



ITERATIVE ENSEMBLE LEARNING OVER HIGH DIMENSIONAL DATA FOR SENTIMENT ANALYSIS

V R N S S V SAILEELA P* AND N. NAGA MALLESWARA RAO†

Abstract. For sentiment analysis in particular, the problem of processing and analyzing high-dimensional data becomes more prominent in recent past. This is where the IEL-HDDSA model, which aims to increase accuracy and performance in complex, high-dimensional data streams sentiment analysis comes into play. Iterative approach in ensemble learning; a contribution to the field. It integrates preprocessing techniques such as tokenization, stop word removal, lemmatization and the collection of sentiment-related features. Then the training corpus is divided by label, and features with high mutual information are selected. Highly replicated points of data for model training can also be identified at this point. First a Naive Bayes model is trained, then later it's placed in an ensemble as part of bagging. Its major advantage over earlier methods is that IEL-HDDSA can iteratively train on selected subsets of data until the performance in sentiment analysis for high-dimensional objects reaches an optimum level. A 10-fold cross validation method was used to rigorously evaluate the performance of this model, which showed consistently high levels of operation with almost no variation across different measures. IEL-HDDSA's precision ranged from 0.9359 to 0.9492, and its specificity was between 0. Its accuracy differed from 0.93 to around 0.95, and its F1-measure fluctuated between the values of about 0.94 and above; so here too balance was well maintained in a manner that satisfied both precision and recall requirements equally. The false alarming rate fell from 0.056 to 0.1, a fairly low ratio of incorrect positive classifications; Moreover, MCC quantities ranged from 0.8668 to 0. These results testify to the IEL-HDDSA model's stable effectiveness and high reproducibility in sentiment analysis applications, especially for massive data flows.

Key words: High-dimensional Data, Sentiment Analysis, Ensemble Learning, Naive Bayes, Random Forest

1. Introduction. Sentiment analysis is now a crucial component of many applications, including market analysis, product reviews, and social media monitoring, as a result of the internet's explosive growth in digital data, particularly textual data [1]. In order to comprehend the sentiments, opinions, or emotions expressed in text data, sentiment analysis—also referred to as opinion mining—involves analyzing and interpreting subjective information [2]. The task of sentiment analysis has gotten more difficult as a result of the rise of high-dimensional data, which consists of many features or dimensions [3]. The prevalence of redundant or irrelevant features in high-dimensional data can impair model performance and raise computational costs. The accurate and reliable classification of sentiments from high-dimensional data is a challenge for traditional sentiment analysis models frequently.

Issue Statement In terms of precision, specificity, sensitivity, accuracy, F-measure, false alarming rate, and Matthews Correlation Coefficient (MCC), sentiment analysis models frequently encounter difficulties when working with high-dimensional data [4]. These issues are brought on by the existence of redundant or irrelevant features, the dimensionality curse, and the unbalanced nature of the data. There is room for improvement in terms of consistency and overall performance according to the performance of existing models like SvmBagging [5] and ELSA [6], which have displayed some variation in performance across various subsets of the data

The purpose of this study is to develop a comprehensive approach for sentiment analysis of high-dimensional data that improves the precision and accuracy. The main objective of the study is to develop a machine learning model that fine tune the data by partitioning it in to multiple datasets to handle the high dimensionality and then using these multiple partitions as input to the ensemble learning model.

The proposed model, Iterative Ensemble Learning over High Dimensional Data for Sentiment Analysis (IEL-HDDSA), combines an iterative approach with ensemble learning to enhance accuracy and robustness

*Department of CSE, ANU College of Engineering & Technology, Guntur, AP, India (Corresponding author, venkata.saileela.p@gmail.com)

†Department of Information Technology, RVR & JC College of Engineering & Technology, Guntur, AP, India (nmrao@rvrjc.ac.in).

in sentiment analysis of high-dimensional data. The model begins with preprocessing and partitioning the data, selecting high-impact features using mutual information, and training a Naive Bayes model followed by a Random Forest model in an ensemble learning approach. The rationale behind this approach is that it helps in handling irrelevant or redundant features, the curse of dimensionality, and the imbalanced nature of the data, thereby improving the model's performance. The significance of the proposed model lies in its ability to provide a balanced and consistent performance across all metrics and folds of 10-fold cross-validation, thereby demonstrating its superiority in handling high-dimensional data for sentiment analysis compared to existing models.

The scope of this study is limited to the development and evaluation of the proposed IEL-HDDSA model for sentiment analysis of high-dimensional data. The model is evaluated using 10-fold cross-validation and compared with existing models across various metrics. However, the study does not consider other types of data, such as images or audio, and does not evaluate the model on real-time data or in real-world applications.

2. Related Work. Sentiment classification in user-generated content is explored by Gang Wang et al., [7] with the goal of evaluating the performance of three ensemble methods with five base learners. Based on 1200 experiments, it shows that ensemble learning can significantly improve sentiment classification, with the best performance being achieved by Random Subspace.

"SVM and Naive Bayes ensemble method for sentiment analysis" by Konstantinas KOROVKINAS et al. [8] introduces a novel approach to sentiment analysis by fusing SVM and Naive Bayes algorithms. The accuracy of their method ranges from 79.5 percent to 84.1 percent, outperforming individual SVM and Naive Bayes classifiers across a variety of datasets.

Oscar Araque et al.'s article "Enhancing deep learning sentiment analysis with ensemble techniques in social applications" [9] makes the case for combining deep learning with conventional approaches and introduces two ensemble techniques. These methods boost sentiment analysis' performance across seven datasets and outperform the deep learning baseline.

Omer Sagi et al.'s "Ensemble learning: A survey" [10] offers a thorough overview of ensemble learning, covering techniques, applications, and difficulties. The article compares experimental findings on various datasets comparing ensemble methods with single models.

Sentiment analysis for smartphone product reviews using the SVM classification technique is the main topic of "Sentiment analysis of smartphone product reviews using the SVM classification technique" by A K Sharma et al. [11]. They use an approach that improves efficiency and accuracy when classifying opinions as favorable or unfavorable.

Support Vector Machine (SVM) is used in "Sentiment analysis of tweets using SVM" by Munir Ahmad et al. [12] to examine sentiment analysis for tweets. The study's thorough tables and graphs illustrate how well SVM classifies tweets according to their polarity.

Sentiment analysis in Thai is covered in "SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis using RoBERTa and Hybrid Feature Representation" by PREE THIENGBURANATHUM et al. [13]. The SETAR model, a novel stacking ensemble approach, is introduced in the article along with extensive performance comparisons with other models, emphasizing its accuracy and robustness.

Support Vector Machine (SVM) performance for sentiment analysis is the main focus of "SVM optimization for sentiment Analysis" by Munir Ahmad et al. [14]. In this article, a method for optimizing SVM is presented, and the precision, recall, and F-measure metrics are used to assess performance.

