IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 May 2023, accepted 2 June 2023, date of publication 7 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283461

RESEARCH ARTICLE

Toward Machine Learning Based Binary Sentiment Classification of Movie Reviews for Resource Restraint Language (RRL)–Hindi

ANKITA SHARMA AND UDAYAN GHOSE, (Member, IEEE)
University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University, New Delhi 110078, India

Corresponding author: Ankita Sharma (ankitasharma2711@gmail.com)

This work was supported by the Guru Gobind Singh Indraprastha University through Indraprastha Research Fellowship (IPRF) under Award GGSIPU/DRC/2022/1247.

ABSTRACT Sentiment analysis has significantly progressed in English, whereas Hindi research is still nascent. Despite being the third most spoken language worldwide, Hindi remains an RRL. Movie reviews are a treasure trove of opinionated content fueled by people's passionate engagement with film industry. The proliferation of great use of Hindi in writing reviews has catalyzed our endeavor to devise an approach for bipolar sentiment classification of movie reviews. We compiled and manually annotated a Hindi Language Movie Review (HLMR) dataset comprising 10K reviews for experiments, and challenges associated with Hindi have also been identified. In addition to HLMR, two publicly available IIT-P movie and product review datasets are used. Following dataset preprocessing, we explored TF-ISF with word-level N-gram features for text representation. Studies suggest that performance of machine learning approaches can be enhanced by hyperparameter tuning and ensemble learning. Several baseline classifiers were initially applied, and their parameters were hyper-tuned using Grid search. Subsequently, ensemble-based classifiers were applied individually. Lastly, we propose a simplistic yet powerful stacked ensemble-based architecture (SEBA), which effectively classifies Hindi reviews by leveraging the strengths of both approaches. Comprehensive experiments were conducted on all deployed datasets. Empirical results demonstrate that SEBA outperformed individual baselines and exhibited superior performance with unigrams and TF-ISF as features across deployed datasets. SEBA achieved an accuracy, precision, and recall of 0.808% and an F1-score of 0.807% on the HLMR dataset. These findings strongly advocate for the effectiveness of proposed solution and indicate its suitability for online deployment in binary review classification tasks.

INDEX TERMS Hindi, machine learning, movie reviews, NLP, opinion mining, sentiment analysis, stacking ensemble, TF-ISF.

I. INTRODUCTION

Hindi is the official national language of India, along with English, and is ranked the third most spoken language worldwide. It is written in the Devanagari script. With over 600 million native speakers, Hindi is prevalent in India and the world. Textual Sentiment analysis (SA) or sentiment classification (SC), or opinion mining, is defined as the process of determining the attitude of the writer concerning the overall polarity of the text [1]. The attitude may be the emotional state while writing or the emotional effect the writer wishes to have on the reader. SA research in English has yielded triumphing achievements, while research in Hindi SA is still blooming. Resource restraint language (RRL) is a language lacking various linguistic resources. Hindi is an RRL. With technological advancements, numerous websites have adopted Hindi language text (HLT) as their content language, like youtube.com, Facebook.com, Instagram.com, Twitter.com, Wikipedia.org, whatsapp.com, wordpress.com, etc. Hindi as the language of the content on the web is noteworthy in the

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

present-day world. The recent outburst in using Hindi (Devnagari Script) while using social media, expressing opinions, writing reviews posts, and participating in forums has led to a massive amount of opinionated content in Hindi on the web, which needs to be analyzed to mine pure gold. The availability of substantial online data in Hindi in the past decade with the advent of the Unicode UTF-8 standard made it an exciting research area. It has become critical to analyze this HLT and recover valuable insights and relevant information from them. Word of mouth is an integral phenomenon in any society and a powerful weapon of 21st century. People often trust the recommendations made by people, and it provides ultimate validation and social proof. It is known that individuals share their sentiments and opinions through reviews and get quite directed by others' opinions too [2]. The reviews express viewpoints, thoughts, and opinions regarding a particular entity. Review text classification or review text mining assigns a category or class to the review text from a set of pre-defined categories, in our case, the sentiment polarity labels according to its content. Formally, suppose R = {r1, r2, r3, ….., rn} is the set of training review sentences. Assume that each review text is assigned to a set of sentiment polarity labels P = {pos, neg}. The review text classification develops a model whereby given a new review that r'$\notin$ R, it will be assigned to a sentiment polarity label in P [3], [4].

Movie reviews (MRs) are one of the ways words of mouth can be distributed. There is no higher compliment than positive MRs. Positive reviews are a shout-out to the masses to go for the movie. MRs are opinionated text; therefore, they can never be standardized, and one person's appreciation or depreciation may differ from others. Movie reviews in Hindi (MRH) are one such opinionated material present in leaps and bounds owing to people's craziness regarding the movie/film industry. The written opinions, in the form of reviews, are significant and influential and affect everyone, from movie audiences and critics to the production company. The conviction and sentiment expressed in the form of written MRs give convenience and freedom to find the collective likes and dislikes of the individual regarding the movie [5].

Sentiments expressed in written reviews of the movie domain precisely reflect the emotion being conveyed and allow new explorations to find collective word of mouth. However, delving through many such reviews is monotonous and beyond human power. Thus, to provide collective positive and negative opinions, i.e., a scalable solution to this problem is sentence-level sentiment analysis (SLSA). SLSA has attracted the attention of academicians, researchers, and industry professionals. Confirming the fact, SLSA involves two subtasks: (A) Identifying whether a sentence is subjective or objective. (B) Determining the polarity of the sentiment expressed in a sentence. The second subtask is dealt within this work. In the present work, we discriminate only between positive and negative sentiment reviews, turning the task into a standard binary SC problem. Based on the analysis level, SA or SC can be categorized into three levels – document level, sentence level, and aspect or phase level [1], [6]. In the present work, our study will be confined to sentence-level analysis of MRH.

A. MOTIVATION

The proliferation of such a great use of Hindi (Devanagari script) while writing MRs has inspired us to devise an approach for SC of MRs. Our work is motivated by the fact that the ensemble methods provide a propitious way of dealing with classification and overfitting problems [7], [8]. In this work, ensemble-based architecture is used for HLT SC which is not much considered in the literature. Much of the earlier work in HLT in the movie domain was centered around the lexicon-based approach (LBA) and machine learning-based approach (MLA) [9], and not much work in ensemble learning-based approaches is done in Hindi SC and hence is a problem worthy of research. The main objective of our study is as follows. We study, analyze, and propose a stacked ensemble-based architecture (SEBA) to classify the polarity of MRs with two opposing polarities, such as positive or negative. Several state-of-the-art (SOTA) MLAs have been applied to the MRH, and we, in our proposed architecture, combined them to achieve a new state-of-the-art. In this work, we have used HLMR, an author-made corpus in addition to the well-known and publicly available IIT-P movie and product review datasets for Hindi SA. The achieved results on all the deployed datasets show performance improvement compared to MLAs applied individually. We realized without any doubt that a combination of classifiers performs better than any individual MLAs, and the proposed architecture benefits from complimentary MLAs with a diverse set of techniques.

B. CONTRIBUTIONS

The critical point of the present work is to propose a simplistic yet powerful machine-learned based solution for SC of MRH into bipolar sentiment category, i.e., positive, or negative.

- We compiled and annotated Hindi Language Movie Review (HLMR) dataset comprising 10,000 MRs. The HLMR dataset is used in this study.
- To validate the results and potential of the author-made HLMR dataset, all the experiments were performed on two datasets, namely, IIT-P Movie and IIT-P Product review dataset along with HLMR.
- Analysis and pre-processing on all three datasets used were performed. We explore TF-ISF and word-level N-gram features consisting of unigram, bigram, and trigram for text representation and feature extraction. Ours is the first work that uses hand craft-based feature TF-ISF for Hindi review SC.
- MLAs performance can be improved by both hyperparameter tuning and ensemble methods. Both ways were deployed to enhance the performance of MLAs for binary SC of MRH. Firstly, several SOTA classifiers

were applied, and their parameters were hyper-tuned using grid search. Secondly, several MLAs based on an ensemble approach were applied individually. To the best of our knowledge, both hyperparameter tuning and ensemble-based classifiers and methods have not been applied to the MRH dataset in a single work.
- We argue that in SC for an RRL – Hindi, even a minute enhancement in performance requires complex model architecture. This paper proposes a simplistic yet powerful machine learning-based solution, namely, SEBA for SC of MRH which is efficient, easily implemented, requires less computational resources, and is suitable for overcoming overfitting problems. In addition to gaining a comprehensive understanding of SC of Hindi reviews, this work also identifies the challenges and issues encountered while working with RRL-Hindi.
- Results were evaluated using accuracy, F-measure, recall, precision, AUC-ROC score, and MCC score.

C. ARTICLE ORGANIZATION

The rest of the article is ordered as follows. After an introduction, Section II describes the complexities and challenges of dealing with a resource-restraint language - Hindi. Section III provides an overview of the literature review. In Section IV, the material and methods are discussed. Section V is dedicated to the approach and design methodology followed. In Section VI, the achieved results and discussions are done. Finally, we finish with the conclusion and perspectives on future directions in Section VII.

II. COMPLEXITIES AND CHALLENGES FOR DEALING WITH A RRL–HINDI

SC is a hot research topic in text mining for low-resourced Asian languages. SC of HLT is one of the most escalating tasks in the era of NLP, and Hindi is one of the RRLs for which little work has been done during the last decade [10]. The ethnic mélange, access to indigenous language keyboards, affinity, and zeal to communicate in one's native languages adds to challenges and complexities while dealing with RRL like Hindi [11], [12]. There are a lot of challenges and difficulties that one comes across while doing this task. Some of the significant challenges that researchers face while dealing with HLT are mentioned below:

- *No word order followed*: Hindi is liberal in the sense that in Hindi, there is no restriction on word order that a verb should follow a subject should be followed by an object, unlike in English. This means words can be in any order. The exact words with slight variations in their order affect the word polarity of the text to a greater extent. E.g.: {सीता बाजार गई। (S-O-V), बाजार गई सीता।(O-V-S)} both the sentences have the same meaning that ''Sita went to market,'' but the order of the words is different.
- *Resource Restraint:* It is resource-poor language. There are insufficient tools and resources available to deal with HLT. Though resources are available for HLT, they are scarce and not of gold-standard quality; in other words, they are not as efficient as those available for Western languages like English. The resources present are either in the developing stage or are not authentic, they still require a lot of improvisation to efficiently perform the task of SC.
- *Hindi Text Corpus:* There is no gold standard Hindi text corpus available, so getting the correct annotated corpus with the required quality and quantity of data is challenging.
- *Variation in spellings:* In Hindi, the same word with the same meaning can have different spellings; therefore, it becomes tough to incorporate all such words in lexical resources and when training a classifier. E.g., The word ''Hindi'' can have various spellings such as ''हिन्दी,'' ''हिंदी.'' Similarly, the word ''Delhi'' can be written in Hindi using different spellings such as ''डेली,'' ''देहली,'' ''दिल्ली.''
- *Morphologically rich nature:* Hindi is morphologically rich; much information is amalgamated in Hindi words. For example, the words in Hindi text provide an idea about the gender of the person speaking, tense, etc. E.g., Alex is a gender-neutral name. The sentence in English, ''Alex goes to school,'' gives no information about the gender of Alex. On the other hand, In Hindi, we can write this sentence as ''एलेक्स स्कूल जाती है'' and '' एलेक्स स्कूल जाता है'' in Hindi sentence, we can know the gender of Alex implicitly in both the sentences.
- *Binary Gender:* There are only two genders in Hindi – masculine and feminine; the gender information of the inanimate object is unpredictable in HLT.
- *Negation Handing:* Handling negations is more complex in Hindi. For instance- अजय देवगन की कॉमेडी फिल्म से कॉमेडी गायब। translates to ''Comedy is missing from Ajay Devgan's comedy film''.
- *Hinglish:* In the era of the multilingual world, people often use an amalgamation of Hindi and English words that is Hinglish while writing in Hindi over web. For instance- फुक्रे फिल्म को देखने के बाद ये कहना गलत ना होगा कि फिल्म का विषय और फिल्म के किरदार बहुत ही टची और बहुत ही एंटरटेनिंग है । (After watching film Fukrey it will not be wrong to say that film's subject and its characters were very touchy and entertaining). Getting a pure Hindi dataset is tedious.

