IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 13 January 2024, accepted 1 February 2024, date of publication 6 February 2024, date of current version 16 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3363158

RESEARCH ARTICLE

SPSO-EFVM: A Particle Swarm Optimization-Based Ensemble Fusion Voting Model for Sentence-Level Sentiment Analysis

DIMPLE TIWARI[1,2], BHARTI NAGPAL[1], BHOOPESH SINGH BHATI[3], MANOJ GUPTA[4], (Member, IEEE), PANNEE SUANPANG[5], (Member, IEEE), SUJIN BUTDISUWAN[6], AND AZIZ NANTHAAMORNPHONG[7], (Member, IEEE)

[1]NSUT East Campus (formerly Ambedkar Institute of Advanced Communication Technologies & Research), GGSIPU, Delhi 110031, India
[2]ABES Engineering College, Ghaziabad 201009, India
[3]Indian Institute of Information Technology, Sonepat 131029, India
[4]Department of Electrical Engineering, Guru Ghasidas Vishwavidyalaya Bilaspur, Koni, Bilaspur, Chhattisgarh 495009, India
[5]Department of Information Technology, Faculty of Science and Technology, Suan Dusit University, Bangkok 10300, Thailand
[6]Faculty of Education, Metharath University, Bang Toei, Pathum Thani 12160, Thailand
[7]College of Computing, Prince of Songkla University, Phuket 83120, Thailand

Corresponding authors: Aziz Nanthaamornphong (aziz.n@phuket.psu.ac.th) and Manoj Gupta (manojgupta35@yahoo.co.in)

ABSTRACT Sentiment analysis has received incremental growth in recent years for emerging applications, including human-robot integration, social platforms monitoring, and decision-support systems. Several neural or transformer model-based solutions have been provided in the field of sentiment analysis that relies on the decision of a single classifier or neural model. These are erroneous to encode contextual information into appropriate dialogues and increase extra computational cost and time. Hence, we proposed a compact and parameter-effective Particle Swarm Optimization-based Ensemble Fusion Voting Model (PSO-EFVM) that exploited the combined properties of four ensemble techniques, namely Adaptive-Boost, Gradient-Boost, Random-Forest, and Extremely-Randomized Tree with Particle Swarm Optimization (PSO)-based hyperparameter selection. The proposed model is investigated on five cross-domain datasets after applying the foremost initialization and feature extraction using Information Gain (IG). It employs adaptive and gradient learning to incorporate the automatic attribute selection with the arbitrary loss function optimization. In short, a generalized two-block composite classifier is designed to perform context compositionality and sentiment classification. A population-based meta-heuristic optimization PSO is applied to each base ensemble learner that calculates weights for the best parameter selection. Comprehensive investigations of different domains reveal the superiority of the proposed PSO-EFVM over established baselines and the latest state-of-the-art models.

INDEX TERMS Natural language processing, sentiment analysis, adaptive-boosting, gradient-boosting, ensemble learning, particle swarm optimization.

I. INTRODUCTION

Sentiment analysis has gained attention among various computational linguistics and text mining researchers over the past few years. Sentiment analysis classifies text based on opinions, also known as opinion mining, opinion extraction, and effects analysis. With the development of information technology, users can now easily express their opinions and views on different domains using social platforms [1]. Analyzing and predicting the polarity of reviews is required to understand the societal trends. Sentiment analysis is helpful for companies to analyze the stock market demands and predict the value of stock market assets [2]. Additionally, it is a major part of the decision support systems [3]. The progress of businesses is completely dependent on the accurate decisions taken by the decision-makers [4]. Teaching and

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

learning methods are improving using students' feedback [5], and the government is using sentiment analysis to get a person's opinion for making the policies [6]. This field of study has expanded over the decades due to the large amount of text data stored on web 2.0, such as blogs, newsgroups, discussion forums, and social sites. Sentiment analysis is an approach which includes different kinds of tasks like subjectivity classification, sentiment classification, opinion summarization, and sentiment extraction [7]. To perform these tasks, researchers face many challenges like group detection, link prediction, finding an expert, studying trust and distrust, etc. [8]. The success of sentiment analysis depends on the features that are extracted from the text or reviews. Thus, Natural Language Processing (NLP) enriches various techniques related to feature extraction, i.e., Feature Vectorization, Term Frequency-Inverse Document Frequency (TF-IDF), Word Embedding (Global Vectors for Word Representation (GLoVe), WordToVec), and Topic Modeling. Machine learning is a suitable and popular technique that performs the sentiment analysis process effectively and is divided into three parts: supervised learning, unsupervised learning, and ensemble learning. Researchers can select any one from them to meet the needs of their work or proposed aim. In supervised learning, the text is classified based on pre-defined labels, and a mapping function is used to map the input with the output. Various techniques like Naïve Bayes (NB) [9], Support Vector Machine (SVM) [10], Logistic Regressor, and K-Nearest Neighbor (KNN) [11] are commonly applied by different researchers in the field of sentiment extraction. Unsupervised learning is based on previously undetected patterns without pre-existing labels, and the researchers also choose various clustering algorithms to perform sentiment extraction [12]. Deep learning is also a handy approach for solving different kinds of problems. Yutao et al. [13] used a Deep Neural Network (DNN) to develop a multiplex interaction service of cold-start recommendation, and [14] applied a DNN to secure intelligent grids with blockchain technology. Deep learning is a part of machine learning that effectively works with large datasets. However, as the research on sentiment analysis has increased, ensemble learning has come into demand, which increases the efficiency and accuracy of the sentiment analysis process. Ensemble methods are meta-algorithms that combine base algorithms into one predictive model to decrease bias (boosting), variance (bagging), or improve predictions. These approaches also improve machine learning techniques and provide better predictive performance than a single learner [15]. This work presents an ensemble model for the sentiment analysis, in which four base learners are combined to improve the performance of the model. Initially, two base learners of the bagging approach- Random-Forest and Extremely-Randomized Tree classifiers, and two base learners of the boosting approach- Adaptive-Boost and Gradient-Boost classifiers, were trained independently on five datasets. After that, the best parameter weights have been calculated by exploiting the PSO. Finally, a voting ensemble approach integrated all four ensemble methods to acquire more effective results. We combined ensemble learners to build a novel model instead of simple weak learners and tuned their parameters using the bio-mimetic optimization technique PSO. Five popular datasets, including Sentiment-140 Reviews, Reddit-App Reviews, Amazon-Shopping Reviews, Alexa-Product Reviews, and SMS-Collection Reviews, evaluated the proposed model's performance. Experimented results on the five cross-domain reviews datasets prove the efficiency of the proposed PSO-EFVM for the sentiment classification task. Integrating the four ensemble base learners by PSO selected hyperparameters is responsible for the promising results with low computational cost. The major contributions of this research are the following:

- A novel PSO-EFVM optimized model is proposed for effective sentiment analysis tasks. The proposed fusion model has the potential for four powerful learners to extract opinions from sentences.
- Advanced feature extraction techniques are exploited to enrich the raw text with utility information, and the IG feature selection approach reduces dimensionality.
- A bio-mimetic PSO technique is exploited to calculate the hyperparameter weights of Adaptive-boost, Gradient-boost, Random-Forest, and Extremely-Randomized Tree for supreme model training.
- Five cross-domain reviews datasets are investigated to check the authenticity of the proposed PSO-EFVM for sentiment classification, and results are measured in terms of accuracy, weighted precision, weighted-recall, weighted-f1-score, and ROC-AUC score.
- An extensive comparative discussion has been presented, including machine learning, deep learning, and transfer learning paradigms, to prove the authenticity of the novel model.

The remaining sections of this research work are organized as follows: Section II provides a literature review of existing sentiment analysis techniques. Section III describes information regarding the exploited preliminaries. Section IV presents the proposed methodology. Section V demonstrates the experimental results. Section VI presents a comparative study of the proposed model with other baselines and state-of-the-art models. Section VII concludes the work and suggests future research direction.

II. LITERATURE REVIEW

The sentiment analysis process is invented to extract the polarity from reviews, comments, and opinions, applied to product reviews from blogs, news feeds, and social networks. Due to the broad research in this area, Medhat et al. [16] presented an in-depth review paper on various sentiment analysis algorithms and applications. The central focus of this literature is to present the extent of the literature and the direction of research in the field of sentiment analysis using machine learning and deep learning. However, sentiment analysis can be applied using a Lexicon approach.

This work mainly focuses on machine learning-related sentiment analysis. When researchers first dove into this field, single classifiers were used for sentiment classification. For instance, Kang et al. [17] improved the NB algorithm to analyze restaurant reviews' sentiments. By using unigram and bigram as a feature, the gap between the positive and negative polarities was narrowed by about 3.6%, and accuracy was increased. Yang et al. [18] proposed a Support Vector Based (SVM) approach for emotion classification, including four parts: segmentation of words, word emotion establishment, training, and testing of the database. Accordingly, a model was established to forecast the reader's emotion. Samilovic et al. [19] performed sentiment analysis on tweets using SVM to categorize the tweets into negative, positive, and neutral categories and acquire improved predictive results.

Table 1 summarizes previously proposed ensemble methods related to sentiment analysis, notably those employing ensemble and deep learning models. Kontopoulas et al. [20] proposed an original ontology-based technique to increase the accuracy of sentiment analysis of Twitter posts. Tripathy et al. [21] classified a movie reviews dataset based on admiration (positive) and criticism (negative), using Naïve Bayes and SVM algorithms, concluding that SVM outperforms the former. Fang et al. [22] tackled a fundamental sentiment polarity categorization problem using three machine learning algorithms, i.e., NB, Random-Forest, and SVM, on an Amazon Product Reviews dataset with promising outcomes. A sentence-to-sentence attention network has been proposed for the online social reviews sentiment analysis that outperforms cross-domain information [23]. A generic framework for sentiment analysis is proposed to achieve good performance [24]. Some researchers employed a single algorithm for their sentiment classification process in all the works mentioned above. In contrast, others selected multiple algorithms and applied them individually on datasets, but most of the researchers used ensemble learning for sentiment analysis. As research in this field grows, ensemble learning is directed to acquire better accuracy and more efficient results.

III. MATERIALS AND METHODS

This section outlines the prerequisite methods adopted to design and formulate the proposed POS-EFVM model.

A. ENSEMBLE LEARNING

The ensemble learning process is developed in the machine learning approach, in which various base learners are trained together to solve a particular problem [50]. In comparison, where a single machine-learning algorithm tries to learn one hypothesis from the training set, an ensemble model integrates various theories and improves the accuracy of base learners [51]. Devaraj et al. [52] proved that an ensemble lexicon pooled approach could achieve higher accuracy than the standalone NB learner. The generalization ability of the ensemble model is more than the single model; that is why researchers predominantly use it.

1) BAGGING ENSEMBLE APPROACH

Bagging is one of the earliest ensemble approaches, intuitive and straightforward to implement, and provides excellent performance. Bootstrap replicas are performed in the training data for generating the mixture by bagging, and different random subsets of data are calculated by replacing all training datasets [53]. Every subset of the training data is used to train different base learners simultaneously.

Algorithm 1 The Process of Bagging Ensemble Learning

Input: Data Set $DA = \{(A_1B_1), (A_2B_2), \ldots, (A_nB_n)\}$
The base learning algorithm is AL
The total number of learning iterations is I
Process:
for $t = 1, 2, \ldots, I$ **do**
$D_t = \text{Bootstrap}(DA)$; ▷ Generate a sample of bootstrap from DA
$h_t = AL(D_t)$; ▷ Training of base learner h_t from the sample of bootstrap
end for
Output: $H(X) = \text{Simple Avg} = \frac{1}{N}\sum_{i=1}^{N} L_i$

Algorithm 1 describes the bagging algorithm, which shows that every element has the same probability of being randomly derived by the subset. In the subsequent training stage of bagging, each learner is trained in parallel, and each model is built independently. Finally, a simple average method is applied for predicting the results without providing any weights on N learners. We selected two powerful ensemble methods (Random-Forest and Extremely-Randomized Tree) to formulate the proposed model from the bagging concept.

