An Effective Sentiment Analysis in Hindi-English Code-Mixed Twitter Data using Swea Clustering and Hybrid BLSTM-CNN Classification

Abhishek Kori[1*], Jigyasu Dubey[2]

[1*] *Research School, Information Technology, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, Madhya Pradesh 453111, India.*
[2] *Professor, Head Information Technology Department, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, Madhya Pradesh 453111, India.*
[*]*Email: abhishekkoriphd05@gmail.com*

Abstract

Sentiment Analysis is the process of examining the individual's emotions. In tweet sentiment analysis, opinions in messages are categorized into positive, negative and neutral categories. A clustering-based classification approach is used to increase the accuracy level and enhance the performance in sentiment classification. The input dataset comprises of Hindi-English code-mixed text data. Initially, the input text data is pre-processed with different pre-processing techniques such as stop word removal, tokenization, Stemming, lemmatization. This effectively pre-processes the data and makes it appropriate for further processing. Afterwards, effective features such as Count Vectors, Modified term frequency-inverse document frequency (MTF-IDF), Feature hashing, Glove feature and Word2vector features are extracted for enhancing the classification performance. Afterwards, Sentiment word embedding-based agglomerative (SWEA) clustering is presented for effective sentiment feature clustering. Finally, a hybrid Bidirectional long short-term memory-convolutional neural network (Hybrid BLSTM-CNN) is used to accurately classify tweet sentiments into positive, negative, and neutral. Here, modified horse herd optimization (MHHO) approach is used for weight optimization in Hybrid BLSTM-CNN. This optimization approach further enhances the performance of classification. The dataset used for the implementation is a Hindi-English mixed dataset. The experimental result significantly improves the different existing approaches in terms of accuracy, precision, recall, and F-measure.

Keywords: Sentiment Analysis, Hindi-English, Twitter, code-mixed text data, modified horse herd optimization

1. Introduction

Over the last two decades and the increase in the population, social media users have also increased tremendously [1]. Thelanguages are mixed in several forms of communication due to different cultures. People started to communicate online to share and express their thoughtson social network websites like Twitter, YouTube, and Facebook. These media are a better platform to interact with each other [2]. Most of the words are used generally in one language, and the translation in another language is not very popular. Hence, when the person utilizes those words in a text, they are like the most fashionable language [3]. This makes a sentence in two different languages and arranges a grammar part of one language.The text in social media isfamiliar with many linguistic variations. In multilingual countries like India, commonly people combine the English and Hindi languages with their native language [4]. The method of switching sentences

between the many languages is known as code-mixing, also called code switching [5]. It is amodification phenomenon among many languages, generally two, within a single sentence [6].

Sentiment Analysis (SA) in mixed languagehas gained popularity due to the increasing number of non-English speaking users [7]. SA can offer precise insight from product reviews to capture trending themes to design business models. Today, most institutions depend on SA of social media text to monitor the performance of their products and fetch feedback [8]. SA is used in product reviews and applications like reputation management, social media monitoring, and brand monitoring [9].In addition, people post their suggestions, opinion and results in a huge amount of text information accessible for analysis. Humans can understand the paragraph written in a language they know and identify the paragraph as positive, negative, or neutral feelings [10]. However, the computer doesn't know the languages around the world, and it cannot interpret them. SA overcomes this limitation of the computer by Natural Programming Language (NLP) for recognizing the word and classifying them [11]. A mixed language tweet has many unseen complexities to NLP tasks like language identification, semantic processing, machine translation and Parts-of-Speech (POS) tagging.Hence it is important to develop a technology for mixed language text [12, 13].

Based on the report of KPMG Group in India, users of the Indian language can expand up to 537 million in 2022 [14]. People states their opinion on changing subjects varies from sports to politics and movies [15]. In addition, people express their suggestions in mixed languages like English-Urdu, English-Hindi, English-Tamil, and English-Bengali [16]. Hindi is a national language of India, which is majorly spoken in different states. Due to many Hindi speaking people contributes majorly in several social media about various social activities [17]. The formation of the Hindi language utilized in social networks is mixed with English and accessible in roman scripts [18].Some of the existing sentiment analysis approaches are Naive Bayes, convolutional neural network (CNN) [19], decision tree (DT), support vector machines (SVM), recurrent neural network (RNN), and RF (random forests) [20].

Motivation: The phenomenon of mixed language can learn by analyzing different applications. SA is circumstantial text analysis, finding the social sentiment for better understanding the source. SA on mixed language helps to understand the sentiment of sentences and phrases. Multilingualism is the potential of people to communicate efficiently in many languages. There are three categories of SA approaches: Machine learning (ML), Lexicon based and Deep learning (DL). The Lexicon-based models depend on the predefined rules for determining the sentence from the text. An ML technique utilizes semantic mining for identifying sentiments. ML techniques are semi-supervised, supervised and unsupervised. ML techniques require hand-crafted features extraction, which is time consuming, needs expertise and is expensive. DL models are more efficient in learning features automatically from text and achieve better results. The most commonly used methods in NLP tasks are Recurrent Neural Network (RNN), Convolutional Neural Network (Convnet), and Long Short-Term Memory Network (LSTM). The text representation and the good classification approach are necessary to improveclassification performance. The major contributions of the presented methodology are described as,

- To effectively pre-process the Hindi mixed English tweets, different pre-processing approaches like stop word removal, tokenization, stemming, and lemmatization are utilized.
- To extract the most important features from the pre-processed data, count vectors, MTF-IDF, Feature hashing, Glove feature and Word2vector features are extracted. The effective feature extraction process is a necessary one for accurate sentiment prediction.
- A Sentiment word embedding-based agglomerative clustering approach is presented to cluster the sentiment features effectively. This clustering process further improves the performance of sentiment prediction.

- To accurately classify the sentiments as positive, negative and neutral, a hybrid BLSTM-CNN framework is presented. Here, the modified horse herd optimization approach improves this classification network performance. This MHHO approach is used for updating the optimized weights in the hybrid BLSTM-CNN framework. Finally, the performance of the sentiment analysis is well improved using these combinations of approaches.

2. Related Work

Jhanwar and Das [21] proposed an ensemble method for SA of Hi-En mixed data using DL models. In this work, LSTM and Multinomial Naive Bayes (MNB) were utilized for identifying the Hi-En sentiments. The ensemble model integrated the LSTM and keywords polarity from a probabilistic method for identifying sentiments in inconsistent and sparse mixed data. Both models were averaged and weighted based on the accuracy. The experimental outcomes proved the system's performance compared with other DL models.

Sasidhar et al. [22] used the DL model CNN-BLSTMto identify the emotions expressed via Hi-En mixed languages in many social media sites. To evaluate the detection method, 12,000 CM sentences from various sources have many emotions. A bilingual model was used for generating the vector, and the DL model was used for classification. This model provided higher performance with an accuracy of 83.1%.

Singh et al. [23] presented a unique language and POS tagged database of CM Hi-En tweets. It was based on five happenings in India that led to many twitter activities. The database used in this model has two factors: it was longer than the prior annotated database and was like real-world tweets. Then the POS was trained on this database to show how this database can be utilized. This model has attained a better performance with an F-measure value of 88.6%, but the model has overfitting issue despite the usage of regularization.

Singh and Lefever, [24] proposed two stages for the SA task for Hi-En sentiments. In the first stage, baseline methods and monolingual embedding were initialized. Then, in a second stage, cross lingual embedding's for Hi-En were constructed. The transfer learning-based classifieris trained on En sentiment and implemented on code-mixed information.The task comprises three sentiments, and the experimental outcomes proved that this model improved the results in fully supervised and can utilize as a baseline for a distant base.

Garg and Kamlesh Sharma, [25] presented the model of creating a corpus for Hi-En sentiments. The method utilized for annotating the corpus into 5 categories. The inner agreement measure was computed for positive and negative tweets. This model provided a standard corpus for code switching in Hi-En. The words utilized for sarcasm and slang were also discussed in this work. This model overcame spelling which was inconsistent and misspelt words.