III. RELATED LITERATURE

Hindi is one of the RRLs for which little work has been done in SC during the last decade. Although SC in resource-affluent languages like English has remarkably accelerated, SC in HLT has always lagged. SA is one of the most engrossing, fascinating advancements in NLP. Lately, there has been more development in SA in HLT due to increased interest and demand of NLP enthusiasts worldwide.

Nearly 4% of the world's population speaks Hindi; consequently, the past few years in Hindi SC have been fruitful, and SC in HLT has been a topic of interest for researchers. Nowadays, a lot of researchers are working in mining areas of HLT, such as SA [13], [14], text summarization [15], [16], [17], and information retrieval [18], [19]. The research into Hindi text classification is limited compared to research in Western languages like English. The characteristics of the Hindi language and the unavailability of linguistic resources are significant reasons.

Textual SA is an exigent and emerging field of text mining and NLP, which is concerned with extracting meaningful information from user-generated content for tracking people's moods regarding political events, topics, products, etc. Textual SA can be considered a binary classification problem with a motive to classify written text for, e.g., Review, post, comment, tweet, etc., into positive or negative. [20], [21]. It usually consists of three main steps: data preparation, feature extraction, and SC. The development in the UTF-8 standard led to a surge in HT movie review data over the web, which needs to be analyzed to produce valuable information from it, helping viewers decide whether to watch a Bollywood movie or not, also making makers realize good and bad aspects of the movie thus helping in forecasting success. Below is a literature review related to machine learning-based SA.

Nanda et al. [22] attempt to perform SA on MRs in Hindi using Random Forest (RF) and Support vector machine (SVM) for classification and Hindi Senti Word Net for polarity finding. Neutral class to be included for future work. Jha et al. [23] have proposed HOMS for movie data SA at the document level. Naive Bayes (NB) was used for classification, and adjectives were considered polarity-bearing words. Also, negative handling was used to increase accuracy. Discourse relation is considered for future work.

Mumtaz and Ahuja [24] attempted SA on Twitter and retrieved movie reviews using the Senti-lexicon algorithm. A separate list of polarity words and emoticons was included. Data sets are enhanced, and forged review and sarcasm detection are considered for future work. Galvao and Henriques [25] use decision trees (DT), Neural Networks (NN), and regression to predict box office success using the SEMMA approach—other text mining techniques to be applied in the future. Jha et al. [26] performed document-level SA on MRs in Hindi using dictionary-based approaches and employing NB, Maximum Entropy (ME), and SVM. The limitation is the small dataset.

In [27], success class prediction of Indian films is made using NN—unsupervised MLMs to be employed for future work. In [28], a dataset of 755 MRs is taken for Box Office prediction of movies using NB, SVM, RF, Logistic Regression (LR), AdaBoost, Multilayer perceptron (MLP), and stochastic gradient descent. MLP performed best among all. Genre and sequel will be considered for future work. Kanitkar [29] prognosticated the Box office success of Indian movies using KNN, DT, NB, RF, and NN. Only a dataset of 250 MRs is taken. In [30], a back-propagation NN model is proposed for prognosticating box office success using data from movie databases and social networking sites. The highest accuracy was obtained using the proposed approach. The results showed that both SVM and NN faced trouble in correctly classifying flop movies. In the future, more movie categories and new input data attributes will be added. In [31], Korovkinas et al. introduced a novel method of combining SVM and NB for performing binary SA on three different datasets from Amazon reviews, movie reviews, and sentiment140. Results show that the proposed method performs better than individual classifiers in all three domains. The proposed method achieved an accuracy of 89.19% with the Amazon review dataset, 88.66% with the movie's dataset, and 78.31% with the tweet's dataset. The point of study in [32] was to propose a hybrid method of combining weak SVCs using boosting ensemble techniques for performing SA on movies and hotel domain reviews. A dataset consisting of 2K reviews was used for the study. Results show that boosted SVM outperforms individual SVM in terms of accuracy and has achieved an accuracy of 93% for the MRs domain. The study was restricted to only binary SA of reviews. An ensemble consisting of NN and SVM was proposed by Sangam and Shinde [7] for performing SA on social media reviews. A movie review dataset consisting of a total number of 2K MRs was used for this study, with equal positive and negative reviews. The baseline models employed were NB, MAXENT, and SVM. The most persistent feature selection (MPFS) method was used for the experiment, and the genetic algorithm was used to optimize the feature set. Results show that the proposed ensemble performed better than the baseline models and achieved the highest accuracy, followed by SVM and NB. Maxent performed the worst in comparison to other models.

Madan and Ghose [33] conducted SA on Hindi tweets collected from Twitter regarding MRs. They applied three different approaches: LBA, MLA, and Hybrid. The researchers found that the hybrid approach was more effective than LBA. They also found that the DT classifier had the highest accuracy, achieving a rate of 92.97%. Hussaini et al. [34] conducted a score-based SA on Hindi book reviews. They created an annotated dataset of 700 sentences related to Hindi book reviews to carry out their analysis. The researchers used verbs, nouns, adjectives, and adverbs as opinion words and applied three techniques: H-SWN, word sense disambiguation, and the Hindi subjectivity lexicon. Their results showed that the Hindi subjectivity lexicon performed the best and achieved an accuracy of 87.4%.

Kumar et al. [35] extended the Indian sentiment lexicon and performed SA on Indian tweets. Their analysis used co-occurrences at the sentence level and DTs. To collect the data for their study, they obtained corpora consisting of 2,358,708 sentences in Hindi and 109,855 sentences in Bengali from an online newspaper. The researchers found

that their approach achieved an accuracy of 43.20% and 42% for the Bengali corpora and an accuracy of 49.68% and 46.25% for the Hindi corpora for constraint and unconstraint submission, respectively. Kaur et al. [36] introduced a new approach for SA in Hinglish. They collected a dataset of 100 positive and 100 negative MRs to test their approach. They used unigrams, bigrams, and trigrams for feature extraction and applied four classification methods: SVM, NB, LR, and NN. Sharma and Moh [37] conducted SA on Hindi tweets to predict the results of the Indian elections. They collected 42,235 tweets and used both supervised and unsupervised approaches in their analysis. Their results showed that the NB and SVM approaches predicted a win for the BJP, while a dictionary-based approach predicted a win for the INC. The SVM approach had the highest accuracy, achieving a rate of 78.4% among the three methods used. Sharma et al. [38] conducted SA on Hindi using a modified subjectivity lexicon. To obtain data for analysis, they collected a dataset of 50 Hindi tweets from Twitter with hashtags ''JAIHIND'' and ''WORLDCUP2015''. The researchers compared the accuracy of their modified lexicon with a method using unigrams and found that the modified lexicon performed better. They achieved an accuracy of 73.53% and 81.97% for the respective hashtags. Puri and Singh [39] introduced an HTC-SVM model for Hindi document classification. Their study utilized a dataset consisting of four Hindi documents categorized into bipolar categories. Notably, their proposed model achieved a remarkable accuracy, indicating a perfect classification performance. Soni and Selat [40] conducted a study on the classification of general headlines in Hindi, collected from multiple news websites. They constructed a dataset consisting of 10,000 rows and proceeded with preprocessing techniques. Feature Engineering (FE) was then applied using the TF-IDF approach. Subsequently, several classifiers, including SVM, RF, LR, and NB, were employed for classification. The results revealed that SVM outperformed the other classifiers, achieving an accuracy of 80.18%. Conversely, NB demonstrated the lowest accuracy, reaching only 53.08%. Sharma and Ghose [41] put forward a majority voting ensemble model for binary SC of Hindi MRs. They compiled a dataset of 1200 MRs in Hindi, excluding objective or neutral reviews. After pre-processing and feature extraction through TF-IDF, they applied various baseline classifiers, such as SVM, DT, NB, KNN, LR, RF, and AdaBoost (AB), followed by the proposed voting ensemble. Results showed that their voting ensemble of classifiers based on maximum voting outperformed all individual classifiers and achieved a reasonably good performance. The proposed Voting model achieved the highest accuracy of 88%. The researchers' future plans involve using a stacking ensemble technique and expanding the dataset size.

SC has been one of the most escalating tasks in the era of NLP since the last decade. The research into Hindi text classification is limited compared to the research in a Western language like English, as seen from the literature review. The literature on Hindi SA primarily focuses on sentence-level analysis, with limited attention given to document-level SA. In a research-deficient scenario, most researchers have utilized MLAs for SC, with SVM and NB being the most commonly employed methods. While some studies have employed LBAs, ensemble learning techniques have not been extensively utilized for Hindi SA. Furthermore, the majority of the existing work is domain-specific, highlighting the need for domain-independent approaches. Another challenge arises from the usage of private datasets created from online platforms, which often contain Hinglish (a mix of Hindi and English) instead of pure Hindi dataset. This scarcity of linguistic resources, including datasets of sufficient size and quality, poses a significant challenge for Hindi SA. For this review of literature in SC of MRH, we can conclude that publicly available datasets in the movie domain are very sparse, making the development of a movie review dataset critical to carry out studies in this language. Therefore, in the present work, we have presented author made HLMR dataset. Compared with other research domains, MRH still have significant scope of improvement. To fill this research gap, this paper proposes SEBA for SC of MRH and introduces HLMR dataset with adequate quality and quantity. Present work distinguishes itself from the existing works in the following ways. It is known that the performance of machine learning classifiers (MLCs) can be enhanced by hyperparameter tuning and ensemble learning. This paper considers both ways of improving the performance of MLCs which have yet to be considered so far in Hindi SC literature. To the author's knowledge, ours is the first work that uses handcraft-based features such as TF-ISF and word-level N-gram features for SC in a resource restraint scenario. This paper also provides a comprehensive understanding of SC of Hindi reviews; and also identifies the challenges and issues encountered while working with RRL-Hindi. A concise summation of some literature review done is given in Table. 1.

IV. MATERIAL AND METHODS

A. DETAILS OF MRH COLLECTION AND FORMATION OF HLMR DATASET

The primary issue with Hindi research is the procurement of accurately labeled datasets. Due to the unavailability to open access gold-standard adequate dataset, researchers who wish to conduct SA on HLT prefer to build their dataset. A sufficient dataset with the required quantity and quality is a prerequisite for developing an efficient supervised ML model that is not prone to overfitting. Generally, a dataset can be comprised of written or spoken bodies and can be open or closed. Currently, in this work, the HLMR dataset is comprised of written MRH and is of closed nature as it is related to the movie domain and is developed for SC. MRH from various recognized and widely known websites such as [1,2,3,4,5]

[1]https://navbharattimes.indiatimes.com/movie-masti/movie-review/
[2]https://www.aajtak.in/entertainment/film-review
[3]https://hindi.webdunia.com/bollywood-movie-review/
[4]https://hindi.filmibeat.com/reviews/
[5]https://www.amarujala.com/entertainment/movie-review

TABLE 1. A concise summation of some literature Review done.

Techniques Used	SA Level	Domain Oriented & Category	Reference
Voting Ensemble, SVM, DT, NB, KNN, LR, RF and AB	Sentence	Yes, 1200 MRs	[41]
RF, SVM, NB and LR	Sentence	No, General News Headlines	[40]
HTC- SVM	Document	No, Hindi Documents	[39]
H-SWN, Word Sense Disambiguation and Hindi Subjectivity lexicon	Sentence	Yes, 700 Hindi book reviews	[34]
LBA, MLA and Hybrid	Sentence	Yes, Hindi tweets of MRs	[33]
NB, SVM and MAXNET	Sentence	Yes, 2K MRs	[7]
KNN, DT, NB, RF and, NN	Sentence	Yes, 250 MRs	[29]
RF, SVM and Hindi Senti Word Net	Sentence	Yes, MRs	[22]

were collected. The Unicode UTF-8 standard recognizes and reads Hindi reviews by machine. All the collected reviews were annotated with two opposing positive or negative polarities. The objective reviews were not taken into consideration. An MRH is classified as positive if it contains a feeling of excitement, delight, triumph, optimism, confidence, faith, appreciation, etc. [42]. The (interesting), "सराहनीय" (commendable), "उल्लेखनीय" (notable), "शानदार" (fabulous), "अच्छी" (good), "मसाला" (Spice), "रंग" (colors), "दिलचस्पी" (interest), "मनोरंजन" (entertainment), "पसंद" (like), "हल्की-फुल्की" (light-hearted), "नयापन" (novelty), "मजेदार" (interesting) etc. In contrast, MRH will be classified as negative if it contains negative terms or expresses negative feelings such as regret, anger, sorrow, violence, etc. The following terms are very frequent in negative MRH such as "रंगहीन" (colorless), "फीकी" (spice less), "सिरदर्द" (headache), "कमजोर" (weak), "निराश" (disappointment), "उबाऊ" (boring), "औसत" (average), "घटिया" (rubbish), "समस्या" (problem), "बुरी" (bad) etc.

