Applied Soft Computing Journal 96 (2020) 106651

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering

R. Purushothaman [a,*], S.P. Rajagopalan [a], Gopinath Dhandapani [b]

[a] Department of Computer Science and Engineering, GKM College of Engineering and Technology, Chennai, India
[b] Department of Mechanical Engineering, GKM College of Engineering and Technology, Chennai, India

ARTICLE INFO

Article history:
Received 17 September 2019
Received in revised form 3 July 2020
Accepted 17 August 2020
Available online 20 August 2020

Keywords:
Feature selection
Preprocessing steps
GWO
GOA
FCM
And text clustering

ABSTRACT

Text analysis in the field of text mining requires complex techniques for handling several text documents. Text clustering is among the most effective tactics in the field of text mining, machine recruitment and pattern recognition. Computers can start organizing a corpus document in certain organizational structure of conceptual clusters using reasonable text-clustering method. Informative and un-informational functionalities of the text documents contain noisy, inconsequential and superfluous features. The main method of finding a new subset of informative feats for each document is the unsupervised selection of text features. The functional selection technique has two aims: (1) maximize text clustering algorithm reliability, (2) minimize the number of uninformative traits. The proposed technique is that it produces a mature convergence rate and requires minimal computational time and is trapped in local minima in a low dimensional space. The text data is fed as the input and pre-processing steps are performed in the document. Next, the text feature selection is processed by selecting the local optima from the text document and then selecting the best global optima from local optimum using hybrid GWO–GOA. Furthermore, the selected optima are clustered using the Fuzzy c-means (FCM) clustering algorithm. This algorithm improves the reliability and minimizes the computational time cost. Eight datasets are used in the proposed algorithm and the performance is envisaged efficaciously. The evaluation metrics used for performing text feature selection and text clustering are accuracy, precision, recall, F-measure, sensitivity, specificity and show better quality when comparing with various other algorithms. When comparing with GWO, GOA and the proposed hybrid GWO–GOA algorithm, the proposed methodology reveals 87.6% of efficiency.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

The websites and other computerized implementations offer a large amount of text information. In terms of making efficient use of details, the text information should be divided up due to the growing use of those applications. Transcripts are categorized as per their provenance and a considerable volume of information is ended up saving as text on these digital documents [1]. Text clustering strategies are extensively used for categorizing text information. Text clusters are used with implementations such as text mining, retrieval of text information and text classification. In all areas of text mining and in specific in the field of text document clustering the substance is a hurdle for text analysis techniques [2].

For the text clustering process, text feature sampling is essential. The text information contains informative and non-informative functionalities. These un-informing characteristics hoodwinked and minimize its reliability by the classification algorithms and strategies. The main purpose of the function selection process is to remove un-informing functions and to select information. Weight-based techniques for selecting functions are used in subsequent years. These text characteristic categorizations influence the efficiency and performance of clustering procedures in which uninformative functionality is inappropriate, irrelevant and noisy [3]. Thus it is important to establish an effective approach for identifying features. Several strategies are used for the availability of features, including embedded methods, wrapping methods and filter methods. The wrapper method examines every subset of components from the training model and realizes the best subset of functionality. To perform the classification or clustering tasks and to select features simply efficiently, the filter procedure first chooses the standard features of all the features using certain criteria of analysis. To find the best feature subsection, the wrapper method uses a training model directly to determine each feature subset as the outcomes of the selection [4].

* Corresponding author.
E-mail address: rpurushothaman1006@gmail.com (R. Purushothaman).

https://doi.org/10.1016/j.asoc.2020.106651
1568-4946/© 2020 Elsevier B.V. All rights reserved.

Nomenclature

$tw_{u,v}$	term weight v in the text document u
α	Alpha
δ	Delta
D	distance between the prey and the wolf
X_P	beheads position vector
P_k	Dictates the position of the kth grasshopper
L_k	gravity force
rn_2, rn_3	random numbers in [0, 1]
R	intensity of attraction
q, L	components
e_f	unity vector towards the center of earth
e_u	unity vector in wind direction
Sf_i	number of text feature selected from document i
P_r	Parameter
$xf_{u,v}$	frequency of term v in the text document u
β	Beta
ω	Omega
X	wolf's vector of position
A, C	random vectors
Q_k	is the social interaction
B_k	wind advection
d_{kl}	distance between kth and the lth grasshopper
t	length scale
f	gravitational constant
v	constant drift
M	number of grasshoppers
$\overline{m_l}$	mean value of the vector i
Q_u	control parameter

Method of feature selection is classified as half-supervised, unmonitored and controlled learning. The monitored and semi-monitored methods change based on label information to mentor the analysis of features by coding discrimination data on labels. In many practical applications, indeed, data are frequently collected at large scales without labels. These methods are quantitatively complex and expensive to implement due to the requirement for label information. Because of this, unattended learning techniques are becoming more prominent and entice most of the validation of researchers and make many machine teaching tasks like classification, clustering, recuperation and evaluation much easier [5].

In this work, a recent innovation in the area of text mining is the vector-space model (VSM), which depicts the text features of each document as a vector of terms of weight. Each term weight is portrayed in this model as a single-dimensional space. The clustering methodology reliability is thus impacted by dimensional space and non-informational peculiarities. The inconveniences are surmounted by Gray Wolf Optimization (GWO) [6] hybrid with Grasshopper Optimization Algorithm (GOA) [7]. GWO is inspired by gray wolves to attack praise for hunting purpose. The GWO is based on behavior of the gray wolf. The better optimum solutions with lower computation burden can be found in GWO compared to the existing stochastic search techniques. GOA is inspired by the biological behavior shown in swarms of grasshoppers. The special adaptive mechanism in this algorithm smoothly balances exploration and exploitation. These characteristics make the GOA algorithm potentially able to cope with the difficulties of a multi-objective search space and outperform other techniques

First, we use a hybrid GWO–GOA to select the text features. GWO selects local characteristics of text documents and GOA selects the best global qualities from local functions. The hybrid GWO–GOA minimizes un-informational functions for enhancing text classification algorithm quality. We use the Fuzzy c-means (FCM) algorithm [8] to cluster the document after the selection of global text features. In past related works explained about different methods to text feature selection and clustering, but all these related works do not sufficiently take into account optimization problems on optimum using hybrid GWO–GOA based algorithm development.

The remaining article is structured as follows: Section 2 describes the field of text-feature selection for text clustering provides various works about the use of meta-heuristic algorithms and dimension minimization. Section 3 describes the proposed methodology, illustrates the steps of the feature selection problem and illustrates the proposed Hybrid GWO–GOA.

Section 4 presents the experimental results. Section 5 defines the conclusion and future work.

2. Related work

The comprehensive textures selection and the vibrant dimension deterioration to text clustering were provided by Abualigah et al. [9]. Text clustering strategies are extensively used for categorizing text information. Text clusters are used by implementations such as text mining, retrieval of text information and text classification. For the text clustering process, text feature sampling is essential. The text information contains informative and non-informative functionalities. These un-informing characteristics hoodwinked and minimize its reliability by the classification algorithms and strategies. The text characteristics have been chosen using the harmony search algorithm, optimization of swarm particles and genetic algorithm. For the implication of function selection techniques, a weighting scheme called length–weight was incorporated. The vibrant dimensional mitigation technique lowered the number of features used for clustering. They clustered the text document using the k-means clustering method. In contrast to the existing technologies, the implementation of the proposed method is strengthened.

Hybrid feature selection approaches to text categorization has been established by Abdullah Saeed Ghareb, [10] using Genetic Algorithm. This approach uses a hybrid search technique that combines the advantages of filter feature selection methods with an enhanced GA (EGA) in a wrapper approach to handle the high dimensionality of the feature space and improve categorization performance simultaneously.

Ibrahim Aljarah, [11] proposed simultaneous feature selection and support vector Machine optimization using the Grasshopper Optimization algorithm. GOA is utilized for performing feature selection and optimizing the parameters of SVM simultaneously. The goal of the model is to maximize the classification accuracy of SVM with the smallest possible number of features.

For short text clustering, Xu et al. [12] have optimized a self-teaching neural network. Therefore, the raw characteristics were transferred into versatile binary codes with an uncontrolled dimension eliminating technique. The neural network frequency domain discovered deep feature depictions. Then, the methodology k-means were used to cluster the document.

Majdi Mafarja, [13] has proposed a method evolutionary population dynamics and Grasshopper Optimization approaches to feature selection problems. This paper presents an efficient GOA-based optimizer with EPD and selection operators to improve the efficacy of the basic GOA in dealing with FS tasks.

D. Thiyagarajan and N. Shanthi, [14] conveyed in text classification a multi-objective heuristic text selection approach. For function selection the algorithm Modified artificial fish swarm was used. For text classification, the SVM classifier is used. In comparison to the existing methods, the proposed method improved accuracy.

The artificial bee colony algorithm of the chaotic gradation for clustering was implemented by Bharti and Singh, [15]. The heuristic of the artificial colony of bees chooses the best appropriate cluster hubs. The method implemented was superior to the current methods. Arun Kumar Sangaiah et al. [16] have established a clustering classifier for Arabic text documents. A certain work exploited a non-controlled a technological method like k-means, accumulative k-means and semi-controlled strategies such as threshold and k-means. Compared to conventional strategies, the precision of the proposed method is huge.

