Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Hybrid Model for Deep Learning- Machine Learning of Hindi Sentiment Poetic Analysis with a Metaheuristic Optimization Algorithm

Hariom[1], Avinash Kumar Sharma[1], Ashwani Kumar[2], Mohit Kumar[3]
[1]1Department of CSE, Sharda School of Computing Science & Engineering (SSCSE), Sharda University, Greater Noida, India. Email: hariomsoniiimtgn@gmail.com, avinashsharma2006@gmail.com
2School of Computer Science Engineering & Technology, Bennett University, Greater Noida, UP India
Email: ashwani.kumarcse@gmail.com
3Department of Computer Science and Engineering, Amity University, Ranchi, Jharkhand, India, pin- 834001
Email: mohitsmailbox13@gmail.com
*Corresponding Author: Avinash Kumar Sharma Email: avinashsharma2006@gmail.com

ARTICLE INFO

Received: 22 Dec 2024

Revised: 07 Feb 2025

Accepted: 18 Feb 2025

ABSTRACT

An essential function of natural language processing is sentiment analysis. Which holds substantial significance in understanding public opinion across diverse domains. However, while sentiment analysis methodologies abound in English, there exists a notable scarcity of research addressing sentiment analysis in languages like Hindi. In response, the above paper provides a pioneering aspect to Hindi sentiment analysis through the development of a hybrid deep learning-machine learning model integrated with a metaheuristic optimization algorithm. By amalgamating the strengths for normal machine learning (ML) techniques and deep learning (DL), this model endeavours to boost accuracy and robustness in sentiment classification tasks specific to Hindi text. Furthermore, the inclusion of a metaheuristic optimization algorithm aims to optimize crucial model parameters, thereby improving convergence speed and overall performance. The proposed approach is motivated by the need for more comprehensive sentiment analysis techniques tailored for multilingual social media data, particularly in languages like Hindi, which are prevalent on various online platforms. Through empirical evaluation and comparative analysis, this paper demonstrates the efficacy and potential applications of the proposed hybrid model in real-world sentiment analysis scenarios. This research contributes to bridging the gap in sentiment analysis research for non-English languages and lays the foundation for further advancements in multilingual sentiment analysis methodologies.

Keywords: CNN-LSTM multi-feature fusion, Hindi poetry-based text sentiment analysis, Natural language processing, Grey wolf optimization.

INTRODUCTION

Sentiment analysis is an essential activity in natural language processing (NLP) that is critical to comprehending the sentiment of the public towards different entities, goods, or events. While sentiment analysis has gained considerable attention in English, there is a noticeable gap in the literature concerning sentiment analysis in languages other than English, particularly for languages like Hindi. Hindi, being one of the most spoken languages globally, holds significant importance for sentiment analysis, especially in the context of social media platforms where multilingual content is prevalent. The instantaneous advancement in computing technology has led to notable strides in analyzing and processing monolingual text collections through diverse Natural Language Processing (NLP) methodologies. Situated within the range domain of computational intelligence (AI), NLP involves the examination of linguistic components, including sentences structured according to grammatical rules. Although code-switching and code-mixing are frequently conflated, they denote distinct phenomena within natural language data. Computational techniques play a pivotal role

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

in both human and machine interaction [1]. From 1-word frequency analysis to more intricate tasks like comprehending complete human expressions [2]. As a result, code switching takes place between sentences, while code mixing unfolds within a given language's framework, integrating elements from another language into the discourse. A detailed exploration of the factors influencing code mixing is provided in [3]. Bilingualism, along with factors such as etymological circumstances of speakers and their casual cohorts, contexts, lexis accessibility, and language status, taken as a whole, affect the frequency of code mixing on social networking sites. In [3, 4], the reasons and motives behind code mixing and code switching are discussed in further detail. In response to this gap, This study report offers a fresh perspective. to Hindi sentiment analysis by proposing a Hybrid DL -ML model coupled with a metaheuristic optimization algorithm. The integration between conventional statistical learning methods and DL aims to leverage the strengths of both paradigms, offering enhanced accuracy and robustness in sentiment classification tasks. Additionally, the incorporation of a metaheuristic optimization algorithm further enhances the model's performance by optimizing key parameters and improving convergence speed.

The importance of this study is in its capacity to further the development of sentiment analysis techniques for languages like Hindi, thereby enabling a more comprehensive understanding and analysis of sentiment in multilingual social media data. Furthermore, the proposed hybrid model with metaheuristic optimization holds promise for application in various domains, including marketing analytics, public opinion monitoring, and social media sentiment tracking, facilitating informed decision-making and strategic planning. Through empirical evaluation and comparative analysis, this paper aims to demonstrate the effectiveness and applicability of the proposed approach in real-world sentiment analysis scenarios.