There is a total of 10,000 MRH in the HLMR dataset, and it is a CSV file comprising two columns, namely MRH and label annotation. The MRH contains the review text, while the label annotation column contains the binary sentiment polarity – positive or negative. Later, each MRH in the HLMR dataset was given to two language experts to confirm the polarity annotation label. Cohen's Kappa evaluated the annotation quality from sklearn.metrics.cohen_kappa_score, which came out to be ∼85%, which is equivalent to perfect agreement. The most used Inter annotator agreement (IAA) in two annotators-based work is the kappa coefficient [43] which is calculated as mentioned in "(1)":

$$k = (Pa - Pc)/(1 - Pc) \tag{1}$$

Pa represented the proportion of cases where both the annotators agree on sentiment polarity.

Pc represented the proportion of cases, where both the annotators agree by chance.

We have received the perfect agreement level in our case. The HLMR dataset comprised a maximum of twenty-five reviews per movie. HLMR dataset has no standard train test split; hence 10-fold cross-validation is used. Some of the MRH of the HLMR dataset is shown in the Table. 2.

B. STATISTICS OF DEPLOYED DATASET

In this work, we have used two well-known publicly available datasets, the IIT-P movie, and the IIT-P product review, as mentioned in [44], in addition to the HLMR dataset. Since our motive is binary SC of reviews in Hindi, only positive and negative polarity reviews are considered. A brief statistical description of the deployed datasets is mentioned in the Table. 3.

C. CHALLENGES WITH THE DEPLOYED DATASETS

The MRs posted online are not very formal, lack solid grammatical structures, and are dirty, informal, and unstructured. Therefore, their preconditioning and pre-treatment are required to convert them into a suitable form; if left untreated, they might result in low classification performance. This issue is looked after by the pre-processing step, as mentioned in sub-section V-A of section V. Another challenge with the deployed datasets is finding the appropriate feature set for efficient classification, as in the deployed datasets, there is a dearth of distinctive and discriminative features; this issue is taken care by the text representation and feature extraction step as mentioned in subsection V-B of section V.

V. PROPOSED METHODOLOGY

Recently there has been a good hike in the usage of Hindi while writing MRs on various websites. With the expeditious growth of HLT in the form of reviews over the web, the demand for automatic classification of review data has increased by leaps and bounds. Therefore, it is necessary to have an efficient way to predict the writer's sentiments that are expressed in a written MR. Individuals face many decisions daily; one of the common decisions is to decide whether or not a film or movie should be seen; SA or SC of MRH can automate the process of coming to a decision based on others' opinions, for instance, classifying the reviews into

TABLE 2. Some of MRH in HLMR dataset.

MRH	Label Annotation
शादियों को सफल बनाने की दिलचस्प कहानी जुग जुग जियो। {Interesting story of making marriages successful is Jug Jug Jiyo.}	Positive
फिल्म शाबाश मिथु इरादे से नेक फिल्म है। {The film Shabaash Mithu is a noble film with intentions.}	Positive
कुमुद मिश्रा का किरदार जरूर करीने से लिखा गया है और उनकी मेहनत उनके किरदार को फिल्म का सबसे अच्छा किरदार बना भी देती है। {Kumud Mishra's character has definitely been written neatly and her hard work also makes her character the best character of the film.}	Positive
फ्रीकी अली में एक भी ऐसा कारण नहीं है जिसके लिए टिकट खरीदा जाए। {There isn't a single reason to buy a ticket to Freaky Ali.}	Negative
सिटिलाइट्स इंसानी जज्बातों और मानवता से भरी एक बहुत ही खूबसूरत फिल्म है। {Citylights is a very beautiful film full of human emotions and humanity.}	Positive
फिल्म एबीसीडी दर्शकों को अंत तक बांधे रखती है और अंत में दर्शकों के दिल में एक जीत और खुशी का एहसास छोड़ जाती है। {The film ABCD keeps the audience hooked till the very end and at the end leaves a feeling of victory and happiness in the heart of the audience.}	Positive
फिल्म 420 आईपीसी देखना विशुद्ध रूप से समय की बर्बादी है। {Watching 420 IPC movie is purely wastage of time.}	Negative
फिल्म बेलबॉटम की सबसे कमजोर कड़ी फिल्म का संगीत है। {The weakest link of the film Bellbottom is the music of the film.}	Negative
फिल्म शेरशाह की सबसे कमजोर कड़ी है इसकी पटकथा। {The weakest link of the film Sher Shah is its screenplay.}	Negative
फिल्म बंटी और बबली 2 उत्तीर्ण है मनोरंजन तो इसमें भरपूर है। {The movie Bunty Aur Babli 2 has been a success, there is a lot of entertainment in it.}	Positive
कुल मिलाकर फिल्म तेरा सुरूर का सुरूर नहीं चढ़ता। {Overall, the film Tera Suroor doesn't justify its title.}	Negative
कहानी रोचक है फिल्म चेहरे की। {The story of the film Chehre is interesting.}	Positive

two opposing categories. The followed methodology and proposed framework for the classification of MRH into bipolar

TABLE 3. Statistical description for the deployed datasets.

Dataset	Parameter	Description
HLMR Dataset	Dataset size	1.98 MB
	Domain oriented	yes
	No. of positive reviews	5,744
	No. of negative reviews	4,256
	Review language	Hindi (Devanagari)
IIT-P Movie Review Dataset	Dataset size	284 KB
	Domain oriented	yes
	No. of positive reviews	823
	No. of negative reviews	530
	Review language	Hindi (Devanagari)
IIT-P Product Review Dataset	Dataset size	668 KB
	Domain oriented	yes
	No. of positive reviews	2,290
	No. of negative reviews	712
	Review language	Hindi (Devanagari)

polarity categories can be well explained from the flowchart as shown in Fig 1. and involves four main modules, namely, Online MRH collection and polarity annotation module as mentioned in section IV, pre-processing module, text representation and feature extraction module, model development and sentiment classification module and lastly, the performance evaluation module. A detailed explanation of each module is given in the subsections below.

A. PRE-PROCESSING OF HINDI REVIEW TEXT

The MRHs posted online are not very formal reviews but lack strong grammatical structures and are dirty, informal, and unstructured. Preliminary or pre-processing is needed to prepare the corpus before applying any algorithm [45]. The primary processing helps reduce the dataset's dimension, vocabulary size, and memory overhead, resulting in better classification performance. Non-discriminative words, the words that occur with similar frequency in both positive and negative reviews, were eliminated. Generic stop words were removed, while domain-specific stop words were left untreated [46]. Punctuation and special character removal are done. Non- Hindi words, emoticons, and numbers are substituted by their textual counterparts/word equivalents. Negation words – the logical complement are the words that change the polarity, thus the sentence's meaning. Hence, they were left untreated, as removing them would affect the model's

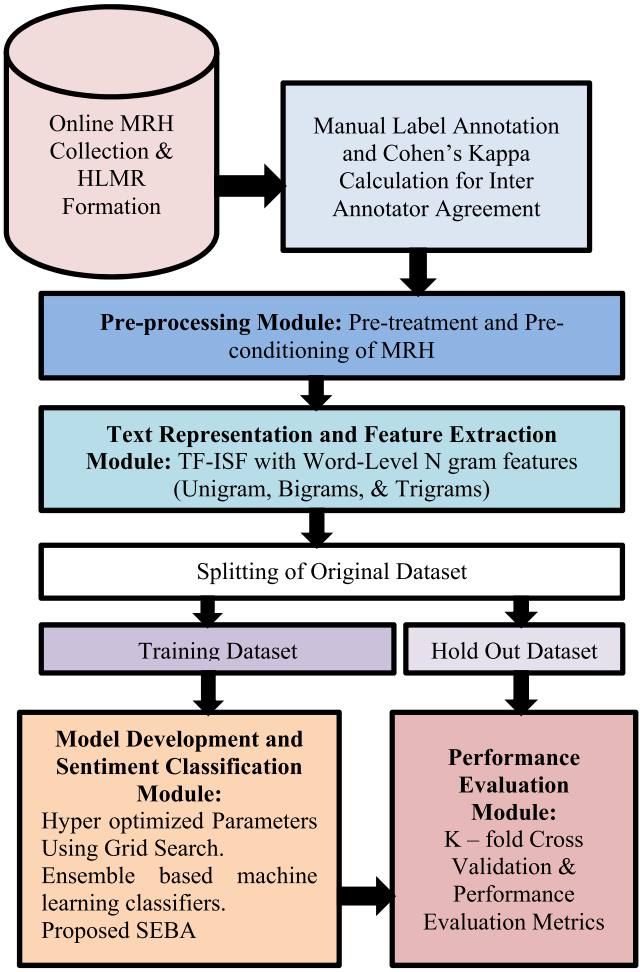

FIGURE 1. Basic Framework followed for Bipolar SC of MRH.

performance. Some examples of negation words are as follows: नही {nahee}{no}, न {na}{no}, कुछ नहीं {kuch nahi}{nothing}, and कभी नहीं {kabhi nahi} {never}etc. Movie Review tokenization is the process of separating the review sentences into word groups; it is done to extract relevant information from review sentences. Review sentences are divided into their constituent words by using spaces as separators. Subsequently, tokenization is performed by removing the individual words from the review text. The tokens need to be encoded to integer values when working with MLAs. Therefore, TF-ISF is employed for the same.

B. TEXT REPRESENTATION AND FEATURE EXTRACTION

Following the preliminary processing, the next step is to analyze the review sentence to find relevant features that may affect the polarity of reviews. MRH in Hindi lacks solid grammatical structure and uses informal jargon. Therefore, for MRH representation and relevant feature extraction, handcrafted features such as Term frequency–inverse sentence frequency (TF-ISF) along with word level N-gram features (Unigram, bigram & trigram) have been deployed [47], [48]. Word-level N-gram features are a straightforward, easily implemented, data-driven approach to building MLCs. Word-level N-gram model is utilized as a feature along with TF-ISF. The grams are the text terms where n-grams find the adjacent terms of n terms from the dataset [49], [50]. N can be any number. Current work is confined to unigram, bigram, and trigram only. Unigram means the gram size of one, i.e., single terms; bigram means two adjacent terms are taken together, i.e., the pair of consecutive words, while trigram means three adjacent words. The term weighting (TW) schemes are considered for the information retrieval process meaning; thereby, the prime function of term weight (TW) is to measure the salient features of the term in a dataset. Therefore, datasets are ranked based on their words or weight. TF-ISF has been explained beneath.

TF-ISF: The significance of a word (*wr*) in a review dataset depends upon its frequency inside a review sentence. The *TF*is mentioned in "(2)".

$$TF(wr, rs) = \text{number of times a word w' appears in review sentences } rs'. \quad (2)$$

Evaluation of whole document for significance of a word w' can be determined through *ISF* i.e., *ISF (wr)* given in "(3)".

$$ISF(wr) = log(|RS|)/NRS \quad (3)$$

The total number of review sentences in the dataset is represented by $|RS|$.

Where *NRS (w)* determines the number of review sentences in which work w' appears.

The *TF-ISF* highlights' importance of a word inside a review sentence of review dataset as given in "(4)".

$$TF - ISF(wr', rs) = TF(wr, rs\ )ISF(wr) \quad (4)$$

C. MODEL DEVELOPMENT AND SENTIMENT CLASSIFICATION

The HLMR corpus is pre-processed, prepared, cleaned, and normalized; relevant features are extracted for model development and sentiment classification. As stated before, the performance of MLAs can be enhanced by using the following two strategies - Hyperparameter Tuning and ensemble learning methods. In the present work, both techniques are utilized initially. Some SOTA classifiers were applied, and their parameters were optimized using grid search. Second ensemble-based classifiers were applied.

