ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20231004000989
Original scientific paper

Sentiment Analysis on Big Data: A Hybrid SED-TABU Feature Selection Method

Sabitha RAJAGOPAL, Sreemathy JAYAPRAKASH*, Karthik SUBBURATHINAM

Abstract: Big data mining is a crucial component of contemporary decision support systems linked to social networks and other data sources. Sentiment Analysis (SA) is the process by which text analytics is used to mine many data sources for opinions. This research seeks to create a feature selection method for sentiment analysis that is efficient and robust against noise and high dimensionality in Big data environments. The objective is to choose a condensed collection of useful features that increases sentiment categorization precision. It is suggested to use a novel hybrid feature selection method that combines Tabu Search (TS) and Stream Evolution Dynamics (SED). SED offers exploratory power, and TS offers exploitation. The classifier assesses the performance for each feature subset that SED-TS chose. Instances are classified using the AdaBoost classifier. The suggested method was assessed using data from Amazon product reviews. As a result, our technique outperforms wrapper and filter-based feature selection methods. By extracting a small feature subset, the SED-TS hybrid technique attained the best accuracy of 93% and an *F*1 score of 0.95. The work effectively combined SED and TS for feature selection specifically suited to sentiment analysis on Big data. The hybrid strategy offers higher accuracy and better generalization by utilizing the complementing characteristics of the two strategies. This shows how metaheuristic approaches can be used to classify sentiment in high-dimensional noisy data.

Keywords: adaboost classifier; Big data; feature selection; sentiment analysis (SA); stream evolution dynamics (SED); tabu search (TS)

1 INTRODUCTION

The collection of unstructured datasets gathered from numerous sources such as the Internet, cloud drives, social media, and so on is referred to as Big data. This has been made possible due to the extensive distribution of information technologies to all aspects of society, the Internet's growth and usage, and the provision of the Internet's diverse service types. Big data also refers to the data's exponential growth as a result of the utilization of various digital technology applications in transportation, public administration, astrology, bioinformatics, and health [1-5]. An assembly of huge datasets obtained from diverse domains [6] is another definition for Big data. In Big data, there is an immense data amount comprised of datasets, which are unstructured, semi-structured, and structured. These datasets increase the data's complexity and make it hard to process the data with conventional data processing methods. Summarization of the Big data's attributes [7]. The tremendous data amount which is generated every second is known as volume. The rate at which data is generated and processed to meet requirements is called velocity. The data's trustworthiness or messiness is known as veracity. The capability to transform the data into value is known as value. The comprehensive range of data types, as well as data sources, is known as variety. Veracity is a matter of validity, signifying that the data has been correctly identified and absolute. Data with meanings that constantly change is known as variability. The duration of the data's validity is known as volatility. Big data's toughest aspect, which is related to making all the huge data amounts easily comprehensible as well as easily visualizable, is known as visualization. The MapReduce framework's [8] open-source implementation is the Hadoop software library, which facilitates distributed and parallel processing of large datasets. In addition to that, it has a provision for distributed storage on a cluster of computers. MapReduce and Hadoop Distributed File System (HDFS) constitute the Hadoop core. The HDFS's responsibility is to store huge datasets on the cluster that have been fragmented into blocks and distributed into nodes. Distributed processing across a cluster's many nodes is facilitated by the MapReduce model. This model is comprised of a map and a reduce function procedure, which is respectively referred to as a mapper and reducer. There is the partitioning of the input data into the mapper phase. Later, it is transferred to the workers for the map function's execution. After data processing, the outputs of every worker will be in key-value pairs. The shuffle phase will sort the output and categorize it by the key. The reducer will call for all unique keys and will obtain a value set that is related to the key. MapReduce framework will manage the adjustment of loads, distribution of information amongst the nodes, adjustment to internal failure as well and intrinsic parallelization. Both reliability, as well as accessibility, are enhanced with the data's replication and distribution over the nodes. Sentiment Analysis (SA) is a type of Natural Language Processing (NLP) that can be utilized to track public opinion on a particular topic or product. Also referred to as opinion mining, SA [9] constitutes constructing a system for the collection and examination of product-related opinions from reviews, tweets, blog posts, or comments. There are multiple applications of SA. For instance, in the marketing field, SA can judge a new product launch/ad campaign's success for the determination of the popular services or product versions as well as for identification of the demographics that would like/dislike a specific feature. Document classification and its polarity determination are the SA's major jobs. Neutral, negative, or positive are the three distinct expressions of polarity. SA performance [10] can be conducted at three distinct levels: Document-level: The entire document is classified as neutral, negative, or positive. This is typically called the document-level sentiment classification; Sentence level: The sentences are classified as neutral, negative, or positive. This is generally termed the sentence-level sentiment classification; Aspect & Feature level: The documents/sentences are classified as neutral, negative, or positive, depending on their aspects.This is often referred to as aspect-level sentiment classification. At the time of the classification, nature-inspired algorithms have put a lot of focus on accuracy. Loss of diversity and outdated memory are the two critical issues that affect

