Journal of Computer Science

Research Article

Sentiment Analysis using Light Weight - Gradient Boosting Machine based Feature Selection

[1]Bikku Ramavath, [2]Srikanth Kadainti and [3]Nemani Subash

[1]Department of Information Technology, Malla Reddy University, Hyderabad, India
[2]Department of Data Science, Malla Reddy University, Hyderabad, India
[3]Department of Electronics and Communication Engineering, Malla Reddy University, Hyderabad, India

Article history
Received: 13-12-2024
Revised: 26-02-2025
Accepted: 19-03-2025

Corresponding Author:
Bikku Ramavath
Department of Information Technology, Malla Reddy University, Hyderabad, India
Email: bikkucs@outlook.com

Abstract: Sentiment analysis is a significant task in Natural Language Processing (NLP) that differentiates the emotions and opinions expressed in text or reviews. The sentiment analysis is challenging due to the complex language patterns and inappropriate or redundant features used for classification. In this research, the Light Weight - Gradient Boosting Machine (LWGBM) based feature selection is proposed to choose relevant features for classification to eliminate inappropriate or redundant features and learn the complex language patterns. Then, the classification is performed by using H2O Automatic Machine Learning (H2O AutoML) algorithm which classifies the sentiments as positive, neutral and negative with high accuracy. The performance of the proposed method is analyzed with different metrics: accuracy, precision, recall and f1-score. The proposed LWGBM and H2O ML method attains an accuracy of 95.39% on the Internet Movie Data Base (IMDB) dataset, and 92.41% accuracy on SemEval - 2016 dataset, which is more effective than the conventional methods namely, Extra-Long Neural Network (XLNet) and Arabic Bidirectional Encoder Representation Transformer (AraBERT).

Keywords: H2O Automatic Machine Learning, Light Weight - Gradient Boosting Machine, Natural Language Processing, Sentiment Analysis

Introduction

Sentiment analysis, also known as opinion mining, is a key task in Natural Language Processing (NLP). It identifies emotions and opinions expressed in the text (Steinke *et al.*, 2022). This application covers a wide range of features, from analyzing customer feedback to monitoring brands and sentiments on social media (Pradhan *et al.*, 2022). Sentiment analysis plays a crucial role in organizations by aiding data-driven decision-making, understanding public perception, and efficiently responding to emerging sentiments and trends (Alantari *et al.*, 2022; Tesfagergish *et al.*, 2022; Pavitha *et al.*, 2022). Many people utilize social media platforms to express their emotions or sentiments about movies, products, and so on (Srinivasan & Subalalitha, 2023). Sharing text with other users is one of the most common formats. Consumers use these reviews to assess the value of products, movies, and other items (Kora & Mohammed, 2023). This is achieved by identifying words and phrases related to positive or negative sentiments and utilizing Machine Learning (ML) algorithms for classifying reviews (Ishac *et al.*, 2024). The extraction of opinions helps organizations, particularly in the entertainment industry, gain essential insights into audience preferences, enhance marketing strategies, make informed decisions, and improve the overall viewer experience (Danyal *et al.*, 2024a).

Automated Machine Learning (AutoML) refers to the task of automating the process of engineering solutions for a specific problem. This includes selecting and applying prediction techniques to a given dataset (Lin *et al.*, 2023). It involves the integration, parameterization, and selection of ML algorithms as the basic components of a pipeline. This process generates a model through the AutoML tool, which is utilized for training the concrete method on the dataset (Agarwal, 2023). Compared to basic ML algorithms like Support Vector Machine (SVM) that address the learning problem, the AutoML tool solves the learning-to-learn problem (Zulqarnain *et al.*, 2024). For standard problem categories, such as single-label, binary, or multi-class classification and regression, numerous tools have been developed in recent years, showing impressive performance in various experiments (Danyal *et al.*, 2024b; Zhao *et al.*, 2024). Sentiment analysis is challenging due to complex language patterns and inappropriate or redundant features used for classification (Aarthi *et al.*, 2024).

Sentiment analysis is further complicated by complex language structures that involve negations, sarcasm, and context-dependent expressions. Conventional ML algorithms fail to capture these nuances due to feature redundancy, irrelevant feature selection, and overfitting. These issues lead to reduced classification accuracy and poor generalization. The significant contributions of this research are described as follows:

- The TF-IDF-based feature extraction technique is used to convert text or reviews into numerical formats that capture the importance of words in the context of both the document and the corpus.
- The LWGBM-based feature selection method is proposed to choose the relevant and appropriate feature subsets from the extracted features. The feature selection process eliminates irrelevant or inappropriate features to ensure enhanced classification performance.
- The H2O Auto ML-based classification method, which includes six different ML algorithms that allow for automatic hyperparameter tuning and method selection, is employed. This process allows for the selection of the most effective methods for sentiment analysis with high Accuracy.

Literature Review

In recent times, numerous Deep Learning (DL) and Machine Learning (ML) algorithms have been developed for sentiment analysis, demonstrating effective performance. This section analyzes and describes the recent methods, highlighting their advantages and limitations.

Kai Ning Loh *et al.* (2024) presented a hybrid deep learning algorithm that integrated a Masked and Permuted Network (MPNet), Gated Recurrent Unit (GRU), and Bidirectional GRU (Bi-GRU). The MP-Net was a transformer-based pre-trained method that enhanced the understanding of language through permutation and masking. By integrating the benefits of these methods, the presented method provided a more efficient solution for sentiment analysis. However, the proposed model did not calculate feature importance, which resulted in the presence of inappropriate features during classification, which minimized the model's performance.

