March 2019 to September 2020. Lwin et al. [22] investigated global Twitter patterns concerning COVID-19. Prabhakar et al. [23] conducted COVID-19 topic modeling and sentiment analysis using 18,000 tweets and the National Research Council (NRC) sentiment lexicon. Nemes et al. [24] employed a Recurrent Neural Network (RNN) model to classify the emotional content of tweets into positive or negative categories. Samuel et al. [25] explored public opinion on COVID-19 using Twitter data and achieved a maximum accuracy of 74% by employing the Naïve Bayes classifier, Logistic regression, and linear regression techniques. Mittal et al. [26] developed an annotated corpus for the Hindi language and obtained an 80% classification accuracy using HindiSentiWordNet (HSWN). However, most of the existing research on sentiment analysis has predominantly focused on analyzing social media networks. Gupta et al. [27] evaluated various machine learning techniques, including logistic regression, Naive Bayes, Support Vector Machine (SVM), and Decision Tree, for sentiment analysis of Hindi tweets. They utilized the NRC Emotion Lexicon and Hindi Senti-WordNet Lexicon to identify emotions associated with phrases. Finally, an integrated Convolutional Neural Network (CNN) was proposed for sentiment analysis of 23,767 Hindi tweets, achieving an accuracy of 85% in classifying them as positive, negative, or neutral.

Table 1: Sentiment Classification and Text analysis of Code-Mixed text of Diverse Languages

Existing work	Language	Objective	Dataset(s)	ML/DL Approach	Performance Evaluation
Sasidhar, T. T et. al. [5]	Hindi-English	Development of annotated dataset, classification of emotions	Tweets Facebook posts Instagram comments	CNN-BiLSTM	Accuracy: 83.21%
Kumar & Dhar [6]	Hindi-English	Sentiment Analysis	Facebook posts	BiLSTM	Accuracy: 83.54% F1-score: 0.827.
Veena et. al. [7]	Hindi-English	Language Identification	Facebook posts, Tweets, WhatsApp chats	SVM	Data 1 (f-score =98.70) Twitter data (f-score=93.94) Data 3 (f-score=77.60)
Vijay, Deepanshu, et al [8]	Hindi-English	Sarcasm detection	Tweets	SVM RF	SVM F1 score=0.77 RF F1 score=0.72
Wu, Wang & Huang [9]	Hindi-English Spanish-English	Sentiment Analysis	Tweets	BiLSTM	F1-score: 0.730
Raha, Tathagata, et al. [10]	Bengali-English	POS Tagging	Tweets	LSTM	Accuracy: 75.29%
Pratapa, A et. al.[11]	Hindi-English	POS Tagging, Sentiment Analysis	Tweets	LSTM	F1-score : 0.56
Prabhu, Ameya, et al .[12]	Hindi-English	Corpus creation, Sentiment Analysis	Facebook post	LSTM	Accuracy 69.7%
Gopal & Das [13]	Hindi-English	Sentiment Analysis	Facebook posts	Ensemble (LSTM MNB)	Accuracy 70.8 F1-Score 0.661

Sentiment evaluation of social media data analysis plays a vital role in contemporary commerce and governance. Traditionally, sentiment analysis systems were primarily developed for analyzing product reviews. However, with the advancement of Natural Language Processing (NLP) tools and technologies, sentiment analysis has expanded its scope to various other tasks. Code-mixed text data sentiment analysis poses a unique set of challenges, starting from data collection to classification. Extensive research has been conducted in the domains of Cross-Lingual Information Retrieval (CLIR), Multilingual Information Retrieval (MLIR), and Mixed Script Information Retrieval (MSIR). CLIR focuses on enabling users to query in one language and retrieve information in multiple languages. MLIR, on the other hand, allows queries in one or more languages, with information retrieval across multiple languages. However, MSIR presents a more intricate retrieval task, particularly when dealing with Romanized text of non-English languages. Additionally, social media text exhibits numerous non-standard forms, including misspellings, grammar errors, letter substitutions, non-standard abbreviations, and other ambiguities, making pre-processing an essential step in the code-mixed scenario. To address these challenges, several tools

have been developed for Part-of-Speech (POS) tagging, language identification, and named entity recognition (NER) in code-mixed data. However, the development of automatic text analysis tools remains challenging due to limited datasets, particularly annotated datasets for specific language pairs, and the lack of linguistic resources for most native Indian languages. Furthermore, the absence of linguistic catalogues for informal code-mixed text further exacerbates these challenges in tool development. Almeida et al. use the CNN to model and solve the problem of emotion classification. Rn has a good effect on processing information containing time series data, so it is often applied to natural language processing tasks [28]. Liu et al. used the CNN neural network algorithm to model the feature information existing in the text data and then to deal with the emotion classification problem and achieved good results [29]. Zeng et al. put forward a thesaurus-based algorithm to solve the problem of emotional tendency. This algorithm adopts the bootstrapping strategy, and its emotional tendency ultimately depends on the sum of emotional tendency scores of all emotional words in a sentence [30]; Dang et al. put forward the strategy of goal dependence, considering the influence of the context on Weibo's information emotion. The main realization methods are goal dependence and situational awareness. The goal dependence method refers to judging emotion according to syntactic features, while the situational awareness method is classified by considering the related tweets of each tweet [31].