2) EXTREMELY-RANDOMIZED TREE ENSEMBLE CLASSIFIER

The Extremely-Randomized Tree classifier refers to extremely randomized trees based on an ensemble learning approach. The results are predicted based on the aggregation of multiple trees from the different types of trees collected in a forest. The original training sample is used for every decision tree in the Extremely-Randomized Tree forest, and each tree provides a random selection of k features from the feature set; each decision tree must select the best feature for splitting the dataset based on mathematical rules. This random feature sampling creates multiple de-correlated decision trees [54]. For determining a better split of a tree, entropy (Equation 1) and gain (Equation 2) are calculated as follows.

$$\text{Entropy}(E) = \sum_{k=1}^{n} -p_k \log_2(p_k) \tag{1}$$

TABLE 1. Summary of existing ensemble methods for sentiment analysis task.

Author and Year	Findings	Techniques	Type of Data
Ying et al. 2012 [25]	Ensemble framework for sentiment classification.	NB, Centroid-based classification, KNN, Maximum Entropy (ME), and SVM	Reviews
Martin-Valdivia et al. 2013 [26]	Spanish reviews sentiment detection with the combination of supervised and unsupervised learning.	SVM, NB, KNN, Bayesian Logistic Regression (BBR)	Reviews
Saha et al. 2013 [7]	Vote ensemble classifier for named entity recognition.	NB, Decision Tree, Memory Based Learner (MBL), Hidden Markov Model, HMM, ME, Conditional Random Field (CRF), and SVM	News Corpus
Govindraj et al. 2014 [27]	Movie reviews sentiment classification is performed by a bagged ensemble classifier.	NB, SVM, and Genetic algorithm	Reviews
Winkler et al. 2015 [28]	Multiclass ensemble model for sentiment prediction.	Decision Tree, Adaptive Boosting, Gaussian Processes, KNN, SVM, Artificial Neural Network (ANN), and Random-Forest	Reviews
Silva et al. 2014 [29]	Twitter sentiment classification using ensemble classifier.	SVM, Random-Forest, Logistic Regression, and Naive Bayes	Tweets
Wang et al. 2014 [30]	Ensemble learning for sentiment classification on 10 reviews datasets.	NB, ME, Decision Tree, K-Nearest Neighbor, and SVM	Reviews
Chalothorn et al. 2015 [31]	Twitter sentiment analysis using an ensemble approach.	SentiStrength, NB, SVM	Tweets
Ouyang et al. 2015 [32]	Designed a suitable architecture using Convolutional Neural Network (CNN) with Word2vec for sentiment analysis.	CNN with Word2Vec	Reviews
Alnashwan et al. 2016 [33]	Improved ensemble model for meta-level features.	SVM, Bayes, Logistic Regression, and Decision Forest	Tweets
Catal et al. 2017 [34]	Multiple-classifiers sentiment classification model.	NB and SVM	Reviews
Korovkinas et al. 2017 [35]	Ensemble model proposed for sentiment classification using NB and SVM.	SVM, NB	Reviews
AL-Sharuee1 et al. 2017 [36]	Clustering classifier ensemble approach proposed with automatic contextual analysis.	Clustering Ensemble	Reviews
Akhtar et al. 2018 [37]	Two-step, aspect-based sentiment analysis for ensemble construction and feature selection.	ME, Conditional Random Field, and SVM	Tweets, News, Facebook posts, letters, blogs
Liao et al. 2017 [38]	They designed a simple CNN model for sentiment analysis.	CNN	Tweets
Ramadhani et al. 2017 [39]	Sentiment analysis of tweets using deep learning approaches.	DNN and Multilayer Perceptron	Tweets
Lu et al. 2017 [40]	Investigated SVM, CNN and RNN for twitter sentiment analysis.	SVM, CNN, and RNN	Tweets
Jianqiang et al. 2018 [41]	Introduced a word embedding methods, which contain co-occurrence statistical characteristics and latent contextual semantic relationships.	CNN	Tweets
Hanafy et al. 2018 [42]	Presented a combined approach of classical and deep learning models for Twitter sentiment analysis.	Bag-of-Words (BOW) with TF-IDF, MaxEnt with Word2Vec, SVM with Word2Vec, Long Short-Term Memory (LSTM), and CNN-static	Tweets
AL-Sharuee et al. 2018 [43]	An ensemble clustering approach for sentiment analysis and comparison.	Clustering Ensemble	Reviews
Fouad et al. 2018 [44]	The efficient sentiment analysis system of Twitter with ensemble classifier and feature selection.	SVM, NB, Logistic Regression	Tweets
López et al. 2019 [45]	Ensemble method for domain adaptation in sentiment analysis.	Average Based Model (AVG), Neutral Penalty Based Model, and Genetic Algorithm	Tweets
Yang et al. 2019 [46]	Multiple classifiers for sentiment classification.	Bagging-SVM, Boosting-NB, and Random Subspace Method (RSM)-SVM	Reviews
Yang et al. 2019 [47]	Train XLNet transformer model for sentiment classification.	XLNet	Reviews
Xie et al. 2020 [48]	Presented an Unsupervised Data Augmentation (UDA) model for semi-supervised learning.	BERT-Fine-Tuned	Reviews
Kunte et al. 2020 [49]	Extreme Gradient-Boost (XGB) ensemble model for predicting the personality of social network users.	XGB	Tweets

$$\mathrm{Gain}(E, A) = \mathrm{Entropy}(E) - \sum_{v \in \mathrm{Values}(A)} \frac{|E_v|}{|E|} \mathrm{Entropy}(E_v) \tag{2}$$

where n is the number of unique class labels of the dataset, and p_k is the proportion of rows with an output label k.

3) RANDOM-FOREST ENSEMBLE CLASSIFIER

In Random-Forest, various decision trees are constructed at the time of the training process, and the predictions from all the learners are collected for the final prediction via the mean or mode method. Feature importance is calculated based on the reduction in node impurity weights and node probability calculated by the number of samples reached by node/total number of samples. The higher the value of a feature, the more important that feature node holds [55], which is calculated (Equation 3) as:

$$ni_j = w_j \cdot c_j - w_{\mathrm{left}}(i) \cdot c_{\mathrm{left}}(i) - w_{\mathrm{right}}(i) \cdot c_{\mathrm{right}}(i) \tag{3}$$

where n_{ij} depicts node j importance; w_j is the weighted samples of node j; c_j is node j impurity value; left_j indicates the left split child node from node j, and right_j represents the right split child node from node j. The importance of each feature is calculated as follows:

$$fi = \frac{\sum_{j:\text{node } j \text{ splits on feature } i} n_{ij}}{\sum_{k \in \text{all nodes}} n_{ik}} \tag{4}$$

where fi_i represents the importance of feature i, and ni_j is the importance of node j. After that, the total value of feature importance is calculated in Equation 5.

$$RFfi_i = \frac{\sum_{j \in \text{all trees}} \text{normfi}_{ij}}{T} \tag{5}$$

where $RFfi_i$ is the importance of feature i calculated by all of the trees in the Random-Forest model; normfi_{ij} is the normalized feature importance of feature i in tree j, and T is the total number of trees. Jotheeswaran et al [56] used Random-Forest to improve the precision of feature selection in a Twitter movie reviews dataset.

4) BOOSTING ENSEMBLE APPROACH

Boosting is a powerful ensemble approach that seeks to improve the power of the predicted result by training a sequence of weak learners, where each learner compensates for the weakness of their predecessor. Unlike bagging, boosting does not learn independently but instead depends on previous learners. Algorithm 2 describes the algorithm for the boosting ensemble learning process.

Algorithm 2 The Process of Boosting Ensemble Learning

Input: Data Set $DA = \{(A_1B_1), (A_2B_2), \ldots, (A_nB_n)\}$
  The base learning algorithm is AL
  The number of learning iterations is I
Process:
for $t = 1, 2, \ldots, I$ **do**
  $h_t = AL(D, D_t)$; ▷ Training of base learner h_t by D using Distribution D_t
  $\varepsilon_t = Pr_{i \sim D_t}[h_t(x_t) \neq y_i]$; ▷ h_t error measuring
  $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$; ▷ h_t weight determination
end for
Output: $H(X) = \text{Weighted Avg} = \frac{1}{N} \sum_{i=1}^{N} w_i L_i$

5) ADAPTIVE-BOOST ENSEMBLE CLASSIFIER

Adaptive-Boost is one of the first boosting algorithms that implement the boosting ensemble concept and helps to integrate various weak learners into a single strong learner. Decision trees are weak learners, and their single split is called a decision stump. The central concept of Adaptive-boost is that more weight is put on complicated instance classifiers, and lighter weights are assigned to those that work well. This method is useful for both regression and classification problems. Adaptive-Boost is also a successive classifier that builds support for instances that had been misclassified by previous classifiers [57].

$$\varepsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \tag{6}$$

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right) \tag{7}$$

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \tag{8}$$

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right) \tag{9}$$

where D_t represents weak learners; i is the i-th training sample; ϵ denotes an element of sample; x is a training sample; h_t is the hypothesis set that aims to select low error weights; α is the weight for the classifier; Pr is probabilities; h_t is the hypothesis/classifier; exp is Euler's e : 2.71828, and Z_t is a normalization factor. In summary, (Equation 6) calculates a low weighted error; (Equation 7) is used to assign weights to the learners; (Equation 8) is the update learner process; and (Equation 9) produces the final hypothesis results.

6) GRADIENT-BOOST ENSEMBLE CLASSIFIER

Gradient-Boost is the generalized form of Adaptive-Boost, which depends on three elements: loss function for optimization, weak learner for making the predictions, and an additive model for adding the weak learners to minimize the loss function. While the loss function works based on the problem, it must be differentiable in Gradient-Boost regression trees, where decision trees are used as weak learners. Traditionally, gradient descent is used to minimize the parameters, like weights and coefficients, in neural networks [58].

$$F_0(x) = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} L(y_i, \gamma) \tag{10}$$

$$\text{rim} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \tag{11}$$

$$h_m(x) = \{(x_i, \text{rim})\}_{i=1}^{n} \tag{12}$$

$$\gamma_m = \underset{\gamma}{\text{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{13}$$

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{14}$$

where F_0 refers to the constant model; γ is a constant value; m is the base learner; L is model loss. In Equation (10), the model is initialized with a constant value; Equation (11) is used to calculate the pseudo-residuals; Equation (12) is applied to fit the base learner (tree) $h_m(x)$ according to pseudo-residuals; Equation (13) performs optimization via multiplier computation γ_m; and Equation (14) provides the final output $F_m(x)$.

B. PARTICLE SWARM OPTIMIZATION

PSO is a meta-heuristics optimization technique that adopts the nature of birds' simulation and movement to search for their food. Each swarm particle that flies in a search space represents the new candidate solution. It is an optimization scheme that refines the candidate solutions iteratively. Their local best-fitted location influences the transition of particles in iteration i concerning perfect p_{best}^{i-1} and the best fitting place in the search area g_{best}^{i-1} in iteration $i-1$. Velocity and location of particle t in s dimension optimization space are represented as ($f_{\text{avg}}, f_{\text{cur}}, f_{\min}$) and ($x_{tj}^{i}$), respectively, which is formulated in (Equation 15) and (Equation 16).

$$v_t^i = wv_t^{i-1} + c_1 b_1^i (pbest_t^{i-1} - x_t^{i-1}) + c_2 b_2^i (gbest_s^{i-1} - x_t^{i-1}) \tag{15}$$

$$x_t^i = x_t^{i-1} + v_t^i \tag{16}$$

The particle exploits the best location explored, and their neighbour shifts closer to the optimum solution. PSO can be structured for multi-dimension feature selection problems [59].