Danyal *et al.* (2024c) suggested a sentiment analysis method using the Extra-Long Neural Network (XLNet), Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN-LSTM). The XLNet understood word contexts from every slide, enabling the capture of complex language patterns. The LSTM effectively modeled long-term dependencies, while the CNN-LSTM integrated the global and local contexts for effective feature extraction. The suggested method had the capability to extract challenging linguistic patterns and contextual data from the raw textual information. However, the features in the dataset were not fully represented due to limited resources.

Fadel *et al.* (2024) introduced Multi-Task Learning (MTL) that utilized a pre-trained language method called Arabic Bidirectional Encoder Representation Transformer (AraBERT) to extract Arabic aspect terms and classes. Additionally, the introduced method integrated AraBERT, one pair classification, and Bidirectional LSTM (Bi-LSTM) for Aspect term Polarity Classification (APC) and Aspect category Polarity Classification (ACPC). The introduced method successfully performed multiple tasks and data sharing alongside leveraging the interdependency among the tasks. However, the introduced method learned noise and irrelevant features, which led to overfitting.

Aziz *et al.* (2024) implemented a Semantic-Syntactic Dependency Parsing (SSDP) approach that employed both syntactic and semantic data. This method was incorporated with the Core Natural Language Processing (NLP) library to process the input text and identify patterns efficiently. This process extracted complex relationships that accurately reflected the sentiments conveyed toward the opinion target. The outcomes of the implemented method demonstrated that patterns were effectively captured based on the semantic data. However, the implemented method faced difficulties in capturing significant patterns due to the presence of irrelevant and redundant features.

Mendon *et al.* (2021) presented a method for analyzing user sentiments on Twitter during natural disasters by utilizing pre-processing methods, hybrid ML algorithms, statistical modeling, and lexicon-based methods. The Term Frequency-Inverse Document Frequency (TF-IDF) and K-means methods were deployed for classification between hierarchical and affinitive clustering. The Latent Dirichlet Allocation (LDA), Doc2Vec, and K-means were utilized to capture themes with multiple-stage polarities classification and time series analysis. However, the presented method encountered significant challenges in interpretation due to the high number of features used for classification.

From the above analysis, the existing algorithms have the following limitations: Feature importance was not calculated, features in the dataset were not completely represented, struggles with overfitting issues, and difficulties in capturing and interpreting the data. These limitations reduce Accuracy, result in high misclassification rates, and lead to poor generalization. In this proposed methodology, an LWGBM-based feature selection approach is suggested to select the relevant features from the extracted set. This process eliminates inappropriate, irrelevant, or redundant features, which helps minimize overfitting and maximizes classification performance. The LWGBM method calculates the feature importance between the target and key work

Materials and Methods

This research proposes an effective feature selection and classification method for sentiment analysis. The IMDB, SemEval–2016, and World Cup soccer datasets are used and then pre-processed through lemmatization, stemming, and stopword removal approaches. The features are extracted using the Term Frequency–Inverse Document Frequency (TF–IDF) technique, and then the relevant features are chosen using the proposed Light Weight – Gradient Boosting Machine (LWGBM) method. Then, the sentiments are classified by employing the H2O Automated Machine Learning (H2O AutoML) method that classifies the sentiments into positive, neutral, and negative classes. Figure (1) illustrates the process of sentiment analysis.

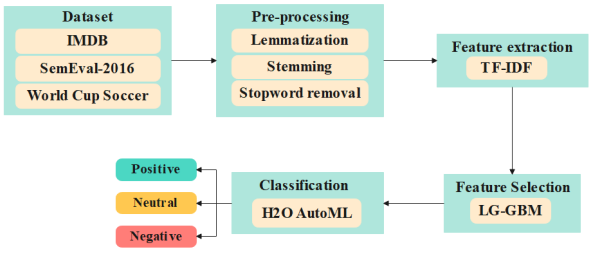

Fig. 1: Sentiment analysis using the proposed method

Dataset

The three datasets used in this research for sentiment analysis are Internet Movie Data Base (IMDB), SemEval-2016 and World Cup Soccer. These datasets respectively contain movie reviews, restaurant reviews and hashtags from Twitter. The detailed explanation of these datasets is given below.

IMDB

The IMDB dataset (IMDB(n.d)) contains 50,000 movie reviews, each annotated with sentiments. This dataset is commonly used in NLP and sentiment analysis, containing a wide range of opinions on various movies. Each review is labeled with either positive or negative sentiments, providing a background for understanding how people felt about the movie. The sentiment classes are evenly distributed, with 25,000 positive and 25,000 negative reviews.

SemEval-2016

The SemEval-2016 dataset (Kaggle, 2016) was presented in scientific competitions of SemEval and is categorized into three classes: Positive, neutral, and negative. The training set consists of 1657 positive reviews, 749 negative reviews, and 101 neutral reviews. This distribution provides a rich dataset with a strong emphasis on sentiments and positive and negative sentiments, offering valuable information for method training. The testing set is composed of 611 positive reviews, 204 negative reviews, and 44 neutral reviews.

World Cup Soccer

Part of the raw data is gathered from Twitter [20] dataset with hashtags such as #brazil2014, #worldcup2014 and game hashtags like #ALGRUS. This dataset focuses on tweets related to the 2014 World Cup soccer tournament held in Brazil, with tweets containing specific hashtags collected. The total number of tweets exceeds 3.5 million, representing a large volume of information that needs to be processed with limited computing resources.

Pre-Processing

The pre-processing techniques used in this research are: Lemmatization, stemming and removal of stopword [23]. The detailed explanation of these techniques is explained as follows.

