FEATURE SELECTION METHOD FOR SENTIMENT CLASSIFICATION USING MACHINE LEARNING CLASSIFIERS

Vinoth Kumar BALASUBRAMANIAN

, Karpagam MANAVALAN

Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

Abstract

Online purchasing is becoming increasingly popular at all stores. Textual user reviews posted on e-commerce websites are a primary source for determining the quality of a product. Customers cannot read all online customer reviews to fully comprehend the product dependability, customer service, and quality. The fastest-growing discipline is sentiment analysis, which uses machine learning algorithms to correctly identify positive or negative text reviews. Scalability, overfitting, and underfitting arise during the training of data using machine learning. Selecting suitable machine learning model training features eliminates these concerns. Thus, genetic algorithms use their powerful searching abilities to choose the best characteristics from the training data. In this paper, we address the key issues in using genetic algorithms for feature selection are slow convergence speed and long computation time. The objective of this work is to propose a method that injects domain-specific knowledge into genetic algorithms to address scalability, overfitting, and underfitting problems. The research work proposes KBGA (Knowledge-based Genetic Algorithm) and KBDE (Knowledge-based Differential Evolution) algorithms to improve the scalability problem and reduce the computation time. This work performs an extensive performance analysis and a statistical significance test to confirm the efficiency of proposed algorithms. Furthermore, the proposed work excellency is compared with existing sentiment analysis model in order to find the proposed work efficiency using the metrics namely precision, recall, F-measure and accuracy.

Keywords

Product Quality, Customers' Perspective, Sentiment Analysis, Machine Learning

1. Introduction

People enthusiasm for making purchases via any kind of e-commerce website is at an all-time high right now. Reviews written by actual consumers and published on various online marketplaces are a valuable source of information about the reliability of the products being provided by these marketplaces. A potential buyer may find it challenging to sift through the many customer reviews available to draw any firm conclusions about the product reliability, customer service, or general quality [1]. Sentiment analysis, which employs machine learning methods, is currently the fastest-growing area of study. This is due to the fact that sentiment analysis attempts to correctly predict whether positive or negative perspectives are being expressed in user-posted text evaluations. Products have been rated on a scale from one to five stars based on this independent assessment [2].

Building algorithms that fully leverage data attributes is the main focus of feature engineering. However, the very large dimensionality of the data they offer could make training these features challenging. One of the most common and useful techniques for transforming high-dimensional information into low-dimensional, more manageable representations is dimensionality reduction [3].

When it comes to dimensionality reduction techniques, you may classify them into either feature extraction or feature selection [4]. For feature extraction, it is necessary to first generate new features by combining existing features, and then to project those features into a space with less dimensions. In contrast, Feature Selection (FS) use a criterion measure to narrow down a large set of candidate features to a more manageable subset. Feature selection is an optimization technique that takes a big pool of data and narrows it down to a more manageable subset of features, all the while boosting classification accuracy and speed. As part of this procedure, a vast number of features is reduced to a more manageable subset [5].

Depending on the presence or absence of human oversight, supervised file system operations include the use of filters and wrapper techniques. There are many other instances where this occurs, including as Wrapper models centralize in one place certain parts of the entire set of features that are functionally comparable to one another. After gauging the efficacy of a machine learning algorithm trained with only those features, they make an ad hoc call on which features to employ []. Wrapper methods, on the other hand, become impractical and computationally expensive when there are many features. Overfitting is possible with wrapper approaches because they employ numerous parameters with less training data [6].

To simplify the task of feature selection, a genetic algorithm (GA) can be used as a wrapper technique. A genetic algorithm can be used to organize and classify a wide variety of optimization problems in an intuitive structure. The GA [7] is a heuristic approach to solving problems that heavily draws inspiration from Darwin theory of evolution and its central concept of natural selection.

Machine learning classifiers can be trained with the data we have, but it probable that in most cases they won't be able to produce a reliable decision boundary for classification. It was conceivable for overfitting or underfitting to occur if the classifier showed significant bias or fluctuation. Even still, issues with scalability, overfitting, and underfitting become apparent during the training phase of machine learning. To relieve some of the above problems, select features that are pertinent to the work at hand, such as training the machine learning model [8]. Therefore, there are several methods [9]–[10] for picking out features to employ in the training data, with genetic algorithms favoring the most effective ones due to their propensity for finding answers. Training many classifiers concurrently and using their combined predictions is one approach to resolving this problem.

When it comes to feature selection, genetic and differential evolutionary algorithms present some fundamental issues that we attempt to address in our work. Some of the problems with these methods are that they take a long time to compute and have a slow convergence speed. One of the aims of this study is to evaluate the various approaches used by FSs to the study of sentiment.

In this paper, we address the key issues like slow convergence speed and long computation time of genetic and differential evolutionary algorithm for feature selection. The novelty of this research is to perform extensive evaluation of different FS techniques for sentiment analysis.

The main contribution of the work involves the following: A propose of a KBGA (Knowledge-based Genetic Algorithm) and KBDE (Knowledge-based Differential Evolution) method that injects domain-specific knowledge to address scalability, overfitting, and underfitting problems.

2. Related works

Feature vocabulary is a good starting point, it just as important to narrow it down to the elements that are truly necessary for conveying the intended meaning. Therefore, FS methods are crucial for boosting the effectiveness of the training algorithm and speeding up the learning

process [11]. To conduct sentiment analysis on Chinese media [12]. Combining IG and genetic algorithms into a single inference engine, the Entropy Weighted Genetic Algorithm (EWGA), enhanced the accuracy of sentiment classification. To determine which criteria to utilize in categorizing viewpoints, Fisher discriminant ratio is applied.