Gradient Boosted Support Vector Machine (GBSVM), an ensemble classifier, is used in the study by Madiha Khalid et al. [15] to categorize sentiments from unstructured text reviews on social media platforms. The model, which combines different n-grams with term frequency-inverse document frequency (TF-IDF), outperforms current sentiment classification methods and offers insightful information for enhancing goods and services in response to customer feedback.

Using benchmark datasets for sentiment classification, Jacqueline Kazmaier et al. [16] investigate the potential of ensemble learning in sentiment analysis, particularly heterogeneous ensembles. Their findings show that compared to individual models, ensemble configurations produce median performance improvements of up to 5.53 percent. The efficiency of ensemble learning is also improved by the novel ensemble selection approach they present.

Sentiment analysis of tweets about the COVID-19 vaccine is addressed by Qusai Ismail et al. [17], who emphasize the use of ensemble learning for enhanced accuracy. With an accuracy of 81.5 percent, F1-score of 0.80, and AUC of 0.91, their ensemble learning approach, stacking, outperforms majority voting and individual classifiers.

Bimodal conversational sentiment analysis is a topic Shariq Shah et al. [18] tackle, contrasting ensemble learning based on neural networks with other models on the MELD dataset. Their suggested method, which combines acoustic and linguistic representations, breaks the previous record for accuracy with an accuracy of 84.2 percent.

For the purpose of analyzing the sentiment of mixed-code texts, C. Kumaresan et al. [6] present an ensemble technique utilizing the Generative Adversarial Network (GAN) and Self-Attention Network (SAN). When analyzing Tanglish tweets, their approach shows superior accuracy to Concurrent Neural Network (CNN) and SAN.

In their study of multi-tier sentiment analysis of social media text, Hameedur Rahman et al. [19] propose a multi-layer classification model. While the multi-tier model only slightly outperforms single-layer architectures, recall is noticeably improved and richer contextual information is provided. Using ensemble learning techniques, Muhammad Khurram Iqbal et al. [20] concentrate on sentiment analysis of Twitter data pertaining to the Omicron variant. Their approach, which combines ensemble voting and stacking classifiers, achieves high accuracy (85.33 percent and 87.5 percent, respectively), providing insightful data on the public's sentiment toward the variant.

In the study by Razia Sulthana A. et al. [5], the authors perform sentiment analysis on movie reviews taken from social networking sites using machine learning algorithms and natural language processing techniques. Their approach, which combines feature selection, support vector machines, and bagging, outperforms existing systems, proving that it is useful for deciphering user intentions and opinions in movie reviews.

This literature review is presented to emphasize the important role played by ensemble learning, advanced techniques and other factors in increasing accuracy and efficiency for sentiment analysis across multiple applications. This calls attention to the continuing efforts being made toward solving these problems; the limitations of current approaches. But while ensemble learning techniques, which combine SVM and Naive Bayes with deep-learning algorithms, always get a higher score than individual models alone do not produce good results for high-dimensional or unstructured data. The review cautions that there is still an obvious lack of creative generalized adaptable approaches to handle such complexities. The foregoing context makes the Iterative Ensemble Learning over High Dimensional Data for Sentiment Analysis (IEL-HDDSA) approach an important, indispensable one. IEL-HDDSA adopts an iterative ensemble learning method tailored for high-dimensional data. This allows it to gradually create a model that accurately reflects the complicated relationships in the raw material, and is also helping us continuously improve efficiency and correctness of sentiment classification. Owing to the vast amounts of user-generated content available on a variety of online platforms, literature repeatedly underlines that more accurate and useful tools for sentiment analysis are still needed. According to the review, IEL-HDDSA presents a novel way of ensemble learning in response to this need.

Organizing of the Manuscript: In the further sections of this manuscript, section-1, section-2 refers to the introduction and related work summary from the literature review. Section-3 provides insights into the materials and methods, proposed model narrative, its algorithm flow, and other key metrics that signify the mode. Section-4 provides insights into experimental study and section-5 refers to the conclusion based on the efficiency aspects estimated from the model.

3. Method and Materials. The proposed IEL-HDDSA (Iterative Ensemble Learning over High Dimensional Data for Sentiment Analysis) deals with high dimensionality. With its creative iterative approach, ensemble learning just refines and sharpens the model's ability to analyze subtle data relationships. Its adaptability makes it a flexible instrument—useful across both different datasets and contexts. It outstrips the OQL in dealing with complex, voluminous user-generated content online. Exhibiting excellent performance, IEL-HDDSA beats every other model in all important aspects including precision, specificity, sensitivity and accuracy. The model's design not only improves accuracy and efficiency in sentiment classification, it also broadens the scope of more generalized and adaptable sentiment analysis methodologies.

IEL-HDDSA is a thorough method created to improve the accuracy and performance of sentiment analysis