Majdi et al. [17] has presented a method of an efficient optimizer depended on the simultaneous utilization of the GOA, selection operators, and Evolutionary Population Dynamics (EPD) is presented in the form of four various methods to mitigate the immature convergence and stagnation drawbacks of the conventional GOA. In the first two stages, one of the top three agents and a randomly generated one are chosen to reposition a solution from the worst half of the population. In the third and fourth stage, to give a chance to the low fitness solutions in reforming the population, Roulette Wheel Selection (RWS) and Tournament Selection (TS) are used to select the guiding agent from the first half.

Ibrahim Aljarah et al. [18] suggested a hybrid tabu search (TS) and GWO based text clustering method. With the help of TS algorithm the performance of GWO has been improved. Yet this hybrid method has drawback like high running time. H Rashaideh et al. [19] introduced a GWO based text clustering. In this text clustering algorithm to optimize the distance between clusters and documents authors considered the average distance of documents to the cluster centroid (ADDC) as an objective function. However the accuracy and the robustness of the suggested method were not achieved the desired level. Also the computational complexity of work is higher and it is represented by $O(t_1 d n_2)$, where t_1 indicates the number of iterations, d is the number of variables, and n shows the number of solutions. Szymon Łukasik et al. [20] investigated about using GOA in text clustering. Calinski–Harabasz index is used for clustering validation of formed solutions. The computational complexity of work is $O(t \times S \times N_2)$, where t is the current iteration, S is dimension, and N represents total number of grasshoppers.

2.1. Motivation of the work

GWO in fact is a meta-heuristic algorithm, inspired by the leadership behavior and unique mechanism of hunting of gray wolves. This population-based meta-heuristic has the ability to avoid local optima stagnation to some extent. It also has good convergence ability towards the optima. In general, GWO advances itself strongly to exploitation. However, it cannot always implement exploration well. Thus, in some cases, GWO cannot always deal with the problem successfully and fails to find the global optimal solution [21,22]. GOA is a recently proposed meta-heuristic algorithm which mimics the intelligent behavior swarms of grasshoppers [23–25]. The local search strategy of the GOA is not very much efficient. Both GOA and GWO are not always guaranteed to find a good solution because they depend on each other in this way and the time complexity of this method is not good compared to the proposed method. Considering the strengths of GWO and GOA, these two algorithms are ideal for hybridization [26]. Therefore, in this study, a hybrid algorithm comprising GWO and GOA, termed GWO–GOA is proposed which combines two algorithms in order to achieve a more suitable trade-off between diversification and intensification, and offer significantly better results than the conventional GWO and GOA in terms of solution accuracy and convergence speed.

3. Proposed methodology

A sequence of substantial text documents is reconfigured into a relatively small group of lattices premised on their inherent characteristics in the text clustering. A benchmark pre-processing step is used to pre-process text documents before implementing clustering techniques. In order to mitigate computing time and focus on improving text clustering defensive effort, the text preprocessing steps is being used to designate optimum thought-provoking aspects. The proposed method, therefore, identifies a new subset of information-oriented text-features with a few spaces. The main advantage of the GWO algorithm over most of the well-known meta-heuristic algorithms is that the GWO algorithm operation requires no specific input parameters. Additionally, it is straightforward and free from computational complexity. Further, its advantages include — ease of transformation of such concept of the programming language and ease of comprehensibility [27]. The strength of GOA lies in the ability to avoid local optima easily when dealing with complex, high dimensional and multimodal problems.

In GWO–GOA, random error is a deterministic, random-like method found in non-linear, dynamical system, which is non-period, non-converging and bounded [28]. Mathematically, error is randomness of a simple deterministic dynamical system and chaotic system may be considered as sources of randomness. The random error of the proposed algorithm is reduced by running the algorithm at several times. The framework of the proposed method is illustrated in Fig. 1.

3.1. Pre-processing of text documents

For converting texts into numerical form, pre-processing measures are utilized. The foregoing is a detailed breakdown of the benchmark pre-processing [10,29] phases:

3.1.1. Tokenization

Tokenization signifies the hierarchy of a text stream into words or terms and the elimination of a blank series, wherein each expression or symbol is chosen from the first to the final the character referred to as a token [30].

3.1.2. Removal of stop words

The removal process deletes certain most common phrases, like, if, this, that, even and several other obscure words which take little variance, and also high intensity and brief responsive words, usually form a part of the document and impact the rising numbers of features, thus reducing the efficiency of the text document group. A list of certain words comprising 571 words can be encountered in (http://www.unine.ch/info/clef/).

3.1.3. Stemming

By eliminating the prefixes and suffixes for each word, Stemming infuses the conjugated verbs pertinent forms of certain words with almost a similar root. The effective, efficacious and effectual, for instance, have the same root, effect, which is a feature. We use the Porter stemmer in this paper; it is an effective method of stemming [31].

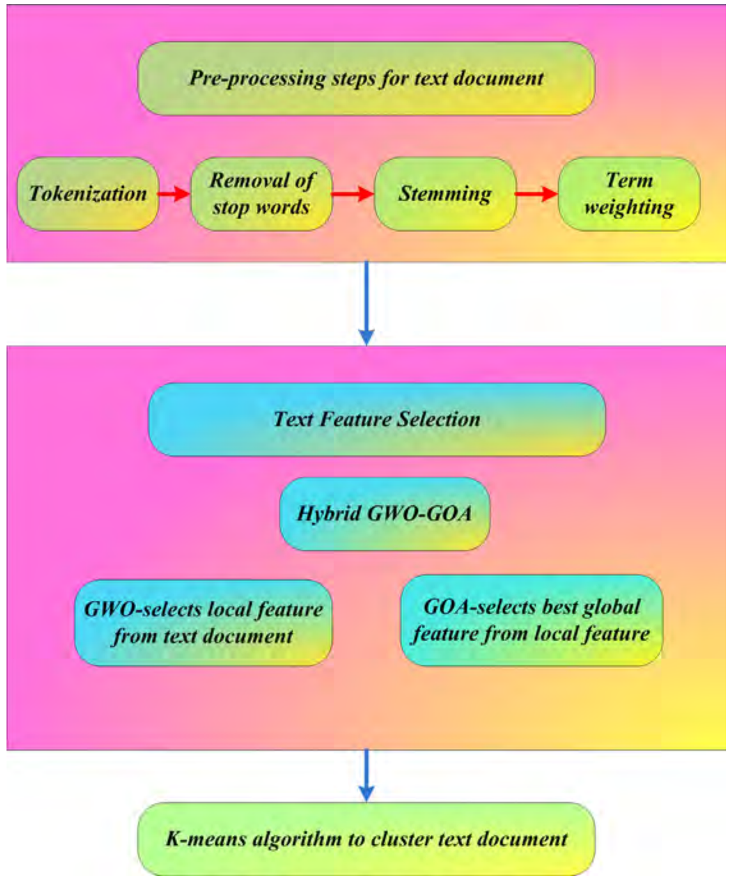

Fig. 1. Overall Structure of the proposed model.

3.1.4. Term weighting

To convert the textual form of data into a numerical form is term weighting [32]. For measuring the term weighting for document representation within the field of text mining the term frequency–inverse document frequency (TFIDF) is used. The term weighs vector of every document presented as in Eq. (1),

$$T_i = (tw_u, 1, tw_u, 2, \dots, tw_{u,v}, \dots, tw_{u,r}) \tag{1}$$

For each term, based on the current magnitude of each manuscript and other determinants, the term weighting is entrusted. We infer that perhaps this term is beneficial to make a distinction among supporting documents if the magnitude is huge and the similar term is used in many relevant documents. The term weighting is formulated as in Eq. (2),

$$tw_{uv} = xf(u,v) * yf(u,v) = xf(u,v) * \log(n/zf(v)) \tag{2}$$

where the term weight v in the text document u denoted in $tw_{u,v}$ and the frequency of term v in the text document u denoted in $xf_{u,v}$. In Eq. (3) to enhance the term with limited amplitude in document by $yf_{u,v} = \log(n/zf(v))$, where a number of documents in the dataset and $zf(v)$is the number of documents that obtains the term u. By using the vector space model format the formulated expression is as follows,

$$VSM = \begin{bmatrix} tw_{1,1} & \cdots & tw_{1,(r-1)} & tw_{1,r} \\ \vdots & \ddots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots \\ tw_{(n-1),1} & \cdots & \cdots & tw_{(n-1),r} \\ tw_{n,1} & \cdots & tw_{n,(r-1)} & tw_{n,r} \end{bmatrix} \tag{3}$$

3.2. Gray wolf optimization algorithm

In 2014, Mirjalili et al. [6] suggested the Gray Wolf Optimizer (GWO). It is a novice in meta-heuristic algorithms that are influenced by evolution. The leadership and hunting features of the gray wolves are emulated. Gray wolves belong to the family of Canidae and are very rigid class hierarchies. In a pack of wolves, they tend to favor hunting of prey. In the classic GWO for its optimal virtual environment, which covers the four tiers of Wolves' hierarchy which are alpha (α), beta (β), delta (δ) and omega (ω) commonly do have some inferences. (α) Wolf is the leader of the wolf pack at the highest level. It could be a wolf of a male or a female kind. Hunting, discipline, sleeps, and time to wake is vicariously liable to make all sorts of decisions. Secondly, (β) are the subjugated wolves and assists the (α) leader in decision making or any other pursuits. As the second greatest wolf in the cluster, the β wolf is perhaps most likely to become a (α) leader. The third degree of the gray wolves, for instance (δ) wolves, annihilates the wolves in the front, and the last degree is called the (ω) wolves that ensure the perceived safety and the competence of the wolf packs [33,34]. GWO is designed quantitatively into the proceeding four processes:

3.2.1. Hierarchy rigidly structured

The GWO algorithm is focused on the dominance hierarchy of the wolves quantitatively designed. The best approach (α) is the high level of the intellectual hierarchy [35]. In the same way, (β) and (δ) are both perceived as the second and third highest quality viable alternative. The other solution which pursues (α), (β) and (δ) wolves is surmised to be the remaining candidates (ω).

