SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I1P106

Volume 11 Issue 1, 67-85 January 2024

Original Article

Sentiment Analysis Using Self-Adaptive Stacking Ensemble Method for Classification

K.R. Srinath[1], B. Indira[2]

[1]*Department of Informatics, Osmania University, Telangana, India*
[2]*Department of Master of Computer Applications, CBIT, Telangana, India.*

[1]*Corresponding Author : srinath.kr1022@gmail.com*

Received: 16 October 2023 Revised: 26 November 2023 Accepted: 17 December 2023 Published: 13 January 2024

Abstract - *The primary purpose of sentiment analysis is to classify the polarity of the data, such as whether the data should be positive, negative, or neutral. Most sentiment analyses used single classifiers, but they do not provide an accurate polarity. There should also be drawbacks, like a lack of keywords, high dimensional space, etc. This paper used the polarized word embedding technique and Remora Optimization algorithm for distance ranking; then, the classification is done by both machine learning and deep learning classifiers that are integrated using the self-adaptive stacking ensemble method to select the finest base classifier and hyper-parameters of base classifiers with the use of the genetic algorithm. Then, the model is trained and tested employing four datasets utilizing cross-validation, and the performance is calculated using recall, accuracy, precision, F1 score, and AUC that is compared using four state-of-the-art models. The comparison shows that the proposed method provides the most accurate predicted value with the highest accuracy of 99.3%.*

Keywords - *Sentiment analysis, Word embedding, Attention CNN, Bi-GRU, HSVM, Bayesian network.*

1. Introduction

The evaluation of technology provides a new type of communication through user-generated content like social media, e-commerce sites, etc.; using this advanced technology, people can express their opinions about a particular subject. This development leads to a vast amount of data on the internet. From this, getting the necessary information about a specific topic is hard to extract, so sentiment analysis has recently become a hot research field in natural language processing.

Sentiment analysis analyzes and extracts information about a particular subject online. The main aim is to classify the sentiments and opinions in the text the people created. The machine learning method uses supervised learning for training with the labeled dataset in a classification model; some supervised learners are Naïve Bayes, SVM, K-NN algorithm, and RF. It also includes deep learning methods like CNN, RNN, LSTM, GRU, etc.; these deep learning approaches use a distributed representation method for large datasets. It extracts the features from the data to analyze many different problems with the best accuracy and efficiency in prediction, and it also reduces the prediction time. In sentiment analysis, there are three levels of process: sentence, document, and feature level [1]. This paper is based on the feature level, which uses the aspects of entities to classify opinions. Since this level uses the features of the sentences in-depth, it extracts the expressions hidden in the large text. So, it provides a supervised learning method on labeled data. In sentiment analysis, word embedding is a crucial step in the classification and feature selection.

The bag of word method is highly recommended for the text classification method. Word embedding will represent the text document in a dense space with a fixed length to improve the performance, and sparsity will also be reduced using the low dimension in a bag of words. Word2vec is a convolutional scheme that computes the mean of word embedding and improves the performance of supervised and unsupervised learning in the Natural Learning Process (NLP) [2].

Deep Learning is used in sentiment analysis for decision-making, classification, and recommendation problems. It is a powerful computational model that will find the complicated semantic illustration of text by itself [3]. Using the different types of classifiers with machine learning algorithms to train the model is more efficient than the individual classifier [4]. Then, ensemble learning is the finest technique for improving machine learning models, which involves combining multiple models to improve the accuracy and robustness of classifications. Three broad categories can categorize ensemble learning: boosting, bagging, and stacking.

Moreover, the various machine learning issues, such as clustering, regression, and classification, have been tackled by utilizing ensemble learning. Finding the ideal base classifier combination and parameter choices for classical ensemble learning can be challenging. Regarding the Boosting techniques, they exhibit sensitivity to anomalies that arise from the weak classifier. All of the prediction functions for the Bagging algorithm are directly related to weights and have the potential to result in significant inaccuracies. Furthermore, an adaptable optimization technique that works well for stacking is currently lacking. Stacking performance is heavily influenced by base classifiers' input qualities and learning procedures [5]. Stacking is an advanced technique that combines several base models' predictions by training a meta-model. Stacking has been successfully used in both supervised and unsupervised classification fields.

In this article, the merits of ML and DL are concatenated by the ensemble method to get the best classification of opinions from the user. Before the classification process, the word embedding method is taken using the polarised embedding model, which is used to develop the embedding space after the pre-processing of input words. The feature selection addresses the optimization problem's limitation; this paper uses Mutual Information (MI) and the Remora Optimization Algorithm (ROA). Then, different classifiers from deep learning and machine learning are used to find the sentiments from the text.

Bi-directional Gated Recurrent Unit (Bi-GRU) and Attention Convolutional Neural Network (CNN) are the deep learning classifiers, the supervised machine learning method used for classifications are Heterogeneous Support Vector Machine (HSVM) and Bayesian network and the self-adaptive stacking method as an ensemble method to get the best prediction. These are the approaches involved in the proposed sentiment analysis. The main contribution of the recommended sentiment analysis model is summarized below.

- The word embedding is performed using the Polarised word embedding technique, which polarizes the words into positive and negative without overlapping and sparsity problems.
- The features are selected using the bi-stage feature selection method. The two stages are Mutual Information (MI) and the Remora Optimization Algorithm (ROA). MI used to find the strong correlation between the variable and the distance ranking is done by the ROA.
- The classification is done by the Attention CNN and Bi-GRU deep learning classifier, which is used to develop text dependencies and detect the important features of the given data.
- Finding the relation between the most significant texts and learning the new data with new words by itself is done by a Bayesian network classifier. A Heterogeneous Support Vector Machine classifier deals with heterogeneous and imbalanced data.
- Self-adaptive stacking ensemble learning is employed to enhance the performance of the model. The main aim of this ensemble model is to convert the weak classifiers into robust classifiers. In this research, four weak classifiers, such as Bayesian network, Attention CNN, HSVM, and Bi-GRU, are combined in the ensemble method by utilizing the self-adaptive stacking technique to provide a strong classifier, which gives better outcomes.
- In the stacking approach, k-fold cross-validation mainly aids in preventing the model from becoming overfit to the training set. Moreover, k-fold cross-validation yields a more accurate evaluation of the model's efficacy on fresh data since it uses distinct data sections for testing and training. As a result, the model's generalization performance is enhanced.
- The classifiers are trained and tested using four datasets. The performance is measured using Precision, Accuracy, Recall, AUC and F1-Score. Then, the evaluation criteria values are compared with the four current methods in sentiment analysis.

This article is detailed below; Section 2 explains the literature review. Section 3 describes the proposed method of the paper, Section 4 presents the pre-processing, word embedding, feature selection, and sentiment classification methods. Section 5 explains the experimental results of the suggested model, performance evaluation utilizing the dataset, and comparison with the existing method. Section 6 provides the conclusion of the proposed experiment.

2. Related Works

Using sentiment analysis, a lot of research has been conducted in recent years; some of the study is briefly explained in this section. Onan, AytuÄ (2020) [6] proposed a sentiment analysis using a leveraging word embedding approach, which is word2vec with the hyper-parameters for better performance of word embedding.

The supervised machine learning algorithm Random Forest Algorithm trains the model. Without using the feature selection method, it gives an accuracy of 75% in the negative class, 70% in the positive class, and 62% in the neutral class. Rezaeinia, Seyed Mahdi, et al. (2019) [7] Proposed the Improved word2vec in sentiment analysis to overcome problems in word2vec. Improved word2vec is used to increase the accuracy of the pre-trained vector. This sentiment analysis model uses the lexicon-based approach to train the model. It was evaluated using five different datasets, giving it the best performance.

Zhang et al. [8] proposed three-word embedding methods: sentiment, semantic, and lexicon. The multi-modality classification with a fusion of CNN and LSTM using the Attention mechanism and the Cross-modality regression is applied for feature extraction. This proposed model is evaluated using two different datasets. Then, the result from the implementation is compared with the existing methods, revealing that the suggested technique performs better than others.

Usama, Mohd, et al. [9] used a distributed DL method to learn the word embedding and implement it through the word2vec model. The classifiers used in this paper are CNN and RNN, with the attention mechanism. The model is tested utilizing the three datasets, and efficiency is compared using four previous attention models. This experiment reveals the accuracy with three datasets, which are 83.64%, 51.14%, and 89.62%.

Pan, Yaxing, Liang, and Mingfeng [10] proposed a sentiment analysis model to reduce the high complexity and improve the efficiency of sentiment analysis. They used BiGRU and attention mechanism for the classification with the pre-trained word embedding. They introduced the multi-head self-attention mechanism to reduce the external parameters, assign a weight to the word vector, and highlight the text feature. The experimental results give an accuracy of 87.1%.

Huawen Liu et al. [11] proposed two primary sentiment analyses using the ensemble models M_{SG}, M_{GA}, and M_{SGA}. After using the deep learning model as the baseline, the word embedding method is employed for the feature extraction for developing the sentiment analysis model. The result of this sentiment analysis model is the comparison of classification accuracies of classifiers using 16 datasets with four feature selection algorithms. As a result, the proposed method outperforms well in all measurements.

Basiri, Mohammad Ehsan, et al. [12] presented a sentiment analysis using the fusion method with deep learning and a machine learning classifier called 3-way fusion of one deep learning with a convolutional technique. It was implemented using the drug review dataset. The precision score is 0.886, the F1 score is 0.8836, Recall is 0.883, and the model's accuracy is 88.3%.

Araque, Oscar, et al. [13] use an ensemble technique to enhance deep learning sentiment analysis. They developed two ensemble techniques for collecting the baseline classifier with the other surface classifier in sentiment analysis. They used two models that combined the baseline and deep features to get the information from many sources. The result of the suggested technique is implemented by employing seven datasets. The performance is compared with the existing fusion method, and the F1 score of the model gives a high performance using an IMDB movie review. Traditional ensemble learning faces challenges in achieving optimal parameter settings and base-classifier combinations while Boosting approaches are sensitive to weak classifier anomalies. Gaye B et al. (2021) [14] suggested a novel strategy that uses deep learning classifiers, machine learning, and linguistic dictionaries. This research utilized a stacked ensemble for three LSTM classifiers as base classifiers and Logistic Regression (LR) as a meta-classifier to classify the tweets according to TextBlob's retrieved sentiments. Since the suggested model does not need feature extraction because LSTM extracts features automatically, it has proven efficient and time-saving. Word embedding techniques are proposed to analyze sentiments in textual documents like social media posts and online product reviews, but capturing intricate inter-dependencies is challenging.