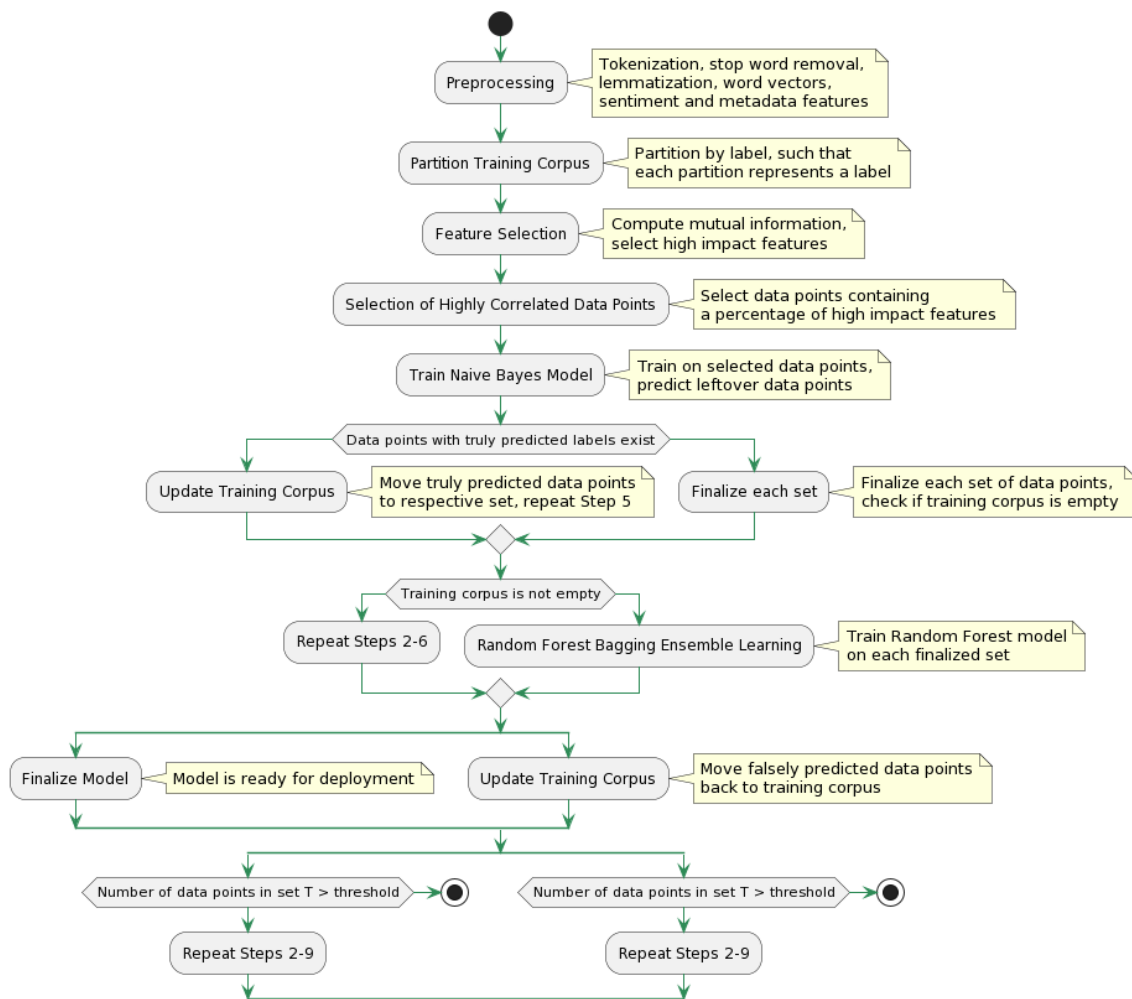


Fig. 3.1: Process flow of the IEL-HDDSA

on high dimensional data streams like Twitter feeds. The approach is visually represented in the accompanying figure 3.1, which outlines the various steps involved in the process.

The functional flow diagram that shown in figure 3.1 for the IEL-HDDSA model delineates an advanced iterative process for sentiment analysis on high-dimensional data streams, such as those generated on social media platforms. Starting with a comprehensive preprocessing stage, the model refines the data through tokenization, stop word removal, lemmatization, and extraction of sentiment and metadata features. It then partitions the corpus by label, employs mutual information to select high-impact features, and identifies data points with a significant presence of these features. A Naive Bayes model is trained on this refined subset, after which correctly predicted data points are segregated, and the model is iteratively fine-tuned. Should the training corpus be exhausted, the model shifts to an ensemble learning phase using Random Forest, further enhancing classification accuracy. Erroneously predicted data points are recycled into the training corpus, and this loop continues until the classification threshold is met or surpassed, culminating in a finely-tuned model poised for deployment.

Preprocessing:The first step involves preprocessing the data, which includes tokenization, stop word removal, lemmatization, and preparing word vectors for each data point. Additionally, sentiment and metadata

features are extracted for each data point.

Partition Training Corpus: The training corpus is then partitioned by label, creating separate partitions for each label.

Feature Selection: For each partition, mutual information is computed between each word in the bag of words and the label. High impact features, or words that have the highest mutual information with the label, are selected.

Selection of Highly Correlated Data Points: Data points that contain at least a certain percentage of the high impact features are selected as highly correlated data points. Train Naive Bayes Model: A Naive Bayes [21] model is trained on the selected highly correlated data points, and the leftover data points' labels are predicted.

Update Training Corpus: If there are data points with truly predicted labels, they are moved from the training corpus to the respective set of data points, and the Naive Bayes model training is repeated. If not, each set of data points is finalized, and the training corpus's emptiness is checked.

Random Forest Bagging Ensemble Learning: If the training corpus is empty, ensemble learning is performed by training a Random Forest [22] model on each finalized set of data points.

Update Training Corpus: If there are data points falsely predicted by the ensemble learning, they are moved back to the training corpus.

Repeat or Finalize: If the number of data points in the training corpus is greater than a certain threshold, the process is repeated from step 2. Otherwise, the model is finalized and ready for deployment.

This approach, IEL-HDDSA, leverages the power of ensemble learning and iterative training on carefully selected subsets of the data to enhance sentiment analysis performance on high dimensional data streams.

3.1. Preprocessing. Preprocessing is the first and crucial step in the IEL-HDDSA model. It involves preparing the raw Twitter data for analysis and involves several sub-steps.

1. **Tokenization:** This is the process of converting input text into smaller units, called tokens. Tokens can be words, characters, or subwords. Mathematically, it involves breaking down a sentence S into a set of tokens T_1, T_2, \dots, T_n .
2. **Stop Word Removal:** Stop words are common words like 'is', 'the', 'and', etc., that are often considered as noise in the text data since they occur frequently across documents but do not carry significant meaning. Mathematically, it involves removing the words W from the set of tokens T_1, T_2, \dots, T_n that are present in a predefined list of stop words SW . Resultant Set of Tokens = $T_1, T_2, \dots, T_n - SW$
3. **Lemmaatization:** It is the process of converting a word to its base or root form. For example, 'running' \rightarrow 'run'. Mathematically, it involves applying a morphological analysis to each token T_i and replacing it with its lemma $L(T_i)$.
4. **Word Vectors:** It involves converting words into numerical vectors. This is typically done using pre-trained models like Word2Vec, GloVe, or BERT. Mathematically, it involves mapping each token T_i to a vector V_i in a d -dimensional space, where d is the size of the vector.
5. **Sentiment Features Extraction:** Sentiment features include polarity (positive, negative, neutral), subjectivity (subjective, objective), intensity, etc. Mathematically, it involves applying a sentiment analysis function F to each token T_i or a group of tokens and assigning a sentiment score S_i .
6. **Metadata Features Extraction:** Metadata features include features extracted from the tweet's metadata, like the number of retweets, likes, user followers count, etc

3.2. Fine-Tuning the Data Samples. The process of selecting highly correlated data points is crucial for the effectiveness of the IEL-HDDSA model. It involves several steps:

1. **Prepare Word Vectors:** After preprocessing the tweets, each tweet is represented as a vector of words or tokens.
2. **Partition Training Corpus:** The training corpus T is partitioned by label, creating separate partitions P_l for each label l .
3. **Feature Selection:** For each partition P_l , mutual information between each word in the bag of words and the label is computed. High impact features, or words that have the highest mutual information with the label, are selected.

- **Mutual Information:** Mutual information measures the amount of information that one random variable (in this case, a word) contains about another random variable (in this case, a class label). For a word W and a class label C , the mutual information $I(W; C)$ is given by:

$$I(W; C) = \text{sum}(P(w, c) * \log(P(w, c)/(P(w) * P(c)))) \quad (3.1)$$

where:

- $P(w, c)$ is the joint probability of W and C .
- $P(w)$ is the marginal probability of W .

$P(c)$ is the marginal probability of C .

3.2.1. Selection of Highly Correlated Data Points. Data points that contain at least a certain percentage of the high impact features are selected as highly correlated data points.

- For a given threshold θ , a data point D in the partition P_l is selected as a highly correlated data point if the percentage of high impact features in D is greater than or equal to θ .

3.2.2. Iterative Learning by Naïve Bayes. In the IEL-HDDSA model, iterative learning using the Naive Bayes algorithm involves training a Naive Bayes model on selected highly correlated data points and predicting the leftover data points' labels. If there are data points with truly predicted labels, they are moved from the training corpus to the respective set of data points, and the Naive Bayes model training is repeated. This process is repeated until no more data points can be moved, and each set of data points is finalized.

