Optimization of opinion mining classification techniques using dragonfly algorithm

Mikanshu Rani, Jaswinder Singh
Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India

Article Info

Article history:

Received Oct 8, 2023
Revised Mar 1, 2024
Accepted Mar 14, 2024

Keywords:

Decision tree
Deep neural networks
Dragonfly algorithm
Naive bayes
Opinion mining
Random forest
Sentiment analysis

ABSTRACT

With the rapid evolution and growth of the internet, many individuals are using websites, blogs, and social media, and sharing their opinions about any product or service on online social platforms. Opinion mining (OM) is a field of analyzing opinions or reviews given by the public about services or products on online resources into positive, negative, or neutral classes. Governments, businesses, and researchers are using OM to analyze the reviews or opinions of the public. Thus, OM is helping individuals and businesses in better decision making. This paper mainly focuses on the feature extraction, performance analysis of OM classifiers and optimization using swarm intelligence (SI). Our proposed work: i) evaluates the performance of OM classification techniques after data collection, pre-processing, and feature extraction, ii) applies the dragonfly algorithm (DA) for optimization, and iii) evaluates the performance of OM classification techniques after applying DA and compares it with the observed performance of OM classifiers before optimization. The experimental results show that OM classification techniques perform better after optimization using DA in terms of precision, recall, f-score, and accuracy.

Corresponding Author:

Mikanshu Rani
Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology
Hisar, Haryana, India
Email: mikanshu6406@gmail.com

1. INTRODUCTION

The fast growth of the internet and online resources like social media applications and e-commerce sites, gave rise to the presence of enormous amounts of textual data as reviews and opinions expressed by people about products or services on the internet. Since analyzing this vast data manually is unachievable, some technique is needed that can extract the polarity of opinions and reviews from this textual data. So, opinion mining (OM) can be used to classify the user-generated data using different classifiers [1], [2]. OM is the fastest growing field because of the internet and the world wide web. OM, also called sentiment analysis (SA) is a technique to classify opinions, and sentiments expressed by people towards objects like products and organizations into positive, negative, or neutral classes; and helps in decision-making [3]. The task of OM seems to be very easy at first, but it is very challenging and also very useful. OM is a field of analyzing textual data to gain information about the mood of individuals towards specific objects [4].

The purpose of this work is mainly to analyse the performance of OM classifiers and apply dragonfly algorithm (DA) to improve the performance of OM classifiers. DA has the ability to optimize real-world problems. For this work, three datasets are collected from Kaggle, and data is cleaned and preprocessed to make it ready for analysis. Further, feature extraction is applied to extract the relevant features from the datasets and the performance of OM classifiers is evaluated. For optimization, DA is designed and applied, which improves the performance of OM classifiers.

Birjali *et al*. [5] presented a survey on SA, its approaches, and the fields related to it. They compared the classification approaches of SA and also presented the applications of SA and challenges observed while doing research. Mudgil *et al*. [6] proposed an analysis model using the grasshopper optimization algorithm (GHO). They combined swarm intelligence (SI) and machine learning (ML) by introducing a reward mechanism (GHO) based on SI and validating it using ML techniques. Elangovan and Subedha [7] proposed firefly and levy flight models for extracting features from online reviews and applied multilayer perceptron (MLP) to classify the sentiments of the DVD dataset. Hendardi *et al*. [8] discussed SI and DA which work according to the swarming nature of dragonflies. They presented the implementation of DA, its pros, and cons, and discussed in detail a review of DA in 2020. While these earlier studies have explored the impact of GHO, and firely and levy flights models on the performance of OM or SA classifiers for one dataset, they have not explicitly addressed their influence on the performance of OM classifiers for more than one dataset.

This research aims to analyze the performance of OM classifiers and improve their performance with SI. Our proposed framework combines the strengths of SI and ML and led to the following key contributions: i) the performance of different OM classifiers is analyzed/evaluated; ii) DA is introduced for optimization which is based on the concept of SI; and iii) the performance of OM classifiers is evaluated after optimization using DA and compared with the performance of OM classifiers before optimization.

In this section, we presented the introduction of OM, research work of other researchers and the contributions of our research. The remaining work is organized as: section 2 discusses the method used, section 3 presents the proposed work for this research, section 4 covers the results and discussions, and the last section 5 concludes our work.