some of these algorithms [11]. Despite that, enhanced approaches such as SED-Tabu with AdaBoost can be utilized to overcome these issues by increasing the system performance with regard to accuracy. Sentiment analysis on big social data is critical for applications like recommendation systems but suffers from noisy, low-quality data [12]. Feature selection is key to improving sentiment classification by removing irrelevant features [13]. However, many current methods like IG, ReliefF, and wrappers have limitations in computational expense or ignoring feature dependencies [14]. The objective of this work is to develop an efficient hybrid feature selection technique that balances exploration and exploitation for enhanced sentiment analysis. The proposed solution hybridizes the explorative Stream Evolution Dynamics (SED) with exploitative Tabu Search (TS) to leverage their complementary strengths. Unlike existing hybrids, SED provides stochastic search to avoid local optima while TS uses adaptive memory for intensification [15]. The integration with AdaBoost further improves sentiment classification performance. Experiments on benchmark datasets demonstrate up to 5% better accuracy than current techniques. In summary, this work aims to advance sentiment analysis on Big data by proposing a novel SED-TS feature selection technique that outperforms existing hybrids. The hybrid approach balances explorative and exploitative search to find optimal feature subsets. The remaining portions of the inquiry have been organized as follows. In Section Two, the related published works are listed. The work's multiple utilized techniques are described in Section Three. The outcomes of the experimentations are detailed in Section Four, while the work's conclusions are provided in Section Five.

2 RELATED WORKS

Rodrigues & Chiplunkar [16] presented the Hybrid Lexicon-Naive Bayesian Classifier (HL-NBC) approach for sentimental analysis. Moreover, there was a proposal for an SA engine preceded by topic classification that could classify tweets into diverse categories as well as filter irrelevant tweets. This proposed technique was compared against Lexicon and Naïve Bayesian Classifier (NBC) for unigram and bi-gram features. When compared against various other techniques, the proposed HL-NBC technique could improve sentiment classification as well as offer an accuracy of 82%. In addition to that, the SA was executed at a time that was shorter in comparison to conventional techniques, and for larger datasets, it accomplished a 93% improvement in processing time. Liu [17] proposed a text SA method that used deep learning with the Bag of Words (CBOW) language model to address problems with accurate and timely SA of text comments in network Big data environments. At first, the text's vector representation is built using a CBOW language model, which is dependent on feed-forward neural networks. Afterward, there is the training of the Convolutional Neural Network (CNN) through the labeled training set to capture the text's semantic features. Eventually, there is the introduction of the Dropout strategy in conventional CNN's Softmax classifier for effective prevention of the model's over-fitting as well as to gain better classification capability. Experimental outcomes have demonstrated that this technique's robustness as well as its ability to accurately determine the text's emotional category with an accuracy of 90.5% for the COAE2014 dataset and 87.2% for the IMDB dataset. Hammou et al. [18] developed an SA technique that focused on the use of rapid text with Recurrent Neural Network (RNN) versions for efficient textual data representation. This technique would later utilize these new representations to execute the task of classification. This proposed technique's key goals are enhancement of the renowned RNN variants' performances with regard to classification accuracy as well as management of large-scale data. This work also proposes a distributed intelligent framework for social Big data analytics. This proposed system's purpose is the real-time ingestion, storage, processing, indexing, and visualization of an immense amount of information. For the decision-making procedures' enhancement, the proposed system has adopted distributed machine learning along with the proposed technique. This work is quite beneficial for practitioners and researchers who want the real-time collection, management, analysis, and visualization of multiple information sources. This proposed method, together with updated versions of the most powerful distributed deep learning and machine learning algorithms, can provide a deeper knowledge of user behavior and public opinion. Furthermore, SA can improve the classification accuracy of several previous RNN model-based efforts. A Twitter SA was presented by Rahmani [19], who collected the tweets and utilized numerous machine-learning approaches for the assessment of these tweets' sentiments. This paper uses correlation-based feature selection (CFS) to evaluate sentiment categorization using Twitter data. There is real-time extraction of data from Twitter. Then, there is text pre-processing followed by the application of feature extraction on the textual data. There is the utilization of correlation-based attribute selection techniques. Comparisons based on several performance metrics of Machine Learning (ML) classifiers (such as SVM, Naïve Bayes, Random Forest (RF), Meta classifier, Stochastic Gradient Descent (SGD), and Logistic Regression) are done to demonstrate which classifier provides better outcomes. Within the same set-up, the outcomes have demonstrated that, for the combination of StringToWordVector (STWV) with Attribute Selection techniques, the classifiers provide accuracy between 78 and 88% with about 0.88 true positive rates and 0.15 false positive rate, which is much better in comparison to non-utilization of any attribute selection technique. Sharma & Jain [20] proposed an ensemble learning approach for sentiment categorization of social media data, along with an empirical assessment of several ensemble features and classifiers. Twitter API was employed for the real-time collection of the data from Twitter.Afterward, the textual data was applied with text pre-processing as well as ranking-based feature selection. There was a presentation of a hybrid ensemble learning model's framework. This model made use of an implementation that blended ensemble classifiers such as AdaBoost with Sequential Minimal Optimization (SMO)-SVM and Logistic Regression with ensemble features, namely Information Gain (IG) and CHI-Squared. SA is employed as a feature for categorizing Twitter data. The proposed model has an