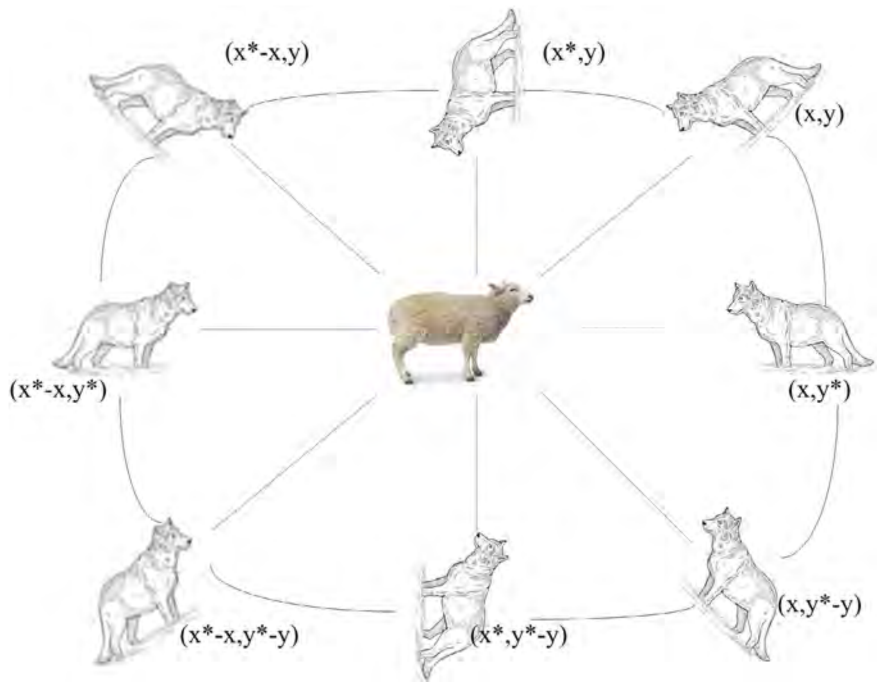

Fig. 2. Mechanism of GWO.

3.2.2. Prey encircling

The gray wolves have an intrinsic functionality of hunting around the prey. Eqs. (4) and (5) depicts the quantum mechanics designed at GWO to surround the wolves [36].

$$D = |C.X_P(t) - X(t)| \tag{4}$$

$$X(t+1) = X_P(t) - A.D \tag{5}$$

The distance between the prey and the wolf here is D. X Is the wolf's vector of position and X_P implies beheads position vector at the iteration t. A and C are random vectors that are determined in Eqs. (4) and (5),

$$A = 2a.r_1 - a \tag{6}$$

$$C = 2.r_2 \tag{7}$$

The random vectors in the range of [0, 1] here are, r_1 and r_2. They make wolves to reach among the prey and the wolf at any point.

Vector a works to regulate the GWO algorithm phenomenon and is used as a basis for A computations. The vector a aspect standards decline dynamically among 2 and 0 overtime.

3.2.3. Prey hunting

The gray wolves can feasibly circle it, because they can track the movement of the prey. The (α) wolf tends to lead nearly the entire process of hunting. All gray wolves are chased by (α), (β) and (δ) wolves. They will also keep updating their positions to the optimal position of the wolves (α), (β) and (δ). It is articulated in Eqs. (8)–(10) in statistical terms.

$$D_\alpha = |C_1.X_\alpha - X|\,, D_\beta = \left|C_2.X_\beta - X\right|, D_\delta = |C_3.X_\delta - X| \tag{8}$$

$$X_1 = X_\alpha - A_1.D_\alpha, X_2 = X_\beta - A_2.D_\beta, X_3 = X_\delta - A_3.D_\delta \tag{9}$$

Eq. (7) can be used to measure the upgraded status of the gray wolf.

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{10}$$

3.2.4. Prey searching and attacking

Gray wolves will only attack the prey when they are no longer moving. It is designed as per the vector in Eq. (3), mathematically. A is a random vector, which is within the range $[-a, a]$ and which has been reduced from 2 to 0 by an Eq. (11),

$$a = 2 - (2 * t/M_i) \tag{11}$$

Thus, if $|A| < 1$, the wolf is compelled to annihilate the prey in a way that $|A| > 1$ is directed at, then the wolf is distinguished from the prey and looks for fitter prey. The search of gray wolves takes place in the (α), (β) and (δ) wolves, depending on the position of the wolves. Indeed the values of A and C vectors impinge on exploitation and exploration. We can use A random values to differentiate the wolf from the prey. C Vector random values lie within [0, 2] which are a critical factor in preventing local optimal stagnation. It increases the random weight of the capabilities so that the distance from the battlements to the wolf is harder for gray wolves. $C > 1$, means that C exemplifies the impact of the prey and that if $C < 1$, the effect of C is stochastically de-emphasized (shown in Fig. 2).

The main focus in the entire ordeal is on calibrating A and C vectors. Almost all parameters stress or highlight exploitation or exploration. Finally, the GWO algorithm is suspended when the last benchmark is met and the result is the best location for the alpha wolf [33,34].

3.3. Grasshopper optimization algorithm

Grasshopper, insects whose harm to soil fertility and cultivation is regarded a pest. Grasshoppers' life span cycle is shown in Fig. 3. While grasshoppers are generally personally witnessed singly in nature, each of the biggest swarms of all organisms is joined with them [37]. The swarm size can be the continental extent and a farmer's disaster waiting to happen. The notable feature of the grasshopper swarm is that those nymph and adult lives find swarming actions. Billions of nymph grasshoppers, like spinning cylinders, bounce and transfer. They eat nearly all the grasslands on their route. During this behavior, they create a

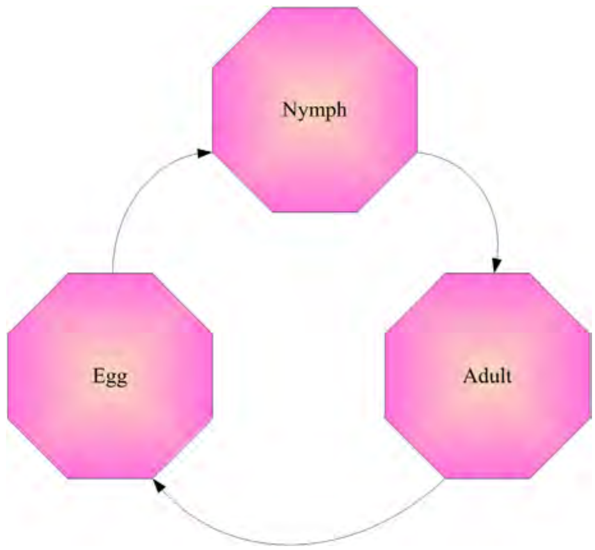

Fig. 3. Life span cycle of a grasshopper.

swarm throughout the air once they become adults. That is how huge intervals of grasshoppers relocate [38].

Slow-motion and steady progress of the grasshoppers are the defining quality of the swarm throughout the larval phase. In comparison, the important component of swarm in adult lives is long-range and unexpected motion. Source of energy searching is also another essential feature of grasshoppers swarming. The evaluation process is philosophically divided into two impulses by nature-inspired algorithms: discovery and exploitation. In discovery, the check attorneys are expected to move briefly, however, during exploitation they tend to drive regionally. Those little two operations, and also the check for targets, are practiced by natural grasshoppers.

The numerical simulation used to accurately measure grasshoppers' swarming actions is shown as follows,

$$P_k = Q_k + L_k + B_k \tag{12}$$

where P_k Dictates the position of the kth grasshopper, Q_k is the social interaction, L_k is the gravity force and B_k defines the wind advection.

To provide random behavior, the equation can be written as $P_k = rn_1Q_k + rn_2L_k + rn_3B_k$ where rn_1, rn_2 and rn_3 are random numbers in [0, 1].

$$Q_k = \sum_{\substack{l=1 \\ l\neq 1}}^{m} q(d_{kl})d_{kl} \tag{13}$$

where d_{kl} are the distance between kth and the lth grasshopper.

Computed as $d_{kl} = |x_1 - x_2|$, q is a function to state the strength of social forces, as shown in Eq. (11), and $d_{kl} = (x_1 - x_2)/d_{kl}$ is a unit vector from the kth grasshopper to the lth grasshopper.