- Lemmatization – The lemma is the standard form of the lexeme. A lexeme defines the group of all forms with similar meanings, and a lemma is the form selected to describe the lexeme. This process involves minimizing the words to their dictionary or base format (i.e., lemma). For instance, the word "running" becomes "run". This process normalizes the text and makes sure that differences in words are considered as one entity.
- Stemming–Similar to lemmatization, the stemming technique reduced these words to their root format by removing the end of words. For instance, the words “worst” and “worse” are reduced to their root word "worst". This process minimizes the number of features used for minimizing the computational costs of the model.
- Stopword Removal–Stop words are general words (i.e., "is," "the," etc.) which do not contribute much to the text sentiments. Eliminating them enables their minimization of data dimensionality and maximization of the performance of the sentiment analysis.

Feature Extraction

The pre-processed data are provided as input to the feature extraction phase to convert the text into numerical features. The Term Frequency – Inverse Document Frequency (TF-IDF) feature extraction method [23] is used in this research to capture the significance of words in a document by considering their relationship with other documents in a similar corpus. The TF-IDF method analyzes the frequency of words appearing in a document and the frequency with which those words appear in other documents within the corpus. The numerical expression for the TF-IDF method is given in Eq. 1:

$$tfidf\,(w,d,D) = tf\,(w,d) \times idf\,(w,D) \quad (1)$$

In the above Eq. 1, the $f\,(w,d)$ represent the frequency of the word within the document, where w

denotes the word, and D represents the document. The TF-IDF method weights the word's term frequency by inversing its document frequency in the corpus. By extracting the relevant terms and down-weighing general terms, the TF-IDF generates significant and discriminative feature sets. The extracted features are then provided as input to the feature selection phase, where relevant features are selected for classification.

Feature Selection

Amongst numerous feature selection algorithms, the LWGBM method is a robust method for selecting the relevant features. This method handles large, sparse, high-dimensional textual data and offers a reliable feature that is important for ranking for interpretability. The LWGBM method is an improved version of the gradient boosting decision tree, designed to develop boosted trees efficiently and process them in parallel. The objective of the method is to construct trees that are processed effectively to attain feature scores and determine the feature's importance. The method calculates the importance by utilizing "gain," "frequency," and "cover." The gain defines tree branches of feature importance, while frequency defines the number of features in the developed trees, and the cover represents the related value of observed features. The restricted precision and counting trees that attain similar values on all training sets are considered as one or similar trees. Additionally, it is considered that inputs are mapped to $R^{|H|}$ by $\emptyset(x) = \left[h_1(x), \ldots, h_{|H|}(x)\right]^T$. The mathematical expression for learning the linear classifier in the transformed space is given in Eq. 2:

$$\min_{\beta} \sum_{(\phi(x_i),y_i)} l\left(\phi(x_i), y_i, \beta\right) + \lambda|\beta|_1 + \mu q_{\in}(\beta) \quad (2)$$

In the above Eq. 2, β represents the sparse linear vector that selects the trees, $\phi(x_i)$ represents the feature vector, $l\left(\phi(x_i), y_i, \beta\right)$ represents the loss function that calculates the difference between the feature values and actual target y_i. This function minimizes the loss over each data point, while λ represents the regularization parameter, which controls the strength of L1 regularization. Furthermore, considering the generalization loss, the trees in H are arranged such that the initial T of β is non-zero. The mathematical expression of this process is given in Eq. 3:

$$H(x) = \sum_{t=1}^{T} \beta_t h_t(x) \quad (3)$$

In the above Eq. 3, $H(x)$ represents the final feature vector, β_t represents the weights, $h_t(x)$ represents tth extracted feature, and T represents the total number of features. Eq. 2 contains two penalty terms for norm l_1 and capped norm l_1. The initial terms help minimize overfitting, while the following terms focus on feature selection. In the current form, the capped l_1 norm selects trees rather than individual features. The total number of features extracted through the ensemble trees is represented as $F \in \{0,1\}^{d\times T}$, where $F_{ft} = 1$ indicates that the h_t feature utilizes feature f. The mathematical expression for feature f is given in Eq. 4:

$$f = \sum_{t=1}^{T} |F_{ft}\beta_t| \quad (4)$$

In the above Eq. 4, T represents the total number of features, F_{ft} represents the feature value at time t, and β_t represents the weight integrated with feature f at time t. The final mathematical expression for optimization is given in Eq. 5:

$$\min_{\beta} l(\beta) + \lambda|\beta|_1 + \mu\sum_{f=1}^{d} q_{\in}\left(\sum_{t=1}^{T}|F_{ft}\beta_t|\right) \quad (5)$$

In the above Eq. 5, $\min_{\beta}$ denotes the minimization of the fitness function concerning the β vector that represents the weights integrated with features. Furthermore, $l(\beta)$ denotes the loss function that quantifies the difference between actual and optimized features.

Feature Importance

The LWGBM uses the metric of feature importance to retrieve the values of each feature according to its significance after the boosted tree is developed. This scoring technique determines the significance of every feature when making decisions when developing decision trees. Typically, feature importance provides a score that determines the important role of every attribute, and this significance is executed by ranking and comparing every feature in the dataset. The significance of the decision tree is measured by the number of attribute split points, weighted by the number of observations at each node. The split point is used to enhance the effectiveness and performance of the method. Particularly, the Gini Index (purity) is utilized to choose the split points or, alternatively, to find a much more specific error function. The feature significance of each tree is averaged across all decision trees in this method. The LWGBM-based feature selection is employed to transform the extracted features into subsets by utilizing the most promising features. The method's focal point is embedded in pre-processing to minimize training time by eliminating inappropriate features from the extracted features.