D. HYPERPARAMETER TUNING WITH GRID SEARCH

Various MLAs were applied, and their hyperparameters were fine-tuned using Grid search [51]. Hyperparameter refers to characteristics external to the model, and their values cannot be guesstimated from data. Their values are always set before the process of learning begins. It is known that the performance of MLAs highly depends on the value of

hyperparameters. Grid search is a way of tuning hyperparameters by finding the optimal value of hyperparameters in models. It has been observed that there exists no way by which we get to know the optimal values of hyperparameters in advance. So, we have to apply all possible combinations to find the optimal value manually; doing this is very time-consuming; therefore, we used GridSearchCV, which comes with sklearn.model_Selection package. This automated our hyperparameter tuning process; it coils through various predefined hyperparameters and fits them into our model. So, by the end, we can choose the best parameter out of the list of stated hyperparameters [52].

E. FOLLOWING ARE MLC WHOSE HYPERPARAMETERS WERE TUNED WITH GRID SEARCH (GS) [53]

- **Decision Trees (DT):** DT is a non-parametric supervised MLC that can be employed for both classification and regression problems. A tree-like structure follows if-else conditions to visualize and classify the data. The DT classifier can be imported from sklearn.tree class. The intent is to create a tree that predicts the value of the target label or class, in our case, the sentiment class of MRs, using decision rules deduced from features in data. This classifier suffers from high variance, meaning thereby it is very much sensitive to its training data. If there is a minute change in underlying data, then resultant tree changes also prediction of model changes sharply. The value of the hyperparameter Max Depth, Max leaf nodes, and Max features are set using GS.
- **K-Nearest Neighbors (KNN):** KNN is a lazy, non-parametric, supervised MLC that can be utilized for solving classification and regression problems. KNN makes use of nearest neighbor or feature similarity. This algorithm is lazy in the sense that it lacks training. All the data is used for making the prediction. At its basic level, it is based on the principle that similar points lie in close proximity. It can be imported from sklearn.neighbors class. In KNN only parameter is K, and its optimal value we calculated using GS.
- **Support Vector Machine Radial Basis Function (SVM-RBF):**It is a very commonly used supervised MLC that is used to perform classification. SVM can be imported from sklearn.svm.SVC class. SVM aims to separate data points into two classes using a hyperplane to correctly place any new data points to any of these two classes in the future. It is used primarily for linearly separable problems and non-linearly separable problems by transforming non-linearly separable problems into linearly separable problems using a kernel function. Here RBF kernel is used, the default kernel of sklearn's SVM classification algorithm. Different kernel functions such as RBF, linear, and polynomial were tried using GS; also, the optimum value of gamma and c was found using GS.
- **Multinomial Naïve Bayes (MNB):** MNB can be imported from the class sklearn.naive_bayes .MultinomialNB. It is a classification algorithm based on the Bayesian learning approach. In the present work, the algorithm guesses the polarity of the annotation label using the Bayes theorem. It calculates each label's likelihood for a given MRH and outputs the annotation polarity label with the greatest chance. It consists of several algorithms that all have one common thing: each feature being classed is independent and unrelated to any other feature. The alpha value was set to one, and classes_ were assigned to two.
- **Logistic Regression (LR):** LR is the most straightforward technique and offers efficacy for various classification tasks. It is low in variance. It is mainly used when the dependent or response variable is categorical. In binary LR, the response variable can be of two types; in our movie review case, it can be negative (0) or positive (1). It uses functions to deduce relationships among dependent and independent variables by prognosticating probabilities, and these obtained probabilities can be converted into binary values for further predictions. The optimal value of c was obtained through GS.

F. BY EMPLOYING ENSEMBLE LEARNING (EL) BASED CLASSIFIERS

The committee of different people often performs better than the individual expert when making real-life decisions. Likewise, in EL methods, committees of efficient classifiers are applied to a problem. Each constituent classifier of the committee will make a prediction, and their prognosis will be combined to predict the outcome. A committee of classifiers, when chosen wisely, can bring different expertise and reduce errors under the averaging effect. Integrating many MLCs to form a single robust classifier model is called EL. The MLCs combined to form a committee are called base estimators, also called weak learners. We get a single, efficient, robust, better-performing classifier with reduced errors through EL. There are multiple classifiers based on EL; the details of some of the deployed classifiers are given below [54], [55]:

- **Random Forest (RF):** RF is an ensemble of DTs, meaning it has the easiness of DTs along with the power of the ensemble method. In RF, bagging is implemented internally. It is a non-parametric supervised MLC that reduces the overfitting problem of DTs. Ensembling is implemented by combining decisions from multiple DTs and by taking an average of n-estimates, the number of DTs in a forest. It is used for both classification and regression problems. It can be imported from the class sklearn.ensemble.RandomForestClassifier.
- **Extra-Trees (ET):** ET can be imported from sklearn.ensemble.ExtraTreesClassifier class. ET is an ensemble-based classifier based on DTs. It randomizes certain decisions and data subsets to minimize overfitting. The node splitting is done based on random splits among the random subsets of selected features. This randomness comes from random splits of all observations.

- **Adaboost (AB):** Adaptive Boosting is abbreviated as AdaBoost and is an extensively used iterative ensemble method. It randomly chooses training data and repetitively trains by choosing training data based on prior's training's correct prediction. In this algorithm, the order does matter. The errors in the first step influence how the second step proceeds, and so on, until as many errors as possible have been considered.
- **Gradient Boosting Machine (GBM):** Boosting is a method of incorporating weak learners into a better-performing model. The core idea is to use the most informative data for training every individual weak learner. GBM primarily uses DTs as weak learners. The idea is to train the second model on a gradient of the error concerning the loss predictions of the first model. This way, multiple simple models compensate for each other's weaknesses to better fit the model.
- **eXtreme Gradient Boosting (XGB):** XGBoost, abbreviated as XGB, is an optimized distributed version of GBM designed for efficient and scalable training of MLCs. It is an ensemble learning method that combines the predictions of multiple weak models to produce a more robust prediction, and it usually works with an ensemble of DTs. It is efficient at handling large datasets and usually achieves SOTA performance in classification tasks. The issue with GBM as it is very slow at computing outputs, while XGB has built-in support for parallel processing, making it possible to train models on large datasets in less time.

G. PROPOSED SEBA ARCHITECTURE

We present our proposed solution SEBA for doing binary SC of Hindi reviews of the movie domain using a stacked ensemble of models comprising five independent classifiers refer to Fig. 2. The concept of stacking is based on the wisdom of multi-fold classifiers at different levels to boost predictive capabilities. A layering concept is used for making predictions in the stacked generalization method.

There can be many layers in stacked models, but two-layered stacked models are commonly used as many levels add complexity. Several weak learners or primary classifiers (s1, s2 . . . . . .sn) are employed in the first layer. In SEBA, ET, XGB, SVM, MNB, and LR are used as weak learners at layer 1. It has been observed that less related MLCs produce better results. Usually, the whole dataset is split into training and testing. The training dataset is divided into G portions (G1, G2, G3, ... Gn).

First, weak or primary classifier s1 is trained on g-1 portions, making predictions on the gth portion. This process is repeated until predictions have been made on all g portions. Likewise, the same procedure is repeated for all weak or primary classifiers placed at layer 1. The predictions generated by all weak classifiers are stacked together. These predictions are input to the layer two classifiers, which are meta classifiers. In the case of classification, probabilities or class labels are predicted by the weak classifiers. The predictions from weak classifiers are used as features to fit the meta classifier placed at layer 2, in our case LR. In stacking, cross-validation is used to avoid overfitting.

The meta classifier at layer 2 uses both predictions from weak classifiers and predictions made on original data to make final predictions. The strength of stacking lies in the meta classifier, which learns the strengths and weaknesses of models at level 1 and rationally combines their predictions to get the final result [54], [56].

Applying different classification techniques at various stages of sentiment label prediction may give a better overall result than using the same classifier at all stages.

The proposed stacking ensemble model is a two-layer architecture. The first layer comprises ET, XGB, SVM, MNB, and LR as estimators, and predictions from the first layer are used as features for the second layer estimators.

Second-layer estimators comprise LR, which is also a meta-learner. Lastly, predictions from the first layer are fed to the second layer estimator- meta learner LR. Meta learner makes the final prediction of the movie review label. It was found that more than one classification technique used at one stage allows verification and validation of class label prediction of Hindi reviews generated by incumbent methods. Algorithm for proposed SEBA is stated beneath.

Algorithm 1 Proposed SEBA With G – Fold Cross-Validation

Input: MRH Set MR = (mr1, mr2, mr3, mr4, mr5, mr6, mr7, mr8, mr9, mr10, ... , mr1000);
Polarity label class set Plabel = (p1, p2, p3, p4, p5, p6, p7, p8, p9, ... , pn);
Weak learner set W = (ET, XGB, SVM, MNB, LR);
Output: Predicted binary sentiment polarity label for MRH
1: SEBA S
2: Adopt Cross Validation approach in preparing a training set of weak learners
3: Randomly split MR into G equal size subsets: MR = {MR1, MR2, MR3, MR4, ... , MRg}
4: for g ← 1 to G do
5: Step 1: Learn Weak learners at level 1
6: for n ← 1 to K do
7: Learn a stacker G from MR/MRg
8: end for
9: end for
10: Step 2: Learn meta level estimator: LR
11: Step 3: Re-train weak learners
12: for n ← 1 to K do
13: Train an estimator gk based on MR
14: end for
15: return G (mr) = g'(g1(mr), g2(mr), g3(mr), g4(mr), g5(mr), g6(mr), ... , gp(mr))

H. PERFORMANCE EVALUATION METRICS

To gauge the performance of the proposed architecture and the applied classifier, we measure their efficacy in terms of

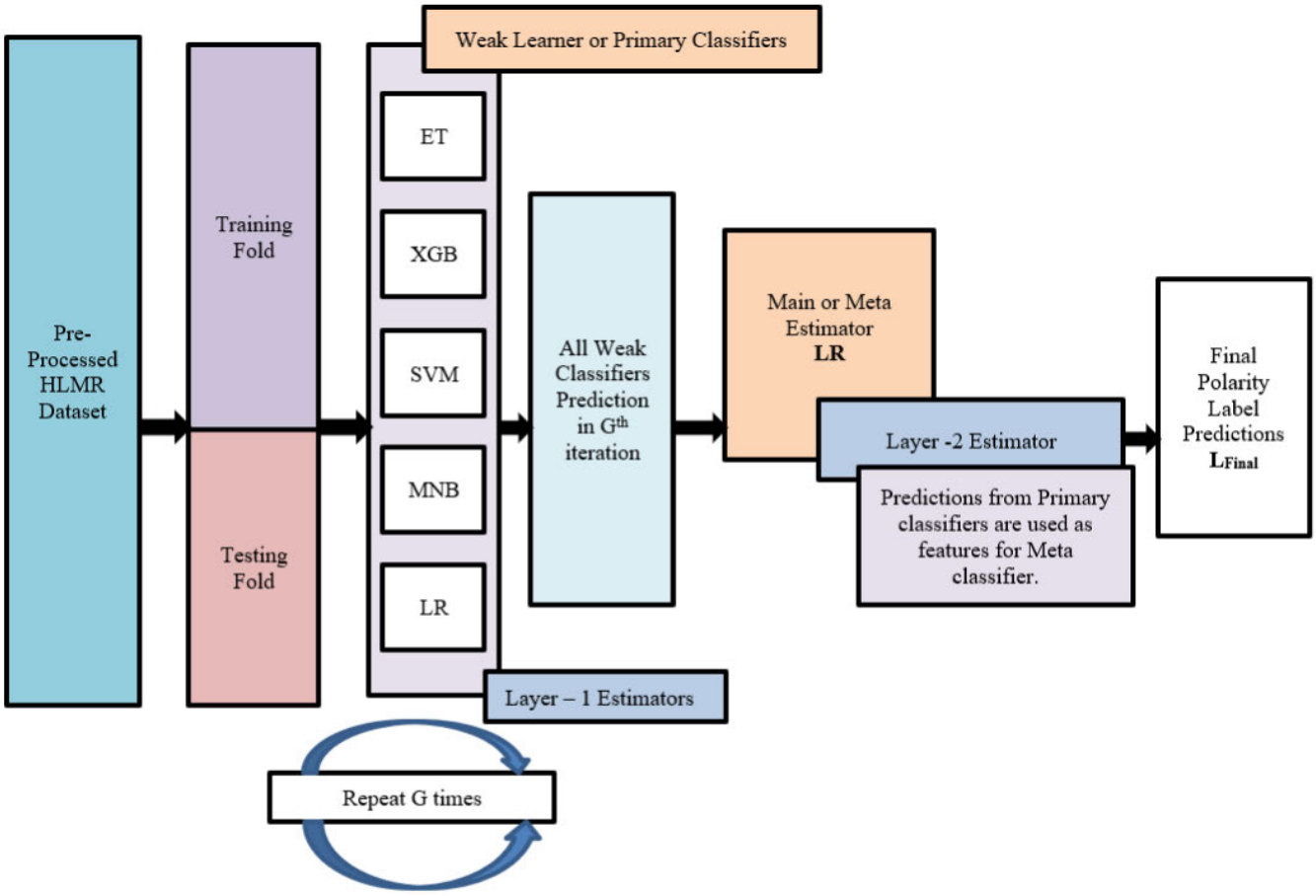

FIGURE 2. Proposed SEBA for SC of movie reviews in Hindi.