Gormezi et al. [15] proposed a feature-based stacked ensemble method that systematically integrates six FE techniques and triple classifiers. The techniques used for feature extraction are as follows: unigram TF-IDF, hierarchical softmax skip-gram, unigram TF negative sampling continuous bag of words, and negative sampling endless bag of words. The classifiers used are logistic regression and multi-layer perceptron in the 1st stage and support vector machine in the 2nd stage.

To tackle this problem, Subba, B., & Kumari, S. (2021) [16] presented a computationally effective sentiment analysis method based on stacking ensembles and multiple-word embedding. In Mohammadi, A. & Shaverizade, A. (2021) [17] addresses the problem of Aspect-Based Sentiment Analysis (ABSA), which aims to extract the opinions or attitudes towards specific topics or entities in a text. The adoption of deep learning techniques for ABSA is encouraged by this article since they have demonstrated better results in tasks involving natural language processing.

Zhou, Yanling, et al. [18] proposed a fusion deep learning approach for hate speech detection. They used machine learning, deep learning, and BERT for text classification. These classifiers are fused using a classifier fusion method. The result shows that the value of the F1 score of the classifications is improved.

Teragawa Shoryu et al. [19] proposed a sentiment analysis using two deep learning techniques, CNN and LSTM. CNN is used to extract the features by optimizing the network, and LSTM is used to remove the consecutive data from the text. For classification purposes, they combined both CNN and LSTM. The experiment using the commodity review of e-commerce reveals that they deliver a good result compared with the individual CNN and LSTM. Dohaiha, Hai Ha et al. [20] proposed the sentiment polarized word embedding model for multi-label sentiment classification to find the critical emotional semantic words and a Relax loss function to optimize the objective function. The experimental results expose that it can decrease the approximate degree and overfitting of the model to the target label by using the multi-labeled comments dataset.

Zhang, Dejun, et al. [21] combined the merits of CNN and Bi-GRU for sentiment expression classification, which extracts the features from the words and contexts using a pre-trained vector. The experiment is validated using four datasets, and the result is measured using the performance metrics. The highest accuracy is 95.1%.

Harleen Kaur et al. [22] proposed the Hybrid Heterogeneous Support Vector Machine to classify the sentiment classification. The classification was done using the deep learning and machine learning algorithms RNN and SVM, which rank the Covid-19 dataset as positive, negative, and neutral.

3. Proposed Methodology

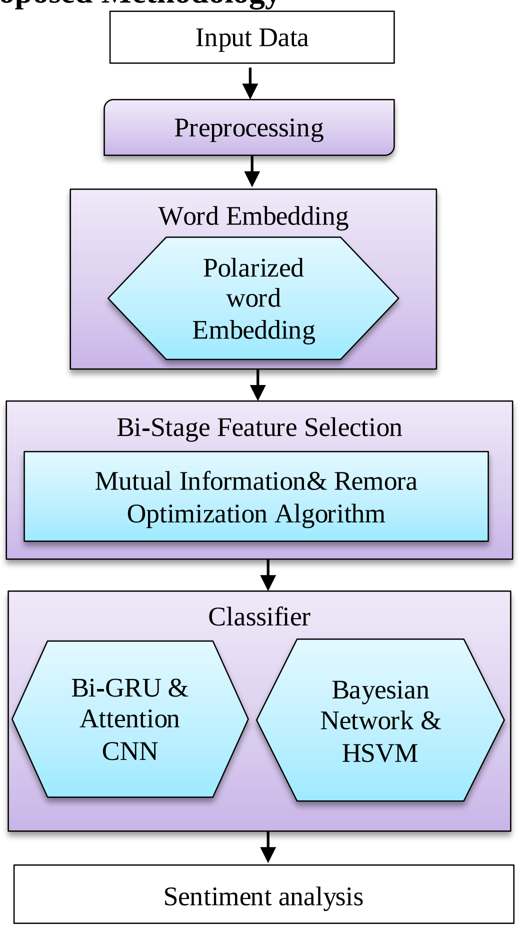

Fig. 1 Proposed methodology

The steps involved in the proposed method are listed in the flow chart shown in Figure 1. The initial step is to collect data from the internet platform. The crawled dataset is pre-processed by removing the unwanted data from the whole dataset, and the information in the text is mined using the polarized word embedding process [23]. The mined data are exposed for the feature selection method to reduce the dimensionality and unnecessary data from the reviews by using the Mutual Information algorithm [24-26], and the distance ranking is done by the ROA [27].

Using these processed datasets, the model was trained using the fusion of deep learning and machine learning classifier. The deep learning classifiers are Attention CNN and BiGRU, and the machine learning classifiers are HSVM [20] and Bayesian networks [28]. Here, four datasets are used for both training and testing the model. Then, the ensemble method combines the four classifiers to give an accurate prediction.

3.1. Pre-Processing

The pre-processing techniques involved in the sentiment analysis are Tokenization, Lemmatization, Removal of stop words, Removal of hashtags, Removal of non-alphabetic words, and stemming, shown in Figure 2. The input data should be converted from unnecessary upper case letters to lower case letters, and then the sentences are split into words that are meant as tokens.

Lemmatization is combining the tokenized word into the proper sentences, which means it will change the word to its root word. Most of the dataset is from social media, so it contains many hashtag symbols (#) and emoticons, so it should be removed from the dataset. Then, the stop words like a, as, was, etc., and non-alphabetic words like numbers symbols, emoticons, etc., are removed from the sentences, which are redundant for the further process.

Stemming removes the prefixes and suffixes of the words, like lemmatization, because it also converts the word to its root word. These are the processes used in the pre-processing techniques; these steps are taken to remove noise from the input data, improving the proposed model's accuracy.

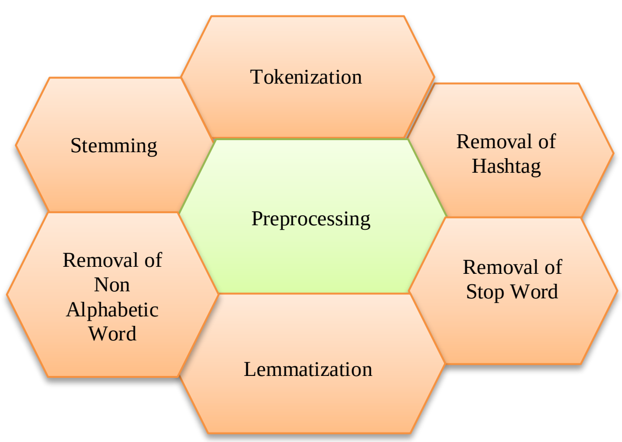

Fig. 2 Pre-processing

3.2. Polarized Word Embedding

Word embedding is the text mining technique from sentiment analysis used to build the vector representation of the words in the dense space by the fixed length vector to develop the performance. It also represents the text with a low dimension, ignoring the sparsity and high dimensionality problems. This proposed paper used a polarized word embedding technique that polarizes the word embedding into positive and negative words extracted from the SentiWords.

Normal sentences are expected to have positive and negative words in similar contexts. If a positive word is changed from a sentence to a negative word, then the polarity of the sentence will be reversed. In the vector space, the positive and negative words are represented by close word embedding, which causes the overlapping.

The polarized word embedding avoids overlapping positive and negative words in similar contexts. In the sentiment polarity classification, the pre-trained embedding projects use the new space for each word's polarity and take the details of each word. This permits the development of embedding space for the polarity classification, which is mainly associated with positive or negative opinions.

It involves k-means clustering for clustering positive and negative words; they used two clusters.c_1 and c_2 with random centroids. The accuracy of the clustering is calculated in Equation (1)

$$accuracy = \frac{\max\{(c_1^+ + c_2^-),(c_1^- + c_2^+)\}}{c_1^+ + c_1^- + c_2^+ + c_2^-} \qquad (1)$$

Where c_1^+, c_1^- and c_2^+, c_2^- are the numbers of the negative and positive words in the cluster c_1 and c_2. The experiment gives the best accuracy of using k-means clustering in the separation of positive and negative words.

3.3. Feature Selection

The FS method selects the relevant features from the dataset under the classification. It eliminates the redundant and irrelevant features from the dataset, improving the classifier's overall performance and providing the best accuracy. The proposed paper used the bi-stage feature selection method. The two stages are MI and ROA, which are used to select the most relevant features from the dataset to train the deep learning and machine learning algorithms to enhance the classification.

3.3.1. Mutual Information

Mutual information is an FS technique used to find the strong correlation between the variables, improving classification performance and filtering the feature details from the selected criteria with the class labels. The properties of mutual information are under the transformation, and it is invariant in feature space.

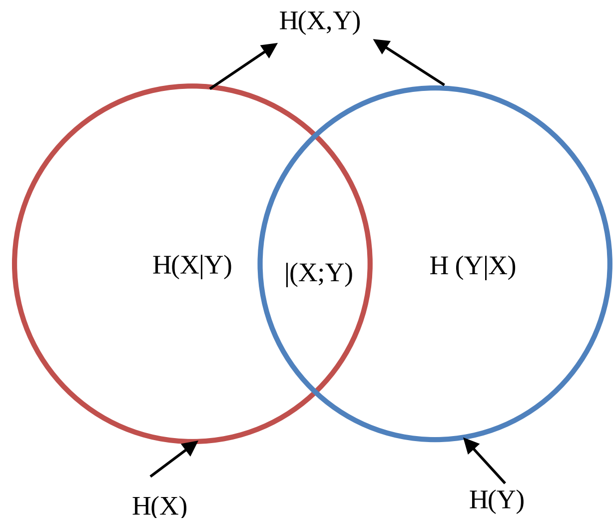

Fig. 3 Correlation of MI function

Let X be the random variables with discrete values; the entropy is measured using H(X) and the joint entropy *H(X, Y)*, which is expressed in Equation (2),

$$H(X,Y) = -\sum_{y \in Y} \sum_{x \in X} p(x,y) \log p(x,y) \qquad (2)$$

Where the density function is represented as (x, y), $H(X, Y)$ is an entropy function that depends on the random variables of a probability distribution.