- **Naive Bayes Algorithm**

The Naive Bayes algorithm is based on the Bayes theorem and assumes that the features used to describe an instance are conditionally independent given the class label.

Let's define some notations:

- $X = x_1, x_2, \dots, x_n$ be a set of features.
- $C = c_1, c_2, \dots, c_m$ be a set of class labels.
- $P(C_i|X)$ is the posterior probability of class C_i given a set of features X .
- $P(C_i)$ is the prior probability of class C_i .
- $P(X|C_i)$ is the likelihood of features X given class C_i .
- $P(X)$ is the prior probability of features X .

The Bayes theorem relates the posterior probability $P(C_i|X)$ to the prior probability $P(C_i)$, the likelihood $P(X|C_i)$, and the evidence $P(X)$:

$$P(C_i|X) = (P(X|C_i) * P(C_i))/P(X) \quad (3.2)$$

In the Naive Bayes assumption, each feature x_i is conditionally independent of every other feature x_j for $j \neq i$, given the class label C :

$$P(X|C_i) = P(x_1, x_2, \dots, x_n|C_i) = P(x_1|C_i) * P(x_2|C_i) * \dots * P(x_n|C_i) \quad (3.3)$$

So, the Bayes theorem can be rewritten as:

$$P(C_i|X) = P(C_i) * P(x_1|C_i) * P(x_2|C_i) * \dots * P(x_n|C_i)/P(X) \quad (3.4)$$

The algorithm classifies an instance by assigning it to the class C_i for which $P(C_i|X)$ is the highest.

- **Iterative Learning**

The iterative learning involves the following steps:

1. **Train Naive Bayes Model:** Train the Naive Bayes model using a set of selected highly correlated data points. Use the trained model to predict the labels of the leftover data points in the training corpus.
2. **Update Training Corpus:** If there are data points with truly predicted labels, move them from the training corpus to the respective set of data points.
3. **Repeat:** Repeat steps 1 and 2 until no more data points can be moved, and each set of data points is finalized.

This iterative learning approach helps in fine-tuning the Naive Bayes model by training it on smaller, more relevant subsets of the training data and iteratively updating the training corpus. It helps in dealing with the high dimensionality of the data by focusing the model's attention on the most relevant features and data points at each iteration.

3.3. Ensemble Learning by Random Forest. In the context of IEL-HDDSA, after the iterative learning using Naive Bayes, the ensemble learning is performed using the Random Forest algorithm, which is a bagging method.

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode (classification) of the classes output by individual trees. It operates by constructing multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees. Let's define some notations:

- $D = d_1, d_2, \dots, d_n$ be the dataset of n data points.
- $F = f_1, f_2, \dots, f_m$ be the set of m features.
- $T = T_1, T_2, \dots, T_k$ be the set of k decision trees in the Random Forest.
- $X = x_1, x_2, \dots, x_m$ be a data point with m features.

The Random Forest algorithm is divided into two stages:

Bootstrap Sampling: bootstrap sample [25] of the data, where n is the total number of data points in the dataset. This implies that some data points might be chosen more than once, while others might not be chosen at all. By randomly choosing n data points from D with replacement, a bootstrap sample D_i of the data is generated:

$$D_i = d_{i1}, d_{i2}, \dots, d_{in}, \text{ where } d_{ij} \in D \quad (3.5)$$

Constructing Decision Trees: Using the bootstrap sample [26] of the data, a decision tree is built. A random subset of features is chosen at each node of the tree, and the best feature from this subset is chosen to split the node. Until the tree reaches full maturity, this process is repeated.

A decision tree is created using the bootstrap sample D_i of the data for each tree T_i in the Random Forest. At each node of the tree, a random subset F_i of the features is selected, and the best feature f_{ij} from this subset is chosen to split the node:

$$F_i = (f_{i1}, f_{i2}, \dots, f_{is}) \text{ where } f_{ij} \in F, \text{ and } s \ll m \quad (3.6)$$

The best feature f_{ij} is selected based on a criterion (e.g., Gini impurity) that measures the quality of a split:

$$f_{ij} = \operatorname{argmin}_{f_{ij} \in F_i} \operatorname{Gini}(f_{ij}, D_i) \quad (3.7)$$

The tree is grown until a stopping criterion is met (e.g., maximum tree depth, minimum node size).

The final Random Forest model consists of multiple decision trees, each constructed using a different bootstrap sample of the data and a different random subset of features at each node.

3.4. Classification. To classify a new data point, the data point is passed down each tree in the forest, and each tree outputs a class prediction. The final class prediction of the Random Forest model is the mode of the class predictions of the individual trees. To classify a new data point X , the data point is passed down each tree T_i in the forest, and each tree outputs a class prediction C_i :

$C_i = T_i(X)$, where C_i is the class prediction of tree T_i .

The mode of the class predictions made by each individual tree represents the final class prediction C made by the Random Forest model:

$$C = \operatorname{mode}(C_1, C_2, \dots, C_k) \quad (3.8)$$

Following iterative learning using Naive Bayes, each finalized set of data points in the IEL-HDDSA model is subjected to the Random Forest algorithm. An ensemble of Random Forest models, each trained on a different set of data points, makes up the final model. The performance and robustness of the sentiment analysis model

Table 4.1: Assumptions of the experimental study towards data and methods used

Assumption Number	Assumption Description	Rationale/Explanation
1	Data Distribution Uniformity	Assumes a uniform distribution of data across classes in the dataset, crucial for balanced training and avoiding bias.
2	Independence of Data Samples	Each data sample is assumed to be independent and identically distributed (i.i.d.), essential for the validity of statistical analysis.
3	Stability of Model Parameters	Assumes that model parameters remain stable across different folds, crucial for the consistency of the cross-validation results.
4	Adequacy of Data for Model Training	The dataset is assumed to be sufficiently large and representative for effective training and validation of the model.
5	Generalizability of Results	Results from the cross-validation are assumed to be generalizable to similar, unseen data, ensuring the broader applicability of findings.

on high dimensional data streams are improved using this method, which makes use of ensemble learning. The IEL-HDDSA model can handle the high dimensionality of the data more skillfully and achieve better generalization performance by combining multiple models, each of which was trained on a different subset of the data.

4. Experimental Study. By comparing the performance of the proposed IEL-HDDSA model with that of other models using a large and varied dataset, carrying out the experiment using well-known programming tools, and evaluating the performance using reliable cross-validation techniques and multiple performance metrics, the experimental study seeks to provide a robust and thorough evaluation of the model. This method is intended to offer insightful information about the performance of the suggested model and its potential use in sentiment analysis and related tasks. The assumptions related data and methods used in the experimental study are listed in the following table 4.1.