Nagamanjula and Pethalakshmi, [26] developed an innovative model using a logistic adaptive network that depends on a neuro fuzzy inference system (LAN2FIS) to classify sentiments on Twitter data. Here, features were chosen by using the bi-objective optimization scheme. The sentiment analysis using this approach provided enhanced performance. But the analysis was performed only in public reviews. A bigger data set is a necessity to improve the accuracy performance. The performance of the approach can be improved by utilizing a better combination of approaches.

Naresh and Venkata Krishna, [27] presented a proficient sentiment analysis methodology utilizing the machine learning approach. Here, optimization was utilized to enhance the performance ofthe machine learning technique. At first, input data was taken and pre-processed. Next, the optimized data was attained through the feature extraction process. The trained feature

data was utilized to categorize the sentiment classes through the machine learning classifier in the final stage. The attained accuracy of this approach was 89.47%. The examination performance of the methodology can be improved by using the deep learning approaches in future.

Kanika Garg and Lobiyal, [28] developed a methodology for evaluating the feature values through KL (Kullback-Leibler) divergence methodology. The features were used for finding the membership values with the neuro fuzzy and fuzzy logic methodology. The obtained accuracy of the methodology was 89.93%. In future work, an effective feature selection and classification approach was suggested to improve the performance.

3. Proposed Methodology

This work presents an effective sentiment analysis in Hindi-English mixed twitter texts. Initially, the input twitters in the combination of Hindi and English texts are collected using the Twitter data set. Afterwards, input texts are effectively pre-processed using different pre-processing techniques like tokenization, stop word removal, lemmatization and stemming. Then, effective combinations of features like count vectors, feature hashing, Glove feature vectors and modified TF-IDF are extracted from the pre-processed data. Subsequently, the extracted sentiment features are clustered using the presented SWEA clustering approach. Finally, the hybrid BLSTM-CNN approach accurately predicts sentiments into negative, positive, and neutral classes. Here, the performance of sentiment prediction is improved by updating the optimized weights through the modified horse herd optimization approach. The schematic diagram of the presented methodology is depicted in figure 1.

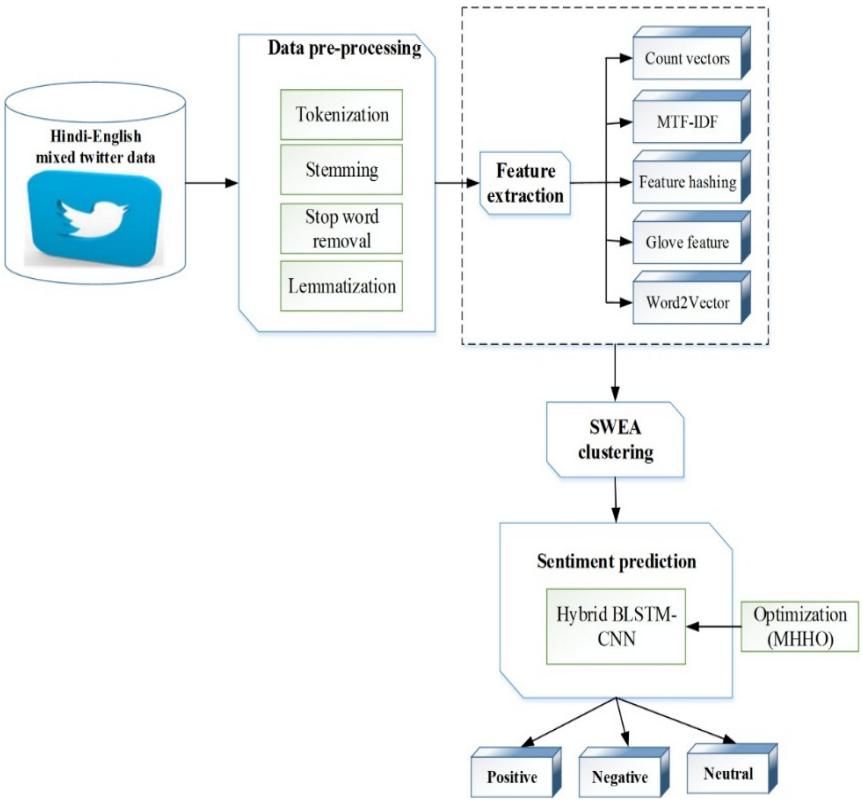

Figure 1: *Schematic diagram of the presented methodology*

The SA process has four stages: (a) Pre-processing is the first stage to clean and prepare data for sentiment analysis. It reduces computational process and feature space, enhancing the system's

accuracy. (b) Feature extraction is a second stage to extract the important features for the analysis. (c) Clustering is the third stage to group the data points and (d) Sentiment prediction is the fourth stage to classify the sentiments of data into positive, negative or neutral.

3.1 Pre-processing

Pre-processing is an important process for SA since textual information has unstructured and noisy data. The Pre-processing step is applied to the dataset to improve the quality and classify the text. This step is used to clean the dataset from noise like spelling error correction, disambiguation of ambiguous abbreviations and reduction of repetitive characters. Hence in these cases, pre-processing techniques like stop word removal, tokenization, Stemming and lemmatization can enhance the dataset's quality.

3.1.1 Tokenization

It is complex for machines to understand the context and semantics of a paragraph and sentences. Initially, the tokenization process breaks sentences into punctuations and words called tokenization. For tokenizing the words, Natural Language Toolkit (NLTK) is used.
Example: Input data- "I am playing"
Output data- (I) (am) (playing)

3.1.2 Stop word removal

In the text data processing, the words which have high existence in the documents are called stop words like "is", "am", "and", "the". These words are helpful in sentence formation but do not provide any importance in language processing. Applying these words on lexical resources has less emotional meaning and doesn't affect a sentimental score. Hence these words are filtered from the tweets. The example representation of stop word removal is provided below.
Example: Input data- I am playing
Output data- Playing

3.1.3 Stemming

It is the process of converting the tenses of words to their basic form. This stemming process avoids the processing time of words. The example of the stemming process is described as,
Example: "Playing" to "play","Arguing" to "Argue", "Fishing" to "Fish", "noises" to "noise".

3.1.4 Lemmatization

This is the process of integrating two or three words into one word. This finds the Word morphology and removes the end of the word.

Example: "Matched" to "Match", "taught" to "teach"

3.2 Text Feature extraction

The most important features like count vectors, MTF-IDF, Feature hashing, Glove feature and Word2vector features are extracted from the pre-processed data. The effective feature extraction process is a necessary one for accurate sentiment prediction. The presented feature extraction techniques are described in the subsequent sub-sections.

3.2.1 Count Vectors

Count vectors feature just count the number of word occurrences in a document. Afterwards, the count value is utilized as a weight in the feature vector. This feature extraction process counts the words and outputs in integers. The count feature vector is similar to the TF-IDF feature. In TF-IDF feature extraction, a score of the words are computed, and in the case of count vectors, floats are computed from the words. The count vectors are illustrated in the subsequent example shown in table 1.

Example data: {"Good", "Boy", "Play", "Well", "Top", "Good", "Student", "School"}

Table 1: *Example illustration of output Count Vectors*

Data	Good	Boy	Play	Well	Top	Student	School
Count vectors	2	1	1	1	1	1	1

3.2.2 Feature hashing

This is the process of changing separate tokens into their corresponding integers. This process generates the dictionary of words. After the process completion of dictionary generation, the words in the dictionary are transformed into respective hash values. This hash value representation is utilized to find the feature is already utilized in the process or not. In this feature hashing, the model results in the group of columns for every text data in rows and one column for every hashed feature.

The feature hashing is equivalent to one hot encoding process and results in the hashes to mention each text word. Example illustration is provided in table 2.