The reduction of the improbability of the variable is done by controlled entropy. The controlled entropy of X and Y is determined as $H(X|Y)$ with Y. Y is a variable expressed in Equation (3),

$$H(X|Y) = -\sum_{y \in Y} \sum_{x \in X} p(x,y) \log p(x|y) \qquad (3)$$

If X depends on Y, then $H(X|Y)$ is zero. The information transformed from the variables X and Y are measured using *I(X; Y)*, defined as Mutual Information derived in Equations (4 & 5).

$$I(X;Y) = H(y) - H(X|Y) \qquad (4)$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)P(y)} \qquad (5)$$

Here, the value of $I(X;Y)$ is high and $I(X;Y) = H(y) - H(X|Y)$.

The first stage is MI, which calculates the amount of information in which the random variables are about the other variable. The MI between two variables X and Y are derived in Equation (6).

$$H(X;Y) = \int_X \int_Y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \qquad (6)$$

Where joint probability is *p(x, y)* with the density function of *x* and *y*. p(x) and p(y) is described as marginal function. Figure 3 shows the correlation of the MI function in the form of a Venn diagram.

3.3.2. Remora Optimization Algorithm

The second feature selection stage is distance Ranking using the Remora Optimization Algorithm optimization algorithm. It is a meta-heuristic algorithm inspired by the species remora, which changes the host according to the location. It uses the whale optimization algorithm and swordfish algorithm as examples to stimulate the living habits of parasitic feeding on various hosts. The hosts are whales and swordfish, with the best movement characteristics to change the different modes. The ROA is further proposed to adjust the local renewal. The idea used in this algorithm is that when the remora attach themselves to the swordfish, it will change its position at that time.

It is supposed that the solution is remora and the variable is position R. The position vector will change according to the swimming dimension of the fish. The current position is mentioned as $R_n = \{R_{n1}, R_{n2} \dots \dots R_{nd}\}$. Here, *n* denotes the number of remora, and *d* denotes the dimension of the search space. Meanwhile, the random selection of remora is added to confirm the search for space exploration. This algorithm mainly depends on whether the fitness value increased or not.

ROA uses two phases: the exploitation phase and the exploration phase. In the exploration phase, the formula of the changed location is described in Equation (7).

$$R_i^{t+1} = R_{best}^t - \left(rand(0,1) * \left(\frac{R_{best}^t + R_{rand}^t}{2}\right) - R_{rand}^t\right) \quad (7)$$

Where the position of i^th remora is R_i^{t+1}, R_{best}^t is the best position, R_{rand}^trandom position, and t is the current iteration. Checks the current position of the host to change the position using Equation (8)

$$R_{att} = R_i^t\left(R_i^t - R_{pre}\right) * randn \quad (8)$$

Where R_{pre} is the previous position, R_{att} is the small step to change the position with the *randn*. This mechanism is used to overcome the local optimum issue.

The exploitation phase updates the position derived from Equations (9) to Equation (12).

$$R_{i+1} = d \times e^{a} \times \cos(2\pi a) + R_i \quad (9)$$

$$\alpha = rand(0,1) \times (a - 1) + 1 \quad (10)$$

$$a = -\left(1 + \frac{t}{T}\right) \quad (11)$$

$$D = R_{best} - R_i \quad (12)$$

D is the present optimal solution, α is the random number in [-1, 1], t represents the iteration, and R_{rand} represents the random location of the remora. Then, it reduces the space of the area by using the derivation from Equations (13) to (16).

$$R_i^t = R_i^t + A \quad (13)$$

$$A = B * (R_i^t - C * R_{best}) \quad (14)$$

$$B = 2 * V * rand(0,1) - V \quad (15)$$

$$V = 2 * \left(\frac{1}{max - iter}\right) \quad (16)$$

Where A is denoted as the small step movement for the volume (V) space, C is the constant number. B is used to pretend any space volume. Number, dimension, and maximum iterations are the related factors that are related to the algorithm of computational complexity.

3.4. Sentiment Classification

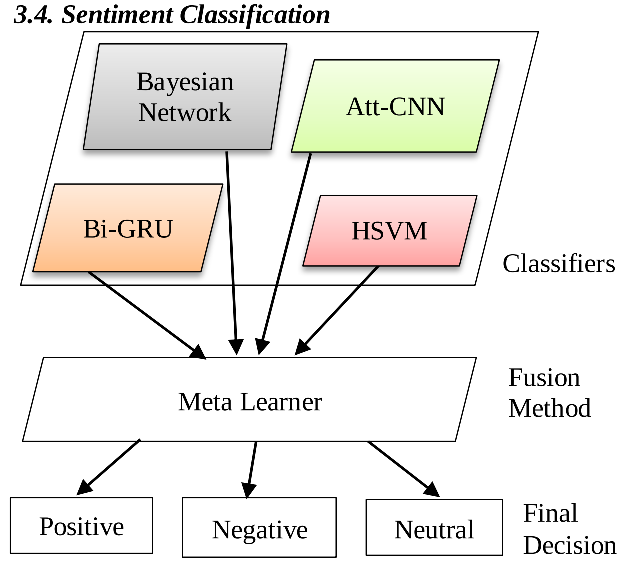

Fig. 4 Overall structure of sentiment classification

This proposed model is based on the fusion method using deep learning and a machine learning classifier for sentiment classification. Using both classifiers will enhance the performance and improve the confidence to get the highest accuracy. The outline of the classification process is displayed in Figure 4. Two deep learning and two machine learning classifiers are trained, and the output is fused to get an accurate prediction using the meta-learner.

Bi-GRU: Figure 5 shows the Layers of the Bi-GRU classifier. A Gated Recurrent Neural Network (GRU) is used to develop the long dependencies in text. This proposed method uses Bi-GRU as a classifier that extracts both the backward and forward sequential dependencies. The context of sentiment words is a significant problem in sentiment analysis, and this feature will be considered. This model uses an embedding layer for mapping the phrase with a pre-trained word vector. Then, the GRU layer is applied to extract the backward and forward contexts, which consist of the update gate (u) and reset gate (r); this mechanism is expressed in Equations (17 & 18).

$$u_t = \delta(W_u h_{t-1} + U_u X_t + b_u) \quad (17)$$

$$r_t = \delta(W_r h_{t-1} + U_r X_t + b_r) \quad (18)$$

Where the logistic softmax function is denoted as δ, the Weight matrix of the memory cell c_t is denoted as U and W of the gate u_t and r_t Which is the input and hidden state, and b represents the bias vector. The hidden state is expressed in the Equations (19 and 20).

$$h_t = (1 - r_t) \odot h_{t-1} + r_t \odot \tilde{h}_t \quad (19)$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}_t}(h_{t-1} \odot r_t) + U_{\tilde{h}_t} x_t) \quad (20)$$

Two hidden layers are combined to extract features and proceeding context, which flow the information in both directions. The output from the word embedding is fed into the Bi-GRU layer to extract the dependencies in both directions forward and backward to recall the previous data by using the derivation expressed in Equations (21, 22 & 23).

$$\vec{h}_{t_{GRU}} = \overrightarrow{GRU}(c_t), t \in [1, m] \quad (21)$$

$$\overleftarrow{h}_{t_{GRU}} = \overleftarrow{GRU}(c_t), t \in [m, 1] \quad (22)$$

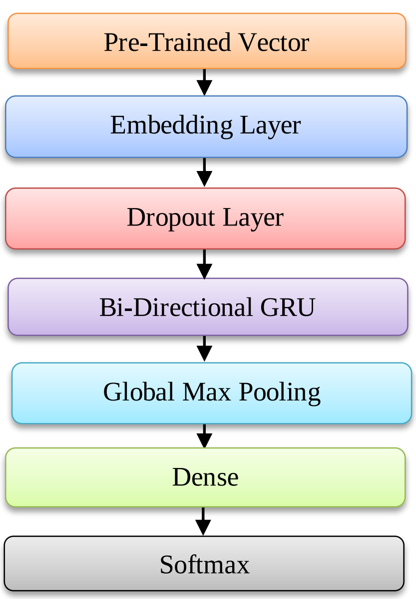

Fig. 5 Layers of Bi-GRU classifier

$$h_{t_{GRU}} = [\vec{h}_{t_{GRU}}, \overleftarrow{h}_{t_{GRU}}] \quad (23)$$

The global max-pooling layer is used to the output of the Bi-GRU model to gain the feature maps. This feature map is then integrated and fed into the dense output layer to provide output.

3.4.1. Attention CNN

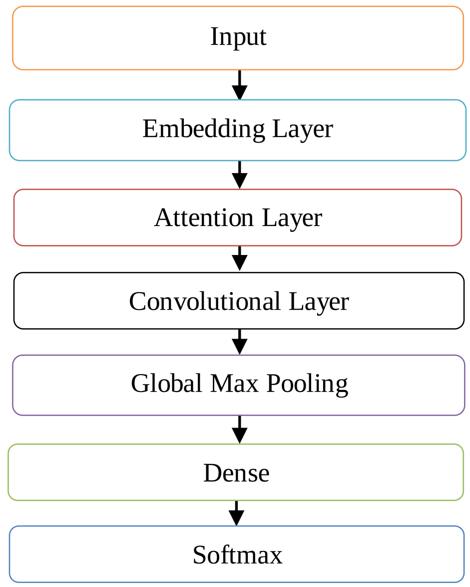

Fig. 6 Layers of attention CNN classifier

CNN is a deep learning technique containing many layers for feature extraction, as shown in Figure 6. Convolutional operation is performed in the input feature through the filters. CNN used a word embedding layer to assign an embedding space for the vocabulary of the sentence S with the words W, where $W = \{w_1, w_2, \ldots\ldots w_W\}$ These words are converted into an embedding matrix vector.

The embedding output is an input of the pooling layer, a standard method used to decrease the feature map size. The suggested method used the max-pooling layer to select the most necessary features of the feature map. The output of the max-pooling layer is fed into the attention mechanism, which is used to predict the attention score by getting attention to the context of the feature generated by the CNN filter. Then, finally, the output is concatenated and passes through the dense layer to get the feature vector as an output.

Final sentence representation and then CNN with attention layer is applied in sentence S with the word W. CNN contains several layers and performs using the linear filter. This filter is used iteratively in the sub-matrices to produce the feature map $M = \{m_0, m_1, \ldots\ldots m_s - h\}$ is expressed in Equation (24).

$$m_i = F.S_{i:i+h-1} \quad (24)$$

Where $i = 0, 1, \dots \dots s - h$ and $S_{i:j}$ is denoted as the submatrix with rows i to j. To reduce the feature map, the CNN uses a global max-pooling layer, which is used to choose the most related and required features b from the feature map that is illustrated in Equation (25).

$$b = \max_{0 \le i \le s-h} (m_j) \tag{25}$$

Then, the result from the pooling layer is concatenated and passed to the dense layer.