2. DRAGONFLY ALGORITHM

Data optimization is done to find the best solution from different possible solutions. SI is an optimization technique that finds the best solution or optimizes the problems [9]. SI is a part of computational intelligence and is influenced from the natural behavior of dragonflies, ants, and fishes, fighting for their life [8], [10]. The DA is a swarm optimization technique developed by Mirjalili in 2016. It is based on the swarming behavior of dragonflies for their life existence [11], [12]. Swarm movements of the dragonfly are divided into five segments, these are [8], [12]: i) separation: separation means static collision avoidance with others in neighbourhood; ii) alignment: alignment is the individual's velocity matching to neighbours in that group; iii) cohesion: it is the tendency of swarm towards the center of the group; iv) the attraction towards the food source: this is the search agent's reaction or response when a dragonfly has found the food; and v) the distraction from the enemy: this is the response that occurs when a search has found some surrounding objects that are a threat to them. DA examines the dynamic and static behavior of dragonflies, which build the exploitation and exploration phases of SI. Balancing these two phases effectively is crucial for the success of the DA. These phases of DA are briefly described [12], [13]:

– Exploration phase: in exploration, dragonflies make groups that are small in size and fly to a shorter distance to search for food and attract flying prey. During the exploration phase, the dragonflies in the swarm have the goal of covering as much as possible search space to find regions that can contain optimal or near-optimal solutions. By following certain strategies, dragonflies explore new areas in the search space.
– Exploitation phase: in exploitation, a huge number of dragonflies causes the swarm to move away in one direction, distracting the enemy. The exploitation phase aims to refine the information gathered for exploiting the most promising solutions discovered during the exploration phase.

3. PROPOSED WORK

OM extracts and analyzes the human intention from their opinions towards products, and individuals [14]. Our research is aimed at analyzing the performance of OM classifiers in terms of different performance metrics and improving the performance of OM classifiers using DA. The proposed framework for this research work is depicted in Figure 1.

3.1. Data collection from Kaggle

In our research work, we obtained three datasets from Kaggle. These datasets correspond to different domains i.e. first dataset contains tweets from twitter, the second dataset contains movie reviews, and the last dataset contains tweets corresponding to the mental state of a person. So, we will refer to them as twitter dataset, movie reviews dataset and depression dataset in our work. Brief detail of the datasets is given:

– Sentiment140 dataset with 1.6 million tweets: this dataset has 1.6 million tweets taken from Twitter API. This dataset is used for SA and the tweets have sentiment labels: 0 (for negative), 2 (for neutral) and 4 (for positive). In our research work, we have used 100,000 tweets or records from this dataset. The link for this dataset is https://www.kaggle.com/datasets/kazanova/sentiment140.

- Movie reviews: this sentiment polarity dataset version 2.0 contains the movie reviews expressed by people. The opinion or sentiment labels for this dataset are pos (for positive reviews) and neg (for negative reviews). 60,000 records or reviews are considered from this dataset in our research. This dataset can be downloaded from https://www.kaggle.com/datasets/nltkdata/movie-review.
- Sentiment analysis for tweets: this dataset contains tweets to perform SA for finding whether the person is depressed or not, from their sentiments expressed on social media. Sentiment labels used in this dataset are 0 (if a person is not depressed) and 1 (if a person is depressed). In our research, we have used 10,000 records or tweets from this dataset. This dataset can be downloaded from https://www.kaggle.com/datasets/gargmanas/sentimental-analysis-for-tweets.

