ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Big Data Sentimental Analysis Using Document to Vector and Optimized Support Vector Machine

Sozan Abdulla Mahmood, Qani Qabil Qasim

Department of Computer Science, University of Sulaimani, Sulaymaniyah, Iraq

ABSTRACT

With the rapid evolution of the internet, using social media networks such as Twitter, Facebook, and Tumblr, is becoming so common that they have made a great impact on every aspect of human life. Twitter is one of the most popular micro-blogging social media that allow people to share their emotions in short text about variety of topics such as company's products, people, politics, and services. Analyzing sentiment could be possible as emotions and reviews on different topics are shared every second, which makes social media to become a useful source of information in different fields such as business, politics, applications, and services. Twitter Application Programming Interface (Twitter-API), which is an interface between developers and Twitter, allows them to search for tweets based on the desired keyword using some secret keys and tokens. In this work, Twitter-API used to download the most recent tweets about four keywords, namely, (Trump, Bitcoin, IoT, and Toyota) with a different number of tweets. "Vader" that is a lexicon rule-based method used to categorize downloaded tweets into "Positive" and "Negative" based on their polarity, then the tweets were protected in Mongo database for the next processes. After pre-processing, the hold-out technique was used to split each dataset to 80% as "training-set" and rest 20% "testing-set." After that, a deep learning-based Document to Vector model was used for feature extraction. To perform the classification task, Radial Bias Function kernel-based support vector machine (SVM) has been used. The accuracy of (RBF-SVM) mainly depends on the value of hyperplane "Soft Margin" penalty "C" and γ "gamma" parameters. The main goal of this work is to select best values for those parameters in order to improve the accuracy of RBF-SVM classifier. The objective of this study is to show the impacts of using four meta-heuristic optimizer algorithms, namely, particle swarm optimizer (PSO), modified PSO (MPSO), grey wolf optimizer (GWO), and hybrid of PSO-GWO in improving SVM classification accuracy by selecting the best values for those parameters. To the best of our knowledge, hybrid PSO-GWO has never been used in SVM optimization. The results show that these optimizers have a significant impact on increasing SVM accuracy. The best accuracy of the model with traditional SVM was 87.885%. After optimization, the highest accuracy obtained with GWO is 91.053% while PSO, hybrid PSO-GWO, and MPSO best accuracies are 90.736%, 90.657%, and 90.557%, respectively.

Index Terms: Document to Vector, Grey Wolf Optimizer, Particle Swarm Optimizer, Hybrid Particle Swarm Optimizer_Grey Wolf Optimizer, Opinion Mining, Radial Bias Function Kernel-based Support Vector Machine, Sentiment Analysis, Support Vector Machine Optimization, Twitter Application Programming Interface

Access this article online	
DOI:10.21928/uhdjst.v4n1y2020.pp18-28	**E-ISSN:** 2521-4217 **P-ISSN:** 2521-4209

1. INTRODUCTION

Nowadays, the use of the internet has become inseparable from our daily routines. Social media networks such as Facebook and Twitter have also been developed to give a right to people to easily share their viewpoints about any

Corresponding author's e-mail: Qani Qabil Qasim, Department of Computer Science, University of Sulaimani, Sulaymaniyah, Iraq. E-mail: qani.qabil@gmail.com

Received: 23-10-2019 Accepted: 06-01-2020 Publishing: 13-02-2020

product or service in the form of short text. This makes them to be rich sources of data that can be valuable for various organizations and companies to find their fans' or customers' opinions about their products and services. In spite of companies, well-known people such as politicians and athletes may need to exploit those opinions and attitudes as well as to help them for making better decision-making in the future. However, data diversity and sparsity make it impossible for human to be able to analyze it. Here, the role of machine learning and automation can take a part to solve the problem of big data. Sentiment analysis (SA) or opinion mining techniques could be used [1].

SA refers to the task of finding the opinions of authors about specific entities that expressed in a written text [2].

In recent years, Twitter has become one of the most popular social media and microblogging platform where it is a convenient way for users to write and share their thoughts about anything within 280-characters length (called tweets). Twitter is used extensively as a microblogging service worldwide. Tweets consist of misspellings, slangs, and symbolic forms of words, which poses a major challenge for the conventional natural language processing or machine learning systems to be used on tweets [3].

Sentiment analyzer model can be built in three main approaches – lexicon-based approach, machine learning-based approach, and hybrid of both lexicon-based and machine learning approach. The machine learning approach is one of the most popular techniques that are widely used to build an automated classification model with the help of algorithms such as support vector machine (SVM), Naïve Bayes (NB), and so on. This is due to their ability to handle a large amount of data [4].

In this study, we propose a technique to promote SVM performance for SA by implementing four different meta-heuristic optimizers, namely, particle swarm optimizer (PSO), modified PSO (MPSO), grey wolf optimizer (GWO), and hybrid of PSO-GWO. The sentiment classification goes through four phases: Data collection, data pre-processing, feature extraction, and classification. In the first phase, Twitter Application Programming Interface (Twitter-API) enables developers to collect tweets about any keyword they desire and then followed by preprocessing phase to remove least informative data such as URL, hashtags, numbers, and so on. In the third phase, Document to Vector (Doc2Vec) approaches were used for vectorizing cleaned text, which is the numerical representation of text. PSO, GWO, and hybrid PSO-GWO are used to select the best parameters for the classifier (SVM) to classify generated features from the previous step.