Classification Using H2O Auto ML

H2O AutoML is an ML technique that automates the processing and includes the H2O system. It is easy to implement and understand for enterprise environments, generating high-quality models. H2O AutoML supports various types of tasks, such as binary and multi-class

classification, as well as regression problems. In this research, six learning techniques are used for sentiment analysis: Generalized Linear Model (GLM), Feed-Forward Neural Network (FFNN), Gradient Boosting Machine (GBM), Random Forest (RF), Xtreme Gradient Boosting (XGBoost), and LW-GBM.

Generalized Linear Model (GLM)

The GLM involves spatiotemporal filtering of $31 \times 31 \times 30$ patches around an evaluation center in the respective area of the neuron. Here, b represents the bias term, f represents the sigmoid nonlinearity, and the Poisson spike production for generating the response is denoted as r_t. The mathematical expression for GLM is given in Eq. 6:

$$r_t \backsim Poiss\left[f\left(\vec{w}_s^T\left(X_t\vec{w}_t\right)+b+\sum_i h_i r_{t-i}\right)\right] \quad (6)$$

In Eq. 6, h represents the post-spike history filter, and f represents the nonlinearity. The $\backsim Poiss[.]$ represents the variable of Poisson distributed in mean parameter, $\vec{w}_s^T\left(X_t\vec{w}_t\right)$ represents the linear transformation of the input feature vector, X_t represents the feature vector, $\vec{w}_t$ represents the weight parameter which controls the influence of various features, b denotes the bias term, and $\sum_i h_i r_{t-i}$ denotes previous response value. Here, the rank-1 approximation of all spatiotemporal filter methods is used effectively to enhance the subset of analyzed neurons, resulting in a vectorized spatial and temporal filter that spans 250 ms. The algorithms with spike history are suited by initializing the method fit without spike history. The filter processes cover the spikes or generate spikes through the method. Nonlinearity is defined as a logistic sigmoid, which enhances the fitting of exponential nonlinearity to model the RCG responses.

Feed-Forward Neural Network (FFNN)

The FFNN, also known as a multilayer perceptron, consists of input, hidden, and output layers. The neurons in every layer are merged with all neurons in the following layer. Every neuron receives signals from the neuron in the prior layer, producing results in the following layer. All connections among neurons are associated with a real number, which is the weight. Every layer of the neurons is classified through an input layer through an activation threshold, known as the bias. It is assumed that x_i represents the ith output, m represents the number of inputs, and their size is based on the number of features in the input. The o_r denotes the outcome of the rth neuron in the output layer. Here, k represents the number of neurons in the output layer, with their size determined by the number of classes in the input. The number of neurons in the hidden layers has a significant influence on the prediction performance of the FFNN in classification. The mathematical expression for the hidden layer neurons n is given in Eq. 7:

$$n = 2m + 1 \quad (7)$$

Gradient Boosting Machine (GBM)

The GBM is the boosting algorithm that is majorly utilized for regression and classification issues. The GBM has three major factors: Weak learner, loss function, and additive method. The additive method in GBM reduces the loss function by integrating various weak learners that handle the imbalanced data effectively. The aim of boosting is to improve the strength of the method to detect its weaknesses and replace them with powerful learners to generate close, accurate results. The GBM performs tasks by gradually, sequentially, and additively training using various methods.

Random Forest (RF)

The RF, a tree-based ensemble method, is an improved version of the Decision Tree (DT) utilized for handling the issues in supervised learning. The RF integrates various weak learners that offer greatly accurate predictions. By utilizing the various samples of Bootstrap that utilize the bagging method for training several DTs through sub-sampling, the training dataset attains the samples of Bootstrap. The bootstrap samples are similar to the training dataset size. In ensemble classification, two or more classifiers are trained, and their outcomes are integrated by utilizing the stacking process. The mathematical expression for RF is given in Eqs. (8 and 9):

$$p = mode\left\{T_1(y), T_2(y), T_3(y), \ldots, T_m(y)\right\} \quad (8)$$

$$p = mode\left\{\sum_{m=1}^{m} T_m(y)\right\} \quad (9)$$

In the majority of classification tasks, the Gini index is utilized as the cost function to estimate the split dataset. The mathematical expression for the Gini index is given in Eq. 10:

$$Gini = 1 - \sum_{i=1}^{classes} p\left(\frac{i}{t}\right)^2 \quad (10)$$

In Eq. 10, the $Gini$ represents Gini Index and the $\sum_{i=1}^{classes} p\left(\frac{i}{t}\right)^2$ denotes the addition of squared probabilities for all classes.

Xtreme Gradient Boosting (XGBoost)

The XGBoost technique is a type of Gradient Boosting Decision Tree (GBDT) used for both regression and classification tasks. GBM is an ensemble learning technique that integrates a group of weak classifiers to develop a strong classifier. GBM attempts to correct the residuals of each weak learner by adding new weak learners. Ultimately, multiple learners are added to make the final prediction, which improves Accuracy compared

to using a single learner. This is known as GBM, which uses the gradient descent method to minimize the training loss while incorporating new models. Typically, gradient boosting is slow due to the need to develop and add trees to the entire method sequence each time. XGBoost, however, is a classifier that offers fast calculation speed and performance. The mathematical expression for the objective function of XGBoost is separated into the loss function and the regularization term, as given in Eq. 11:

$$Obj\left(\circledast\right) = L\left(\circledast\right) + \Omega\left(\circledast\right) \tag{11}$$

In Eq. 11, $Obj\left(\circledast\right)$ represents the objective function, $L\left(\circledast\right)$ represents the loss function, and $\Omega\left(\circledast\right)$ represents the regularization term. The loss function is well-suited to this method, incorporating the regularization term that penalizes complex methods and encourages simpler methods.

