International Journal of Scientific & Engineering Research Volume 10, Issue 3, March-2019
ISSN 2229-5518

Parameter Tuning in Decision Tree Based on Genetic Algorithm for Text Classification

Ahmed I. Taloba, Adel A. Sewisy, Safaa S. I. Ismail

Abstract— One of the most complicated tasks in text mining field is text classification. In this paper, we propose a new efficient hybrid of the decision tree algorithm (DT) and a genetic algorithm (GA) approach known as (GADT) for text classification. This approach focuses on increasing the performance of DT approach to categorize texts into one or more predefined classes according to their contents. There are several parameters of decision tree algorithm that have to be adapted in order to get optimal classifier. One of these parameters is the confidence factor that has a great effect on the classification accuracy. In our work, we propose to use genetic algorithm to find the optimal value of confidence factor of decision tree approach and this is an important step for improving decision tree algorithm performance for text classification. Thus, the optimal value of confidence factor helps to construct an optimal, accurate and small, decision tree for achieving high quality results. Our experimental results indicate that our proposed approach (GADT), in comparison to traditional DT algorithm alone (with default value of confidence factor) gives more achievable and efficient results in a number of cases of text datasets.

Index Terms— Text Mining, Text Classification, Decision Tree, Confidence Factor, Genetic Algorithms, Hybrid Technique.

1 INTRODUCTION

A huge amount of electronic data is available such as digital libraries, electronic books, electronic newspaper, electronic publications, emails, etc. Also, these electronic data are rapidly increasing which raises the level of challenge to manage that data. The main aim of text mining is to enable users to extract useful information from texts and deals with the tasks like summarization, classification and retrieval. One of the most important tasks of text mining is text classification (TC).

Text classification (also known as text categorization) is the process of assigning the documents with pre-defined categories. The task of classifying millions of text document manually is an expensive and time-consuming task. Thus, automatic management of texts becomes an essential research issue of text mining. Many traditional supervised machine learning approaches have been developed to deal with automatic text classification [1] including decision tree (DT), random forest (RF), K-nearest neighbor (KNN), naive bayes (NB), support vector machine (SVM) and hybrid techniques. Hybrid techniques are both combination of two or more different classifiers. Researchers have proved that using hybrid techniques improves the performance of each individual technique and compensates for each other's weakness [2]. The hybrid classification techniques handle the classification tasks by utilizing the strengths of the used classifiers.

Among these classifiers, we use decision tree that has proved superiority experimentally when compared with other widely used text classifiers. In addition, decision tree is generally fast even with large datasets as text datasets. It also can create non-linear decision

Boundaries that fit large data very well. As the decision tree grows, it naturally tends to cause data overfitting. Pruning is a process by which the size of the decision tree is reduced through eliminating sections of the tree that provide little power to classify instances.

Pruning decreases the complexity of the final classifier, and hence improves the classifier accuracy through the reduction of data overfitting. In our work we use another approach, genetic algorithm, to find the optimal value of the parameter of confidence factor which determines how aggressive the pruning process will be. This parameter avoids unnecessary complexity and helps in optimizing the classification accuracy.

Decision tree (DT) is one of the most well-established classifiers based on their transparency in describing rules that lead to a prediction [3]. Decision tree doesn't need any domain knowledge for its construction, it is able to handle both numerical and categorical data. It also is robust as it can handle noise and high dimensional data as text data. Also, it is well known that the construction of DTs and the interpretation of the resulting model to classification rules is an easy work. This makes them extremely useful tools for many real-world problems for classification [4] such as text classification problems. One of the main drawbacks of decision tree classifiers is that they are sensitive to noisy data and that multiple output attributes are not allowed [5].

There are many algorithms for building decision tree models such as CHAID [6], ID3 [7], CART [8], C4.5 [9] and C5.0 [10]. In our work, we use the java implementation of the C4.5 algorithm which is the J-48 algorithm [11]. Tree building and tree pruning [11],[12] are the two steps of decision tree modeling.