The rest of the paper is structured as follows: In section 2, some previous related works in this field that has been conducted before being discussed, section 3 describes the material and methods used in this work, section 4 describes the problem statement, section 5 illustrates the proposed system model and methodology of analyzing the datasets, section 6 shows the results obtained from the model and discussed in detail, and finally, the conclusion and future work are stated in section 7.

2. RELATED WORK

Many researches and works have been developed in the field of SA. Researchers have proposed different solutions to different issues of SA in terms of improving performance of classification models, enhancing topic specific corpus, reducing feature-set size to shrink execution time of algorithms and space usage using different techniques.

Das *et al.* [5] review basic stages to be considered in SA, such as pre-processing, feature extraction/selection, and representation along with some data-driven techniques in this field such as SVM and NB as well as to demonstrate how they work and the measuring metrics such as (Precision, Recall, F1-Score, and Accuracy) to evaluate the model efficiency. They concluded that all the SA tasks are challenging and need different techniques to deal with each stage.

Naz *et al.* [6] illustrate the impact of different weighting feature schemes such as term frequency (TF), TF-inverse document frequency (TF-IDF), and binary occurrence (BO) to extract features from tweets along with different n-gram ranges such as unigram, bigram, trigram, and their combination, followed by feeding extracted feature from SemEval2016 dataset to train SVM. The best result they achieved is 79.6% for TF-IDF with Unigram range. They also used the sentiment score vector package to calculate the score of tweets into positive and negative forms to improve the performance of SVM, along with different weighting schemes and n-gram range, the highest accuracy achieved with SVC is 81.0% for BO with unigram range.

Seth *et al.* [7] proposed a hybrid technique for improving the efficiency and reliability of their model by merging SVM with the decision tree. The model performs a classification of tweets on the basis of SVM and adaboost decision tree individually. Then, a hybrid technique will be applied by feeding the outputs obtained from the two above mentioned algorithms as the input to the decision tree. Finally, they compared traditional techniques to the proposed model and obtained the accuracy of 84%, while prior accuracies were 82% and 67%.

Sharma and Kumari [8] applied SVM to find the polarity of four smartphone product review texts, whether positive or negative. Before applying SVM, they used part of speech (POS) tagging with tokens, then used clustering for TF-IDF features to find more appropriate centroids. The accuracy of the model was evaluated based on (Precision, Recall, F-score, and Accuracy) metrics, compared to previous studies on the same datasets where no POS and no clustering were performed. They obtained the accuracy of 90.99% while the best previous study accuracy was 88.5%.

Rajput and Dubey [9] made a comparative study between two supervised classification algorithms, namely, NB and SVM for making binary classification of customers review about six Indian stock market. The results show that SVM provides better accuracy, which was 81.647%, while NB accuracy was 78.469%.

Rane and Kumar [10] worked on a six major US Airline datasets for performing a multi-class (Positive, Negative, and Neutral) SA. Doc-2Vec deep learning approach has been used for representing these tweets as vectors to do a phrase-level analysis – along with seven (7) supervised machine learning algorithms (Decision Tree, Random Forest, SVM, K-Nearest Neighbors, Logistic Regression, Gaussian NB and AdaBoost). Each classifier was trained with 80% of the data and tested using the remaining 20% data. Accuracy of all classifiers was calculated based on (Precision, Recall, F1-Score) metrics. They concluded that the classification techniques used include ensemble approaches, such as AdaBoost, which combines several other classifiers to form one strong classifier which performs much better. The maximum achieved accuracy was 84.5%.

Shuai *et al*. [11], these authors carry out a binary SA on Chinese hotel reviews by using Doc2vec feature extraction technique and SVM, logistic regression and NB as a classifier. After making a performance comparison between classification algorithms based on the precision, recall rate, and F-measure metrics, SVM achieved the best accuracy in their experiment as follows: 79.5%, 87.92%, and 81.16% for all three metrics.

Bindal and Chatterjee [3] described two-step method (lexicon-based sentiment scoring in conjunction with SVM, point-wise mutual information utilized to calculate sentiment of tweets. They also discussed the efficacy of several linguistic features, such as POS tags and higher-order n-grams (Uni + Bi Gram, Uni + Bi + Tri Gram) in sentiment mining. Their proposed scheme had better "F-Score" average than commonly used one-step methods such as Lexicon, NB, Maximum Entropy, and SVM classifier, i.e., for Unigram range lexicon-SVM outperforms other classification methods with F-score of 84.39% while other methods F-score is 82.44%, 81.85%, 80.18%, and 83.56%, respectively.

Mukwazvure and Supreethi [12] used a hybrid technique which involves lexicon-based approach for detecting "news comments" polarity in (Technology, Politics, and Business) domains. Then, the outcome of lexicon-based is then fed to train two supervised machine learning algorithms: SVM and K-nearest neighbor (kNN) classifiers. Investigational results revealed that SVM performed better than kNN which were 73.6, 61.38, and 58.00 while kNN results were 74.24%, 56.27%, and 55.58%.

Flores *et al*. [13] made a comparative analysis of SVM algorithm-sequential minimal optimization with synthetic minority over-sampling technique (SMOTE) and Naive Bayes multinomial (NBM) algorithm with SMOTE for classification of two SA datasets gathered by students of University of San Carlos. The outcomes have shown that with 10-folds cross-validation SA for their datasets could perform better compared to 70:30 split. Performance of NBM with SMOTE was 72.33% and 78.02% and SVM with SMOTE were 83.16% and 82.22% in the term of accuracy.