The thorough literature review of machine learning techniques demonstrates how the suggested approaches speed up text pattern analysis and can be utilized to automate sentiment analysis [19, 20]. Deep learning can manage enormous amounts of data, analyze text patterns more quickly using artificial neural networks, and fix misalignment issues by gleaning local features from sentiments. To achieve high accuracy on these kinds of tasks, a deep learning algorithm accepts word embedding as input. The performance of the deep learning models LSTM and CNN has been examined in the proposed work using the most popular machine learning techniques.

III. METHODOLOGY

Machine Learning & Deep Learning Algorithms incorporated in this research work: -

- Multinomial Naïve Bayes: The Multivariate Event model is referred to as Multinomial Naive Bayes. When most people want to learn about Naive Bayes, they want to learn about the Multinomial Naive Bayes Classifier. However, there is another commonly used version of Naïve Bayes, called Gaussian Naive Bayes Classification.

- Random Forest: Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

- Logistic Regression: Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables.

- Convolution neural network model (CNN): The CNN model contains a convolution layer that minimizes the dimension of input data. A total of 50 filters of 3x3 window is used for feature extraction from input data.

- Long short-term memory model (LSTM): LSTM is a specific recurrent neural network (RNN) model. It uses three gates to maintain and control the long-term dependency along with the state information of each node. The model can also solve the vanishing gradient problem. The feasibility and accuracy of the model are validated through the Hindi poetry sentiment corpus. Artificial neural network research is where the idea of "deep learning" (DL) first emerged. For the IL (input layer), word vectors trained by Word2Vec are used and then input into the CNN-LSTM joint model. Then, the output of the joint model is weighted and summed by self-attention and finally input into the SoftMax classifier, to realize the emotion classification of the text. By creating and putting into practice pertinent comparative experiments, the usefulness of the proposed model is confirmed. The outcomes demonstrate that this model outperforms the other three comparison models [CNN, LSTM, CNN-LSTM] for the quantification of evaluation indices in terms of overall performance. The vector obtained after processing is passed through the SoftMax classifier to achieve the classification of sentiment, and the loss function is set to categorical_crossentropy.

Text pre-processing is a method of formatting text, which needs to process unstructured or semi-structured data such as word segmentation, sentence segmentation, and stop word removal, to extract the required information from rough original data. This section discusses the improved text segmentation method. The main idea of the improvement is that firstly, the maximum matching method is used to roughly segment the text, then, the hidden Markov model is used to mark the part of speech of the segmented words, and then, the part of speech marking and segmentation results are evaluated to get the best segmentation results. The goal of part-of-speech tagging is to reduce the amount of calculation required and increase the algorithm's operating efficiency by removing words that are unnecessary or contribute little to the text's content.

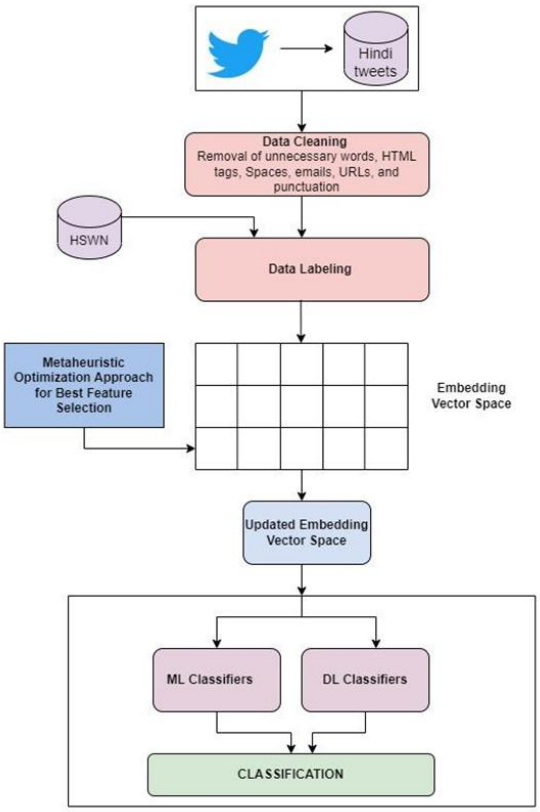

Fig 1: Proposed Framework

Algorithm 1 Hindi text sentiment analysis of HindiSentiwords.net

Dataset: HindiSentiwords.net

Procedure: Sentiment classification as positive, negative, or neutral

1. Scraping of tweets using the Twint library.
2. Store the scrapped data frame as a dataset
3. Apply data cleaning & pre-processing on data frame ghl
   a. ghl <= Delete the null value
   b. ghl <= Remove stop words, @, and URL from dataset.
   c. ghl <= Remove emoticons & punctuations from ghl
   d. ghl <= Tokenization
4. ZA["qt"] <= Calculate the polarity score of tweets in the dataset ghl.
5. Assign sentiment to the text-based on the polarity score.
6. for each ZA[a], where a=0,1,2,3,4……. n do
7. if ZA["qt"] score > 0 then