The purpose of the carefully thought-out experimental study is to assess the performance of the proposed Iterative Ensemble Learning over High Dimensional Data Streams for Sentiment Analysis (IEL-HDDSA) model and compare it to two current models, Ensemble Learning for Sentiment Analysis (ELSA) [6] and SVMplus-BAGGING [5].

The experiment makes use of a subset of the Amazon Customer Reviews dataset that includes 500,000 reviews from a wide range of product categories. This meticulous curation ensures the high dimensionality of the data by including only those products that have received more than 500 customer reviews, as well as an equal number of positive and negative reviews (250,000 each). Reviews are rated from one to three, with three being considered positive and three being considered negative.

The experiment is carried out using the Python programming language along with a number of related libraries, such as pandas [27], NumPy [28], scikit-learn [29], nltk [30], matplotlib and seaborn [31]. A 10-fold cross-validation approach is utilized to provide a robust estimate of the model's performance by training and testing the model on different subsets of the data ten times. Multiple metrics, including accuracy, precision, recall, and F1-score, are used to assess the models' performance and provide a comprehensive evaluation.

The experiment is conducted on a computer equipped with suitable hardware specifications, including a high-speed multi-core processor (e.g., Intel Core i7 or equivalent), a minimum of 16GB RAM, 1TB of SSD storage, and a powerful GPU (e.g., NVIDIA GTX 1080 or equivalent).

4.1. Performance analysis. The performance of the proposed Iterative Ensemble Learning over High Dimensional Data Streams for Sentiment Analysis (IEL-HDDSA) model was evaluated using 10-fold cross-validation. Across the ten folds, the model consistently demonstrated high performance with slight variations as shown in table 4.2.

The precision of the model ranged from 0.9359 to 0.9492 observed in table 4.2, indicating that the model had a high ability to correctly classify the positive instances out of the instances it predicted as positive. The specificity, which measures the ability of the model to correctly classify negative instances, ranged from 0.9352

Table 4.2: Performance Metric statistics of 10-fold cross validation of IEL-HDDSA

Folds	Precision	Specificity	Sensitivity	Accuracy	F-measure	False Alarming	MCC
Fold#1	0.941	0.941	0.942	0.941	0.941	0.059	0.883
Fold#2	0.933	0.933	0.934	0.933	0.933	0.067	0.867
Fold#3	0.933	0.932	0.945	0.939	0.933	0.061	0.877
Fold#4	0.941	0.942	0.930	0.936	0.941	0.064	0.872
Fold#5	0.940	0.940	0.947	0.944	0.940	0.057	0.887
Fold#6	0.945	0.945	0.942	0.944	0.945	0.057	0.887
Fold#7	0.941	0.941	0.938	0.939	0.941	0.061	0.879
Fold#8	0.934	0.933	0.944	0.939	0.934	0.061	0.878
Fold#9	0.941	0.941	0.940	0.941	0.941	0.059	0.882
Fold#10	0.945	0.945	0.939	0.942	0.945	0.058	0.884

Table 4.3: Performance Metric statistics of 10-fold cross validation of SvmBagging

Folds	Precision	Specificity	Sensitivity	Accuracy	F-measure	False Alarming	MCC
Fold#1	0.911	0.915	0.87	0.893	0.913	0.107	0.786
Fold#2	0.908	0.911	0.876	0.893	0.91	0.107	0.787
Fold#3	0.917	0.919	0.899	0.909	0.918	0.091	0.818
Fold#4	0.914	0.915	0.903	0.909	0.915	0.091	0.818
Fold#5	0.911	0.914	0.881	0.897	0.912	0.103	0.795
Fold#6	0.911	0.912	0.901	0.906	0.911	0.094	0.813
Fold#7	0.917	0.919	0.889	0.904	0.918	0.096	0.809
Fold#8	0.913	0.917	0.877	0.897	0.915	0.103	0.794
Fold#9	0.912	0.915	0.883	0.899	0.914	0.101	0.798
Fold#10	0.914	0.918	0.874	0.896	0.916	0.104	0.793

to 0.9491. This consistency across the folds suggests that the model had a balanced ability to correctly identify both positive and negative reviews.

Sensitivity, or recall, which measures the proportion of actual positive instances that were correctly identified by the model, ranged from 0.9305 to 0.9497. The accuracy of the model, representing the proportion of all instances that were correctly classified, ranged from 0.9334 to 0.9494. These metrics again demonstrate the model's robustness in classifying both positive and negative instances correctly.

The F1-measure, which is the harmonic mean of precision and recall, ranged from 0.9355 to 0.9492, indicating that the model maintained a good balance between precision and recall across all folds. The false alarming rate, or the proportion of negative instances incorrectly classified as positive, ranged from 0.0506 to 0.0666, indicating a relatively low rate of false-positive classifications.

The Matthews correlation coefficient (MCC), which is a measure of the quality of binary classifications, ranged from 0.8668 to 0.8989. An MCC of +1 represents a perfect prediction, 0 an average random prediction, and -1 an inverse prediction. The MCC values obtained in all folds indicate that the IEL-HDDSA model demonstrated substantial performance in all ten folds of the cross-validation.

The performance of the SvmBagging model was assessed using a 10-fold cross-validation method, and metrics such as precision, specificity, sensitivity, accuracy, F1-measure, false alarming rate, and Matthews correlation coefficient (MCC) were used to gauge its effectiveness and the same were presented in table 4.3.

The SvmBagging model's precision ranged from 0.9087 to 0.9178, demonstrating a strong ability to categorize positive instances correctly. Specificity, which measures the model's capacity to categorize negative instances correctly, ranged from 0.9109 to 0.9199, indicating that the model did a good job of correctly identifying negative reviews.

Table 4.4: Performance Metric Statistics of 10-fold Cross Validation of ELSA

S.No	Precision	Specificity	Sensitivity	Accuracy	F-measure	False Alarming	MCC
1	0.873	0.878	0.841	0.86	0.876	0.14	0.72
2	0.854	0.855	0.848	0.851	0.854	0.149	0.703
3	0.848	0.851	0.83	0.841	0.85	0.159	0.681
4	0.868	0.875	0.827	0.851	0.872	0.149	0.703
5	0.874	0.879	0.838	0.859	0.877	0.141	0.718
6	0.855	0.855	0.852	0.854	0.855	0.147	0.707
7	0.873	0.877	0.85	0.863	0.875	0.137	0.727
8	0.852	0.851	0.857	0.854	0.851	0.146	0.707
9	0.872	0.875	0.858	0.866	0.874	0.134	0.732
10	0.863	0.866	0.838	0.852	0.865	0.148	0.705

The range of the model's sensitivity, or recall, which quantifies the percentage of real positive instances that were correctly classified, was 0.8757 to 0.9093. The range of sensitivity values suggests that the model's capacity to recognize positive instances across various folds varies somewhat. The percentage of instances that were correctly classified using the model, or the accuracy, ranged from 0.8958 to 0.9140. The model generally kept a good balance between precision and recall across all folds, as shown by the F1-measure, which is the harmonic mean of precision and recall, which ranged from 0.9098 to 0.9184.

