Computers, Materials & Continua
DOI:10.32604/cmc.2021.019047
Article

Tech Science Press

Swarming Behavior of Harris Hawks Optimizer for Arabic Opinion Mining

Diaa Salam Abd Elminaam[1,2,*], Nabil Neggaz[3], Ibrahim Abdulatief Ahmed[4,5] and Ahmed El Sawy Abouelyazed[4]

[1]Department of Information Systems, Faculty of Computers and Artificial Intelligence, Benha University, 12311, Egypt
[2]Department of Computer Science, Faculty of Computer Science, Misr International University, Egypt
[3]Universite des Sciences at de la Technologie d'Oran Mohamed Boudiaf, USTO-MB, BP 1505, EL M'naouer, 31000 Oran, Laboratoire Signal Image Parole (SIMPA), Department d'informatique, Faculte des Mathematiques et Informatique, Algerie
[4]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, 12311, Egypt
[5]Department of Computer Science, Obour High Institute for Computers and Informatics, Egypt
[*]Corresponding Author: Diaa Salam Abd Elminaam. Email: diaa.salama@fci.bu.edu.eg
Received: 31 March 2021; Accepted: 05 May 2021

Abstract: At present, the immense development of social networks allows generating a significant amount of textual data, which has facilitated researchers to explore the field of opinion mining. In addition, the processing of textual opinions based on the term frequency-inverse document frequency method gives rise to a dimensionality problem. This study aims to detect the nature of opinions in the Arabic language employing a swarm intelligence (SI)-based algorithm, Harris hawks algorithm, to select the most relevant terms. The experimental study has been tested on two datasets: Arabic Jordanian General Tweets and Opinion Corpus for Arabic. In terms of accuracy and number of features, the results are better than those of other SI based algorithms, such as grey wolf optimizer and grasshopper optimization algorithm, and other algorithms in the literature, such as differential evolution, genetic algorithm, particle swarm optimization, basic and enhanced whale optimizer algorithm, slap swarm algorithm, and ant–lion optimizer.

Keywords: Arabic opinion mining; Harris hawks optimizer; feature selection; AJGT and OCA datasets

1 Introduction

The development of information technology and associated services forums, specialized sites, etc.—has opened the doors to a vast mode of opinion expression on a wide range of subjects. It prompted us to study the opinions of the public and particular people, such as consumer reviews. The comments and responses expressed in various blogs, forums, and social media platforms are considered essential sources of textual data that can be analyzed to derive useful information [1]. Opinion mining or opinion analysis is a branch of the automatic extraction of knowledge from data that uses computational linguistics techniques to assess the opinions expressed in the textual form [2]. The richness of social media in terms of opinion and sentiment has sparked research interest. This interest is intense for the Arabic language, given the massive number of internet users speaking Arabic and its dialects. The term "opinion mining" refers to the automatic processing of opinions, feelings, and subjectivity in texts. It is well known as word polarity extraction or sentiment analysis (SA), often associated with a classification problem on evaluative texts, such as those available on Amazon and Facebook. Recently, with the COVID-19 pandemic, great importance has been attached to social networks and online shopping. Therefore, analyzing opinions has become essential in daily activities; it is paramount for business enterprises to respect consumers' opinions to increase their profits [3].

The process of massive data resulted from social media, such as Facebook and Twitter, required to apply SA over the text. However, several features contain irrelevant information, negatively influencing the classification results based on machine learning (ML) techniques. Thus, feature selection (FS) has been employed for several natural language processing (NLP) applications [4]. FS can be classified into three as a filter, wrapper, and embedded techniques [5]. In the first technique, the process of FS is based on the results of learner performance, and the correlation between the features is used during the process of evaluation (no external evaluators are involved). In the third technique, the classifier is trained by the available features. The obtained results are used for evaluating the correlation of each attribute. FS-based wrapper method engages the classifier into the ranking process using a subset of features. Various meta-heuristic optimization algorithms have been proposed to solve complex optimization problems, such as text document clustering, data mining, image segmentation, computer vision, and opinion mining. Several inspirations are derived from natural biological behavior such as genetic algorithm (GA) [6], differential evolution (DE) [7], and genetic programming [8], swarm intelligence (SI) artificial bee colony [9], grey wolf optimizer (GWO) [10], whale optimization algorithm (WOA) [11], Improved whale optimization algorithm [12], sports volleyball premier league [13], league championship algorithm [14], and football optimization algorithm [15] and physical/mathematical rules sine–cosine algorithm (SCA) [16], thermal exchange optimization [17], and Henry gases solubility optimizer [18], fruit fly optimization [19], and Big data analytics using spark [20].

Several studies on natural languages, such as English, French, and Spanish, have been conducted using SA. This due to their formal nature, unlike the Arabic language, which can be depicted using formal and informal language or dialectical language (Algerian, Moroccan, Tunisian, Egyptian, and Jordanian dialects, to mention only a few), spoken by 423 million people. Therefore, Arabic SA (ASA) is still challenging due to its vast vocabulary, different dialects, and the Qur'an language. Besides, several SI- and physical/mathematical-inspired algorithms are used for FS [21,22], which motivated us to treat ASA using Harris hawks optimizer (HHO).

The main contributions of this paper are as follows:

- Designing a new framework of Arabic sentiment by imitating the behavior of Harris hawks.
- We are introducing the wrapper FS using HHO for Arabic opinion mining (AOM).
- Comparing the performance of HHO with well-known optimizers, such as GWO, SCA, and grasshopper optimizer algorithm (GOA), using two Arabic opinion datasets—Arabic Jordanian General Tweets (AJGT) and Opinion Corpus for Arabic (OCA).
- Comparing the efficiency of HHO to state-of-the-art methods, such as DE, GA, particle swarm optimization (PSO), primary and modified WOA, slap swarm algorithm (SSA), and ant–lion optimizer (ALO).

The remainder of this paper is laid out as follows: We present a detailed related work in Section 2. The preprocessing stage of the NLP, GWO, and k-NN classifiers is discussed in Section 3. The architecture of FS for AOM based on HHO is defined in Section 4. Section 5 describes the data and metrics that were used, as well as the findings that were obtained. Finally, in Section 6, we summarise our findings and discuss future research directions.

2 Related Work

Several studies have been conducted for the ASA. For example, five ML classifiers, such as support vector machine (SVM), stochastic gradient descent, naive Bayes (NB), multinomial NB (MNB), and decision tree (DT), have been employed on a large scale Arabic book review dataset. The obtained results showed that the MNB classifier has tremendous potential compared with other algorithms. The authors used several feature extraction models based on these classifiers. The experimental study showed that the best performance is obtained by the MNB classifier using the unigram. Finally, GA is introduced by [23] as a new contribution to select relevant features for the MNB classifier, which enhanced the classification rate to 85%.