IV. PROPOSED METHODOLOGY

In this section, we present a proposed novel sentiment analysis model. Fig. 1 presents the working procedure of the proposed fusion model. The performance of an off-the-shelf PSO-EFVM relies on feature extraction, feature selection, ensemble learning, and a PSO-based optimization mechanism. Although, it is known that the ensemble learner addresses the limitation of the individual learner and performs well because, in ensemble learning, various base classifiers integrate to form a better model in terms of accuracy, performance, and efficiency. However, the challenge in building the ensemble model is how to learn and optimize the contribution of each individual learner that can better perform on the cross-domain datasets by solving critical linguistic problems. Hence, a novel fusion model is developed that calculates the contribution of four ensemble classifiers with PSO selected optimal hyperparameters of each individual learner. For better training purposes, all the datasets get cleaned first by following the fundamental preprocessing steps and, after that, the combination of different feature selection paradigms (N-Gram, Part-of-Speech (PoS) tagging, Negation-words handling, Sentiment-words, and Position-of-words calculation) have been applied.

The GLoVe word embedding is exploited to calculate the word vectors of extracted features. After that, the Information Gain (IG) is calculated to reduce the dimension of a large word vector. Finally, the selected feature vectors have fed into the fusion ensemble training, and hyperparameters get tuned by the PSO mechanism to develop a robust model for an effective sentiment analysis task. We claim that our novel PSO-EFVM model obtains higher results than traditional baselines, advanced deep learning, and transformer models on different domain datasets within a minimum time span. Algorithm 3 describes the novel algorithm of sentiment analysis, which illustrates the whole procedure of the proposed PSO-EFVM.

TABLE 2. Major information regarding all the investigated datasets.

Dataset	Columns	Rows	Polarity
Sentiment-140 Twitter Reviews	2 (Review, Sentiment)	50000	Positive / Negative
Reddit-Twitter Reviews	2 (Reviews, Category)	37249	Positive / Negative
Amazon-Shopping Reviews	2 (Review, Sentiment)	23486	Positive / Negative
Alexa-Product Reviews	2 (Review, Feedback)	3150	Positive / Negative
SMS-Collection Reviews	2 (Review, Sentiment)	5574	Positive / Negative

A. REVIEWS ACQUISITION AND FUNDAMENTAL PREPROCESSING

To evaluate the ensemble model's efficiency for different domains, datasets are collected from five benchmark sources, including Sentiment-140 Reviews, Reddit-App Reviews, Amazon-shopping Reviews, Alexa-Product Reviews, and SMS-Collection Reviews. Table 2 describes information regarding all datasets used for experiments, and Fig. 2 presents the ratio of positive and negative reviews of five investigated datasets.

Before implementing the training model, fundamental preprocessing is required to select the essential features and eliminate useless information [54]. Initial preprocessing holds the major three steps to clean the raw database.

- **Noise Removal**: The elements in all five review corpus, such as hashtags, URLs, numbers, and punctuations, are extracted and removed.
- **Syntactic Correction**: Reviews are written in informal language that holds acronyms as well as spelling mistakes that affect the accuracy of the model. Hence, a python based Aspell library[9] is exploited for syntactic correction.
- **Replacement with Describing Words**: The emotion icons are replaced by their describing words using the python emoji package.

B. FEATURE EXTRACTION

The feature vectors have been extracted from cleaned reviews tokens for accurate polarity calculation and sentiment classification.

- **Vocab Vector** ($v2v$): It is built based on the N-gram vector scheme, which is the most popular and straightforward portrayal model. Here, N-grams holding 1-gram, 2-gram, and 3-gram have been employed for mapping a vector of TF-IDF values related to the word [60]. N-Grams (Equation 17) are sequences of elements that appear in a text: characters, words, or POS-tags [61].

$$N\text{-}Grams = W - (N - 1) \tag{17}$$

Here, W refers to words present in sentence S and N-grams present in sentence S. Then, the corpus is

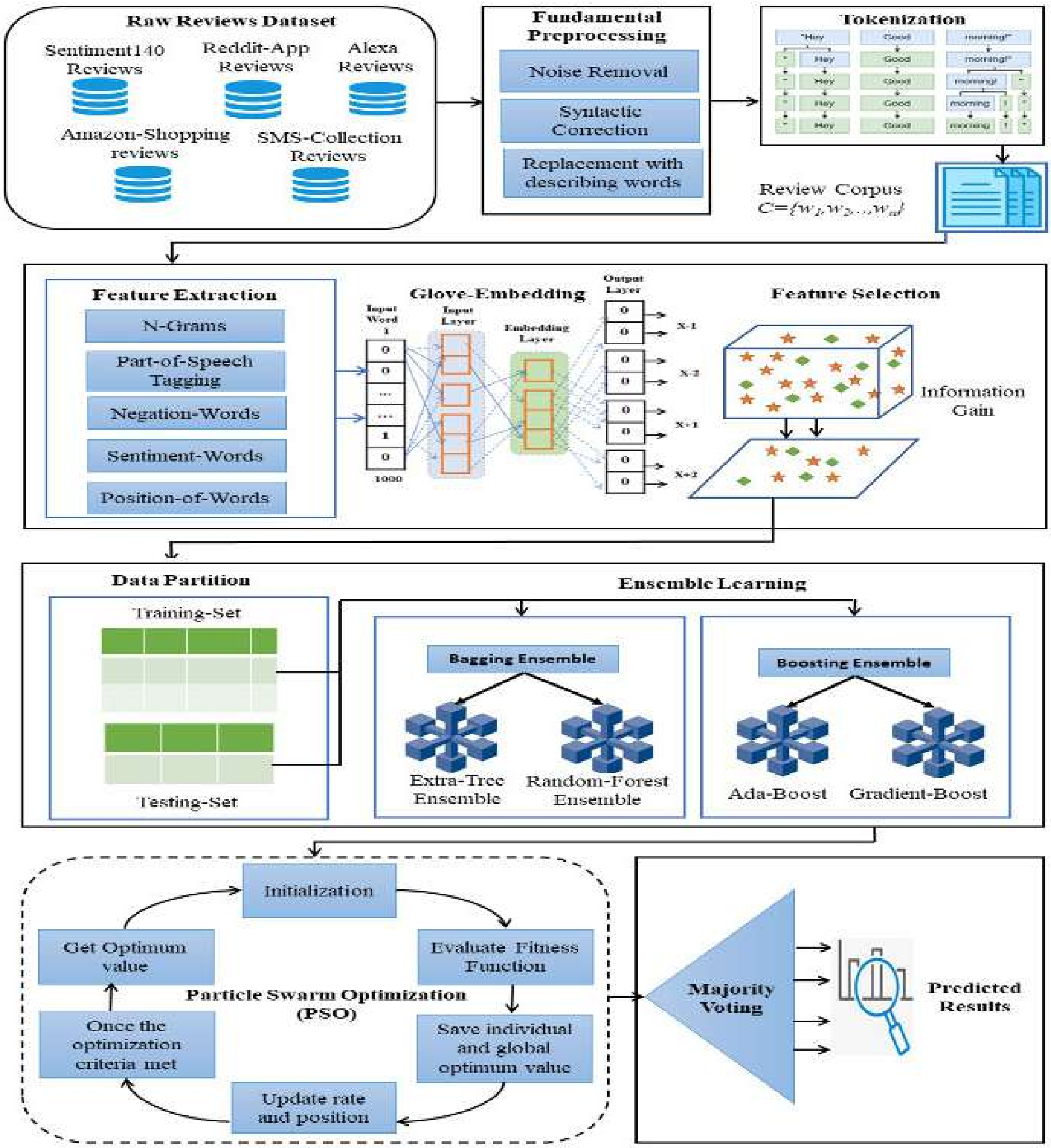


FIGURE 1. The architecture of the proposed PSO-EFVM model for reviews sentiment analysis.

collected in the form of individual tokens as $v2v = \{v_1, v_2, v_3, \ldots, v_n\}$.

- **PoS Tagging Vector** ($pt2v$): Developed using the PoS tags available in reviews r. The PoS tags provide more information regarding the words, such as nearest neighbors, syntactic classification (adjective, verb, adverb, or noun, etc.), and their relation. PoS tagging is an essential phase in sentiment analysis, required to assign each word related to the exact PoS tag [61]. The NLTK toolkit is utilized for this task, and $pt2v$ vectors are collected as $pt2v = \{ps_1, ps_2, ps_3, \ldots, ps_n\}$.
- **Sentiment Vector** ($s2v$): It is built by extracting negation words (no, wouldn't, shouldn't) and sentiment words like sentiment score value, respectively. Here, $s2v$ is collected as $s2v = \{s_1, s_2, s_3, \ldots, s_n\}$.
- **Position Vector** ($p2v$): It is built by calculating the position of the words required for machine learning training. The position of the word is calculated by identifying the distance of a word from the remaining words. Representation of position vector calculation for the sentence ''the iPhone has a good quality'' is depicted in Table 3.

The position vector is extracted in as

$$p2v = \{p_1^v, p_2^v, p_3^v, \ldots, p_n^v\}.$$

These four feature vectors $v2v$, $pt2v$, $s2v$, and $p2v$ are concatenated into a single vector.

Finally, the GLoVe word embedding is utilized for training word vectors from the global co-occurrence matrix using an

Algorithm 3 A Proposed PSO-EFVM Algorithm of Optimized Ensemble Sentiment Classification

Initial Corpus: {Sentiment-140 Reviews, Reddit-Twitter Reviews,
Amazon-Shopping Reviews, SMS-Collection Reviews, Alexa Reviews}
Fundamental Preprocessing: This step removes the initial noise from the raw corpus.
• Noise Removal
• Syntactic Correction
• Replacement of emotions with describing words
Feature Extraction: The required features are collected from a
cleaned corpus to perform effective textual processing.
• w2v = $\{v_1, v_2, v_3, \ldots, v_n\}$, a collection of word phrases from reviews r.
• pt2v = $\{ps_1, ps_2, ps_3, \ldots, ps_n\}$, a collection of PoS tag word vectors.
• s2v = $\{s_1, s_2, s_3, \ldots, s_n\}$, a collection of sentiment word vectors.
• p2v = $\{p_1, p_2, p_3, \ldots, p_n\}$, a collection of word position vectors.
Feature Vector: The GLoVe word embedding is exploited to
generate vectors from selected features.
• $M(\theta) = \frac{1}{W}\sum_{n,m=1}^{W} f(P_{n,m})(x_n \cdot y_m - \log P_{n,m})^2$
Feature Selection: The IG feature reduction is employed to reduce
the dimensionality of the feature vectors in order to enhance the
interpretation of the training model.
• $IG(v) = -\sum_{k=1}^{n} p(k)\log p(k) + \ldots$
Ensemble Learning: The selected ensemble learners are trained on
reduced feature vectors.
• **Extremely-Randomized Tree (E1):** Implements a number
of randomized trees to control predictive accuracy.
$Entropy(E) = -\sum_{k=1}^{n} pk \log_2(pk)$
$Gain(E, A) = Entropy(E) - \sum_{v \in Values(A)} \frac{|E_v|}{|E|} Entropy(E_v)$
• **Random-Forest (E2):** Reduces the variance and improves
the performance.
$n_i = w_j \cdot c_j - w_{\text{left}}(j) \cdot c_{\text{left}}(j) - w_{\text{right}}(j) \cdot c_{\text{right}}(j)$
• **Adaptive-Boost (E3):** Exploited to reduce loss function.
$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
• **Gradient-Boosting (E4):** Implemented to manipulate the
differential loss.
$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$
PSO (Optimal Hyperparameter Selection): A weight parameter
optimization is calculated for each implemented ensemble in
combination to improve their performance.
Initialize:
• Set the location parameter x^i_{tj}.
• Set the velocity parameter v^i_{ts}.
While ($k <$ last-iteration) **Do**
• Fitness calculation $f'_{avg}, f'_{cur}, f_{min}$
• Update local $pbest^{i-1}_{ts}$ and global $gbest^{i-1}_{s}$ optimum position
• Update w inertia weight
• Update velocity v^{i+1}_{ts} and location x^{i+1}_{tj}
• $k++$;
End While
Got the optimum parameters value.
Majority Voting: Predicted the class label based on the majority
voting of optimized ensemble learners.
$\bar{R} = \underset{a}{\text{argmax}}\{E1(a), E2(a), E3(a), E4(a)\}$
Final Prediction: Predicted result = $\bar{R}$

objective function (Equation 18).