A theoretical framework of grasshopper interaction with the comfort zone is shown in Fig. 4. It is worth noting that this social interaction was in a simplified form, the motive of certain earlier models of mussel swarming. While the Q function can divide the space between two grasshoppers into a repellent area, comfort zone and area of appeal, this function returns values near zero with distances above 10. This function cannot indeed be used among grasshoppers with greater distances. We have sketched the grasshoppers' distance from 1 to 4 to overcome the problem.

The q function, which defines the social forces, is calculated as follows,

$$q(rn) = \mathrm{Re}^{\frac{-rn}{t}} - e^{-rn} \tag{14}$$

where R denotes the intensity of attraction and t is the length scale. The shape of the function q have L component and is computed as,

$$L_k = -fe_f \tag{15}$$

where f is the gravitational constant and e_f shows a unity vector towards the center of earth.

The B component is computed as follows,

$$B_K = ve_u \tag{16}$$

where v is a constant drift and e_u is a unity vector in wind direction. Nymph grasshoppers consist of no wings, so the movements of them are correlated with wind direction. Putting q, L and B in Eq. (12), the equation can be written as,

$$P_k = \sum_{\substack{l=1 \\ l\neq 1}}^{m} q(|x_l - x_k|)((x_l - x_k)/d_{kl}) - fe_f + ve_u \tag{17}$$

where $q(rn) = \mathrm{Re}^{\frac{-rn}{t}} - e^{-rn}$ and M is the number of grasshoppers.

The model is used in free space for the swarm Eq. (13) is utilized and can simulate the interaction in a swarm between grasshoppers.

However, there is no destination in an actual search space since we do not understand exactly in which the global maximum, the main goal, is. Furthermore, for each optimization step, we need to pick targets for grasshoppers. To GOA, the fit grasshopper (still with the best competitive value) is suspected to be the destination throughout the optimization. It will also help GOA to do the same in every installment the most successful goal in the search space and this will actually-want grasshoppers to shift towards it. These are accomplished with the promise that a better and clearer goal will be found as the finest approximation in the search space for the true global optimum [39].

3.4. Hybridizing GWO–GOA

Grasshopper Optimizing Algorithm (GOA) with hybridizing Gray Wolf Optimization (GWO) to surmount premature convergence, extensive calculation time and is trapped in local minimums in large size space. First, we use hybrid GWO–GOA to select text traits. GWO selects local traits from the texts and GOA selects the best global traits from the local trait. In order to enhance the efficiency of the text clustering algorithm, the Hybrid GWO–GOA minimizes its uninformative features. The first step of any meta-heuristic GWO–GOA optimization algorithm, which impacts the convergence heuristic and solution quality, is to the initialize population. Because the real issue is not resolved, the discrete dynamic allocation will be the most important strategy used to produce the preliminary solutions. The population with a random position was generated by the meta-heuristic algorithm (i.e. the initial solution is a random 0 or 1 which means if there is a position = 0, this feature will not be selected as an information feature and if there is any position = 1, it is an informative feature. A certain heuristic improves the prey to provide the best solution possible (i.e. an optimum subset). The solution is evaluated according to the Fitness function and it seems that every predicament is an element of the space (Fig. 5).

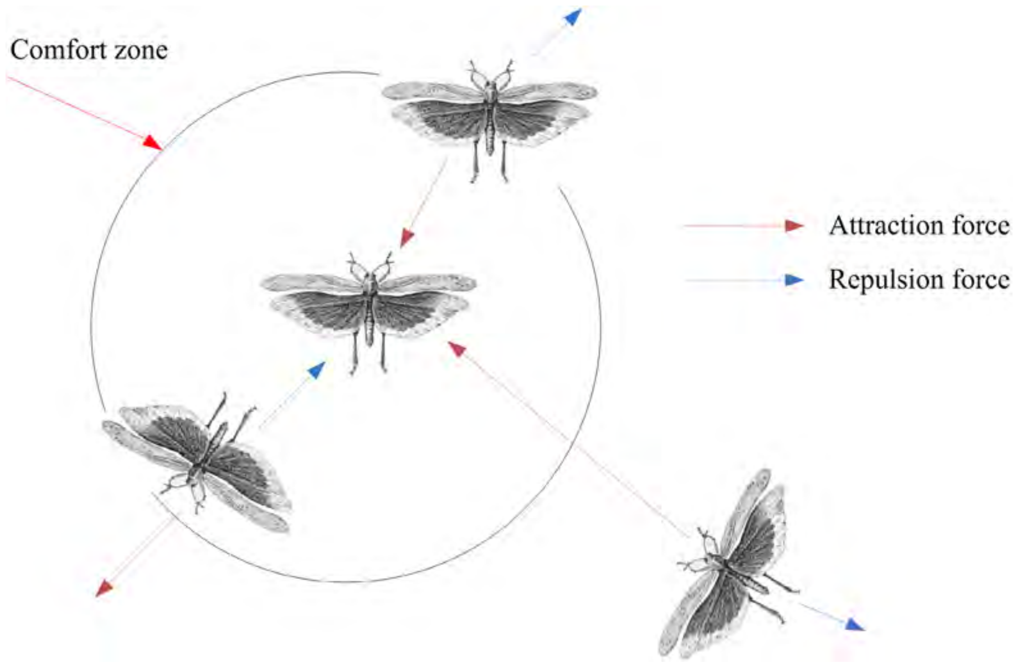

Fig. 4. Patterns among individuals in a swarm of grasshopper.

Updating of position. Based on its current position, target status, the locations of all other grasshoppers and a grasshopper's next position shall be defined. Probably clarify the location of the present grasshopper regarding other grasshoppers is considered by the first component of the Eq. (15) of practical terms, the status of all grass-hopper has been perceived to define where search mediums are located around the target. Premised on the use of the current location, global best, and the position of all other search agents, GOA upgrades the search medium position. All search mediums are obliged by GOA to participate in interpreting each search medium's next position.

$$R_i^c = b\left(\sum_{\substack{j=1\\ j\neq 1}}^{N} b\frac{u_c - l_c}{2} m(\left|r_j^c - r_i^c\right|)\frac{r_j - r_i}{c_{ij}}\right) + S_c \tag{18}$$

In Eq. (15), the first b from the left has the same weight as inertial (w) in GOA. It minimizes grasshopper's movements throughout the goal. Exploration and exploitation of the whole prey balance the parameter. The second b decreases the attraction zone, comfort zone, and the repulsion zone between grasshoppers. By using the element $b\frac{u_c-l_c}{2}m(\left|r_j - r_i\right|)$, $b\frac{u_c-l_c}{2}$ linearly diminishes the space that the grasshoppers should explore and exploit. The element $m(\left|r_j - r_i\right|)$ implies if a grasshopper should be repelled from (exploration) or attracted to (exploitation) the target. The b internal contributes to a decrease of the repulsion/attraction between grasshoppers proportionally to the number of iterations, and the b external decreases the search covers around the target with the rise of the iteration count.

In GWO–GOA, a search medium upgrades its position by using (α) and (β) as shown in Eq. (19),

$$R(t+1) = R + f1 * rand * ((R_1 - R) + (R_2 - R))/2 \tag{19}$$

A further mural for the upgrade of the alpha and beta direction is not upgraded by all people of the population, but by alpha only in the GWO–GOA proposed to maintain their workforce homogeneity. The proposed algorithm acts as a declining strategy to avoid the local optimum.

$$R(t+1) = R + f1 * rand * (R_1 - R) \tag{20}$$

The pseudo code of the proposed GWO–GOA algorithm is portrayed in Algorithm 1.

ALGORITHM: 1 Hybridizing GWO-GOA

Input: initialize the population and GWO-GOA parameters;

Output: Optimal solution (subset of texture features)

Initialize the population of P_k (k = 1,2,3,……..,n) ;

Initialize the parameters l, M and N ;

Calculate the fitness of every search medium (wolf);

P_α = Global best feature search medium;

P_β = Local best feature search medium;

While ($r < M_{itera}$)

  For single search medium

  Normalize Grasshopper's distance;

  Update Current position of search medium;

If (current search medium>boundary)

Retrieve current search medium;

End For

Update l value by eqn. (13);

Update parameters M and N ;

Calculate all search mediums fitness;

Update P_α and P_β ;

$r = r + 1$;

End while

Return a new subset of text features

Control parameters. In balancing the exploration and exploitation of a search agent, the M parameter performs a rather prominent role. The significant control of the search phase direction is crucially dependent on l. Larger value of l performs exploration phase and smaller value of l performs exploitation phase. The appropriate selection of an exploration and exploitation balance may be sturdy and can give rise to enhanced quality. The generation of control parameter values l in the optimization phase of the strategy (as shown in Eq. (21)),

$$l = 2 - (\cos(rand()) * 1/M_{itera}) \tag{21}$$

3.4.1. Fitness function

The fitness function is used to measure candidate solutions after the solutions are initialized. As the fitness function for the unpredicted text selection problem, we applied to mean absolute difference (MAD). It is a good measure to assess the value of the documents' functions in accordance with weight (i.e., TFIDF and VSM). MAD is employed by calculating the differences between the mean values of the text features to assign relevance score for each feature in Eq. (22),

$$MAD(Sf_i) = \frac{1}{Sf_i}\sum_{k=1}^{j}\left|m_{i,k} - \overline{m_l}\right| \tag{22}$$

Where,

$$\overline{m_l} = \frac{1}{Sf_i}\sum_{k=1}^{j} m_{i,k}$$

where Sf_i is the number of text feature selected from document i, $\overline{m_l}$ is the mean value of the vector i, j is the weighting value of feature k in document i, j is the number of text features in the original text dataset [10,21].