Light GBM

The Light GBM offers a quick and reliable gradient-boosting performance based on the DT method deployed for classification, ranking, and various ML tasks. The Light GBM is an ensemble technique that integrates the predictions of numerous DTs to generate the last prediction and generalize it effectively. This method trains numerous tree methods in an additive manner, with every tree method trained for predicting the residuals of previous methods. Hence, the Light GBM method with T tress is developed, and the mathematical expression for the additive training process is given in Eq. 12:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k\left(x_i\right) = \hat{y}_i^{(t-1)} + f_t\left(x_i\right) \tag{12}$$

In the above Eq. 12, $\hat{y}_i^{(t)}$ represents the ith sample and $t-th$ iteration, and f_t represents the learned function. In every iteration, the present method $\hat{y}_i$ has an addition function f. The mathematical formula for f's of every iteration is learned by reducing, as given in Eq. 13:

$$L^{(t)} = \sum_i^n l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{t=1}^{T} \Omega\left(f_t\right) \tag{13}$$

Table 1: Hyperparameters of each model

Method	Hyperparameters
GLM	Regularization – L1/L2, Link function – Logit
FFNN	Hidden layers [128, 64], Activation function – ReLU, Optimizer – Adam, and 0.001 learning rate
GBM	Number of trees – 100, 0.1 learning rate, 6 Max depth and 0.8 subsampling
RF	Number of trees – 200, 10 Max depth.
XGBoost	Number of trees – 150, 0.05 learning rate, 8 Max depth.
LGBM	Number of leaves – 31, 0.05 learning rate, 0.8 feature fraction.

In the above Eq. 13, $L^{(t)}$ represents the loss function that calculates the difference among target y_i and prediction $y_i^{(t)}$, while $\sum_{t=1}^{T} \Omega\left(f_t\right)$ represents the regularization term that penalizes the method's complexity. The Table (1) represents the hyperparameters of each model.

Stacked H2O Auto ML

The stacked ensemble learning method H2O is a supervised learning technique utilized for determining the optimum combinations from numerous classification techniques. The procedure for identifying an optimum combination from numerous classification techniques is known as stacking. This stacking method helps all types of issues, including binary or multi-class classification, and supports regression issues. This research utilizes the RF classifier as the base and GBM as the meta-estimator for classifying sentiment analysis. It includes the classification algorithms of GLM, FFNN, GBM, RF, XGBoost, and Light GBM. The highest target value is selected as the result of the algorithm by combining the advantages of the integrated method, which offers significant results. Additionally, the Decision Function (DF) is used to make correct sentiment predictions, and the mathematical expression is given as Eq. 14:

$$DF = \max\left[\frac{1}{N_C}\sum_{cls}\begin{matrix}\left(Average_{(Pb_{DL}-P, Pb_{ML}-P)}\right)\\ \left(Average_{(Pb_{DL}-N, Pb_{ML}-N)}\right)\\ \left(Average_{(Pb_{DL}-Neu, Pb_{ML}-Neu)}\right)\end{matrix}\right] \tag{14}$$

In the above Eq. 14, N_C represents the number of classifiers, Pb_{DL} and Pb_{ML} denote the probability values of DL and ML algorithms, respectively. The P and N denote the positive and negative feedbacks, respectively and Neu represents the neutral feedback. Based on the mean possibility value, the final classification is performed using the proposed method, rendering high accuracy.

Experimental Analysis

The performance of the proposed method is simulated in a Python environment with the required system configurations being i5 processor, windows 10 OS, and 16 GB RAM. The performance metrics considered in this research for analysis of the proposed method are Accuracy, precision, recall, and f1-score. The mathematical expressions for performance metrics are given in Eqs. (15-18):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

$$Precision = \frac{TP}{TP+FP} \tag{16}$$

$$Recall = \frac{TP}{TP+FN} \tag{17}$$

$$F1-score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{18}$$

Accuracy is measured by dividing the number of predicted reviews by the whole number of reviews, while precision is measured by dividing the number of reviews that are accurately predicted as positive by the total

number of reviews predicted as positive. The recall is calculated by dividing the number of accurately predicted positive reviews by the total number of positive reviews, while the F1-score measures the method's performance by considering both precision and recall. In the above equations, TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative.

Evaluation of the IMDB Dataset

The performance of the proposed method is analyzed on three different datasets: IMDB, SemEval and World Cup Soccer. The classifier's performance is evaluated based on actual features after selecting features with different performance metrics of Accuracy, precision, recall, and f1-score. The existing methods considered in this research are GBM, RF, GLM, FFNN, XGBoost and LW-GBM.