Table 2: *Example of feature hashing*

Word	**Hash1**	**Hash2**
Blue	0	0
Black	0	1
Red	1	0
Pink	1	1

The main objective of using feature hashing is to decrease the size of features. This can be used to mention the texts in variable length to the feature vectors in equivalent numeric size and attains the reduced size of features.

3.2.3 Word2Vec feature

In the Word2Vec feature extraction model, the input is word data, and the output data is vector data. This makes the words into their corresponding numeric form. A similar form of words has the same embedding. This word2vec feature extraction process combines a continuous bag of words (CBoW) and the skip-gram process. Here, CBoW provides the occurrence probability of words. This makes the output target vector for the words in embedding. The word2vec feature extraction model is represented in figure 2.

The continuous bag of words is utilized for finding the target text, and the skip-gram process finds the target context data from the predicted words.

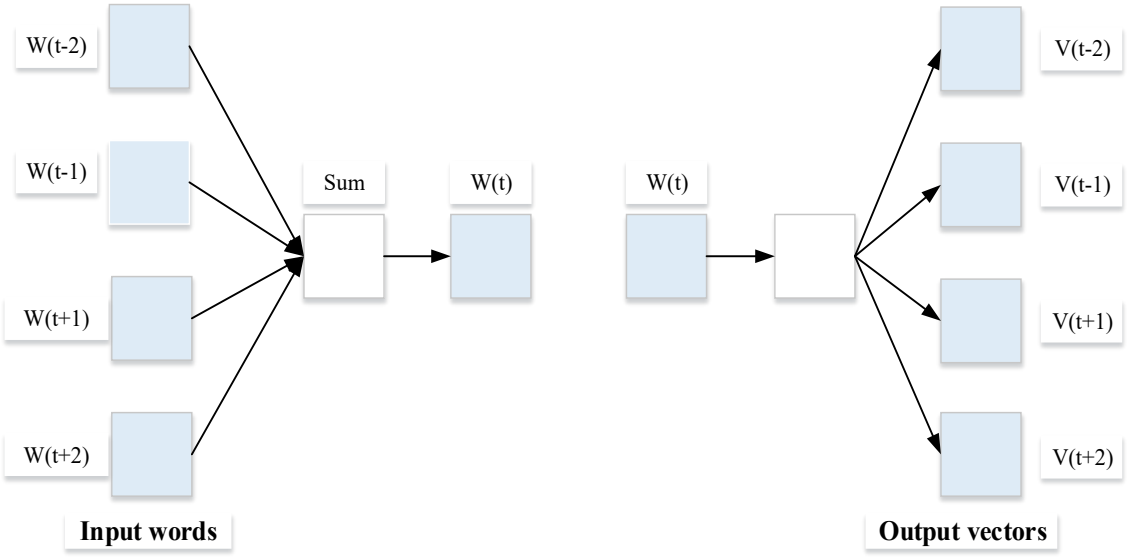

Figure 2: *Word2Vec feature extraction model*

3.2.4 Modified TF-IDF (MTF-IDF)

This feature extraction approach is utilized to find the importance of particular words in a document. Here, the term frequency of a particular word is computed by the number of times that the particular word occurred in the document to the whole number of words in that same document. This is utilized to remove unnecessary words like "a", "an", "the" in the document. The modified form of the TF-IDF feature is computed here to advance the feature extraction process. At first, the term frequency is computed through the subsequent condition (1).

$$\overline{T}_F'' = \frac{m_{l,m}}{\sum_p m_{p,m}} \tag{1}$$

Here, $\overline{T}_F''$ signifies the extracted term frequency feature vector, $m_{l,m}$ signifies the number of the word t_l occurs in the document d_m, and the denominator portion in condition (1) describes the addition of word occurrences of the document. The inverse document frequency is computed through the subsequent condition (2),

$$\overline{Idf}'' = \log\left[\left(\frac{\overline{M}}{Df_l}\right)+1\right] \tag{2}$$

Here, $\overline{\overline{Idf}}''$ signifies the IDF feature vector, $\overline{M}$ signifies the total amount of documents in the dataset and Df_l signifies the feature vector of documents. The TF-IDF feature extraction process is enhanced by updating the weight calculated using the term frequency. The improved form of feature vector based on the weight of term frequency is computed by the subsequent condition (3).

$$\overline{W}_T = \frac{\overline{T}_F{}'' + \overline{\overline{Idf}}''}{2} \tag{3}$$

Here, $\overline{T}_F{}''$ signifies the term frequency of words in each tweet, $\overline{W}_T$ signifies the weight of the word frequency, and $\overline{\overline{Idf}}''$ signifies the feature vector computed by the condition (3) and. The higher weight words are considered as a important word. This weight factor is updated in the TF-IDF feature extraction process to attain the modified TF-IDF feature. This computed weight factor is updated in the subsequent condition (4),

$$\overline{M}_{Tf-Idf} = \overline{T}_F{}'' \times \log\left[\left(\frac{M}{FD_l}\right) * \overline{W}_T + 1\right] \tag{4}$$

Here, $\overline{W}_T$ signifies the updated weight factor computed by the condition (3), the first term in condition (4) describes the term frequency and the next term describes the IDF. Based on this MTF-IDF feature extraction using the condition (4), improved form of feature vectors is attained.

Significance of MTF-IDF: The word frequency of each tweet data are considered for the calculation of weight and it is updated in the condition (4). The average word frequency of words is computed through the condition (3). The condition (3) computed the word frequency weights and it is also updated in condition (4) to improve the process of feature extraction. This weight parameter is used to predict the relevancy of word related to the category. This updated feature extraction process improves the decision on classification and increases the output accuracy. This way of feature extraction makes easier the process of sentiment classification. In existing TF-IDF, only the term frequency and IDF of tweets only considered. This general TF-IDF provides the feature vector counts of all the words in the considered data. But, the improved form of TF-IDF provides the optimal feature vector to the existing to further improve the process of classification. The updated weights improve the TF-IDF feature extraction and it enhances the further classification performance. This feature extraction process effectively supports the accurate sentiment classification as positive, neutral and negative.

3.2.5 Glove feature vectors for words representation

This model is the representation of the text of the global word vector. The idea used in this feature extraction process is a word by word estimation of the counting matrix. The correlation among the random words is determined by finding the ratio between their occurrence probabilities. The relation of co-occurrence is described in the subsequent conditions (5),

$$G_F = \left(\left(K_l - K_m\right)^{\bar{t}} K_P\right) \tag{5}$$

Here, G_F represents the glove feature and K represents the counting matrix. This creates the counting matrix for words between tweets (l, m) to the random words (p). This condition creates the transposable feature matrix based on its transpose ($\bar{t}$). Moreover, the additive shift is utilized in this process as described by condition (6),

$$\log\left(G_F\right) \Rightarrow \log\left(1 + G_F\right) \tag{6}$$

Here, this logarithmic condition (6) maintains the sparsity of G_F and avoids the divergences when calculating the co-occurrence matrix.