3.4.2. Bayesian Network Classifier

Bayesian is supervised learning processed using the Bayes theorem to find the relations among the words. An acyclic graphical model directs it; the nodes in the graph denote the variables, and the edges represent the probabilistic influence relationship.

The Bayesian network can handle incomplete datasets, and in the classification process, it can read relations among the attributes. The Bayes theorem is applied to achieve the posterior probability of the input data in a class variable C. The new input data $a_1, a_2, \dots . a_n$is classified by Equation (26).

$$C = \underset{c}{\operatorname{argmax}} \quad p(C = c | X_1 = a_1 \dots \dots, X_n = a_n) \tag{26}$$

The posterior probability after the application of the Bayes theorem is expressed in Equation (27).

$$p(C = c | X_1 = a_1 \dots \dots, X_n = a_n) \propto p(C = c) p(X_1 = a_1, \dots \dots X_n = a_n | C = c) \tag{27}$$

Selection of the base classifier is based on high diversity and low complexity between the base classifier. This paper uses two machine learning algorithms and two deep learning algorithms as base classifiers; they are Bayesian Network (BN), Heterogeneous SVM (HSVM), BI-GRU, and Attention CNN (Att-CNN). This proposed method used an ensemble method called stacking to combine these classifiers. Self-adaptive stacking ensemble method is used in this paper.

Here, $p(C = c)$ is the prior probability that can be straightforwardly estimated by the variable. The second part is hard to complete, so the naïve assumption is used in the attributes, and then it is called the Naïve Bayes classifier. The assumption of conditional independence is released, and then the posterior probability is expressed as Equation (28).

$$p(C = c | X_1 = a_1 \dots \dots, X_n = a_n) \propto p(C = c) \prod_{i=1}^{n} p(X_i = a_i | \pi_i) \tag{28}$$

Where $\pi_i$is a parent node of X_i in the Naïve Bayes classifier.

3.4.3. Heterogeneous SVM Classifier

An SVM is a supervised learning algorithm employed to classify the data. It will plot the data in the form of a vector in space. SVM classifier with heterogeneous data is called Heterogeneous SVM (HSVM). It is used to classify heterogeneous data by mapping the nominal attributes into real space by reducing the error. The objective function of HSVM is defined in Equation (29).

$$H = \frac{R^2}{\gamma^2} = R^2 * \|w\|^2 \tag{29}$$

Where the radius of enclosing all the samples isR^2, γ^2 denotes the margin between classes in the feature space. After mapping, the nominal and numerical attributes are combined. The numerical qualities are involved in the whole training procedure and in the mapping of values to reduce the generalization error. The process is initialized by using the heterogeneous data $H = \{h_1, h_2, \dots \dots . h_n\}$ then initializing the nominal values $a_i^k$with the probability $p(k|h_i)$, which is expressed in Equation (30).

$$p(k|h_i) = \frac{N_{a_i,k,c}}{N_{a_i,k}} \tag{30}$$

Where $N_{a_i,k,c}$ is the total number of times of a_i in-class c and $N_{a_i,k}$ denotes the total times of all classes, which are positive, negative, and neutral.

3.5. Stacking Ensemble Method

The stacked ensemble is a heterogeneous ensemble that stimulates the diversity of classifiers since the base classifiers in the ensemble method have a different algorithm for learning. From Figure 7, the training dataset is represented as $n \times m$ where *n* is the number of rows and f is the number of columns. Then, the number of base classifiers is represented as b. In the training process, the datasets are the input to the 1st algorithm, and then the representation of the b classifier in the testing set is ψ^b and so on.

Without cross-validation, the classification algorithm learns the training data set and creates a classification; this classification generates a metadata z for all four classifiers using cross-validation, as explained in Figure 8. The two principal tasks are repeated four times for the four algorithms. They created four metadata; z^1, z^2, z^3, z^4 are column-combined using the class label to generate the metadata Z.

This generated metadata is the input data to the meta learner. The testing steps include the base classifier $\psi^1, \psi^2, \psi^3, \psi^4$ use testing data as input and provide a prediction; these predictions are combined and fed to the trained meta-learner. The prediction from the meta-learner is considered as a final prediction output.

K-fold cross-validation is the basic approach to creating the metadata with five folds. The training datasets are divided into five equal folds {F, F2, F3, F4, F5}. The k-folds, which are blue-colored, are used to train the classifier, and then the yellow-colored folds are used as test data to predict, and the expected folds are combined into metadata.

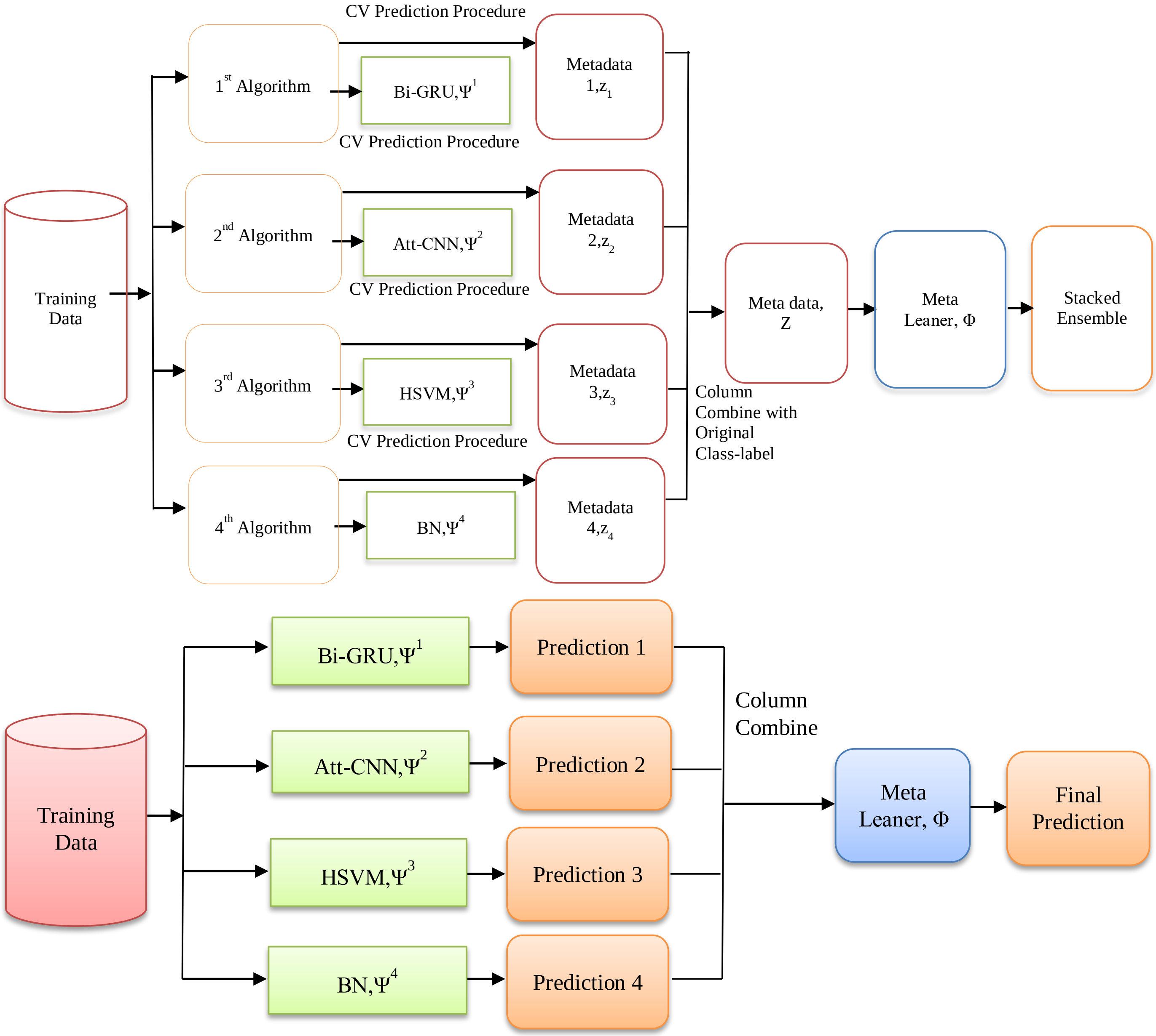

Fig. 7 Architecture of stacking ensemble method

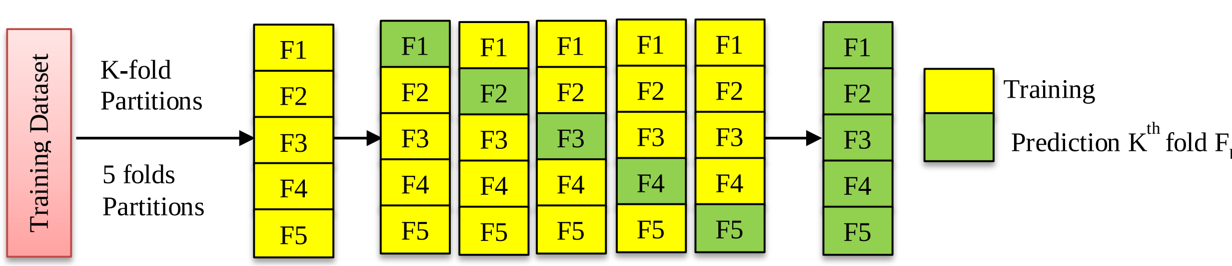

Fig. 8 K-fold cross-validation framework

3.6. Self-Adaptive Ensemble Method

The best combination of dissimilar base classifiers and their hyper-parameters for various datasets. The feature vector n is represented as $x = [x_1, x_2, \dots, x_n]$, $f = [f_1, f_2, \dots f_k]$ is denoted as the k train base classifiers, and the output of m train base classifiers are denoted as $z = [z_1, z_2, \dots . z_n]$, then the outcome of the train-based classifier is expressed in Equation (31).

$$z_i = f(x_i)n \tag{31}$$

The k train base classifier selected the meta-learner g. Therefore the output $\hat{Y} = [\hat{y}_1, \hat{y}_2, \dots \hat{y}_n]$ is expressed in Equation (32)

$$\hat{y}_i = g(Z_i) \tag{32}$$

The prediction accuracy of the base learner classifier is represented using the Minimum Mean Square Error. If the value of MMSE is low, then the base classifiers' performance is high. M base learner can produce the combination of meta learner and k base classifier, and the combination of the base classifier is expressed as Equation (33).