LITERATURE REVIEW

The goal of Sasidhar, T. T. et al. [5] was to create an annotated dataset and categorise emotions in code-mixed Hindi-English text taken from Instagram comments, Facebook posts, and tweets. Using CNN-BiLSTM, they were able to attain an accuracy of 83.21%. Utilising a BiLSTM model, Kumar & Dhar [6] performed sentiment analysis on code-mixed Hindi-English text taken from Facebook postings, attaining an accuracy of 83.54% and an F1-score of 0.827. Language identification of Hindi-English code-mixed data from Facebook posts, tweets, and WhatsApp chats was the main focus of Veena et al.'s study [7]. Using an SVM technique, they were able to obtain a range of f-scores (the highest being 98.70%) on various datasets. Using both SVM and Random Forest (RF) classifiers, Vijay, Deepanshu, et al. [8] tackled the problem of sarcasm identification in Hindi-English code-mixed tweets, getting F1 scores of0.77 and 0.72, respectively. Wu, Wang & Huang [9] conducted sentiment analysis on Hindi-English and Spanish-English code-mixed tweets using a BiLSTM model, achieving an F1-score of 0.730. Raha, Tathagata, et al. [10] focused on segment of speech (POS) tagging in Bengali-English code-mixed tweets using an LSTM model, attaining an accuracy of 75.29%. Pratapa, A et al. [11] performed POS tagging and sentiment analysis on Hindi-English code-mixed tweets using an LSTM model, achieving an F1-score of 0.56. Prabhu, Ameya, et al. [12] created a corpus and conducted sentiment analysis on Hindi-English code-mixed Facebook posts using an LSTM model, achieving an accuracy of 69.7%. Gopal & Das [13] performed sentiment analysis on Hindi-English code-mixed Facebook posts using an ensemble approach combining LSTM and Multinomial Naive Bayes (MNB), achieving an accuracy of 70.8% and an F1-score of 0.661. The Proportional Rough Feature Selector (PRFS) is a filter-based technique for selecting features, using clumsy set theory. It improves the performance of various classifiers, such as SVM, decision trees, KNN, and Naive Bayes, with a confidence level of 95%. [16] Jain et al. [17] developed a method for reducing feature sets in sentiment analysis using the Apriori algorithm and a feature selection technique based on association rule mining. Their experiments involved supervised classification methods, including Naive Bayes, random forests, logistic regression, and support vector machines. Rodrigues et al. [18 created a method that uses pattern analysis to identify aspects and analyze sentiment. They extract specific syntactic patterns from product reviews and determine the sentiment polarity of sentences using Senti-Wordnet and bigram features. Their study shows that a multi-node clustering approach performs better than a single-node approach. Using Native datasets from Twitter, the authors developed a Cooperative Binary-Clustering Framework specifically for sentiment analysis. They further split the clusters into positive and negative groups by applying the confusion matrix. Word polarity is used in feature selection, TF-IDF, and unigram techniques [14,15]. Learning methodologies how the proposed strategies expedite text pattern analysis and offer avenues for automating sentiment analysis [19, 20]. DL, capable of handling vast datasets, accelerates

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

text pattern analysis through artificial neural networks, addressing alignment issues by extracting DL algorithms that leverage word embeddings as inputs. To determine sentiment, the researchers used the Long Short-Term Memory (LSTM) building. Their research focused on the analysis of a dataset that included 150,000 tweets on COVID-19 pertaining to India that were gathered between March 2019 and September 2020. The researchers looked at COVID-19-related worldwide Twitter trends. They used 18,799 tweets utilising the sentiment lexicon of the National Research Council (NRC) to perform topic modelling and sentiment examination on COVID-19. By leveraging an architecture of Recurrent Neural Networks (RNN),the researchers Tweets' psychological content has been classified into positive and negative sentiment categories. Using Twitter data, they studied the public's views on COVID-19 and used logistic regression, linear regression, and the Naïve Bayes classifier to get a maximum accuracy of 74% [21–25]. The authors used (HSWN) to achieve an 80% classification accuracy after creating an interpreted quantity for the Hindi language. The authors evaluated a plethora of machine learning (ML) methods, like decisions trees, LR, and Naive Bayes (SVM), for sentiment analysis of Hindi tweets. [26, 27]. Kumar et al. proposed novel skin cancer detection algorithm using deep CNN [28-30]

Almeida et al. employed (CNNs) to address the task of reaction cataloguing. Recurrent Neural Networks (RNNs) demonstrate effectiveness in processing sequential data, making them frequently utilized NLP [31]. Liu et al. utilized the model countenance info inherent for emotion classification, yielding promising outcomes [32]. Zeng et al. introduced an algorithm to tackle emotional tendency issues. This approach incorporates a bootstrapping strategy, with the emotional tendency being determined [33]. Dang et al. proposed goal dependency approach, which considers the contextual influence on Weibo's emotional content. Implementation methods include goal dependency and situational awareness. Goal dependency involves emotion assessment based on syntactic features, while situational awareness entails classification considering related tweets for each post [34].

METHODOLOGY

Here I describe the most important aspect of set up for how I process the raw data where I collect form the raw twitter data 25000 tweets about poetry then I processed them to these basic step word processing I described the four important aspect as follows. Bag-of-Words: This method is like counting up the number of "happy" words versus "sad" words. For example, if a tweet has more words like "great," "awesome," and "love," it might be classified as positive. However, this method can be limited because it doesn't consider the context of the words or the relationships between them. This creates a high-dimensional feature space where documents are represented as sparse vectors. TF-IDF: This method is like giving more points to words that are common in the tweet but rare in most tweets. For example, if a tweet uses the word "fantastic" multiple times, it might be given a higher weight because it's a less common word. This helps to identify words that are more indicative of sentiment.