3. MATERIALS AND METHODS

3.1. VADER

VADER stands for Valence Aware Dictionary and sEntiment Reasoner. It is a lexicon and rule-based SA tool that was developed by Hutto and Gilbert [14] in 2014. It is specifically attuned to do calculate the sentiment scores of texts expressed on social media. VADER uses a combination of a sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the positivity and negativity score but also tells us about how much positive or negative a sentiment is? VADER produces four sentiment metrics from these word ratings, the first three, positive, neutral, and negative, represents the proportion of the text that falls into those categories and the final metric is compound score which is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). According to their experiment, it is more effective than other existing lexicon-based approaches, for example, **SentiWordNet.**

3.2. Word Embedding and DOC2VEC

Word embedding, also known as (Word2Vec), is a technique for unique vector representation of each word with its semantic meaning of the word taken into consideration. Unlike bag of words, which is one of the most common techniques used for numerical representation of words that convert word to a fixed-length feature vector, it has some shortcomings. First, it does not consider the ordering of the words, ignores semantics of the words. For example, “powerful,” “strong,” and “Paris” are equally evaluated and generate a high dimensional feature set, so, it needs a lot of memory space [15].

In Word2Vec approach, each word is mapped to a vector in a predefined vector space. These vectors are learned using neural networks. The learning process can be done with a neural network model or by using an unsupervised process involving document statistics. Word2Vec can be implemented in two different architectures, first is continuous bag of word (CBoW), as shown in Fig. 1 which is designed to predict current words at an input of future words and history words and the second is skip-gram (SG) which is used to maximize the probability of surrounding words given the current word being used in word embedding [15], [16].

Doc2Vec, also called paragraph vector (PV), is a (Word2Vec) based learning approach that converts entire paragraph to a unique vector which is represented by a column in matrix D and every word is mapped to unique vector mapped in matrix W. The word and PVs are then concatenated to predict the next word. CBoW and SG methods have been tuned for Doc2Vec and converted into two methods, namely, distributed bag of words version of PVs (PV-DBOW) and distributed memory of PVs (PV-DM) [10], as shown in Figs. 2 and 3.

The DBOW model ignores the context words in the input, but force the model to predict words randomly sampled from the paragraph in the output [15].

In DM model, to predict the next word in a context, the paragraph and word vectors either being averaged (mean) which is called DM mean (DMM), or concatenated which is called DM concatenation (DMC) [15].

3.3. PSO Algorithm

PSO is a type of meta-heuristic algorithm developed by Dr. Kennedy and Dr. Eberhart in 1995 to optimize numeric problems iteratively. PSO simulates the behaviors of the animals’ groups searching for food, especially bird flocking or fish schooling. PSO starts through a randomly distributed group of agents called particles in a search space; every particle has self-own velocity [17].

Each particle has two “best” achieved positions; the first one is its best position or (local best position) referred to as “pbest.” And the second one is (global best position) referred to as “gbest.”

At each time the particles will move toward “pbest” and “gbest” based on a new “velocity” and some constant

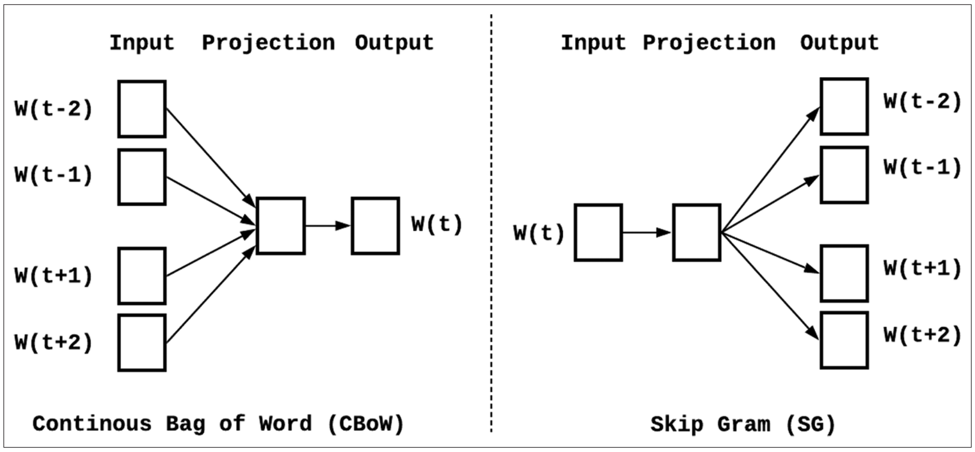


Fig. 1. Continuous bag of word and skip-gram.

coefficient parameters such as c_1, c_2, and w (inertia weight) and two random numbers.

In D-dimensional space, PSO algorithm can be described as follows:

$X_i = (X_{i1}, X_{i2}, X_{i3}, \ldots, X_{iD})$ represents the current position of the "particle," $V_i = (V_{i1}, V_{i2}, V_{i3} \ldots V_{iD})$ and it refers to its velocity, the local best location is denoted as Pbest,i = $(P_{i1}, P_{i2}, P_{i3} \ldots P_{iD})$, and global best position of all particles refers to Pgbest,i = $(P_{g1}, P_{g2}, P_{g3} \ldots P_{gD})$.