$$minJ(\theta) = \frac{1}{2}\sum_{n=1}^{s}\|y - \hat{y}\|^2$$

$$s.t\begin{cases}0 \le i \le k\\ f_i \in f\\ g \in f\end{cases} \tag{33}$$

$b_i = [b_{i1}, b_{i2} \dots b_{is_i}]$ is represented as a parameter vector of the i^th^ classifier when b_{ij} is the j^th^ parameter from several parameters s_i in the i^th^ base classifier. Then, in the self-adaptive stacking ensemble method, the m classifier generated p is expressed in Equation (34).

$$p = \left[\prod_{i=1}^{m}\left(\prod_{j=1}^{s_i}\|b_{ij}\| + 1\right) - 1].m\right] \tag{34}$$

The integration of the base classifier is selected using the incorporated parameter. The ranges of b_i with s_i are $\left(a_i\left[a_{i1}a_{i2} \dots a_{is_i}\right]\right)$ to $\left(c_i\left[c_{i1}c_{i2} \dots c_{is_i}\right]\right)$.

$$minj(\theta) = \frac{1}{s}\sum_{n=1}^{s}\|y - \hat{y}\|^2$$

$$s.t\begin{cases}0 \le i \le k\\ f_i \in f\\ g \in f\\ b_i \in [a_i, c_i]\end{cases} \tag{35}$$

4. Experiment Result

4.1. Dataset

This study uses four datasets to calculate the performance of the sentiment classifier. The datasets are IMDB dataset with 50k movie reviews, which is divided into two parts, 25k for testing and 25k for training, which is taken from (https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews).

Twitter entity sentiment analysis dataset contains the messages about the entities it is used to predict the model positive, negative or neutral, which is taken from (https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis).

The Twitter Airline Sentiment dataset https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment contains the airline reviews used to classify as negative, positive, and neutral.

Amazon calls phone reviews dataset https://www.kaggle.com/code/mamunalbd4/amazon-cell-phones-reviews/data contains the review messages of the cell phone from Amazon. These are used for both the testing and training of the proposed method. Some reviews from the four datasets and their predicted score are tabulated in Table 1.

Table 1. Datasets and their predicted score

Dataset	Text	Sentiment	Model Prediction
Dataset 1	The plot was incredibly implausible and unclear. This is a real Oprah movie. (In Oprah's universe, women are victims, and men are villains.)	Negative	Negative
	Adrian Pasdar is fantastic in this movie. He makes an exciting partner.	Positive	Positive
Dataset 2	This Scene Hit me every time	Positive	Positive
	In Hearthstone, anyone who uses an albatross deck for bad luck is a real cop.	Neutral	Negative
Dataset 3	Lost Luggage	Negative	Negative
	wow, this just blew my mind	Positive	Positive
Dataset 4	Not a good product	Negative	Negative
	A good little phone	Positive	Positive

4.2. Evolution Criteria

In this suggested paper, there are four evolution metrics: precision, AUC, Accuracy, F1-Score, and Recall, which are used to evaluate the performance of the proposed model. These are illustrated in Equations (36) to (40).

$$Precision = \frac{TP}{TP+FP} \tag{36}$$

$$Recall = \frac{TP}{TP+FN} \tag{37}$$

$$F1\ score = \frac{2(Precision \times Recall)}{Precision+Recall} \tag{38}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{39}$$

$$AUC = \int_0^1 TPR(FPR^{-1}(x))dx \tag{40}$$

Where TP is a True positive, TN is a true negative, FP is a false positive, and FN is a false negative.

4.3. Parameter Settings

The proposed sentiment analysis model is implemented using Python software in the Windows 10 operating system. The parameter setting for the deep learning classifier is tabulated in Table 2, and the parameter of the feature selection method is tabulated in Table 3.

4.4. Confusion Matrix

The confusion matrix is obtained using the different datasets as input. This matrix is constructed to estimate the performance of the proposed technique utilizing different datasets. The values from the confusion matrix are taken to calculate the accuracy, precision, Recall, AUC, and F1 score values. Figures (9) to (12) show the confusion matrix of four datasets. Using all the datasets gives a high true positive value, true negative, and true neutral value. The IMDB dataset it provides an accuracy of 96%, the Twitter-Entity-Sentiment Analysis gives 98% accuracy, the highest value of accuracy is issued by the Twitter-Airline-Sentiment dataset with an accuracy of 99%, and the Amazon cell phone review dataset gives an accuracy of 98%.

Table 2. Parameters of classifiers

Classifiers	Parameter	Values
HSVM	Kernel type	Multi kernel
	Distance measure	Euclidean distance and Heterogeneous Euclidean Overlap Metric (H-EOM).
Bayesian Network	Posterior probability calculation	Bayes theorem
Att-CNN	Optimizer	RMSprop
	Learning rate	0.0001
	Batch size	64
	Epochs	30
	Iteration	100
	Loss Function (LF)	Categorical cross entropy
	Activation Function (AF)	Softmax
BI-GRU	Number of nodes	64
	Optimizer	Adam
	LF	Binary cross entropy
	AF	Softmax
	Learning rate	0.001
	Dropout rate	0.01

Table 3. Parameter settings of feature selection method

Algorithm	Parameters	values
Remora Optimization Algorithm	Number of population	30
	Number of iteration	100
	Fitness function	Minimization of classification error
	Remora factor, C	0.1

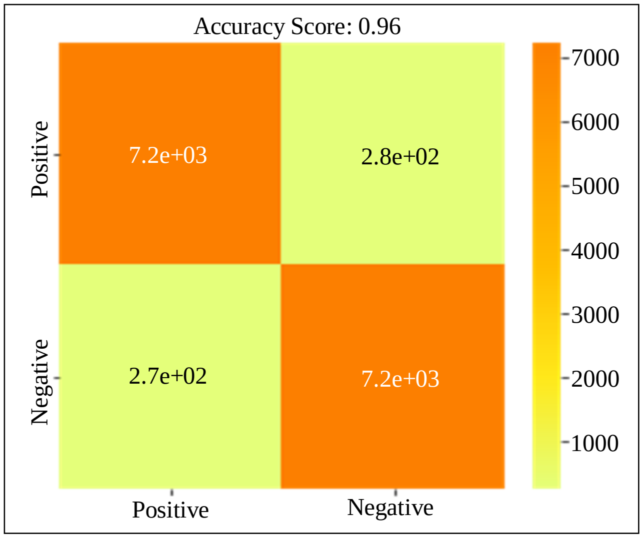

Fig. 9 Confusion matrix using IMDB-50k-movie review

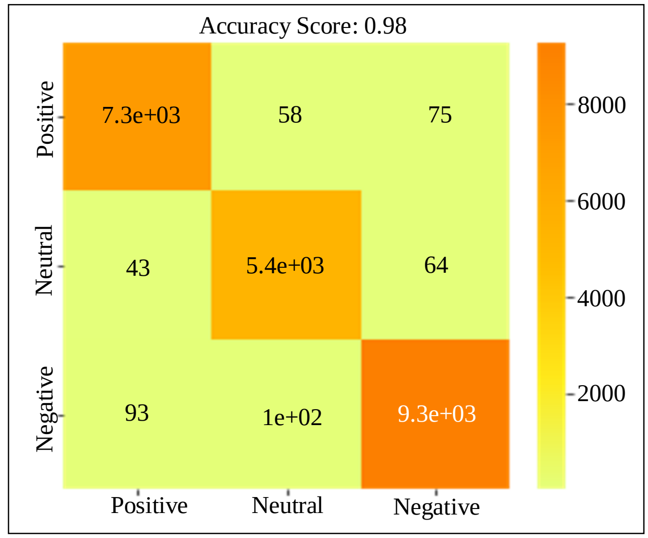

Fig. 10 Confusion matrix using twitter-entity-sentiment-analysis

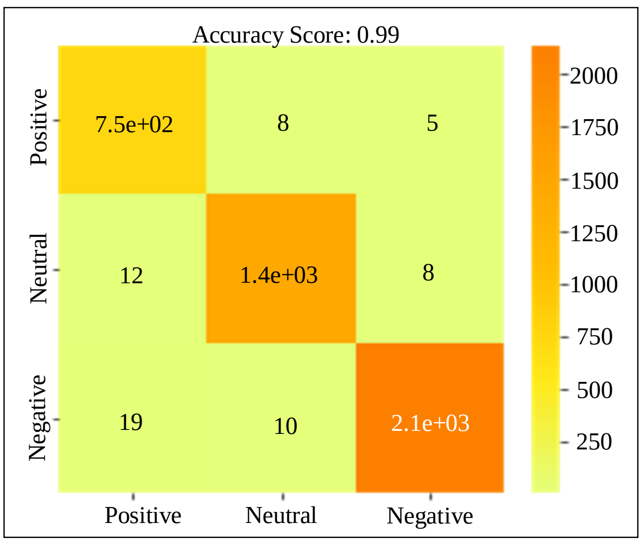

Fig. 11 Confusion matrix using Twitter-airline-sentiment dataset

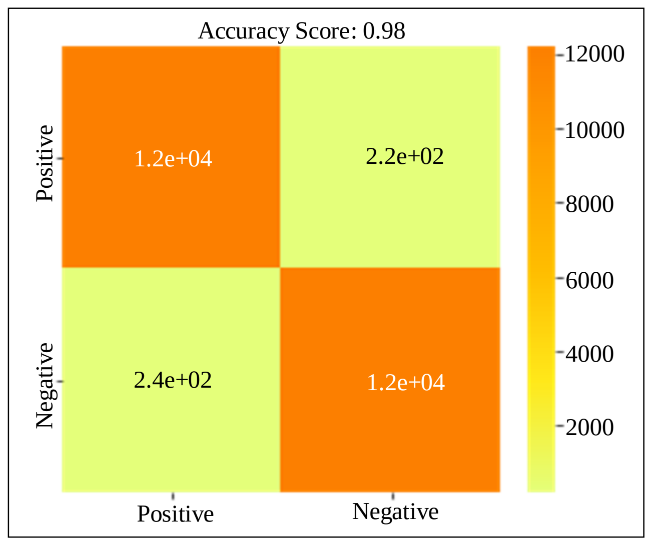

Fig. 12 Confusion matrix using Amazon-cell-phone-reviews datasets

4.5. Training Accuracy and Loss Curve

The model's performance during the training period is observed in each iteration. This proposed model takes 100 iterations using four datasets. The accuracy of this iteration level is plotted as a curve, which is displayed in Figure 13. This graph increases the accuracy value by increasing the iteration; the accuracy remains stable when it reaches the maximum iteration. This stable value is considered the accuracy of the model. Compared with existing models, the proposed sentiment analysis using deep and machine learning models provides high accuracy. Training loss is a performance metric that reports the loss during the experiment iteration. Figure 14 shows the loss curve during the iteration; the graph shows that the loss decreases when the iteration improves. Compared with other methods, the proposed method has minimum loss during training. HSVM, Bayesian network, Att-CNN, and Bi-GRU were used as the three base learners in the ensemble model's construction with the Meta-learner.