As part of the research conducted by [24], a novel dataset for ASA called AJGT was designed. The authors compared the efficiency of SVM and NB classifiers using different scenarios of preprocessing fusions. Mainly, they compared three techniques for extracting characteristics based on N-grams, such as unigrams, bigrams, and trigrams, tested using the AJGT dataset. Besides, a fair comparison was realized using the TF/TF–IDF weighting technique (TF: term frequency; IDF: inverse document frequency). The experimental study showed that the fusion between SVM and TF–IDF weighting method outperformed other techniques and achieved an accuracy rate of 88.72% and 88.27%, respectively, in terms of F-measure.

A set of ML classifiers based on the majority voting algorithm combined with four classifiers, including NB, SVM, DT, and ***k***-NN, has been proposed [25] for ASA. The experiments showed that the set of ML classifiers have better performance compared with the basic classifiers. The voting method highlighted a practical classification approach for ASA. It uses different classifiers to classify each case. The majority vote of all classifiers' decisions is combined to predict the instance under test.

In [26], the authors enhanced the basic WOA for solving the problem of AOM based on FS by adopting an improved WOA (IWOA). The novelty of their work is the merging of two phases of dimensional reduction. The first phase used a filter based on the information gain (IG) method, which a wrapper WOA will optimize. The IWOA employed several operators, such as elite opposite learning and evolutionary operators, inspired by DE optimizer to produce a new generation. The IWOA obtained a significant result in terms of classification accuracy and the selected ratio compared with other optimization algorithms over several datasets.

A new hybrid system was designed for ASA based on filter and wrapper FS as IG and SSA, respectively [27]. The proposed method was assessed using the AJGT dataset, and the obtained results achieved 80.8% accuracy.

The authors of [28] designed a new tool for ASA using GWO. Their idea comprises selecting the features using wrapper GWO to determine the polarity of opinions. The experiment was conducted using two datasets (AJGT and OCA). The GWO achieved approximately 86% and 95% for AJGT and OCA, respectively.

3 Background

3.1 Preprocessing Step

Before the learning phase for ASA, the preprocessing steps are required to convert text features to vectors, crucial. This study employed steps such as tokenization, noise removal, stop word removal and stemming [29].

3.1.1 Tokenization

The process of tokenization comprises identifying words and phrases in a text. Simple tokenization can use white space or the carriage return as a word separator. Notably, punctuation marks ("?" "!" and ".") are very useful in separating sentences.

3.1.2 Noise Removal

The result of the tokenization process provides two types of tokens.

- The first corresponds to recognizable units, including punctuation marks, numeric data, and dates.
- The second requires deep morphological analysis. In this context, the tokes defined by one or two characters, non-Arabic language, and digit numbers are eliminated.

3.1.3 Stop Word Suppression

Stop words correspond to terms that appear in texts but do not contain useful information. This process is to eliminate stop words. These words are usually personal pronouns, articles, prepositions, or conjunctions. A dictionary of stop words is usually employed for eliminating the same from the text.

3.1.4 Stemming

Stemming is the extraction of lexical root or stem by adopting morphological heuristics to remove affixes from words before indexing them. For example, the Arabic words عامل ، عمل ، معمل share the same root عمل.

3.2 Feature Extraction

After the preprocessing phase, the dataset must be prepared in a suitable form to start the learning phase. Consequently, the most relevant text features are extracted and converted into vectors. The vector space is represented as a two-dimensional matrix, where the columns denote the features, and the rows denote the documents (reviews). The entries of the matrix are the weights of the features in their corresponding reviews. The TF–IDF scheme is employed to assign weights to terms [30]. The weight is determined from Eqs. (1)–(3) as follows:

$$TF(i,j) = \frac{\text{Frequency of term i in review } j}{\text{Total number of terms in review } j} \tag{1}$$

$$IDF(i,j) = \log\left(\frac{\text{Total number of reviews in the datasets}}{\text{Number of reviews which include i term}}\right) \tag{2}$$

$$W(i,j) = TF(i,j) \times IDF(i,j) \tag{3}$$

TF(i, j) is the frequency of term i in review j, IDF(i, j) is the frequency of features concerning all reviews. Finally, the weight of feature i in review j, W(i, j) is calculated by Eq. (3).

3.3 Harris Hawks Optimization

The HHO [31] as a new SI algorithm is inspired by the cooperative behaviors of Harris hawks in hunting preys. Harris hawks demonstrate various chasing styles depending on the dynamic nature of circumstances and escaping patterns of a prey. In this intelligent strategy, several Harris hawks try to attack from different directions cooperatively and simultaneously converge on a detected prey outside the cover, showing different hunt strategies. The candidate solutions are the Harris hawks, and the intended prey is the best candidate solution (nearly the optimum) in each step. The three phases of the HHO algorithm are highlighted as follows: exploration phase, the transition from exploration to exploitation phase, and exploitation phase.

3.3.1 The Exploration Phase

The hunting is modeled as follows:

$$x_{t+1}^{i} = \begin{cases} x_{\text{rand}} - \tau_1 |x_{\text{rand}} - 2\tau_2 x_t^i| & \text{if} \quad \tau_5 \geq 0.5 \\ (x^{\text{best}} - \overline{x_t}) - \tau_3 |lb^j + \tau_4 (ub^j - lb^j)| & \text{else} \end{cases} \quad t \in [1 \cdots T], \quad i \in [1 \cdots N] \tag{4}$$

where the current position of i$^{\text{th}}$ hawk and its new position in iteration t + 1 is represented by x_t^i and x_{t+1}^i; x_{rand} and x^{best} Are randomly selected hawk location and the best solution (target: rabbit). Lower and upper bounds of the j$^{\text{th}}$ dimension are denoted by lb^j and ub^j; τ_1–τ_5 are random numbers in the interval [0, 1]. The average hawk position $\overline{x_t}$ is defined as follows:

$$\overline{x_t} = \frac{1}{N} \sum_{i=1}^{N} x_t(i); \quad N, \textit{ is the number of hawks} \tag{5}$$

In Eq. (4), the first scenario ($\tau_5 \geq 0.5$) grants a chance to the hawks to hunt randomly, spreading in the planned space; meanwhile, the second scenario explains context when the hawks hunt beside other hawks close to a target.

3.3.2 The Transformation from Exploration to Exploitation

In this phase, the prey attempt to escape from the capture, so the escaping energy E_n level of the prey decreases gradually. The energy is given by

$$E_n = 2 * E_{n0} * \left(1 - \frac{t}{T}\right) \tag{6}$$

where the initial energy (E_{n0}) is defined by $E_{n0} = 2 * \text{rand} - 1$, randomly changed inside (−1, 1), and T is the maximum number of iterations. HHO remains in the exploration mode as long as $|E_n| \geq 1$, and hawks continue exploring global regions, whereas it swaps into exploitation mode when $|E_n| < 1$. R refers to escaping probability of the target.

3.3.3 The Exploitation Phase

It aims to avoid fall into local optima. According to the value of energy escaping and the value of R, four strategies are applied named: surrounding soft, surrounding hard, surrounding soft beside advanced rapid dives, and surrounding hard beside advanced rapid dives.