Table 2: Performance of classifier on IMDB dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
With actual features				
GBM	81.13	71.93	74.85	72.11
RF	84.29	75.29	79.08	75.82
GLM	87.61	80.83	83.33	81.79
FFNN	91.81	86.55	86.75	86.91
XGboost	83.03	90.95	92.03	91.82
Light GBM	86.10	86.66	87.37	86.63
Proposed	91.81	90.95	92.03	91.82
With selected features				
GBM	81.39	77.14	77.98	76.82
RF	84.89	81.13	83.69	81.47
GLM	89.53	86.75	88.39	87.10
FFNN	95.39	92.28	93.43	90.30
XGboost	86.24	95.60	97.23	96.20
Light GBM	91.27	92.23	93.94	90.83
Proposed	95.39	95.62	97.23	96.24

Table 2 presents the outcomes of the performance of the proposed classifier on the IMDB dataset, which is evaluated based on different performance metrics. The performance of the classifier is analyzed based on the actual and selected features on the IDMB dataset. The proposed classifier attains 91.81% accuracy, 90.95% precision, 92.03% recall, and 91.82% f1-score on the actual features. By using the based feature selection method, appropriate features are selected by eliminating the irrelevant or inappropriate features from the feature subset. This process improves the classifier's sentiment analysis performance, offering high Accuracy on the IMDB dataset. After selecting the relevant features, the performance of the classifier improves with 95.39% accuracy, 95.62% precision, 97.23% recall, and 96.24% f1-score, proving more efficient than the existing algorithms. Figures (2-4) show the graphical representation of classifiers on the IMDB, SemEval–2016, and World Cup Soccer datasets, with the selected features. The AutoML optimizes the hyperparameters, offers good feature selection, and performs model tuning. This integration of multiple models enhances generalization, and the stacked ensemble method reduces overfitting compared to single methods.

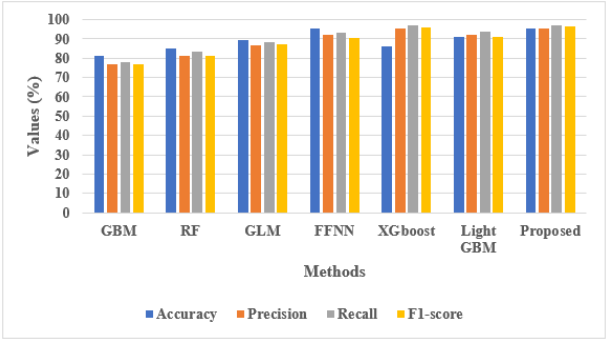

Fig. 2: Graphical representation of classifiers on IMDB dataset with selected features

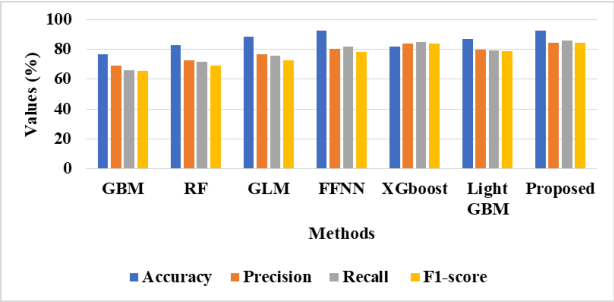

Fig. 3: Graphical representation of classifiers on SemEval – 2016 dataset with selected features

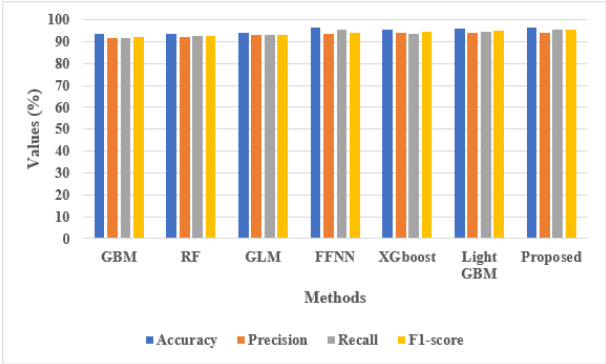

Fig. 4: Graphical representation of classifiers on the World Cup Soccer dataset with selected features

Evaluation of SemEval-2016 Dataset

Table 3 presents the performance of the proposed classifier on the SemEval-2016 dataset based on different performance metrics. The performance of the classifier is analyzed on both the actual and selected features on the SemEval-2916 dataset. The proposed classifier attains 87.40% accuracy, 80.00% precision, 80.14% recall, and 79.13% f1-score on the actual features. After selecting the relevant features, the classifier exhibits an improved performance on the SemEval-2016 dataset with 92.41%

accuracy, 84.35% precision, 85.72% recall, and 84.51% f1-score, proving its efficiency over the existing algorithms.

Table 3: Performance of classifier on SemEval – 2016 dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
With actual features				
GBM	75.36	61.61	64.13	60.86
RF	78.47	66.50	68.01	65.02
GLM	84.26	69.50	71.55	70.43
FFNN	87.40	74.80	75.54	74.68
XGboost	76.09	80.00	80.14	79.13
Light GBM	81.40	76.65	75.09	74.51
Proposed	87.40	80.00	80.14	79.13
With selected features				
GBM	76.56	68.89	66.01	65.70
RF	82.56	72.61	71.53	69.26
GLM	88.25	76.69	75.90	72.41
FFNN	92.40	80.38	81.62	78.36
XGboost	81.58	84.00	85.00	84.00
Light GBM	86.69	79.65	79.15	78.93
Proposed	92.41	84.35	85.72	84.51

Evaluation of World Cup Soccer Dataset

Table 4 displays the performance outcomes of the proposed classifier on the World Cup Soccer dataset based on different performance metrics. The performance of the classifier is analyzed based on the actual and selected features on the World Cup Soccer dataset. The proposed classifier attains 92.60% accuracy, 90.61% precision, 91.95% recall, and 91.79% f1-score on actual features. After selecting the relevant features, the performance of the classifier is enhanced on the World Cup Soccer dataset with 96.54% accuracy, 94.24% precision, 95.25% recall, and 95.37% f1-score, demonstrating greater efficiency than existing algorithms.