N-grams: This method looks at not just single words, but also pairs or groups of words to understand the context better. For example, the phrase "not so bad" might be classified as positive, even though the word "bad" is negative. N-grams can capture the nuances of language and improve the accuracy of sentiment analysis. Word Embedding: This method is like giving each word a personality based on how it's used in other texts. Word embedding represent words as dense vectors in a high-dimensional space, capturing semantic relationships between words. This allows the model to understand the meaning of words and how they relate to each other, which can improve the accuracy of sentiment analysis. After all these basic data pre-processing I followed standard parameter of sentiment +1 for positive, 0 for negative and +0.5 for neutral for single tweet. For computational efficiency CNN use for Short Pattern and LSTM use for long pattern in strings that used in data set.The proposed methodology is validated using a Hindi poetry sentiment corpus. The study introduces a CNN-LSTM model with attention mechanism (AM) for poetic aesthetic implication analysis, addressing challenges in semantic and emotional information preservation followed by self-attention weighting and summation of the model's output, leading to sentiment classification using a SoftMax classifier. The proposed solution outperforms the other structure in terms of overall performance metrics, according to comparative trials. Based on processed vectors that are extracted from the model, the SoftMax classifier—which has Utilising a categorical cross-entropy loss function for sentiment classification.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

To handle sorting challenges, this research combines a variety of ML and DL strategies. Multinomial Naive Bayes, tailored for text classification, is utilized to classify data. Random Forest, an ensemble learning algorithm, is employed to enhance predictive accuracy and prevent overfitting by aggregating multiple decision trees. Logistic Regression, a statistical method for binary classification, predicts the likelihood of instances belonging to specific classes The architecture of Convolutional Neural Networks (CNNs) is used to extract features and lower the dimensionality of input. Furthermore, recurrent neural networks (RNNs) with Long Short-Term Memory (LSTM) are used to identify long-term relationships in sequential data, providing superior results in sequential analysis as well as minimising the vanishing gradient issue.Text pre-processing is conducted to format and prepare the text data for analysis. The pre-processing method described in this study involves utilizing the maximum matching method for rough text segmentation, followed by part-of-speech tagging using hidden Markov models to evaluate and refine the segmentation results

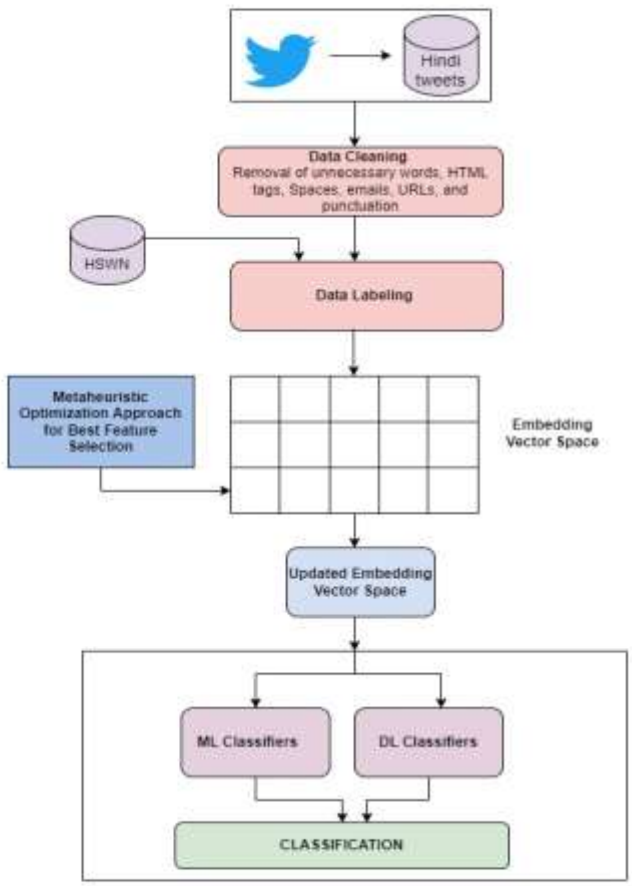


Fig 1: Proposed Framework

Algorithm 1 outlines the process for sentiment analysis of data collected from

This flowchart outlines a comprehensive process for classifying Hindi tweets. The pipeline begins with data acquisition from Twitter, followed by a thorough data cleaning phase to remove irrelevant elements such as HTML tags, unnecessary words, and punctuation. Subsequently, the cleaned data is subjected to a meticulous labelling process, assigning appropriate categories to each tweet Fig [1]. HindiSentiwords.net. Initially, tweets are scraped using the Twint library, and the resulting data is stored in a data frame. Data cleaning and pre-processing are then performed on the data frame to prepare it for analysis. This includes removing null values, stop words, mentions, URLs, emoticons, and punctuation, as well as tokenization. Subsequently, the polarity score of each tweet is calculated using a designated function, and

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

sentiments are assigned based on these scores. Each sentence is then processed to calculate the index value of each word. Following this, a fitness function is applied to update the weight matrix, and the optimized weight matrix is obtained using The GWO algorithm, or Grey Wolf Optimisation. Lastly, the optimised weight matrix is used to train the suggested hybrid DL model, which then analyses the dataset for sentiment.