The first task (surrounding soft): The surrounding soft can be formulated mathematically when R ≥ 12, and the level of energy is greater than 1, 2 (i.e., $|E_n| \geq 12$), given by

$$\begin{aligned} x_{t+1}^{i} &= \Delta x_t^i - E_n |J x^{\text{best}} - x_t^i| \\ \Delta x_t^i &= x^{\text{best}} - x_t^i, \quad J = 2(1-\tau_6) \end{aligned} \tag{7}$$

where Δx_t^i denotes the distance between the best prey (a rabbit) and the i$^{\text{th}}$ hawk's current location. J denotes the prey's random strength jump, and τ_6 is a random number between 0 and 1.

The second task (surrounding hard): When the level of energy is less than 12 ($|E_n| < 12$) and R ≥ 12, the rabbit becomes exhausted, and the possibility of escaping low (or escaping becomes hard) because the level of energy is decreased. This behavior can be modeled by

$$x_{t+1}^{i} = x^{best} - E_n |\Delta x_t^i| \tag{8}$$

The third task (surrounding soft beside advanced rapid dives): This task is applicable when the level of energy is greater than 1 2 ($|E_n| > 1\ 2$) and R < 1 2, where the rabbit still has sufficient energy to run away. Hence, the hawk tries progressive dives to take the best position to catch the rabbit. This behavior is modeled by integrating the Lévy flight function [32].

The position of i$^{\text{th}}$ hawk should be modified

$$\begin{aligned} x_{t+1}^{i} &= \begin{cases} y & \text{if fit } (y) < \text{ fit}(x_t^i) \\ z & \text{if fit } (z) < \text{ fit}(x_t^i) \end{cases} \\ y &= x^{best} - E_n |J x^{best} - x_t^i| \\ z &= y + r_v \times Lv(D) \end{aligned} \tag{9}$$

with

$$Lv(D) = 0.01 \times \frac{\text{rand}(1, D) \times \sigma}{|\text{rand}(1, D)|^{\frac{1}{\beta}}} \tag{10}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{11}$$

where D is the dimensionality space, r_v contains D components generated randomly in the interval (0, 1), Lv represents the Lévy flight function, β is a constant with default $\beta = 1.5$, and fit indicates the fitness function computed by Eq. (9).

The fourth task (surrounding hard beside advanced rapid dives): In this task, it is assumed that R < 1 2 and the level of energy is less than 1 2 ($|E_n| < 1\ 2$), the prey has a lower energy level to escape, and the hawks are close to realizing successive dives for catching. This process can be described by

$$
\begin{aligned}
&x_{t+1}^{i} = \begin{cases} y & \text{if } \; fit(y) < fit(x_t^i) \\ z & \text{if } \; fit(z) < fit(x_t^i) \end{cases} \\
&y = x^{best} - E_n \lvert J x^{best} - \bar{x} \rvert \\
&z = y + r_v \times Lv(D)
\end{aligned}
\tag{12}
$$

The general steps of HHO are depicted in Algorithm 1.

Algorithm 1: Pseudo-code of HHO algorithm

Inputs: The population size N and maximum number of iteration T
Outputs: The position of rabbit and its cost
Initialize a random population x i (i = 1, 2,..., N)
t = 1
while (t $\leq$ T) **do**
  Compute the fitness values of hawks
  Set x^{best} as the position of rabbit (best solution)
  For (each hawk (x^i)) **do**
    Update the initial energy E_{n0} and strength jump J E_{n0} = 2rand( )–1, J = 2(1-rand( ))
    Update the En using Eq.
    if (|En| $\geq$ 1)
    Adjust the current vector using Eq. (4)
    end if
    if (|En| < 1) **then** Exploitation mode
      if (R $\geq$ 0.5 and $|E_n| \geq 0.5$) **then** Surrounding
        Adjust the current vector using Eq. (7)
      else if (R $\geq$ 0.5 and |En| < 0.5) **then** Surrounding
        Adjust the current vector using Eq. (8)
      else if (R<0.5 and $|E_n| \geq 0.5$), **then** Surrounding Soft beside advanced rapid dives
        Adjust the current vector using Eq. (9)
      else if (R < 0.5 and $|E_n| \geq 0.5$) **then** Surrounding Hard beside advanced rapid dives
        Adjust the current vector using Eq. (12)
      end if
    end if
  end for
  t = t + 1
end while
Return x^{best}

4 HHO for AOM

This section explains the process of AOM deeply using HHO. After extracting the TF–IDF matrix, HHO aims to keep the relevant terms by ensuring a compromise between high accuracy and a low number of selected features. The following steps summarize the required steps for AOM.

4.1 Initialization Phase

In this step, HHO generates N swarm agents in the first population, where each individual represents a set of terms (features) to be selected for evaluation. The population X is generated as follows:

$$x_i^j = \mathrm{Min}^j + \delta^j \times (\mathrm{Max}^j - \mathrm{Min}^j), \quad i = 1, \ldots, N; \quad j = 1, \ldots, D \tag{13}$$

The minimum and maximum bounds, Max^j and Min^j Respectively, for each candidate solution, i, are in the range of [0, 1]. The δ^j is a random number between 0 and 1. An intermediate binary conversion step is necessary before fitness evaluation to select a subset of terms. So, each solution x^i is converted using binary operator (x^i_{bin}) as follows:

$$x^i_{\mathrm{bin}} = \begin{cases} 1 & \text{if} \quad x^i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

For example, we generate a solution x^i that contains five terms in TF–IDF as $x^i =$ [0.6, 0.2, 0.9, 0.33, 0.75]. The operation of conversion is applied using Eq. (14) to generate a binary vector $x^i_{\mathrm{bin}} =$ [1, 0, 1, 0, 1], where one should be selected, and zero should be deselected. It means that the first, third, and last terms in original datasets are relevant and should be selected, whereas the others are irrelevant features and should be eliminated. After determining the subset of the selected terms, the fitness function is calculated for each agent x^i_{bin} To determine the quality of these features. The fitness of the i^{th} solution is defined by