Online product reviews were compiled by Morinaga [13] by the use of a frequency analysis of key terms and phrases. This paved the way for opinion mining to examine client input. We found that machine learning classifiers like support vector machines and naive bayes outperformed manually constructed features when trained to categorize the sentiment of product reviews [14]. Bag-of-Words feature extraction was used to create the feature vocabulary. The most crucial, yet least frequent, words are typically overlooked when selecting phrases based on the count threshold. As a result, the TF-IDF [15] was developed as a more efficient method of feature extraction.

Pang et al. [16] were the first researchers to utilize machine learning to analyze the tone of film critic writings. They were able to reach an accuracy of 70–80%, which is significantly higher than the 70% attained by human baseline sentiment analysis. Sentiment analysis at the phrase level has been shown to be possible by Santos and Gatti [17]. Sentiment analysis makes use of both textual elements (words) and graphical elements (characters). Unfortunately, the lengthy computation time can be traced back to the massive number of features that were constructed.

A feature selection strategy is required for accurate machine learning categorization. It is the goal of some academic researchers to use as few features as possible in their studies. The feature count (FC) method was proposed by Manurung [18]. FC prioritizes the most commonly occurring subfeatures when making a decision. Only selecting an additional feature is chargeable. However, it possible that select the features that is not significant to the output class, since high incidence does not always guarantee that a feature is important to that class.

The evolutionary methods in an effort to enhance well-established features is common. Following a Relief-F filter technique examination of the entire feature set, Zhang et al. [19] fed the resulting subset into a genetic algorithm to determine which set of features is most likely to yield the best outcomes. The proposed solution to the character recognition problem was tested on a dataset made up entirely of Chinese characters.

While working with high-dimensional micro-array datasets, Apolloni et al. [20] suggested a combination of the filter and wrapper methods for selecting features. The strategy used a hybrid

of the binary differential evolution method and the IG feature ranking technique to make feature selections.

To use the filter and wrapper, Hsu et al. [21] devised a three-stage method. They first filtered features using the intersection of F-score and IG, then we used sequential backward search to eliminate superfluous features, and finally, we used sequential forward search to add features from the X-OR portion of the filter techniques, all in an effort to boost the classifier performance.

In an effort to improve the reliability of classifiers used in the Alzheimer disease diagnosis process, Zhang et al. [22] created a multivariate technique that collected textual information from brain images using stationary wavelet entropy. The technique was used to decipher text from scans of the brain. The authors introduced a unique chaotic PSO that utilized the Hu moment invariant to suit the requirements of the medical robot alcoholism detection competition.

According to Basari et al. [23], evolutionary algorithms have seen relatively little exploration in the field of sentiment analysis. Onan et al. [24] employed a genetic algorithm to aggregate the results from multiple filter-based feature selection procedures by integrating feature lists. Researchers utilize Spearman footrule to narrow the gap between the many existing feature lists and arrive at the optimal feature list possible for their investigation. Experiments were conducted on sentiment analysis datasets from a wide variety of domains and fields, with the NB and k-nearest neighbor algorithms serving as base learners. Nicholls et al. [25] came up with the idea for the entropy-weighted genetic algorithm (EWGA) to improve the process of feature selection utilized in sentiment analysis.

3. Proposed Method

In this section, we proposed two different machine learning models that involves 1) KBGA (Knowledge-based Genetic Algorithm) and 2) KBDE (Knowledge-based Differential Evolution) algorithms to improve the scalability problem and reduce the computation time as illustrated in Figure 1.