accuracy of 88.2% and a low error rate when compared to sophisticated approaches. Numerous machine-learning approaches have been utilized in a lot of research for the analysis of sentiments. Despite that, the complete system's efficiency is reduced by these research error rates. To address this problem, a unique Big data and ML method for the evaluation of the SA procedures was proposed by Liu et al. [21] and Alarifi et al. [22]. There is data collection from a tremendous dataset volume. This is beneficial for the systems' efficient analysis. A pre-processing data mining idea is utilized for the elimination of the data's noise. Using a Greedy technique, which selects the best features to be processed by an ideal classifier called the Cat Swarm Optimization-based Long Short-Term Memory Neural Network (CSO-LSTMNN), an efficient feature set is obtained from this cleansed sentiment data. The classifiers use the behavior of cats to examine the sentiment-related features. The minimization of the error rate occurs during the examination of features. From the analysis of this approach's experimental outcomes in terms of accuracy, recall, precision, and error rate, it has been observed that this approach has aided in the enhancement of system efficiency. A novel Big data classification framework was propounded by Hassib et al., [23], which comprised three developed phases. The feature selection phase was the initial phase in which the Whale Optimization Algorithm (WOA) was employed for the identification of the best feature set. The pre-processing phase was the second phase in which the SMOTE algorithm and the Locality Sensitive Hashing (LSH)-SMOTE algorithm were utilized for the class imbalance problem's resolution. The WOA+Bidirectional Recurrent Neural Network (BRNN) algorithm, which used WOA for the first time to train a deep learning method known as BRNN, was the last, third stage. The proposed WOA +BRNN algorithm's experimental outcomes have shown that it has accomplished favorable accuracy as well as high local optima avoidance, and, with regard to Area Under Curve (AUC), has also surpassed the performances of the four most frequently utilized machine learning algorithms, and the Multi-Layer Perceptron utilizing Gray Wolf Optimizer (GWO-MLP). The whale hunting mechanism is replicated in WOA, which is among the latest metaheuristic optimization algorithms. In spite of this, WOA exhibits the same problem as many other optimization algorithms and has a propensity to enter local optima. Liu et al. [24] proposed two improvements to the WOA algorithm in order to address these problems. Using Elite Opposition-Based Learning (EOBL) during the WOA's initiation phase was the first improvement. The second improvement was adding evolutionary operators-mutation, crossover, and selection-to the Differential Evolution (DE) algorithm at the conclusion of each WOA iteration. Moreover, Information Gain (IG) was used in this work as a filter features selection strategy with WOA using an SVM classifier in order to reduce the search space that is explored by WOA. Four Arabic benchmark datasets were used in the verification of the proposed algorithm for SA. This is due to the fact that there was significantly less research conducted in South Africa on the Arabic language than there was on the English language. The results of the extensive experiments clearly show that the suggested approach, which finds the optimal solutions while lowering the amount of selected features, outperforms all other algorithms in terms of SA classification accuracy.

3 METHODOLOGY

The feature selection problem in the classification can be formulated as a combinatorial optimization problem for two main reasons [23]. To create a total of 2N possible subsets, it first entails choosing a feature subset from a set of N features. The performance of a classification model created using a subset is frequently used to evaluate the quality of a subset. Unfortunately, using a sophisticated classifier to create a model can take a lot of time, which makes using optimization techniques difficult when dealing with Big datasets. In this section, feature selection methods based on Tabu Search (TS), SED-AdaBoost, TS-AdaBoost, and SED-TS AdaBoost are covered. The overall process of investigation is shown in Fig. 1.

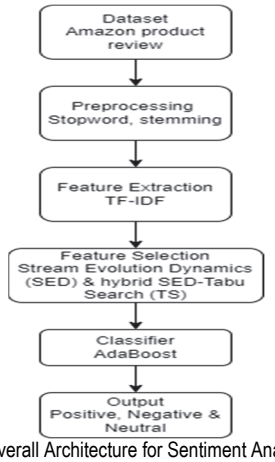

Figure 1 Overall Architecture for Sentiment Analysis

The Amazon review dataset is used for evaluating the methods. The reviews are preprocessed using stop words and Stemming. The reviews' features (text, ratings, and votes) are extracted using Term Frequency-Inverse Document Frequency (TF-IDF) links at the end, along with product metadata (picture features, brand, price, category information, and descriptions). The Amazon dataset is extensively used as a benchmark in sentiment analysis, owing to its scale, diversity, and authenticity of real-world opinions. In our experiments, we focused solely on review text and sentiment labels. Text preprocessing involved lowercase conversion, punctuation removal, stop word elimination, and stemming. This streamlines vocabulary and extracts meaningful features. Textual features from unigrams and bigrams were retrieved using Term Frequency-Inverse Document Frequency (TF-IDF) weighting after preprocessing.

3.1 Feature Selection using Stream Evolution Dynamics (SED) Algorithm

Feature selection that utilized the SED algorithm was earlier employed for the resolution of Non-deterministic Polynomial (NP)-hard problems like path detection in graphs (for example, the problem of the traveling salesman). The application of this kind of heuristic technique is justified as the problem of a traffic sign's path

detection can be turned into a graph problem within specific conditions. This algorithm can optimize the path length as well as take into consideration additional dependencies, like its travel time and usage of vehicular energy. The task of path detection with time, velocity, and acceleration constraints among polyhedral obstacles is of great significance and is known in the literature as an NP-hard problem [24]. The following is the SED algorithm's overview. Every node will be allocated an amount of soil. Subsequently, the movement of the drops will deposit the carried material (thereby raising the nodes' elevations) or produce erosion of their routes by removing the silt from the nodes. The diminishing gradient, or the height difference between the node where the droplet dwells and its neighbor, determines the probability that the next node will be selected. Initially, the environment that is produced will be level, meaning that all nodes will have the same height, with the exception of the goal node, which will remain at zero height throughout the entire process. To aid in more site research, the drops are positioned in the first node. This will lead to the discovery of the best route. A set of drops will go through the area one after the other at each step, eventually wearing down the nodes that were visited. Following is the pseudo-code representation of the modified SED scheme:

initializeNodes()
initializeDrops()
while (not endConditionMet())
moveDrops()
analyzePaths()
erodePaths()
depositSediments()
end while