3.4.2. Crossover and mutation rate in GWO-GOA

The heuristic implemented also utilizes the mutated algorithm to focus on improving the rates of convergence by crossover and mutation operations.

Crossover rate. The extensive cross-over operation used in the GOA is adapted to enhance GWO exploration and exploiting capacities. It helps the GWO–GOA avoid untimely convergence. The crossover possibility P_r is the parameter used when a consistently disseminated random value is emitted among [0, 1] for the calibration of a crossover operator. The crossover operation is as defies,

$$z_{k,l} = \begin{Bmatrix} z_{v,l} & \text{if } rand < P_r \\ z_{k,l} & else \end{Bmatrix} \tag{23}$$

where $P_r = 0.5$, $r \in 1, 2, \ldots, k-1, k+1, \ldots, t$. It is indeed a fresh start for the best global solution and can be completely altered as the fitness function is boosting and shrinking [10].

Mutation rate. Introducing mutation into the GWO–GOA as a method for higher diversity and thereby enhancing algorithm exploration functionality. Q_u is used as the control parameter for tuning the mutation operator by accumulating a random value optimized between [0, 1]. The posting can be outlined in this operation for GWO–GOA,

$$z_{k,l} = \begin{Bmatrix} z_{Sl} + & \mu(z_{a,l} - z_{b,l}) & \text{if } rand < Q_u \\ & z_{k,l} & else \end{Bmatrix} \tag{24}$$

where $Q_u = 0.07$, $a, b \in 1, 2, \ldots, k-1, k+1, \ldots, t$ and μ is value among [0, 1]. This likelihood is a major feature for the mutation operator to attain the best global solution and alterations as the fitness function rises and falls [10].

3.4.3. Text clustering technique

This section presents the steps of the clustering technique after getting a new subset of essential and informative features using the GWO–GOA algorithm.

3.4.3.1. Mathematical model of the text document clustering problem. The text clustering method is defined as: given D a huge set of text documents.

$$D = d1, d2 \ldots dj \ldots dn \tag{25}$$

where n represents the number of documents in the given document collection, d_1 represents the recorder (document) number 1, cos di, cj is an objective function to maximize the cosine similarity measure between the document number i and the cluster centroid number j .

3.4.3.2. Compute clusters centroid. To cluster a huge set of text documents into a subset of coherent clusters, each cluster represents one centroid, which requires updating in each iteration using Eq. (11). Each document is assigned to the similar cluster based on its similarity with the clusters centroids [40]. Ck is the centroid of cluster k, which is represented as a vector.

$$Ck = (ck1, ck2 \ldots ckj, \ldots, ckt), ckj \tag{26}$$

Is the centroid of cluster j and t is the length of the cluster centroid.

$$c_{kj} = \frac{\sum_{i=1}^{n}(a_{ki})d_i}{\sum_{j=1}^{r_i} a_{kj}} \tag{27}$$

where di denotes the document number i that belongs to cluster centroid number i, akj is the total number of text documents that belong to cluster j, ri is the number of text documents in cluster i.

3.4.3.3. Similarity measure. Cosine is the standard measure used in the document clustering technique to compute the similarity score between two vectors (i.e., document and cluster centroid) as $d1$ is document number 1 and $d2$ is the cluster centroid.

$$\cos(d1, d2) = \frac{\sum_{j=1}^{t} w(t_j, d_1) \times w(t_j, d_2)}{\sqrt{\sum_{j=1}^{t} w(t_j, d_1)^2}\sqrt{\sum_{j=1}^{t} w(t_j, d_2)^2}} \tag{28}$$

where $w(tj, d_1)$ is the weight of term j in the document number 1, $\sum_{j=1}^{t} w(t_j, d_1)^2$ is the square of the summation of terms score for the document number 1 from $j = 1$ to t and $\sum_{j=1}^{t} w(t_j, d_2)^2$ is the square of the summation of terms score for the document number 2 from $j = 1$ to t which $d2$ denotes the cluster centroid [41].

4. Results and discussions

4.1. Dataset

Text clustering of the conventional benchmark datasets is made accessible by numerical form regarding extraction of the aspects "http://sites.labic.icmc.usp.br/textcollections/". Reuters21578 contains 200 random documents in four groups in the initial dataset (DS1). The second dataset, known as DS2, contains 100 random documents belonging to five groups, named 20 newsgroups. Reuters21578 is the third dataset (DS3) that comprises 100 documents in eight groups. The fourth (DS4) dataset, known as 20Newsgroups, comprises 200 random documents in ten groups. Dmoz-Business, the fifth dataset (DS5), contains 300 random documents from ten groups. DMOZ-Science is called the sixth dataset (DS6) and incorporates 1000 random documents in twelve groups. Reuters 21578 provides 5000 random documents in the seventh dataset (DS7) having access to 15 groups. Finally, there are 10.000 random documents in the 8th dataset (DS8), known to be 20 Newsgroups, from the twenty groups [9].

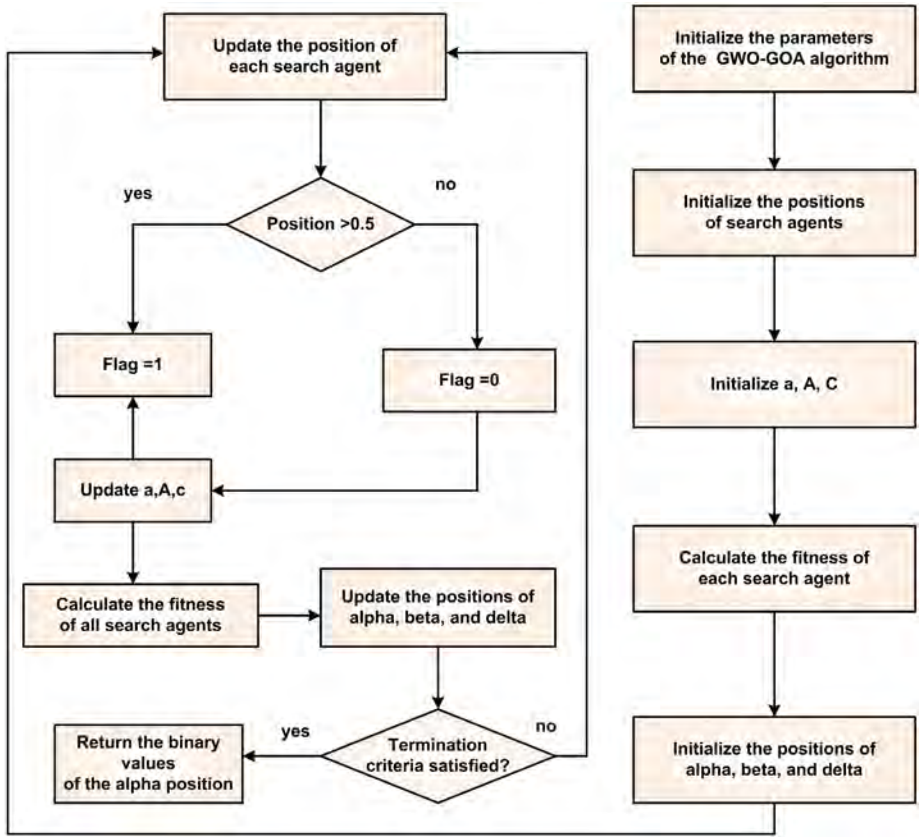

Fig. 5. Flow chart of GWO–GOA.

4.2. Evaluation metrics

4.2.1. Accuracy

Accuracy (AM) is the measure of exactness which can be measured by the ratio value of the documents assigned to each cluster that is represented in Eq. (29),

$$AM = \frac{1}{n}\sum_{p}^{q} n_{p,p} \tag{29}$$

where n, q, p is a cluster p number of correct members, n are the number of all documents; q is cluster number [10].

4.2.2. F-measure

The pervasive quantification in the text clustering arena is F-measure (F) [42]. Two metrics impinge on the F-Measure which are precision and recall. These metrics are widely used in information retrieval. Precision is referred as the fraction of related information among the all recovered information, while recall referred as the fraction of the total amount of recovered information which is truly retrieved. The pervasive metrics in text mining used among Eqs. (30) and (31) are precision and recall. Premised on use for these two measures, F-measure is ascertained in Eq. (32),

$$PR(k,l) = \frac{n_{k,l}}{n_l} \tag{30}$$

$$RE(k,l) = \frac{n_{k,l}}{n_l} \tag{31}$$

where $n_{k,l}$ is the number of members of class k in cluster l, n_l is the number of cluster l and n_k is the number of members of class k.

$$F_M(l) = \frac{2 * PR(k,l) * RE(k,l)}{PR(k,l) * RE(k,l)} \tag{32}$$

where $PR(k,l)$ is precision of members of class k in cluster l, $RE(k,l)$ is the recall of members of class k in cluster l, and F-measure for every cluster is formulated by Eq. (33),

$$F_M = \sum_{l} \frac{n_l}{n} \max_{k} \{n(k,l)\} \tag{33}$$

4.2.3. Sensitivity

This measure is used to quantify accurately identified substantial part of positive variations [43].

$$S_E = \frac{ps}{ps + qn} \tag{34}$$

4.2.4. Specificity

This measure is used for ascertaining the proportion of negatively labeled structures [43].

$$S_P = \frac{pd}{pd + qn} \tag{35}$$

4.3. Performance evaluation

The performance evaluation of the proposed methodology is compared with PSO-k means [44], SCA-k means [45] and k-means clustering methodology [46] and the following outcomes are being put forth.