Ten prediction experiments were conducted to evaluate the suggested adaptive stacking ensemble model's stability and efficacy. After implementing the proposed ensemble model for ten rounds, the following prediction outcomes were attained in Table 4. The suggested ensemble model's overall prediction performance was determined by averaging the values of the ten rounds.

The suggested ensemble model's overall prediction performance was determined by averaging the values of the ten rounds. If the threshold value of AUC is more significant than 0.9, then it is referred to as good performance, an AUC value less than 0.9 is considered fair performance, and the AUC value with less than 0.7 is considered bad performance.

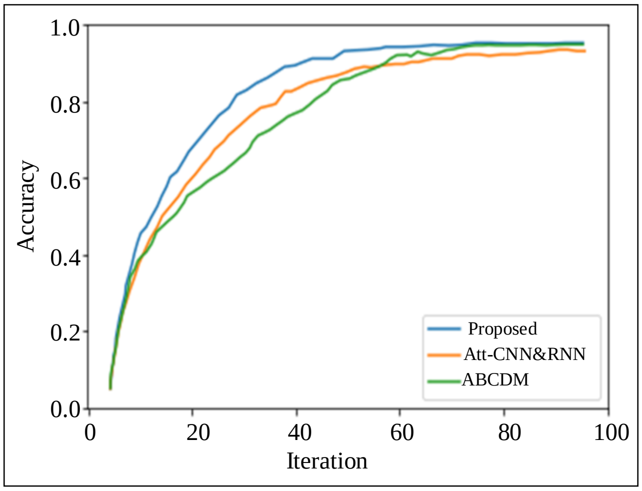

Fig. 13 Training accuracy curve

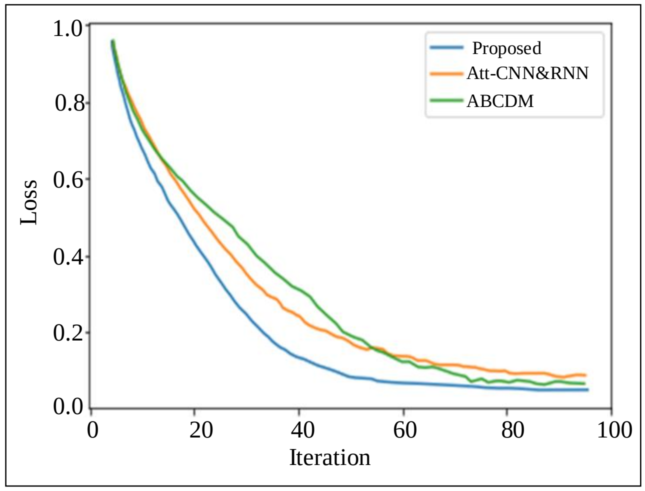

Fig. 14 Training loss curve

Table 4. Ten runs of testing the suggested ensemble model's results

Class Label	Recall	Accuracy	F1	Error	AUC
0	0.8928	0.8511	0.8725	0.1417	0.9376
1	0.8554	0.8869	0.8614		
0	0.8841	0.8584	0.8779	0.1473	0.9338
1	0.8517	0.8724	0.8638		
0	0.8883	0.8658	0.8775	0.1405	0.9263
1	0.8716	0.8712	0.8656		
0	0.8734	0.8574	0.8635	0.1511	0.9294
1	0.8528	0.8669	0.8594		
0	0.8811	0.8585	0.8731	0.1448	0.9298
1	0.8537	0.8789	0.8674		
0	0.8948	0.8576	0.8775	0.1413	0.9261
1	0.8542	0.8888	0.8638		
0	0.8833	0.8627	0.8744	0.1457	0.9296
1	0.8645	0.8734	0.8639		
0	0.9054	0.8695	0.8876	0.1341	0.9275
1	0.8525	0.8917	0.8781		
0	0.8971	0.8625	0.8734	0.1341	0.9299
1	0.8924	0.8868	0.8785		
0	0.8931	0.8644	0.8813	0.1341	0.9289
1	0.8673	0.8827	0.8795		
Minimum	0.8517	0.8511	0.8614	0.1341	0.9261
Maximum	0.9054	0.8888	0.8876	0.1511	0.9376
Average	0.8812	0.8734	0.8798	0.1415	0.9296

Table 5. Comparison of classification outcomes using individual algorithms

Approaches	Class Label	Accuracy	Recall	F1	Error	AUC
HSVM	0 1	0.7857 0.8301	0.8484 0.7928	0.8258 0.7949	0.1839	0.8566
Bayesian Network	0 1	0.7924 0.7508	0.7479 0.8031	0.7939 0.7762	0.2087	0.7809
Att-CNN	0 1	0.7620 0.7288	0.7172 0.7928	0.7688 0.7500	0.1967	0.8024
BI-GRU	0 1	0.7601 0.7133	0.7069 0.7935	0.7856 0.7823	0.1578	0.7856
Proposed	0 1	0.8559 0.9023	0.8788 0.8952	0.9014 0.8829	0.1028	0.9342

A proportional study was conducted in which multiple approaches were chosen for performance comparison to gain additional insight into the prediction performance for the suggested model, shown in Table 5. The findings show that the proposed ensemble model has superior prediction performance compared to the other four models, each with a single technique. The model concentrates on the facts of the indicators designated as one since the prediction findings are used for competition candidate selection.

The SVM model comes in second place, with the precision of the suggested model receiving the highest score of 0.8710. There is an 87.1% chance that pupils will be accurately identified as 1 when using the proposed methodology. The decision tree comes after the recall value in the suggested model, which is 0.8351. With the AUC value of 0.9138, the presented model performs the best out of the five. In addition, when compared to other methods, the suggested model has the lowest error rate. The AUC of all five models is significantly bigger than 0.5, as seen from the comparison of ROC curves in Figure 15. Based on the five indicators, Figure 16 data demonstrate that the suggested model performs the best. Statistical significance was inferred from the differences between the models when p was less than 0.05.

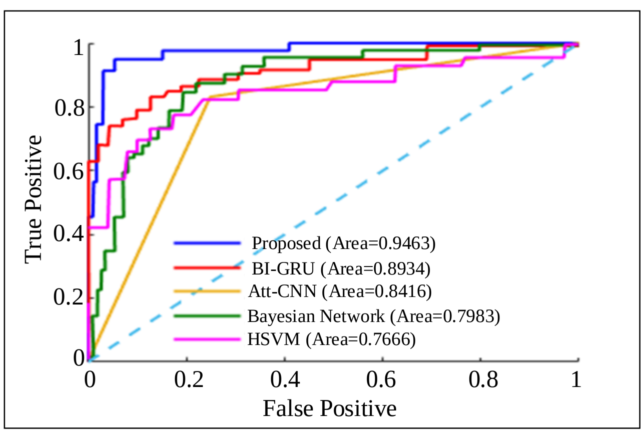

Fig. 15 The suggested model's receiver operating characteristic curve in comparison to other compared algorithms

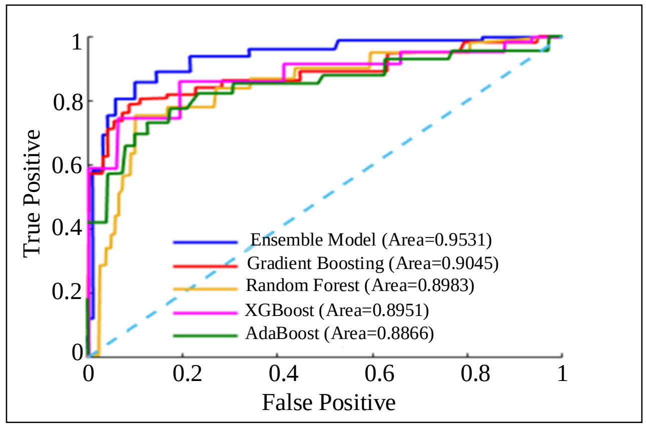

Fig. 16 Characteristic curves of the proposed model's receiver and the existing ensemble methods

4.6. Comparison of Evaluation Criteria

The estimation criteria are mandatory to observe the performance of the proposed model. The evaluation criteria utilized in this model are F1-score Accuracy, Recall, Precision, and Kappa. Kappa is the metric used to measure the inter-rater reliability for categorical instances, and K varies from 0 to 1. If the value of K is more significant than (0. 6), it denotes the extensive contract between the predicted and actual class. Then, these metrics are compared with four existing methods using four datasets. The K value of the proposed method with four datasets is 0.74, 0.76, 0.77, and 0.76.

Figures 17 and 18 show the model's true negative and true positive predictions using the IMDB dataset. Then, the performance metrics are compared with the four previous methods to observe the efficiency of the suggested model. Both graphs show that the proposed method is highly valued in all metrics. The Precision value for the positive is 94%, the negative is 96%, the recall value for the positive is 95%, and the negative is 94%. The F1 positive score value is 95%, and the negative is 96%. The accuracy is about 96%.

Figure 19, Figure 20, and Figure 21 show the true negative, true positive, and True Neutral prediction of the model utilizing the Twitter-entity-sentiment-analysis dataset then, the performance metrics are compared with the four other methods to observe the efficiency of the presented model. The three graphs show that the proposed method is highly valued in all metrics. The precision value for the positive is 94%, the negative is 93%, the recall value for the positive is 96%, and the negative is 94%.

The F1 positive score is 95%, and the negative is 93%. The accuracy is about 98%. Figure 22, Figure 23, and Figure 24 show the true negative, true positive, and True neutral prediction of the approach utilizing the Twitter-Airline-Sentiment dataset then, the performance metrics are compared with the four other techniques to observe the effectiveness of the suggested technique.