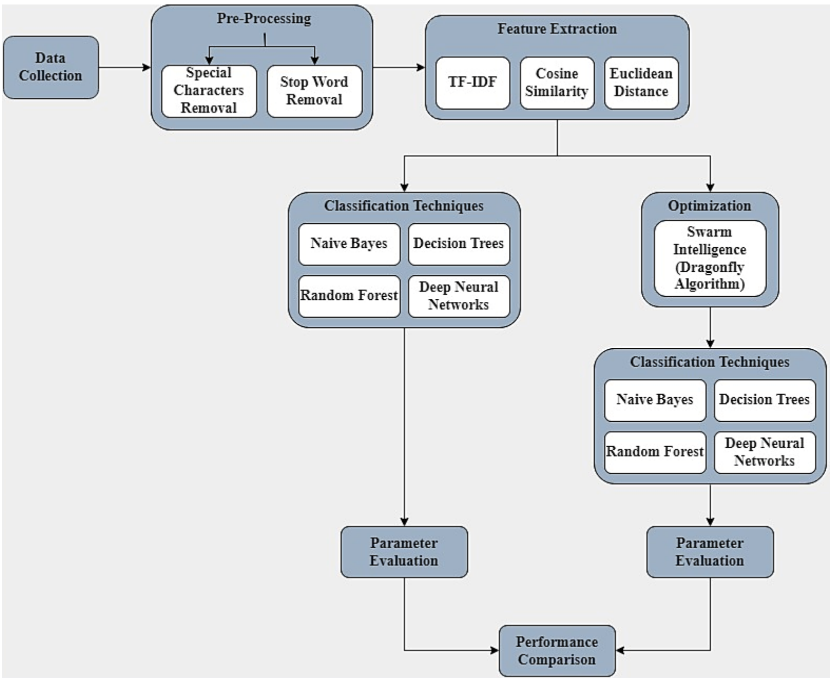

Figure 1. Proposed framework

3.2. Pre-processing of data

A large amount of data on the internet or web contains lots of irrelevant data and noise. Words like he, that, and there, are called stop words and do not contribute to decision-making, therefore it is crucial to eliminiate these words. Reviews or opinions are given in capital or small letters by people, so normalization is required to make all opinions in the same caption either large or small. There are some symbols used by people while giving opinions like @, #, and !. These punctuations or special characters should be removed from the data. That is why pre-processing is needed. In our research work, we have applied normalization, removed stop words, and removed special characters to make the data relevant for use.

3.3. Feature extraction

Feature extraction identifies and extracts relavant features from raw data that enhance the outcome of classification process. There are not many options in terms of exact features for textual data. Feature extraction methods used in our research are briefly described.

3.3.1. Term frequency-inverse document frequency

Term frequency-inverse document frequency (TF-IDF) finds some usual information from reviews or opinions expressed in textual form. It measures the significance of a word/term in a given document and assigns weight to the term based on its occurrence in the document, as shown in (1) and (2) [15], [16].

$$\mathrm{TF(t)} = \frac{(\text{Number of term t in a document})}{(\text{Total number of terms in a document})} \quad (1)$$

$$\text{IDF(t)} = \frac{\text{(Total number of documents)}}{\text{(Number of documents containing term t)}} \tag{2}$$

Finally, TF-IDF is the multiplication of TF and IDF score and is shown as (3):

$$\text{TF} - \text{IDF} = \text{TF} * \text{IDF} \tag{3}$$

3.3.2. Cosine similarity

It is used to measure the similarity of two non-zero vectors. It is used in many applications like information retrieval and textual mining, as a useful metric for measuring how similar or different two text phrases are. Cosine similarity (CS) between vectors A={x1, x2,…,xn} and B={y1, y2,…, yn} can be expressed as (4) [17]:

$$S_c(\text{A}, \text{B}) = \cos(\theta) = \frac{\text{A.B}}{||\text{A}||\,||\text{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 . \sum_{i=1}^{n} B_i^2}} \tag{4}$$

3.3.3. Euclidean distance

Numerous disciplines, including computer science, data analysis, physics, and ML, heavily rely on the euclidean distance (ED). It represents the straight-line distance between two points in Euclidean space. For points x = ($x_1, x_2, \ldots, x_n$) and y = ($y_1, y_2, \ldots, y_n$), ED is represented as (5) [17], [18]:

$$\text{d}(\text{x}, \text{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

3.4. Classification techniques

In the field of data analysis, ML has grown rapidly in recent years. ML makes the system able to gain knowledge and evolve from experience. Deep learning, a part of ML approaches enables the system to analyze data intelligently [19]. We are using four OM classification techniques in our research to analyze their performance. These techniques are briefly described.

3.4.1. Naïve Bayes

Naïve Bayes (NB) algorithm is majorly used for classifying text and spam filtering and makes predictions based on the probability of a specific object. The advantages of NB are it is an effective method, performs well in practice and processing is easy. NB is an algorithm of probability based on Bayes theorem, which is represented as (6) [20]:

$$\text{P}(\text{A} \mid \text{B}) = \frac{\text{P}(\text{B} \mid \text{A}) * \text{P}(\text{A})}{\text{P}(\text{B})} \tag{6}$$

where, P(A|B)=posterior probability, P(B|A)=likelihood, P(A)=prior probability, and P(B)=marginal likelihood.