$$fit^i = \zeta \times Er^i + (1 - \zeta) \times \frac{d^i}{D} \tag{15}$$

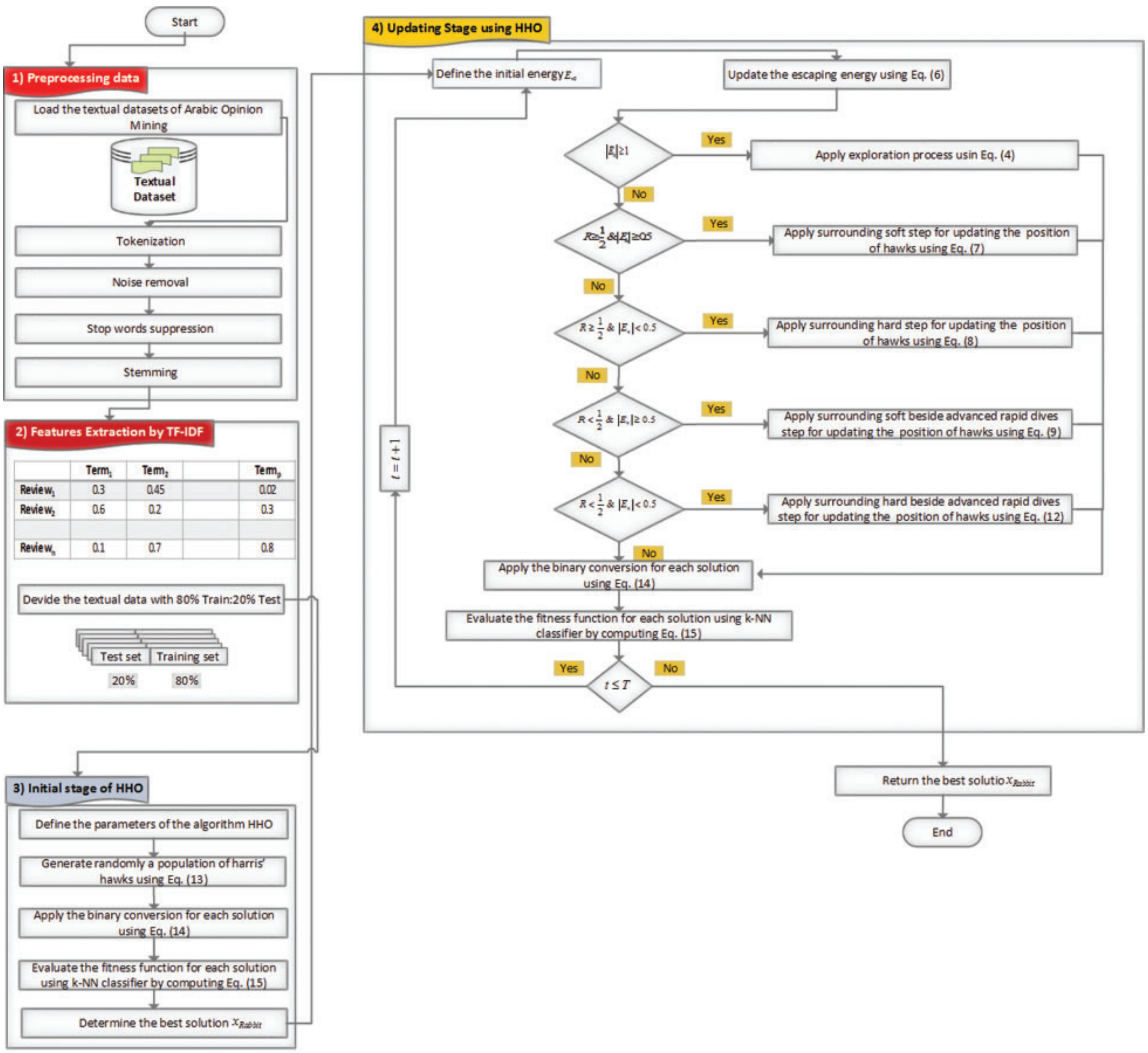
where $\zeta = 0.99$ represents the equalizer parameter employed to ensure a relationship between the error rate of classification ($Er^i = 1 -$ Accuracy) and the size of selected terms (d^i); D is the total size of terms in the original dataset. The k-NN is utilized as a classifier in the FS cycle. The hold-out strategy is used as a classification strategy, which divides the dataset into training and test sets of 80% and 20%. Er^i denotes the error rate of test datasets computed by k-NN [33]. The lower value of fitness through all agents is assigned to the best prey ($\mathrm{x}^{\mathrm{best}}$).

4.2 Updating Phase

The process of updating solutions consists of the exploration phase, which aims to apply a global search when the energy is more significant than one. Afterward, the transformation from exploration to exploitation is applied. Then, the exploitation phase is employed, which contains four tasks surrounding soft, surrounding hard, surrounding soft beside advanced rapid dives, and surrounding hard beside advanced rapid dives.

The process is reproduced while the termination condition is met. The stop criterion corresponds to the maximum amount of iterations that evaluate the HHO algorithm's performance. Then, the best solution x^{best} Returned and converted to determine the number of relevant features.

The ASA framework required three steps preprocessing data, features extraction, and FS using HHO. In the first step (preprocessing data), the Arabic reviews are treated by tokenization, noise removal, stop words suppression, and stemming. The second step consists of converting the text to a vector shape model by weighting each term using TF–IDF. The third step used HHO as a wrapper FS. The detailed ASA framework using HHO is shown in Fig. 1.



	$Term_1$	$Term_2$		$Term_p$
$Review_1$	0.3	0.45		0.02
$Review_2$	0.6	0.2		0.3
$Review_n$	0.1	0.7		0.8

Figure 1: The proposed model of HHO for Arabic opinion mining

5 Experimental Results and Discussion

In this section, several tests and experiments were performed to determine the efficiency of HHO for ASA. Two datasets are exploited to automatically determine the nature of opinion review (positive/negative) OCA and AJGT datasets. First, the experiment results are compared with well-known population-based algorithms GWO, SCA, and GOA tested 30 times using ten search agents and T = 100. Second, the performance of HHO is compared with some works in the literature, which used the same datasets as WOA, IWOA, SSA, GA, PSO, DE, and ALO.

5.1 Datasets Description

- AJGT dataset: This data was gathered from Twitter on various subjects, including arts and politics. It contains 2000 Arabic tweet reviews, 1000 of which are positive and 1000 of which are negative. Due to the differences between Modern Standard Arabic and the Jordanian dialect, this data presents a significant challenge [34].
- OCA dataset: This data was compiled from Arabic film blogs and web pages devoted to Arabic film reviews. There are 500 Arabic reviews, evenly divided into binary categories (250 positives and 250 negatives) [35].

After the preprocessing steps, the TF–IDF allows determining 3054 and 9404 terms for AJGT and OCA datasets, respectively.

5.2 Parameters Settings

Parameters settings of the GWO, SCA, GOA and HHO algorithms are listed in Tab. 1.

Table 1: Parameters settings of the SI algorithms

Algorithms	Parameters setting
GWO	$a \in [2; 0]$ Wolves number (N = 10) T = 100 D indicates to features number.
SCA	$a \in [2; 0]$ N = 10 T = 100 D indicates to features domain
GOA	Cmax = 1 and Cmin = 0.00004 number of agents (N = 10) T = 100 D indicates to features number
HHO	N = 10 T = 100 D indicates to features number $\beta =$ 1.5 used in Lévy flight function [29]

5.3 Evaluation Measures

To investigate the efficiency of the HHO algorithm in the area of ASA-based FS. First, we define the confusion matrix depicted by Tab. 2. Then, specific metrics must be evaluated, such as Accuracy (Ac), Recall (Re), Precision (Pr), and F-score (Fsc).