- *Ahmed I. Taloba is in Faculty of Computer and Information, Assiut University, Egypt, E-mail: Taloba@aun.edu.eg*
- *Adel A. Sewisy is in Faculty of Computer and Information, Assiut University, Egypt, E-mail: dr.adelsewisy@yahoo.com.*
- *Safaa S. I. Ismail is in faculty of Science, New Valley University, Egypt, E-mail: safaasobhy1982@scinv.au.edu.eg*

In the building step, the obtained tree may have a large number of branches which may cause overfitting. Therefore, the tree needs to be pruned for better classification accuracy of new data.

Genetic algorithm (GA) is an adaptive heuristic search algorithm premised on the evolutionary mechanism of natural selection [13]. GA is usually good for searching for near optimal value in complex and large landscapes. Also, GA is more useful in highly dimensional data such as text datasets.

The key advantage of our new hybrid technique is the high accuracy from a powerful supervised learning algorithm, decision tree. This work compares the performance of standard DT classifier (with default value of confidence factor) with that of our new hybrid technique GADT on a set of well-known textual datasets. The experimental results are conducted on seven different text datasets that will be described below.

The rest of our paper is organized as follows. A brief presentation of related work to the use of hybrid techniques for text classification is introduced in Section 2. Section 3 provides some fundamental concepts about genetic algorithm and decision tree that are the main components of our (GADT) technique. Our new hybrid technique (GADT) that improves the performance of decision tree algorithm is described in Section 4. In Section 5, we provide our experimental results and a comparison against standard decision tree algorithm. Finally, in Section 6 we summarized the main results followed by references.

2 Related Works

In recent years, the field of text classification attracted a lot of interest and has been studied by many researchers. In this section, several hybrid algorithms for text classification are described.

In [14], a novel hybrid parallel architecture using different types of classifiers is combined and trained on different subspaces giving a significant performance improvement over single classifier working on full data space. The features are extracted dynamically by using maximum significance values. The experiments are run on the Reuters and LSHTC corpora. Also, it shows that the improvement of accuracy and training time of the hybrid classifier is higher with the LSHTC corpus than with the Reuters corpus.

In [15], a hybrid neural classifier fully integrated with a novel boosting algorithm is used for text classification task in a non-stationary environment. This hybrid technique is conducted on the modapte version of the Reuters news text corpus. Results show the improved performance of the hybrid neural classifier even with minimal number of neurons in constituting structures. Also results show that the performance of the classifiers did not just occur by chance, but based on learned information.

In [16], a combination of both classifiers, Nave Bayes and Maximum Entropy, is used for text classification task. Results, in this paper, show that this combination of these classifiers gives better performance than the individual ones. First, these classifiers are used separately on the dataset and then, combined the results of both the classifiers using combining operation like Mean, Max, and Average etc. in order to get more accurate results than any of the single classifier. Results show that the best accuracy is obtained by Max combining operator.

In [17], Nave Bayes and the modified versions of Maximum Entropy are combined using three merging operators for text classification. Nave Bayes is extremely simple and fast technique. The Maximum Entropy classifier gives a great deal of flexibility for parameter definitions and follows assumptions closer to real world scenario. The combination of these classifiers is done through operators like Mean, Max, and Average etc. that linearly combine the results of the two classifiers to predict a class of documents in query. Results show that this combination achieves more accuracy.

3 Background

Our proposed technique GADT is based on DT and GA. In the following section, the necessary background on these two techniques are presented.

3.1 Decision Tree (DT)

Decision tree classifiers are used to give a rapid and effective solution for classifying instances in high dimensional datasets with a large number of features such as text datasets [18].
Decision tree is a flowchart as the tree representation, in which each leaf node represents an attribute (feature) in an instance to be classified. Each branch represents a value of which the node can be assumed. As we mentioned above, we use J-48 algorithm [19] that builds the decision tree from labeled training dataset using information gain. It examines the same that results from choosing an attribute for splitting the data. To make the decision, the attribute with highest normalized information gain is used.
Then, the algorithm recurs on smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Finally, the leaf node is created in a decision tree telling to choose that class.