$$M(\theta) = \frac{1}{2}\sum_{n,m=1}^{W} f(P_{n,m})(\mathbf{x}_n^T \mathbf{y}_m - \log P_{nm})^2 \tag{18}$$

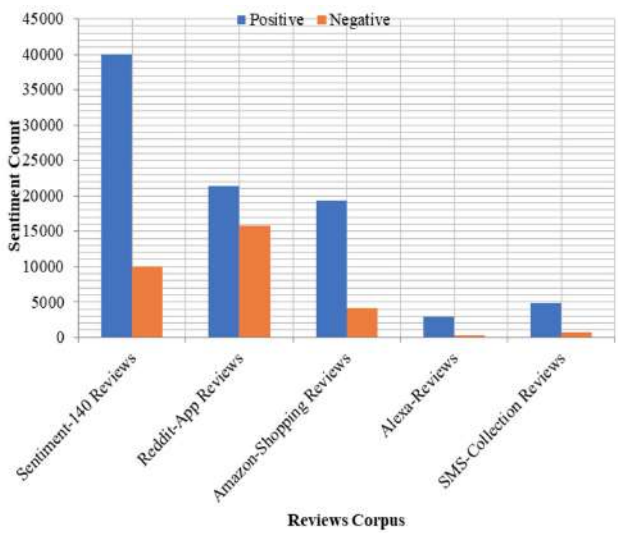


FIGURE 2. Polarity distribution of five cross-domain reviews corpus.

The given n, and m are two co-occurred different words that can optimize by reducing the dot product of the n, m word vectors. The key calculation of single pair is calculated as

$$M(\theta) = \left(\mathbf{x}_n^T \mathbf{y}_m - \log P_{nm}\right)^2.$$

Here, $M(\theta)$ is an objective function, $\mathbf{x}_n^T \mathbf{y}_m$ is a dot product of the vector $\mathbf{x}_n$ and $\mathbf{y}_m$, and P_{nm} depicts the co-occurrence of

TABLE 3. Example of the position vector calculation.

Words	the	iPhone	has	a	good	quality
the	0	1	2	3	4	5
iPhone	-1	0	1	2	3	4
has	-2	-1	0	1	2	3
a	-3	-2	-1	0	1	2
good	-4	-3	-2	-1	0	1
quality	-5	-4	-3	-2	-1	0

n and m. Minimizing the difference between n and m is not enough when these never co-occur. For this, the weighting function $f(P_{n,m})$ is added to the objective function as

$$M(\theta) = f(P_{n,m}) \left(\mathbf{x}_n^T \mathbf{y}_m - \log P_{nm} \right)^2 .$$

It helps in balancing the weight problem of very common and uncommon words. Finally, the entire objective function is multiplied by $\frac{1}{2}$ for squaring the variation between the co-occurrence and dot-product count of n and m. Final combined feature vectors are represented as $C = \{w_1, w_2, w_3, \ldots, w_n\}$.

C. FEATURE SELECTION

Every word of review is represented as a number vector based on the feature extracted in the above phase. The dimension of these vectors is dramatically increased due to different feature extraction. The curse of huge dimensionality exists in almost all the applications of NLP, including sentiment analysis. Hence, an IG feature selection technique is exploited to minimize the size of input feature vectors. The information gain weight has been calculated (Equation 19) for each vector of the final corpus, $C = \{w_1, w_2, w_3, \ldots, w_n\}$ and those features with a greater weight than 0.01 are selected for ensemble learning.

$$IG(v) = -\sum_{k=1}^{n} p(l_k) \log p(l_k) + p(i) \sum_{k=1}^{n} p(l_k|i) \log p(l_k|i) + p(\bar{i}) \sum_{k=1}^{n} p(l_k|\bar{i}) \log p(l_k|\bar{i}) \quad (19)$$

where $p(c_t)$ is the fraction of labeled review with class c_t, $p(i)$ is the fraction of review that feature i occurs, $p(c_t|i)$ is the fraction of review having class c_t that has feature i.

D. ENSEMBLE LEARNING FOR SENTIMENT CLASSIFICATION

The major step of the proposed system is to select the best ensemble learners for the sentiment classification task. The ensemble should select that have the potential to effectively segregate positive and negative opinions. Everyone knows that the ensemble model performs better than the individual one, as they join the main system's search space and improve the performance of multiple weak learners to build a single powerful model. However, the challenge of ensemble learning is choosing each base learner's contribution as a powerful joint ensemble solution. Therefore, to build a single robust model, we joined two different ensemble approaches, bagging and boosting. To capture the capability of the models, first, we implemented the Extremely-Randomized Tree classifier that extracted random samples from the training set, distributed them among various trees, and used aggregation of these trees for predicting results. The second is the Random Forest, which performed deterministic splits for result calculation. Third, Adaptive-Boost boosted the performance of weak learners by adding extra weights on difficult instances rather than well-performing instances, and fourth, Gradient-Boost minimized the loss by adding binary trees and regression trees. All the implemented ensemble models are briefly discussed in Section III-A. We argued that the PSO-EFVM has the ability to find the right contribution to the set of ensembles. To implement these four ensembles, the first datasets are categorized into two parts, 80% for the training set and 20% for the testing set. To avoid the problem of overfitting, split the data into k-fold cross validation where k = 5 that run on train and test. During each run, one fold is dedicated for the test set, and the rest of the four sets are for training purposes. The k-value is selected five because the too large a value of k-fold leads to less variance and limits the variations across the multiple iterations.

E. PSO-BASED HYPERPARAMETER OPTIMIZATION

To improve the performance of investigated ensembles and avoid the high labor cost of manually adjusting the weights of the parameters, the PSO technique selects optimum hyper-parameters for each implemented-based ensemble model. Selecting appropriate hyperparameters directly affects the performance of the training model. PSO randomly initializes a set of a particle in the solution space where every particle flies through the solution space with a particular speed according to the current optimum particle and reaches the optimum solution by searching in a successive generation [62]. The process of generating the best hyperparameter for each investigated ensemble is depicted in Algorithm 4.

Table 4 presents the optimum parameters selected by PSO bio-mimetic algorithm for investigated ensembles. Here, d is the dimension selected for the hyperparameter of a particular technique. PSO-based hyperparameter tuning generates the best parameter range and enhances the performance of each investigated ensemble learner. The optimized ensemble learners better contribute to the majority voting scheme for final predictions.

F. FINAL PREDICTION VIA MAJORITY VOTING SCHEME

It is notable that there is not any single classification model that always produces the most accurate solution because every algorithm has elementary discriminant bias, and they differentially assume about the data. The global error rate of a single learner in different instances can decrease by formulating a combination of suitable classifiers [63]. Hence, this model implies the diversity optimization principle for selecting the base classifiers. The diversity principle assumes

Algorithm 4 The Process of Hyperparameter Selection of Classification Model Using PSO Technique

Initialization:
Initialize the position $a_n(0)\ \forall n \in \{1, \ldots, K\}$
Initialize the optimum location of the particle to its beginning position $p_n(0) = a_n(0)$
Calculate fitness for each particle and if $f(a_m(0)) \geq f(a_n(0))\ \forall n \neq m$, initialize best global as $g = a_m(0)$
Process:
while the stopping criteria are met **do**
  Update the velocity of the particle:
  $v_n(i+1) = v_n(i) + c_r \cdot (p_n - a_n(i)) \cdot R_n + c_2 \cdot (g - a_n(i)) \cdot R_2$
  Update the location of the particle:
  $a_n(i+1) = a_n(i) + v_n(i+1)$
  Particle fitness evaluation $f(a_n(i+1))$.
  If $f(a_n(i+1)) \geq f(p_n)$, update local best $p_n = a_n(i+1)$
  If $f(a_n(i+1)) > f(g)$, update global best $g = a_n(i+1)$
end while
Finally, g is presented as the best solution.

that there is no perfect classifier available to deal with a large quantity of data, space dimensionality, and distribution hypothesis learning. The diversity principle works on three fundamental approaches for choosing a suitable combination. First is to train different algorithms on a single dataset. The second is to apply a single algorithm with different parameters to a single dataset. The third is to train a single algorithm on different sub-samples of the dataset. This model adopted the first approach of selecting the multiple algorithms on a single dataset. Here, each ensemble gives its self-prediction for each test sample, and the *majority of votes* taken by all the four ensembles decide the final prediction. Consider L the target label, with $E_i, \forall n \in \{1, 2, 3, \ldots, L\}$ presents the n-th target label predicted by the ensemble learner. Given an input a, each ensemble generates the prediction regarding the target label, yielding a total of L prediction, i.e., $P_1, P_2, \ldots, P_L$. The majority voting aims to generate a collaborative prediction for input $P(a) = n, \forall n \in \Lambda$ from all the L predictions, i.e., $P_l(a) = n_l, l = 1, \ldots, L$. Algorithm 5 depicts the procedure of majority vote selection of multiple base classifiers.

Here, majority votes act as a multi-expert optimized recommendation that reduces the possible chance of false prediction. We exploited the self-ensemble models with PSO-based hyperparameter optimization to build robust fusion for sentiment classification. However, multiple neural models related combinations previously existed, but they have few limitations. First, neural models are too complicated by themselves, and once combined in a group with others, they increase exhaustive complexity. Second, neural models need large training corpora to train; these are unable to generate accurate predictions for the small-scale datasets. One major point should be noted that the weak learners

TABLE 4. Optimal hyperparameter value selected by PSO.

Method	Hyperparameter	Type	Range	PSO-Value
Random Forest (d-5)	n_estimators	int	10-100	70
	criterion	string	"gini", "entropy"	"gini"
	max_depth	int	5-50	30
	min_samples _split	int/float	5-20	8
	max_features	string	"auto", "sqrt", "log2"	"sqrt"
Extra Tree (d-5)	n_estimators	int	10-100	80
	criterion	string	"gini", "entropy"	"entropy"
	max_depth	int	5-50	35
	min_samples_leaf	int/float	5-20	6
	min_impurity _decrease	int/float	0.0-5.0	0.1
Adaptive-Boost (d-3)	n_estimators	int	10-100	90
	learning_rate	float	1.0-6.0	5.0
	algorithm	string	'SAMME', 'SAMME.R'	'SAMME'
Gradient-Boost (d-5)	n_estimators	int	10-150	120
	criterion	string	'friedman_mse', 'squared _error', 'mse'	'friedman_mse'
	learning_rate	float	1.0-3.0	3.0
	subsample	float	1.0-5.0	2.0
	min_samples_leaf	int/float	5-20	7

in the combination should be coherent in a way that is aggregated. It must be that if a model has been chosen with low variance and high bias, it should be aggregated with the model that tends to reduce the bias, conversely. Therefore, PSO-EFVM combines the potential of bagging (focus on getting an ensemble with less variance) and boosting (concentrate on producing the model with less bias) to cope with the trade off of variance-bias. We have chosen the combination of two bagging ensembles (Random-Forest and Extremely-Randomized Tree) and two boosting ensembles (Adaptive-Boost and Gradient-Boost) that balance the problem of high variance and bias proportionally.