3.3 Sentiment word embedding based agglomerative (SWEA) clustering

Sentiment word embedding based agglomerative clustering is a hierarchical based clustering approach. It is otherwise called bottom-up clustering. In this clustering, every data point is considered a separate cluster is, known as a leaf. At first, the input data points are clustered arbitrarily. Afterwards, the distance among two pair clusters are computed and, based on the shortest distance of the data points in the clusters, are grouped further. This process is repeated until getting the one optimal form of cluster. The process of clustering formation is stopped when reaching one optimal cluster known as root. The Euclidean distance between the considered data points are calculated and described in the subsequent condition (7),

$$\overline{E}_D = \sqrt{\sum_{k=1}^{m} \left(y_k - z_k\right)^2} \tag{7}$$

Here, y_k and z_k represents the two data points, $\overline{E}_D$ represents the Euclidean distance. A distance matrix is generated according to the estimated distance between data points. Afterwards,a connection between the clusters is computed through the linkage criteria. This linkage criterion computes the shortest distance among the data points in the clusters. It is computed by the subsequent condition (8),

$$\overline{S}_D(Y,Z) = \underset{y \in Y, z \in Z}{Min} \overline{E}_{D(y,z)} \tag{8}$$

Here, $\overline{S}_D(Y,Z)$ represents the linkage criterion among the data points (Y,Z), $\overline{E}_{D(y,z)}$ represents the Euclidean distance amongst data points. This criterion is utilized to merge the clusters. Furthermore, the distance matrix is updated after this merging. This process is continued until reaching one optimal form of cluster. This process of clustering provides the tree based model for clustering. Here, the word to vector model is utilized for word embeddings. Every neighbouring word is semantically correlated in the word embedding-based clustering process. This clustering process considers that the centroid of the clusters is associated with the neighbouring data point with the minimum local density. The local density of the data points are computed by the subsequent condition (9),

$$\delta_k = \sum_m \chi\left(D_{lm} - D_{cut-off}\right) \tag{9}$$

Here, D_{lm} represents the distance among data points, $D_{cut-off}$ represents the cut-off distance and $\chi(\cdot)$ is represented in condition (10),

$$\chi(\cdot) = \begin{cases} 1 & if \quad D_{lm} < D_{cut-off} \\ 0 & else \end{cases} \tag{10}$$

The local density measure δ_k is equivalent to the total quantity of data points that are closest to the data point k than $D_{cut-off}$. Moreover, ρ_k is computed by the subsequent condition (11),

$$\rho_k = \begin{cases} \underset{l \cdot \delta_m - \delta_l}{Min}\left(D_{lm}\right) & if \quad D_{lm} < D_{cut-off} \\ \underset{l}{Max}\left(D_{lm}\right) & else \end{cases} \tag{11}$$

The Word embedding based clusters are formed as per the illustrations of δ_k and ρ_k. The maximum value of δ_k and ρ_k is considered as a cluster centre. The flow diagram of SWEA clustering is depicted in figure 3.

According to the steps provided in figure 3, sentiment features are clustered. This process clusters the features effectively and is given as an input to the hybrid BLSTM-CNN framework.

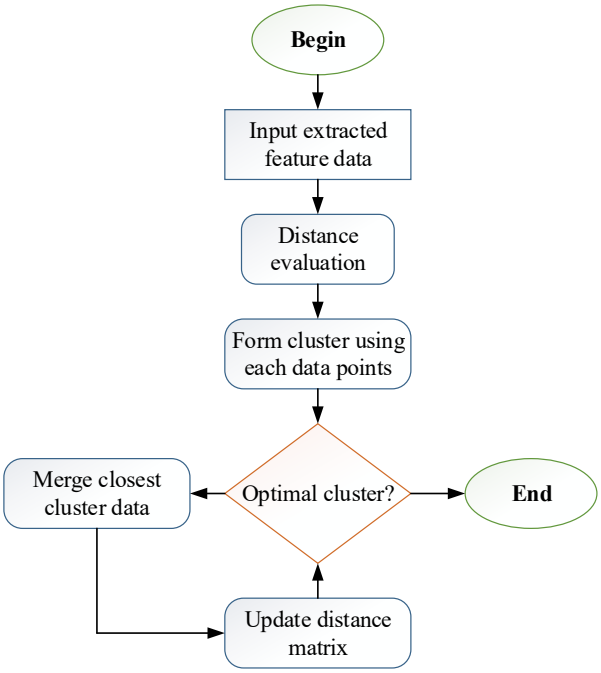

Figure 3: *Flow diagram of SWEA clustering*

3.4 Hybrid BLSTM-CNN

In the presented approach, hybrid BLSTM-CNN is utilized to get the entire context information to predict the accurate output. Combining BLSTM and the CNN layers can provide important information to the output layer through deep learning. Here, the CNN is utilized in the hybrid form is to attain enhanced performance. Here, the weights are updated optimally to increase the performance of classification. The presented hybrid framework comprises CNN layers and the BLSTM layers. The schematic diagram of hybrid BLSTM-CNN is depicted in figure 4.

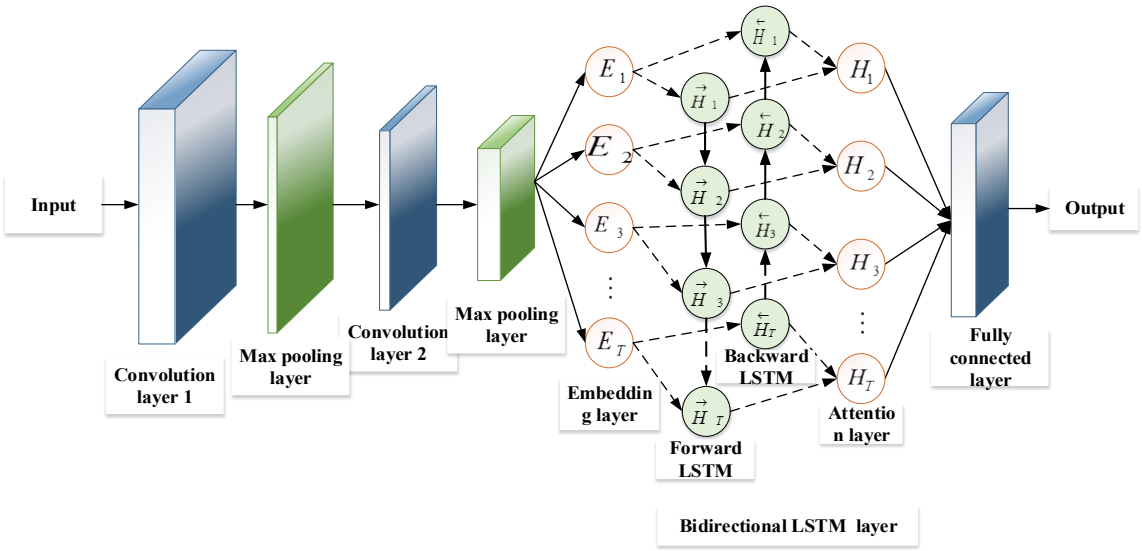

Figure 4: *Structure of hybrid BLSTM-CNN*

3.4.1 Weight optimization

The main motivation of using a modified horse herd optimization algorithm for optimizing the weight is the hierarchical organization of horse herds. The optimization approach is modified here by updating the weight factor in condition (10).In the hierarchical organization, horse A is more dominant than horse B. Similarly, horse B is more dominant than horse C. The optimized weights are attained through the presented optimization approach. The weight of the presented hybrid framework is described by the subsequent condition (12).

$$\Delta\overline{W}_t = -\frac{z\lambda}{m'}\overline{W}_t - \frac{z}{m'}\frac{\partial\overline{b}}{\partial\overline{W}_t} + N\Delta\overline{W}_t(n') \tag{12}$$

The first priority weight of the optimization technique is used to update the hybrid BLSTM-CNN. Here, the hierarchical order is attained through the fitness estimation of horse herds. The number of horses (M) and function $(P)$are considered as per the corresponding conditions.

$$\overline{H} = \left\{\overline{h}_1, \overline{h}_2, \overline{h}_3 .... \overline{h}_M\right\} \tag{13}$$

$$\overline{P} = \overline{H} \rightarrow \{1,2,3....M\} \tag{14}$$

Here, $\overline{H}$ represents the horse herd. The fitness of the horse herd is computed by the subsequent condition (13). If $\overline{F}\left(\overline{h}_u\right) < \overline{F}\left(\overline{h}_v\right)$, where $u \neq v$ and $u, v \in \{1,2,3...m\}$ then

$$\overline{P}\left(\overline{h}_u\right) > \overline{P}\left(\overline{h}_v\right) \tag{15}$$