8. Assign as “Positive”
9. else if ZA[“qt”] score < 0 then
10. Assign as “Negative”
11. else
12. Assign as “Neutral”
13. end if
14. end for
15. ZA[“Sentence”] <= Divide all emoticons assigned in above steps into three categories such as (i) positive, (ii) negative, (iii) neutral.
16. For each ZA[“Sentences”] do
17. The index value of each word is calculated.
18. end for
19. Create a weight matrix for each sentence with indexed value of a token.
20. Evaluate the fitness value of each weight by applying fitness function.
21. Updated_weight_matrix <= Fitness_function[weight_matrix]
22. Optimized_weight_matrix <= GWO[Updated_weight_matrix]
23. Train the proposed hybrid deep learning with Optimized weight_matrix

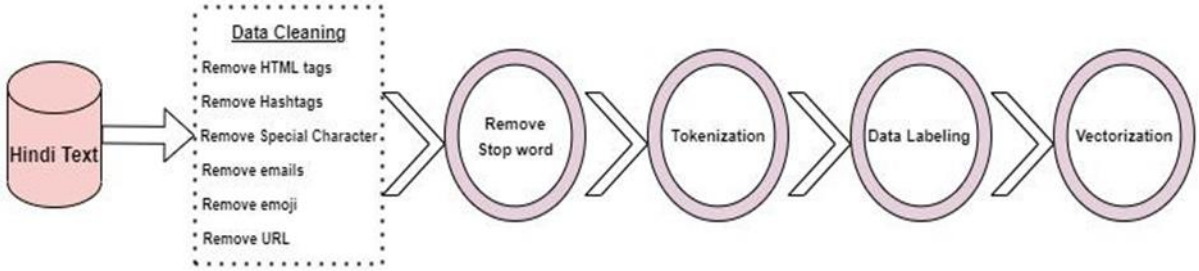

Fig 2: Pre-processing steps

One of the pre-processing techniques that is most frequently applied across many NLP applications is stop word removal. Simply eliminating the words that appear often across all of the corpus's papers is the notion. Pronouns and articles are typically categorized as stop words.

Fig 3. List of Hindi stop words

Tokenization is the division of text into a collection of meaningful fragments. These objects are known as tokens. For instance, we could break up a passage of text into words or phrases. We can create our conditions to separate the input text into relevant tokens depending on the task at hand. In machine learning, data labeling is the process of identifying raw data (images, text files, videos, etc.) and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it.

The rearranging of sentences the final stage of pre-processing, known as vectorization, converts the pre-processed data into a vector of real values.

Grey wolf optimization: The Grey wolf algorithm incorporates four quantitative methods, namely, (i) Hierarchical structure of Grey wolf’s, (ii) Prey encircling, (iii) Prey hunting, and (iv) Prey searching and attacking. Each method is briefly discussed in the following subsection.

Algorithm 2: Algorithm of GWO technique.

1. Set M as the maximum iteration.
2. Set population Bv(v=1,2…q).
3. Initialize d, E and G.
4. Determine the wolf's fitness level.
5. X(α) = Most effective search agent.
6. X(β) = Second effective search agent.

7. X(δ) = Third effective search agent.
8. while Y < M do
9. for each search agent do
10. Reposition the active search agent.
11. end for
12. Update the value of d, E, and G.
13. Determine the fitness level of all search agents.
14. Update the value of (α), (β), and (δ).
15. Y = Y+1
16. end while
17. return B(α)

- Hierarchy structure of Grey Wolf's algorithm: The Grey Wolf algorithm focuses on the quantitative hierarchy of the wolf pack leadership. Alpha (α) represents the highest level of the intellectual hierarchy, whereas (β) denotes the second and third most essential traits, respectively. The goal of the entire process is to determine the calibration of both vectors E and G. The exploitation and exploration are emphasized in almost every dimension. Finally, the GWO algorithm returns an optimal weight matrix denoted as the optimized weight matrix and given as input to the classification model.

Sentiment Categories for Poetry based text:

- Strongly Positive: For the whole poetry, if the poet is using only positive language such as expression of support, motivation, admiration, positive attitude, cheerfulness, forgiving nature, positive emotional state, etc, then the emotional states identified tend to the positive side of Russell's model. These types of poetries were classified as strongly positive.
- Strongly Negative: For the whole poetry, if the poet is using only negative language such as expression of hate, judgment, fear, anger, failure, criticism, negative attitude, etc. These types of poetries were classified as strongly negative.
- Positive: For most of the poetry, if the poet is using positive language but also using negative language in some instances, then those types of poetries were classified as positive.
- Negative: For most of the poetries, if the poet is using negative languages but also using positive language in some instances, then those poetries were classified as negative.
- Neutral: If the poet is using both positive and negative language at equal intervals, then it is hard to tell what type of sentiment is present in the poetry. Those types of poetries were classified as neutral.