Experiments were conducted on Jupyter Notebook Python version 3.7 with Windows-10, 4 GB RAM, and Intel i5 8th generation processor and used five benchmark datasets sequentially to evaluate the authenticity of the proposed ensemble framework. Small and large datasets were included in the experiments to test the reliability of the proposed model.

V. EXPERIMENTAL RESULTS

A. EVALUATION CRITERIA

The four base ensemble classifiers and one proposed ensemble model were applied to five online review

Algorithm 5 The Process of Selecting an Ensemble Classifier for Majority Voting

Initialization:
  Validation Corpus $C = \{a_1, a_2, a_3, \ldots, a_n\}$
  Ensemble Classification Model $E = \{e_1, e_2, \ldots, e_m\}$
  Positive= 0, Negative= 0
Process:
for $a_i = 1$ to n **do**
    Positive= 0, Negative= 0
    for $e_j = 1$ to m **do**
        $a^e_{i.class} \leftarrow$ predict(a_i, e_j) ▷ extract the class label
        if $a^e_{i.class}$ = Positive **then** ▷ jth model predicts the 'Positive'
            Positive←Positive+1;
        else if $a^e_{i.class}$ = Negative **then** ▷ jth model predicts the 'Negative'
            Negative←Negative+1;
        end if
    end for
    if Positive > Negative **then**
        $a_{i.class} \leftarrow$ 'Positive';
    else
        $a_{i.class} \leftarrow$ 'Negative';
    end if
end for
Output:
  Prediction of each sentiment as positive or negative.

TABLE 5. Classification confusion matrix.

	Actual Positive	Actual Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

datasets. Standard statistical measures, namely True Positive Rate (TPR), False Positive Rate (FPR), accuracy (Equation 20), weighted-precision (Equation 21), weighted-recall (Equation 22), weighted-F1-score (Equation 23), ROC-AUC curve, and runtime of the model, were used to obtain authentic results. Since the confusion matrix can provide a correct and incorrect prediction summary [64], we use the outcomes of the confusion matrix to generate the ROC-AUC curve. Table 5 depicts the confusion matrix. The associated formulas are presented below [65].

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{20}$$

$$\text{Precision}_{\text{Weighted}} = \frac{\sum_{i=1}^{m} |y_i| \frac{TP_i}{TP_i+FP_i}}{\sum_{i=1}^{m} |y_i|} \tag{21}$$

$$\text{Recall}_{\text{Weighted}} = \frac{\sum_{i=1}^{m} |y_i| \frac{TP_i}{TP_i+FN_i}}{\sum_{i=1}^{m} |y_i|} \tag{22}$$

$$F1\text{-Score}_{\text{Weighted}} = \frac{\sum_{i=1}^{m} |y_i| \frac{2TP_i}{2TP_i+FP_i+FN_i}}{\sum_{i=1}^{m} |y_i|} \tag{23}$$

The primary objective of the ensemble model is to rectify the misclassification of individual classifiers. The ROC-AUC curve demonstrates the performance of the proposed ensemble model according to various thresholds, including the probability and degree of separability. Specifically, a higher AUC score depicts a better prediction of class labels. ROC-AUC curves were plotted based on the True Positive Rate (TPR) (Equation 24) and False Positive Rate (FPR) (Equation 25), which represent the y-axis and x-axis, respectively. The curves were calculated according to previous work [66].

$$TPR = \frac{TP}{TP + FN} \tag{24}$$

$$FPR = \frac{FP}{TN + FP} \tag{25}$$

Further subsections present the evaluated results of the proposed model for discriminant domain datasets. All the experimented datasets hold different lengths and features, which helps to check the operational reliability of the proposed model on different domains.

B. RESULTS

In the sentiment-140 dataset, the proposed model has been trained on 40000 reviews, where 10000 reviews are used for training purposes. Table 6 summarises the results obtained by the proposed PSO-EFVM model for all the investigated datasets. The Sentiment-140 dataset is very large; still, the proposed ensemble model acquires greater training accuracy and TPR. However, testing accuracy is a little low in comparison to training accuracy but provides a prominent score. Fig. 3 presents the ROC-AUC curve of the proposed model for sentiment-140 twitter reviews. Here, the proposed model generates a 80 AUC score, which better distinguishes between positive and negative reviews. In the Reddit-Reviews dataset, the proposed model scored higher training accuracy and TPR value, which shows that the proposed model can train on a large dataset with effective results. In addition, this model generates minimum false alarms as it scored very low FPR.

The testing accuracy of this model is also good for the Reddit-reviews dataset. Fig. 3b depicts the ROC-AUC curve of the proposed model for Reddit-Reviews. Considering the AUC score, we can argue that this model better classifies the extensive review set of the Reddit social website. The proposed model scores higher training accuracy, testing accuracy, w-precision, w-recall, w-f1-score, and TPR within 661 seconds on Amazon-Shopping dataset. Fig. 3c shows the ROC-AUC curve of the proposed model for Amazon-Shopping reviews, where the proposed model obtained an 88 AUC score for classifying the reviews of Amazon products. The deep learning models are unable to handle small datasets. Hence, the Alexa reviews dataset has been selected to check the capability of the proposed model for a small dataset. The obtained results depict the proposed model's effectiveness for a small dataset. Fig. 3d presents the

TABLE 6. Performance metrics of various datasets.

Dataset	Training Accuracy (%)	Testing Accuracy (%)	W-Precision	W-Recall	W-F1-Score	TPR (%)	FPR (%)	Run-Time (Sec)
Sentiment-140	98.96	83.08	81	83	81	96.09	69.33	4286
Reddit-Reviews	99.49	85.73	86	86	86	84.60	13.44	2489
Amazon-Shopping Reviews	99.70	87.16	87	87	85	98.81	67.27	661
Alexa-Reviews	99.00	93.80	93	94	93	99.47	66.66	1143
SMS-Collection Reviews	99.93	97.93	98	98	98	99.78	12.72	112

TABLE 7. Comparative results of the proposed model with traditional classifiers.

Dataset	Algorithm	Training Accuracy (%)	Testing Accuracy (%)	Precision	Recall	F1-Score
Dataset 1: Sentiment-140 Twitter Reviews	Complement-NB	76.31	74.27	82	74	77
	Gaussian-NB	47.38	45.38	78	45	48
	Decision-Tree	79.97	80.11	64	80	71
	K-Neighbors	85.44	80.63	77	81	77
	Logistic-Regression	79.97	80.11	64	80	71
	Proposed-Model	**98.96**	**83.08**	**81**	**83**	**81**
Dataset 2: Reddit-Twitter Reviews	Complement-NB	82.61	79.28	80	79	79
	Gaussian-NB	80.57	75.87	76	76	76
	Decision-Tree	57.38	57.98	34	58	43
	K-Neighbors	66.26	62.59	69	63	54
	Logistic-Regression	83.01	82.25	83	82	82
	Proposed-Model	**99.49**	**85.73**	**86**	**86**	**86**
Dataset 3: Amazon-Shopping Reviews	Complement-NB	82.97	83.22	85	83	77
	Gaussian-NB	60.01	55.64	82	56	60
	Decision-Tree	86.94	85.20	73	85	78
	K-Neighbors	89.25	85.14	83	85	83
	Logistic-Regression	82.20	82.37	68	82	74
	Proposed-Model	**99.70**	**87.16**	**87**	**87**	**85**
Dataset 4: Alexa-Reviews	Complement-NB	88.25	82.85	92	83	86
	Gaussian-NB	66.34	61.26	87	61	70
	Decision-Tree	91.94	91.42	84	91	87
	K-Neighbors	93.65	91.26	87	91	88
	Logistic-Regression	91.94	91.42	84	91	87
	Proposed-Model	**99.00**	**93.80**	**93**	**94**	**93**
Dataset 5: SMS-Collection	Complement-NB	94.30	91.92	94	92	93
	Gaussian-NB	84.23	82.33	90	82	84
	Decision-Tree	86.94	85.20	73	85	78
	K-Neighbors	96.29	93.45	94	93	93
	Logistic-Regression	86.94	85.20	73	85	78
	Proposed-Model	**99.93**	**97.93**	**98**	**98**	**98**
Average Score	Complement-NB-AVG	84.88	82.31	86.6	82.2	82.4
	Gaussian-NB-AVG	67.68	64.10	82.6	64	67.6
	Decision-Tree-AVG	80.63	79.98	65.6	79.8	71.4
	K-Neighbors-AVG	85.65	82.61	79.4	82.6	79
	Logistic-Regression-AVG	84.81	84.27	74.4	84	78.4
	Proposed-Model-AVG	**99.42**	**89.54**	**89**	**89.6**	**88.6**

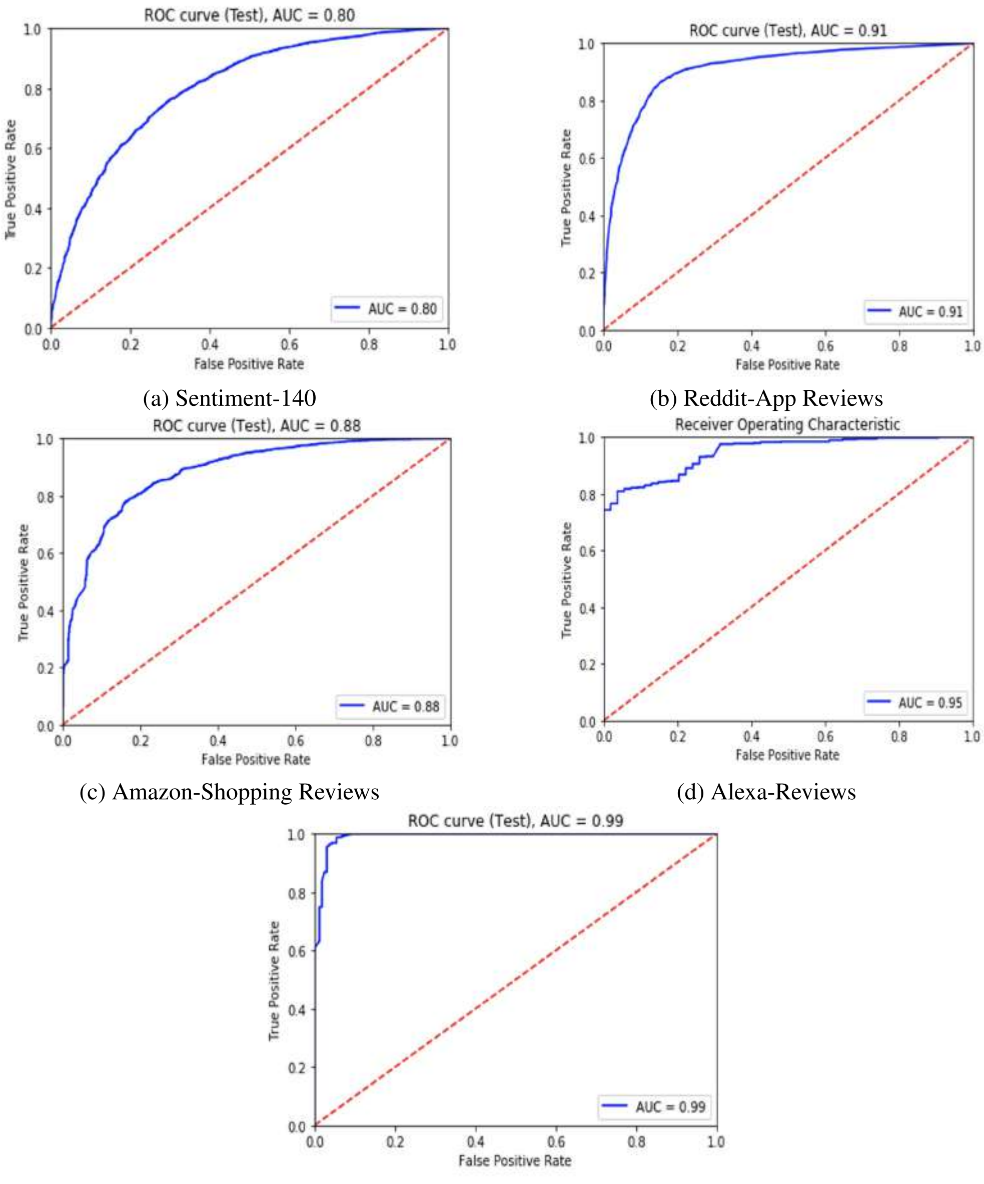
ROC-AUC curve of the proposed model for Alexa reviews, where the model obtained a higher AUC score for a small dataset. Our model achieves the highest score for the SMS collection dataset as all the measurement units reflected the best values. It has taken around 112 seconds for the training and testing process on the SMS collection dataset. Fig. 3e shows the ROC-AUC curve of the proposed model for the SMS collection dataset where the proposed model obtains a 99 AUC score, which represents the accurate separation of the positive and negative SMSs.