Pruning is removing non-predictive parts of the decision tree that cause overfitting in the training data. Pruning reduces the tree size to improve the accuracy and the comprehensibility of the resulting classifier. Ideally, pruning should only eliminate those parts of the tree that are due to noise, and never eliminate any structure that is truly predictive. Pruning should never remove predictive parts of a classifier. The parameter known as confidence factor is used for pruning versus classification accuracy. Confidence factor is one of the most essential parameters of J-48 decision tree which decides when to stop the expanding of the tree.

3.2 Genetic Algorithm (GA)

The genetic algorithm is a major metaheuristic approach that simulates the phenomena of natural evolution. Comprehensive details about GA can be found in [13]. An iterative stochastic process is followed such that high-quality or exact solutions are found by depending on bio-inspired operators such as mutation, crossover and selection. Each solution in the population is encoded and associated to a score with regard to fitness function. In each generation, two parents are selected based on their corresponding fitness value. These chromosomes are used by crossover and mutation operations to produce two offspring for the new generation. After several generations, the optimal solutions that have better fitness are selected. The algorithm of standard GA approach is as following:

1) Choose an initial population of chromosomes;
2) Evaluate the initial population;
3) While termination condition not satisfied do
    Select the current population from the previous one;
    If crossover condition satisfied, perform crossover;
    If mutation condition satisfied, perform mutation;
    Evaluate fitness of offspring;
  End while

The goal of setting optimal parameter (confidence factor) value in DT classifier is to achieve the highest classification accuracy. In our work, we use GA that meets the need of our parameter optimization.

4 Our Proposed Method

In this section, we propose our new hybrid algorithm GADT to improve the text classification accuracy. The processes of GADT is presented in Fig. 1.

Pruning has a great effect in improving the efficiency of the decision tree. In our algorithm, we employ GA to find the optimal value of a parameter named confidence factor that control the pruning of our decision tree.

The idea of our proposed algorithm GADT is that we firstly use J-48 to generate the text classification rules and then according to accuracy we build the fitness function of GA to find the optimal value of confidence factor. The larger the value of the fitness is, the more the optimal value of confidence factor will be.

The detailed explanation for each phase of our hybrid algorithm GADT is described in the following subsections:

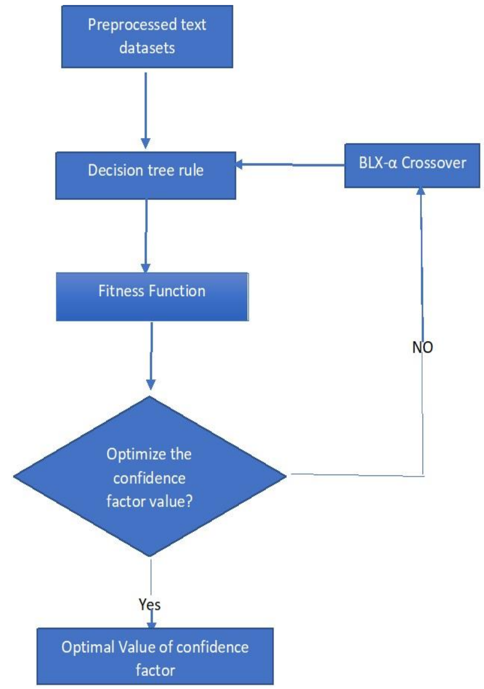

Fig.1 Processes of GADT algorithm

4.1 Text Preprocessing for Classification

After collecting texts as raw data, the preprocessing phase is applied in order to present the collected texts into clear word format where the classifiers can be used. The presentation of raw texts into sequence of words is known as Tokenization is used to present raw texts into sequence of tokens. Then, the stop words (a, an, is, are and etc.) are removed. After removing the stop words, stemming is applied where the words are returned to their root such as the words″connecting, connects, connected, connection″ which have the same stem″ connect″. In our work, we use one of the most known stemming algorithms, Porters algorithm [20. This algorithm removes the commoner morphological and in flexional affixes from words. After these steps, the texts are represented as″Bag of Words (BOW)″ and now texts are ready for″ indexing″ or″ term weighting″. In indexing step, the bag of words that is extracted from the raw documents is converted to a feature vectors where each feature is a word (term) and the feature's value is a term weight. One of the most popular used approaches for indexing is, that we use in our work, TF_i . IDF_i (term frequency - inverse document frequency) scheme [21]. We use this scheme to evaluate the importance of each word (weight) in the document (W_i). Accordingly, let tf_i (term frequency) be the number of occurrences of t_i in the document and df_i (document frequency) be the number of the document in which the t_i term is seen at least once. The idf_i (inverse term frequency) is calculated by the following equation:

$$IDF = \left(\frac{|D|}{df_i}\right) \qquad (1)$$

Where D is the number of all documents in the training sets and the weight of each word in the d document is given by:

$$W_i = tf_i \, . idf_i \qquad (2)$$

After preprocessing phase, the original textual data are represented as a feature vector where the most significant features used to differentiate text classification are identified. This representation reduces the complexity of the documents and make them easier to handle by the classifier.

4.1 Finding Optimal Value of Confidence Factor using GA

When we use GA for finding optimal value of confidence factor our problem must first be adapted to the genetic algorithm where the basic components of GA such as chromosome and population should be determined. In order to use GA, one must define the various steps of GA s follow:

- Problem Encoding and Initial Population: In our work, real value representation is used here for encoding the chromosomes. Each chromosome is a vector of decimal number between 0 and 1. The initial population was generated randomly (random values of confidence factor).
- Genetic Operators: In order to guide the genetic algorithm towards the optimal value of confidence factor, we adapt two major operations that we use in GA technique. These operations are selection and crossover.

Selection is the process of choosing individuals in the current generation for crossover operator to obtain the offspring in the next generation. In our implementation, we use one of the most popular selection methods for select offspring for crossover which is the Roulette Wheel selection scheme.

The crossover operation of genetic is used to adjust the fitness function, so the fitness value will reach to the maximum value, and the value of confidence factor will be optimization.

In the crossover phase, two individuals are selected for mated to create offspring. In our implementation, the blend crossover scheme (BLX-α) is used. BLX-α is one of the most popular crossover schemes that used with real representation. It shows good search ability and perform better for this problem domain [22],[23]. In general, the better performance of BLX-α is due to its implementation that allows for generating offspring in the neighborhood outside the two parent solutions as well rather than taking only the interpolated values between two points. This is helpful when there exist several solutions to the problem.

The algorithm of BLX-α [24] is as follows:

1) Select two parents X^1 and X^2 from the initial population.
2) Create two offspring X^{t+1} and Y^{t+1} as follows:
   For i=1 to n do
   $d_i = |X_i^t - Y_i^t|$
   Choose a uniform random real number
   $u \in < \min(x_i^t, y_i^t) - \alpha d_i, \max(x_i^t, y_i^t) + \alpha d_i > \; x_i^{t+1} = u$
   Choose a uniform random real number
   $u \in < \min(x_i^t, y_i^t) - \alpha d_i, \max(x_i^t, y_i^t) + \alpha d_i > y_i^{t+1} = u$
   End do // Where α = {0.2,0.5}.

- Fitness Function: The fitness function is used to evaluate the quality of different values of the confidence factor represented by the individuals. In order to evaluate the optimal value of the confidence factor of J-48 decision tree classifier, the fitness of each new value is computed. The fitness function takes the value of confidence factor as an input and produces as output the accuracy of J-48 decision tree classifier. Every individual of higher value of fitness function has more chance to appropriate as a problem solution.