IV. RESULT & DISCUSSION

To prove the validity of the CNN-LSTM aesthetic implication analysis model of Hindi language poetics based on the thought of self-attention, it is verified by several different groups of experiments. This paper selects the public Hindi poetry dataset. There are two categories of datasets, positive and negative, which represent positive and negative, respectively. Text sentiment analysis experiments based on DL in this paper are all designed on the open-source DL library Keras, which is a highly modular neural network library written in the Python language. The ReLU function is employed as the activation function in the experiment for the network structure model suggested in this paper, and many groups of convolution kernels are used for training. Convolution kernel window sizes are set to 2, 3, and 4, there are 100 convolution neurons in each filter and 300 HL neurons, and Softmax is used to classify the output layer. The experimental comparative analysis is presented in Table 1 to demonstrate the efficacy of the CNN-LSTM aesthetic implication analysis model of based on the concept of self-attention. Figure 3 displays the outcomes of data visualization. It can be seen that the overall performance of this model is better than that of the other three comparison models for the quantification of evaluation indexes. The accuracy and F1 of this paper are 93.362% and 90.886%, both of which are higher than other models, and the LSTM model has also achieved good classification results. It shows that the model in this paper is superior to the other three models under the same situation of introducing AM at the same time and can achieve a better classification effect.

- Classification results and discussion:

Table 3: Parameter values set as input for DL models.

Parameter name	CNN	LSTM
Epoch	175	143
Batch size	264	232
Max-pooling layer size	2	2
Activation function	Relu	Relu
Pooling layer padding	Same	Same
Learning rate	0.001	0.001
Kernel size	5	5

Table 4: Comparison of proposed models.

Model	Precision (%)	F1-Score (%)	Recall (%)	Accuracy (%)
RF	90.12	89.07	91.47	87.75
LR	88.14	86.32	85.63	89.11
NB	91.44	90.87	89.63	94.55
CNN	87.45	90.25	91.63	91.46
LSTM	88.65	89.75	90.47	88.72

The results show that the most used ML classifier for the sentiment classification of code-mixed Indian language text is SVM followed by NB and RF. Ensemble approaches are also used to classify the code-mixed text. The study also showed that in terms of accuracy and f1- measure, Neural Network approaches perform better than the traditional models. Typically, LSTM and BiLSTM algorithms are being used by researchers for the classification of sentiment in code-mixed datasets. The study reveals that Twitter is the first choice of data collection followed by Facebook and movie/product reviews. Also, appreciable research has been carried out in the Hindi-English public networking site's text followed by Bengali-English. Research has also been carried out in other code-mixed Indian languages such as Punjabi-English, Marathi-English, Telugu-English, and Malayalam-English. However, limited or no annotated datasets, text analysis tools, and SentiWordNets are not available in most of the code-mixed Indian language text.

Table 5: Sentiment Analysis with Word-Level TF-IDF Features

Model	Features	Class	Precision	F1-Score
Random Forest	uni	Positive	0.851	0.867
		Negative	0.907	0.870
	uni-bi	Positive	0.882	0.867
		Negative	0.872	0.884
	uni-bi-tri	Positive	0.886	0.861
		Negative	0.856	0.878
Naïve Bayes	uni	Positive	0.882	0.863
		Negative	0.862	0.880
	uni-bi	Positive	0.863	0.868
		Negative	0.869	0.869
	uni-bi-tri	Positive	0.872	0.871
		Negative	0.832	0.882
Logistic Regression	uni	Positive	0.890	0.851
		Negative	0.843	0.871
	uni-bi	Positive	0.891	0.857
		Negative	0.890	0.873
		Positive	0.891	0.842
		Negative	0.831	0.870

Table 6: Observation:

Range	Interpretation
≤0	Poor agreement
0.01-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-1	Almost perfect agreement

TF-IDF was calculated for unigrams, uni-bigrams, and uni-bi-trigrams. Table 5 illustrates the results of the same for the three classifiers. Even though 733 poetries is a sizable corpus for the task in hand, it does not show a significant increase in accuracy, especially with added bi-gram and tri-gram features. This is because most bi-grams and tri-grams occur sparsely in the entire corpus. Here, on average, Linear-SVM performed better than all the other classifiers. To tackle the problem of sparsity, we conducted experiments using n-grams at the character level. For the baseline, 2-6- and 3-6-character n-grams were used to calculate character-level TF-IDF features.

The dataset was split into a ratio of 4:1 for training and testing. For the baseline experiments, TF-IDF features for word n-grams and character n-grams were used for the task of sentiment classification. All the experiments were conducted using 'scikit-learn' an open-source Python library. Precision, Recall, and F1-score are the three-evaluation metrics that were calculated using 5-fold cross-validation.

For baseline experiments Naive Bayes, Logistic Regression and Random Forest were the classifiers used for baseline experiments. The 733 poetries of the corpus were also passed through various pre-processing phases. Then using TF-IDF weights, vector representations were obtained for each poetry. TF-IDF is a technique to quantify a word in documents. We generally compute a weight to each word which signifies the importance of the word in the corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Text from m Kaggle was taken to examine the various supervised learning techniques for assessing the sentiment analysis process. This data set contains restaurant review text containing 1000 text reviews. Here Anaconda Python platform is used to evaluate and pre-process the restaurant reviews. 70% of the reviews are used for training, while 30% are used to test the supervised learning technique. NLTK is used for pre-processing, and Keras, Tensor flow (backend) is used to create LSTM (RNN with memory) and CNN neural network models [30]. Experiments are carried out by Google Collaboratory which provides a Python development environment and runs code in Google Cloud.