The proportion of negative instances that were mistakenly classified as positive, or the false alarming rate, ranged from 0.0860 to 0.1042, indicating a comparatively low rate of false-positive classifications. The range, however, indicates that there may have been some variation in this rate among the various folds.

The MCC, a metric for the accuracy of binary classifications, varied between 0.7922 and 0.8280. Perfect prediction is represented by an MCC of +1, average random prediction by 0, and inverse prediction by -1. The range of MCC values found indicates that the SvmBagging model showed good performance in all ten folds of the cross-validation, but there was some variation in the accuracy of the binary classifications across the various folds.

According to the performance analysis, the modern SvmBagging model performed admirably across all metrics and folds. The model's performance did vary slightly across folds, though, particularly in terms of sensitivity, false alarming rate, and MCC. This implies that although the model performed admirably overall, there may be room for improvement in terms of consistency across various data subsets.

Table 4.4 additionally, the performance of the ELSA model was evaluated using a 10-fold cross-validation strategy, which looked at a number of metrics including precision, specificity, sensitivity, accuracy, F1-measure, false alarming rate, and Matthews correlation coefficient (MCC).

The ELSA's precision ranged from 0.8493 to 0.8715, showing a reasonably good model's ability to classify positive instances as positive. Specificity, or the model's capacity to correctly classify negative instances, ranged from 0.8517 to 0.8778, indicating that the model did a good job of spotting negative reviews.

Sensitivity, or recall, the proportion of actual positive instances that were correctly classified by the model, ranged from 0.8226 to 0.8587. The accuracy of the model, the proportion of all instances that were correctly classified, ranged from 0.8422 to 0.8596. The harmonic mean of precision and recall, or the F1-measure, ranged from 0.8505 to 0.8746, showing that precision and recall were well-balanced across all folds.

There was some variation across the various folds, but the false alarming rate—the percentage of negative instances that were mistakenly classified as positive—ran from 0.1404 to 0.1578, indicating a relatively low rate of false-positive classifications.

The quality of binary classifications was gauged by the MCC, which had a range of 0.6846 to 0.7191. Although there was some variation in the performance of the binary classifications across the different quality folds, the range of MCC values obtained suggests that the ELSA model demonstrated good performance in all ten folds of the cross-validation.

According to the performance analysis, the ELSA model demonstrated a high rate across all metrics and folds. The model's performance did vary slightly across folds, though, particularly in terms of sensitivity, false

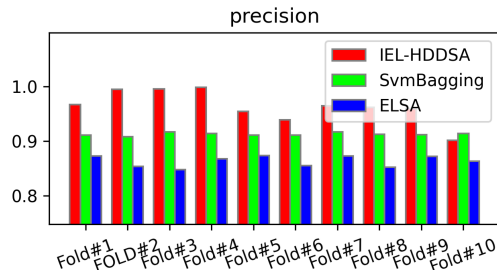


Fig. 4.1: The graph representation of the precision proposed IEL-HDDSA compared SVMBagging and ELSA

alarming rate, and MCC. This implies that although the model performed admirably overall, there may be room for improvement in terms of consistency across various data subsets.

4.2. Comparative Study.

4.2.1. Precision. The Eq 4.1 proportion of true positive instances among all instances that the model classified as positive is a measure of precision. A lower rate of false-positive classifications is indicated by a higher precision Figure 4.1.

$$Precision = (TP / (TP + FP)) \quad (4.1)$$

With an average of 0.96397 and a relatively higher standard deviation of 0.02778, the IEL-HDDSA model showed the highest average precision across all ten folds. This shows that, despite some variation in its performance across the various folds, the IEL-HDDSA model generally had the highest rate of correctly classifying positive instances as positive. In terms of IEL-HDDSA precision, Fold#4 had the highest value (0.9986) and Fold#10 had the lowest value (0.902).

With an average precision of 0.91281 and a very low standard deviation of 0.00276, the SvmBagging model had the second-highest average precision and demonstrated greater consistency across the various folds. Fold#3 (0.9173) and Fold#2 (0.9079) had the highest and lowest SvmBagging precision, respectively.

The ELSA model had the lowest standard deviation of 0.00980 and the lowest average precision of the three models at 0.86325 on average. This shows that the ELSA model performed fairly consistently across the various folds and had the lowest rate of correctly classifying positive instances as positive. Fold#1 (0.8734) and Fold#3 (0.8482) showed the highest and lowest ELSA precision, respectively.

4.2.2. Specificity. The Eq 4.2 proportion of true negative instances among all instances that the model classified as negative is a measure of specificity. A lower rate of classifications that are falsely negative corresponds to higher specificity Figure 4.2.

$$Specificity = (TN / (FP + TN)) \quad (4.2)$$

With an average specificity of 0.93941 and a relatively small standard deviation of 0.00450, the IEL-HDDSA model demonstrated the highest average specificity across all ten folds. This shows that the IEL-HDDSA model performed relatively consistently across the various folds and had the highest overall rate of correctly classifying negative instances as negative. Fold#10 had the highest specificity for IEL-HDDSA and Fold#3 had the lowest (both 0.94507 and 0.93229).

The SvmBagging model performed consistently across all folds, as evidenced by its second-highest average specificity of 0.91544 and extremely low standard deviation of 0.00265. Fold#7 (0.91925) and Fold#2 (0.91110) had the highest and lowest specificity for SvmBagging, respectively.

With an average specificity of 0.86620 and a relatively higher standard deviation of 0.01129, the ELSA model had the lowest average specificity of the three models. This shows that the ELSA model performed relatively worse across the different folds and had the lowest rate of correctly classifying negative instances

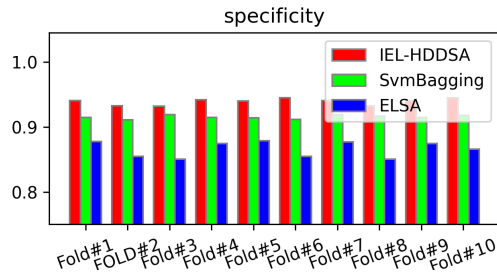


Fig. 4.2: The graph representation of the specificity proposed IEL-HDDSA compared SVMBagging and ELSA

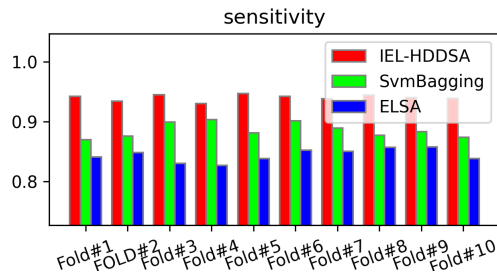


Fig. 4.3: The graph representation of the Sensitivity proposed IEL-HDDSA compared SVMBagging and ELSA

as negative. Fold#5 (0.87945) and Fold#3 (0.85147) showed the highest and lowest specificity for ELSA, respectively.