Algorithm 1 Sentiment Analysis of HindiSentiwords.net

1. **Utilize** Twint library for harvesting tweets.
2. **Save** the gathered dataset.
3. **Conduct** data refining and preprocessing:
   a. *Eliminate null values from the dataset.*
   b. *Strip off stop words, mentions, and URLs.*
   c. *Purge emoticons and punctuations.*
   d. *Segment into tokens.*
4. **Determine** the tweets in the dataset's polarity score and store it in ZA["qt"].
5. **Associate** sentiment with content according to polarity score.
6. **Iterate** over each element ZA[a], where a ranges from 0 to n:
   a. If ZA["qt"] score is greater than 0, label as "Positive".
   b. If ZA["qt"] score < 0, label as "Negative".
   c. Otherwise, label as "Neutral".
7. Loop through each ZA["Sentences"].
8. **Employ** GWO for optimizing the weight matrix.
9. **Train** the weight matrix.

From Fig [2] ,The pre-processing pipeline for Hindi text involves several sequential steps to prepare the data for further analysis. Initially, the raw Hindi text undergoes a data-cleaning process where various cleaning operations are applied to remove irrelevant or redundant information. This includes removing any stop words, which are commonly occurring words that typically do not carry significant meaning in the context of the analysis. Following stop word removal, the text is tokenized, breaking it down into individual words or tokens to facilitate further processing. Once tokenized, the data is labelled, assigning appropriate labels or categories to each piece of text based on the intended analysis, such as sentiment analysis or topic classification. Finally, the labelled text is vectorised, converting it into numerical representations suitable for input into ML algorithms. Overall, this pre-processing pipeline ensures that the Hindi text is properly cleaned, organized, and transformed into a format conducive to subsequent analysis tasks.

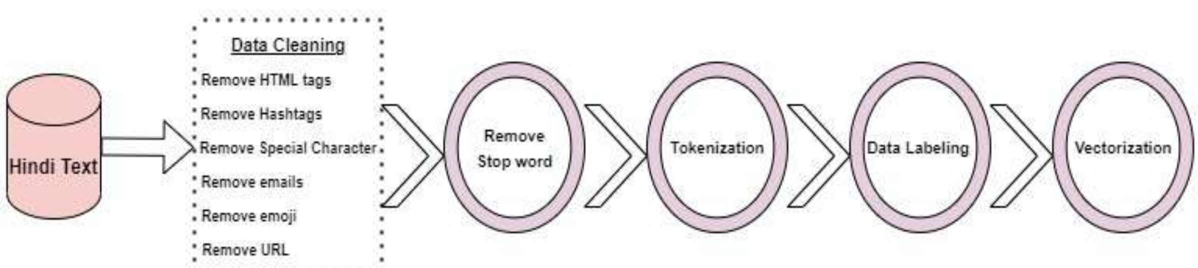


Fig 2: Pre-processing steps

Initially, the maximum iteration limit (Z) is set, along with the population of wolves (Bv). Parameters d, E, and G are adjusted accordingly. The fitness levels of the wolves are then regulated, with the most effective search agents identified as M(α), the second most effective as M(β), and the third most effective as M(δ). Subsequently, the algorithm enters a loop where the wolves are repositioned iteratively. Each search agent's position is updated based on certain rules or strategies. Positions of the most effective search agents (α, β, δ) are updated as well. This process continues until the maximum iteration limit (Z) is reached. Finally, the algorithm returns the position of the most effective search agent B(α), which represents the solution obtained by the GWO algorithm for the optimization problem at hand.

Algorithm 2: Algorithm of GWO procedure.

1. **Put** Z as max. steps.
2. **Put** populace $cu(u = 1,2 \dots q)$.
3. **Adjust** G, E, & d.
4. **Regulate** the wolves' aptness level.
5. $M(\alpha)$ = Most Valuable search employ.
6. $M(\beta)$ = Second effective search employ.
7. $X(\delta)$ = Third effective search agent.
8. **while** $Y < Z$ **do**
9. **for** each search agent **do**
10. resituating the active search employ.
11. **end for**
12. Apprise the value of G, E, and d.
13. **Regulate** the fitness level of all search agents.
14. **Apprise** the value of $(\alpha), (\beta), and\ (\delta)$.
15. $Y = Y + 1$
16. **end while**
17. **return** $B(\alpha)$

Hierarchical Structure in Grey Wolf Optimization: *The Grey Wolf algorithm operates with a hierarchical structure mirroring the leadership dynamics within a wolf pack. Within this framework, Alpha (α) symbolizes the pinnacle of intellectual dominance, while Beta (β) represents the subsequent levels of leadership importance. Throughout the optimization process, the objective is to fine-tune the calibration of vectors E and G, with a focus on both exploitation and exploration across various dimensions. Ultimately, the GWO algorithm yields an optimized weight matrix, which serves as input for the classification model, enabling enhanced performance in solving optimization problems.*

RESULT & DISCUSSION

Table 1 provides a comparative analysis of various proposed models based on their performance metrics including Precision, F1-Score, Recall, and Accuracy. Random Forest achieved a Precision of 90.22%, F1-Score of 89.06%, Recall of 91.47%, and Accuracy of 87.75%. Logistic Regression yielded a Precision of 88.09%, F1-Score of 86.31%, Recall of 85.53%, and Accuracy of 89.01%. Naive Bayes exhibited a Precision of 91.39%, F1-Score of 90.86%, Recall of 89.43%, and Accuracy of 94.55%. CNN attained a Precision of 87.35%, F1-Score of 90.22%, Recall of 91.33%, and Accuracy of 91.26%. LSTM showcased a Precision of 88.70%, F1-Score of 89.79%, Recall of 90.27%, and Accuracy of 88.22%. This comparative analysis offers insights into the relative performance of each model across multiple evaluation metrics, aiding in the selection of the most suitable model for specific applications based on desired performance criteria. Fig [3] is line graph representation of table [1] of different method used.