Evaluation Parameters Precision, recall, F1-score, accuracy, AUC score, and training time were used to assess the classifier's performance.

They are calculated by,

Precision = T P/ T P + FP

Recall = T P/ T P + FN

F1 score = 2 × (Recall × Precision) / Recall + Precision

Accuracy = (T P + T N) / (TP + FP + FN + T N)

Here True positive (TP) refers to restaurant reviews that were initially classified as positive and are also expected to be positive. False Positive (FP) reviews are those that are initially identified as negative but are predicted to be positive. True Negative (TN) refers to restaurant reviews that were initially categorized as negative and that the classifier predicted to be negative. False Negative (FN applies to restaurant reviews that are initially positive but are expected to be negative. AUC score specifies the area under the ROC curve from the prediction score.

- Result analysis of Machine Learning Technique for Sentiment Analysis:

Table 5: Performance Analysis of Machine Learning Classifiers for Sentiment Analysis

Classifier	Classification Precision, Recall, F1 Score	Accuracy Train, Test	AUC Score	Time to train (in seconds)

Naive Bayes Multinomial	0.79,0.75,0.77s	0.76,0.7633	0.7642	0.009278
Logistic Regression	0.73,0.61,0.70	0.73,0.73	0.736	0.06633
Random Forest	0.89,0.54,0.67	0.72,0.7233	0.732	0.57832

Initially, pre-processing is carried out in sentiment reviews by removing non-character data such as digits and symbols, as well as punctuation and converting the sentence into lower-case. After pre-processing, cleaned text reviews are converted to numerical data that contain sentiment tokens and sentiment scores called feature vectors. The feature vector is formed TF-IDF vectorizer. The TF-IDF stands for Term-Frequency Times Inverse Document-frequency and it assigns weight to each word based on how often it appears in the review text. The TF-IDF refers to term-frequency times inverse document-frequency which assigns weight to each word based on the frequency of that word appearing in review text. After extracting features with a vectorizer, six machine learning classifiers are used for sentiment analysis: naive Bayes, logistic regression, random forest, linear SVC (Support Vector Classifier), K-nearest neighbor, and decision tree. The performance of the classifier is assessed by precision, recall, F1-score, accuracy, AUC score, ROC curve, and training time. The classification report of the classifier is shown in Table 1. From this table, the highest AUC score obtained for Naïve Bayes Classifier is 0.7642. So Naive Bayes model provides better prediction compared to other machine learning classifiers for this restaurant data set. The table also shows that the time taken for the Naive Bayes model is low compared to other machine learning algorithms. The ROC curve of the machine learning classifier is depicted in Figures 4 to 6.

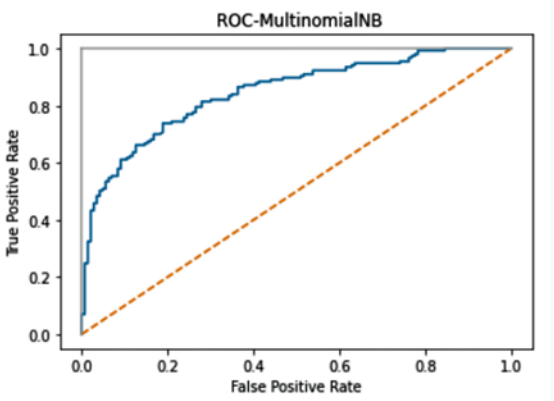

Fig 4: ROC Curve of Multinomial Naïve Bayes

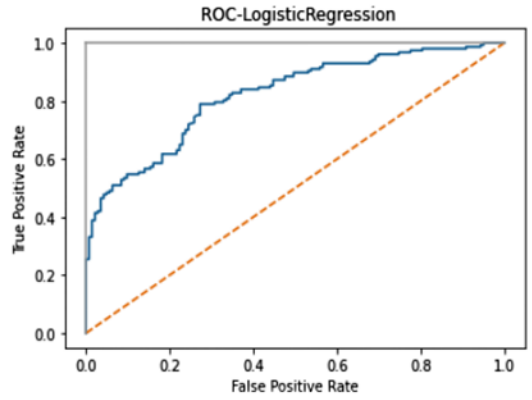

Fig 5: ROC Curve of Logistic Regression

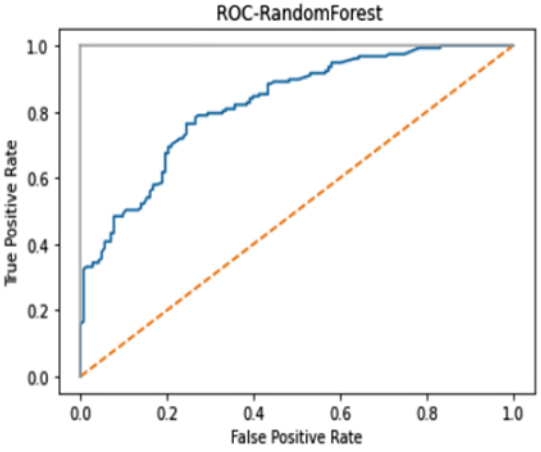

Fig 6: ROC Curve of Random Forest

- Result Analysis of deep learning techniques in Sentiment Analysis:

Table 6: Performance Analysis of Deep Learning Techniques for Sentiment Analysis

Classifier	Classification Precision, Recall, F1 Score	Accuracy Train, Test	AUC Score	Time to train (in seconds)
CNN	0.83,0.79,0.81	0.98,0.845	0.84	7.3
LSTM	0.80,0.67,0.73	0.96,0.77	0.748	9.08

Sentiment analysis is carried out by deep learning techniques CNN and LSTM. After pre-processing the review text is converted into tokens. Following tokenization, the sentiment text is passed to the word2vec model which converts words to vectors. The total number of words obtained for training data is 5118 words, with a vocabulary size of 1839 and a maximum sentence length is 18, while the total number of words obtained for test data is 574 words, with a vocabulary size of 415 and a maximum sentence length of review text is 15. The CNN model passes the input data to a series of layers. It uses a convolution layer to extract features from input data, a max pooling layer to reduce the dimensionality of trainable parameters, and a sigmoid activation function in the output dense layer. Similarly, the input data in the LSTM model is passed to the embedding layer, spatial dropout, LSTM, and output layer. The spatial dropout rate is taken as 0.2 to avoid overfitting. For compilation, both the CNN and LSTM models use the Adam optimizer. The classification report of CNN and LSTM classifiers is shown in Table 6. The ROC curve of CNN and LSTM is shown in Figure 7,8 Finally, K-fold cross-validation (K=10) is carried out to evaluate the machine learning and deep learning algorithm with random seed = 20. The results of the accuracy score for each algorithm obtained are shown in Table 7.

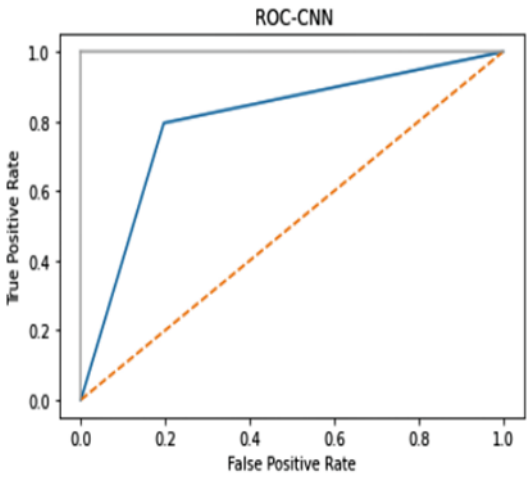

Fig 7: ROC Curve of CNN

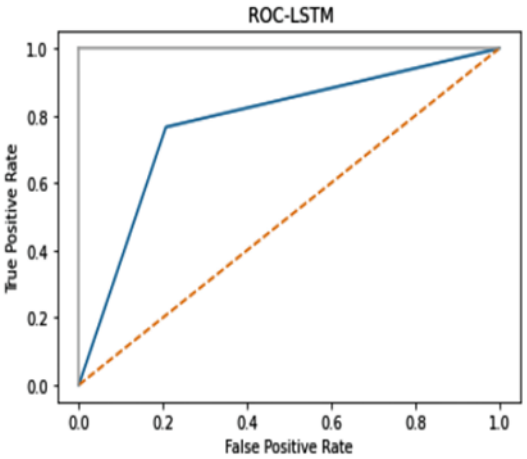

Fig 8: ROC Curve of LSTM

Table 7: Comparison results of different models.

Model	Accuracy	Recall	F1
CNN	84.282	89.37	87.865
LSTM	87.33	90.28	85.985
CNN-LSTM	93.362	91.147	90.886

Analysis of computational complexity: The computational complexity of the entire prediction process is analyzed as: Data-cleaning process requires a total of O (Twitter posts × total word count). The feature extraction step requires T (f) = O(f2) + parsing time. Here, f is the total number of tweets in the sample. The computational complexities of the GWO algorithm for feature selection are summarized as:

Table 8: Computational complexity of the proposed model

Steps	Time Complexity
Data-cleaning process	O (Twitter posts × total word count)
Feature Extraction	T (f) = O (f 2) + parsing time
Features selection with GWO	O (W × q × MI)
Evolution of sentiments	O (b × s (ac + cx + xy))
Forecasting	O (1)

GWO initialization requires O (W ×q) time, here W and q denote population density and problem dimension, respectively. Calculation of GWO process parameters requires O (W × q). O (W × q) time is required to update the wolf position. Evaluation of fitness value requires O (W × q) time. The entire time complexity of the GWO is denoted as O (W × q × MI) and here, MI denotes the optimum iterations. The computational complexity of supervised learning algorithms depends on the number of iterations or the number of classification classes. The time complexity of the CNN-LSTM model is given as O (b × s(ac + cx + xy)), here, a, c, x, and s denote the number of input layer nodes, second layer node, third layer node, and training examples respectively. by and y denotes the total number of epochs and output layer nodes, respectively. The prediction step requires a temporal complexity of O (1). The overall time complexity is O(f). The time complexity of each stage is shown in Table 8.