To define a set that is generated by the site's cell decomposition, the initialization of the algorithm's nodes (initialize nodes ()) is the first stage. All nodes also possess extra information, such as the distance to the destination and the amount of time needed to reach it, in addition to knowing whether there are any obstacles in their path. Conversely, during drop initialization (initializeDrops()), the first node is assigned the proper number of drops. The procedure is run through until it reaches the end condition (endConditionMet()). This suggests that each drop travels along the same route. Additionally, in order to shorten the computation time, a condition to confirm that the previous loops did not enhance the answer has been added, along with an upper limit on the number of iterations.

$$P_k(i,j)=\begin{cases}\dfrac{\text{gradient}(i,j)}{\text{sum.}\left(d_j\right)^{\alpha}}, \text{for } j\in V_k(i)\\ \dfrac{\omega/\left|\text{gradient}(i,j)\right|}{\text{sum.}\left(d_j\right)^{\alpha}}, \text{for } j\in U_k(i)\\ \dfrac{\delta}{\text{sum.}\left(d_j\right)^{\alpha}}, \text{for } j\in F_k(i)\end{cases} \quad (1)$$

Drops travel in groups of one until they reach their destination or run out of room; at this point, they evaporate and reenter the news loop. Eq. (1) provides the likelihood that the drop k, located in node I, will select node j after it:

The equation denotes that $V_k(i)$ represents the positive gradient neighbor set, wherein the node's altitude is greater than that of node j; $U_k(i)$ represents the negative gradient neighbor set, wherein node j has a higher altitude; and $F_k(i)$ represents the flat gradient neighbors, wherein the altitudes of both nodes and node j are similar. A selection of neighboring nodes (up to 8) is available, except those that are situated in cells with obstacles. The altitude difference between successive nodes is defined as the gradient, while coefficients ω and δ are fixed, tiny values. Here, the sum will represent the total of each neighbor's weights from several collections (the numerators). The length from node j to the objective will be shown by the coefficient d_j, and a convergence tuning coefficient will be indicated by α. The program makes the assumption that there is a probability of migrating on edges with an increasing slope. A low probability draw will perform in the event that such an edge is selected; this likelihood will drop with each repetition. If the pull is unsuccessful, the drop will evaporate and leave behind sediment corresponding to the transported sediment of the drop minus a certain amount. The route analysis (analyzePaths()) determines the optimal drop to carry out the extra erosion by detecting the optimum solution. The objective function is used to determine the solutions. The mean squared error (MSE) serves as the foundation for the objective function in this study. Better solutions have a lower MSE. Every path that is traversed will experience erosion as a result of the nodes' elevations dropping in proportion to the gradient with each subsequent node. To increase convergence, the best drop's deduction amount will be doubled by a parameter. Eq. (2) below expresses the notion of node erosion:

$$\begin{aligned}&\text{erosion}(i,j)=\\ &=\begin{cases}\varepsilon_V.\text{gradient}(i,j)/(N-1).M.\text{pathLength}_k,\\ \text{for } j\in V_k(i)\\ \varepsilon_U/\left|\text{gradient}(i,j)\right|.(N-1).M.\text{pathLength}_k,\\ \text{for } j\in U_k(i)\\ \varepsilon_F/(N-1).M.\text{pathLength}_k, \text{for } j\in F_k(i)\end{cases}\end{aligned} \quad (2)$$

In this equation, ε_V, ε_U, and ε_F are the corresponding parameters of the positive gradient neighbor group, the negative gradient neighbor group, and the flat gradient neighbor group. The length of the drop's travelled path will be indicated by PathLength_k. N will show the number of nodes, whereas M will show the number of drops.

3.2 Feature Selection using SED Algorithm with AdaBoost Classifier

In comparison to the majority of other learning algorithms, the AdaBoost algorithm has less susceptibility to the over-fitting issue even though boosting has a lot of sensitivity to noisy data as well as outliers. Hence, mislabelled cases are the reason for the over-fitting issue. Since new classifiers focus more on cases that are wrongly classified, there is the provision of a huge amount of weak classifiers such that a better performance is attained. A novel boosting algorithm termed the "SED-AdaBoost" has been developed in this work. This novel algorithm [25] will

optimize the number of weak classifiers as well as their corresponding weights through an SED utilization for performance enhancement of the boosting procedure. Since the SED can regulate how the outliers get affected, it can choose a feasible fitness function that restricts the number of weak classifiers and, therefore, boosts the prediction accuracy. In comparison to other boosting algorithms, this proposed algorithm is more advantageous as it reduces the model's complexity.