3.4.2. Random forest

Random forest (RF) is a simple, robust, versatile, and effective algorithm with the ability to handle complex data. RF is a supervised ML technique, and it is a collection of different DTs that run in parallel to each other, without any contact among these trees. In this way, it considers multiple DTs which are constructed on different set of datsets and merges them to gain stable value. It performs classification by transmitting the class which is the mean prediction of the DTs [21].

3.4.3. Deep neural networks

Deep learning is a technique of ML where the machine has the capability to learn from experience. Sometimes, deep learning models give efficient performance and accuracy that exceeds human-level intelligence. Deep neural network (DNN) is a part of artificial intelligence and is used for visual and textual OM. The word "deep" in the DNN indicates multiple hidden layers in the neural network. DNN works with more than two layers: one is the input layer which includes input data, then hidden layers which include neurons; and the output layer which gives output [22], [23]. DNN is able to learn hierarchical representation of complex data, extract features, and make accurate predictions.

3.4.4. Decision trees

Decision trees (DT) is widely used in many applications like finance, education, healthcare, and marketing. DT is the strongest algorithm for classification and regression and is organized in the tree structure form. In DT, every node specifies a test on an element and the branch specifies the test result. At the end, the terminal node represents a class label [20]. DT in the form of a tree forms some of the classification rules and has advantages over other techniques, which are: simple representation makes it easy to understand and classification of test data is done very fast [24].

3.5. Performance parameters

Parameters used for performance evaluation of OM classifiers are described [25], [26]: i) precision: it is also referred to as correctness, is the percentage of correctly classified instances; ii) recall: it is the ratio of the count of correctly classified positive reviews to the total count of reviews; iii) f-score: f-score or f-measure is the harmonic mean of precision and recall; and iv) accuracy: it is the fraction of predictions our model got right and is given by the count of correct predictions to the total count of predictions. These performance metrics can be obtained from TP, TN, FP, and FN as shown in the [20]:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})} \tag{8}$$

$$\text{F-Score} = \frac{(2*\text{Precision}*\text{Recall})}{(\text{Precision}+\text{Recall})} \tag{9}$$

$$\text{Accuracy} = \frac{(\text{TP}+\text{TN})}{(\text{TP}+\text{TN}+\text{FP}+\text{FN})} \tag{10}$$

where, TP=true positives, TN=true negatives, FP=false positives, and FN=false negatives.

3.6. Optimization using proposed dragonfly algorithm

Any optimization algorithm is evaluated in many numbers of iterations or steps because in one or two iterations it can't reach its best-fit value. DA is iterated multiple times, which are called generations or levy flights. DA is shown in Figure 2 and is used here to select or reject the records/rows in the data.

```
R_explo = 0, R_expl = 0
P_dragon_per = 0.20, 0.50, L_expl = 10
L_explo = 1, L_flight = 1
[rows, cols] = size (AllFeature)
accepted_set = [], accepted_labels = []
accounter = 0, indexcount = 0
{Repeat for exploitation and exploration} for i = 1 to rows do
    {Loop over total number of dragons} current_dragon = AllFeature (i,: )
    current_dragon_group = kid (i) for L_explo = 1 to L_flight do
        D_p = round (rand(1, numel(current_dragon_group) * P_dragon_per))
        D_p = findandreplace(D_p, 0, 1)
        V_pair = V_explore (D_p , :)
        C_index = C(current_dragon_group , :)
        A = calc_alignment(current_dragon, V_pair)
        C = calc_cohesion(current_dragon, C_index)
        D_AC = |(A−C)/A|
        P_diff = D_AC * 100
        if P_diff <= 75 then
            R_expl (L_flight) = 1
        end if
    end for
    R_total = R_explo + R_expl
    for each row selection in different dataset: do
        Perform row selection using R_total as a guide.
        Add the selected row to accepted_set and its label to accepted_labels.
    end for
end for
selected_set = accepted_set
selected_labels = accepted_labels
```

Figure 2. Dragonfly algorithm

DA displayed here works for both the exploitation and exploration phases. In this algorithm, the population volume for the exploitation phase is 20% and for the exploration phase is 50%. This algorithm runs for ten levy flights or generations. In the algorithm, alignment (A) and cohesion (C) are calculated, which are used for finding the core factor (D_{AC}). (D_{AC}) is calculated by subtracting cohesion (C) from alignment (A), and then dividing the result by alignment (A). P_{diff} is D_{AC} in terms of percentage. If P_{diff} is less than or equal to 75, then the reward is set to 1. Total reward (R_{total}) is the sum of rewards for the exploitation and exploration phase and is used as a guide for row selection. This should be repeated for both the exploitation and exploration phases. Selected rows are stored in selected_set and their labels in selected_labels.