CONCLUSION

This research paper presents a comprehensive analysis of sentiment in Hindi tweets utilizing both deep learning and machine learning algorithms. The incorporation of the meta-heuristic-based Grey Wolf Optimization technique and a hybrid deep learning model constitutes a focal point of this study. The process involves critical steps such as data pre-processing and labeling through natural language processing, which are pivotal in constructing an optimized feature vector. The optimal features are subsequently identified by employing the Grey Wolf Optimization technique. Moreover, a comparative evaluation is conducted, contrasting the outcomes of the proposed model with those of traditional machine learning models. The proposed hybrid model demonstrates superior performance, achieving a classification accuracy of 95.54%, precision of 91.44%, recall of 89.63%, and F-score of 90.87%. These results signify the efficacy of the model in sentiment classification for Hindi tweets, surpassing the performance of conventional machine learning models.

In a parallel investigation focused on sentiment analysis within a restaurant dataset, a meticulous pre-processing phase is initiated to streamline features and expedite the subsequent classification task. Machine learning methods, including Multinomial Naive Bayes, Logistic Regression, and Random Forest, are juxtaposed with deep learning methods such as LSTM and CNN. The former employs a bag-of-words model approach (TFID-Vectorizer) for text-to-vector conversion, while the latter leverages word embedding methods. The performance of each classifier is rigorously evaluated using diverse metrics, encompassing precision, recall, F1 score, AUC score, and training time. Employing k-fold cross-validation, accuracy scores are determined for each classifier. The findings underscore that neural network-based learning, despite exhibiting higher training accuracy, incurs longer running times compared to machine learning classifiers. A comprehensive analysis of time complexity is incorporated in this investigation. As a prospect for future research endeavors, enhancements to classifier performance are proposed through the adoption of diverse feature selection techniques, along with an exploration of text prediction in multilingual contexts. The study acknowledges the burgeoning landscape of text analysis employing deep learning neural networks, particularly in providing a robust tool for profound semantic analysis of text. Based on an exploration of emotional polarity and similar text classifications, the research constructs a CNN-LSTM aesthetic implication analysis model specific to Indian Hindi poetics, incorporating self-attention mechanisms. The model's overall performance surpasses that of three alternative comparison models, demonstrating accuracy and F1 scores of 93.362% and 90.886%, respectively. The ensuing numerical experiments affirm the efficacy of the proposed

method in text classification while maintaining acceptable running efficiency. The intersection of poetry and emotions is characterized by a profound interconnection, as poets imbue their compositions with rich emotional content, sentiments, and values. The distinctive features inherent in poetry, including diction, rhyme, rhythm, and imagery, contribute to its divergence from conventional textual forms. Consequently, this research endeavors to establish a Hindi poetry corpus, specifically curated to facilitate the development of an automated system for categorizing poetry based on the identification of polarity and the emotional states encapsulated within. The corpus comprises 733 Hindi poems inscribed in the Devanagari script, meticulously annotated to reflect the emotions elicited by each. The annotation process is guided by a comprehensive questionnaire addressing polarity identification and emotional nuances embedded in the poetic expressions. Subsequently, four distinct classifiers are trained using TF-IDF word-level features extracted from the annotated corpus. Notably, the Random Forest classifier outperforms its counterparts, establishing its superiority in effectively discerning and categorizing emotional states within Hindi poetry. These endeavors contribute valuable insights to the research landscape, establishing a robust foundation for the automated categorization of poetry based on emotional attributes. The efficacy of the Random Forest classifier, as demonstrated in this study, provides a promising baseline for subsequent investigations in this domain.

REFERENCES

[1] Vilares, D., Alonso, M. A., & Gómez-Rodríguez, C. (2017). Supervised sentiment analysis in multilingual environments. Information Processing & Management, 53(3), 595-607.

[2] Mujahid, M.; Lee, E.; Rustam, F.; Washington, P.B.; Ullah, S.; Reshi, A.A.; Ashraf, I. Sentiment Analysis and Topic Modeling on Tweets about Online Education during COVID-19. Appl. Sci. 2021, 11, 8438. https://doi.org/10.3390/app11188438.

[3] Balahur, A., & Turchi, M. (2014). Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. Computer Speech & Language, 28(1), 56-75.

[4] Kim, E. (2006). Reasons and motivations for code-mixing and code-switching. Issues in EFL, 4(1), 43-61.

[5] Singh, Vinay, Aman Varshney, Syed Sarfaraz Akhtar, Deepanshu Vijay, and Manish Shrivastava. (2018)" Aggression detection on social media text using deep neural networks." Proceedings of the 2nd Workshop on Abusive Language Online (ALW2): 43-50.

[6] Baroi, S. J., Singh, N., Das, R., & Singh, T. D. (2020, December). NITS-Hinglish-SentiMix at SemEval-2020 Task 9: Sentiment Analysis for Code-Mixed social media Text Using an Ensemble Model. In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 1298-1303).

[7] Si, S., Datta, A., Banerjee, S., & Naskar, S. K. (2019, July). Aggression detection on multilingual social media text. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[8] Wu, Q., Wang, P., & Huang, C. (2020). MeisterMorxrc at SemEval2020 Task 9: Fine-tune bert and multitask learning for sentiment analysis of code-mixed tweets. arXiv preprint arXiv:2101.03028.