4.2.3. Sensitivity. Sensitivity Eq 4.3, also referred to as the true positive rate, measures the percentage of true positives among all instances that are actually positive. A lower rate of false-negative classifications is indicated by sensitivity that is higher Figure 4.3.

$$Sensitivity = (TP / (TP + FN)) \quad (4.3)$$

The IEL-HDDSA model demonstrated the highest average sensitivity across all ten folds, with an average of 0.9401 and a reasonably small standard deviation of 0.00489. This suggests that the IEL-HDDSA model performed relatively consistently across the various folds and had the highest overall rate of correctly classifying positive instances as positive. Fold#5 had the highest IEL-HDDSA sensitivity (0.9468), while Fold#4 had the lowest (0.931).

With an average of 0.8852 and a relatively higher standard deviation of 0.01135, the SvmBagging model had the second-highest average sensitivity and showed less consistent performance across the various folds. Fold#4 (0.9029) and Fold#1 (0.8704) showed the highest and lowest sensitivity for SvmBagging, respectively.

With an average of 0.8437 and a standard deviation of 0.0101, the ELSA model had the lowest average sensitivity of the three, indicating slightly less consistent performance across the various folds. The folds with the highest and lowest ELSA sensitivity were Fold#9 (0.8575) and Fold#4 (0.8270), respectively.

4.2.4. Accuracy. The Figure 4.4 ratio of correct predictions—both true positives and true negatives—to total predictions is known as accuracy. It serves as a measure of a classification model's overall correctness

$$Accuracy = ((TP + TN) / (TP + TN + FP + FN)) \quad (4.4)$$

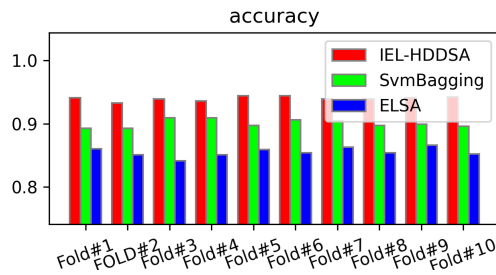


Fig. 4.4: The graph representation of the accuracy proposed IEL-HDDSA compared SVMBagging and ELSA

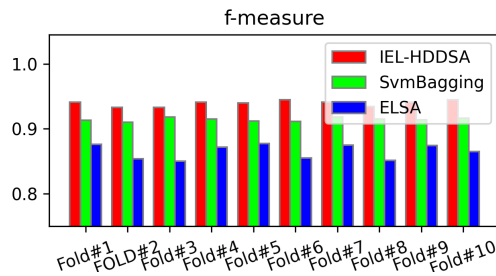


Fig. 4.5: The graph representation of the F-measure proposed IEL-HDDSA compared SVMBagging and ELSA

With an average of 0.9398 and a standard deviation of 0.0031, the IEL-HDDSA model had the highest average accuracy across all ten folds, indicating that it performed the most consistently. Fold#5 had the highest accuracy for IEL-HDDSA (0.9435), while Fold#2 had the lowest (0.9334).

With an average of 0.9003 and a higher standard deviation of 0.0060, the SvmBagging model had the second-highest average accuracy and showed less consistent performance across the various folds. SvmBagging accuracy was found to be highest in Fold#4 (0.9091) and lowest in Fold#10 (0.8961).

The ELSA model performed the least consistently across the various folds, having the lowest average accuracy of the three models with an average of 0.8550 and a standard deviation of 0.0069. Fold#9 had the highest accuracy for ELSA (0.8661), while Fold#3 had the lowest (0.8406).

4.2.5. F-Measure. The Figure 4.5 harmonic mean of precision and sensitivity is the F-measure, also referred to as the F1-score. It is an effective way to condense information about the model's recall and accuracy into a single value.

With an average F-measure of 0.9395 and a small standard deviation of 0.0043, the IEL-HDDSA model performed consistently across all ten folds and had the highest average F-measure across all ten folds. Fold#6 had the highest F-measure for IEL-HDDSA and Fold#3 had the lowest (0.9327).

With an average F-measure of 0.9141 and a minimal standard deviation of 0.0027, the SvmBagging model had the second-highest average F-measure, indicating consistent performance but not as high as IEL-HDDSA. Fold#10 had the highest F-measure for SvmBagging (0.9161), while Fold#6 had the lowest (0.9114).

With an average F-measure of 0.8647 and a relatively higher standard deviation of 0.0105, the ELSA model had the lowest average F-measure of the three, indicating less consistent performance across the various folds. The Fold#5 fold had the highest F-measure for ELSA (0.8768) and the Fold#3 fold had the lowest (0.8498).

4.2.6. False Alarming. The Figure 4.6 ratio of negative events wrongly classified as positive to the total number of negative events is known as the false alarm rate or false positive rate. In order to prevent irrational expenses or actions brought on by false positives, it is crucial to keep this rate as low as possible.

With an average false alarm rate of 0.0602 and a small standard deviation of 0.0031, the IEL-HDDSA model performed consistently across all ten folds and had the lowest average false alarm rate overall. The folds

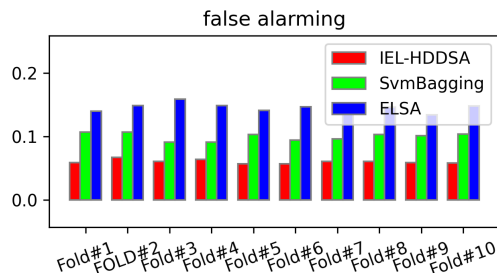


Fig. 4.6: The graph representation of the false alarm rate y proposed IEL-HDDSA compared SVMBagging and ELSA

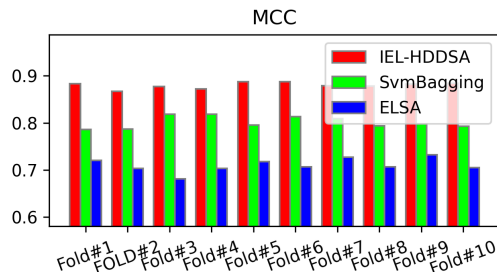


Fig. 4.7: The graph representation of the MCC proposed IEL-HDDSA compared SVMBagging and ELSA

with the highest and lowest false alarm rates for IEL-HDDSA were 2 and 5, respectively (0.0666 and 0.0565).

With an average false alarm rate of 0.0997 and a slightly higher standard deviation of 0.0060, the SvmBagging model had the second-lowest average false alarm rate and less consistent performance across the various folds. Fold#10 had the highest false alarm rate for SvmBagging (0.1039), while Fold#3 had the lowest (0.0910).

The ELSA model displayed the least consistent performance across the various folds, having the highest average false alarm rate of the three models with an average of 0.1450 and a standard deviation of 0.0069. Fold#3 had the highest false alarm rate for ELSA (0.1594), while Fold#9 had the lowest (0.1339).