The fitness for feature selection is compared for various iterations and Fig. 6 shows the graph plotted for iterations versus fitness for feature selection. It is noted from the graph that as the number of iteration increases, the fitness value increases. For 5 iterations, the fitness value for the feature selection is about 175.45. For 10 iterations, the fitness value increases to 197.347. For 15 iterations, the fitness value is 225.343. For 20 iterations,

Table 1
The parameter settings for all the algorithms.

Algorithm	Main parameter settings
PSO-k means	Particle number $n = 30$; Learning factor $c1 = 2$, $c2 = 2$; Inertia weight $w = 0.9$
SCA-k means	The population size is 100, $a = 2$
k-means clustering methodology	Neighborhood $\in = 2$, minimum points parameter $N_{\min} = 0.8$
Hybridizing GWO–GOA	Wolves number $N = 30$; filter capacitance $Cf = 2500$, filter inductance $Lf = 0.94$ mH

Table 2
Computational complexity analysis of GWO–GOA.

Operations	Computational complexity
Initialize the positions	$O(N)$
Update the position	$O(N)$
Calculate the fitness	$O(n^3)$
Termination	$O(n^2.N)$

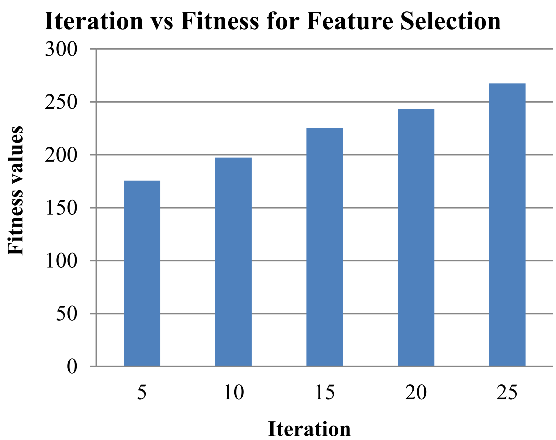

Fig. 6. Iteration vs fitness for feature selection.

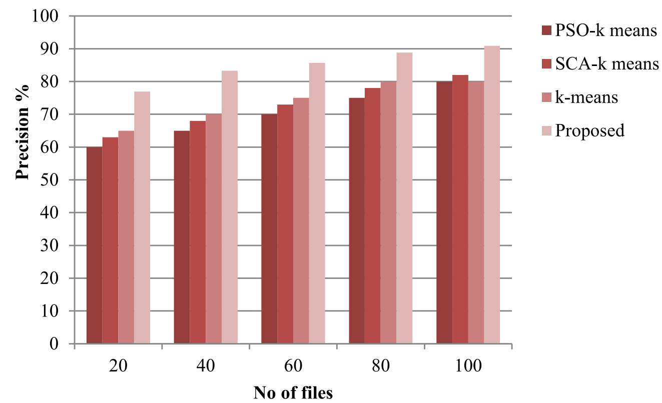

Fig. 7. Precision value with number of files.

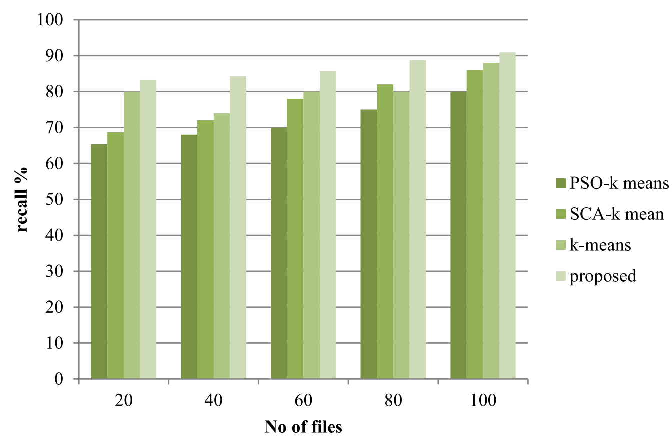

Fig. 8. Recall value with number of files.

the fitness value raises to 243.43 and for 25 iterations, the fitness value reaches to 267.34.

Inspecting the results, the comparison of the PSO-k means methodology and proposed methodology is tabulated in Table 1 for precision, recall, F-measure and accuracy with the number of files. Table 2 represents the comparison of k means methodology with the proposed methodology for sensitivity, specificity, execution time and memory usage with a number of files. (See Tables 3 and 4.)

The graphical representation of the comparison with particle swarm optimization based K means (PSO-k means) methodology [44], sine cosine algorithm based KM clustering (SCA-k means) [45], k-means clustering [46]and proposed methodology for precision with a number of files is shown in Fig. 7.

The experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 76.9% precision whereas the PSO-k means and SCA-k means, and k-means clustering show 60% and 63% and 65% respectively. The proposed method precision increases by 28% and 22% and 20% when compared to PSO-k means, SCA-k mean, and k-means clustering. As the file number increases the precision value also increases gradually. For 40 files, the precision is increased to 83.3% for the proposed methodology. For 60 files, the precision is increased to 85.7% for the proposed methodology. For 80 files, the precision is increased to 88.8% for the proposed methodology and for 100 file, the precision is increased to 90.9% for the proposed methodology. Considering the overall performance, the precision value of the proposed is increased by an average of 9% when compared with the PSO-k means, SCA-k means methods and k-means clustering. The graphical representation of the comparison between existing K means methodology and the proposed methodology for recall with number of files is shown in Fig. 8.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 83.3% recall whereas the PSO-k means, SCA-k means and k-means clustering show 65.4%, 68.7% and 80% respectively. The proposed method recall increases by 27%, 21% and 18% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the recall value also increases gradually. For 40 files, the recall is increased to 84.3% for the proposed methodology. For 60 files, the recall is increased to 85.7% for the proposed methodology. For 80 files, the recall is increased to 88.8% for the proposed methodology and for 100 files; the recall is increased to 90.9% for the proposed methodology. Considering the overall performance, the recall value of the proposed is increased by an average of 8.8% when comparing with the PSO-k means, SCA-k means and k-means clustering methods.

The graphical representation of the comparison between existing PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for F-measure with number of files is shown in Fig. 9.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 80% F- measure whereas PSO-k means SCA-k means and k-means

Table 3
Comparison of k-means methodology with the proposed methodology for precision, recall, F-measure and accuracy.

No. of files	Precision (%)		Recall (%)		F-Measure (%)		Accuracy (%)	
	k-means	Proposed	k-means	Proposed	k-means	Proposed	k-means	Proposed
20	68.4	76.9	67.4	83.3	71.3	80	68.3	75
40	71.3	83.3	72.3	83.3	74.2	83.3	71.5	80
60	78.4	85.7	79.3	85.7	79.4	85.7	74.3	83.3
80	80.5	88.8	82.4	88.8	82.3	88.8	81.2	87.5
100	83.3	90.9	86.3	90.9	84.2	90.9	83.4	90

Table 4
Comparison of k means methodology with proposed methodology sensitivity, specificity, execution time and memory usage.

No. of files	Sensitivity (%)		Specificity (%)		Execution time (s)		Memory usage (KB)	
	k-means	Proposed	k-means	Proposed	k-means	Proposed	k-means	Proposed
20	72.3	83.3	60.3	62.5	12.3	10.4	5822387	5421775
40	78.3	83.3	71.3	75	18.1	16.6	6044683	5732321
60	80.3	85.7	73.4	80	24.3	21.9	6520500	6033761
80	82.3	88.8	80.3	85.7	29.3	27.5	7065100	6422786
100	85.4	90.9	82.3	88.8	37.4	33.8	7540340	7077865

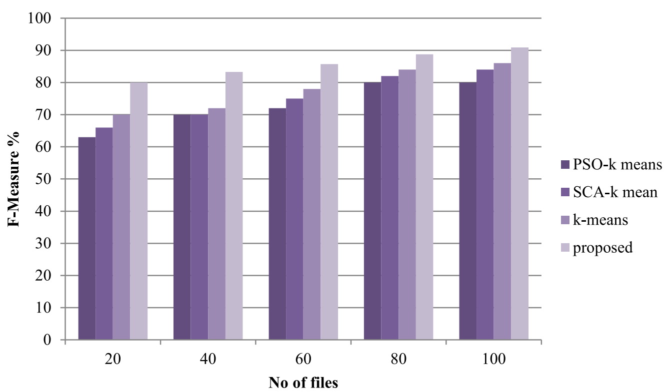

Fig. 9. F-measure value with number of files.