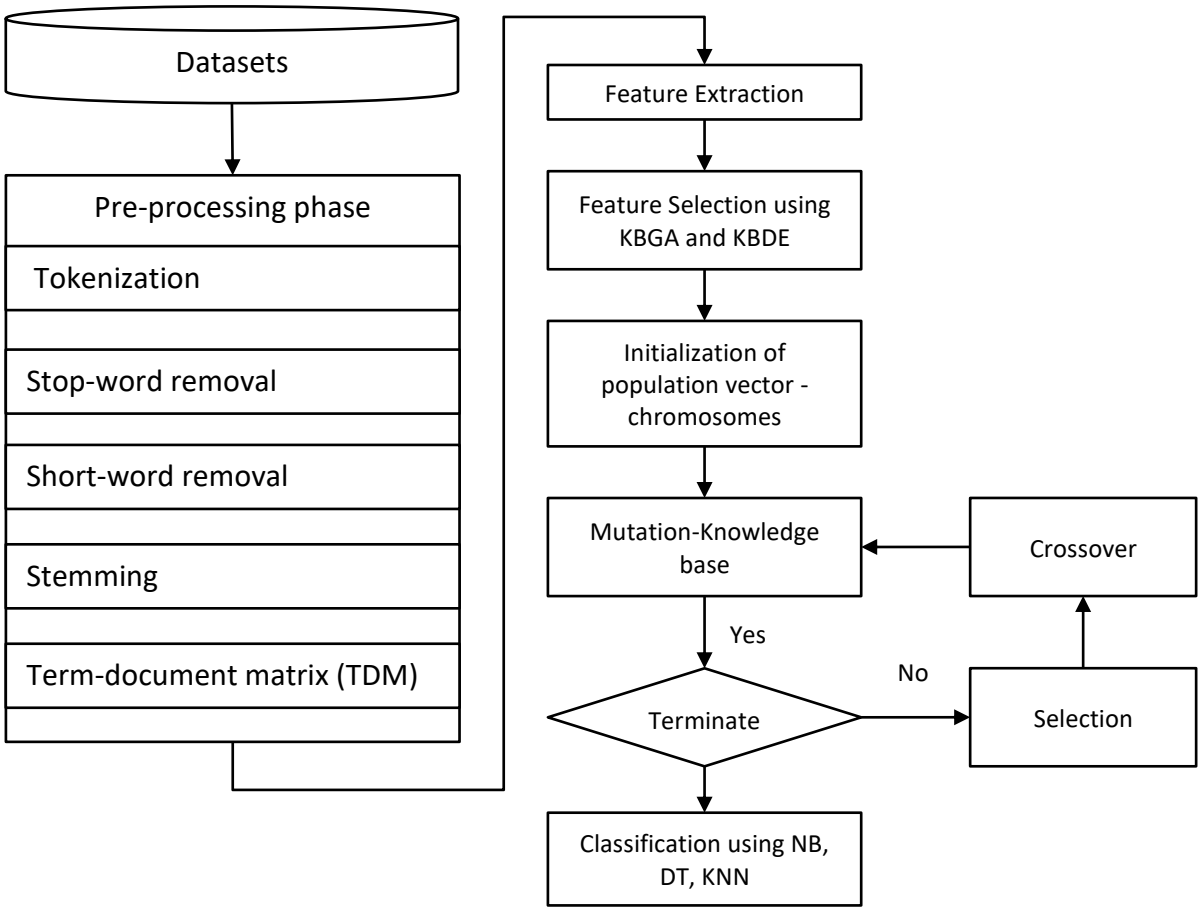

Figure 1: Proposed Framework

Pre-Processing Phase

In this framework, a data-cleaning component is integrated into the preliminary stages of the data-analysis procedure. This process involves reading information from files and transferring it in a steady stream into memory, where it will stay until it is cleared. This process will continue until the memory is completely erased. There are three distinct phases contained within this overall one.

Tokenization

Tokenization refers to the process of splitting a string of text into individual words, phrases, symbols, or other relevant parts. To complete the tokenization procedure, we use the punctuation-preserving LingPipeTokenizer, which is part of the Apache Lucene package. We initially experimented with the StringTokenizer, but after realizing its limits, we switched to the more robust LingPipeTokenizer. Tokens (keywords) and phrases are stored in unique data structures that are constructed for each document (list of keywords).

Stop-word removal

The technique known as stopword deletion includes eliminating the most frequently used words and phrases in a language. In most cases, NLP is unaffected by the words in question. With the help of CMU Rainbow stopword list, it is easy to find every stopword that could be present in the data.

Short-word removal

It is necessary to preprocess raw data gathered from various sources in order to eliminate terms that are irrelevant to user ratings. During data processing, we get rid of blanks and cut off words.

Stemming

Words that have undergone inflection are stemmed, or reduced to their uninflected root form. Tokens are stemmed using the porter-2 method and both the stemmed and original tokens are persisted in the key word object.

Term-document matrix (TDM) conversion

The filtering step can be applied to remove any remaining special characters like $, @, or # once all remaining tokens have been reduced to their roots. This occurs once the remaining tokens have been reduced to their essence. TDM, which stands for term distribution model, is a tabular display of how frequently each word appears in the corpus. It is implemented to transform the filtered stream. The final step is to analyze the TF-IDF data for patterns to determine which terms are most important.

In many cases, a TF-IDF analysis significance has two parts. In most instances, this is accurate. The first term estimates the normalized term frequency of the document (TF). The second concept, the inverse document frequency (IDF), is calculated by dividing the logarithm which include the word being searched for.

TF can be used to count how many times a word or phrase appears in a given text. Due to differences in document length, some documents may include significantly more instances of a given term than others. That because there more room for detail in longer documents. A common method of standardizing term frequency is to divide it by the total number of terms in the document. This strategy is often used.

$TF(t)$ = Number of times term t appears in a document/Total number of terms in the document

The importance of a given phrase is quantified by the IDF. All phrases are given the same weight in the final TF calculation. The following must be determined so that fewer-used phrases have less weight and more-used phrases have more weight:

$IDF(t)$ = log (Total number of documents/Number of documents with term t in it)

If there are a hundred words in the paper and five of them are man, we can say that there are five occurrences of the word. Therefore, the occurrence frequency, or TF, for people is 5/100, or 0.05. Consider a set of documents that numbers in the millions, but only a thousand of them actually mention man. If that the case, then the formula log(10,000,000/1,000) = 4 can be used to determine the inverse document frequency (IDF). Multiplying (0.05 * 4) by 4 yields (0.2) as the TF-IDF value.

3.2. Preliminaries of Feature Selection

In this section, the conventional genetic algorithm and evolutionary algorithm is presented below:

3.2.1 Genetic Algorithm

It mimics biological evolution through processes including mutation, crossover, inheritance, and selection, the Genetic Algorithm (GA) is sometimes referred to as a evolutionary algorithm.

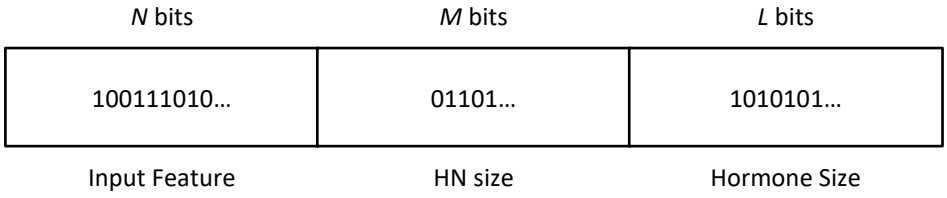

Fig. 2 Binary coded GA representation

The parameters of the GA algorithm are stored in chromosomes made up of characteristics, nodes, and hormones, as depicted in Figure 2. The algorithm might contain this logical framework. Keep in mind that the chromosomes are represented by the binary code GA. The genetic code clearly displays this. The feature fraction consists of N bits, where N is the total

number of features in the set. A 0 indicates that the feature is not chosen at that location, while a 1 indicates that it is. The node is the second section and it has M bits.

Up until the second decimal place is double the value of the first, the number of chromosomes will rise proportionally (feature number). Since there are M possible binary bit values for each node, the maximum number that can be stored in HN is 2M. For each chromosome, the final letter represents the total number of hormones it carries as a L value. This presentation is preferred as it can be implemented with a small to medium amount of code, making it scalable to any size of network.