4. RESULTS AND DISCUSSION

For our research work, we used MATLAB which is a programming platform to analyze and design systems and products. In this platform, we first designed the user view and then implementation was done. We used twitter, movie reviews and depression datasets (in Excel format) in our research.

The first task in the implementation was uploading a dataset at a time. Once uploaded, pre-processing of data was done for normalization, special character removal and stop word removal. After pre-processing, feature extraction methods: TF-IDF, CS, and ED were applied. After this, training and classification were done and the performance of OM classification techniques (NB, RF, DT, and DNN) was evaluated in terms of precision, recall, f-score, and accuracy for the uploaded dataset. This task is repeated for the other two datasets. The performance analysis results (up to three decimal places) of OM classifiers (without optimization) for all the datasets are displayed in Table 1.

We analyzed from these results that all the OM classification techniques performed well, but to improve the performance of these OM classifiers, we proposed an optimization algorithm, which is DA. The DA displayed in Figure 2 was applied to perform optimization by selecting and/or rejecting the records. Table 1 also illustrates the performance results (up to three decimal places) of OM classifiers after applying optimization using DA for all the datasets. Table 1 shows that OM classifiers gave improved performance after optimization using DA.

Table 1. OM classifiers performance with and without optimization for all datasets

Dataset	Classifiers	Without optimization				With optimization			
		Precision	Recall	F-sdcore	Accuracy	Precision	Recall	F-score	Accuracy
Twitter	NB	0.834	0.609	0.704	0.590	0.984	0.837	0.904	0.861
	RF	0.843	0.644	0.730	0.619	0.984	0.887	0.933	0.901
	DNN	0.836	0.620	0.712	0.598	0.982	0.821	0.894	0.850
	DT	0.836	0.654	0.734	0.620	0.984	0.891	0.935	0.904
Movie review	NB	0.838	0.601	0.700	0.588	0.983	0.840	0.906	0.864
	RF	0.858	0.675	0.756	0.651	0.984	0.898	0.939	0.909
	DNN	0.844	0.649	0.734	0.623	0.983	0.859	0.917	0.879
	DT	0.838	0.612	0.707	0.595	0.983	0.889	0.934	0.903
Depression	NB	0.841	0.684	0.755	0.644	0.983	0.866	0.921	0.885
	RF	0.859	0.691	0.766	0.662	0.984	0.899	0.940	0.910
	DNN	0.844	0.654	0.737	0.627	0.983	0.832	0.901	0.858
	DT	0.847	0.687	0.759	0.651	0.985	0.889	0.934	0.902

From Table 1 it is difficult to display and compare the performance of OM classifiers before and after optimization based on each performance metric by plotting the charts; so, we summarized the results of Table 1 by taking the average of each classifier's value for all datasets, on the basis of different performance metrics considered in our research work. Figures 3 to 6 show the performance comparison of OM classifiers without and with optimization using DA in the form of charts using the summarized results (up to three decimal places) for all the datasets in terms of precision, recall, f-score, and accuracy.

To validate this research work, a comparison with other researcher's work is also shown. Table 2 shows the performance of the research work of Elangovan and Subedha [7] for all the datasets considered in this research on the basis of presision, recall, f-score, and accuracy. To compare the performance with the performance of OM classifiers after optimization, the average of the results (up to three decimal places) for all the parameters from Table 2 is taken. For all the parameters, the average performance of Elangovan and Subedha is compared with the average performance of all OM classifiers with optimization as shown in Figure 7. The comparison shows that the research work of Elangovan and Subedha observes less performance for almost all OM classifiers with optimization.