4.3 Evaluation Method of GADT

Several measurements are used to evaluate the performance of our hybrid technique GADT for text classification which are f-measure, recall, precision and accuracy. These popular measurements are mostly used to evaluate the efficiency of the classifier for text classification. There are four cases as a result of a classifier for text classification which are:

TP (True Positive): the number of documents that are correctly classified to that class.
FP (False Positive): the number of documents that are not correctly classified to that class.
TN (True Negative): the number of documents that are correctly classified to the other class.
FN (False Negative): the number of documents that are not correctly classified to the other class.

The f-measure, recall and precision measurements are the most popular measurements which are used to evaluate the accuracy of the result of a classifier for text classification. Precision is the proportion of correctly proposed documents to the proposed document and it is given in equation 3. Recall is the proportion of the correctly proposed documents to the test data that have to be proposed and it is given in equation 4. F-measure is the combination of both precision and recall used in information retrieval and it is given in equation 5.

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

$$Recall = \frac{TP}{TP + FN} \qquad (4)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \qquad (5)$$

5 Experimental Results and Discussion

In the experimental work, efficiency of our new hybrid technique GADT is thoroughly investigated and compared to standard J-48 decision tree approach in the literature.

5.1 Text Datasets

We evaluate the performance of our new hybrid technique GADT for text classification on seven UCI text datasets collection [25] which are described in Table 1. Each text dataset is roughly equivalent to a two-dimensional spreadsheet.

TABLE 1 MAIN CHARACTERISTICS OF THE TEXT DATASETS IN OUR EXPERIMENTS

Text dataset	No. of Attr.	Attr. Types	Instances	Classes
CNAE-9	857	Integer	1080	9
DBWorld e-mails	4702	Nominal	64	10
Classic 03	100	Nominal	3830	3
Oh0	3183	Integer	1003	10
Oh5	3013	Integer	918	10
Oh10	3229	Integer	1050	10
Oh15	3101	Integer	913	10

Experiments are run using a machine Intel Core i7 with 16 GB of RAM, 1.8 GHz CPU, and Windows 10 operation system. Preprocessing, using GA and classification processes are implemented by C sharp software. The code for the basic versions of the J-48 decision tree classifiers is adopted from Weka3, which is open source data mining software [26]. Weka is a collection of machine learning algorithms for different data mining tasks. The classifiers in Weka3 can be either applied directly to a dataset or called from our own coding.
In our approach GADT, GA has population of 50 individuals evolving during 20 generations. For crossover, we use BLX-α crossover where α parameter equal 0.36 for all our text datasets. The presented experimental results are the best of 10 runs. The output chromosomes of values of confidence factor are sorted ascendingly according to their fitness.

5.2 RESULTS

We have extensively evaluated the performance of both J-48 decision tree and GADT across seven different text datasets. Initially J-48 decision tree classifier is applied with the default value of confidence factor on our text datasets. We use 10-fold cross-validation in evaluating the performance of the algorithms. The experimental results of using J-48 decision tree (with default value of confidence factor which is 0.25) on our seven text datasets are summarized in Table 2. As shown in Table 2, Classic 03 corpus gives the highest accuracy with J-48 classifier.

TABLE 2 THE PERFORMANCE ANALYSIS OF J-48 DECISION TREE CLASSIFIER WITH DEFAULT VALUE OF CONFIDENCE FACTOR ON OUR SEVEN CORPUSES.

Text dataset	Value of confidence factor	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
Classic 03	0.25	95.0	95.5	95.3	94.7258
CNAE-9	0.25	90.4	88.8	89.1	88.7963
DBWorld	0.25	81.6	75.0	74.3	75
Oh0	0.25	82.1	80.5	80.8	80.4586
Oh5	0.25	83.4	82.2	82.6	82.244
Oh10	0.25	72.3	71.1	71.3	71.1429
Oh15	0.25	76.1	74.8	75.3	74.8083

TABLE 3 THE PERFORMANCE ANALYSIS OF J-48 DECISION TREE CLASSIFIER WITH OPTIMAL VALUE OF CONFIDENCE FACTOR ON OUR SEVEN TEXT DATASETS