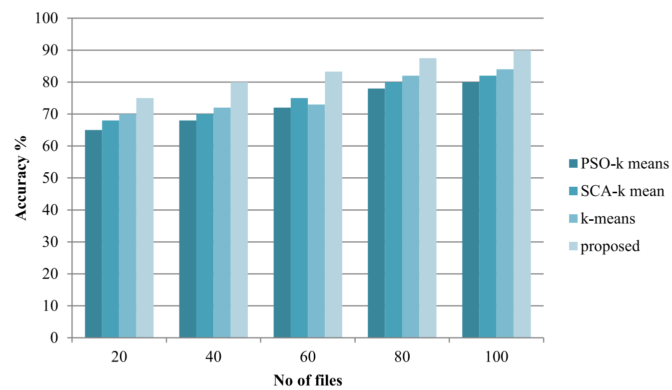

Fig. 10. Accuracy percentage with number of files.

clustering methods show 62%, 66% and 70% respectively. The proposed method's F-measure increases by 29%, 21% and 19% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the F-measure value also increases gradually. For 40 files, the F-measure value is increased to 83.3% for the proposed methodology. For 60 files, the F-measure value is increased to 85.7% for the proposed methodology. For 80 files, the F-measure value is increased to 88.8% for the proposed methodology and for 100 file, the F-measure value is increased to 90.9% for the proposed methodology. Considering the overall performance, the F-measure value of the proposed is increased by an average of 8.8% when comparing with the PSO-k means, SCA-k means and k-means clustering methods. The graphical representation of the comparison between PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for accuracy with number of files is shown in Fig. 10.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 75% accuracy whereas the e PSO-k means. SCA-k means and k-means clustering methods show 65%, 68% and 70%. The proposed method's accuracy increases by 15%, 17% and 20% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the accuracy value also increases gradually. For 40 files, the accuracy value is increased to 80% for the proposed methodology. For 60 files, the accuracy value is increased to 83.3% for the proposed methodology. For 80 files, the accuracy value is increased to 87.5% for the proposed methodology and for 100 file, the accuracy value is increased

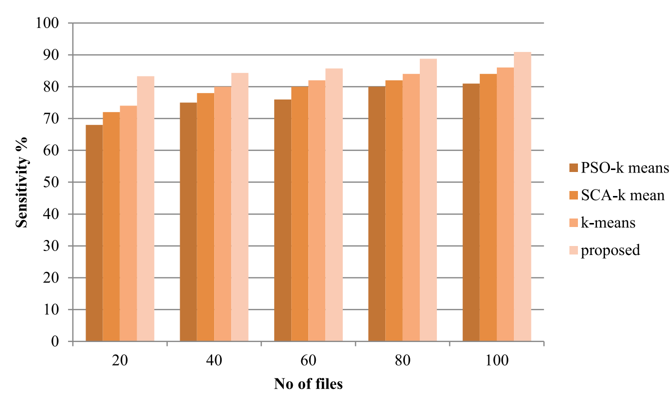

Fig. 11. Sensitivity percentage with number of files.

to 90% for the proposed methodology. Considering the overall performance, the accuracy value of the proposed is increased by an average of 7.6% when comparing with the PSO-k means, SCA-k means and k-means clustering methods. The graphical representation of the comparison between PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for sensitivity with number of files is shown in Fig. 11.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 83.3% sensitivity whereas the PSO-k means SCA-k means and k-means clustering show 68%, 72% and 74% respectively. The

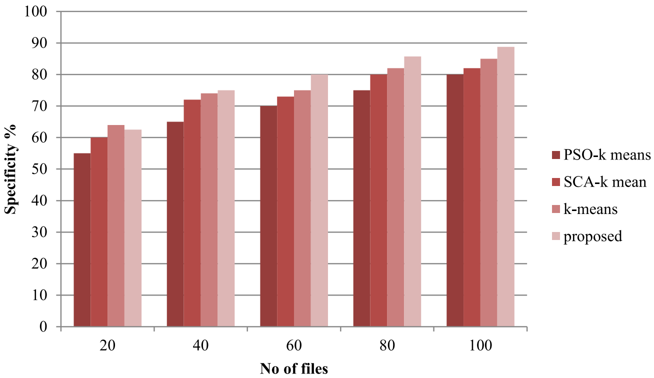

Fig. 12. Specificity value with number of files.

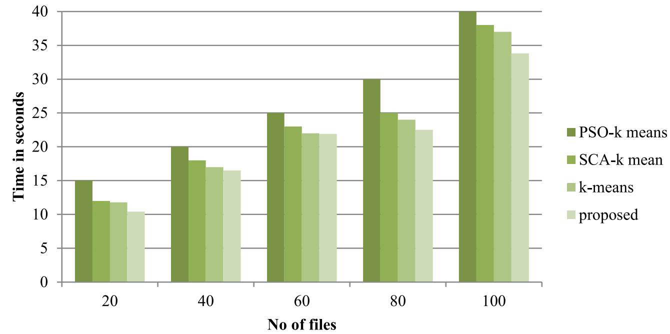

Fig. 13. Execution time vs. number of files.

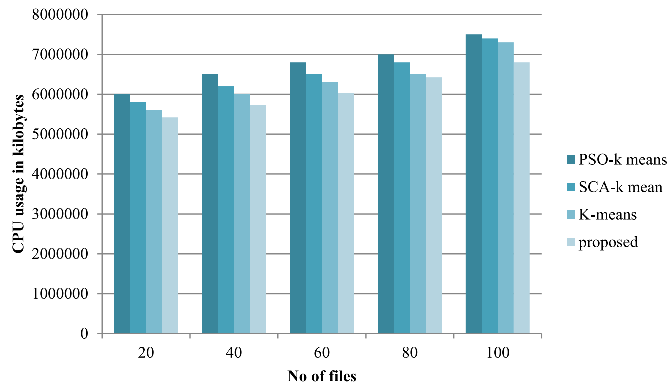

Fig. 14. Memory usage vs. number of files.

proposed method's sensitivity increases by 22%, 15% and 14% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the sensitivity value also increases gradually. For 40 files, the sensitivity value is increased to 84.3% for the proposed methodology. For 60 files, the sensitivity value is increased to 85.7% for the proposed methodology. For 80 files, the sensitivity value is increased to 88.8% for the proposed methodology and for 100 file, the sensitivity value is increased to 90.9% for the proposed methodology. Considering the overall performance, the sensitivity value of the proposed is increased by an average of 7.6% when comparing with the PSO-k means, SCA-k means and k-means clustering methods. The graphical representation of the comparison between PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for specificity with number of files is shown in Fig. 12.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 62.5% specificity whereas the existing methods show 55%, 60% and 64%. The proposed method's specificity increases by 13% and 4% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the specificity value also increases gradually. For 40 files, the specificity value is increased to 75% for the proposed methodology. For 60 files, the specificity value is increased to 80% for the proposed methodology. For 80 files, the specificity value is increased to 85.7% for the proposed methodology and for 100 file, the specificity value is increased to 88.8% for the proposed methodology. Considering the overall performance, the specificity value of the proposed is increased by an average of 4.8% when comparing with the PSO-k means, SCA-k means and k-means clustering methods. The graphical representation of the comparison between PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for execution time with number of files is shown in Fig. 13.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 10.4 s execution time whereas the PSO-k means SCA-k means and k-means clustering show 15 s, 12 s and 11.8 s respectively. The proposed method's execution time decreases by 3%, 13% and 10% when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the execution time decreases gradually. The performance level increases in the proposed methodology as the execution time decreases. For 40 files, the execution time is decreased to 16.5 s for the proposed methodology. For 60 files, the execution time is decreased to 21.9 s for the proposed methodology. For 80 files, the execution time is decreased to 27.5 s for the proposed methodology and for 100 files, the execution time is decreased to 33.8 s for the proposed methodology. Considering the overall performance, the execution time of the proposed method is decreased by an average of 2.5% when comparing with the PSO-k means, SCA-k means and k-means clustering methods. The graphical representation of the comparison between PSO-k means, SCA-k means and k-means clustering methodologies and the proposed methodology for memory usage with number of files is shown in Fig. 14.

Experiment is analyzed for the number of files in the range 20, 40, 60, 80 and 100. For 20 files, the proposed method shows 5421775 KB memory usage whereas the PSO-k means, SCA-k means and k-means clustering methods show 60000000 KB, 5800000 KB and 5600000 KB respectively. The proposed method's memory usage decreases by 400612 KB when compared to the PSO-k means, SCA-k means and k-means clustering methods. As the file number increases the memory usage decreases gradually. The performance level increases in the proposed methodology as the memory usage decreases. For 40 files, the memory usage is decreased to 5732321 KB for the proposed methodology. For 60 files, the memory usage is decreased to 6033761 KB for the proposed methodology. For 80 files, the memory usage is decreased to 6422786 KB for the proposed methodology and for 100 file, the memory usage is decreased to 6800000 KB for the proposed methodology. Considering the overall performance, the memory usage of the proposed method is decreased by an average memory usage of 460900.4 KB when comparing with the PSO-k means, SCA-k means and k-means clustering methods.

All the computations are performed in the same system. In Fig. 15 the proposed method compared with various algorithms like GWO [47], GOA [20] and PSO algorithm [28]. The running time is measured in terms of seconds over 100 iterations.