TABLE 4. Confusion matrix.

Sentiment Polarity Prediction	Actual Sentiment Polarity	
	True	False
Positive	TP	FP
Negative	TN	FN

accuracy, F measure, recall, precision, AUC-ROC score and MCC score [56], [57].

The confusion matrix is also known as the contingency table as mentioned in Table 4.

True Positive (TP) counts the correctly assigned MRs to the positive polarity class.

False Positive (FP) counts the incorrectly assigned MRs to the positive polarity class.

False Negative (FN) counts the positive polarity class's incorrectly rejected MRs.

True Negative (TN) counts the correctly rejected MRs from the positive polarity class.

Accuracy refers to the percentage of correctly classified reviews from both positive and negative classes.

Precision measures how exact the classifier is.

*Recall*measures the completeness of the classifier.

The F1 score is the harmonic mean of precision and recall.

AUC-ROC score determines the classifier's efficiency. The higher the score, the better the model's performance differentiating the positive and negative polarity classes.

MCC score. MCC stands for Matthew's correlation coefficient. This metric evaluates classification performance when the number of instances in two classes is different, i.e., imbalanced.

VI. RESULTS AND DISCUSSIONS

This section presents the experiments' findings and details the software and hardware specifications used. The experiments were conducted on a computing device with an Intel Core i7 processor and 16 GB of RAM. The implementation was done using Python 3.11.0. This section focuses on the empirical findings of the experiments conducted for SC of MRs in Hindi. Our work is limited by the lack of existing research on SC of Hindi, particularly in the movie review domain, and the complexities of the Hindi language itself. The availability of a correct annotated dataset with the required quality and quantity of reviews is another challenge. Additionally,

imbalanced labels in the dataset can lead to biased results. We created the HLMR dataset of 10,000 MRs for coarse-grained sentence-level SC to overcome these challenges. We achieved an almost balanced distribution for both positive and negative polarity class labels in this dataset to avoid the divergence of experimental results. All experiments were conducted on the IIT-P movie and product review datasets for further validation. Our research aimed to develop a machine learning-based solution for binary SC of Hindi reviews. We began with pre-treatment and pre-conditioning of Hindi reviews in the pre-processing module and used handcrafted features such as TF-ISF with word-level N-gram features like unigram, bigram, and trigram for feature extraction and text representation. After that, we applied various SOTA classifiers, including DT, KNN, SVM, MNB, and LR, and used GS to hyper tuned their parameters. The results of these experiments were good. We then experimented with ensemble-based classifiers like RF, ET, AB, GBM, and XGB and observed their competence. This led us to propose the SEBA architecture, which is composed of diverse and complementary machine learning classifiers (MLCs), and we applied this architecture to Hindi review datasets using 10-fold cross-validation to obtain reliable results. In this study, the authors computed the values of different evaluation metrics, including accuracy, recall, F1 score, precision, ROC_AUC score, and MCC score for all the datasets employed.

Previous studies have demonstrated that ensemble methods like stacking can improve the performance of individual MLCs, overcome overfitting problems, and handle imbalanced class label problems if present in a dataset. Therefore, stacking multiple MLCs in the SEBA architecture can lead to a more robust prediction model. Stacking can also handle the imbalance of class-label problems by providing potentially more diverse, robust, and independent sets of predictions. Additionally, stacking is a good way of combining models of different types in such a way as to result in low variance. Finally, bagging and boosting models can also be integrated with stacking models to make them even more robust.

The experiments conducted in this study have shown that ensemble-based classifiers are highly effective. Based on this finding, we proposed the SEBA architecture, which consists of multiple complementary and diverse MLCs. This architecture was applied to the Hindi review datasets described in Section V. In stacking, various classifiers create an improved model by offsetting each other's biases and weaknesses. This leads to the better overall efficacy of the prediction model.

The results of all the individual classifiers applied along with the proposed SEBA on HLMR dataset are shown in Table 5. The subscript + GS implies that the hyperparameter tuning uses grid search.

According to the detailed classification results in Table 5, the SEBA model outperformed all the classifiers used in terms of accuracy (Acc), recall (Rec), F1 score, precision (Pre), and ROC_AUC score when unigrams with TF-ISF were used as features. Among the hyper-optimized classifiers, SVM achieved the highest performance, while ET achieved the highest performance among the ensemble-based classifiers. The experiments were repeated for bigrams and trigrams with TF-ISF as features, and SEBA outperformed all the classifiers in the bigram case, achieving the highest Acc of 0.741%, F1 score of 0.737%, Rec of 0.741%, Pre of 0.740%, and ROC_AUC score of 0.727. Similarly, SEBA outperformed the classifiers in the trigram case, achieving the highest Acc of 0.654%, F1 score of 0.609%, Rec of 0.654%, Pre of 0.686%, and ROC_AUC score of 0.609. Figure 3 shows a bar graph comparing the accuracy and F1 score of all feature sets used, while Figure 4 compares the recall and precision scores of all feature sets used. Figures 5 and 6 imply that the proposed architecture achieves the highest ROC_AUC Score in all the feature sets applied and the Best MCC score in unigram with the TF-ISF feature set.

TABLE 5. Summarizes the performance evaluation of applied classifiers and proposed SEBA on HLMR Dataset with Unigram with TF-ISF as features.

CLASSIFIERS	ACC	F1	REC	PRE	ROC_AUC
DT + GS	0.724	0.704	0.724	0.747	0.689
KNN + GS	0.587	0.579	0.587	0.643	0.614
SVM + GS	0.794	0.790	0.794	0.798	0.776
MNB + GS	0.749	0.740	0.749	0.756	0.724
LR + GS	0.783	0.780	0.783	0.784	0.768
RF	0.787	0.785	0.787	0.787	0.774
ET	0.798	0.796	0.798	0.798	0.787
AB	0.746	0.740	0.746	0.748	0.725
GBM	0.754	0.741	0.754	0.771	0.724
XGB	0.778	0.774	0.778	0.779	0.761
PROPOSED	**0.808**	**0.807**	**0.808**	**0.808**	**0.799**

To validate the results obtained on the HLMR dataset and to show domain adaptability, we evaluate SEBA and all the classifiers applied on two publicly available IIT-P movie and product review datasets.

Based on the results presented in Table 6, the proposed SEBA outperformed all the classifiers applied in the case of unigram with TF-ISF as feature sets for the IIT-P movie review dataset. Additionally, for bigram with TF-ISF features, SEBA performed equally well as the best-performing classifier, ET, obtaining an Acc of 0.742%, F1 score of 0.726%, Rec of 0.742%, a Pre of 0.740%, and ROC-AUC score of 0.687. Furthermore, when trigram with TF-ISF features was applied, SEBA outperformed all the classifiers and achieved the highest Acc of 0.745%, F1 score of 0.726%, Rec of 0.745%, Pre of 0.740%, and ROC-AUC score of 0.687. The comparison of accuracy & F1 score and Recall & precision score on all feature sets used is depicted in Fig. 9 and Fig. 10, respectively.

Moreover, from the results presented in Fig. 7 and Fig. 8, it can be inferred that the proposed architecture achieved the

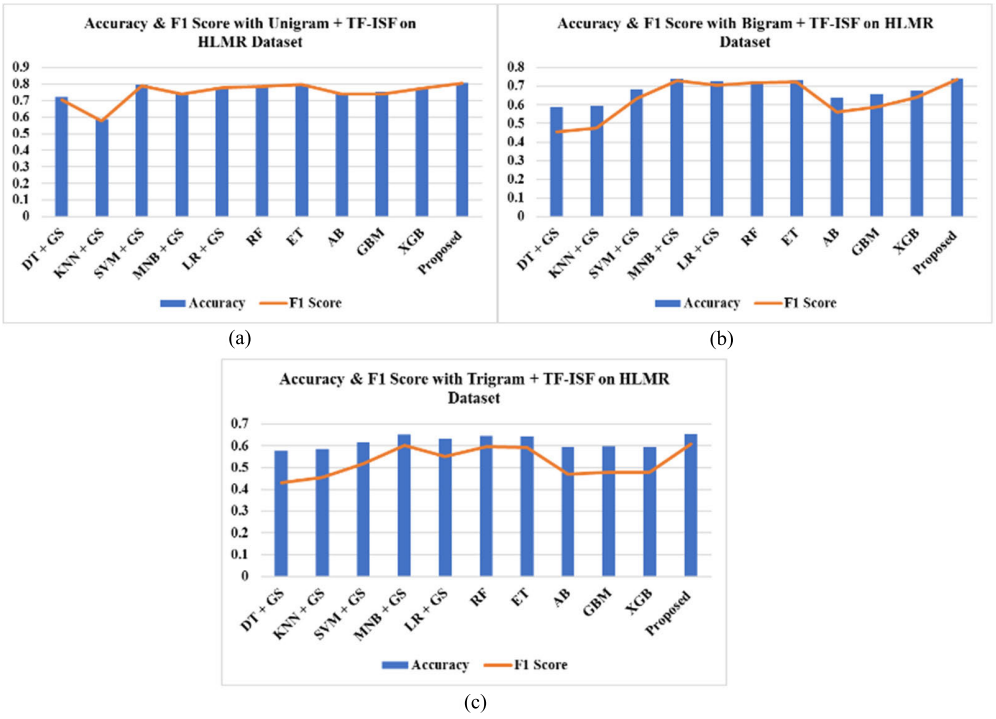

FIGURE 3. Accuracy and F1 score based comparison on HLMR dataset for TF-ISF with (a) Unigram (b) Bigram (c) Trigram.

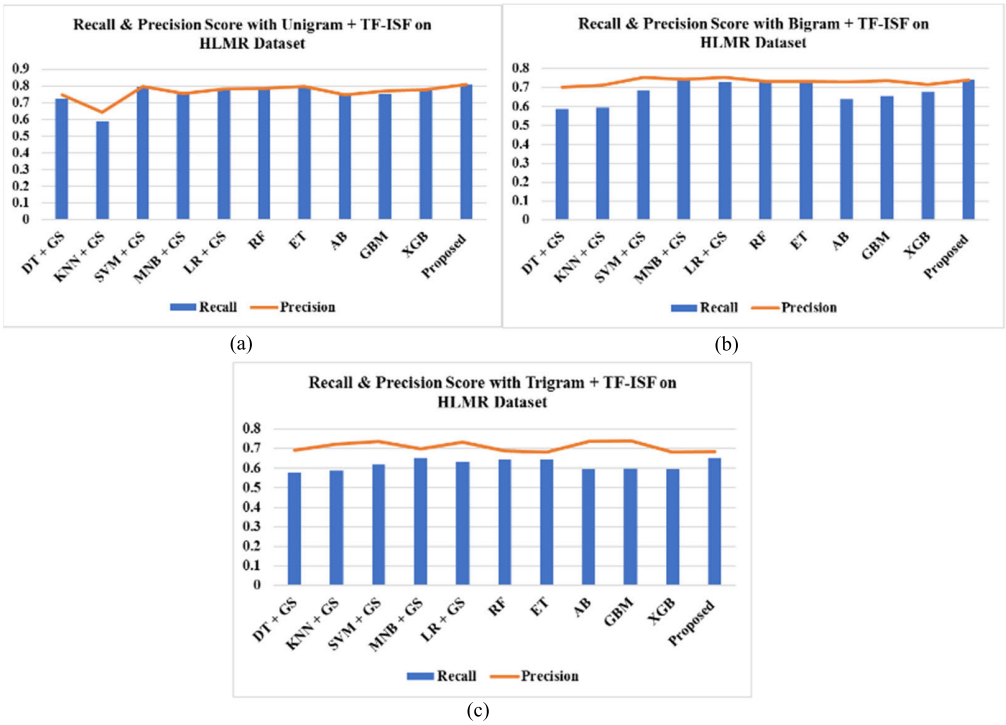

FIGURE 4. Recall and precision score based comparison on HLMR dataset for TF-ISF with (a) Unigram (b) Bigram (c) Trigram.

highest ROC_AUC Score across all the feature sets used and the Best MCC score in the case of unigram with TF-ISF feature set.