Table 1: Comparison of proposed models.

Model	Precision (%)	F1-Score (%)	Recall (%)	Accuracy (%)
RF	90.22	89.06	91.47	87.75
LR	88.09	86.31	85.53	89.01
NB	91.39	90.86	89.43	94.55
CNN	87.35	90.22	91.33	91.26
LSTM	88.70	89.79	90.27	88.22

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

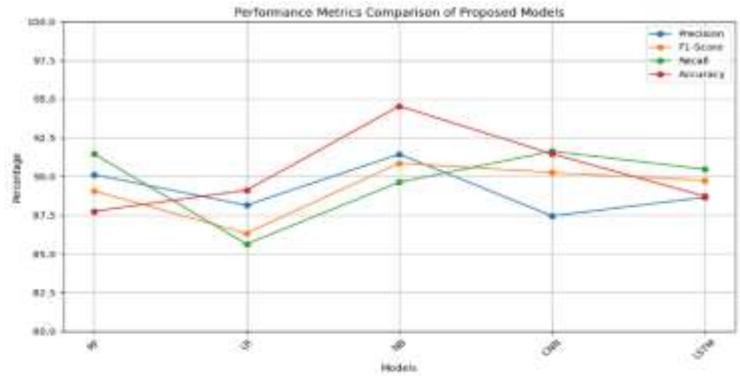
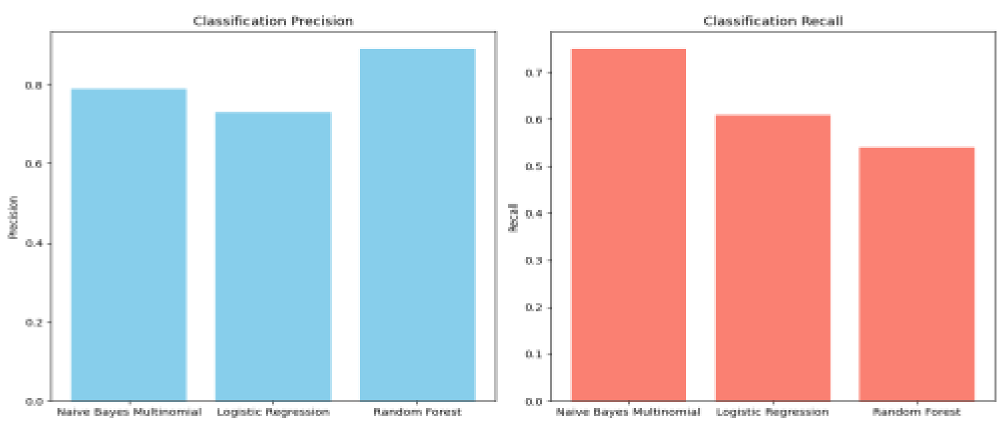


Fig.3 Performance Metrics Comparison of Proposed Models

Table 2: Evaluation of Sentiment Analysis Performance of Machine Learning Classifiers

Classifier	Classification Precision, Recall, F1 Score	Accuracy Train, Test	AUC Score	Time to train (in seconds)
NBM	0.79,0.75,0.77s	0.76,0.7633	0.7642	0.009278
LR	0.73,0.61,0.70	0.73,0.73	0.736	0.06633
RF	0.89,0.54,0.67	0.72,0.7233	0.732	0.57832

Table 2 offers a thorough evaluation of Sentiment analysis using machine learning (ML) classifiers. The Random Forest (RF), Logistic Regression (LR), and Naive Bayes Multinomial (NBM) classifiers are evaluated. A number of metrics, including accuracy, area under the curve (AUC) score, recall, classification precision, F1 score, and training time in seconds, are used to assess each classifier's performance. The Naive Bayes Multinomial has recorded values of 0.79, 0.75, and 0.77. for classification precision, recall, and F1 score, respectively. For the training set, the accuracy is 0.7642, and for the test set, the AUC score is 0.7633 and 0.7676. The model takes 0.009278 seconds to fully train. Classification precision, recall, and F1 score are all attained via logistic regression, with respective values of 0.73, 0.61, and 0.70. With an AUC value of 0.73 for both the test and training sets, the accuracy is 0.736. Logistic Regression (LR) takes 0.06633 seconds to train.



Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

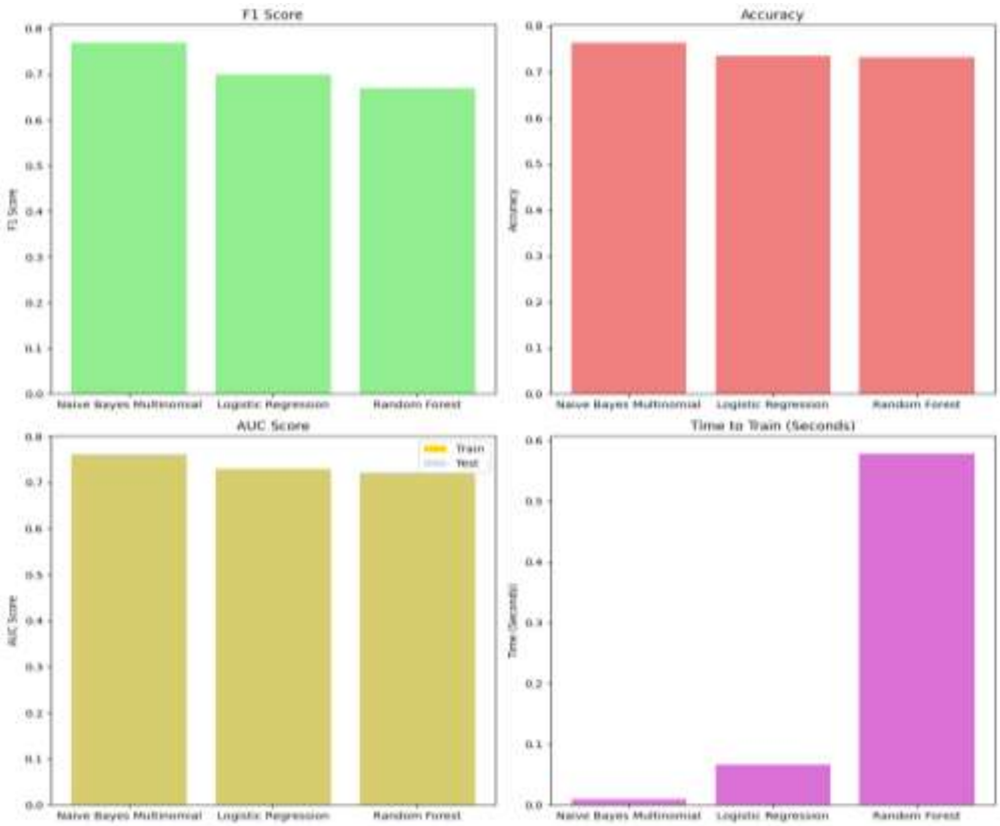


Fig.4 Functioning Examination of ML Classifiers for Sentiment Analysis

Lastly, Random Forest exhibits a higher classification precision of 0.89 but lower recall and F1 score of 00.54 and 00.67 relatively. The accuracy is 0.732, with an AUC score of 0.72 for both training and test sets. However, Random Forest requires significantly more time to train compared to the other classifiers, with a training time of 0.57832 seconds. Overall, this table offers a detailed comparison of the performance of different machine learning (ML) classifiers Fig [4] for sentiment analysis, considering both predictive accuracy and computational efficiency. These insights can inform the selection of the most suitable classifier based on the specific requirements and constraints of the sentiment analysis task at hand.

Table 3: Performance Analysis of Deep Learning Techniques for Sentiment Analysis

Classifier	Classification	Accuracy	AUC Score
CNN	0.83,0.79,0.81	0.98,0.845	0.84
LSTM	0.80,0.67,0.73	0.96,0.77	0.748