The validation dataset Mean Square Error (MSE) serves as the basis for the formulation of the shape function. This is supported by evidence presented in (Eq. 4).

$$MSE = 100\left(pq\right)^{-1}\sum_{m=1}^{p}\sum_{n=1}^{q}\left(O_m^n - T_m^n\right)^2$$

where

q – total separate data sets,

p - number of crop nodes,

T - objective value of m^{th} row output for n^{th} column and

O - the output value of n^{th} column for m^{th} row.

Creation of a new generation is strictly dependent on the process of chromosomal replication. Because of this, chromosomes with a lower MSE are more likely to be chosen as parental chromosomes. Initially, a ranking is made based on the MSE value, which serves as the basis for further selection. A higher score indicates that the studied chromosome is of higher quality. The fitness of a chromosome will therefore be determined by its position in the m^{th} row.

$$fitness\left(m\right) = \frac{100 \times rank\left(m\right)}{\sum_{n=1}^{m} rank\left(n\right)}$$

To determine which sets of chromosomes will serve as the parents, a selection procedure analogous to roulette is used. When both sets of paternal chromosomes are entirely mapped, a crossover will occur. There will be two offspring born as a direct result of the chromosomal exchange, and they will share the genetic features of both parents. These replacement chromosomes will replace the pair of chromosomes with the lowest score. This mimics a

mechanism at work in biological evolution, whereby less advantageous chromosomes (those with lower ranks) are weeded out and replaced with newer, more advantageous ones. These processes culminate in the mutation that all the chromosomes will acquire.

This study employs both a single-point crossover and a modified procedure. No compromises will be made, and only the best set of paternal chromosomes will be used in the process of crossing over to the next generation. One of the first things to do when doing a crossover is to generate a random crossing point. Afterwards, the information in the parent chromosomes will switch places after the created point, giving rise to two new chromosomes. Next, the two new chromosomes will switch places with the chromosomes at the bottom of the current order. It is planned that one day all chromosomes will undergo the mutation process. Depending on how likely it is to change, each piece of chromosome has a different chance of being chosen (0 or 1) or not selected (1).

3.2.2 Evolutionary Algorithm

Initializing training parameters in Differential Evolution involves taking the steps of setting the population size N, the individual dimension N_{par}, the mutation scaling parameter F, and the crossover probability C_R. Using the following steps, you may generate a population X that has a size of N and a dimension of N_{par}:

$$x_i = L_i + rand\left(N, N_{par}\right)*\left(U_i - L_i\right)$$

where

$x_i \in X$, i $\in$ 1, 2, ..., N,

The search space is denoted by the L and U, which stand for the bottom and upper limits, respectively. Utilizing the rand() function allows for the construction of a random matrix that has elements that fall anywhere within the range [0,1].

We will have the ability, with the help of the mutation operator, to make a new individual v_i that is derived from the one that is already there, which is the parent x_i. The DE scheme, commonly referred to as DE/RD/bin, is the one that is responsible for carrying out the mutation process. Equation defines the DE scheme.

$$v_i^t = x_{r1}^t + f*\left(x_{r2}^t - x_{r3}^t\right)$$

where

x_r^1, x_r^2, and x_r^3 – are recorded after being drawn at random from the population during iteration *t*. This selection produced these results. The following is an illustration of one way in which the crossover operator can be utilized to produce a new person who has the offspring's v_i and x_i:

$$z_{ij}^t = \begin{cases} v_{ij}^t & if\ \gamma_j \le CR\, or\, \delta_i \\ x_{ij}^t & otherwise \end{cases}$$

where

c_j - random number generator.

δ_i - random decision variable

The fitness function of the parent, which is denoted by fitx_i, but also the fitness function of the x_i, which is denoted by fitz_i.

Following the completion of the calculation of the fitness function for each parent and child pair, x_i and z_i, the selection operator is then applied in order to choose the more qualified individual out of the two candidates.

$$x_i^{t+1} = \begin{cases} z_i^t & if\ \ fit_{z_i} \le fit_{x_i} \\ x_i^t & otherwise \end{cases}$$

Everything that came before it is repeated in an endless loop until the goal is accomplished. The DE will put a stop to the process as soon as it receives a response that satisfies its standards, at which point it will send back the candidate who performed the best. In that case, it will start the process anew at the mutation step and go through it till completion. It is feasible to adjust the mutation strategy of a DE algorithm in such a way as to broaden the search space's potential for both discovery and exploitation. This can be done in a number of different ways. It is important to differentiate between these approaches by adopting a notation of the type DE/a/b, where DE stands for differential evolution, *a* represents the solution to be modified, and *b* stands for the number of distinct solutions applied. This notation can be found below. In this investigation, just two research approaches are utilized, and they are labeled below as DE/best/1.

$$v_i^t = x_b^t + F * \left(x_{r2}^t - x_{r3}^t \right)$$

DE/best/2 is the one that appears when you select the second choice.

$$v_i^t = x_b^t + F * \left(x_{r2}^t - x_{r3}^t \right) + F * \left(x_{r3}^t - x_{r4}^t \right)$$

where x^t_b - best solution at t^{th} iteration.

3.3. Feature Selection Methods

A dataset is composed of three parts: the rows of instances, the columns of features, and the rows of classes. The most difficult part of working with datasets is figuring out where each piece of unclassified data belongs. We call the set of issues that come under this heading feature selection problems.

Selecting the best dataset to maximize classification accuracy while minimizing false positives is a primary goal of the proposed strategy. Several attributes in the original dataset are irrelevant, unneeded, or duplicated, all of which lower the classifier precision. This is due to the fact that selecting which characteristics to utilize can significantly increase classifier performance and significantly reduce the size of a dataset.