VI. COMPARATIVE STUDY

This section presents a brief comparative study and analysis of the experimental results. To check the efficiency of the proposed model, we compared it with traditional classifiers, base classifiers, and powerful deep learning models.

A. COMPARATIVE ANALYSIS WITH TRADITIONAL CLASSIFIERS

Traditional classifiers such as NB, KNN, decision Tree, and Logistic Regression gained popularity for classifying people's sentiment for a particular domain. Many researchers implemented these classifiers for the process of sentiment analysis. Hence, we selected a few traditional classifiers and implemented them on similar datasets, which were used to experiment with the proposed model. The proposed model's comparison with traditional classifiers has been made in terms of accuracy, w-precision, w-recall, and w-f1-score. Table 7 reported the comparative results of

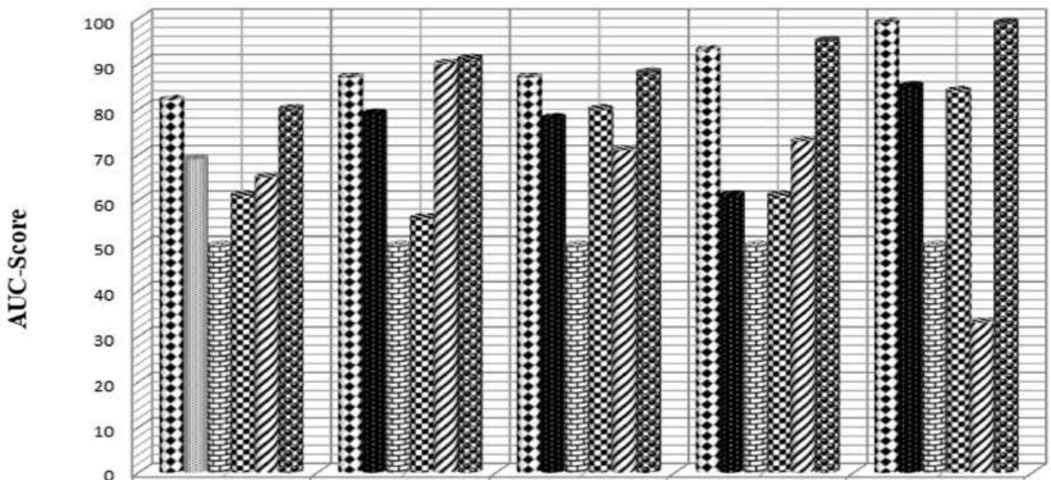


(a) Sentiment-140
(b) Reddit-App Reviews
(c) Amazon-Shopping Reviews
(d) Alexa-Reviews
(e) SMS-Collection Revi

FIGURE 3. **The ROC-AUC scores obtained by the PSO-EFVM model on different datasets.**

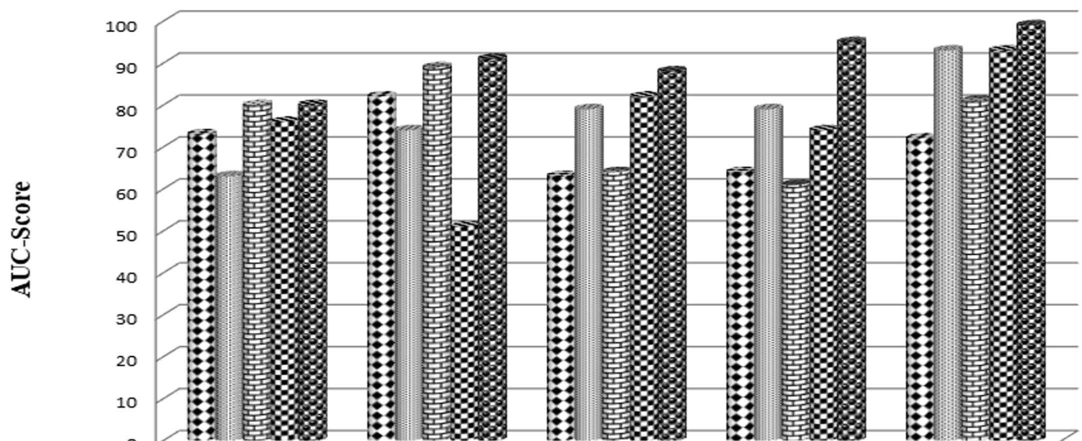
the proposed model with conventional classifiers. It has been observed that the proposed model achieves higher accuracy, w-precision, w-recall, and w-f1-score than all implemented traditional classifiers on all five datasets. Based on five experimented datasets, the average score of all the comparative measures has been calculated to check the range of improvement in the proposed model than traditional classifiers. It has been seen that the proposed model obtains 14.54%, 31.74%, 18.79%, 13.77%, and 14.61% more training accuracy than Complement-NB, Gaussian-NB, Decision-Tree, K-Neighbors, and Logistic-Regression, respectively. In the case of testing accuracy proposed model achieves 7.23%, 25.44%, 9.56%, 6.93%, and 4.67% more results than Complement-NB, Gaussian-NB, Decision-Tree, K-Neighbors, and Logistic-Regression. For the w-precision score, the proposed model generates 2.4, 6.4, 23.4, 9.6, and 14.6; for w-recall values presented model generates 7.4, 25.6, 9.8, 7, 5.6, and for w-f1-score, the proposed model generates 6.2, 21, 17.2, 9.6, and 10.2 more results than Complement-NB, Gaussian-NB, Decision-Tree, K-Neighbors, and Logistic-Regression respectively. The higher rate of the proposed model on all the datasets proves its efficiency with traditional classifiers for sentiment analysis.

Fig. 4 presents the comparative AUC score of the proposed model with traditional classifiers where the proposed model obtains a higher AUC score than Complement-NB,



	Sentiment-140	Reddit-Reviews	Amazon-Shopping	Alexa-Reviews	SMS-Collection
Complement-NB	82	87	87	93	99
Gaussian-NB	69	79	78	61	85
Decision-Tree	50	50	50	50	50
K-Neighbors	61	56	80	61	84
Logistic-Regression	65	90	71	73	33
Proposed-Model	80	91	88	95	99

FIGURE 4. Comparative AUC score of the proposed model with traditional classifiers.



	Sentiment-140	Reddit-Reviews	Amazon-Shopping	Alexa-Reviews	SMS-Collection
Ada-Boost	73	82	63	64	72
Gradient-Boost	63	74	79	79	93
Extra-Tree	80	89	64	61	81
Random-Forest	76	51	82	74	93
Proposed Ensemble	80	91	88	95	99

FIGURE 5. Comparative AUC score of the proposed model with base learners.

Gaussian-NB, Decision-Tree, K-Neighbors, and Logistic-Regression classifiers for Reddit reviews, Amazon-Shopping reviews, Alexa reviews, and SMS-Collection, except Sentiment-140 reviews dataset.

B. COMPARATIVE ANALYSIS WITH BASELINES

Selecting the base learners to form a new ensemble model is a very important task. The best parameters and discriminant capability of the base classifiers contribute to building a robust model. Suppose a novel ensemble model achieved better results in different domains. Then, we can argue that the model can handle the problem in every situation. Table 8 demonstrates the comparative outcomes of the proposed model with base classifiers.

In all five experiments, the proposed model obtains higher results than all other base classifiers (Adaptive-Boost,

TABLE 8. Comparative results of the proposed model with base learners.

Dataset	Algorithm	Training Accuracy (%)	Testing Accuracy (%)	Precision	Recall	F1-Score
Dataset 1: Sentiment-140 Twitter Reviews	Adaptive-Boost	82.15	82.25	80	82	78
	Gradient-Boost	80.20	80.40	82	80	72
	Extremely-Randomized Tree	81.66	81.58	82	82	75
	Random-Forest	81.83	82.03	80	82	77
	Proposed Ensemble	**98.96**	**83.08**	**81**	**83**	**81**
Dataset 2: Reddit-Twitter Reviews	Adaptive-Boost	82.04	81.62	82	82	81
	Gradient-Boost	69.66	69.65	76	70	65
	Extremely-Randomized Tree	82.91	82.59	83	83	82
	Random-Forest	80.16	79.02	80	80	80
	Proposed Ensemble	**99.49**	**85.73**	**86**	**86**	**86**
Dataset 3: Amazon-Shopping Reviews	Adaptive-Boost	82.99	83.05	80	83	77
	Gradient-Boost	82.38	82.54	84	83	75
	Extremely-Randomized Tree	82.19	82.39	68	82	74
	Random-Forest	84.15	83.56	82	84	79
	Proposed Ensemble	**99.70**	**87.16**	**87**	**87**	**85**
Dataset 4: Alexa-Reviews	Adaptive-Boost	92.34	91.58	90	92	88
	Gradient-Boost	92.34	91.58	92	92	88
	Extremely-Randomized Tree	91.94	91.42	84	91	87
	Random-Forest	91.94	91.42	84	91	87
	Proposed Ensemble	**99.00**	**93.80**	**93**	**94**	**93**
Dataset 5: SMS-Collection	Adaptive-Boost	88.58	88.34	87	88	88
	Gradient-Boost	91.85	90.22	91	90	88
	Extremely-Randomized Tree	88.18	86.27	88	87	82
	Random-Forest	92.53	90.58	93	92	91
	Proposed Ensemble	**99.93**	**97.93**	**98**	**98**	**98**
Average Score	Adaptive-Boost	85.62	85.37	83.8	85.4	82.4
	Gradient-Boost	83.29	82.88	85.0	83.0	77.6
	Extremely-Randomized Tree	85.38	84.85	81.0	85.0	80.0
	Random-Forest	86.12	85.32	83.8	85.8	82.8
	Proposed Ensemble	**99.42**	**89.54**	**89.0**	**89.6**	**88.6**

TABLE 9. Comparative results of the proposed model with deep-learning models.