At every iteration, each particle changes its position according to the new velocity.

$$v_i^{t+1} = w^* v_i^t + c_1 r_1 + \left(pBest_i^t - x_i^t\right) + c_2 r_2 + \left(gBest_i^t - x_i^t\right) \quad (1)$$

In this study, instead of multiplying w to only current velocity, after changing the current particle best position and group best position, we multiplied them all to "w." The formulated equation is:

$$v_i^{t+1} = w^* \begin{pmatrix} v_i^t + c_1 r_1 + \left(pBest_i^t - x_i^t\right) \\ +c_2 r_2 + \left(gBest_i^t - x_i^t\right) \end{pmatrix} \quad (2)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

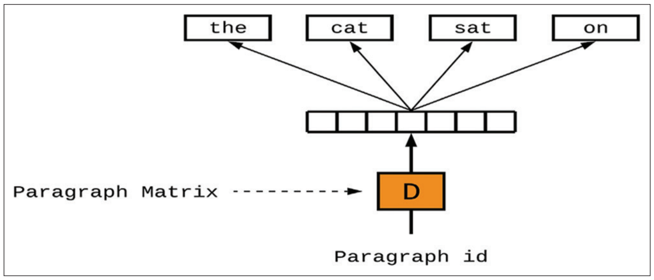


Fig. 2. Distributed bag of word of paragraph vector.

Where "i" refers to a particle, pBest, and gBest as the best particle position, best group position, and the parameters w, c_1 and c_2 are called inertia weighs. r_1 and r_2 are two random numbers in the range of (0, 1), v_i^t is a current velocity, v_i^{t+1} indicates new velocity in the next time or iteration. Furthermore, x_i^t is current particle position, x_i^{t+1} indicates the new particle position.

The pseudocode of PSO is:

```
Initialized number of particles (n_particle), D, n_iterations, c1, c2, and w.
For each particle i ∈ (n_particle)
    Initialize $X_i$, $V_i$
End for
For each particle i in n_particle do
    If $f(X_i) < f(Pi)$
        $Pbest_i = X_i$
    End if
    If $f(Pbest_i) < f Gbest_i$
        $Gbest = Pbest_i$
    End if
End for
For each particle i in n_particle do
    For each dimension d in D
        Update velocity according to equation (1) for PSO and equation (2) for MPSO
        Update position according to equation (3)
    End for
End for
Iteration = Iteration +1
Until iteration > n_iterations.
```

3.4. GWO Algorithm

GWO algorithm is another type of swarm intelligence algorithm, proposed by Mirjalili *et al.* in 2014 [18], that mimics the leadership hierarchy and hunting mechanism of

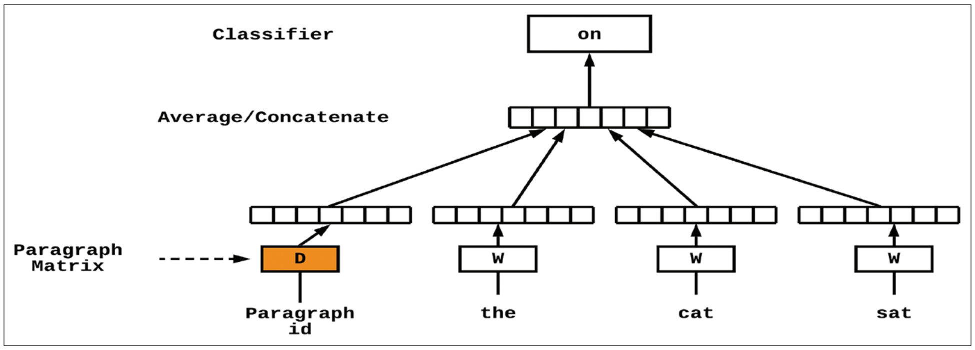


Fig. 3. Distributed memory of paragraph vector.

grey wolves in nature. Four types of grey wolves, such as alpha, beta, delta, and omega, are employed for simulating the leadership hierarchy. Furthermore, the three main steps of hunting, searching for prey, encircling prey, and attacking prey, are implemented.

3.4.1. Social hierarchy

The social hierarchy in this algorithm consists of four groups of wolves, namely, alpha (α), beat (β), and delta (δ), and the other is called omega (ω). In the GWO algorithm, the hunting (optimization) process is guided by α, β, and δ. The ω wolves follow these three wolves [18].

3.4.2. Encircling prey

Encircling prey means that grey wolves surround prey during the hunt, the following mathematical equations form the encircling behavior [18]:

$$\vec{D} = \left|\vec{C}.\vec{X}p(t) - \vec{X}(t)\right| \tag{4}$$

$$\vec{X}(t+1) = \left|\vec{X}p(t) - \vec{X}.\vec{D}\right| \tag{5}$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_P$ is the position vector of the prey, and $\vec{X}$ indicates the position vector of a grey wolf.

The vectors A and C are calculated as:

$$\vec{A} = 2\,\vec{a}\,.\,\vec{r}_1 - \vec{a} \tag{6}$$

$$\vec{C} = 2\,\vec{r}_2 \tag{7}$$

Where $\vec{a}$ linearly decreased from 2 to 0 throughout iterations and r_1, r_2 are two random vectors in the range of 0, 1.