Table 4: Performance of classifier on World Cup Soccer dataset

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
With actual features				
GBM	90.23	86.48	87.08	87.99
RF	87.81	87.34	87.22	88.64
GLM	88.82	87.24	91.95	89.67
FFNN	90.35	90.61	90.25	89.76
XGboost	92.60	88.37	89.51	90.71
Light GBM	90.19	88.35	89.89	91.47
Proposed	92.60	90.61	91.95	91.79
With selected features				
GBM	93.40	91.58	91.58	91.99
RF	93.72	92.27	92.55	92.70
GLM	94.06	93.08	92.94	93.22
FFNN	96.54	93.45	95.25	94.03
XGboost	95.40	94.24	93.50	94.60
Light GBM	95.86	94.13	94.37	95.08
Proposed	96.54	94.24	95.25	95.37

To validate that the observed accuracy improvements are statistically significant, paired t-tests are conducted, comparing LWGBM + H2O AutoML against other classifiers. Table 5 below shows that the p-values are above 0.05, which clearly indicates that the improvements are not due to random variations.

Table 5: Evaluation of Statistical Analysis

Methods	p-value (t-test)
LWGBM+H2O Auto ML vs. GBM	0.013
LWGBM+H2O Auto ML vs. XGBoost	0.007
LWGBM+H2O Auto ML vs FFNN	0.016

Comparative Analysis

In this section, the performance of the proposed classifier is compared with that of the existing algorithms: MPNet-GRUs [16], XLNet [17] on the IMDB dataset and MTL-AraBERT [18], SSDP [19] on the SemEval-2016 dataset. By using the LWGBM-based feature selection method, relevant features are chosen by eliminating the inappropriate features that increase the classification performance. The proposed classifier accomplishes 95.39% accuracy, 95.62% precision, 97.23% recall, and 96.24% f1-score, exhibiting superior performance than the existing methods, namely, MPNet-GRUs [16], XLNet [17] on the IMDB dataset. Then, on SemEval–2016 dataset, the model accomplishes 92.41% accuracy, 84.35% precision, 85.72% recall, and 84.51% f1-score, displaying superior performance than the existing methods such as MTL-AraBERT [18], SSDP [19] on SemEval–2016 dataset. Table 6 displays the comparative analysis of the proposed classifier.

Table 6: Comparative Analysis of the proposed classifier

Datasets	Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
IMDB	MPNet-GRUs (Kai Ning Loh *et al.*, 2024)	94.71	95	95	95
	XLNet (Danyal *et al.*, 2024c)	93.74	91.48	96.54	93.94
	Proposed LWGBM and H2O AutoML	95.39	95.6	97.23	96.2
SemEval - 2016	MTL-AraBERT (Fadel *et al.*, 2024)	NA	80.96	79.70	80.32
	SSDP (Aziz *et al.*, 2024)	90.42	NA	NA	82.68
	Proposed LWGBM and H2O AutoML	92.41	84.35	85.72	84.51

Discussion

The MPNet-GRUs (Kai Ning Loh *et al.*, 2024) method does not calculate feature importance. The XLNet (Danyal *et al.*, 2024c) method fails to fully

represent the features in the dataset. The MTL-AraBERT (Fadel *et al.*, 2024) method suffers from overfitting, while the SSDP (Aziz *et al.*, 2024) method faces difficulties in feature interpretation. In this proposed methodology, LWGBM-based feature selection is introduced to select the relevant features from the extracted set. This process eliminates inappropriate, irrelevant, or redundant features, thereby reducing the overfitting issue and maximizing classification performance. The LWGBM method calculates feature importance between the target and key feature values, further improving feature selection performance. The proposed method achieves an accuracy of 95.39% on the IMDB dataset and 92.41% accuracy on the SemEval 2016 dataset. The LWGBM+H2O AutoML method outperforms classical methods due to enhanced feature selection and ensemble learning. LWGBM removes redundant and noisy features, leading to a 5% accuracy improvement over conventional algorithms. Additionally, H2O AutoML boosts performance by stacking multiple methods, ensuring robustness across various datasets.

Conclusion

This research proposes effective feature selection-based LWGBM and classification-based H2O ML methods are proposed for sentiment analysis. The LWGBM-based feature selection is proposed for the optimal selection of relevant features for classification, which eliminates the inappropriate or redundant features and learns the complex language patterns. Then, the classification is performed using the H2O ML algorithm, which classifies the sentiments as positive, neutral, and negative. The datasets used in the research to evaluate the proposed method are IMDB, SemEval-2016, and World Cup Soccer. The proposed LWGBM and H2O ML method achieves a commendable accuracy of 95.39% on the IMDB dataset and 92.41% accuracy on the SemEval–2016 dataset. This proves the proposed method's superiority over conventional methods, namely, XLNet and AraBERT. In the future, the meta-heuristic optimization-based feature selection can be used to further improve the performance of sentiment analysis.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work, and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

The authors have not received any financial support or funding to report.