Again, the experiment was performed on the IIT-P product review dataset; the product dataset in Hindi is used to check the domain independence. Results show SEBA outperformed

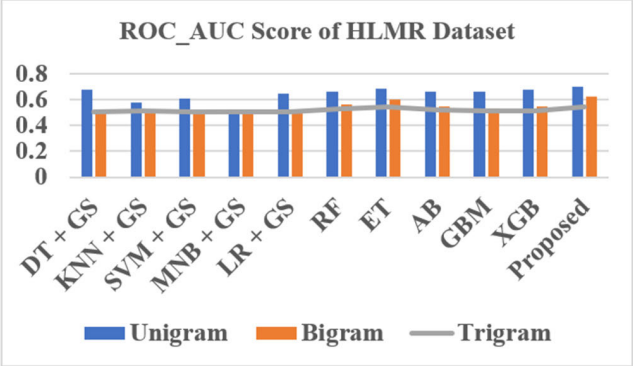

FIGURE 5. ROC_AUC Score based comparison on all three word level N gram features.

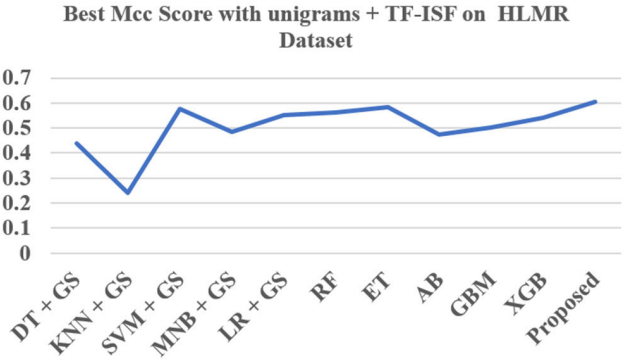

FIGURE 6. MCC Score based of proposed SEBA and other classifiers applied.

TABLE 6. Summarizes the performance evaluation of applied classifiers and proposed seba on IIT-P movie review dataset with Unigram with TF-ISF as features.

Classifiers	Acc	F1	Rec	Pre	ROC_AUC
DT + GS	0.716	0.685	0.716	0.722	0.642
KNN + GS	0.519	0.483	0.519	0.708	0.604
SVM + GS	0.716	0.677	0.716	0.735	0.635
MNB + GS	0.698	0.668	0.698	0.695	0.626
LR + GS	0.706	0.680	0.706	0.703	0.638
RF	0.734	0.722	0.734	0.729	0.685
ET	0.744	0.734	0.744	0.739	0.700
AB	0.710	0.697	0.710	0.702	0.661
GBM	0.716	0.685	0.716	0.722	0.642
XGB	0.724	0.714	0.724	0.717	0.679
PROPOSED	**0.748**	**0.740**	**0.748**	**0.743**	**0.707**

all the other individual classifiers applied for unigram case with TF-ISF features as shown in Table 7. In bigram with TF-ISF case, proposed SEBA obtained the highest Acc of 0.816%, F1 score of 0.784%, Rec of 0.816%, Pre of 0.797%, and ROC-AUC of 0.620. In comparison, it outperformed all

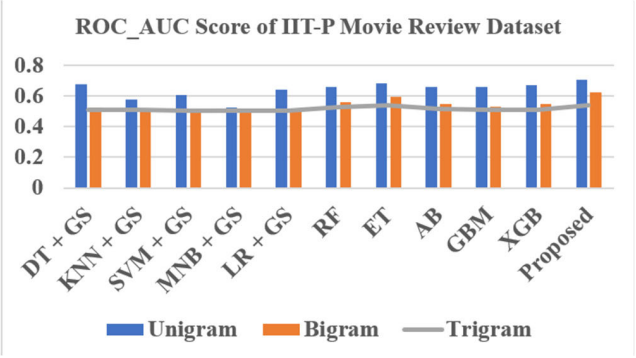

FIGURE 7. ROC_AUC score based comparison on all three-word level N Gram features.

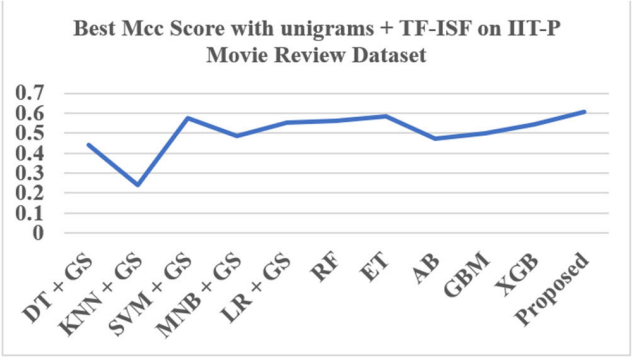

FIGURE 8. MCC score based comparison of proposed SEBA and other classifiers applied.

TABLE 7. Summarizes the performance evaluation of applied classifiers and proposed SEBA ON IIT-P product review dataset with Unigram with TF-ISF as features.

CLASSIFIERS	ACC	F1	REC	PRE	ROC_AUC
DT + GS	0.817	0.805	0.817	0.801	0.676
KNN + GS	0.792	0.750	0.792	0.755	0.574
SVM + GS	0.821	0.777	0.821	0.821	0.603
MNB + GS	0.791	0.713	0.791	0.773	0.524
LR + GS	0.829	0.800	0.829	0.818	0.642
RF	0.831	0.808	0.831	0.817	0.660
ET	0.835	0.818	0.835	0.822	0.681
AB	0.799	0.787	0.799	0.782	0.656
GBM	0.827	0.804	0.827	0.811	0.656
XGB	0.821	0.806	0.821	0.804	0.671
PROPOSED	**0.837**	**0.825**	**0.837**	**0.824**	**0.701**

the applied classifiers regarding F1 and ROC_AUC scores for trigrams with TF-ISF features. However, SEBA performed equally well as the best-performing classifier, ET, regarding accuracy and recall. The ROC-AUC and best MCC score in the case of unigram with TF-ISF feature set is shown in Fig 13 and Fig 14, respectively.

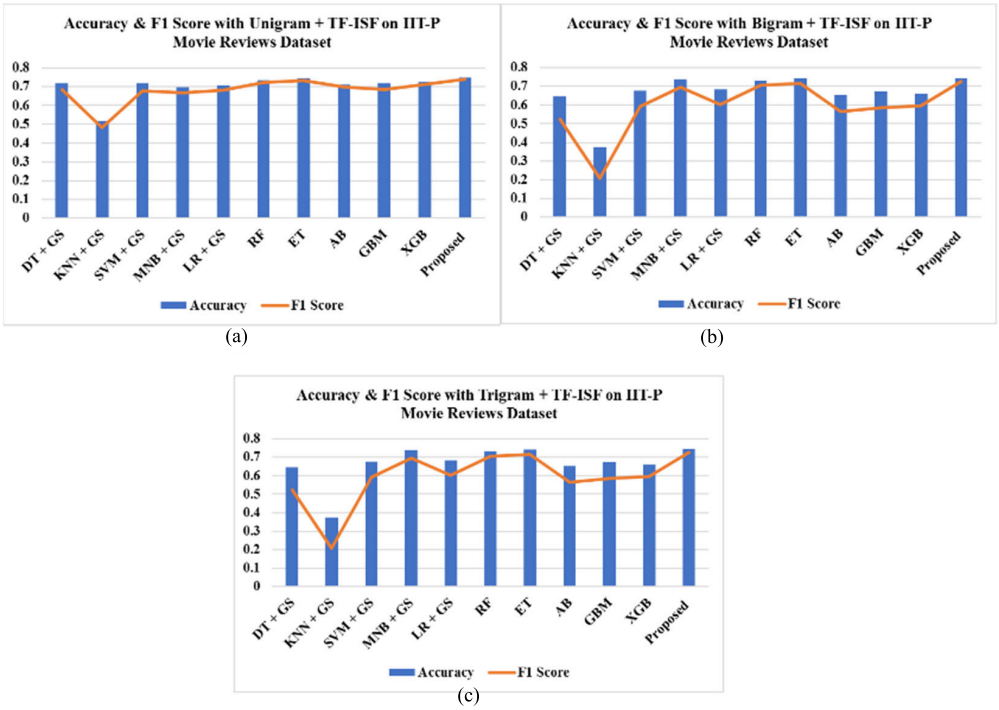

FIGURE 9. Accuracy and F1 score based comparison on IIT-P movie review dataset for TF-ISF with (a) Unigram (b) Bigram (c) Trigram.

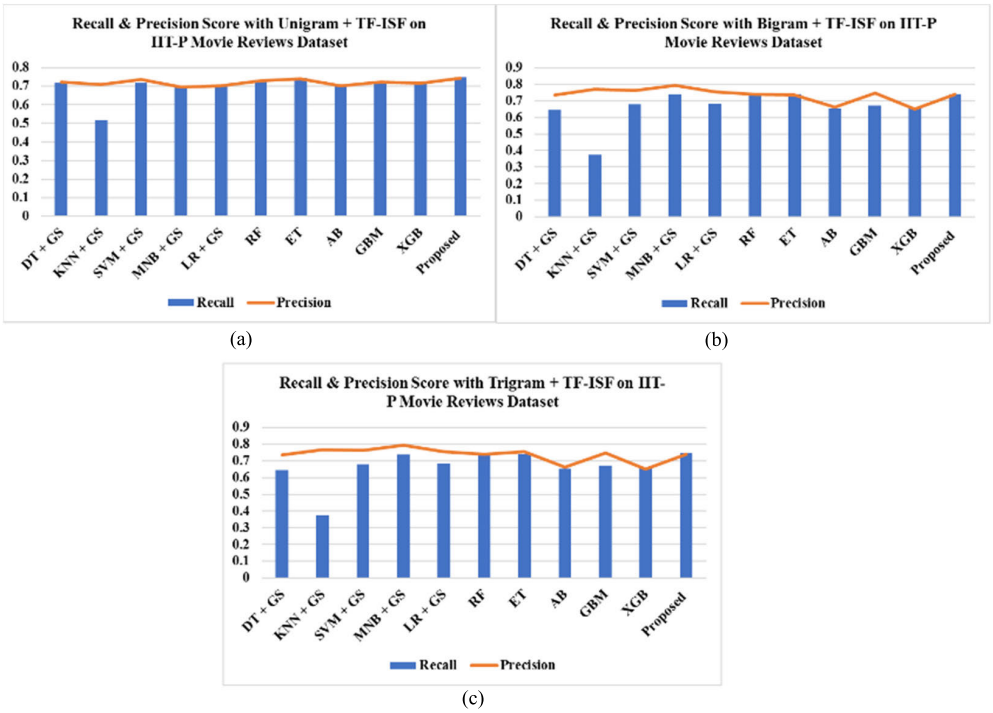

FIGURE 10. Recall and precision score based comparison on IIT-P movie review dataset for TF-ISF with (a) Unigram (b) Bigram (c) Trigram.