3.4.3. Hunting

Grey wolves can identify the location of prey and encircle them. The hunt is usually guided by the alpha. Sometimes beta and delta might also get involved in hunting, alpha (best candidate solution), beta, and delta have better knowledge about the potential location of prey. Thus, the first three best solutions are selected to update their positions according to the position of the best search agents based on the following mathematical formulas [18]:

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|$$

$$\vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right| \tag{8}$$

$$\vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right|$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta) \tag{9}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{10}$$

The pseudocode of GWO as follows:

```
Initialize the grey wolf population Xi (i = 1, 2..., n)
Initialize a, A, and C
Calculate the fitness of each search agent using equations (8) and (9)
$X_\alpha$ = The first best search agent
$X_\beta$ = The second-best search agent
$X_\delta$ = The third best search agent
While (t < Max number of iterations)
For each search agent
Update the position of the current search agent according to equation (9)
End for
a=2−t*(2/(Max_iteration))
Calculate A, C using equations (6) and (7)
Calculate the fitness of each search agent using equations (8) and (9)
Update position of the current search agents according to equation (10)
t=t + 1
End while
Return $X_\alpha$.
```

3.5. Hybrid PSO-GWO

In hybrid PSO-GWO, the first three agents' position is updated in the search space by a mathematical equation 8. Instead of using common mathematical formulas, the exploration and exploitation of the grey wolf in the search space have been controlled by inertia constant [19]. The modified set of dominant equations is as follows:

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - w^*\vec{X}\right|$$

$$\vec{D}\beta = \left|\vec{C}_2 . \vec{X}\beta - w^* \vec{X}\right| \tag{11}$$

$$\vec{D}\delta = \left|\vec{C}_3 . \vec{X}\delta - w^* \vec{X}\right|$$

Where $c_{1,}$ c_2, c_3, and w are constants,

To combine PSO and GWO variants, the velocity and updated equation are calculated as follows:

$$v_i^{t+1} = w^* \left(\begin{array}{l} v_i^t + c_1 r_1 \left(x_1 - x_i^t\right) \\ + c_2 r_2 \left(x_2 - x_i^t\right) + c_3 r_3 \left(x_3 - x_i^t\right) \end{array} \right) \tag{12}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{13}$$

The pseudocode of hybrid PSO-GWO as follows:

```
Initialize $c_{1,}$ $c_2$, $c_{3,}$ $t = 0$,
$w = 0.5 + r/2$, velocity=random (search Agents No. dim),
Postion=dot (random (search Agents No, dim), (ub−lb)) + lb
While (t <Max_iteration)
For each search agent
a=2−t*(2/Max_iteration)
Calculate $A_1$, $A_2$, and $A_3$ according to equation (6)
Calculate the fitness of each search agent using equations (9) and (11)
Update velocity and position of the current search agent according to equations (12) and (13)
End for
t=t + 1
End while.
```

4. PROBLEM STATEMENT

As described in Section 1, machine learning techniques are popular ways of sentiment classification. In this work, to perform sentiment classification, Radial Bias Function kernel-based SVM (RBF-SVM) has been used. The accuracy and performance of this type of SVM mainly depend on the value of two parameters, namely, penalty **"C"** and **"gamma"** which known as hyperplane **"Soft Margin"** parameters. Hence, selecting optimal value for those parameters is a challenge to boost the classification model accuracy. To solve this problem, four meta-heuristic optimizer algorithms: PSO, MPSO, GWO, and hybrid of PSO-GWO have been implemented to select the best values for those parameters.

5. PROPOSED SYSTEM MODEL

In this study, four meta-heuristic optimizer algorithms have been implemented for selecting the best value to ***"Soft Margin" penalty "C" and "gamma"*** parameters to improve the accuracy of the RBF-SVM classifier. The work implemented on Dell Latitude E6540, Intel(R) Core(TM) i7-4610M CPU at 3.00GHz, 8-GB RAM, 64-Bit Windows-7 operating System. Fig. 4 is a flow diagram that displays basic architecture and steps of the proposed sentiment classification model.

5.1. Tweet Collection

To access Twitter and reading tweets from it, you have to make a Twitter developer account that known as Twitter-API. Twitter-API is an interface between the developers and Twitter that enables them to search for tweets based on their desired keyword through some secret key and tokens. In this work a Twitter-API is created called "Twitter-Sentiment-Analysis-20," to collect the most recent tweets according to some keyword such as Trump, Bitcoin, IoT, and Toyota using python code and categorizing to "Positive" and "Negative" using "VADER" [14] lexicon rule-based method then persist in mongo database collection or table. Table 1 shows the details about each keyword dataset size, and Fig. 5 shows a sample of data.

5.2. Pre-processing

Pre-processing means cleaning the text from the least important data. The datasets will go through the following steps for pre-processing task:

Removing duplicate tweets, convert the words to the lowercase, and replace emoticons symbols with a positive or negative opinion, according to Table 2.

The next step is removing URLs, slang correction (omg → oh my god), expand contraction (can't → cannot), stripping punctuation marks, special character and numbers, as well as multiple spaces, clearing from stop words, tokenizing, and

TABLE 1: Dataset size description

Keyword	Positive	Negative	Total
Trump	1339	1626	2965
Bitcoin	4923	2341	7264
IoT	10,700	1929	12,629
Toyota	14,332	6594	20,926
Total	31,294	12,490	43,784