[9] Bhange, M., & Kasliwal, N. (2020). HinglishNLP: Fine-tuned Language Models for Hinglish Sentiment Detection. arXiv preprint arXiv:2008.09820.

[10] Parikh, A., Bisht, A. S., & Majumder, P. (2020, December). IRLab_DAIICT at SemEval-2020 Task 9: Machine Learning and Deep Learning Methods for Sentiment Analysis of Code-Mixed Tweets. In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 1265-1269).

[11] Kumar, V., Pasari, S., Patil, V. P., & Seniaray, S. (2020, July). Machine Learning based Language Modelling of Code-Switched Data. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 552-557). IEEE.

[12] Dahiya, A., Battan, N., Shrivastava, M., & Sharma, D. M. (2019, August). Curriculum Learning Strategies for Hindi-English Code-

Mixed Sentiment Analysis. In International Joint Conference on Artificial Intelligence (pp. 177-189). Springer, Cham.

[13] Singh, P., & Lefever, E. (2020, May). Sentiment Analysis for Hinglish Code-mixed Tweets by means of Cross-lingual Word Embedding‟s. In Proceedings of the The 4th Workshop on Computational Approaches to Code Switching (pp. 45-51).

[14] R. D Endsuy, "Sentiment Analysis between VADER and EDA for the US Presidential Election 2020 on Twitter Datasets", Journal of Applied Data Sciences 2 (2021) 8.

[15] M. Bibi, W. Aziz, M. Almaraashi, I. H. Khan, M. S. A. Nadeem & N. Habib, "A Cooperative Binary-Clustering Framework Based on Majority Voting for Twitter Sentiment Analysis", IEEE Access 8 (2020) 68580.

[16] R. Cekik & S. Telceken, "A New Classification Method Based on Rough Sets Theory", Soft Computing 6 (2018) 1881.

[17] A. Jain & V. Jain, "Sentiment Classification Using Hybrid Feature Selection and Ensemble Classifier" Journal of Intelligent & Fuzzy Systems, 4(2021) 221.

[18] A. P. Rodrigues & N. N. Chiplunkar, "A New Big Data Approach for Topic Classification and Sentiment Analysis of Twitter Data", Evolutionary Intelligence 2 (2019)11.

[19] S. Rani, N. S. Gill & P. Gulia, "Survey of Tools and Techniques for Sentiment Analysis of Social Networking Data", International journal of Advanced computer Science and applications 12 (2021) 222.

[20] R. Cekik & A. K. Uysal, "A novel filter feature selection method using rough set for short text data", Expert Systems with Applications 160 (2020) 113691

[21] Chandra R, Krishna A (2021) Covid-19 sentiment analysis via deep learning during the rise of novel cases. Plos one 16(8):e0255615

[22] Lwin MO, Lu J, Sheldenkar A, Schulz PJ, Shin W, Gupta R, Yang Y (2020) Global sentiments surrounding the covid-19 pandemic on twitter: analysis of twitter trends. JMIR Public Health and Surveillance 6(2):e19447

[23] Prabhakar Kaila D, Prasad DrAV et al (2020) Informational flow on twitter–corona virus outbreak– topic modelling approach. International Journal of Advanced Research in Engineering and Technology (IJARET) 11:3

[24] Nemes L, Kiss A (2021) Social media sentiment analysis based on covid-19. J Inform Telecommun 5(1):1–15

[25] Samuel J, Ali GG, Rahman M, Esawi E, Samuel Y et al (2020) Covid-19 public sentiment insights and machine learning for tweets classification. Information 11(6):314

[26] Mittal N, Agarwal B, Chouhan G, Bania N, Pareek P (2013) Sentiment analysis of hindi reviews based on negation and discourse relation. In: Proceedings of the 11th workshop on Asian language resources, pp 45–50

[27] Gupta V, Jain N, Shubham S, Madan A, Chaudhary A, Xin Q (2021) Toward integrated cnn-based sentiment analysis of tweets for scarce-resource language-hindi. Transactions on Asian and Low-Resource Language Information Processing 20(5):1–23

[28] A. M. G. Almeida, R. Cerri, E. C. Paraiso, R. G. Mantovani, and S. Barbon Junior, "Applying multi-label techniques in emotion identification of short texts," Neurocomputing, vol. 320, no. 3, pp. 35–46, 2018.

[29] H. Liu, M. Shen, J. Zhu, N. Niu, and L. Zhang, "Deep learning based program generation from requirements text: are we there yet?," IEEE Transactions on Software Engineering, vol. 48, no. 4, pp. 1268–1289, 2020.

[30] W. Zeng, H. Xu, H. Li, and X. Li, "Research on the methodology of correlation analysis of sci-tech literature based on deep learning technology in the big data," Journal of Database Management, vol. 29, no. 3, pp. 67–88, 2018.

[31] C. N. Dang, M. N. Moreno-García, and F. Prieta, "Hybrid deep learning models for sentiment analysis," Complexity, vol. 2021, Article ID 9986920, 16 pages, 2021.