The overall results suggest that SEBA performs reasonably well across all three datasets, although the specific results differ. When analyzing the author-made HLMR dataset, which is of sufficient size, SEBA outperformed all the individual classifiers applied in all features. Similarly, in the case of the IIT-P movie review dataset, SEBA outperformed

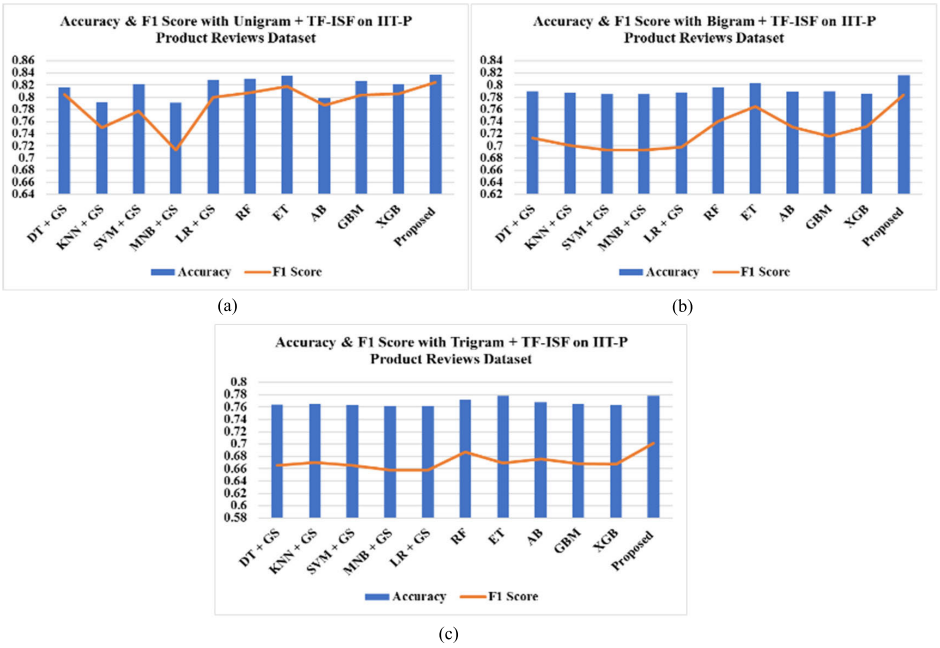

FIGURE 11. Accuracy and F1 score based comparison on IIT-P product review dataset for TF-ISF with (a) Unigram (b) Bigram (c) Trigram.

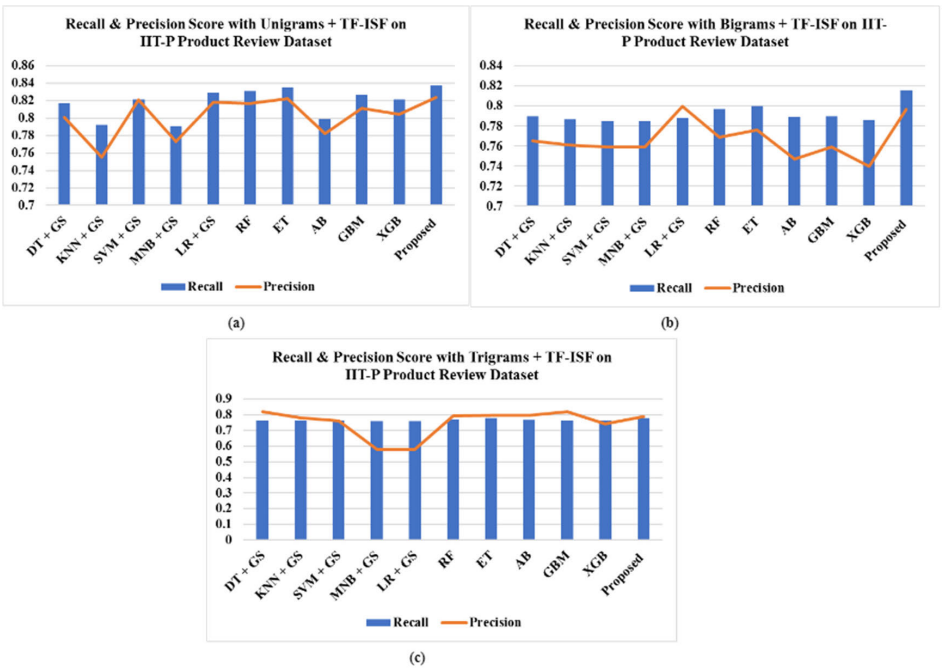

FIGURE 12. Recall and precision score based comparison on IIT-P product review dataset for TF-ISF with (a) Unigram (b) bigram (c) Trigram.

all the individual classifiers used for unigram and trigram with TF-ISF features, while for bigrams with TF-ISF case, it performed equally well as the best-performing classifier, ET.

This variation in performance can be attributed to the dataset's size and imbalanced polarity labels, which were not present in the HLMR dataset.

Based on the study's results, it was observed that specific individual classifiers, including SVM, ET which is DTs based ensemble, MNB, XGB an enhanced version of GBM, and LR, outperformed others in terms of all quality measures. As a result, the proposed SEBA model was developed using these classifiers. The experiments conducted on Hindi reviews indicated that SEBA achieved superior performance in all quality

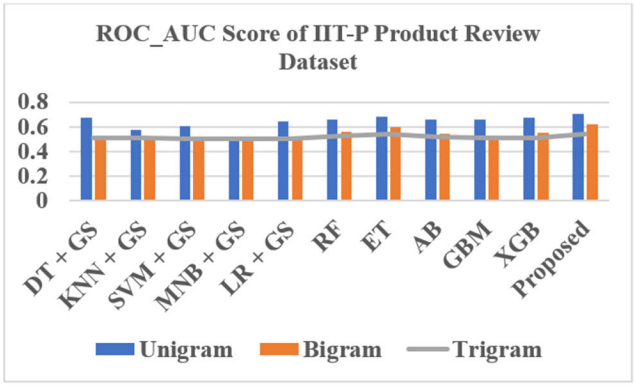

FIGURE 13. ROC_AUC score-based comparison on all three-word level N gram features.

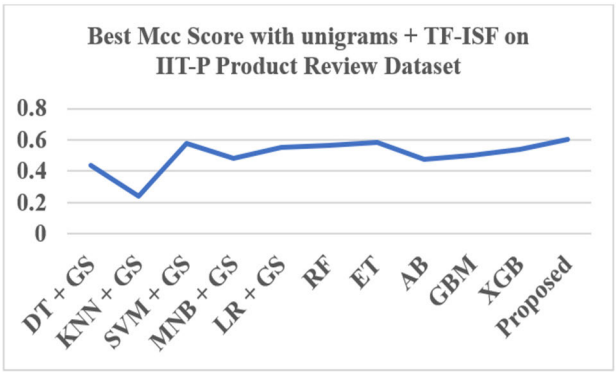

FIGURE 14. MCC Score-based comparison of proposed SEBA and other classifiers applied.

measures across all three datasets, regardless of their size. The TF-ISF feature set with unigrams performed the best, suggesting that SEBA is well-suited for coarse-grained SC; of Hindi reviews and could potentially be deployed online for binary review classification. These findings provide a foundation for future research in this area.

Another observation made in the study was that even when higher word-level N-gram features were used, applying SEBA did not significantly improve the quality measures. This was attributed to the TF-ISF feature set, including unigrams, bigrams, and trigrams. Using this feature set could assign a lower weight to Hindi polarity-bearing words, which carry sentiment and frequently appear in Hindi reviews. This could explain why applying SEBA to higher word-level N-gram features did not yield significant improvements in the results.

In addition, SEBA was found to be highly efficient, as it does not require any additional computational resources and is effective in addressing overfitting issues. The proposed model is expected to aid viewers and reviewers in evaluating online Hindi MRs and determining whether a movie is worth watching. The model can be easily deployed for binary review classification and can prove helpful in various applications, including online movie recommendation systems. The authors hope their work will inspire further research in this field and advance SA in low-resource languages, thus filling a critical research gap.

VII. CONCLUSION AND FUTURE WORKS

SC is a challenging task, especially for RRL, like Hindi. In this study, we proposed a simplistic yet powerful SEBA for classifying Hindi MRs according to their polarity. SEBA is an ensemble method that combines multiple MLAs to achieve high classification performance. Our approach in the present work initially involved applying various SOTA classifiers with hyperparameter tuning and ensemble-based classifiers, followed by the proposed architecture. We are the first to use hand craft-based feature TF-ISF and word-level N-gram features consisting of unigram, bigram, and trigram for text representation and feature extraction when doing Hindi SC. We argue that in SC for an RRL – Hindi, even a minute enhancement in performance requires a complex model architecture. SEBA addresses this problem by providing a simplistic yet powerful solution.

SEBA is efficient, easily implemented, requires less computational resources, and is suitable for overcoming overfitting problems. Our proposed model achieved SOTA results on the HLMR dataset and showed promising results on the IIT-P Movie and product datasets, indicating its effectiveness in analyzing binary sentiment in Hindi review text. Our work contributes to the field by identifying the challenges encountered while working with RRL-Hindi and proposing an approach for bipolar SC of MRs. The proposed model can be easily deployed online for binary review classification and can be helpful for various applications such as online movie recommendation systems. It is important to note that there is minimal work on Hindi as movie review datasets are private. Therefore, our work is significant as it provides a valuable resource for SA in Hindi movie reviews. We hope that our work will inspire further research in this field and help advance SA in low-resource languages, and thus our proposed solution fills a critical research gap. We plan to extend our work to other domains like tourism reviews and reviews related to food, health, education, and other low-resource Indian languages. Our proposed solution can be deployed online for binary review classification, providing a foundation for future research. We plan to propose a stacking ensemble comprising machine and deep learning models and work on character-level features in the future. We also aim to consider negation handling, which has been ignored in the present work. A further paper incorporating all these is planned for future work. Overall, our proposed SEBA model shows great potential for SC in Hindi and could be helpful in various applications such as e-commerce, social media analysis, and public opinion mining.

REFERENCES

[1] A. Sharma and U. Ghose, "Sentimental analysis of Twitter data with respect to general elections in India," *Proc. Comput. Sci.*, vol. 173, pp. 325–334, Jan. 2020, doi: 10.1016/j.procs.2020.06.038.

[2] Q. A. Xu, V. Chang, and C. Jayne, "A systematic review of social media-based sentiment analysis: Emerging trends and challenges," *Decis. Anal. J.*, vol. 3, Jun. 2022, Art. no. 100073, doi: 10.1016/j.dajour.2022.100073.

[3] A. Khan, M. A. Gul, M. Zareei, R. R. Biswal, A. Zeb, M. Naeem, Y. Saeed, and N. Salim, "Movie review summarization using supervised learning and graph-based ranking algorithm," *Comput. Intell. Neurosci.*, vol. 2020, pp. 1–14, Jun. 2020, doi: 10.1155/2020/7526580.

[4] D. S. Kulkarni and S. S. Rodd, "Sentiment analysis in Hindi—A survey on the state-of-the-art techniques," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 21, no. 1, pp. 1–46, Nov. 2021, doi: 10.1145/3469722.

[5] M. Ceyhan, Z. Orhan, and D. Karras, "An approach for movie review classification in Turkish," *Eur. J. Formal Sci. Eng.*, vol. 4, no. 2, pp. 56–65, Sep. 2021, doi: 10.26417/328uno67t.

[6] T. Sharma and K. Kaur, "Aspect sentiment classification using syntactic neighbour based attention network," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 2, pp. 612–625, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.005.

[7] S. Sangam and S. Shinde, "Sentiment classification of social media reviews using an ensemble classifier," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 16, no. 1, pp. 355–363, Oct. 2019, doi: 10.11591/ijeecs.v16.i1.

[8] G. Mesnil, T. Mikolov, M. Ranzato, and Y. Bengio, "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews," 2014, *arXiv:1412.5335*.

[9] A. Sharma and U. Ghose, "Lexicon a linguistic approach for sentiment classification," in *Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2021, pp. 887–893, doi: 10.1109/Confluence51648.2021.9377057.

[10] V. Gupta, N. Jain, S. Shubham, A. Madan, A. Chaudhary, and Q. Xin, "Toward integrated CNN-based sentiment analysis of tweets for scarce-resource language—Hindi," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 20, no. 5, pp. 1–23, Jun. 2021, doi: 10.1145/3450447.

[11] S. Rani and P. Kumar, "A journey of Indian languages over sentiment analysis: A systematic review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1415–1462, Aug. 2019, doi: 10.1007/s10462-018-9670-y.

[12] S. R. Shah and A. Kaushik, "Sentiment analysis on Indian indigenous languages: A review on multilingual opinion mining," 2019, *arXiv:1911.12848*.

[13] A. Joshi, "Towards sub-word level compositions for sentiment analysis of Hindi-English code-mixed text," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, Dec. 2016, pp. 2482–2491. Available: https://aclanthology.org/C16-1234/.