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

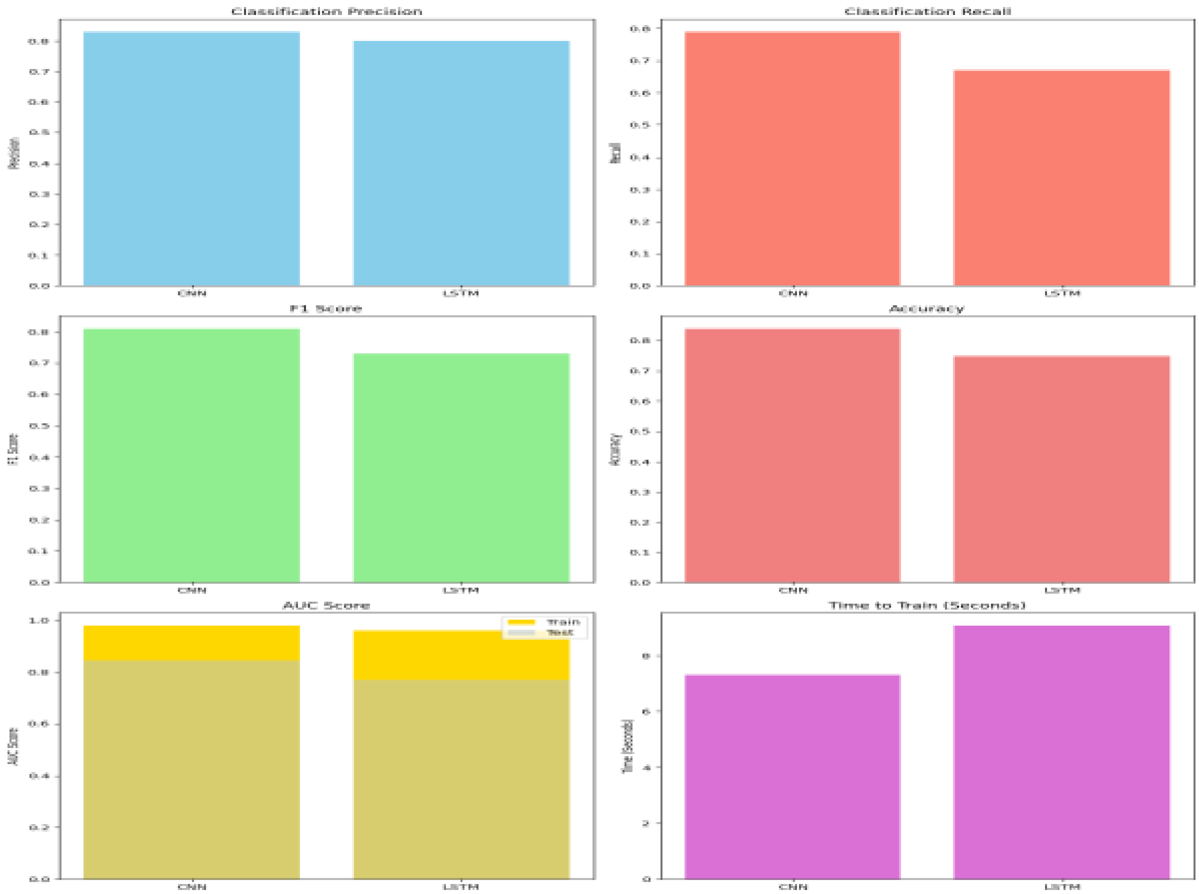


Fig.5 Functioning Examination of DL Procedures for Sentiment Analysis

Table 3 presents an analysis of performance deep learning (DL) techniques, specifically Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), for sentiment analysis. For each classifier, the table reports classification precision, recall, and F1 score, as well as accuracy, area under the curve (AUC) score, and time to train in seconds. The CNN classifier achieves a classification precision, recall, and F1 score of 0.83, 0.79, and 0.81, respectively. The accuracy is reported as 0.84, with an AUC score of 0.98 for the training set and 0.845 for the test set.

In contrast, the LSTM classifier achieves slightly lower values of 0.80, 0.67, and 0.73, respectively. The accuracy is reported as 0.748, with an AUC score of 0.96 for the training set and 0.77 for the test set. LSTM requires a longer training time compared to CNN, with a time to train of 9.08 seconds. Overall, the table provides insights into the performance of DL techniques for sentiment analysis, highlighting the strengths and weaknesses of CNN and LSTM classifiers in standings of predictive accuracy & computational efficiency. These findings can inform the selection of appropriate deep-learning (DL) models for sentiment analysis tasks based on specific requirements and constraints as shown in Fig[5].

Table 4: Contrasting outcomes of diverse models.

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

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

Fig.6 Contrasting outcomes of diverse models.

Table 4 bestows a comparison of diverse models based on their performance metrics, including accuracy, recall, and F1 score. The CNN model achieves an accuracy of 84.282%, a recall of 89.37%, and an F1 score of 87.865%. This indicates that the CNN model correctly identifies 84.282% of the instances and has a good balance between recall & precision. Conversely, though, the LSTM model exhibits higher accuracy, with a value of 87.33%. It also demonstrates a slightly higher recall of 90.28% matching to the CNN blueprint. However, the F1 score for LSTM is slightly lower at 85.985%.

The CNN-LSTM hybrid model achieves an impressive accuracy of 93.362%, indicating its ability to correctly classify a vast majority of instances. Additionally, the recall and F1 scores for the hybrid model are 91.147% and 90.886%, respectively, demonstrating its effectiveness in correctly identifying positive instances while maintaining a good balance between precision and recall. Overall, the comparison results highlight the strengths of each model and provide valuable insights into their performance in the context of the specific task or dataset under consideration as shown in Fig [6].

Table 5: Comparison of average training time of different models.

MODEL	RUN-TIME
CNN	1369.71
LSTM	1553.28
CNN-LSTM HYBRID	1025.14

Table 5 compares the average training time (in seconds) of different models, including CNN, LSTM, and CNN-LSTM hybrid. The CNN model has an average running time of 1369.71 seconds, while the LSTM model takes slightly longer

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

with an average running time of 1553.28 seconds. On the other hand, the CNN-LSTM hybrid model demonstrates the shortest average running time among the three models, with a value of 1025.14 seconds. The running time comparison provides insights into the computational efficiency of each model as shown in Fig [7].

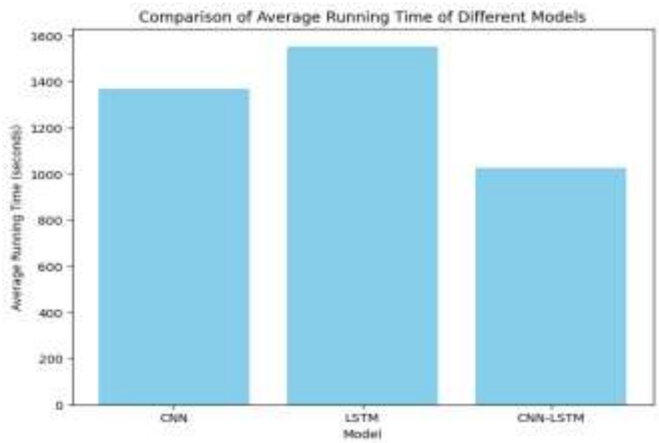


Fig.7 Comparison of Average Running Time of Different Models

While the LSTM model takes the longest to train, the CNN-LSTM hybrid model shows promise as a more time-efficient alternative. This information is crucial for decision-making processes, especially when considering resource constraints or time-sensitive applications.