3.3 Feature Selection using Tabu Search (TS) Algorithm

Glover proposed the TS as a meta-heuristic technique for the resolution of combinatorial optimization problems. Of late, this technique has received a lot of interest. It has swiftly progressed in its application due to its flexible control framework as well as numerous outstanding successes in the resolution of NP-hard problems. When compared against the local search approach, the TS permits shifting towards a new solution that will make the objective function worse such that it does not trap in the local optimal solutions. A short-term memory termed the Tabu List (TL) is employed by the TS for recording and guiding the search procedure. Along with the TL, long-term memories, as well as other prior information about the solutions, are employed by the TS for enhancement of the search's [26] intensification and/or diversification.

3.4 Feature Selection using Tabu Search (TS) Algorithm with AdaBoost Classifier

One can use a Tabu List and the ε parameter simultaneously because the improvements made to the initial AdaBoost algorithm are not mutually exclusive. Both enhancements (Tabu List and constant ε value) can be tested to check if their utilization can further enhance the boosting technique. Therefore, a hybrid algorithm, ε-TabuBoost, is proposed by combining both these enhancements and is presented as the given algorithm. The resulting algorithm is called TabuBoost, and in the [27] iteration of the boosting procedure, it uses a constant value of ε as the base classifier weight.

3.5 Proposed Feature Selection using SED-TS Algorithm with AdaBoost Classifier

There is the development of a hybrid SED-Tabu approach [28] in this work to utilize both approaches' best features. For both approaches' facilitation, an altitude value is conferred to all graph nodes, whilst a Tabu move (in TL) trail value is conferred to all edges. When there is a release of hybrid drop-ant entities into this environment, Tabu move (in TL) trails, as well as altitudes, do have some weight in the determination of these entities' next movement. After the move, there is a modification of the Tabu move (in TL) trail as well as the place's altitude by each approach. The resultant hybrid approach is influenced by certain derivative values (altitude differences) as well as by certain values that are taken as they are (Tabu move (in TL) trails). The hybrid approach's search space analysis has resulted in the enhancement of the solution quality. Through a combination of these two approaches, the hybrid approach can avoid capture in solutions which one approach's local optimum but not for the other approach. Each approach can dominate for some time as there is weight alternation in each approach together with time. Even though both approaches are dependent on distinct driving entities, the intrinsic evolutionary computation model is related. As a result, the partial compatibility in both approaches' mechanics enables their partial collaboration in solution formation. This work has suggested an AdaBoost algorithm improvement method that involves a straightforward adjustment and is inspired by the well-known SED-Tabu approach for optimization issues. Interpretation of any classifier's training is given as an optimization procedure for a given performance criterion like exponential loss, error or classification ratio. For SED algorithm improvement, TS is a well-established method. Its key concept is the maintenance of a list of forbidden solutions that cannot be utilized by the SED algorithm during the search space's exploration. Due to this, the local minima are avoided by these algorithms. This work will initially review the AdaBoost, a fundamental boosting algorithm, and later give the algorithm's interpretation as a SED optimization procedure. From this perspective, there is the learning process's modification through the introduction of a list of forbidden features that are not to be utilized during the consequent base classifiers' learning process within the procedure of boosting. In comparison to the fundamental AdaBoost algorithm, this simple modification has been proved to enhance the final classifier.

4 RESULTS AND DISCUSSION

The Amazon product review dataset contains millions of reviews from Amazon customers (as input texts) and star ratings (as output labels) to learn how fast the text for Sentiment Analysis can be made. A comparative study on an Amazon reviews dataset was conducted using 5-fold stratified cross-validation to minimize variability. To ensure a fair comparison, the hyperparameters of each method were optimized via grid search. The results were averaged over 10 independent runs for each method, and the performance was evaluated using the metrics of accuracy, F1 score, precision, and recall. In this section, the SED feature selection - Adaboost, SED-Tabu feature selection - Adaboost, SED feature selection & Adaboost optimization, and SED-Tabu feature selection & Adaboost optimization methods are used. Tab. 1 through 5 and Fig. 2 through Fig. 6 display the accuracy, recall for (positive, negative, and neutral), precision for (positive, negative, and neutral), f measure for (positive, negative, and neutral), and fitness.

Table 1 Accuracy for SED-Tabu feature selection & adaboost optimization

Methods	Accuracy
SED feature selection - Adaboost	90.05
SED-Tabu feature selection - Adaboost	91.07
SED feature selection & Adaboost optimization	92.08
SED-Tabu feature selection & Adaboost optimization	93.08

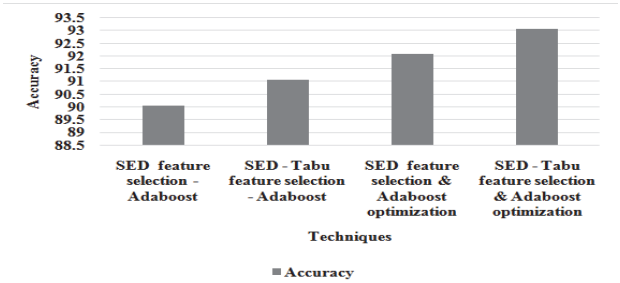

Figure 2 Accuracy for SED-Tabu feature selection & adaboost optimization

Fig. 2 shows that the accuracy of the SED-Tabu feature selection as well as Adaboost optimization is greater by 3.31%, 2.18%, and 1.08%, respectively, for the SED feature selection & Adaboost optimization, SED-Tabu feature selection - Adaboost, and SED feature selection & Adaboost optimization.