If $\overline{F}\left(\overline{h}_u\right) = \overline{F}\left(\overline{h}_v\right)$, where $u \neq v$ and $u, v \in \{1,2....M\}$ then

$$\left[\overline{P}\left(\overline{h}_u\right) - \overline{P}\left(\overline{h}_v\right)\right](u - v) > 0 \tag{16}$$

Similarly, the prioritized orders of every horse in the herd are estimated through the subsequent condition (17),

$$\widetilde{O}_P(\overline{h}_u) = \frac{\overline{P}\left(\overline{h}_u\right)}{M} \tag{17}$$

Here, $\widetilde{O}_P(\overline{h}_u)$ signifies the prioritized order of the horses. Afterwards, the weighted average value is estimated for each horse position in the herd, considered a centre of the horse herd. The centre evaluation of each horse herd through this process is computed in the condition (18),

$$\overline{h}_C = \frac{\sum_{u=1}^{M} y_u \overline{h}_{u,rank}}{\sum_{u=1}^{M} \overline{h}_{u,rank}} \tag{18}$$

Here, $\overline{h}_C$ represents the centre of the horse herd. The distance between the location of the horse and the herd centre is computed by the condition (19),

$$\hat{d}_{(horse,herd)} = \sqrt{\sum_{v=1}^{M}\left(horse_v - \overline{h}_C\right)^2} \tag{19}$$

Here, $\hat{d}$ signifies the evaluated distance between the horse location and herd centre.Then, the velocity function is updated if one particular horse is fitted to the group of horse herd through the condition (20),

$$\widetilde{V}_{u,v}^{T+1} = \widetilde{V}_{u,v}^{T} + \overline{h}_{u,rank} * \overline{W}_F * \left(\overline{h}_{center,v}^{T} - y_{u,v}^{T}\right) \tag{20}$$

Here, $\overline{W}_F$ signifies the weight factor. This factor is updated in the optimization approach instead of arbitrary numbers in0 and 1. Here, T signifies the current iteration and $T+1$ signifies

the newest iteration. The weight factor updated in condition (20) is computed by the condition (21),

$$\overline{W}_F = \overline{e}\left(\frac{\overline{CI}}{\overline{I}_{\max} - 1}\right) \tag{21}$$

Here, $\overline{CI}$ represents the current iteration, $\overline{e}$ represents the exponential term, and $\overline{I}_{\max}$ signifies the maximum of iteration. The horse memory matrix (S) comprisesa number of rows equal to the horse memory pool (HMP) and has the P columns.

$$S_u^{T+1} = \begin{bmatrix} S_{1,u,1}^{T+1} & \cdots & S_{1,u,P}^{T+1} \\ \vdots & \vdots & \vdots \\ S_{HMP,u,1}^{T+1} & \cdots & S_{HMP,u,1}^{T+1} \end{bmatrix} T+1 \tag{22}$$

The memory matrix is updated by the subsequent condition described as,

$$G_{best} = S_{N,u,v}^{T+1} = y_{u,v}^{T+1} * \eta_D\left(0, \delta_d\right) \tag{23}$$

Here, η_D signifies the normal distribution and δ_d signifies the standard deviation. Subsequently, the global best position is attained in maximum iteration $K+1$. This optimization process is significant for updating the optimal weight in thehybrid BLSTM-CNN framework. The attained global best positions (G_{best}) are considered optimal weight to update the hybrid BLSTM-CNN framework.

3.4.2 Layers in hybrid CNN-LSTM

The presented framework comprises CNN and BLSTM layers described in the subsequent sub-sections.

- **Convolutional layer**

This layer performsthe convolution operation on the input data of the deep learning classifier. The operation of the convolution layer is represented in the subsequent condition (22),

$$S_F(l,m) = \left(L * N\right)(l,m) \sum\sum T\left(l+u\ ,\ m+v\right) N\left(u,v\right) \tag{24}$$

Here, T represents the input matrix, N represents the 2-dimensional filter of size (u,v) and S_F represents the output of a 2D feature map.

- **Pooling layer**

This layer is otherwise known as a down sampling layer. In this layer, the dimensionality of the features is reduced for the output of the convolutional layer. Here, the biggest feature value is taken for the average pooling operation.

- **BLSTM layer**

The bidirectional LSTM layer is utilized to obtain a high level of features. The bidirectional LSTM is the advanced feature learning of LSTM. This BLSTM layer performs deep learning features in both forward and backward processes. The concatenation of BLSTM layer feature learning is described in the subsequent condition (25)

$$\overline{H}_{Total} = \overrightarrow{H}_t + \overleftarrow{H}_t \tag{25}$$

Here, $\overrightarrow{H}_t$ signifies the forward pass outputs and $\overleftarrow{H}_t$ signifies the backward pass outputs. The forward and backward process outputs are combined in this condition. The output of the final bidirectional LSTM layer is connected with the fully connected layer.

- **Fully connected layer**

The feature matrix attained from the previous layer is flattened and given as an input to this layer. It acts as an interface between the layers with features deep learning to the output decision. This layer is updated with the softmax activation function.

Softmax: This softmax activation function is utilized to predict the output class probability values accurately. This is computed by the subsequent condition (26),

$$\overline{Y}_t = \frac{e^{x_t}}{\sum_{t=1}^{n} e^{x_t}} \tag{26}$$

Here, $\overline{Y}_t$ represents the system output. According to this process, sentiment on Hindi-English mixed twitter text is predicted and classified into positive, negative and neutral.

4. Results and Discussion

This section provides the experimental results of the presented methodology in an effective sentiment classification on Hindi-English code mixed twitter data. The presented methodology is implemented in the PYTHON working platform. In order to achieve better results, the SWEA clustering approach and Hybrid BLSTM-CNN framework are presented. The performance of the developed approach is examined with the different existing methodologies to prove the effectiveness of the developed approach.

4.1 Dataset description: Hindi-English code-mixed twitter dataset

The dataset consists of English-Hindi mixed social media contents. Here, Twitter data is gathered in this database. This dataset contains the twitter data in both Hindi and English language mixed form. The total number of tweets in the dataset is 10500 with 5249 user location. The available total 10500 numbers of tweets is in the Hindi-English code mixed language.

4.2 Performance metrics

In this section, different performance evaluations are provided to analyze the effectiveness of the presented methodology. Various performance metrics like accuracy, precision, recall, F-measure, and Error rate are described in the subsequent subsections,

4.2.1 Accuracy

This performance measure evaluates the overall accurateness of the sentiment classification.This performance measure evaluates the proportion of accurately predicted classes among the total number of classes. It is computed through the subsequent condition (27),

$$A_Y'' = \frac{\overline{T}_{(+ive)} + \overline{T}_{(-ive)}}{\overline{T}_{(+ive)} + \overline{F}_{(+ive)} + \overline{F}_{(-ive)} + \overline{T}_{(-ive)}} \tag{27}$$

Here, A_Y'' signifies the accuracy, $\overline{T}_{(+ive)}$ signifies the true positive, $\overline{T}_{(-ive)}$ signifies the true negative, $\overline{F}_{(+ive)}$ signifies the false positive, and $\overline{F}_{(-ive)}$ signifies the false negative.

4.2.2 Precision

This performance evaluation characterizes the ratio of accurately predicted tweet classfor the particular sentiment to the total amount of classified tweets in that sentiment. It is computed through the subsequent condition (28),

$$\hat{P}''_N = \frac{\overline{T}_{(+ive)}}{\overline{T}_{(+ive)} + \overline{F}_{(+ive)}} \tag{28}$$

Here, $\hat{P}''_N$ signifies the precision performance.

4.2.3 Recall

This performance metric computes the proportion of accurately categorized tweets of given sentiment to the total quantity of tweets that are actually under that sentiment category. It is evaluated through the subsequent condition (29),

$$\overline{\mathrm{Re}''_L} = \frac{\overline{T}_{(+ive)}}{\overline{T}_{(+ive)} + \overline{F}_{(-ive)}} \tag{29}$$

Here, $\overline{\mathrm{Re}''_L}$ signifies the recall performance.