TABLE 2: Emoticons and their meaning

:-), :-D, :-j, =p, :], :3	positive
:(, :[, ^o), :^), :@, =/	negative

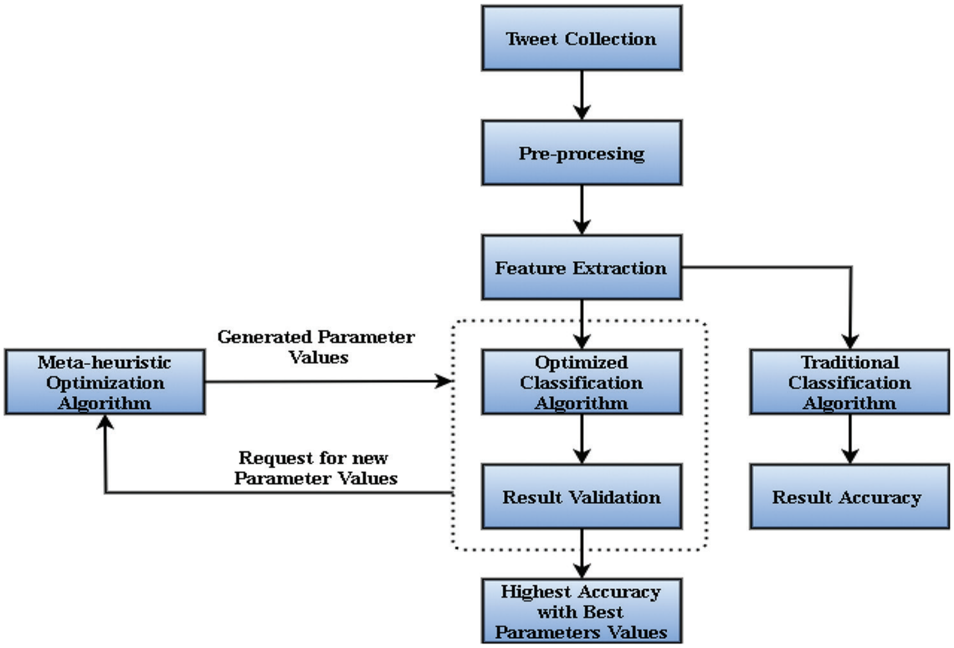
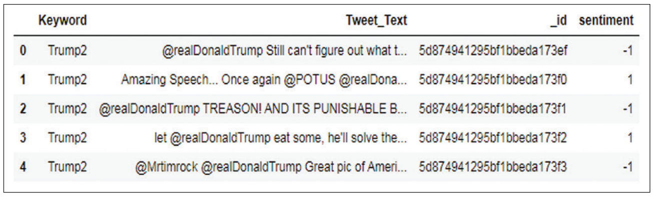

Tweet Collection
Pre-procesing
Feature Extraction
Generated Parameter Values
Meta-heuristic Optimization Algorithm
Optimized Classification Algorithm
Traditional Classification Algorithm
Request for new Parameter Values
Result Validation
Result Accuracy
Highest Accuracy with Best Parameters Values

Fig. 4. Flow diagram of the proposed model.



	Keyword	Tweet_Text	_id	sentiment
0	Trump2	@realDonaldTrump Still can't figure out what t...	5d874941295bf1bbeda173ef	-1
1	Trump2	Amazing Speech... Once again @POTUS @realDona...	5d874941295bf1bbeda173f0	1
2	Trump2	@realDonaldTrump TREASON! AND ITS PUNISHABLE B...	5d874941295bf1bbeda173f1	-1
3	Trump2	let @realDonaldTrump eat some, he'll solve the...	5d874941295bf1bbeda173f2	1
4	Trump2	@Mrtimrock @realDonaldTrump Great pic of Ameri...	5d874941295bf1bbeda173f3	-1

Fig. 5. Sample of collected tweets.

lemmatizing and finally, dropping duplicate tweets after preprocessing and protecting them in another mongo database collection.

5.3. Feature Extraction

Feature extraction is the most important phase. The purpose of this phase is to normalize the data by converting the words into vectors for the classification process. Gensim's deep learning library has been utilized for the numerical representation of each document. Doc2vec is a way of document embedding where each document is mapped to a vector in space. Doc2vec is Gensim's extended library of word2vec, which is used to find vector representations for each word [15]. Doc2Vec was proposed in two models, namely, DBoW and DM. DM is divided into two sub-model, namely, DMC and DMM. After preprocessing, the cleaned tweets will be split into two parts, which are training-set, composed of 80% of tweets, and test-set, composed of 20% of tweets, after that Doc2Vec models has been used to extract features from train-set and test-set. Doc2Vec models and their combination DBoW + DMC and DBoW + DMM are used to extract features from pre-processed tweets.

5.4. Classification

To perform the classification task, the RBF-SVM has been used. SVM one of the well-known supervised machine learning that broadly use in classification and regression tasks due to the ability to work with large amounts of data.

In the first approach, the traditional SVM with default value "1", and "scale" for **C** and **gamma** parameters, used to classify tweets. In the second approach, at each iteration, the RBF-SVM's **"C"** and **"gamma"** parameters took the position value of each agent. After finishing the last iteration, the best accuracy with respect to the best **C** and **gamma** values was presented. Finally, the accuracy of both classification approaches has been compared.

6. RESULTS AND DISCUSSION

Figs. 6-9 show an accuracy comparison between traditional SVM and optimized SVM with different Doc2Vec feature extraction models.