Table 2: Confusion matrix

	Predicted	
Actual	1	0
1	TP	FN
0	FP	TN

- **TP:** The classifier manages to identify the text as being a positive opinion.
- **TN:** The classifier manages to identify the text as being a negative opinion.
- **FP:** The classifier identifies the text as being a positive opinion knowing that the actual label indicates that the review is negative.
- **FN:** The classifier identifies the text as negative, knowing that the actual label indicates positive reviews.

In this study, we note that the HHO algorithm is executed 30 times. So, all metrics are expressed in terms of average with their standard deviation. In addition, for comparing the efficiency of HHO, three meta-heuristic algorithms GWO, SCA, and GOA, were employed under the same conditions.

• Mean accuracy (μ_{Ac}): The accuracy metric (AC) represents the rate of correct data classification, given by

$$Ac = \frac{Tp + Tn}{Tp + Fn + Fp + Tn} \tag{16}$$

The number of runs is fixed to 30, so the mean accuracy μ_{Ac} is calculated as follows:

$$\mu_{Ac} = \frac{1}{30} \sum_{k=1}^{30} Ac_k^{best} \tag{17}$$

• Average recall (μ_{Re}): The recall metric (Re) is also called actual positive rate, which indicates the percentage of predicting positive reviews, given by

$$Re = \frac{Tp}{Tp + Fn} \tag{18}$$

Thus, μ_{Re} is calculated from the best prey (x_{best}) using

$$\mu_{Re} = \frac{1}{30} \sum_{k=1}^{30} Re_k^{best} \tag{19}$$

• Average precision (μ_{Pr}): The precision (Pr) indicates the rate of correctly predicted samples, given by

$$\mathrm{Pr} = \frac{Tp}{Fp + Tp} \tag{20}$$

Thus, the average precision (μ_{Pr}) can be computed by the following equation:

$$\mu_{Pr} = \frac{1}{30} \sum_{k=1}^{30} Pr_k^{best} \tag{21}$$

• Average fitness value (μ_{fit}): The fitness value metric evaluates the performance of algorithms, which puts the relationship between minimizing the error rate of classification and reducing the selection ratio as in Eq. (15). The average fitness value is given by

$$\mu_{fit} = \frac{1}{30} \sum_{k=1}^{30} fit_k^{best} \tag{22}$$

• The average size of selected features (μ_{size}): This metric represents the size of relevant features. It is computed as follows:

$$\mu_{\text{size}} = \frac{1}{30} \sum_{k=1}^{30} d_{(k)}^{\text{best}} \tag{23}$$

where $d_{(k)}^{\text{best}}$ denotes the cardinality of the best agent's selected features for the k^{th} execution.

• Mean F-score ($\mu_{F_{Sc}}$): This metric represents the harmonic average between recall and precision. It is already used for balanced data, which can be computed as follows:

$$F_{Sc} = 2 \times \frac{Re \times Pr}{Re + Pr} \tag{24}$$

Thus, the mean F-score can be determined by

$$\mu_{F_{Sc}} = \frac{1}{30} \sum_{k=1}^{30} F_k^{Sc-best} \tag{25}$$

• Average CPU time (μ_{Cpu}): It is the average of computation time for each method, given by

$$\mu_{Cpu} = \frac{1}{30} \sum_{k=1}^{30} T_k^{best} \tag{26}$$

• Standard deviation (σ): This is the quality of each algorithm and analysis of the obtained results over different executions and metrics. It is calculated for all metrics defined above.

5.4 Results and Discussions

In terms of the average and standard deviations of fitness and CPU time, Tab. 3 reports the mean fitness values obtained by the HHO, GWO, SCA, and GOA algorithms. It can be deduced that HHO outperformed the other for both AJGT and OCA datasets. The GWO and SCA ranked second for AJGT and OCA datasets, respectively. In addition, the GOA is the worst optimizer for both datasets. The CPU time consumed by the HHO and counterparts is listed in Tab. 3. From the results, it can be observed that the SCA is very fast, especially for the OCA dataset, when the number of reviews is lower, whereas the HHO and GOA require more time. The complex exploitation/exploration operators can interpret this behavior. For both datasets, the SCA provides the lowest time due to using a simple updating operator using trigonometric functions.

Table 3: Performance of HHO and counterpart algorithm for AOM based on fitness metric

Algorithms	AJGT datasets		OCA datasets	
	μFit	σFit	μFit	σFit
GWO	0.1471	0.0142	0.0814	0.0165
GOA	0.1958	0.0127	0.0980	0.0229
SCA	0.1771	0.0147	0.0535	**0.0136**
HHO	**0.1227**	**0.0096**	**0.0396**	0.0158

In terms of mean and standard deviations of accuracy and selected features, the performance of four swarm competitor algorithms in terms of accuracy and number of selected features is illustrated in Tabs. 5 and 6. It is essential to highlight that the HHO achieves a high classification accuracy of 88.28% while keeping 2042 features from 3054 for the AJGT dataset. In addition, it can be observed that the HHO recognizes most of the OCA reviews correctly, with 96.40% in terms of accuracy. Moreover, the SCA finds the most informative features, exhibiting high accuracy for both datasets used. However, similar performance was seen between the GWO and SCA for the OCA datasets. A slight advantage is shown for SCA with a margin of 0.8% in terms of average accuracy. From Tab. 4, it can be seen that SCA determines the optimal set of terms by keeping 1178 terms from 3054 provided by TF–IDF for the AJGT dataset. Further, the HHO can eliminate 7459 irrelevant terms for the OCA dataset.

Table 4: Performance of HHO and counterpart algorithms for AOM based on CPU time

Algorithms	AJGT datasets		OCA datasets	
	μCpu	σCpu	μCpu	σCpu
GWO	536.0240	47.3335	169.6582	16.4972
GOA	355.5719	**7.1278**	259.3000	**5.2120**
SCA	**230.9038**	193.5631	**65.6704**	8.0070
HHO	764.8749	36.3333	204.5147	3373.9142

Table 5: Performance of HHO and counterpart algorithms for AOM based on accuracy metric

Algorithms	AJGT datasets		OCA datasets	
	μAc	σAc	μAc	σAc
GWO	0.8589	0.0146	0.9400	0.0175
GOA	0.8072	0.0128	0.9060	0.0232
SCA	0.8250	0.0149	0.9480	**0.0140**
HHO	**0.8828**	**0.0096**	**0.9640**	0.0158

Table 6: Performance of HHO and counterpart algorithms for AOM based on selected features size

Algorithms	AJGT datasets		OCA datasets	
	$\mu Size$	$\sigma Size$	$\mu Size$	$\sigma Size$
GWO	2274.3000	249.4960	3938.9000	1249.0000
GOA	1518.3000	**27.4530**	4686.9000	56.0563
SCA	**1176.5000**	161.1834	**1944.7000**	55.9733
HHO	2041.7000	287.5730	3748.9000	**1050.3627**

In terms of the average and standard deviations of recall and precision metrics, the comparison of the performance of four meta-heuristics algorithms based on recall and precision is illustrated in Tabs. 7 and 8. The performance of HHO in terms of recall and precision is better than all other counterpart algorithms for both datasets. We can observe a clear advantage obtained by the HHO in terms of standard deviation based on recall and precision metrics due to a good balance between the exploration and exploitation operators. It provides more stability to the algorithm.