Text dataset	Value of confidence factor	Precision (%)	Recall (%)	F-measure (%)	Accuracy (%)
Classic 03	0.44	95.7	95.5	95.6	94.9869
CNAE-9	0.391	90.4	88.9	89.2	88.8889
DBWorld	0.516	81.6	79.7	79.7	79.6875
Oh0	0.221	82.3	80.6	80.9	80.5583
Oh5	0.107	83.9	82.5	82.8	82.4619
Oh10	0.033	74.0	73.0	73.1	72.9524
Oh15	0.06	76.3	74.9	75.4	74.9179

Furthermore, GADT is applied across our seven text datasets and the results of using GADT across the text datasets are reported in Table 3. Generally, Table 3 shows that using our hybrid technique GADT largely affected the overall performance of our different text datasets positively. The comparison of the classification accuracy of GADT and J-48 decision tree classifiers is shown graphically in Fig. 2. The figure shows that the accuracy of all text datasets we use is improved with GADT classifier.

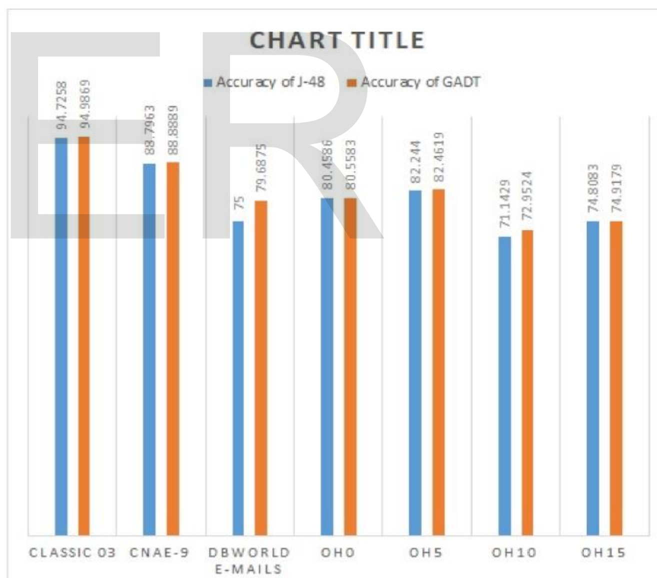

FIG. 2 COMPARISON GRAPH FOR ACCURACY OF OUR HYBRID APPROACH GADT AND STANDARD J-48 DECISION TREE

If Table 2 is compared with Table 3, we can see that among all datasets the highest accuracy is given with Classic 03 text dataset. Furthermore, it is seen that using Classic 03 text dataset with optimal value of confidence factor (0.44) instead of the default value (0.25) positively contributed to the J-48 classifier performances in an affirmative manner. Totally after analyzing Tables 2 and 3, it is seen that using optimal value of confidence factor instead of using default value, with all text datasets, affects performances in positive manner.

6 CONCLUSIONS

In this work, a hybrid text classification method, that we termed as GADT based on combination of decision tree and genetic algorithm, is introduced. In GADT, we use genetic algorithm to find optimal value of a parameter namely confidence factor. Confidence factor not only reduces the tree size but also helps in filtering out statistically irrelevant nodes that would otherwise lead to classification errors. Finding optimal value of confidence factor parameter of decision tree classifier improves its performance and generate the best classifier to classify a new text data.

In this paper, we have compared the performance of GADT approach with the performance of standard J-48 decision tree classifier alone. Our experiments are conducted on seven UCI text datasets. Results show the performance improvement of GADT approach and its ability to handle highly dimensional data as texts.

As a future work, GADT can be applied to different applications related to text classification Furthermore, various evolutionary computing and decision tree algorithm can be combined in order to evaluate the proposed approach with different methodologies and find more accurate solutions for text classification.

REFERENCES

[1] C. Kruengkrai and C. Jaruskulchai, “A parallel learning algorithm for text classification,” in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 201–206, 2002.

[2] A. Mudinas, D. Zhang, and M. Levene, “Combining lexicon and learning based approaches for concept-level sentiment analysis,” in Proceedings of the first international workshop on issues of sentiment discovery and opinion mining. ACM, p. 5, 2012.