4.2.4 F-measure

The F-measure performance is evaluated by incorporating both precision and recall performances. The F-measure performance is computed through the expressed condition (30),

$$\overline{Fm''} = 2 \times \frac{[\overline{\mathrm{Re}''_L} \times \hat{P}''_N]}{[\overline{\mathrm{Re}''_L} + \hat{P}''_N]} \tag{30}$$

4.2.5 Error rate

The error rate performance is computed based on the ratio of difference among the actual value and the predicted value. To illustrate the accuracy of the classifier, the error rate is computed. This performance measure is computed through the subsequent condition (31),

$$E''_{rate} = \frac{\left|\overline{P}'_d - \overline{A}'_d\right|}{\overline{A}'_d} \tag{31}$$

Here, E''_{rate} signifies the error rate, $\overline{P}'_d$ signifies the predicted data and $\overline{A}'_d$ signifies the actual data.

4.3 Performance Analysis

In this section, the performance of the presented approach is examined with different existing methodologies. The accuracy performance comparison with different existing approaches is mentioned in table 3. Table 3 provides the accuracy performance comparison. This proved that the presented approach attains enhanced accuracy performance (97.2%) than the existing KL-NF (Kull back-LeiblerNeuro fuzzy) (89.93%) [28], SVM (support vector machine) (88.13%), Multi-class SVM (70.1%), LAN^{2}FIS (logistic adaptive network depends on Neuro-fuzzy inference system) (89%), KNN (K-nearest neighbour) (67%), KNN+SVM (76%), DT (decision tree) (80%), SMO (sequential

minimal optimization) +DT (89.47%) methodologies [27]. Moreover, the graphical representation of the accuracy performance comparison is provided in figure 5.

In figure 5, accuracy performance is compared with the different existing methodologies. From this illustration, the presented approach attains higher accuracy performance (97.2%) than the existing KL-NF (89.93) [28], SVM (88.13%), Multi-class SVM (70.1%), LAN²FIS (89%) [26], KNN (67%), KNN+SVM (76%), DT (80%), SMO+DT (89.47%) [27]. This proved that the presented approach provides a significant improvement in regards to accuracy than the existing approaches. The performance comparison on precision is mentioned in table 4.

Table 3: *Performance on accuracy*

Methodology	**Accuracy (%)**
KL-NF	89.93
SVM	88.13
Multi-class SVM	70.1
LAN²FIS	89
KNN	67
KNN+SVM	76
DT	80
SMO+DT	89.47
Proposed	97.2

In table 4, the presented methodology performance on precision is portrayed. This representation depicts that the performance of the presented approach is attaining improved performance in precision (96.8%). The existing methodologies are attained only lesser precision values than the presented approach. Furthermore, the performance examination on precision is depicted in figure 6.

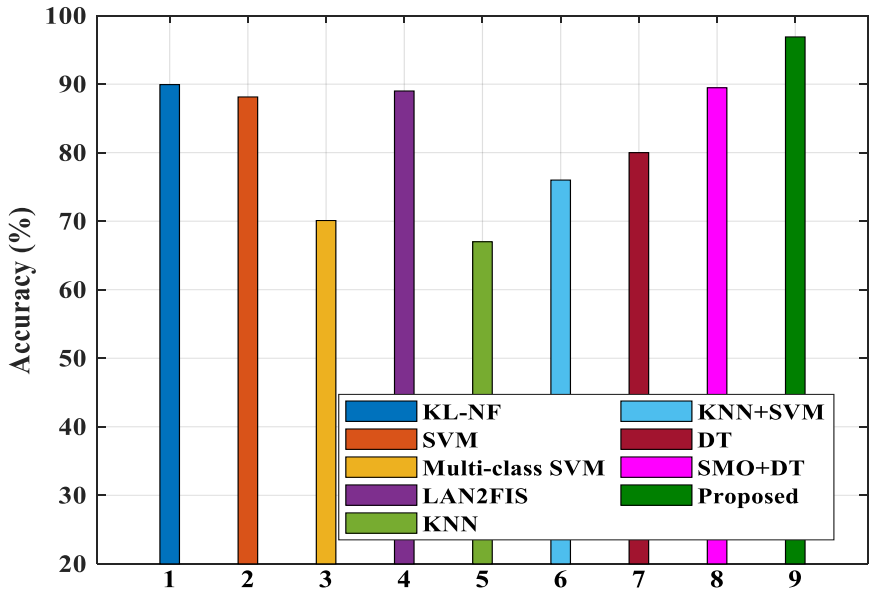

Figure 5: *Comparison examination on accuracy*

Table 4: *Performance comparison on precision*

Methodology	Precision (%)
KL-NF	93.67
SVM	84.15
Multi-class SVM	69.7
LAN²FIS	88.12
KNN	70.5
KNN+SVM	68.45
DT	81.4
SMO+DT	91.6
Proposed	96.8

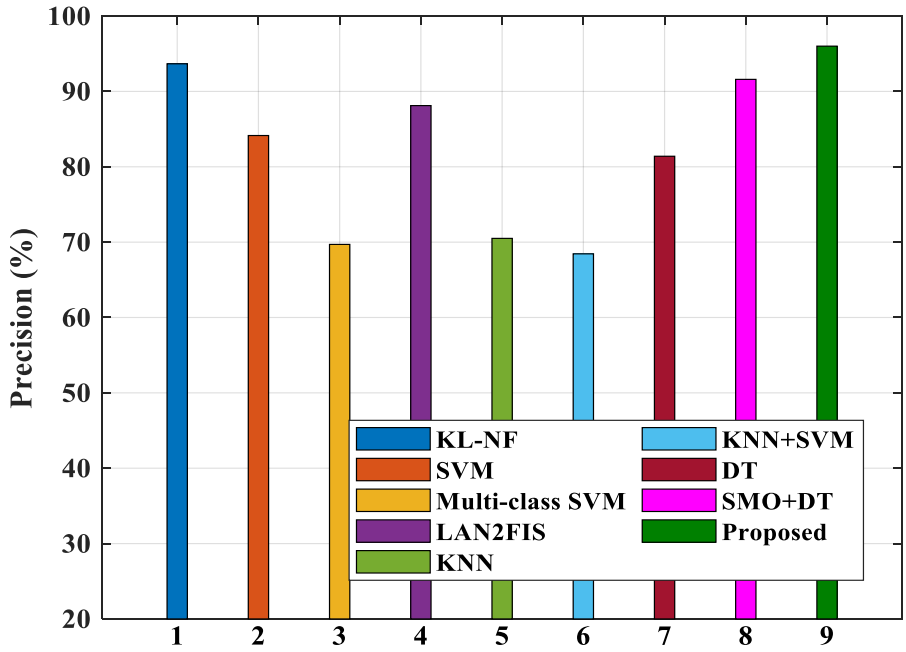

Figure 6: *Comparison examination on precision*

In figure 6, the precision performance is compared with the already existing approaches. The presented approach attains improved precision performance than the existing methodologies. The precision performance of the presented approach is 96.8%, which is higher than the existing approaches like KL-NF (93.67%) [28], SVM (84.15%), Multi-class SVM (69.7%), LAN²FIS (88.12%), KNN (70.5%), KNN+SVM (68.45%), DT (81.4%), SMO+DT (91.6%) [27]. Then, the comparative analysis on recall performance is provided in table 5.

Table 5: *Comparison analysis on recall*

Methodology	**Recall (%)**
KL-NF	93.01
SVM	89.76
Multi-class SVM	70.1
LAN²FIS	89.96
KNN	69.3
KNN+SVM	68.14
DT	81.4
SMO+DT	89.5
Proposed	**96.5**

In table 5, recall performance analysis is compared. This demonstrates that the presented approach providesenhanced recall (96.5%) than the various existing techniques. This further proved the effectiveness of the presented approach in accurate sentiment classification on Hindi English mixed tweets. Moreover, the performance comparison on recall is portrayed in figure 7.