Table 7: Performance of HHO and counterpart algorithms for AOM based on recall metric

Algorithms	AJGT datasets		OCA datasets	
	μRe	σRe	μRe	σRe
GWO	0.8603	0.0144	0.9150	0.0190
GOA	0.8089	0.0130	0.8960	0.0262
SCA	0.8266	0.0145	0.9447	**0.0151**
HHO	**0.8839**	**0.0096**	**0.9591**	0.0185

In terms of mean and standard deviations of the F-score, Tab. 9 summarized the mean and standard deviation of the F-score. For both datasets, the HHO outperforms other algorithms in terms of average F-score. A high advantage is highlighted for the OCA dataset compared with the AJGT dataset due to the standard Arabic language instead of Jordanian dialect. In addition, low standard deviation values are obtained, which indicate stability.

Table 8: Performance of HHO and counterpart algorithms for AOM based on precision metric

Algorithms	AJGT datasets		OCA datasets	
	μPr	σPr	μPr	σPr
GWO	0.8650	0.0134	0.9269	0.0186
GOA	0.8115	0.0151	0.9130	0.0225
SCA	0.8323	0.0119	0.9494	0.0154
HHO	**0.8872**	**0.0101**	**0.9684**	**0.0138**

Table 9: Performance of HHO and counterpart algorithms for AOM based on F-score metric

Algorithms	AJGT datasets		OCA datasets	
	μFs	σFs	μFs	σFs
GWO	0.8650	0.0134	0.9269	0.0186
GOA	0.8115	0.0151	0.9130	0.0225
SCA	0.8323	0.0119	0.9494	0.0154
HHO	**0.8872**	**0.0101**	**0.9684**	**0.0138**

5.5 A Numerical Example of HHO Based AOM

To understand deeply the process of the HHO algorithm for Arabic Opinion Mining based feature selection. A numerical example is illustrated for selecting the important terms extracted using TF-IDF. We consider a population with four solutions (Popinit) that contains 3054 words for the AJGT dataset (features) as shown in Tab. 10.

Table 10: Initial population in the range [0, 1]

Initial population	Word1	Word2	Word3		Word3053	Word3054
Sol1	0.6367	0.4933	0.7766	...	0.4850	0.5653
Sol2	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol3	0.0352	0.0966	0.6581	...	0.0841	0.8054
Sol4	0.4224	0.7952	0.3922	...	0.6085	0.7868

After initializing the first population, we evaluate the fitness of each solution. So, this step required an intermediate process called binary conversion based on thresholding operator as depicted in Tab. 11, i.e., If the value is more significant than 0.5, the word is selected else the word is eliminated. Also, the fitness is computed by introducing a k-NN classifier, which allows assessing the fitness using Eq. (15). It can be seen that the second solution represents the best solution with a fitness value of 0.2431.

For each solution, some control parameters are generated to apply the adequate steps of HHO (Exploration, transition from exploration to exploitation, and exploitation). Tab. 12 shows the value of escaping energy (En) computed by Eq. (6) and a random number (R). The last column indicates the adequate operator of HHO, which will be applied.

Table 11: The binary operator

$x^{bin}_{i,j}$	$x^{bin}_{i,1}$	$x^{bin}_{i,2}$	$x^{bin}_{i,3}$		$x^{bin}_{i,3053}$	$x^{bin}_{i,3054}$	fit
Sol1	1	0	1	...	0	1	0.2653
Sol2	0	0	1	...	0	0	**0,2431**
Sol3	0	0	1	...	0	1	0,2714
Sol4	0	1	0	...	1	1	0,4124

Table 12: The values of (En, R) for applying HHO operators

	Escaping energy	R	Operators
Sol1	0.3322	0.6482	Surrounding hard
Sol2	0.3165	0.9276	Surrounding hard
Sol3	0.7807	0.9860	Surrounding soft
Sol4	0.2077	0.7111	Surrounding Hard

This table evaluates firstly the value of escaping energy (En), while R is randomly generated in the range [0, 1]. By inspecting the obtained results of (En, R), we can conclude that Sol1, sol2, and sol4 will be transmitted to the exploitation step by applying the strategy of surrounding hard using Eq. (8). However, sol3 is updated by surrounding soft using Eq. (7).

The update values using the previous operators (soft and hard surrounding) create a new temporary population, illustrated in Tab. 13.

Table 13: The novel temporary population obtained by exploitation mode

Novel population	Word1	Word2	Word3		Word3053	Word3054
Sol1	−0.1309	−0.0800	0.9247	...	−0.1161	0.0313
Sol2	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol3	−0.0225	−0.0365	−0.3878	...	−0.0739	−1.0652
Sol4	−0.0146	−0.0891	0.8725	...	−0.0855	0.0352

The bounds of each component must be checked to respect the range between 0 and 1. This process is illustrated in Tab. 14.

Table 14: The check of lower and higher bounds

Novel population	Word1	Word2	Word3		Word3053	Word3054
Sol1	0	0	0.9247	...	0	0.0313
Sol2	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol3	0	0	0	...	0	0
Sol4	0	0	0.8725	...	0	0.0352

The second iteration verifies the value of escaping energy provided in Tab. 15. It can be seen that all values are less than 1, which required generating a random number R. This parameter will determine the adequate strategy of exploitation step as illustrated in Tab. 15.

Table 15: The control parameters of En and R

	Escaping energy	R	Operators
Sol1	0.1245	0.4056	Surrounding hard beside advanced rapid dives
Sol2	0.0678	0.6342	Surrounding hard
Sol3	0.0993	0.8122	Surrounding hard
Sol4	0.0541	0,2863	Surrounding hard beside advanced rapid dives

Based on the evaluation of fitness between X1 and each solution described in Tab. 16, we determine the novel population shown in Tab. 17.

Table 16: The fitness comparison

	fit (X(i, :)	fit(X1)
i = 1	0.2653	**0.2386**
i = 2	0,2412	**0.2386**
i = 3	0,2587	**0.2386**
i = 4	0,2448	**0,2368**

Table 17: The novel population

Novel population	Word1	Word2	Word3		Word3053	Word3054
Sol1	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol2	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol3	0.0606	0.0629	0.9984	...	0.0338	0.1645
Sol4	0.0606	0.0629	0.9984	...	0.0338	0.1645

In the third iteration, we can see that the algorithm HHO generates a higher value of escaping energy (En), as shown in Tab. 18 (all values are more significant than 1). So, the HHO applies the exploration mode defined by Eq. (4). for each solution. In this operator, the random number ($\zeta 5$) updates the solution based on two scenarios.