Table 2 Recall for SED-Tabu feature selection & adaboost optimization

	SED feature selection Adaboost	SED Tabu feature selection Adaboost	SED feature selection & Adaboost optimization	SED-Tabu feature selection & Adaboost optimization
Recall for Positive	0.937	0.9443	0.9512	0.9523
Recall for Negative	0.8653	0.8809	0.8984	0.9182
Recall for Neutral	0.86	0.8653	0.866	0.8827

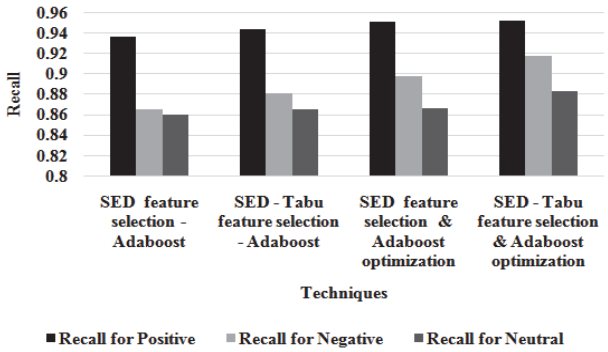

Figure 3 Recall for SED-Tabu Feature Selection & Adaboost Optimization

The average recall for SED-Tabu feature selection & Adaboost optimization is greater by 3.36%, 2.3%, and 1.37%, respectively, for SED feature selection & Adaboost optimization, SED-Tabu feature selection & Adaboost, and Adaboost, as shown in Fig. 3.

Table 3 Precision for SED-Tabu Feature Selection & Adaboost Optimization

	SED feature selection Adaboost	SED Tabu feature selection Adaboost	SED feature selection & Adaboost optimization	SED-Tabu feature selection Adaboost optimization
Precision for Positive	0.927	0.9299	0.9368	0.9531
Precision for Negative	0.9379	0.9461	0.9475	0.9508
Precision for Neutral	0.7235	0.756	0.7916	0.7981

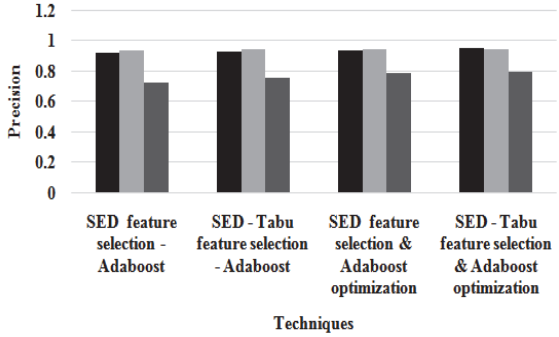

Figure 4 Precision for SED-Tabu feature selection & adaboost optimization

The average precision of the SED-Tabu feature selection and Adaboost optimization is higher, as seen in Fig. 4, by 4.29%, 2.62%, and 0.97%, respectively, for the SED feature selection and Adaboost optimization, Tabu feature selection, and Adaboost optimization.

Table 4 F Measure for SED-Tabu feature selection & adaboost optimization

	SED feature selection - Adaboost	SED Tabu feature selectionAdaboost	SED feature selection & Adaboost optimization	SEDTabu feature selection & Adaboost optimization
F measure for Positive	0.932	0.937	0.9439	0.9527
F measure for Negative	0.9001	0.9123	0.9223	0.9342
F measure for Neutral	0.7859	0.807	0.8271	0.8383

As seen in Fig. 5, the average f measure for SED-Tabu feature selection & Adaboost optimization is higher by 4.01%, 2.56%, and 1.18%, respectively, for SED feature selection & Adaboost optimization, SED-Tabu feature selection, and Adaboost optimization.

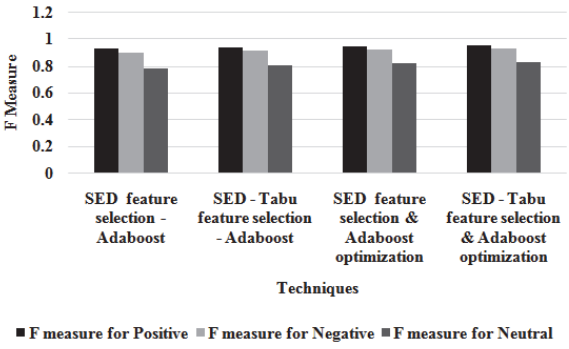

Figure 5 F Measure for SED-Tabu feature selection & adaboost optimization

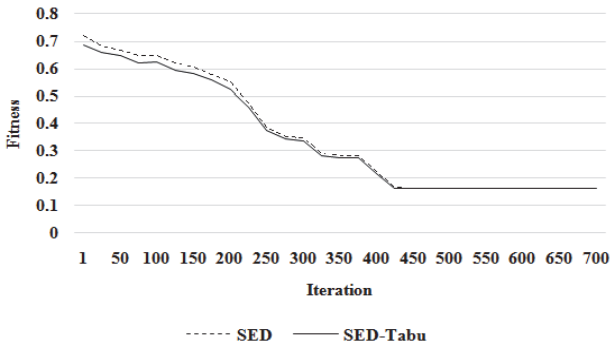

Figure 6 Fitness for SED

Table 5 Fitness for SED

Iteration	SED	SED-Tabu
1	0.719	0.6838
25	0.6812	0.6559
50	0.6659	0.6455
75	0.6462	0.6166
100	0.645	0.6221
125	0.6182	0.5935
150	0.6039	0.5784
175	0.5771	0.5565
200	0.549	0.5235
225	0.4752	0.458
250	0.3871	0.3735
275	0.3537	0.3421
300	0.3484	0.3351
325	0.2913	0.2806
350	0.283	0.2713
375	0.2826	0.2738
400	0.2245	0.2141
425	0.1679	0.1613
450	0.1625	0.1613
500	0.1625	0.1613
550	0.1625	0.1613
600	0.1625	0.1613
650	0.1625	0.1613
700	0.1625	0.1613

From Fig. 6, it can be observed that the SED has higher average fitness by 3.42% for SED-Tabu, respectively.