As it is shown in Fig. 6, all optimizers provide a better result for all Doc2Vec feature extraction methods. The hybrid of PSO-GWO provides better results in DBoW and DMC

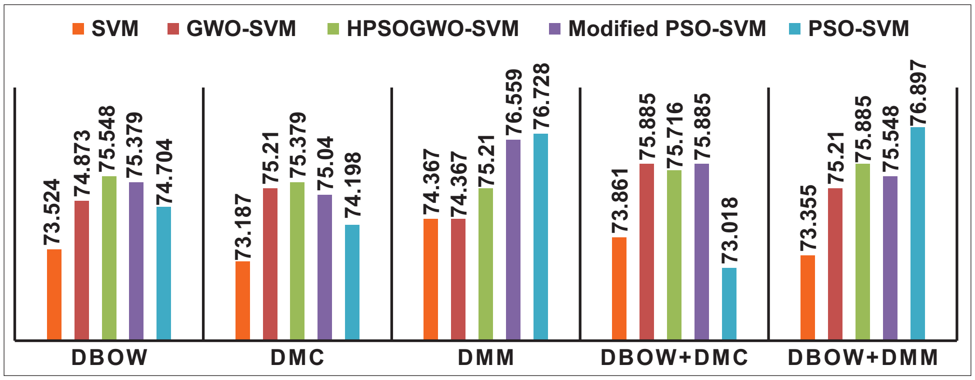


Fig. 6. Results of Trump dataset.

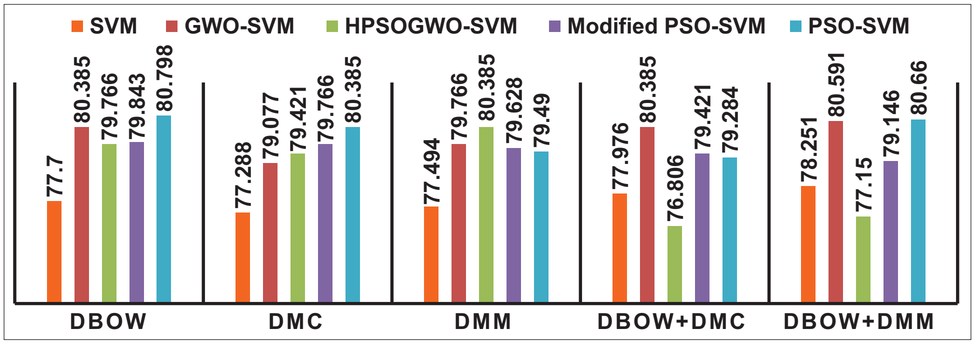


Fig. 7. Results of Bitcoin dataset.

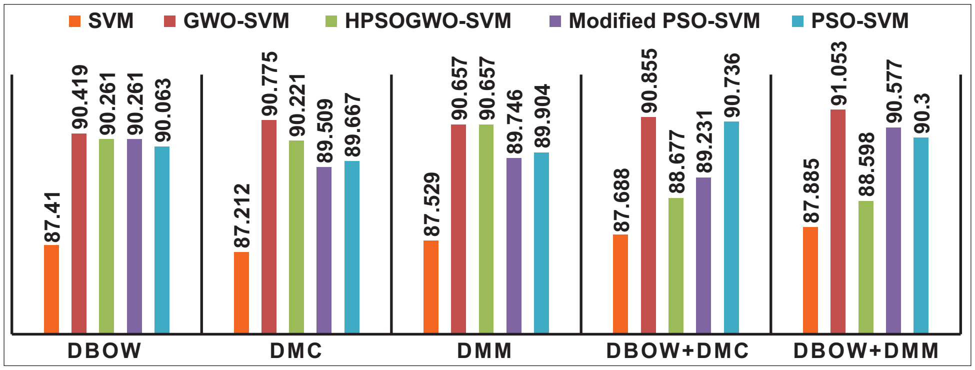


Fig. 8. Results of IoT dataset.

models. Furthermore, MPSO-SVM outperforms original PSO-SVM for DBoW, DMC, and DBoW + DMC models, respectively.

By looking at the “Bitcoin” dataset results, for DBoW, DMC, and DMM models, all optimizers provide a remarkable accuracy compared to traditional SVM, except for hybrid PSO-GWO that could not get expectable result for DBoW + DMC and DBoW + DMM models. MPSO-SVM provides better results than original PSO-SVM for both Doc2Vec DMM and DBoW + DMC models.

The results show that the model accuracy remarkably increased for all optimizers with different Doc2Vec models and their combinations, especially GWO that achieves the highest accuracy result that is 91.093% in DBoW + DMM, followed by MPSO and PSO. In DBoW, DMC, and DMM models hybrid of PSO-GWO provides a better result than PSO and MPSO, but in DBoW + DMC and DBoW + DMM combinations, it increased the model accuracy by <1%.

Finally, Fig. 9 illustrates that all optimizers outperform SVM when used alone, like “IoT” dataset, GWO-SVM

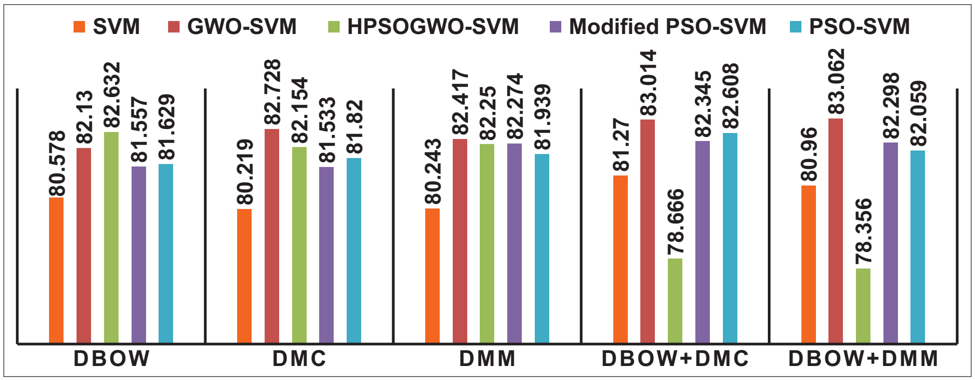


Fig. 9. Results of Toyota dataset.

outperforms other optimizers in all Doc2Vec models. Except for PSO-GWO with SVM that could not grant the expected result for DBoW + DMC and DBoW + DMM the same as the "Bitcoin" dataset.