In the first scenario, hawks hunt randomly spread in the planned space, while, in the second scenario, the hawks hunt beside family members close to a target (Best).

Tab. 19 illustrated the final population determined by exploration mode. In this step, we should compare each solution from the previous iteration and the current population to select the best ones using the fitness metric as illustrated in Tab. 20. Also, we can conclude that the third solution determines the best solution (Rabbit) because it has a lower value of fitness. We conclude that word3 is a significant feature based on this solution, while $Word_1$, $Word_2$, $Word_3$, $Word_{3053}$, and $Word_{3054}$ are irrelevant features.

Table 18: The values of En and $\zeta 5$

	Escaping energy	$\zeta 5$	Operators
Sol1	1.3456	0.2339	Exploration mode (first scenario)
Sol2	1.5478	0.2491	Exploration mode (first scenario)
Sol3	1.2311	0.1444	Exploration mode (first scenario)
Sol4	1.9871	0.8141	Exploration mode (second scenario)

Table 19: The novel population

Novel population	$Word_1$	$Word_2$	$Word_3$		$Word_{3053}$	$Word_{3054}$	fit
Sol1	0.0546	0.0567	0.8993	…	0.0304	0.1482	0.2107
Sol2	0.0231	0.0239	0.3801	…	0.0129	0.0626	0.1914
Sol3	0.423	0.0439	0.6970	…	0.0236	0.1148	**0.1745**
Sol4	0.0606	0.0629	0.9984	…	0.0338	0.1645	0.2368

Table 20: The fitness comparison

-	fit (X(i, :)	fit(X1)
i = 1	**0.2107**	0.2386
i = 2	**0.1914**	0.2386
i = 3	**0.1745**	0.2386
i = 4	0.2458	**0.2368**

5.6 A Comparative Study with Literature Review

Three works from literature are selected. Figs. 2 and 3 show the results of counterpart optimizers (SSA, WOA, PSO, GA, DE, SSA, IWOA, and ALO) [26,27] to investigate the swarming behavior of the HHO deeply for AOM.

In terms of accuracy and selected ratio over ASA (AJGT and OCA datasets), and From Fig. 2, the HHO outperforms all optimizers except the IWOA, which exhibits identical performance in terms of the mean accuracy and selection ratio for the OCA dataset.

In conclusion, the HHO attains the best performance in terms of accuracy and selected ratio because the higher accuracy is equal to 96%. The lower selection ratio reached 40%, which means 60% of irrelevant features are eliminated. So, a good compromise is ensured between accuracy and selection ratio. Also, for the OCA dataset, HHO outperforms four optimizers, including DE, WOA, PSO, and ALO, in terms of accuracy and the same performance compared to IWO and GA. Furthermore, in terms of selection ratio HHO outperforms all optimizer except IWO which provide the same performance to HHO.

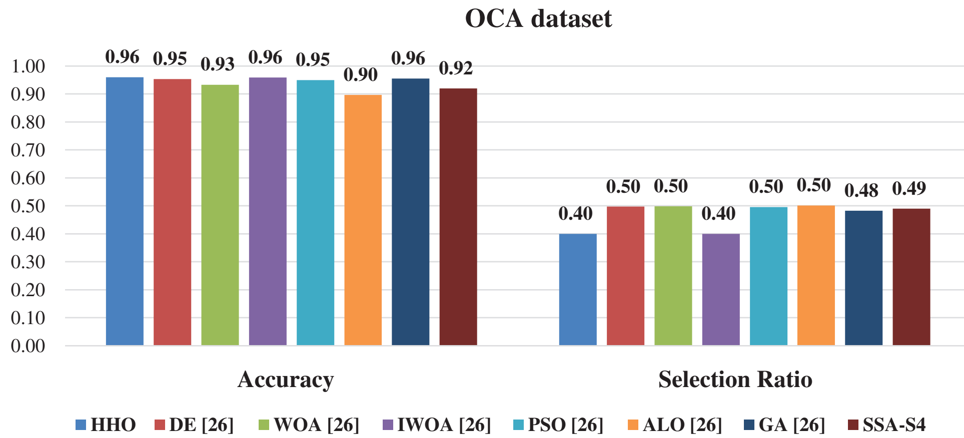


Figure 2: The comparative study of HHO with the state-of-the-art OCA dataset

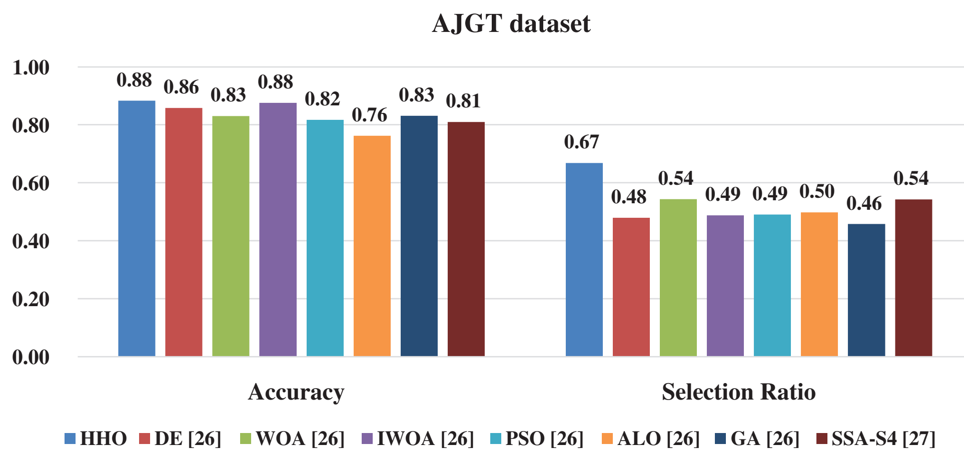


Figure 3: The comparative study of HHO with the state-of-the-art AJET dataset

From Fig. 3, HHO achieved higher performance accuracy with 88%; however, the selection ratio is ranked in the last position. This behavior can be interpreted by the informal language of Jordanian dialect, representing a real challenge in preprocessing step. Also, the application of different types of Steemer (ISRI, KHODJA) influences the selection ratio.

In addition, the initial terms extracted by the works of [26,27] are less than our study (2257 for AJGT in [26] instead of 3054 in our study), which provides a lower selection ratio compared to HHO. By analyzing Fig. 3, a slight advantage for GA in terms of selection ratio for the AJGT dataset.

6 Conclusion

The use of social networks allows people to express their opinions freely. Hence, the automatic SA has become essential, especially in e-commerce, catering, and hotel services. Several studies have been conducted for SA languages, such as English and Spanish. However, few works have devoted to the Arabic language despite their practical use and importance. This study focuses on SA of the Arab language using the HHO technique, which mimics the behavior of Harris hawks.