Assume *A* is a new dataset with *m* instances from *d* features. The set of all *d* features shall be called *D*. Finding the subset of features inside data set *D* (i.e., $n \leq d$) that most effectively maximizes the objective function f is what feature selection is all about $f(X)$. Lets pretend the response is being transmitted using a binary encoding scheme.

$$X=(x_{t1}, x_{t2},\ldots, x_{td});\ \ t = 1,2,\ldots,m;\ \ x_{tk} \in \{0,1\}$$

To be valid, a solution must satisfy the following two conditions: (1) $x_{tk} = 0$ means that the k^{th} feature is not selected, and (2) $x_{tk} = 1$ means that it is. Mathematical expressions like these can help describe the difficulty of making good feature choices:

$$\max \text{ or } \min f(X)$$

s.t.

$$X=(x_1, x_2,\ldots, x_d);\ \text{ where } x_k \in \{0,1\}$$

$1 \leq |X| \leq |D|$

KBGA and KBDE are both presented below as potential models for selecting sentimental features.

KBGA Feature Selection

For the finest possible quantization table, it is crucial to have a firm grasp on the fundamentals of text transformation and the quantization procedure. Transform coding is an approach that relies on the divide and conquer strategy, which entails attempting to tackle a large problem by first dividing it down into smaller, more manageable problems. Three steps make up the text transform coding procedure: applying the transform, quantizing, and entropy coding. Using variable-length coding and fewer links between pixels, transform coding is meant to reduce the amount of data needed to store an image. A sentence is first divided into blocks of size m by n, and then each block is subjected to a discrete cosine transform (DCT) in order to execute the transform coding principle, which is what JPEG does. Making the Discrete Cosine Transform (DCT) a suitable transform for decorrelating signals [1]. The human visual system has been shown to be less sensitive to very high spatial frequencies than lower frequencies in a number of different experiments. An example sentence: [Reference needed] In this case, the citation is required In this case, the citation is required Reconstructing the block in a way that is consistent with the original block is where Inverse DCT truly shines. However, this can only be achieved by employing DCT coefficients with very few frequencies. The degree to which high-frequency DCT coefficients are eliminated affects both compression ratio and text quality. With more coefficients in a block, compression ratio decreases but quality improves, and vice versa.

The same issue is also present in JPEG default quantization tables. These tables reflect extensive visual testing, so the results are consistent with what you see in them. Increasing or decreasing the uncertainty in the top left of the quantization table has the same effect on text quality. Whether there is more or less doubt at the top left is irrelevant to this point. Using the following discussion as input into genetic operators helps to optimize the search, and the resulting DCT and quantization tables are of the highest possible quality.

The initial chromosomes are constructed using KBI and are created at random. The unfitness function is used to assign a ranking to each originating population chromosome. Each chromosome unfitness value, calculated through mathematical analysis, is then utilized to place it in a hierarchy. Selecting superior chromosomes is achieved by employing a low unfitness value. The MSE is computed after the produced text block cluster representations have been quantized and dequantized using superior chromosomes. The KBC procedure is applied to the representative chromosome of the cluster with the highest MSE value. Each cluster has this procedure performed on it.

The unfitness function is used to evaluate the entire offspring population as a whole. These offspring and their parents are screened to see which chromosomes have the best features, and the offspring are then selected for their fitness. The KBSM process further refines these enhanced chromosomes, leading to the birth of new generations. Every kid is given a rating on how unfit they are to take part once more. In KBSM, the existing population is described as the 'father and offspring of the species,' and it is rated according to unfitness. This cycle continues until the target number of generations has been reached through the previous cycles. A chromosome with the highest fitness value and the lowest unfitness value is the best option when the necessary number of generations have passed.

Algorithm: KBGA

Initialisation

Input: N chromosomes,

Output: $m \times n$ chromosomes

Obtain initial chromosomes.

Divide $m \times n$ chromosome into $0.5m \times 0.5n$ tables.

Define range of table from text dataset.

Generate randomly the population individuals.

Find the duplicates and replace the old ones

Evaluation

Input: $m \times n$ tables with λ = compression ratio and $a = 10$

Output: Unfitness value

Apply compression/decompression on table

Find the value of MSE ε and bit rate Br

Apply on the fitness value to compute unfitness value.

Centroid Initialization

Input: K – clusters and N vector set,

Output: Cluster centroids

Sort vectors in an order.

Randomly divide vectors into K.

Calculate mode for bins and find the centroids.

Knowledge-based Feature Selection

Input: Superior chromosomes

Output: Chromosomes for crossover operation

Perform quantization/dequantization on cluster members using superior chromosomes

Find the value of MSE.

Select chromosomes with less MSE on each cluster

Knowledge-based crossover

Input: Superior chromosomes

Output: Offspring

Find the random pairs of chromosomes.

Select the crossover point

Swap the genes between chromosomes.

Replace new ones after finding swapping

Knowledge-based mutation

Input: Superior chromosomes

Output: Offspring

Divide chromosomes into groups.

The group with unfitness values is ranked first and so on

Find duplicates and replace with the new chromosomes.

KBDE Feature Selection

While the DE algorithm excels at global search, it has difficulty with convergence. The convergence rate in DE can be increased by taking advantage of the text features that are already there and by learning how text can be compressed.

The search can be sped considerably by using operators that incorporate knowledge bases based on the DE method and quantization table. To generate $m \times n$ DCT coefficients, the KBDE will perform DCT on $m \times n$ blocks, with the majority of the signal energy being concentrated in the top left corner of the block. It is possible to rebuild the entire block with only a tiny degree of data loss if only a few of the top-left (low-frequency) DCT coefficients are available.