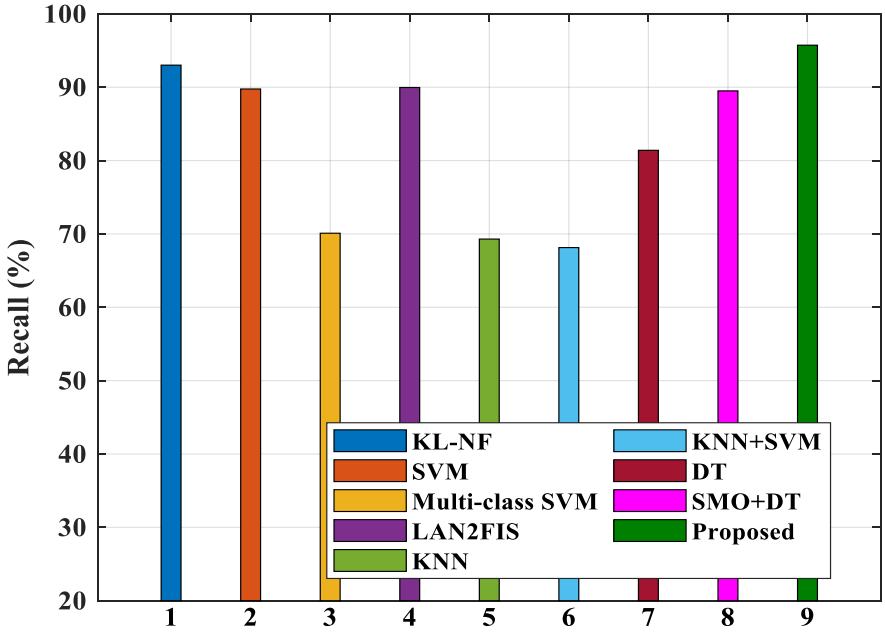

Figure 7: *Comparison examination on recall*

In figure 7, the performance comparison on recall is depicted. This proved that the developed approach attaining improved performance in regards to recall than the existing methodologies. The recall performance of the presented approach is 96.5%, which is significantly higher than the existing approaches like LAN²FIS (89.96%) [26], KL-NF (93.01%) [28], SVM (89.76%), KNN (69.3%), Multi-class SVM (70.1%), KNN+SVM (68.14%), SMO+DT (89.5%), DT (81.4%) [27]. Moreover, the comparison analysis of precision, F-measure and recall performances with varying feature extractions is given in table 6.

Table 6: *Performance comparison based on feature extraction*

Techniques	Recall (%)	Precision (%)	F-measure (%)
Key word extraction with TF-IDF	72.85	59.28	65.37
Key word extraction with WF-TF-IDF	85.57	69.47	76.68
Proposed with MTF-IDF	96.5	96.8	97

Table 7: *Performance examination on Error rate*

Methodology	Error rate (%)
SVM	17.81
Multi-class SVM	29.12
LAN²FIS	11
Proposed	**4.2**

In table 6, the performance comparison is provided by varying feature extractions. This proved that the presented approach provides improved performance with MTF-IDF feature extractionthan the existing TF-IDF and weight frequency based TF-IDF (WF-TF-IDF) [29]. The Comparison analysis on error rate is mentioned in table 7.

Table 7 compares the error rate results to the existing schemes. Here, the presented approach attains a lesser error value (4.2%) than the existing approaches like SVM (17.81%), Multi-class SVM (29.12%), and LAN²FIS (11%) [26]. Lesser error value increases the accuracy level of the developed approach in accurate sentiment classification. Furthermore, the performance comparison on error rate is depicted in figure 8.

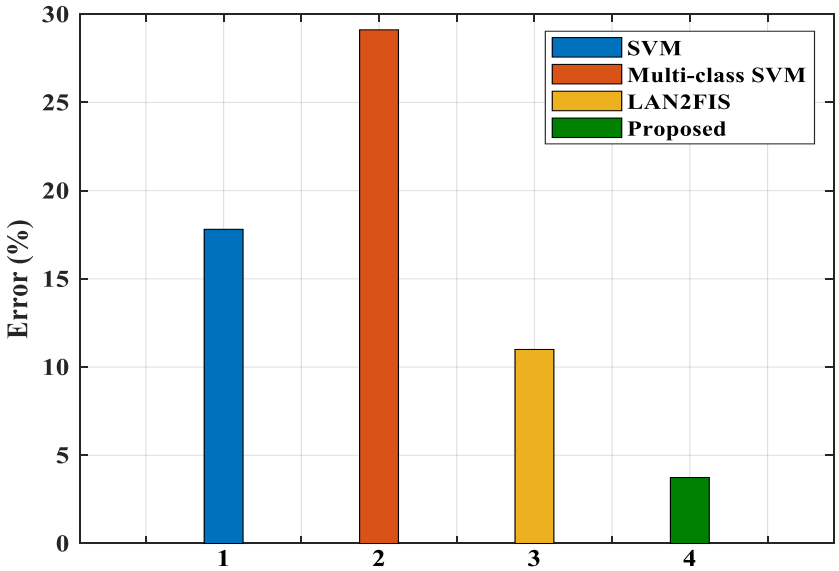

Figure 8: *Performance examination on the Error rate*

Figure 8 compares the error rate shown in sentiment classification using different approaches. The presented approach attains a lesser error rate, and it proves the efficiency of the presented approach in sentiment classification. The error rate of the presented approach is (4.2%), which is

significantly lesser than the different existing classifier approaches like SVM (17.81%), Multi-class SVM (29.12%), and LAN²FIS (11%) [26]. This demonstrates that the presented approach attains improved performance than the existing methodologies.Then the F-measure performance comparison is provided in table 8.

Table 8: *Comparison analysis on F-measure*

Methodology	**F-measure (%)**
KL-NF	93.33
SVM	88.63
Multi-class SVM	69.9
LAN²FIS	89.03
KNN	67.9
KNN+SVM	77.56
DT	80.95
SMO+DT	96.3
Proposed	**97.0**

Table 8 compares the F-measure performance of the presented methodology with the different existing approaches. The F-measure of the presented approach is 97.0%, which is very much higher than the existing approaches. These performance evaluations are increasing the efficiency of the presented approach in sentiment classification. Furthermore, the performance comparisonon F-measure is depicted in figure 9.

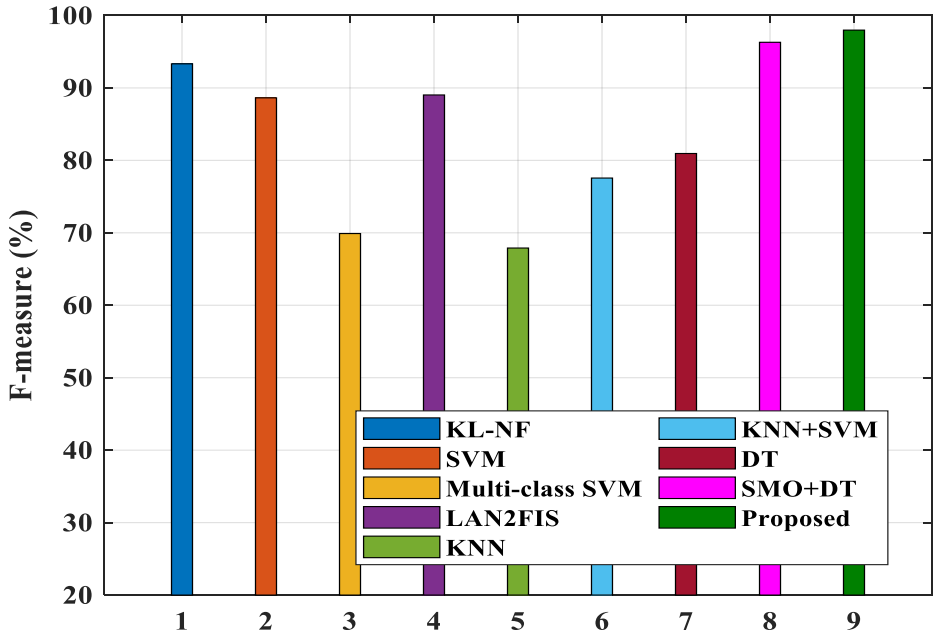

Figure 9: *Performance examination on F-measure*

Figure 9 compares the F-Measure performance with the existing approaches. The attained F-measure performance of the presented approach is 97.0%, which is significantly enhanced than the different existing methodologies like KL-NF (93.33) [28], SVM (88.63%), Multi-class SVM (69.9%),

LAN²FIS (89.03%) [26], KNN (67.9%), KNN+SVM (77.56%), DT (80.95%), SMO+DT (96.3%) [27]. Performancecomparisons are proved the effectiveness of the presented framework in sentiment classification.