The main objective is to ensure a compromise between a high accuracy rate and a reduced number of significant attributes. HHO provides a good balance, especially for OCA instead of AJGT, because the second dataset contains opinions in informal dialectal language. For this reason, the number of significant attributes still higher compared to the literature review, and on the other hand, the choice of steemer (isri or KHODJA) play an essential role in the feature selection process

The studied approach shows a precise performance over both AJGT and OCA datasets in terms of accuracy, but it required more time than the other algorithms. As future work, we will consider more powerful bio-inspired algorithms in terms of performance and response time.

Funding Statement: This research was supported by Misr International University (MIU), (Grant Number. DSA28211231302952) to Diaa Salama, https://www.miuegypt.edu.eg/.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Yadav and D. K. Vishwakarma, "A comparative study on bio-inspired algorithms for sentiment analysis," *Cluster Computing*, vol. 23, no. 4, pp. 2969–2989, 2020.

[2] A. Alsayat and N. Elmitwally, "A comprehensive study for arabic sentiment analysis (challenges and applications)," *Egyptian Informatics Journal*, vol. 21, no. 1, pp. 7–12, 2020.

[3] A. H. Ombabi, W. Ouarda and A. M. Alimi, "Deep learning CNN-lSTM framework for arabic sentiment analysis using textual information shared in social networks," *Social Network Analysis and Mining*, vol. 10, no. 1, pp. 1–13, 2020.

[4] N. Omar, M. Albared, T. Al-Moslmi and A. Al-Shabi, "A comparative study of feature selection and machine learning algorithms for Arabic sentiment classification," in *Asia Information Retrieval Sym.*, Kuching, Malaysia, Springer, pp. 429–443, 2014.

[5] N. Neggaz, A. A. Ewees, M. Abd Elaziz and M. Mafarja, "Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection," *Expert Systems with Applications*, vol. 145, pp. 113103, 2020.

[6] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[7] W. Deng, J. Xu, Y. Song and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization," *Applied Soft Computing*, vol. 100, no. 3, pp. 106724, 2021.

[8] Y. Bi, B. Xue and M. Zhang, "Genetic programming with a New representation to automatically learn features and evolve ensembles for image classification," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1769–1783, 2021.

[9] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[10] H. Faris, I. Aljarah, M. A. Al-Betar and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Computing and Applications*, vol. 30, no. 2, pp. 413–435, 2018.

[11] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[12] W. Guo, T. Liu, F. Dai and P. Xu, "An improved whale optimization algorithm for feature selection," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 337–354, 2020.

[13] A. H. Kashan, "League championship algorithm (lca): An algorithm for global optimization inspired by sport championships," *Applied Soft Computing*, vol. 16, pp. 171–200, 2014.

[14] R. Moghdani and K. Salimifard, "Volleyball premier league algorithm," *Applied Soft Computing*, vol. 64, pp. 161–185, 2018.

[15] B. Alatas, "Sports inspired computational intelligence algorithms for global optimization," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1579–1627, 2019.
[16] S. Mirjalili, "Sca: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
[17] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.
[18] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. AlAtabany and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.
[19] F. Bi, X. Fu, W. Chen, W. Fang, X. Miao *et al.,* "Fire detection method based on improved fruit fly optimization-based SVM," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 199–216, 2020.
[20] Z. Zhou, J. Qin, X. Xiang, Y. Tan, Q. Liu *et al.,* "News text topic clustering optimized method based on TF-iDF algorithm on spark," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 217–231, 2020.
[21] N. Neggaz, E. H. Houssein and K. Hussain, "An efficient henry gas solubility optimization for feature selection," *Expert Systems with Applications*, vol. 152, pp. 113364, 2020.
[22] K. Hussain, N. Neggaz, W. Zhu and E. H. Houssein, "An efficient hybrid sine-cosine harris hawks optimization for low and high-dimensional feature selection," *Expert Systems with Applications*, vol. 176, pp. 114778, 2021.
[23] A. Ghallab, A. Mohsen and Y. Ali, "Arabic sentiment analysis: A systematic literature review," *Applied Computational Intelligence and Soft Computing*, vol. 2020, no. 1, pp. 1–21, 2020.
[24] M. Al-Ayyoub, A. Khamaiseh, Y. Jararweh and N. Al-Kabi, "A comprehensive survey of arabic sentiment analysis," *Information Processing & Management*, vol. 56, no. 2, pp. 320–342, 2019.
[25] L. Canales, W. Daelemans, E. Boldrini and P. Martınez-Barco, "EmoLabel: Semi-automatic methodology for emotion annotation of social media text," *IEE Transaction on Affective Computing*, vol. 14, no. 8, pp. 1–14, 2015.
[26] M. Tubishat, M. A. Abushariah, N. Idris and I. Aljarah, "Improved whale optimization algorithm for feature selection in arabic sentiment analysis," *Applied Intelligence*, vol. 49, no. 5, pp. 1688–1707, 2019.
[27] A. Alzaqebah, B. Smadi and B. H. Hammo, "Arabic sentiment analysis based on salp swarm algorithm with s-shaped transfer functions," in *Proc. ICAIS*, Irbid, Jordan, pp. 179–184, 2020.
[28] N. Neggaz, R. Tlemsani and F. A. Hashim, "Arabic sentiment analysis using grey wolf optimizer based on feature selection," in *Proc Int. Conf. on Research in Applied Mathematics and Computer Science*, Casablanca, Morocco, pp. 1–8, 2020.
[29] S. L. Marie-Sainte, N. Alalyani, S. Alotaibi, S. Ghouzali and I. Abunadi, "Arabic natural language processing and machine learning-based systems," *IEEE Access*, vol. 7, pp. 7011–7020, 2018.
[30] J. Ramos, "Using TF-iDF to determine word relevance in document queries," in *Proc. of the 1st Instructional Conf. on Machine Learning*, vol. 242, no. 1, pp. 29–48, 2003.
[31] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja *et al.,* "Harris hawks optimization: Algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
[32] E. Emary, H. M. Zawbaa and M. Sharawi, "Impact of lèvy flight on modern meta-heuristic optimizers," *Applied Soft Computing*, vol. 75, pp. 775–789, 2019.
[33] L. Dey, S. Chakraborty, A. Biswas, B. Bose and S. Tiwari, "Sentiment analysis of review datasets using naive Bayes and k-nN classifier," arXiv preprint arXiv: 1610.09982, vol. 8, no. 4, pp. 54–62, 2016.
[34] A. Dahou, S. Xiong, J. Zhou and M. Abdelaziz, "Multi-channel embedding convolutional neural network model for arabic sentiment classification," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 4, pp. 1–23, 2019.
[35] M. Rushdi-Saleh, M. T. Martín-Valdivia, L. A. UreñaLópez and J. M. Perea-Ortega, "OCA: Opinion corpus for Arabic," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 10, pp. 2045–2054, 2011.