[14] Md S. Akhtar, A. Ekbal, and P. Bhattacharyya, "Aspect based sentiment analysis in Hindi: Resource creation and evaluation," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, May 2016, pp. 2703–2709. [Online]. Available: https://aclanthology.org/L16-1429

[15] S. Singh, R. Panjwani, A. Kunchukuttan, and P. Bhattacharyya, "Comparing recurrent and convolutional architectures for English-Hindi neural machine translation," in *Proc. 4th Workshop Asian Transl. (WAT)*, Nov. 2017, pp. 167–170. [Online]. Available: https://aclanthology.org/W17-5717

[16] P. Verma, S. Pal, and H. Om, "A comparative analysis on Hindi and English extractive text summarization," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 18, no. 3, pp. 1–39, May 2019, doi: 10.1145/3308754.

[17] K. V. Kumar, D. Yadav, and A. Sharma, "Graph based technique for Hindi text summarization," in *Proc. Inf. Syst. Design Intell. Appl., Proc. 2nd Int. Conf. INDIA* Jan. 2015, pp. 301–310, doi: 10.1007/978-81-322-2250-7_29.

[18] V. Singh, D. Vijay, S. S. Akhtar, and M. Shrivastava, "Named entity recognition for Hindi-English code-mixed social media text," in *Proc. 7th Named Entities Workshop*, 2018, pp. 27–35. [Online]. Available: https://aclanthology.org/W18-2405

[19] V. Kumar, A. Verma, N. Mittal, and S. V. Gromov, "Anatomy of preprocessing of big data for monolingual corpora paraphrase extraction: Source language sentence selection," in *Emerging Technologies in Data Mining and Information Security* (Advances in Intelligent Systems and Computing), vol. 814. Singapore: Springer, 2019, pp. 495–505, doi: 10.1007/978-981-13-1501-5_43.

[20] S. S. Alotaibi and C. W. Anderson, "Extending the knowledge of the Arabic sentiment classification using a foreign external lexical source," *Int. J. Natural Lang. Comput.*, vol. 5, no. 3, pp. 1–11, Jun. 2016, doi: 10.5121/ijnlc.2016.5301.

[21] M. Korayem, K. Aljadda, and D. Crandall, "Sentiment/subjectivity analysis survey for languages other than English," *Social Netw. Anal. mining*, vol. 6, pp. 1–17, Sep. 2016, doi: 10.1007/s13278-016-0381-6.

[22] C. Nanda, M. Dua, and G. Nanda, "Sentiment analysis of movie reviews in Hindi language using machine learning," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2018, pp. 1069–1072, doi: 10.1109/ICCSP.2018.8524223.

[23] V. Jha, N. Manjunath, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, "HOMS: Hindi opinion mining system," in *Proc. IEEE 2nd Int. Conf. Recent Trends Inf. Syst. (ReTIS)*, Jul. 2015, pp. 366–371, doi: 10.1109/ReTIS.2015.7232906.

[24] D. Mumtaz and B. Ahuja, "Sentiment analysis of movie review data using senti-lexicon algorithm," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 592–597, doi: 10.1109/ICATCCT.2016.7912069.

[25] M. Galvao and R. Henriques, "Forecasting model of a movie's profitability," in *Proc. 13th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2018, pp. 1–6, doi: 10.23919/CISTI.2018.8399184.

[26] V. Jha, N. Manjunath, P. D. Shenoy, and K. R. Venugopal, "Sentiment analysis in a resource scarce language: Hindi," *Int. J. Sci. Eng. Res.*, vol. 7, no. 9, pp. 968–980, Sep. 2016.

[27] A. Kaur and A. P. Nidhi, "Predicting movie success using neural network," *Int. J. Sci. Res. (IJSR)*, vol. 2, no. 9, pp. 69–71, Sep. 2013.

[28] N. Quader, Md. O. Gani, and D. Chaki, "Performance evaluation of seven machine learning classification techniques for movie box office success prediction," in *Proc. 3rd Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2017, pp. 1–6, doi: 10.1109/EICT.2017.8275242.

[29] A. Kanitkar, "Bollywood movie success prediction using machine learning algorithms," in *Proc. 3rd Int. Conf. Circuits, Control, Commun. Comput. (I4C)*, Oct. 2018, pp. 1–4, doi: 10.1109/CIMCA.2018.8739693.

[30] T. G. Rhee and F. Zulkernine, "Predicting movie box office profitability: A neural network approach," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2016, pp. 665–670, doi: 10.1109/ICMLA.2016.0117.

[31] K. Korovkinas and P. Danenas, "SVM and Naïve Bayes classification ensemble method for sentiment analysis," *Baltic J. Modern Comput.*, vol. 5, no. 4, pp. 398–409, Dec. 2017, doi: 10.22364/bjmc.2017.5.4.06.

[32] A. Sharma and S. Dey, "A boosted SVM based ensemble classifier for sentiment analysis of online reviews," *ACM SIGAPP Appl. Comput. Rev.*, vol. 13, no. 4, pp. 43–52, Dec. 2013, doi: 10.1145/2577554.2577560.

[33] A. Madan and U. Ghose, "Sentiment analysis for Twitter data in the Hindi language," in *Proc. 11th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Mar. 2021, pp. 784–789, doi: 10.1109/Confluence51648.2021.9377142.

[34] F. Hussaini, S. Padmaja, and S. Sameen, "Score-based sentiment analysis of book reviews in Hindi language," *Int. J. Natural Lang. Comput.*, vol. 7, no. 5, pp. 115–127, Oct. 2018, doi: 10.5121/ijnlc.2018.7511.

[35] A. Kumar, S. Kohail, A. Ekbal, and C. Biemann, "IIT-TUDA: System for sentiment analysis in Indian languages using lexical acquisition," in *Proc. Int. Conf. Mining Intell. Knowl. Explor.*, Jan. 2016, pp. 684–693, doi: 10.1007/978-3-319-26832-3_65.

[36] V. Mangat, "Dictionary based sentiment analysis of Hinglish text," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 816–855, Jun. 2017.

[37] P. Sharma and T. Moh, "Prediction of Indian election using sentiment analysis on Hindi Twitter," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1966–1971, doi: 10.1109/BigData.2016.7840818.

[38] Y. Sharma, V. Mangat, and M. Kaur, "A practical approach to sentiment analysis of Hindi tweets," in *Proc. 1st Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Sep. 2015, pp. 677–680, doi: 10.1109/NGCT.2015.7375207.

[39] S. Puri and S. P. Singh, "An efficient Hindi text classification model using SVM," in *Computing and Network Sustainability* (Lecture Notes in Networks and Systems), vol. 75. Singapore: Springer, 2019, pp. 227–237, doi: 10.1007/978-981-13-7150-9_24.

[40] V. K. Soni and S. Selot, "A comprehensive study for the Hindi language to implement supervised text classification techniques," in *Proc. 6th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Oct. 2021, pp. 539–544, doi: 10.1109/ISPCC53510.2021.9609401.

[41] A. Sharma and U. Ghose, "Voting ensemble-based model for sentiment classification of Hindi movie reviews," in *Computational Intelligence* (Lecture Notes in Electrical Engineering), vol. 968. Singapore, Springer, 2023, pp. 473–483, doi: 10.1007/978-981-19-7346-8_40.

[42] N. Capuano, L. Greco, P. Ritrovato, and M. Vento, "Sentiment analysis for customer relationship management: An incremental learning approach," *Int. J. Speech Technol.*, vol. 51, no. 6, pp. 3339–3352, Jun. 2021, doi: 10.1007/s10489-020-01984-x.
[43] H. Rahab, A. Zitouni, and M. Djoudi, "SANA: Sentiment analysis on newspapers comments in Algeria," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 33, no. 7, pp. 899–907, Sep. 2021, doi: 10.1016/j.jksuci.2019.04.012.
[44] M. S. Akhtar, A. Kumar, and A. Ekbal, "A hybrid deep learning architecture for sentiment analysis," Presented at the 26th Int. Conf. Comput. Linguistics, Technical Papers, Dec. 2016. [Online]. Available: https://aclanthology.org/C16-1047
[45] G. Vinodhini and R. M. Chandrasekaran, "A comparative performance evaluation of neural network based approach for sentiment classification of online reviews," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 28, no. 1, pp. 2–12, Jan. 2016, doi: 10.1016/j.jksuci.2014.03.024.
[46] R. Rani and D. K. Lobiyal, "Performance evaluation of text-mining models with Hindi stopwords lists," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2771–2786, Jun. 2022, doi: 10.1016/j.jksuci.2020.03.003.
[47] J. Singh, G. Singh, R. Singh, and P. Singh, "Morphological evaluation and sentiment analysis of Punjabi text using deep learning classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 33, no. 5, pp. 508–517, Jun. 2021, doi: 10.1016/j.jksuci.2018.04.003.
[48] T. P. Sahu and S. Ahuja, "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms," in *Proc. Int. Conf. Microelectron., Comput. Commun. (MicroCom)*, Jan. 2016, pp. 1–6, doi: 10.1109/MicroCom.2016.7522583.
[49] M. A. Dootio and A. I. Wagan, "Development of Sindhi text corpus," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 33, no. 4, pp. 468–475, May 2021, doi: 10.1016/j.jksuci.2019.02.002.
[50] M. D. Ali Awan, S. Ali, A. Samad, N. Iqbal, M. M. S. Missen, and N. Ullah, "Sentence classification using N-grams in Urdu language text," *Sci. Program.*, vol. 2021, pp. 1–11, Nov. 2021, doi: 10.1155/2021/1296076.
[51] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.
[52] S. M. Alzanin, A. M. Azmi, and H. A. Aboalsamh, "Short text classification for Arabic social media tweets," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6595–6604, Oct. 2022, doi: 10.1016/j.jksuci.2022.03.020.
[53] M. Bilal, H. Israr, M. Shahid, and A. Khan, "Sentiment classification of roman-urdu opinions using Naïve Bayesian, decision tree and KNN classification techniques," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 28, no. 3, pp. 330–344, Jul. 2016, doi: 10.1016/j.jksuci.2015.11.003.
[54] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 1–39, Feb. 2010, doi: 10.1007/s10462-009-9124-7.
[55] M. Thangaraj and M. Sivakami, "Text classification techniques: A literature review," *Interdiscipl. J. Inf., Knowl., Manage.*, vol. 13, pp. 117–135, May 2018.
[56] P. Balaji and D. Haritha, "An ensemble multi-layered sentiment analysis model (EMLSA) for classifying the complex datasets," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 3, pp. 185–190, 2023, doi: 10.14569/IJACSA.2023.0140320.
[57] A. Singh and A. Payal, "CAD diagnosis by predicting stenosis in arteries using data mining process," *Intell. Decis. Technol.*, vol. 15, no. 1, pp. 59–68, Mar. 2021, doi: 10.3233/IDT-200041.

ANKITA SHARMA received the B.Tech. degree in computer science and engineering from Maharshi Dayanand University (MDU), Haryana, in 2017, and the M.Tech. degree in computer science from the University School of Information, Communication and Technology (USICT), Guru Gobind Singh Indraprastha University (GGSIPU), New Delhi, in 2019, where she is currently pursuing the Ph.D. degree in computer science. She has authored two book chapters and presented her papers at various international conferences. Her research interests include machine learning, data mining, data analytics, text analysis, text classification, sentiment analysis, natural language processing, supervised learning, and ensemble learning. She received the Indraprastha Research Fellowship (IPRF) for the top two-ranker of the university.

UDAYAN GHOSE (Member, IEEE) received the master's degree in physics from Banaras Hindu University (BHU), in 1993, the M.Tech. degree in computer science from the Birla Institute of Technology, Jharkhand, in 2001, and the Ph.D. degree in information technology from the University School of Information, Communication and Technology (USICT), Guru Gobind Singh Indraprastha University (GGSIPU), New Delhi, in 2011. He is currently a Professor with USICT, GGSIPU. He is also the Director of the Centralized Career Guidance and Placement Cell (CCGPC), GGSIPU. He has more than 21 years of teaching experience. He has published more than 60 research papers in various international/national journals and conferences. His research interests include data analytics, soft computing, AI, machine learning, information theory, object-oriented, and visual programming.