The three graphs show that the proposed method is highly valued in all metrics. The precision value for the positive is 94%, and the negative is 94%; the recall value for the positive is 95%, and the negative is 96%. The F1 score value of positive is 94%, and the negative is 95%. The accuracy is about 99%.

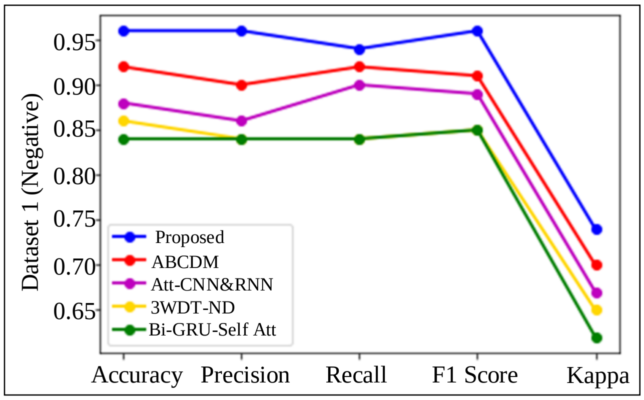

Fig. 17 True negative prediction using IMDB-50k-movie review

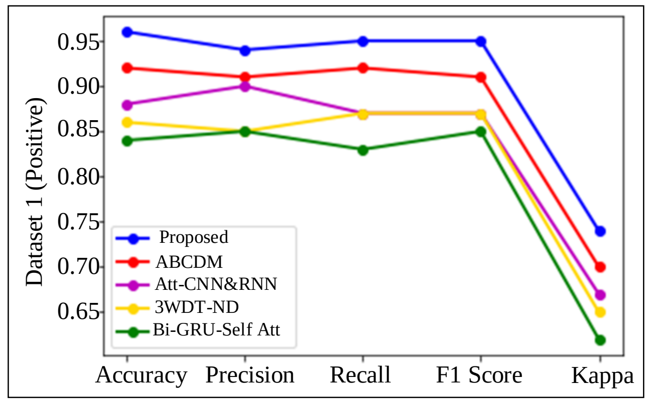

Fig. 18 True positive prediction using IMDB-50k-movie review

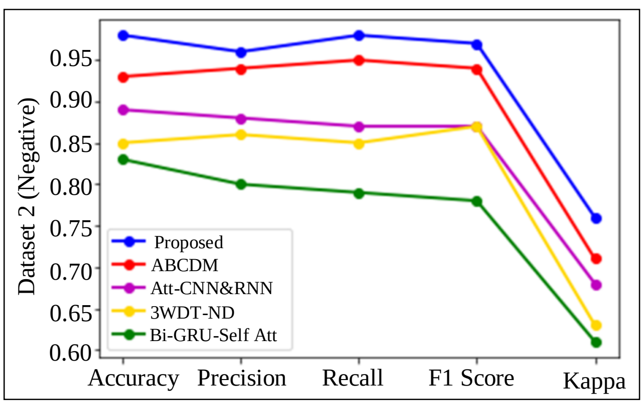

Fig. 19 True negative prediction using Twitter-entity-sentiment-analysis

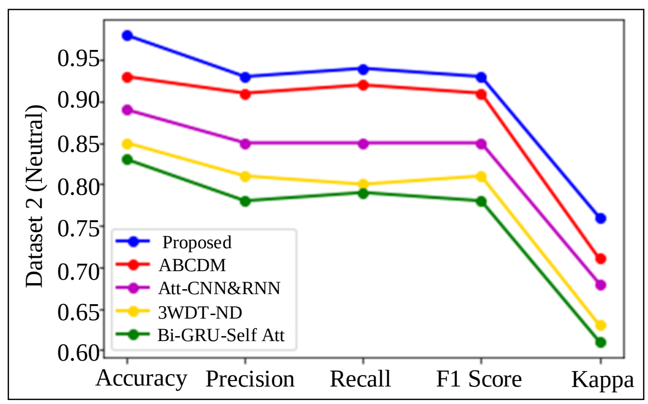

Fig. 20 True neutral prediction using Twitter-entity-sentiment-analysis

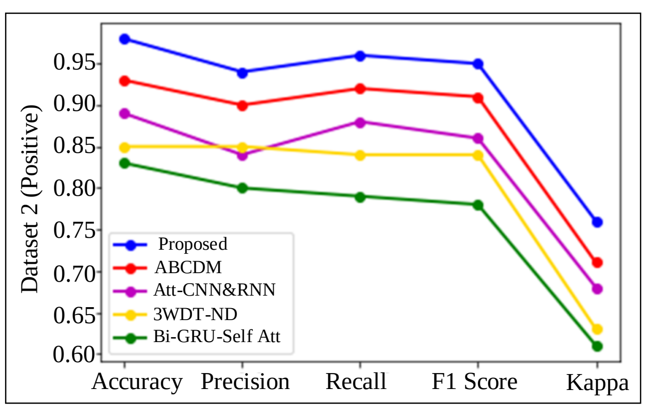

Fig. 21 True positive prediction using Twitter-entity-sentiment-analysis

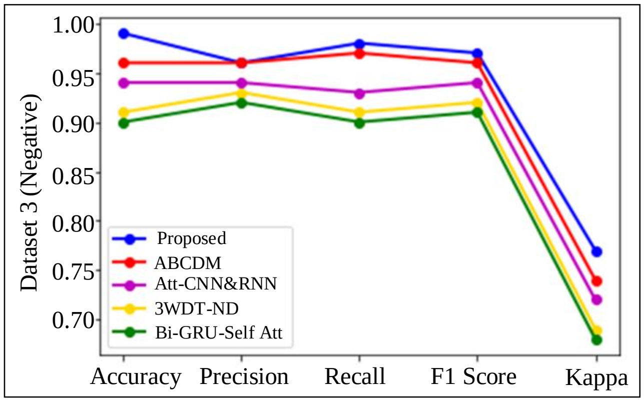

Fig. 22 True negative prediction using Twitter-airline-sentiment dataset

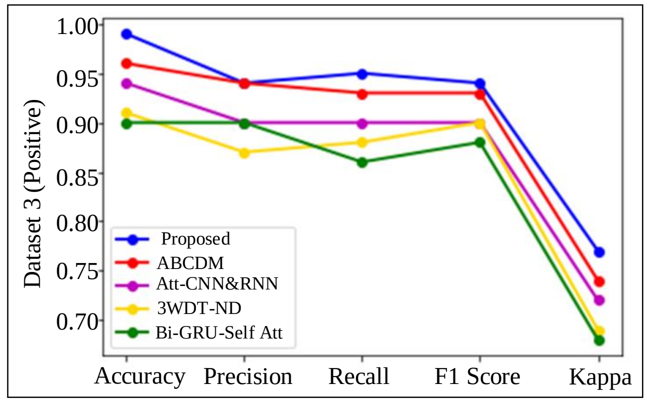

Fig. 23 True positive prediction using Twitter-airline-sentiment dataset

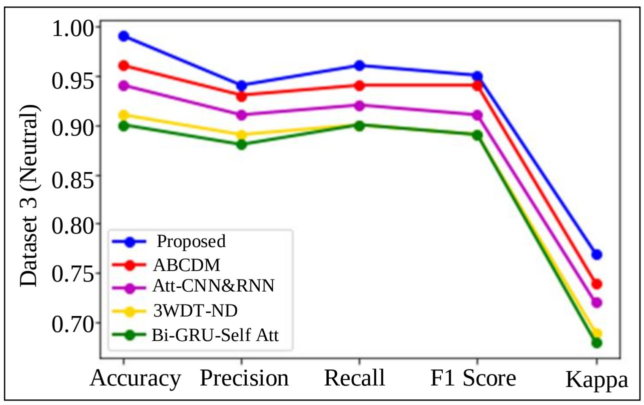

Fig. 24 True neutral prediction using Twitter-airline-sentiment dataset

Figure 25 and Figure 26 show the true negative, true positive of the model employing the Amazon-cell-phone-reviews dataset then, the performance metrics are compared with the four other methods to observe the efficiency of the suggested model.

The two graphs show that the proposed method is highly valued in all metrics. The precision value for the positive is 97%, the negative is 98%, the recall value for the positive is 97%, and the negative is 96%. The F1 score value for the positive is 98%, and the negative is 94%. The accuracy is about 98%.

The average, highest, and lowest accuracy of the existing methods and the presented approach using the four datasets are tabulated in Table 6.

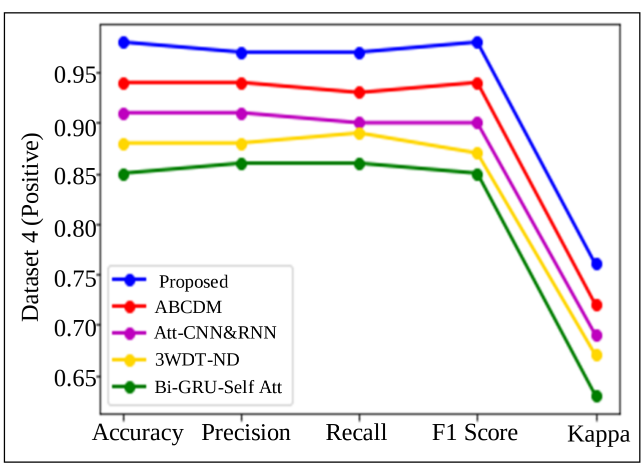

Fig. 25 True negative prediction using Amazon-cell-phone-reviews datasets

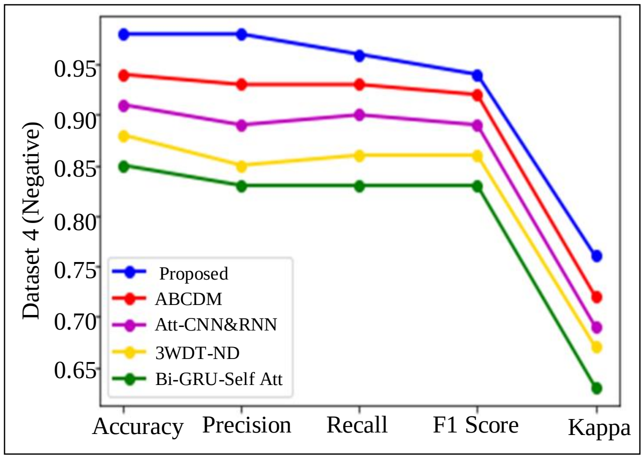

Fig. 26 True positive prediction using Amazon-cell-phone-reviews datasets

Table 6. Average, highest, and lowest accuracy of the models using datasets

Datasets	Technique	Average Accuracy	Highest Accuracy	Lowest Accuracy
IMDB-50k-Movie Review	Proposed	0.94512	0.96753	0.92571
	ABCDM	0.91078	0.91357	0.90764
	Att-CNN & RNN	0.88046	0.88159	0.87339
	3WDT-ND	0.86148	0.86951	0.86295
	Bi-GRU-Self att	0.84364	0.84456	0.84132
Twitter-Entity-Sentiment-Analysis	Proposed	0.98123	0.98452	0.97613
	ABCDM	0.93234	0.93563	0.93014
	Att-CNN & RNN	0.89564	0.89785	0.89126
	3WDT-ND	0.85651	0.85894	0.85369
	Bi-GRU-Self att	0.83124	0.83426	0.82369
Twitter-Airline-Sentiment	Proposed	0.99342	0.99671	0.99145
	ABCDM	0.96496	0.96789	0.96256
	Att-CNN & RNN	0.94432	0.94654	0.94039
	3WDT-ND	0.91258	0.91327	0.91147
	Bi-GRU-Self att	0.90391	0.90756	0.90143
Amazon-Cell-Phone-Reviews	Proposed	0.98319	0.98324	0.98225
	ABCDM	0.92691	0.92761	0.92439
	Att-CNN & RNN	0.89795	0.89856	0.89546
	3WDT-ND	0.87422	0.87783	0.87523
	Bi-GRU-Self att	0.85923	0.85987	0.85674