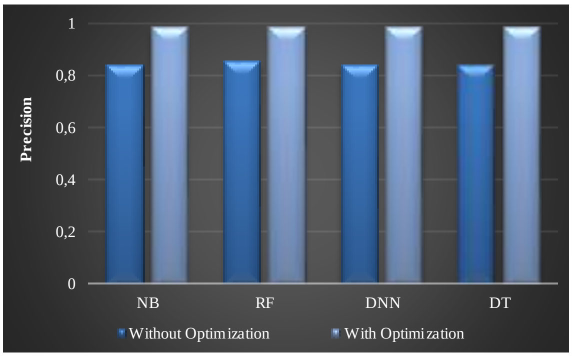

Figure 3. Comparison of OM classifiers without and with optimization in terms of precision

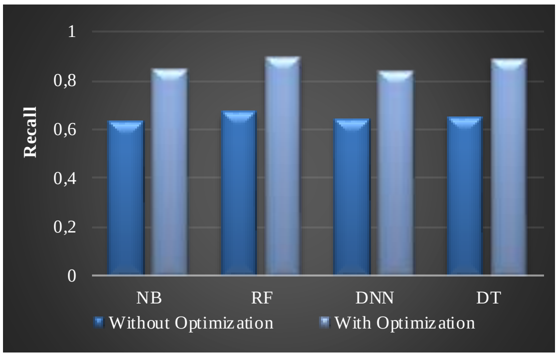

Figure 4. Comparison of OM classifiers without and with optimization in terms of recall

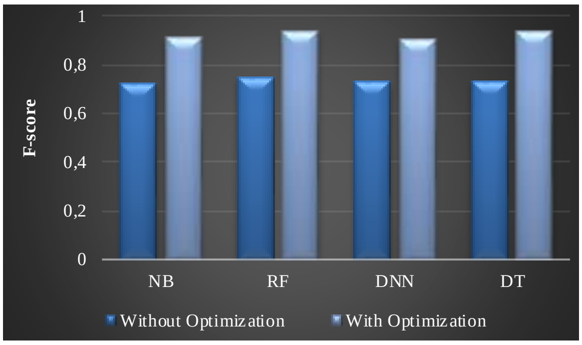

Figure 5. Comparison of OM classifiers without and with optimization in terms of f-score

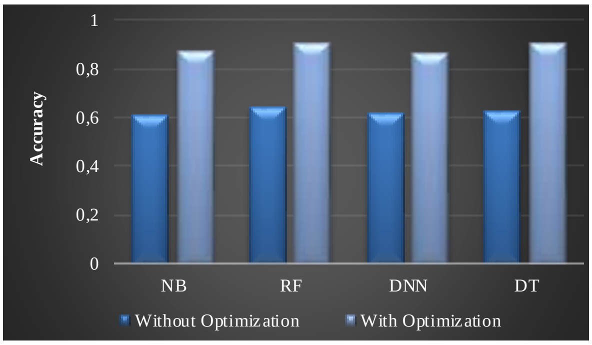

Figure 6. Comparison of OM classifiers without and with optimization in terms of accuracy

Table 2. Performance of Elangovan and Subedha [7] for all the datasets

Datasets	Parameters for evaluation			
	Precison	Recall	F- Score	Accuracy
Twitter	0.910	0.865	0.887	0.892
Movie review	0.858	0.897	0.878	0.882
Depression	0.883	0.893	0.888	0.919

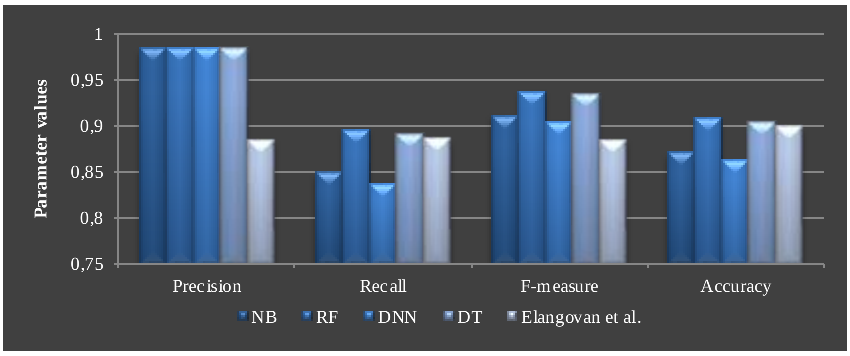

Figure 7. Performance comparison of OM classifiers after optimization and Elangovan and Subedha in terms of performance parameters values

The results observed show that the proposed method in this research tended to have an inordinate performance in comparison to the performances of OM classifiers without optimization and the research work of Elangovan and Subedha. This research explored the usefulness of the proposed method. However, further, and in-depth studies and research may be needed to confirm its effectiveness for other classifiers.