[3] A. J. Myles, R. N. Feudale, Y. Liu and et al., “An introduction to decision tree modeling,” Journal of Chemometrics: A Journal of the Chemometrics Society, vol. 18, no. 6, pp. 275–285, 2004.

[4] S. K. Murthy, “Automatic construction of decision trees from data: A multi-disciplinary survey,” Data mining and knowledge discovery, vol. 2, no. 4, pp. 345–389, 1998.

[5] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects,” Advances in Space Research, vol. 41, no. 12, pp. 1955– 1959, 2008.

[6] M. J. Berry and G. Linoff, “Data mining techniques: for marketing, sales, and customer support,” John Wiley & Sons, Inc., 1997.

[7] J. R. Quinlan, “Induction of decision trees,” Machine learning, vol. 1, no. 1, pp. 81–106, 1986.

[8] L. Breiman, “Classification and regression trees”. Routledge, 2017.

[9] J. R. Quinlan, “C4.5: programs for machine learning”. Elsevier, 2014.

[10] T. Bujlow, T. Riaz, and J. M. Pedersen, “A method for classification of network traffic based on c5. 0 machine learning algorithm,” in Computing, Networking and Communications (ICNC), 2012 International Conference on. IEEE, pp. 237–241, 2012.

[11] J. H. Cho and P. U. Kurup, “Decision tree approach for classification and dimensionality reduction of electronic nose data,” Sensors and Actuators B: Chemical, vol. 160, no. 1, pp. 542–548, 2011.

[12] A. Franco-Arcega, J. A. Carrasco-Ochoa, G. S´anchez-D´ıaz, and et al., “Decision tree induction using a fast splitting attribute selection for large datasets,” Expert Systems with Applications, vol. 38, no. 11, pp. 14 290–14 300, 2011.

[13] E. D. Goodman, “Introduction to genetic algorithms,” in Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM, pp. 205–226, 2014.

[14] N. Tripathi, M. Oakes, and S. Wermter, “Semantic subspace learning for text classification using hybrid intelligent techniques,” Int. J. Hybrid Intell. Syst., vol. 8, no. 2, pp. 99–114, Apr. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2010580.2010585

[15] E. Buabin, “Boosted hybrid recurrent neural classifier for text document classification on the reuters news text corpus,” International Journal ofMachine Learning and Computing, vol. 2, no. 5, p. 588, 2012.

[16] K. Hanika and B. Bala, “Combining nave bayes and modified maximum entropy classifiers for text classification,” Information Technology andComputer Science, vol. 9, pp. 32–38, 2016.

[17] M. Shah, S. Shinde, R. S. Sawant, and et al., “Analysis of text review using hybrid classifier,” International Journal of Engineering Science, vol. 10914, 2017.

[18] D. Lafond, Y. Lacouture, and A. L. Cohen, “Decision-tree models of categorization response times, choice proportions, and typicality judgments.” Psychological Review, vol. 116, no. 4, p. 833, 2009.

[19] I. H. Witten, E. Frank, M. A. Hall, and et al., “Data Mining: Practical machine learning tools and techniques,” Morgan Kaufmann, 2016.

[20]] C. J. Rijsbergen, S. E. Robertson and M. F. Porter, “New models in probabilistic information retrieval,” London: British Library, 1980.

[21] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” Information processing & management, vol. 24, no. 5, pp. 513–523, 1988.

[22] L. J. Eshelman, “Crossover operator biases: Exploiting the population distribution,” in proc. of Int. Conf. on Genetic Algorithms, 1997.

[23] L. J. Eshelman and J. D. Schaffer, “Real-coded genetic algorithms and interval-schemata,” in Foundations of genetic algorithms. Elsevier, vol. 2, pp. 187–202, 1993.

[24] T. D. Gwiazda, “Genetic algorithms reference,” 2006.

[25] D. Dheeru and E. Karra Taniskidou, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, and et al., “The weka data mining software: an update,” ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18, 2009.