Author Contributions

Bikku Ramavath: Conceptualization, Methodology, Software, Field study

Srikanth Kadainti: Data curation, Writing-Original draft preparation, Software, Validation, Field study

Nemani Subash: Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Aarthi, E., Jagan, S., Devi, C. P., Gracewell, J. J., Choubey, S. B., Choubey, A., & Gopalakrishnan, S. (2024). A turbulent flow optimized deep fused ensemble model (TFO-DFE) for sentiment analysis using social corpus data. *Social Network Analysis and Mining*, *14*(1), 41. https://doi.org/10.1007/s13278-024-01203-2

Agarwal, B. (2023). Financial sentiment analysis model utilizing knowledge-base and domain-specific representation. *Multimedia Tools and Applications*, *82*(6), 8899-8920. https://doi.org/10.1007/s11042-022-12181-y

Alantari, H. J., Currim, I. S., Deng, Y., & Singh, S. (2022). An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews. *International Journal of Research in Marketing*, *39*(1), 1-19. https://doi.org/10.1016/j.ijresmar.2021.10.011

Aziz, M. M., Bakar, A. A., & Yaakub, M. R. (2024). CoreNLP dependency parsing and pattern identification for enhanced opinion mining in aspect-based sentiment analysis. *Journal of King Saud University - Computer and Information Sciences*, *36*(4), 102035. https://doi.org/10.1016/j.jksuci.2024.102035

Danyal, M. M., Haseeb, M., Khan, B., Ullah, S., & Khan, S. S. (2024a). Opinion Mining on Movie Reviews Based on Deep Learning Models. *Journal on Artificial Intelligence*, *6*(1), 23-42. https://doi.org/10.32604/jai.2023.045617

Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Ghaffar, M. B., & Khan, W. (2024b). Sentiment analysis of movie reviews based on NB approaches using TF-IDF and count vectorizer. *Social Network Analysis and Mining*, *14*(1), 87. https://doi.org/10.1007/s13278-024-01250-9

Danyal, M. M., Khan, S. S., Khan, M., Ullah, S., Mehmood, F., & Ali, I. (2024c). Proposing sentiment analysis model based on BERT and XLNet for movie reviews. *Multimedia Tools and Applications*, *83*(24), 64315-64339. https://doi.org/10.1007/s11042-024-18156-5

Fadel, A., Saleh, M., Salama, R., & Abulnaja, O. (2024). MTL-AraBERT: An Enhanced Multi-Task Learning Model for Arabic Aspect-Based Sentiment Analysis. *Computers*, *13*(4), 98. https://doi.org/10.3390/computers13040098

IMDB. (n.d.). IMDB dataset. *IMDB Dataset*. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Ishac, W., Javani, V., & Youssef, D. (2024). Leveraging sentiment analysis of Arabic Tweets for the 2022 FIFA World Cup insights, incorporating the gulf region. *Managing Sport and Leisure*, 1-17. https://doi.org/10.1080/23750472.2024.2342258

kaggle. (2016). *SemEval-datasets*. https://www.kaggle.com/datasets/azzouza2018/semevaldatadets

Kai Ning Loh, N., Poo Lee, C., Song Ong, T., & Ming Lim, K. (2024). MPNet-GRUs: Sentiment Analysis With Masked and Permuted Pre-Training for Language Understanding and Gated Recurrent Units. In *IEEE Access* (Vol. 12, pp. 74069-74080). https://doi.org/10.1109/access.2024.3394930

Kora, R., & Mohammed, A. (2023). An enhanced approach for sentiment analysis based on meta-ensemble deep learning. In *Social Network Analysis and Mining* (Vol. 13, Issue 1, p. 38). https://doi.org/10.1007/s13278-023-01043-6

Lin, T., Sun, A., & Wang, Y. (2023). EDU-Capsule: aspect-based sentiment analysis at clause level. In *Knowledge and Information Systems* (Vol. 65, Issue 2, pp. 517-541). https://doi.org/10.1007/s10115-022-01797-z

Mendon, S., Dutta, P., Behl, A., & Lessmann, S. (2021). A Hybrid Approach of Machine Learning and Lexicons to Sentiment Analysis: Enhanced Insights from Twitter Data of Natural Disasters. *Information Systems Frontiers*, *23*(5), 1145-1168. https://doi.org/10.1007/s10796-021-10107-x

Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., Patel, A., & Shashidhar, R. (2022). Movie Recommendation and Sentiment Analysis Using Machine Learning. *Global Transitions Proceedings*, *3*(1), 279-284. https://doi.org/10.1016/j.gltp.2022.03.012

Pradhan, A., Senapati, M. R., & Sahu, P. K. (2022). Improving sentiment analysis with learning concepts from concept, patterns lexicons and negations. *Ain Shams Engineering Journal*, *13*(2), 101559. https://doi.org/10.1016/j.asej.2021.08.004

Srinivasan, R., & Subalalitha, C. N. (2023). Sentimental analysis from imbalanced code-mixed data using machine learning approaches. *Distributed and Parallel Databases*, *41*(1-2), 37-52. https://doi.org/10.1007/s10619-021-07331-4

Steinke, I., Wier, J., Simon, L., & Seetan, R. (2022). Sentiment Analysis of Online Movie Reviews using Machine Learning. *International Journal of Advanced Computer Science and Applications*, *13*(9). https://doi.org/10.14569/ijacsa.2022.0130973

Tesfagergish, S. G., Kapočiūtė-Dzikienė, J., & Damaševičius, R. (2022). Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. *Applied Sciences*, *12*(17), 8662. https://doi.org/10.3390/app12178662

Zhao, C., Sun, X., & Feng, R. (2024). Multi-strategy text data augmentation for enhanced aspect-based sentiment analysis in resource-limited scenarios. *The Journal of Supercomputing*, *80*(8), 11129-11148. https://doi.org/10.1007/s11227-023-05864-2

Zulqarnain, M., Ghazali, R., Aamir, M., & Hassim, Y. M. M. (2024). An efficient two-state GRU based on feature attention mechanism for sentiment analysis. *Multimedia Tools and Applications*, *83*(1), 3085-3110. https://doi.org/10.1007/s11042-022-13339-4