DA is simple and easy to implement and provides a good optimization capability. In comparison to other optimization algorithms, DA can be easily merged with other algorithms. DA uses few parameters for tuning and provides a good optimization capability. For small to medium problems, DA performs well, but for complex prolems DA faces many issues. To overcome these difficulties, DA can be merged with other algorithms.

5. CONCLUSION

In this research work, our focus was on the performance analysis of OM classification techniques and data optimization. After data collection, and pre-processing, we used TF-IDF, CS, and ED for feature extraction; and then performed classification and evaluated the performance of NB, RF, DNN, and DT classification techniques. These OM classifiers performed well but to improve their performance, an optimization algorithm was proposed. The DA, which is a SI algorithm, was used for optimization; and the performance of OM classifiers was evaluated after applying DA. The performances of OM classification techniques before and after applying DA were compared with each other in terms of precision, recall, f-score, and accuracy; and displayed in the form of charts. Our results provide conclusive evidence that the OM classifiers performed well after optimization using DA. Better performance of OM classifiers after optimization using DA, in comparison to the research work of Elangovan and Subedha, also confirmed the effectiveness of DA. So, DA proved to be helpful in the fruitfulness of the existing classifiers. In the future, more research can be done on exploring the design and development of hybrid classifier and merging DA with other algorithms, which can further improve the performance of OM classification techniques.

REFERENCES

[1] V. A. Pulikonda, D. Vanukuru, M. S. Navali, B. Motepalli, and K. Swathi, "Exploring the applications, challenges, and issues of sentiment analysis," in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Feb. 2023, pp. 807–812, doi: 10.1109/ICCMC56507.2023.10083940.

[2] P. Tanwar and P. Rai, "A proposed system for opinion mining using machine learning, NLP and classifiers," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 9, no. 4, pp. 726-733, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp726-733.

[3] M. Malik, S. Habib, and P. Agarwal, "A novel approach to web-based review analysis using opinion mining," *Procedia Computer Science*, vol. 132, pp. 1202–1209, 2018, doi: 10.1016/j.procs.2018.05.035.

[4] Z. Salah *et al.*, "A systematic review on opinion mining and sentiment analysis in social media," *International Journal of Business Information Systems*, vol. 31, no. 4, 2019, doi: 10.1504/IJBIS.2019.101585.

[5] M. Birjali, M. Kasri, and A. B. -Hssane, "A comprehensive survey on sentiment analysis: approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, Aug. 2021, doi: 10.1016/j.knosys.2021.107134.

[6] P. Mudgil, P. Gupta, I. Mathur, and N. Joshi, "An improved context-aware analysis for sentimental grass hopper optimization algorithm and its post affects on Twitter," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 6, pp.10275–10295, Jun. 2023, doi: 10.3233/JIFS-221879.

[7] D. Elangovan and V. Subedha, "An effective feature selection based classification model using firefly with levy and multilayer perceptron based sentiment analysis," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Feb. 2020, pp. 376–380, doi: 10.1109/ICICT48043.2020.9112425.

[8] D. Hendardi *et al.*, "Dragonfly algorithm in 2020: review," *Communications in Mathematical Biology and Neuroscience*, vol. 47, pp. 1–21, 2021, doi: 10.28919/cmbn/5767.

[9] E. Indramaya and S. Suyanto, "Comparative study of recent swarm algorithms for continuous optimization," *Procedia Computer Science*, vol. 179, pp. 685–695, 2021, doi: 10.1016/j.procs.2021.01.056.

[10] C. M. Rahman and T. A. Rashid, "Dragonfly algorithm and its applications in applied science survey," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–21, Dec. 2019, doi: 10.1155/2019/9293617.

[11] C. M. Rahman, T. A. Rashid, A. Alsadoon, N. Bacanin, P. Fattah, and S. Mirjalili, "A survey on dragonfly algorithm and its applications in engineering," *Evolutionary Intelligence*, vol. 16, no. 1, pp. 1–21, Feb. 2023, doi: 10.1007/s12065-021-00659-x.

[12] Y. Meraihi, A. R. -Cherif, D. Acheli, and M. Mahseur, "Dragonfly algorithm: a comprehensive review and applications," *Neural Computing and Applications*, vol. 32, no. 21, pp. 16625–16646, Nov. 2020, doi: 10.1007/s00521-020-04866-y.