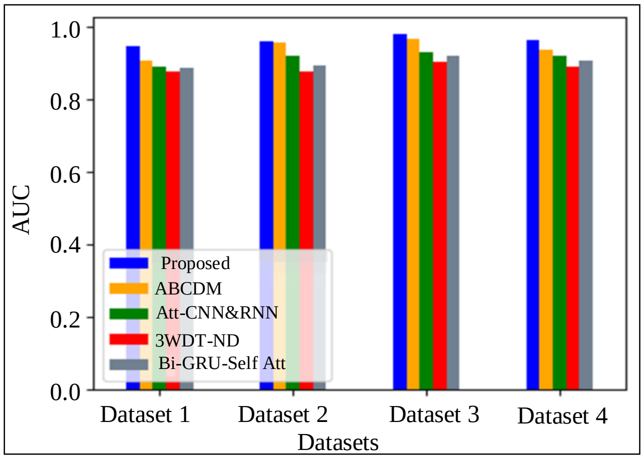

Fig. 27 Comparison of AUC

4.7. Comparison of AUC

Area Under Curve (AUC) is the evaluation of the classifier. Figure 27 compares the suggested technique with the other methods using four datasets. The graph shows that using four datasets, the proposed model gives the highest value of AUC.

4.8. Comparison of Training Time

Figure 28 shows a bar graph to compare the classifier's training time using the four datasets. By using all the datasets, the training time of the classifier is low compared with other existing methods.

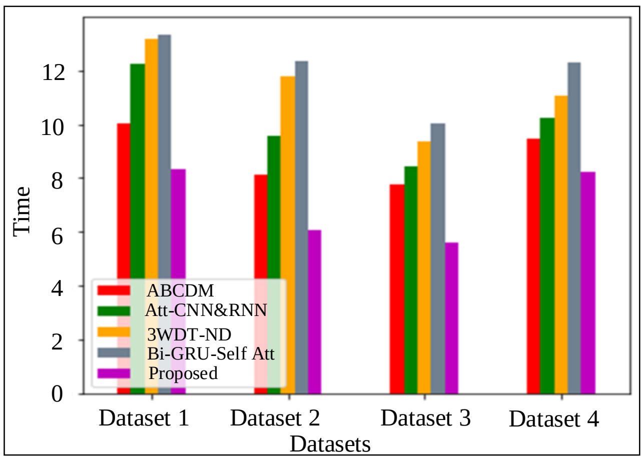

Fig. 28 Training time of the classifiers

5. Conclusion

This paper proposed a sentiment analysis using the fusion of two deep learning classifiers, Bi-GRU and Attention CNN, with two machine learning classifiers, the Bayesian network classifier, and Heterogeneous SVM, with the polarized word embedding techniques and feature selection using MI and ROA. This paper solves the heterogeneous and incomplete data problem, reduces the overlapping, high dimensional space, and optimizes the polarity values.

The classifiers are tested and trained using the IMDB dataset with 50k Movie Reviews, the Twitter-Entity-Sentiment-Analysis dataset, the Twitter-Airline-Sentiment dataset, and the Amazon-Call-Phone-Reviews dataset. Then, the evaluation is compared using the four existing methods, Bi-GRU-self att, 3WDT-ND, Att-CNN & RNN, and ABCDM. The proposed sentiment analysis with the fusion of deep learning and machine learning provides the maximum accuracy of 99.3% using the Twitter-Airline-Sentiment dataset. Future work aims to include more advanced classifiers in the fusion method for better performance.

Acknowledgments

I confirm that all authors listed on the title page have contributed significantly to the work, have read the manuscript, attested to the validity and legitimacy of the data and its interpretation, and agreed to its submission.

References

[1] Seng Zian, Sameem Abdul Kareem, and Kasturi Dewi Varathan, "An Empirical Evaluation of Stacked Ensembles with Different Meta-Learners in Imbalanced Classification," *IEEE Access*, vol. 9, pp. 87434-87452, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Aytuğ Onan, "Sentiment Analysis on Product Reviews Based on Weighted Word Embeddings and Deep Neural Networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Duyu Tang, and Meishan Zhang, "Deep Learning in Sentiment Analysis," *Deep Learning in Natural Language Processing*, pp. 219-253, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mohammad Ehsan Basiri et al., "A Novel Fusion-Based Deep Learning Model for Sentiment Analysis of COVID-19 Tweets," *Knowledge-Based Systems*, vol. 228, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Weili Jiang et al., "SSEM: A Novel Self-Adaptive Stacking Ensemble Model for Classification," *IEEE Access*, vol. 7, pp. 120337-120349, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] B. Oscar Deho et al., "Sentiment Analysis with Word Embedding," *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*, Accra, Ghana, pp. 1-4, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Seyed Mahdi Rezaeinia et al., "Sentiment Analysis Based on Improved Pre-Trained Word Embeddings," *Expert Systems with Applications*, vol. 117, pp. 139-147, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zufan Zhang, Yang Zou, and Chenquan Gan, "Textual Sentiment Analysis via Three Different Attention Convolutional Neural Networks and Cross-Modality Consistent Regression," *Neurocomputing*, vol. 275, pp. 1407-1415, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohd Usama et al., "Attention-Based Sentiment Analysis Using Convolutional and Recurrent Neural Network," *Future Generation Computer Systems*, vol. 113, pp. 571-578, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Yaxing Pan, and Mingfeng Liang, "Chinese Text Sentiment Analysis Based on BI-GRU and Self-Attention," *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, pp. 1983-1988, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Huawen Liu et al., "Feature Selection with Dynamic Mutual Information," *Pattern Recognition*, vol. 42, no. 7, pp. 1330-1339, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mohammad Ehsan Basiri et al., "A Novel Method for Sentiment Classification of Drug Reviews Using Fusion of Deep and Machine Learning Techniques," *Knowledge-Based Systems*, vol. 198, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Oscar Araque et al., "Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications," *Expert Systems with Applications*, vol. 77, pp. 236-246, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Babacar Gaye, Dezheng Zhang, and Aziguli Wulamu, "A Tweet Sentiment Classification Approach Using a Hybrid Stacked Ensemble Technique," *Information*, vol. 12, no. 9, pp. 1-19, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Yasin Gormez et al., "FBSEM: A Novel Feature-Based Stacked Ensemble Method for Sentiment Analysis," *International Journal of Information Technology and Computer Science*, vol. 12, no. 6, pp. 11-22, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Basant Subba, and Simpy Kumari, "A Heterogeneous Stacking Ensemble Based Sentiment Analysis Framework Using Multiple Word Embeddings," *Computational Intelligence*, vol. 38, no. 2, pp. 530-559, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Azadeh Mohammadi, and Anis Shaverizade, "Ensemble Deep Learning for Aspect-Based Sentiment Analysis," *International Journal of Nonlinear Analysis and Applications*, vol. 12, pp. 29-38, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yanling Zhou et al., "Deep Learning Based Fusion Approach for Hate Speech Detection," *IEEE Access*, vol. 8, pp. 128923-128929, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Teragawa Shoryu, Lei Wang, and Ruixin Ma, "A Deep Neural Network Approach using Convolutional Network and Long Short Term Memory for Text Sentiment Classification," *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China, pp. 763-768, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Hai Ha Do et al., "Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review," *Expert Systems with Applications*, vol. 118, pp. 272-299, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Dejun Zhang et al., "Combining Convolution Neural Network and Bidirectional Gated Recurrent Unit for Sentence Semantic Classification," *IEEE Access*, vol. 6, pp. 73750-73759, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Harleen Kaur et al., "A Proposed Sentiment Analysis Deep Learning Algorithm for Analyzing COVID-19 Tweets," *Information Systems Frontiers*, vol. 23, pp. 1417-1429, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mattia Atzeni, and Diego Reforgiato Recupero, "Multi-Domain Sentiment Analysis with Mimicked and Polarized Word Embeddings for Human-Robot Interaction," *Future Generation Computer Systems*, vol. 110, pp. 984-999, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Shibaprasad Sen et al., "A Bi-Stage Feature Selection Approach for COVID-19 Prediction Using Chest CT Images," *Applied Intelligence*, vol. 51, pp. 8985-9000, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Hastari Utama, "Sentiment Analysis in Airline Tweets Using Mutual Information for Feature Selection," *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, pp. 295-300, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jorge R. Vergara, and Pablo A. Estévez, "A Review of Feature Selection Methods Based on Mutual Information," *Neural Computing and Applications*, vol. 24, pp. 175-186. 2014. [CrossRef] [Google Scholar] [Publisher Link]

[27] Heming Jia, Xiaoxu Peng, and Chunbo Lang, "Remora Optimization Algorithm," *Expert Systems with Applications*, vol. 185, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Gonzalo A. Ruz, Pablo A. Henríquez, and Aldo Mascareño, "Sentiment Analysis of Twitter Data during Critical Events through Bayesian Networks Classifiers," *Future Generation Computer Systems*, vol. 106, pp. 92-104, 2020. [CrossRef] [Google Scholar] [Publisher Link]