Study	Model	Accuracy
Ouyang et al. 2015 [32]	Word2vec-CNN	45.4%
Ramadhani et al. 2017 [39]	DNN	75.03%
	Multilayer Perceptron	52.60%
Liao et al. 2017 [38]	CNN	75.39%
Lu et al. 2017 [40]	CNN	67.60%
	RNN	58.50%
Hanafy et al. 2018 [42]	LSTM	82.73%
	CNN-Static	83.87%
Jianqiang et al. 2018 [41]	GloVe-DCNN	87.62%
Yang et al. 2019 [47]	XLNet	85.04%
Xie et al. 2020 [48]	UDA (BERT-Fine-Tuned)	79.05%
Tiwari et al.	Proposed PSO-EFVM	AVG-Accuracy **89.54%**

Gradient-Boost, Extremely-Randomized Tree, and Random-Forest) in terms of training accuracy, testing accuracy, w-recall, and w-f1-score, except w-precision in the Sentiment-140 dataset. The difference between the proposed model results and base learners is huge, which shows the reliability of our proposed ensemble model for sentiment classification. The proposed model generates 13.80%, 16.13%, 14.04%, and 13.29% more training accuracy than Adaptive-Boost, Gradient-Boost, Extremely-Randomized Tree, and Random-Forest, respectively. The proposed model achieves 4.17%, 6.66%, 4.69, and 4.62 more excellent scores for testing accuracy than the base learners. Similarly, for the w-precision score, the proposed model generates 5.2, 4, 8, and 5.2; for the w-recall, the proposed model obtains 4.2, 6.6, 4.6, and 3.8, and for w-f1-score, the proposed model generates 6.2, 11, 8.6, and 5.8 higher results than Adaptive-Boost, Gradient-Boost, Extremely-Randomized Tree, and Random-Forest respectively. Although, Random-Forest generates higher results than other base classifiers in terms of training accuracy, testing accuracy, w-recall, and w-f1-score, but not capable of producing better results than the proposed model. According to the results of base classifiers, it has also been observed that the bagging-based ensemble technique Random-Forest is more capable of classifying the sentiments than Extremely-Randomized Tree and boosting (Adaptive-Boost and Gradient-Boosting) based techniques. Fig. 5 presents the comparative AUC score of the proposed model with base ensemble learners. The proposed model achieves a higher AUC score than all the base ensemble learners in five experimented datasets.

C. COMPARATIVE ANALYSIS WITH STATE-OF-THE-ART MODELS

Deep learning models have gained a lot of popularity to solve the problem of NLP. It can extract high-level features from the data, which helps to improve the capacity of faster learning on a vast amount of data. Nevertheless, deep learning models have a few challenges, such as lack of flexibility, overfitting, being expensive, and requiring massive data for better learning. Several researchers have implemented deep learning models for sentiment analysis. Here in this section, we compared our proposed ensemble model with previously published works based on deep learning techniques for sentiment analysis. Table 9 presents the comparative results of deep learning models with the proposed ensemble model.

According to the results reported in Table 9, our proposed model obtains higher accuracy than previously proposed deep learning models for sentiment analysis. Additionally, the proposed PSO-EFVM has been also compared with advanced transformer-based models such as XLNet and BERT. It has been observed that the proposed model achieves 10.49% more accuracy than the BERT transformer and 4.5% more than XLNet model. It also argued that the proposed model performs well than existing baselines and advanced transformers in minimum duration (within seconds) and on less training resources.

VII. CONCLUSION AND FUTURE WORK

This work proposes a PSO-EFVM framework for sentiment analysis using machine learning ensemble techniques. We conducted a series of experiments on public review datasets to examine the efficiency and performance of the proposed model. Each base learner of the proposed model follows an ensemble approach, which gives a new direction to perform a review classification. According to the literature, numerous studies employed traditional base learners for building the ensemble model of sentiment analysis; therefore, we selected ensemble base learners (bagging and boosting) to create a novel ensemble model based on the majority voting scheme. The proposed model integrates four ensemble learners, two baggings (Extremely-Randomized Tree and Random-Forest classifiers), two boosting (Adaptive-Boost and Gradient-Boost classifiers), and is tested on five public reviews datasets to verify its effectiveness on discriminant domains. Notably, if a model has been chosen with low variance and high bias, it should be aggregated with the model that tends to reduce the bias, conversely. Therefore, PSO-EFVM combines the potential of bagging (focus on getting an ensemble with less variance) and boosting (concentrate on producing the model with less bias) to cope with the tradeoff of variance-bias. Additionally, the hyperparameters of all the four investigated ensemble models have first been optimized by the PSO technique and then fed into the sentiment classification training that enhances the proposed solution's performance and reliability. Furthermore, the advanced feature extraction and selection methods, namely GloVe word-embedding and IG, are exploited to reduce the dimensionality of the reviews datasets to help in better performing the base learners. Our experimental results show that the proposed PSO-EFVM minimized the error rate by avoiding poor selection from the single classifier and ensuring stability. The proposed ensemble model achieved higher average results as 99.42% training accuracy, 89.54% testing accuracy, 89 w-precision, 89.6 w-recall, 88.6 w-f1-score, and 88.6 AUC score. The effectiveness of the proposed model has been compared with traditional classifiers, base ensemble learners, and deep learning models. Furthermore, the proposed ensemble model effectively enhances the performance of sentiment analysis for a discriminant domain, which affirms that the proposed model has certain generalization ability. Accordingly, our novel ensemble model exhibits acceptable reliability and relevancy to social media sentiment analysis and could be easily integrated with different classifiers to build novel frameworks within the minimum learning resources.

For future work, the ensemble task could be improved by integrating other ensemble methods like stacking. In addition, we could expand the proposed approach, utilizing different lexicon sources and machine learning techniques to boost the performance of classifiers in sentiment analysis.

REFERENCES

[1] D. R. Stöckli and H. Khobzi, "Recommendation systems and convergence of online reviews: The type of product network matters!" *Decis. Support Syst.*, vol. 142, Mar. 2021, Art. no. 113475.

[2] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, and B. Davis, "SemEval-2017 Task 5: Fine-grained sentiment analysis on financial microblogs and news," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 519–535.

[3] Q. Deng, B. F. Santos, and W. J. C. Verhagen, "A novel decision support system for optimizing aircraft maintenance check schedule and task allocation," *Decis. Support Syst.*, vol. 146, Jul. 2021, Art. no. 113545.

[4] S. Voorberg, R. Eshuis, W. van Jaarsveld, and G. J. van Houtum, "Decisions for information or information for decisions? Optimizing information gathering in decision-intensive processes," *Decis. Support Syst.*, vol. 151, Dec. 2021, Art. no. 113632.

[5] S. Rani and P. Kumar, "A sentiment analysis system to improve teaching and learning," *Computer*, vol. 50, no. 5, pp. 36–43, May 2017.

[6] P. Singh, R. S. Sawhney, and K. S. Kahlon, "Sentiment analysis of demonetization of 500 & 1000 rupee banknotes by Indian government," *ICT Exp.*, vol. 4, no. 3, pp. 124–129, Sep. 2018.

[7] S. Saha and A. Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data Knowl. Eng.*, vol. 85, pp. 15–39, May 2013.

[8] D. Tiwari and M. Kumar, "Social media data mining techniques: A survey," in *Information and Communication Technology for Sustainable Development (ICT4SD)*. Singapore: Springer, 2020, pp. 183–194.

[9] M. Vadivukarassi, N. Puviarasan, and P. Aruna, "Sentimental analysis of tweets using Naïve Bayes algorithm," *World Appl. Sci. J.*, vol. 35, no. 1, pp. 54–59, 2017.

[10] A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal, "Aspect term extraction for sentiment analysis in large movie reviews using Gini index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, Mar. 2017.

[11] M. R. Irfan, M. A. Fauzi, T. Tibyani, and N. D. Mentari, "Twitter sentiment analysis on 2013 curriculum using ensemble features and K-nearest neighbor," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 8, no. 6, p. 5409, Dec. 2018.

[12] B. Ma, H. Yuan, and Y. Wu, "Exploring performance of clustering methods on document sentiment analysis," *J. Inf. Sci.*, vol. 43, no. 1, pp. 54–74, Feb. 2017.

[13] Y. Ma, X. Geng, and J. Wang, "A deep neural network with multiplex interactions for cold-start service recommendation," *IEEE Trans. Eng. Manag.*, vol. 68, no. 1, pp. 105–119, Feb. 2021.

[14] M. A. Ferrag and L. Maglaras, "DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1285–1297, Nov. 2020.

[15] S. Al-Saqqa, N. Obeid, and A. Awajan, "Sentiment analysis for Arabic text using ensemble learning," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. 1–7.

[16] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[17] H. Kang, S. J. Yoo, and D. Han, "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 6000–6010, 6000.

[18] Y. Yang, C. Xu, and G. Ren, "Sentiment analysis of text using SVM," in *Electrical, Information Engineering and Mechatronics*, X. Wang, F. Wang, and S. Zhong, Eds. London, U.K.: Springer, 2011, pp. 1133–1139.

[19] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Predictive sentiment analysis of tweets: A stock market application," in *Proc. 3rd Int. Workshop Human-Comput. Interact. Knowl. Discovery Complex, Unstructured, Big Data (HCI-KDD)*, Maribor, Slovenia. Berlin, Germany: Springer, 2013, pp. 77–88.

[20] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of Twitter posts," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4065–4074, Aug. 2013.

[21] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentimental reviews using machine learning techniques," *Proc. Comput. Sci.*, vol. 57, pp. 821–829, Jan. 2015.

[22] X. Fang and J. Zhan, "Sentiment analysis using product review data," *J. Big Data*, vol. 2, no. 1, pp. 1–14, Dec. 2015.

[23] P. Wang, J. Li, and J. Hou, "S2SAN: A sentence-to-sentence attention network for sentiment analysis of online reviews," *Decis. Support Syst.*, vol. 149, Oct. 2021, Art. no. 113603.

[24] J. Kazmaier and J. H. van Vuuren, "A generic framework for sentiment analysis: Leveraging opinion-bearing data to inform decision making," *Decis. Support Syst.*, vol. 135, Aug. 2020, Art. no. 113304.

[25] Y. Su, Y. Zhang, D. Ji, Y. Wang, and H. Wu, "Ensemble learning for sentiment classification," in *Proc. 13th Workshop Chin. Lexical Semantics (CLSW)*, Wuhan, China. Berlin, Germany: Springer, Jul. 2013, pp. 84–93.

[26] M.-T. Martín-Valdivia, E. Martínez-Cámara, J.-M. Perea-Ortega, and L. A. Ureña-López, "Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 3934–3942, Aug. 2013.

[27] M. Govindarajan, "Bagged ensemble classifiers for sentiment classification of movie reviews," *IJECS*, vol. 3, no. 2, pp. 3951–3961, 2014.

[28] S. Winkler, S. Schaller, V. Dorfer, M. Affenzeller, G. Petz, and M. Karpowicz, "Data-based prediction of sentiments using heterogeneous model ensembles," *Soft Comput.*, vol. 19, no. 12, pp. 3401–3412, Dec. 2015.

[29] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles," *Decis. Support Syst.*, vol. 66, pp. 170–179, Oct. 2014.

[30] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decis. Support Syst.*, vol. 57, pp. 77–93, Jan. 2014.

[31] T. Chalothom and J. Ellman, "Simple approaches of sentiment analysis via ensemble learning," in *Information Science and Applications*. Berlin, Germany: Springer, 2015, pp. 631–639.

[32] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun. Dependable, Autonomic Secure Comput. Pervasive Intell. Comput.*, Oct. 2015, pp. 2359–2364.

[33] R. Alnashwan, A. P. O'Riordan, H. Sorensen, and C. Hoare, "Improving sentiment analysis through ensemble learning of meta-level features," in *Proc. CEUR Workshop*, vol. 1748. Aachen, Germany: RWTH Aachen University, 2016, pp. 1–13.