Algorithm of KBDE

Generate chromosomes population;

Evaluate chromosomes;

While Maximum number of Generation do

Consider *N* superior chromosomes using low unfitness value

Select chromosomes with better decryption quality

For a sub-population

For entire chromosomes do

Select the fittest chromosome;

Compute mutant chromosome;

Evaluate trial chromosome;

End

End for

End while

Return best chromosome;

4. Results and Discussions

In this section, the proposed KBGA and KBDE models are tested to find the statistical significance of these models over scalability and computational issues.

Table : Parameters for simulation

Parameters	Value
Population	100
Crossover probability	0.6
Mutation probability	0.015
Total Generation	100
Total independent runs	20

The improvements of the proposed work is compared with existing sentiment analysis model like Genetic based models using the metrics namely precision, recall, F-measure and accuracy. Finally, the statistical significance is tested between the proposed models to check if the model reduces the computational issues and improves the scalability of the model.

The proposed and existing models are tested over various datasets that includes: Amazon Dataset, YELP Dataset, IMDB movie review Dataset and Twitter Dataset to selects the appropriate sentiments. The proposed model is implemented by using python programming in the Google Colab environment.

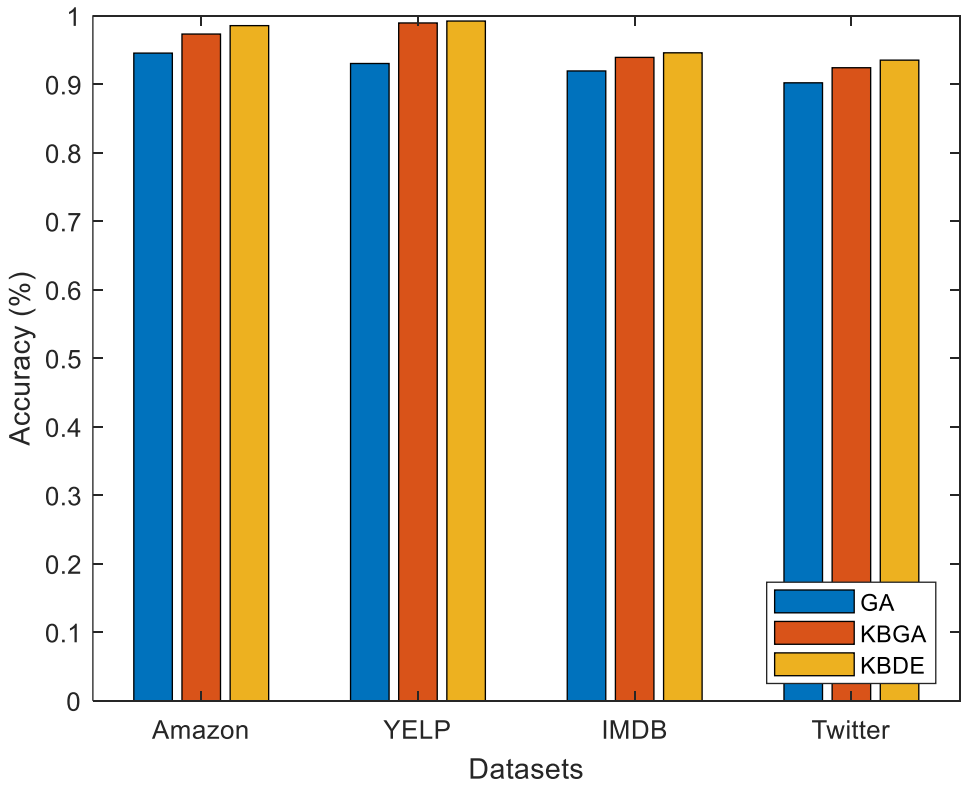

Figure 3: Accuracy - Amazon, IMDB, Yelp and Twitter

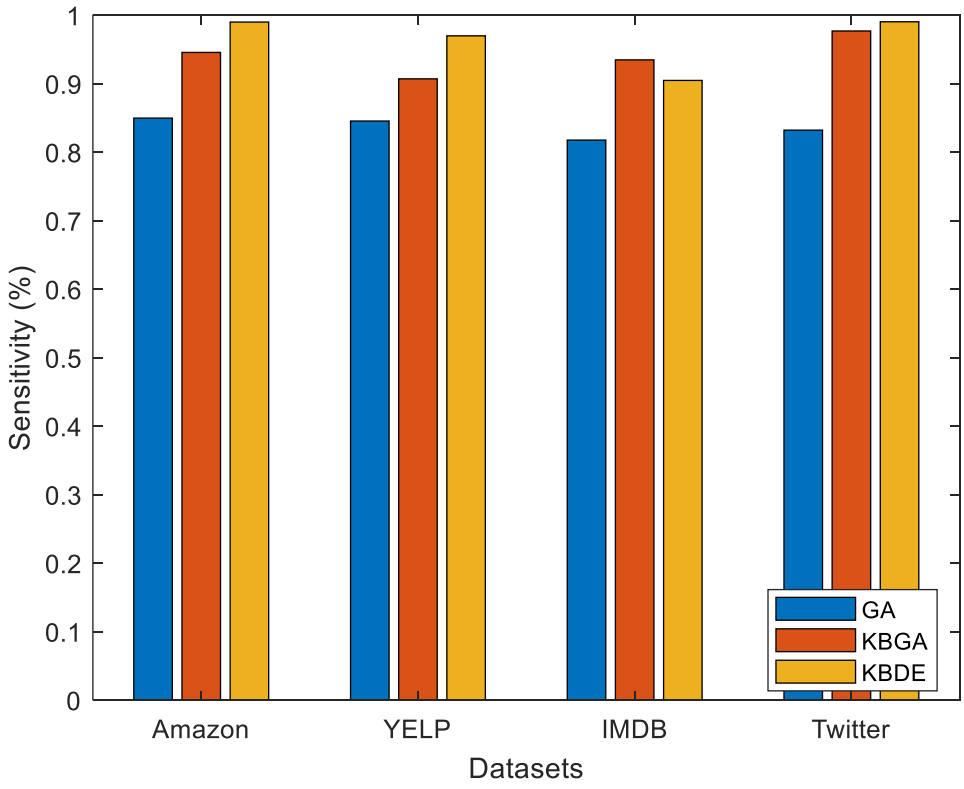

Figure 4: Sensitivity - Amazon, IMDB, Yelp and Twitter

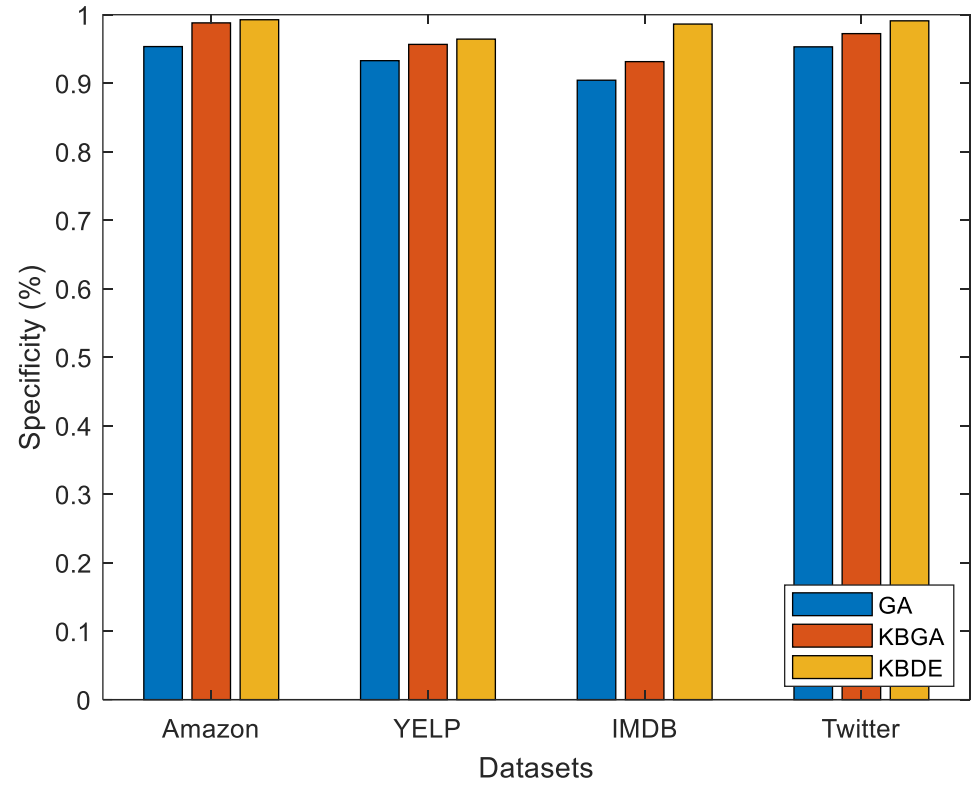

Figure 5: Recall- Amazon, IMDB, Yelp and Twitter

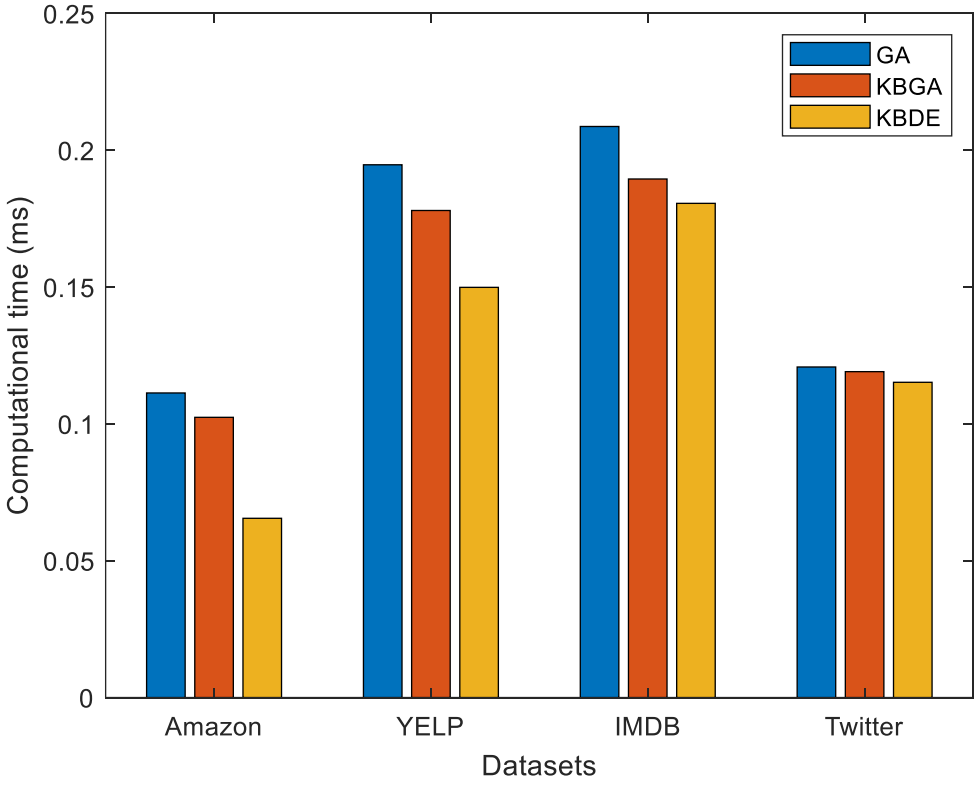

Figure 6: Computation Time comparison with GA, KBGA and KBDE For all datasets with existing models