5 CONCLUSIONS

They proposed a novel hybrid SED-Tabu feature selection approach for sentiment analysis that integrates the exploration strength of SED with the exploitation power of Tabu Search; evaluated the method on the Amazon product review dataset and demonstrated significant improvements in classification accuracy and F1 - score over baseline feature selection techniques. It is demonstrated from the experimental outcomes that the SED-Tabu feature selection & AdaBoost optimization has higher accuracy by 3.31% for SED feature selection-AdaBoost, by 2.18% for SED-Tabu feature selection-AdaBoost, and by 1.08% for SED feature selection & AdaBoost optimization. The key contributions of this work encompass the innovative fusion of SED and Tabu Search to create a potent feature selection framework, leading to improved sentiment classification in high-dimensional noisy text data. The hybrid approach combines strengths from multiple optimization paradigms, providing a broadly applicable technique for sentiment analysis involving Big data. The distributed computing, online learning, approximation, and periodic retraining techniques can allow the SED-Tabu and AdaBoost models to scale effectively to large real-world sentiment data. Future work should focus on improving interpretability, and scalability, reducing tuning, and modeling feature relationships; conducting scalability experiments on Big data platforms like Hadoop/Spark to quantify runtimes and system bottlenecks.

6 REFERENCES

[1] Jankovic, A., Adrodegari, F., Saccani, N., & Simeunovic, N. (2022). Improving Service Business of Industrial Companies Through Data: Conceptualization and Application. *International Journal of Industrial Engineering and Management, 13*(2), 78-87. https://doi.org/10.24867/IJIEM-2022-2-302

[2] Prasetyo, H. N., Sarno, R., Wijaya, D. R., Budiraharjo, R., & Waspada, I. (2023). Sampling Simulation in Process Discovery. *International Journal of Simulation Modelling, 22*(1), 17-28. https://doi.org/10.2507/IJSIMM22-1-619

[3] Tian, S., Zhang, Z., Xie, X., & Yu, C. (2022). A new approach for quality prediction and control of multistage production and manufacturing process based on Big Data analysis and Neural Networks. *Advances in Production Engineering & Management, 17*(3), 326-338. https://doi.org/10.14743/apem2022.2.430

[4] Arasteh, B., Bouyer, A., Ghanbarzadeh, R., Rouhi, A., Mehrabani, M. N., & Tirkolaee, E. B. (2023). Data Replication in Distributed Systems Using Olympiad Optimization Algorithm. *FACTA Universitatis-Series Mechanical Engineering, 21*(3), 501-527. https://doi.org/10.22190/FUME230707033A

[5] Bajic, B., Suzic, N., Simeunovic, N., Moraca, S., & Rikalovic, A. (2020). Real-time Data Analytics Edge Computing Application for Industry 4.0: The Mahalanobis-Taguchi Approach. *International Journal of Industrial Engineering and Management, 11*(3), 146-156. https://doi.org/10.24867/IJIEM-2020-3-260

[6] Feng, C., Zhao, B., Zhou, X., Ding, X., & Shan, Z. (2023). An Enhanced Quantum K-Nearest Neighbor Classification Algorithm Based on Polar Distance. *Entropy (Basel), 25*(1), 127. https://doi.org/10.3390/e25010127

[7] Kosciessa, J. Q., Kloosterman, N. A., & Garrett, D. D. (2020). Standard multiscale entropy reflects neural dynamics at mismatched temporal scales: What's signal irregularity got to do with it?. *PLOS Computational Biology, 16*(5), e1007885. https://doi.org/10.1371/journal.pcbi.1007885

[8] Goswami, A., Krishna, M. M., Vankara, J., Gangadharan, S. M. P., Yadav, C. S., Kumar, M., & Khan, M. M. (2022). Sentiment Analysis of Statements on Social Media and Electronic Media Using Machine and Deep Learning Classifiers. *Computational Intelligence and Neuroscience, 2022*, 9194031. https://doi.org/10.1155/2022/9194031

[9] Chaubey, P. K., Arora, T. K., Raj, K. B., Asha, G. R., Mishra, G., Guptav, S. C., Altuwairiqi, M., & Alhassan, M. (2022). Sentiment Analysis of Image with Text Caption using Deep Learning Techniques. *Computational Intelligence and Neuroscience, 2022*, 3612433. https://doi.org/10.1155/2022/3612433

[10] Jotheeswaran, J. & Kumaraswamy, Y. S. (2013). Opinion mining using decision tree based feature selection through manhattan hierarchical cluster measure. *Journal of Theoretical and Applied Information Technology, 58*(1), 72-80.