Table 6: Computational complexity of the proposed model

Stages	Time Complexity
Data-cleaning process	O(Posts × Word count)
Feature Finding	T (f) = O(f 2) + parsing time
Features labelling with GWO	O(W × q × MI)
Evolution of sentiments	O(b × s(ac + cx + xy))
Forecasting	O(1)

Table 6 presents the computational complexity associated with various steps in the proposed model. The first step involves data cleaning, where the time complexity is represented as O(Twitter posts × total word count). This indicates that the complexity depends on the number of posts collected from Twitter and the total word count within those posts. Next is feature extraction, which has a time complexity denoted as O(f^2) + parsing time = T(f). In this case, 'f' stands for the number of features, and the square of the feature count and the parsing time needed determines the complexity. The Grey Wolf Optimisation (GWO) algorithm is used for feature selection, which comes after feature extraction. For this phase, the temporal complexity is expressed as O(W × q × MI), where 'W' is the word count, 'q' is the class count, and 'MI' is the mutual information.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

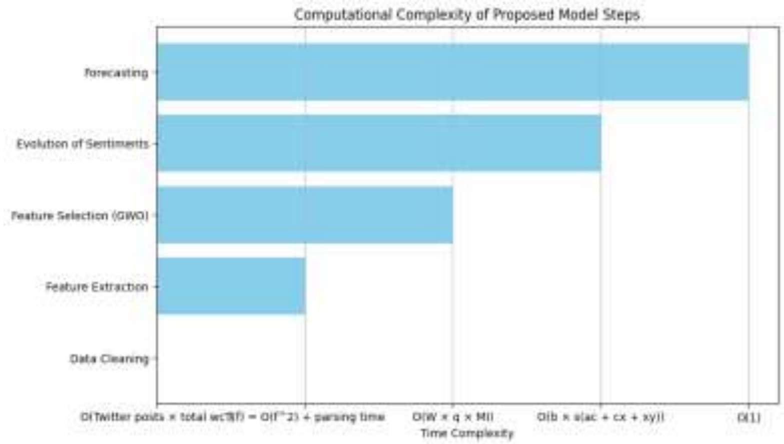


Fig.8 Computational Complexity of Proposed Model Procedure

The evolution of sentiments, which likely involves sentiment analysis or classification, has a time complexity of O(b × s(ac + cx + xy)), where 'b' represents the number of sentiment bins, 's' denotes the number of samples, and 'ac', 'cx', and 'xy' are coefficients corresponding to various operations. Finally, the forecasting step has a constant time complexity denoted as O(1),indicating that it does not depend on the size of the input data but instead executes in constant time. Overall, Table [6] helps to comprehend the overall computational complexity Fig [8] of the strategy by providing insights into the computing needs of each step in the suggested model. This paper is called hybrid because of we use both ML and DL model to computing the sentiment that is we use in feature selection in the paper and in table [4] the hybrid result is shown.

CONCLUSION

This study proposes a novel technique that combines ML with multilingual code-mixed text to analyse sentiment and DL approaches. The study meticulously explores countless representations and methodologies to analyse sentiments expressed in diverse languages across different social media platforms. The study starts with a thorough overview of the body of literature, offering insights into the most recent techniques and difficulties in sentiment analysis, especially when dealing with multilingual code-mixed text. Building on this basis, a number of experiments are created and put into practice to assess the efficacy of various models and methods. Through experimentation, promising results are observed with several models, counting RF, CNN, LSTM, and Multinomial Naive Bayes. Each model exhibits varying levels of F1 score, recall, accuracy, & precision, highlighting their strengths & limitations in handling multilingual sentiment analysis tasks. Furthermore, innovative approaches such as feature selection using Grey Wolf Optimization and ensemble hybrid models are introduced to enhance the performance of sentiment analysis systems. The above methods exhibit increases in correctness and effectiveness, underlining their potential for practical uses. All things considered, this study advances sentiment analysis strategies, especially in the problematic field of multilingual code-mixed text. By leveraging an amalgamation of DL-ML techniques, along with innovative feature selection and ensemble modelling strategies, The foundation is established for more reliable and accurate sentiment analysis algorithms that can process social media data from a variety of language situations. These outcomes show the exceptional performance of the hybrid DL model in accurately classifying sentiments in Hindi tweets, showcasing its efficacy in sentiment analysis applications. The proposed hybrid model, which unifies dl, ml, and metaheuristic techniques. The Real World Application of this paper is to learn more similar 66 language across the country. And can expand in to more literature like poetry.

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.

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[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] Kumar A, Kumar M, Bhardwaj VP, Kumar S, Selvarajan S. A novel skin cancer detection model using modified finch deep CNN classifier model. Sci Rep. 2024 May 16;14(1):11235. doi: 10.1038/s41598-024-60954-2. PMID: 38755202; PMCID: PMC11099129.

[30] Kumar M, Kumar A, Kumar S, Chauhan P, Selvarajan S. An African vulture optimization algorithm based energy efficient clustering scheme in wireless sensor networks. Sci Rep. 2024 Dec 28;14(1):31412. doi: 10.1038/s41598-024-83005-2. PMID: 39733037; PMCID: PMC11682067.

[31] Kumar A, Kumar M, Mahapatra RP, Bhattacharya P, Le TT, Verma S, Kavita, Mohiuddin K. Flamingo-Optimization-Based Deep Convolutional Neural Network for IoT-Based Arrhythmia Classification. Sensors (Basel). 2023 Apr 28;23(9):4353. doi: 10.3390/s23094353. PMID: 37177564; PMCID: PMC10181507.

[32] 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.

[33] 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.

[34] 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.