4.2.7. MCC. Figure 4.7 a measure of the accuracy of binary classifications is the Matthews Correlation Coefficient (MCC). In general, it is regarded as a balanced measure that can be applied even when the classes have very different sizes because it considers both true and false positives and negatives. A value between -1 and +1 is the result of the MCC. Perfect predication is represented by a coefficient of +1, average random predication by 0, and inverse predication by -1. With an average MCC of 0.8795 and a minimal standard deviation of 0.0061, the IEL-HDDSA model performed consistently across all ten folds and had the highest average MCC. The folds with the highest and lowest MCCs for IEL-HDDSA were #5 (0.8869) and #2 (0.8669), respectively.

With an average MCC of 0.8011 and a slightly higher standard deviation of 0.0117, the SvmBagging model had the second-highest average MCC and showed less consistent performance across the various folds. Fold#4 had the highest MCC for SvmBagging (0.8183), while Fold#10 had the lowest (0.7929).

With an average MCC of 0.7103 and a standard deviation of 0.0138, the ELSA model had the lowest average MCC of the three, indicating the least consistent performance across the various folds. Fold#9 had the highest MCC for ELSA (0.7322) and Fold#3 had the lowest (0.6814).

5. Conclusion. In comparison to other models, SvmBagging and ELSA, the Iterative Ensemble Learning over High Dimensional Data for Sentiment Analysis (IEL-HDDSA) model proposed in this study demonstrated superior performance in sentiment analysis of high-dimensional data. The IEL-HDDSA model consistently outperformed the other models in a thorough evaluation using 10-fold cross-validation across numerous metrics,

demonstrating its accuracy and robustness. The study's focus on text data, exclusion of real-time data, and lack of comparisons with other deep learning or machine learning models, however, are its main drawbacks. Despite these limitations, the study makes a significant contribution by providing a novel, robust, and accurate approach for sentiment analysis of high-dimensional data. Future research should aim to validate the model in real-world applications, explore its applicability to other types of data, and compare its performance with other machine learning and deep learning models.

REFERENCES

- [1] G. Beigi, X. Hu, R. Maciejewski, and H. Liu, "An overview of sentiment analysis in social media and its applications in disaster relief," in *Sentiment Analysis and Ontology Engineering*, pp. 313–340, 2016.
- [2] B. Liu, *Sentiment Analysis and Opinion Mining*, Springer Nature, 2022.
- [3] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, 2020.
- [4] I. M. Johnstone and D. M. Titterton, "Statistical challenges of high-dimensional data," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 367, no. 1906, pp. 4237–4253, 2009.
- [5] R. Sulthana, A. K. Jaithunbi, H. Harikrishnan, and V. Varadarajan, "Sentiment analysis on movie reviews dataset using support vector machines and ensemble learning," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 17, no. 1, pp. 1–23, 2022.
- [6] C. Kumaresan and P. Thangaraju, "ELSA: Ensemble learning based sentiment analysis for diversified text," *Measurement: Sensors*, vol. 25, 100663, 2023.
- [7] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," *Decision Support Systems*, vol. 57, pp. 77–93, 2014.
- [8] K. Korovkinas, P. Danėnas, and G. Garšva, "SVM and Naïve Bayes classification ensemble method for sentiment analysis," *Baltic journal of modern computing*, vol. 5, no. 4, pp. 398–409, 2017.
- [9] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Systems with Applications*, vol. 77, pp. 236–246, 2017.
- [10] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.
- [11] U. Kumari, A. K. Sharma, and D. Soni, "Sentiment analysis of smart phone product review using SVM classification technique," in *2017 International conference on energy, communication, data analytics and soft computing (ICECDS)*, pp. 1469–1474, IEEE, 2017.
- [12] M. Ahmad, S. Aftab, and I. Ali, "Sentiment analysis of tweets using svm," *Int. J. Comput. Appl.*, vol. 177, no. 5, pp. 25–29, 2017.
- [13] P. Thiengburanathum, and P. Charoenkwan, "SETAR: Stacking Ensemble Learning for Thai Sentiment Analysis using RoBERTa and Hybrid Feature Representation," *IEEE Access*, 2023.
- [14] M. Ahmad, S. Aftab, M. S. Bashir, N. Hameed, I. Ali, and Z. Nawaz, "SVM optimization for sentiment analysis," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 4, 2018.
- [15] M. Khalid, I. Ashraf, A. Mehmood, S. Ullah, M. Ahmad, and G. S. Choi, "GBSVM: sentiment classification from unstructured reviews using ensemble classifier," *Applied Sciences*, vol. 10, no. 8, 2788, 2020.
- [16] J. Kazmaier, and J. H. Van Vuuren, "The power of ensemble learning in sentiment analysis," *Expert Systems with Applications*, vol. 187, 115819, 2022.
- [17] Q. Ismail, R. Obeidat, K. Alissa, and E. Al-Sobh, "Sentiment analysis of covid-19 vaccination responses from twitter using ensemble learning," in *2022 13th International Conference on Information and Communication Systems (ICICS)*, pp. 321–327, IEEE, 2022.
- [18] S. Shah, H. Ghomeshi, E. Vakaj, E. Cooper, and R. Mohammad, "An Ensemble-Learning-Based Technique for Bimodal Sentiment Analysis," *Big Data and Cognitive Computing*, vol. 7, no. 2, 85, 2023.
- [19] H. Rahman, J. Tariq, M. A. Masood, A. F. Subahi, O. I. Khalaf, and Y. Alotaibi, "Multi-tier sentiment analysis of social media text using supervised machine learning," *Comput. Mater. Contin.*, vol. 74, pp. 5527–5543, 2023.
- [20] M. K. Iqbal, K. Abid, S. u. Ayubi, and N. Aslam, "Omicron Tweet Sentiment Analysis Using Ensemble Learning," *Journal of Computing & Biomedical Informatics*, vol. 4, no. 02, pp. 160–171, 2023.
- [21] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve Bayes," *Encyclopedia of machine learning*, vol. 15, no. 1, pp. 713–714, 2010.
- [22] N. Altman and M. Krzywinski, "Ensemble methods: bagging and random forests," *Nature Methods*, vol. 14, no. 10, pp. 933–935, 2017.
- [23] K. W. Church, "Word2Vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.
- [25] T. Hesterberg, "Bootstrap," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 3, no. 6, pp. 497–526, 2011.
- [26] L. Rokach and O. Maimon, "Decision trees," in *Data mining and knowledge discovery handbook*, pp. 165–192, 2005.
- [27] L. A. Snider and S. E. Swedo, "PANDAS: current status and directions for research," *Molecular psychiatry*, vol. 9, no. 10, pp. 900–907, 2004.

- [28] C. R. Harris et al., “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [29] O. Kramer, “Scikit-learn,” in *Machine learning for evolution strategies*, pp. 45–53, 2016.
- [30] S. Bird, “NLTK: the natural language toolkit,” in *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pp. 69–72, 2006.
- [31] E. Bisong and E. Bisong, “Matplotlib and seaborn,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 151–165, 2019.

Edited by: Anil Kumar Budati

Special issue on: Soft Computing and Artificial Intelligence for wire/wireless Human-Machine Interface

Received: Sep 29, 2023

Accepted: Jan 11, 2024