[11] Sharma, R., Nigam, S., & Jain, R. (2014). Opinion mining of movie reviews at document level. *International Journal on Information Theory, 3*(3). https://doi.org/10.5121/ijit.2014.3302

[12] Alarifi, A., Tolba, A., Al-Makhadmeh, Z., & Said, W. (2020). A Big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. *The Journal of Supercomputing, 76,* 4414-4429. https://doi.org/10.1007/s11227-018-2398-2

[13] Mafarja, M. M. & Mirjalili, S. (2017). Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing, 260,* 302-312. https://doi.org/10.1016/j.neucom.2017.04.053

[14] Ajibade, S. S. M., Ahmad, N. B., & Shamsuddin, S. M. (2019). A novel hybrid approach of Adaboostm2 algorithm and differential evolution for prediction of student performance. *International Journal of Scientific and Technology Research, 8*(07), 65-70.

[15] Hadi, K. A., Lasri, R., & El Abderrahmani, A. (2019). An efficient approach for sentiment analysis in a Big data

environment. *International Journal of Engineering and Advanced Technology (IJEAT), 8*(4), 263-266.
[16] Rodrigues, A. P. & Chiplunkar, N. N. (2019). A new Big data approach for topic classification and sentiment analysis of Twitter data. *Evolutionary Intelligence, 15*(9), 1-11. https://doi.org/10.1007/s12065-019-00236-3
[17] Liu, B. (2020). Text sentiment analysis based on CBOW model and deep learning in Big data environment. *Journal of Ambient Intelligence and Humanized Computing, 11*(2), 451-458. https://doi.org/10.1007/s12652-018-1095-6
[18] Hammou, B. A., Lahcen, A. A., & Mouline, S. (2020). Towards a real-time processing framework based on improved distributed recurrent neural network variants with fastText for social Big data analytics. *Information Processing & Management*, *57*(1), 102-122. https://doi.org/10.1016/j.ipm.2019.102122
[19] Rahmani, A. M., Azhir, E., Ali, S., Mohammadi, M., Ahmed, O. H., Yassin Ghafour, M., Hasan Ahmed, S., & Hosseinzadeh, M. (2021). Artificial intelligence approaches and mechanisms for Big data analytics: a systematic study. *PeerJ Computer Science, 7*, e488. https://doi.org/10.7717/peerj-cs.488
[20] Sharma, S. & Jain, A. (2020). Hybrid Ensemble Learning With Feature Selection for Sentiment Classification in Social Media. *International Journal of Information Retrieval Research (IJIRR), 10*(2), 40-58. https://doi.org/10.4018/IJIRR.2020040103
[21] Zhu, X. (2021). A Face Recognition System Using ACO-BPNN Model for Optimizing the Teaching Management System. *Computational Intelligence and Neuroscience*, 5194044. https://doi.org/10.1155/2021/5194044
[22] Alarifi, A., Tolba, A., Al-Makhadmeh, Z., & Said, W. (2020). A Big data approach to sentiment analysis using greedy feature selection with cat swarm optimization-based long short-term memory neural networks. *The Journal of Supercomputing, 76*(6), 4414-4429. https://doi.org/10.1007/s11227-018-2398-2
[23] Hassib, E. M., El-Desouky, A. I., Labib, L. M., & El-kenawy, E. S. M. (2020). WOA + BRNN: An imbalanced Big data classification framework using Whale optimization and deep neural network. *Soft computing, 24*(3), 5573-5592. https://doi.org/10.1007/s00500-019-03901-y
[24] Liu, W., Guo, Z., Jiang, F., Liu, G., Wang, D., Ni, Z. (2022). Improved WOA and its application in feature selection. *PLoS One, 17*(5), e0267041. https://doi.org/10.1371/journal.pone.0267041
[25] Mousin, L., Jourdan, L., Marmion, M. E. K., & Dhaenens, C. (2016). Feature selection using tabu search with learning memory: learning Tabu Search. *International Conference on Learning and Intelligent Optimization,* 141-156. https://doi.org/10.1007/978-3-319-50349-3_10
[26] Milford, M. & Schulz, R. (2014). Principles of goal-directed spatial robot navigation in biomimetic models. *Philosophical Transactions of the Royal Society B: Biological Sciences, 369*(1655), 20130484. https://doi.org/10.1098/rstb.2013.0484
[27] Hu, W., Gao, J., Wang, Y., Wu, O., & Maybank, S. (2014). Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection. *IEEE Transactions on Cybernetics, 44*(1), 66-82. https://doi.org/10.1109/TCYB.2013.2247592
[28] Bereta, M. (2019). Regularization of boosted decision stumps using tabu search. *Applied Soft Computing, 79,* 424-438. https://doi.org/10.1016/j.asoc.2019.04.003

Contact information:

Sabitha RAJAGOPAL, Professor
Department of Computer Science and Engineering,
SNS College of Technology,
Tamilnadu 641035, India
E-mail: dr.r.sabitha@gmail.com

Sreemathy JAYAPRAKASH, Assistant professor
(Corresponding author)
Department of Computer Science and Engineering,
Sri Eshwar College of Engineering,
Kinathakadavu, Coimbatore, Tamilnadu 641202, India
E-mail: jsreemathybe@yahoo.com

Karthik SUBBURATHINAM, Professor & Dean
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore, Tamilnadu 641035, India
E-mail: profskarthik@gmail.com