[34] C. Catal and M. Nangir, "A sentiment classification model based on multiple classifiers," *Appl. Soft Comput.*, vol. 50, pp. 135–141, Jan. 2017.

[35] K. Korovkinas, P. Danėnas, and G. Garšva, "SVM and Naïve Bayes classification ensemble method for sentiment analysis," *Baltic J. Mod. Comput.*, vol. 5, no. 4, pp. 398–409, 2017.

[36] M. Talib AL-Sharuee, F. Liu, and M. Pratama, "An automatic contextual analysis and clustering classifiers ensemble approach to sentiment analysis," 2017, *arXiv:1705.10130*.

[37] M. Shad Akhtar, D. Ghosal, A. Ekbal, P. Bhattacharyya, and S. Kurohashi, "A multi-task ensemble framework for emotion, sentiment and intensity prediction," 2018, *arXiv:1808.01216*.

[38] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of Twitter data," *Proc. Comput. Sci.*, vol. 111, pp. 376–381, Jan. 2017.

[39] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in *Proc. 7th Int. Annu. Eng. Seminar (InAES)*, Aug. 2017, pp. 1–4.

[40] Y. Lu, K. Sakamoto, H. Shibuki, and T. Mori, "Are deep learning methods better for Twitter sentiment analysis," in *Proc. 23rd Annu. Meeting Natural Lang. Process.*, 2017, pp. 787–790.

[41] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for Twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23253–23260, 2018.

[42] M. Hanafy, M. I. Khalil, and H. M. Abbas, "Combining classical and deep learning methods for Twitter sentiment analysis," in *Proc. 8th IAPR TC3 Workshop Artif. Neural Netw. Pattern Recognit. (ANNPR)*, Siena, Italy. Cham, Switzerland: Springer, Sep. 2018, pp. 281–292.

[43] M. T. Al-Sharuee, F. Liu, and M. Pratama, "Sentiment analysis: An automatic contextual analysis and ensemble clustering approach and comparison," *Data Knowl. Eng.*, vol. 115, pp. 194–213, May 2018.

[44] M. M. Fouad, T. F. Gharib, and A. S. Mashat, "Efficient Twitter sentiment analysis system with feature selection and classifier ensemble," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl. (AMLTA)*. Cham, Switzerland: Springer, 2018, pp. 516–527.

[45] M. López, A. Valdivia, E. Martínez-Cámara, M. V. Luzón, and F. Herrera, "E2SAM: Evolutionary ensemble of sentiment analysis methods for domain adaptation," *Inf. Sci.*, vol. 480, pp. 273–286, Apr. 2019.

[46] K. Yang, C. Liao, and W. Zhang, "A sentiment classification model based on multiple multi-classifier systems," in *Proc. 5th Int. Conf. Artif. Intell. Secur. (ICAIS)*, New York, NY, USA. Cham, Switzerland: Springer, Jul. 2019, pp. 287–298.

[47] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[48] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6256–6268.

[49] A. Kunte and S. Panicker, "Personality prediction of social network users using ensemble and XGBoost," in *Proc. Prog. Comput., Anal. Netw. (ICCAN)*, 2019, pp. 133–140.

[50] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[51] E. A. C. Júnior, V. Q. Marinho, and L. B. dos Santos, "NILC-USP at SemEval-2017 Task 4: A multi-view ensemble for Twitter sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 611–615.

[52] M. Devaraj, R. Piryani, and V. K. Singh, "Lexicon ensemble and lexicon pooling for sentiment polarity detection," *IETE Tech. Rev.*, vol. 33, no. 3, pp. 332–340, May 2016.

[53] G. Wang, J. Ma, and S. Yang, "IGF-bagging: Information gain based feature selection for bagging," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 11, pp. 6247–6259, 2011.

[54] B. S. Bhati and C. S. Rai, "Analysis of support vector machine-based intrusion detection techniques," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 2371–2383, Apr. 2020.

[55] T. Yiu. *Understanding Random Forest. How the Algorithm Works and Why It is*. Towards Data Sci. Accessed: Sep. 1, 2024. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2

[56] J. Jotheeswaran and S. Koteeswaran, "Feature selection using random forest method for sentiment analysis," *Indian J. Sci. Technol.*, vol. 9, no. 3, pp. 1–7, Feb. 2016.

[57] H. Jung. *Adaboost for Dummies: Breaking Down the Math (and Its Equations) Into Simple Terms*. Towards Data Science. Accessed: Sep. 1, 2024. [Online]. Available: https://towardsdatascience.com/adaboost-for-dummies-breaking-down-the-math-and-its-equations-into-simple-terms-87f439757dcf

[58] K. K. Mahto. *Demystifying Maths of Gradient Boosting*. Towards Data Science. Accessed: Sep. 1, 2024. [Online]. Available: https://towardsdatascience.com/demystifying-maths-of-gradient-boosting-bd5715e82b7c

[59] M. Shafipour, A. Rashno, and S. Fadaei, "Particle distance rank feature selection by particle swarm optimization," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115620.

[60] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, Mar. 2020.

[61] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic N-grams as machine learning features for natural language processing," *Expert Syst. Appl.*, vol. 41, no. 3, pp. 853–860, Feb. 2014.

[62] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometric Intell. Lab. Syst.*, vol. 149, pp. 153–165, Dec. 2015.

[63] P. du Jardin, "Forecasting corporate failure using ensemble of self-organizing neural networks," *Eur. J. Oper. Res.*, vol. 288, no. 3, pp. 869–885, Feb. 2021.
[64] N. Omar, M. Albared, A. Al-Shabi, and T. Al-Moslmi, "Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customers' reviews," *Int. J. Advancements Comput. Technol.*, vol. 5, pp. 77–85, Jan. 2013.
[65] B. Behera, G. Kumaravelan, and B. P. Kumar, "Performance evaluation of deep learning algorithms in biomedical document classification," in *Proc. 11th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2019, pp. 220–224.
[66] *How to Assess Quality and Correctness of Classification Models? Part 4—Roc Curve—Algolytics*. Accessed: Sep. 1, 2024. [Online]. Available: https://algolytics.com/how-to-assess-quality-and-correctness-of-classification-models-part-4-roc-curve/

DIMPLE TIWARI received the Ph.D. degree in computer and engineering from NSUT East Campus (formerly Ambedkar Institute of Advanced Communication Technologies & Research), GGSIPU, Delhi, India, in 2023. She is currently an Assistant Professor with the ABES Engineering College, Ghaziabad, India. She has published various research articles in highly reputed, SSCI/SCI/SCIE-indexed journals. Her research interests include natural language processing, social network analysis, sentiment analysis, disease prediction, intrusion detection, information security, and the data science with machine learning and IoT.

BHARTI NAGPAL is currently an Assistant Professor with the Computer Science and Engineering Department, NSUT East Campus (formerly Ambedkar Institute of Advanced Communication Technologies & Research), Delhi, India. She has 21 years of teaching experience. She has published various research papers in reputed international journals/conferences and contributed book chapters. Her research interests include sentiment analysis, artificial intelligence, information security, data mining and data warehouse, the Internet of Things (IoT), and big data.

BHOOPESH SINGH BHATI received the Ph.D. degree in computer and Engineering from the University School of Information Communication and Technology, Guru Gobind Singh Indraprastha University, Delhi, in 2020. He is currently an Assistant Professor with the Indian Institute of Information Technology Sonipat (Institution of National Importance) under the Ministry of Education, Government of India. He has published various research papers in highly reputed, SSCI/SCI/SCIE-indexed journals. His research interests include intrusion detection, information security, and the data science with machine learning and IoT.

MANOJ GUPTA (Member, IEEE) received the B.Tech. degree from the Institute of Engineering & Technology, M. J. P. Rohilkhand University, Bareilly, Uttar Pradesh, India, the M.Tech. degree from HNB Garhwal Central University, Uttarakhand, India, and the Ph.D. degree from the University of Rajasthan, Jaipur, India. His overall experience is more than 18 years. He has published more than 60 research papers in international journals and national/international conferences and book chapters. He is having two patent grants and published five patents in his credit and currently having two Indian research copyright in his credit. He is the Editor-in-Chief (EiC) of the Book Series *Advances in Antenna, Microwave and Communication Engineering* (Scrivener and John Wiley, USA), *Advances in Antenna Design, Wireless Communication and Mobile Network Technology* (CRC Press and Taylor & Francis, USA), *Advances in Digital Signal Processing and Image Processing for Industrial Applications* (CRC Press and Taylor & Francis, USA), and *AAP Book Series on Digital Signal Processing, Computer Vision and Image Processing* (Apple Academic Press, USA). His research interests include wireless and multimedia communication, biomedical image and signal processing, soft computing, intelligent computing and computational intelligence, the Internet of Things (IoT), artificial intelligence and machine learning, healthcare informatics, and embedded systems. He is a member of many professional bodies, such as IACSIT, ISTE, and IAENG. He has served as a Technical Programme Committee (TPC) Member and a Reviewer in various international conferences, such as ICCIA 2020, ICCIA 2019 ICSIP 2018, ICSIP 2017, ICSIP 2016, AIPR 2017, AIPR 2016, ICCIA 2018, ICCIA 2017, ICCIA 2016, and ICNIT 2018. He was invited for a Keynote Speaker/Invited Speaker, in 2017, Second IEEE International Conference on Signal and Image Processing (ICSIP 2017), in August 2017, in Nanyang Executive Centre, Singapore, and Invited for a Keynote Speaker, in 2017, International Conferences on Public Health and Medical Sciences (ICPHMS 2017) in Xi'an, China, in May 2017. He is an editor, an associate editor, and a reviewer of many international journals. His name has been listed in Marquis Who's Who in Science and Engineering®USA and Marquis Who's Who in the World®USA.

PANNEE SUANPANG (Member, IEEE) received the Bachelor of Information Systems (B.I.T) and Master of Information Systems (M.I.S.) degrees from Griffith University, Australia, in 1997 and 2001, respectively, and the Doctor of Technology (D.Tech.) degree in science from the University of Technology, Sydney, Australia, in 2005. She is currently an Associate Professor with the Department of Information Technology, Suan Dusit University. Until now, she has published eight Scopus/WoS indexed documents. Her research interests include advanced information technology in agricultural, big data, the IoT, and smart tourism. She is an editor and a reviewer of many famous journals in the world.

SUJIN BUTDISUWAN received the M.A. degree from Chulalongkorn University, Thailand, and the Ph.D. degree from Illinois State University, USA. He is currently the Dean of Metharath University (Shinawatra University), Thailand. He was the Vice-President of Students' Development and Supports with Mahasarakham University, Thailand, where he was the former Dean of the Faculty of Informatics and the former Director of the Academic Resource Center, known as the university library for four years. He was an Assistant Professor with the Faculty of Informatics, Mahasarakham University. He has been actively involved in the LIS profession in various aspects, such as an invited lecturer, an examiner, and a specialist for some universities. He is a lifetime member of the Thai Library Association (TLA). In 2007, he received the Award for his Outstanding Contributions to the Profession from the TLA.

AZIZ NANTHAAMORNPHONG (Member, IEEE) received the Ph.D. degree from the University of Alabama, USA. He is currently an Associate Professor and the Dean of the College of Computing, Prince of Songkla University, Phuket Campus, Thailand. With an extensive academic background, he specializes in empirical software engineering and data science. His research significantly contributes to the development of scientific software and leverages data science in the field of tourism. In addition to his core focus, he is also deeply engaged in the study of human–computer interaction, pioneering innovative approaches to foster a beneficial interplay between humans and technology.