Dataset	Dataset	Method	Sensitivity	Specificity	Accuracy	Computational time
Training	Amazon	GA	0.8499	0.9535	0.9457	0.0655
		KBGA	0.9458	0.9881	0.9735	0.1024
		KBDE	0.9901	0.9927	0.9858	0.1113
Training	YELP	GA	0.8456	0.9329	0.9305	0.1347
		KBGA	0.9072	0.9567	0.9897	0.1499
		KBDE	0.97	0.9644	0.9925	0.178
Training	IMDB	GA	0.8178	0.9044	0.9196	0.1806
		KBGA	0.9349	0.9315	0.9394	0.1895
		KBDE	0.9049	0.9864	0.9461	0.4587
Training	Twitter	GA	0.8323	0.9531	0.9023	0.1152
		KBGA	0.9771	0.9724	0.9243	0.1191
		KBDE	0.9905	0.9911	0.9354	0.1208

From the results of Figure 3-6, it is seen that the proposed method achieves higher rate of accuracy in selecting the features than the KBGA and GA. The selection of features using DE has a higher accuracy rate than the KBGA and existing genetic algorithm.

5. Conclusions

In this paper, we adopt KBGA and KBDE algorithm to select the features from the four different datasets. The results shows that the KBDE offers increased feature selection ability than the existing methods. From the results of simulation over various datasets, it is found that the proposed method achieves higher rate of accuracy in finding the proposed samples than the conventional Genetic Algorithms.

References

[1] Langley, D. J. (2022). Digital Product-Service Systems: The Role of Data in the Transition to Servitization Business Models. *Sustainability*, *14*(3), 1303.

[2] Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*, 107134.

[3] Malekloo, A., Ozer, E., AlHamaydeh, M., & Girolami, M. (2022). Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, *21*(4), 1906-1955.

[4] Suzuki, K., Laohakangvalvit, T., Matsubara, R., & Sugaya, M. (2021). Constructing an emotion estimation model based on EEG/HRV indexes using feature extraction and feature selection algorithms. *Sensors*, *21*(9), 2910.

[5] Ozyurt, F., Tuncer, T., & Subasi, A. (2021). An automated COVID-19 detection based on fused dynamic exemplar pyramid feature extraction and hybrid feature selection using deep learning. *Computers in Biology and Medicine*, *132*, 104356.

[6] Kernbach, J. M., & Staartjes, V. E. (2022). Foundations of Machine Learning-Based Clinical Prediction Modeling: Part II—Generalization and Overfitting. *Machine Learning in Clinical Neuroscience*, 15-21.

[7] Wahyudi, Mochamad, and Dwi Andini Putri. "Algorithm application support vector machine with genetic algorithm optimization technique for selection features for the analysis of sentiment on twitter." *Journal of Theoretical and Applied Information Technology* 84.3 (2016): 321.

[8] Xue, Y., Zhu, H., Liang, J., & Słowik, A. (2021). Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. *Knowledge-Based Systems*, *227*, 107218.

[9] Abualigah, L., & Dulaimi, A. J. (2021). A novel feature selection method for data mining tasks using hybrid sine cosine algorithm and genetic algorithm. *Cluster Computing*, *24*(3), 2161-2176.

[10] Lappas, P. Z., & Yannacopoulos, A. N. (2021). A machine learning approach combining expert knowledge with genetic algorithms in feature selection for credit risk assessment. *Applied Soft Computing*, *107*, 107391.

[11] O'Keefe, T., & Koprinska, I. (2009, December). Feature selection and weighting methods in sentiment analysis. In *Proceedings of the 14th Australasian document computing symposium, Sydney* (pp. 67-74).

[12] Tan, S., & Zhang, J. (2008). An empirical study of sentiment analysis for chinese documents. *Expert Systems with applications*, *34*(4), 2622-2629.

[13] Morinaga, S., Yamanishi, K., Tateishi, K., & Fukushima, T. (2002, July). Mining product reputations on the web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 341-349).

[14] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *In: Proceedings of the ACL-02 conference on empirical methods in natural language processing - Vol 10,EMNLP '02, pp 79–86*.

[15] Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

[16] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*.

[17] Dos Santos, C., & Gatti, M. (2014, August). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers* (pp. 69-78).

[18] Manurung, R. (2008). Machine learning-based sentiment analysis of automatic indonesian translations of english movie reviews. In *Proceedings of the International Conference on Advanced Computational Intelligence and Its Applications (ICACIA)* (pp. 1-6).

[19] Zhang, L. X., Wang, J. X., Zhao, Y. N., & Yang, Z. H. (2003, November). A novel hybrid feature selection algorithm: using ReliefF estimation for GA-Wrapper search. In *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)* (Vol. 1, pp. 380-384). IEEE.

[20] Apolloni, J., Leguizamón, G., & Alba, E. (2016). Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, *38*, 922-932.

[21] Hsu, H. H., Hsieh, C. W., & Lu, M. D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, *38*(7), 8144-8150.

[22] Zhang, Y., Wang, S., Sui, Y., Yang, M., Liu, B., Cheng, H., ... & Gorriz, J. M. (2018). Multivariate approach for Alzheimer's disease detection using stationary wavelet

entropy and predator-prey particle swarm optimization. *Journal of Alzheimer's Disease*, *65*(3), 855-869.

[23] Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, *53*, 453-462.

[24] Manurung, R. (2008). Machine learning-based sentiment analysis of automatic indonesian translations of english movie reviews. In *Proceedings of the International Conference on Advanced Computational Intelligence and Its Applications (ICACIA)* (pp. 1-6).

[25] Nicholls, C., & Song, F. (2010, May). Comparison of feature selection methods for sentiment analysis. In *Canadian Conference on Artificial Intelligence* (pp. 286-289). Springer, Berlin, Heidelberg.