[13] M. Alshinwan *et al.*, "Dragonfly algorithm: a comprehensive survey of its results, variants, and applications," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 14979–15016, Apr. 2021, doi: 10.1007/s11042-020-10255-3.

[14] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: a survey," *Natural Language Processing Journal*, vol. 2, Mar. 2023, doi: 10.1016/j.nlp.2022.100003.

[15] R. K. Mishra, S. Urolagin, and A. A. J. J., "A sentiment analysis-based hotel recommendation using TF-IDF approach," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dec. 2019, pp. 811–815, doi: 10.1109/ICCIKE47802.2019.9004385.

[16] Prathyakshini, P. S. K, P. B. R, T. Nargis, and S. Salian, "A novel approach for classification of online product reviews using various machine learning techniques," in *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, Dec. 2022, pp. 878–884, doi: 10.1109/ICECA55336.2022.10009447.

[17] P. Marjai, P. L. -Kéry, and A. Kiss, "Document similarity for error prediction," *Journal of Information and Telecommunication*, vol. 5, no. 4, pp. 407–420, Oct. 2021, doi: 10.1080/24751839.2021.1893496.

[18] O. E. Oduntan, I. Adeyanju, A. S. Falohun, and O. Obe, "A comparative analysis of euclidean distance and cosine similarity measure for automated essay-type grading," *ARPN Journal of Engineering and Applied Sciences*, vol. 13, no. 11, pp. 4198–4204, 2018, doi: 10.3923/jeasci.2018.4198.4204.

[19] I. H. Sarker, "Machine learning: algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00592-x.

[20] A. Kumar, T. Jain, P. Tiwari, and R. Sharma, "Opinion mining on amazon musical product reviews using supervised machine learning techniques," in *2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)*, Feb. 2023, pp. 1–6, doi: 10.1109/IEMECON56962.2023.10092288.
[21] P. Karthika, R. Murugeswari, and R. Manoranjithem, "Sentiment analysis of social media network using random forest algorithm," in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, Apr. 2019, pp. 1–5, doi: 10.1109/INCOS45849.2019.8951367.
[22] N. C. Dang, M. N. M. -García, and F. D. L. Prieta, "Sentiment analysis based on deep learning: a comparative study," *Electronics*, vol. 9, no. 3, Mar. 2020, doi: 10.3390/electronics9030483.
[23] F. Correia, A. M. Madureira, and J. Bernardino, "Deep neural networks applied to stock market sentiment analysis," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124409.
[24] M. Rathi, A. Malik, D. Varshney, R. Sharma, and S. Mendiratta, "Sentiment analysis of tweets using machine learning approach," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Aug. 2018, pp. 1–3, doi: 10.1109/IC3.2018.8530517.
[25] R. A. Laksono, K. R. Sungkono, R. Sarno, and C. S. Wahyuni, "Sentiment analysis of restaurant customer reviews on tripadvisor using naïve bayes," in *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Jul. 2019, pp. 49–54, doi: 10.1109/ICTS.2019.8850982.
[26] V. Gopalakrishnan and C. Ramaswamy, "Patient opinion mining to analyze drugs satisfaction using supervised learning," *Journal of Applied Research and Technology*, vol. 15, no. 4, pp. 311–319, Aug. 2017, doi: 10.1016/j.jart.2017.02.005.

BIOGRAPHIES OF AUTHORS

Mikanshu Rani is Ph. D. Scholar, in the Department of Computer Science & Engineering at Guru Jambheshwar University of Science and Technology, Hisar, Haryana and working as a Project Associate at the Center for Development of Advance Computing, Mohali, Punjab. She completed her M.Tech. in Computer Science and Engineering from Chaudhary Devi Lal University, Sirsa, Haryana. She has done a few publications in national and international conferences and journals. Her research area includes opinion mining, machine learning, and data mining. She can be contacted at email: mikanshu6406@gmail.com.

Dr. Jaswinder Singh is working as a Professor in the Department of Computer Science & Engineering at Guru Jambheshwar University of Science and Technology, Hisar, Haryana. He has teaching experience of more than 20 years, and he has published more than 30 research papers in international journals and conferences. He completed his Ph.D. in Computer Science and Engineering from Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonepat, Haryana, and completed his M.Tech. in Computer Science and Engineering from Kurukshetra University, Kurukshetra, Haryana. His areas of research include opinion mining, web information retrieval, search engine optimization, information systems, and social network analysis. He can be contacted at email: jaswinder_singh@gjust.org.