7. CONCLUSION AND FUTURE WORK

In this work, we have carried out a comparative analysis between classification with traditional RBF-SVM and optimized RBF-SVM using four meta-heuristic optimizers, namely, PSO, MPSO, GWO, and hybrid of PSO and GWO. These optimizers are implemented for selecting the best values for hyperplane "***Soft Margin***" penalty "C" and ***gamma*** parameters of the RBF-SVM classifier. After testing our model on each dataset and with different Doc2Vec feature extraction methods. We came to the point that these optimizers have an important role in enhancing the accuracy of the classifier.

The results show that with a small dataset, MPSO provides a better result than the original PSO. In contrast, with increasing the dataset size, SVM with GWO achieves better accuracy compared to the rest optimizers.

Hybrid of PSO-GWO is effective in improving SVM accuracy in Doc2Vec DBoW, DMC, and DMM models, but it is not work well for combinations of DBoW + DMC and DBoW + DMM because of feature set nature was generated by merging these two models.

In future works, we will try to use these optimizers for parameter optimizing of some deep learning algorithms, i.e., rectified neural network weights to examine whether it performs better results than existing RBF-SVM model or not.

REFERENCES

[1] A. Go, R. Bhayani and L. Huang. "Twitter Sentiment Classification using Distant Supervision". Technical Report, Stanford University. p. 6, 2009.

[2] R. Feldman. "Techniques and applications for sentiment analysis: The main applications and challenges of one of the hottest research areas in computer science". *Communication of the ACM*, vol. 56, no. 4, pp. 82-89, 2013.

[3] N. Bindal and N. Chatterjee. "A two-step method for sentiment analysis of tweets." In: *15th International Conference Information Technology 2016*, Bhubaneswar, pp. 218-224, 2017.

[4] S. K. Jain and P. Singh. "Systematic Survey on Sentiment Analysis". In: *2018-1st International Conference on Secure Cyber Computing and Communication*, Jalandhar, pp. 561-565, 2019.

[5] M. K. Das, B. Padhy and B. K. Mishra. "Opinion mining and sentiment classification: A review". In: *Proceedings of the International Conference on Inventive Systems and Control 2017*, Malaysia, pp. 4-6, 2017.

[6] S. Naz, A. Sharan and N. Malik. "Sentiment Classification on Twitter Data Using Support Vector Machine". *2018 IEEE/WIC/ACM International Conference on Web Intelligence*, Santiago, pp. 676-679, 2019.

[7] P. Seth, A. Sharma and R. Vidhya. "Sentiment analysis of tweets using machine learning approach". *International Journal of Engineering and Technology*, vol. 7, no. 3.12, p. 434, 2018.

[8] A. K. Sharma. and D. S. U. Kumari. "Sentiment Analysis of Smart Phone Product Review using SVM Classification Technique". *International Conference on Energy, Communication, Data Analytics and Soft Computing*, Chennai, India, pp.1469-1474, 2017.

[9] V. S. Rajput and S. M. Dubey. "Stock market sentiment analysis based on machine learning". In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Dehradun, pp. 506-510, 2017.

[10] A. Rane and A. Kumar. "Sentiment classification system of twitter data for us airline service analysis". *International Computer Software and Applications Conference*, vol. 1, pp. 769-773, 2018.

[11] Q. Shuai, Y. Huang, L. Jin and L. Pang. "Sentiment Analysis on Chinese Hotel Reviews with Doc2Vec and Classifiers". In: *018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 1171-1174, 2018.

[12] A. Mukwazvure and K. P. Supreethi. "A Hybrid Approach to

Sentiment Analysis of News Comments". In: *2015 4th International Conference on Reliability, Infocom Technologies and Optimization*, Noida, 2015.

[13] A. C. Flores, R. I. Icoy, C. F. Pena and K. D. Gorro. "An Evaluation of SVM and Naive Bayes with SMOTE on Sentiment Analysis Data Set". In: *2018-4th International Conference on Engineering, Applied Sciences, and Technology, Explor Innovative Smart Solutions Social*, Phuket, pp. 1-4, 2018.

[14] J. Hutto and E. E. Gilbert. "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text". In: *8th International Conference on Weblogs and Social Media*, Michigan, 2014.

[15] Q. Le and T. Mikolov. "Distributed Representations of Sentences and Documents". *31st International Conference on Machine Learning*, vol. 4, pp. 2931-2939, 2014.

[16] M. Bilgin and İ. F. Şentürk. "Sentiment Analysis on Twitter Data with Semi-supervised Doc2Vec". In: *2nd International Conference on Computer Science and Engineering UBMK 2017*, Turkish, pp. 661-666, 2017.

[17] R. Eberhart and J. Kennedy. "New Optimizer Using Particle Swarm Theory". In: *Proceedings International Symposium on Micro Machine and Human Science*, New York, pp. 39-43, 1995.

[18] S. Mirjalili, S. M. Mirjalili and A. Lewis. "Grey wolf optimizer". *Advances Engineering Software,* vol. 69, pp. 46-61, 2014.

[19] N. Singh and S. B. Singh. "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance". *Journal of Applied Mathematics*, vol. 2017, pp. 15, 2017.