5. Conclusion

This paper presented an effective classification of sentiments in Hindi-English mixed twitter data. At first, the input twitter data is pre-processed with Stemming, stop word removal, tokenization, and lemmatization processes. Subsequently, effective features like Glove feature, count vectors, Feature hashing, Modified term frequency-inverse document frequency, and Word2vector features are extracted for improved sentiment analysis. Then the extracted features are clustered by utilizing sentiment word embedding based agglomerative clustering. Lastly, a hybrid Bidirectional long short term memory-convolutional neural network (Hybrid BLSTM-CNN) is utilized for accurately categorizing the tweet sentiments into positive, negative and neutral. Here, the modified horse herd optimization (MHHO) approach is utilized to update the optimized weights in Hybrid BLSTM-CNN. The presented methodology effectively predicts the sentiments in Twitter data. The experimental results of the presented Hybrid BLSTM-CNN framework in sentiment analysis provided improved performance than the different existing approaches in regards to accuracy (97.2%), precision (96.8), Error rate (4.2%), recall (96.5%), and F-measure (97.0%).

References

[1] Himelboim I, Smith MA, Rainie L, Shneiderman B and Espina C (2017). Classifying Twitter topic-networks using social network analysis. *Social media+ society*, 3(1):2056305117691545.

[2] Pravalika A, Oza V, Meghana NP and Kamath SS (2017). Domain-specific sentiment analysis approaches for code-mixed social network data. *In 2017 8th international conference on computing, communication and networking technologies* (*ICCCNT*), (pp. 1-6). IEEE.

[3] Jain K, Deshpande A, Shridhar K, Laumann F and Dash A (2020). Indic-transformers: An analysis of transformer language models for Indian languages. arXiv preprint arXiv:2011.02323. 2020.

[4] Kunchukuttan A and Bhattacharyya P (2020). Utilizing language relatedness to improve machine translation: A case study on languages of the indian subcontinent. arXiv preprint arXiv:2003.08925.

[5] Abbas A and Setiawan HI (2020). How talk show presenter using code mixing and code switching on TV program in Indonesia. *Klasikal: Journal of Education, Language Teaching and Science,* 2(2):20-39.

[6] Gupta A, Menghani S, Rallabandi SK and Black AW (2021). Unsupervised self-training for sentiment analysis of code-switched data. arXiv preprint arXiv:2103.14797.

[7] Mahadzir NH (2021) Sentiment Analysis of Code-Mixed Text: A Review. *Turkish Journal of Computer and Mathematics Education* (*TURCOMAT*) 12(3):2469-78.

[8] Drus Z and Khalid H (2019). Sentiment analysis in social media and its application: Systematic literature review. *Procedia Computer Science,* 161:707-14.

[9] Gadekallu T, Soni A, Sarkar D and Kuruva L (2019). Application of sentiment analysis in movie reviews. *In Sentiment Analysis and Knowledge Discovery in Contemporary Business,* (pp. 77-90). IGI global.

[10] Madani Y, Erritali M, Bengourram J and Sailhan F (2020). A multilingual fuzzy approach for classifying Twitter data using fuzzy logic and semantic similarity. *Neural Computing and Applications,* 32(12):8655-73.

[11] Singh K, Sen I and Kumaraguru P (2018). A Twitter corpus for Hindi-English code mixed POS tagging. *In Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media* (pp. 12-17).

[12] Chakrawarti RK, Bansal J and Bansal P (2022). Machine translation model for effective translation of Hindi poetries into English. *Journal of Experimental & Theoretical Artificial Intelligence,* 34(1):95-109.

[13] Sreelakshmi K, Premjith B and Soman KP (2020). Detection of hate speech text in Hindi-English code-mixed data. *Procedia Computer Science,* 171:737-44.

[14] Aneez Z, Chattapadhyay S, Parthasarathi V and Nielsen RK. Digital transition of newspapers in India: *Dainik Jagran, Hindustan Times, and Malayala Manorama*.

[15] Wadhawan A and Aggarwal A (2021). Towards Emotion Recognition in Hindi-English Code-Mixed Data: A Transformer Based Approach. arXiv preprint arXiv:2102.09943.

[16] Kortelainen H. Forms and functions of codeswitching to Hindi/Urdu in Indian English and Pakistani English.

[17] Sane SR, Tripathi S, Sane KR and Mamidi R (2019). Stance detection in code-mixed hindi-english social media data using multi-task learning. *InProceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (pp. 1-5).

[18] Ansari MZ, Ahmad T and Arshad Ali M (2018). Cross Script Hindi English NER Corpus from Wikipedia. *InInternational Conference on Intelligent Data Communication Technologies and Internet of Things* (pp. 1006-1012). Springer, Cham.

[19] Kim H and Jeong YS (2019). Sentiment classification using convolutional neural networks. *Applied Sciences,* 9(11):2347.

[20] Singh J and Tripathi P (2021). Sentiment analysis of Twitter data by making use of SVM, Random Forest and Decision Tree algorithm. *In2021 10th IEEE International Conference on Communication Systems and Network Technologies* (*CSNT*), (pp. 193-198). IEEE.

[21] Jhanwar MG and Das A (2018), An ensemble model for sentiment analysis of Hindi-English code-mixed data. arXiv preprint arXiv:1806.04450.

[22] Sasidhar TT, Premjith B and Soman KP (2020). Emotion detection in hinglish (hindi+ english) code-mixed social media text. *Procedia Computer Science,* 171:1346-52.

[23] Singh K, Sen I and Kumaraguru P (2018). A Twitter corpus for Hindi-English code mixed POS tagging. *In Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media* (pp. 12-17).

[24] Singh P and Lefever E (2020). Sentiment analysis for hinglish code-mixed tweets by means of cross-lingual word embeddings. *In LREC 2020–4th Workshop on Computational Approaches to Code Switching* (pp. 45-51). *European Language Resources Association* (*ELRA*).

[25] Garg N and Sharma K (2020). Annotated corpus creation for sentiment analysis in code-mixed Hindi-English (Hinglish) social network data. *Indian Journal of Science and Technology* 13(40):4216-24.

[26] Nagamanjula R and Pethalakshmi A (2020). A novel framework based on bi-objective optimization and LAN2FIS for Twitter sentiment analysis. *Social Network Analysis and Mining* 10(1):1-6.

[27] Naresh A and Venkata Krishna P (2021). An efficient approach for sentiment analysis using machine learning algorithm. *Evolutionary Intelligence,* 14(2):725-31.

[28] Garg K and Lobiyal DK (2021). KL-NF technique for sentiment classification. *Multimedia Tools and Applications,* 80(13):19885-907.

[29] Shao L, Zhang F, Wang R, Zhou F and Lin S (2018). An Efficient Expansion Word Extraction Algorithm for Educational Video. *In 2018 7th International Conference on Digital Home* (*ICDH*) (pp. 131-136). IEEE.