5. Conclusion and future scope

We utilize JAVA Net beans software for performance evaluation to our work. GWO algorithm has been enhanced with the

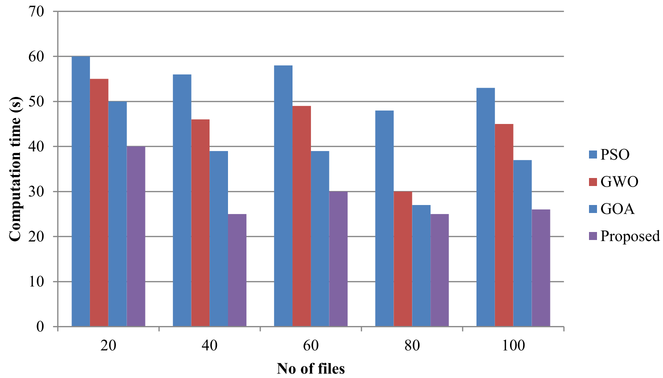

Fig. 15. Comparison results of files vs. computation time.

GOA approach and the pragmatic approach is called a GWO–GOA. This approach is intended to improve meta-heuristic algorithms convergence levels. A certain transition amplifies the algorithm's search potential. Compared to the existing text categorization algorithms, this hybrid algorithm is found to be effective and optimal. GWO algorithms can rapidly congregate in this hybrid approach, even if it has viable search functions in the problem area and can productively find minimum feature subset. The proposed findings of the tasks indicate its consistent performance and categorically state it. A good sub-set of functionalities in the data set is ended up finding in the GOA algorithm. The hybrid algorithm of GWO–GOA ends up working productively on the selected feature conundrum. GWO–GOA demonstrates much higher results, as well as its resilience and rapid convergence speed. Improvement to our work might involve the use of other functional selection algorithms and different fitness functions that are expected to strengthen the success rates.

CRediT authorship contribution statement

R. Purushothaman: Conceptualization, Methodology, Writing - original draft. **S.P. Rajagopalan:** Supervision. **Gopinath Dhandapani:** Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] D. Agnihotri, K. Verma, P. Tripathi, Variable global feature selection scheme for automatic classification of text documents 81 (2018) 268–281.
[2] L. Abualigah, A. Khader, E. Hanandeh, A novel weighting scheme applied to improve the text document clustering techniques, Innov. Comput. Optim. Appl. 30 (2017) 5–320.
[3] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, J. Gan, Unsupervised feature selection by self-paced learning regularization, Pattern Recognit. Lett. (2018).
[4] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, S. Zhang, Graph self-representation method for unsupervised feature selection 220 (2018) 130–137.
[5] M. Luo, F. Nie, X. Chang, Y. Yang, A. Hauptmann, Q. Zheng, Adaptive unsupervised feature selection with structure regularization, IEEE Trans. Neural Netw. Learn. Syst. 29 (4) (2018) 944–956.
[6] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[7] N. Kushwaha, M. Pant, Link based BPSO for feature selection in big data text clustering, Future Gener. Comput. Syst. 82 (2018) 190–199.
[8] A. Fathy, Recent meta-heuristic grasshopper optimization algorithm for optimal reconfiguration of partially shaded PV array, Sol. Energy 171 (2018) 638–651.
[9] K. Bharti, P. Singh, Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering, Appl. Soft Comput. 43 (2018) 20–34.
[10] A. Ghareb, A. Bakar, A. Hamdan, Hybrid feature selection based on enhanced genetic algorithm for text categorization, Expert Syst. Appl. 49 (2016) 31–47.
[11] I. Aljarah, A. Al-Zoubi, H. Faris, M. Hassonah, S. Mirjalili, H. Saadeh, Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm, Cogn. Comput. 10 (2018) 478–495.
[12] N. Kushwaha, M. Pant, Link based BPSO for feature selection in big data text clustering, Future Gener. Comput. Syst. 82 (2018) 190–199.
[13] M. Mafarja, I. Aljarah, A. Heidari, A. Hammouri, H. Faris, A. Al-Zoubi, et al., Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, Knowl.-Based Syst. 145 (2018) 25–45.
[14] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, J. Zhao, B. Xu, Self-taught convolutional neural networks for short text clustering, Neural Netw. 88 (2018) 22–31.
[15] M. Tutkan, M. Ganiz, S. Akyokuş, Helmholtz principle based supervised and unsupervised feature selection methods for text mining, Inf. Process. Manage. 52 (5) (2018) 885–910.
[16] D. Thiyagarajan, N. Shanthi, A modified multi objective heuristic for effective feature selection in text classification, Cluster Comput. (2017) 1–11.
[17] M. Mafarja, I. Aljarah, A.A. Heidari, A.I. Hammouri, H. Faris, A.Z. Ala'M, S. Mirjalili, Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, Knowl.-Based Syst. 145 (2018) (2018) 25–45.
[18] I. Aljarah, M. Mafarja, A.A. Heidari, H. Faris, S. Mirjalili, Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach, Knowl. Inf. Syst. 62 (2) (2020) 507–539.
[19] H. Rashaideh, A. Sawaie, M.A. Al-Betar, L.M. Abualigah, M.M. Al-Laham, M. Ra'ed, M. Braik, A grey wolf optimizer for text document clustering, J. Intell. Syst. 29 (1) (2018) 814–830.
[20] S. Łukasik, P.A. Kowalski, M. Charytanowicz, P. Kulczycki, Data clustering with grasshopper optimization algorithm, in: 2017 Federated Conference on Computer Science and Information Systems, FedCSIS, 2017, IEEE, 2017, pp. 71–74.
[21] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, J. Comput. Des. Eng. 5 (2017) 458–472.
[22] z. Decoder, Behavior of grey wolf optimization (GWO) algorithm using meta-heuristics method, in: Transpire Online 2019, 2019, Available at: https://transpireonline.blog/2019/08/09/behavior-of-grey-wolf-optimization-gwo-algorithm-using-meta-heuristics-method/. (Accessed on: Sep, 2019).
[23] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[24] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[25] z. Decoder, Grasshopper optimization algorithm (GOA): Propelled from swarming behavior of grasshoppers for solving optimization problems, in: Transpire Online 2019, 2019, Available at: https://transpireonline.blog/2019/07/25/grasshopper-optimization-algorithm-goa-propelled-from-swarming-behavior-of-grasshoppers-for-solving-optimization-problems/. (Accessed on: Sep, 2019).
[26] Q. Bui, Q. Nguyen, X. Nguyen, V. Pham, H. Nguyen, V. Pham, Verification of novel integrations of swarm intelligence algorithms into deep learning neural network for flood susceptibility mapping, J. Hydrol. 581 (2020) 124379.
[27] E. Elhariri, N. El-Bendary, A.E. Hassanien, A. Abraham, Grey wolf optimization for one-against-one multi-class support vector machines, in: 2015 7th International Conference of Soft Computing and Pattern Recognition, SoCPaR, IEEE, 2015, pp. 7–12.
[28] R. Janani, S. Vijayarani, Text document clustering using spectral clustering algorithm with particle swarm optimization, Expert Syst. Appl. 134 (2019) 192–200.
[29] P. Singh, Chaotic gradient artificial bee colony for text clustering, Soft Comput. 20 (3) (2015) 1113–1126.
[30] S. V, R. J, Text mining: open source tokenization tools – an analysis, Adv. Comput. Intell. 3 (2016) 37–47.
[31] A. Jabbar, S. Iqbal, A. Akhunzada, Q. Abbas, An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach, J. Exp. Theor. Artif. Intell. 30 (5) (2018) 703–723.

[32] L. Zhang, L. Jiang, C. Li, G. Kong, Two feature weighting approaches for naive Bayes text classifiers, Knowl.-Based Syst. 100 (2016) 137–144.
[33] A. Sangaiah, A. Fakhry, M. Abdel-Basset, I. El-henawy, Arabic text clustering using improved clustering algorithms with dimensionality reduction, Cluster Comput. (2018) 1–15.
[34] L. Abualigah, A. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[35] W.T. Howe, A. Hinderaker, "The rule was the rule": new member socialization in rigidly structured totalistic organizations, Atl. J. Commun. 26 (3) (2018) 180–195.
[36] V.K. Bohat, K.V. Arya, S.S. Rajput, Prey phase based grey wolf optimizer, in: 2018 Conference on Information and Communication Technology, CICT, IEEE, 2018, pp. 1–5.
[37] S.Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, Appl. Intell. 48 (4) (2018) 805–820.
[38] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[39] S. Arora, H. Singh, M. Sharma, S. Sharma, P. Anand, A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection, IEEE Access 1 (1) (2019) 1–19.
[40] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[41] L.M. Abualigah, A.T. Khader, Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering, J. Supercomput. 73 (11) (2017) 4773–4795.
[42] M. Kohli, S. Arora, Chaotic grey wolf optimization algorithm for constrained optimization problems, J. Comput. Des. Eng. 5 (4) (2018) 458–472.
[43] L.M. Abualigah, A.T. Khader, E.S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, J. Comput. Sci. 25 (2018) 456–466.
[44] S.J. Nanda, I. Gulati, R. Chauhan, R. Modi, U. Dhaked, A k-means-galactic swarm optimization-based clustering algorithm with Otsu's entropy for brain tumor detection, Appl. Artif. Intell. 33 (2) (2019) 152–170.
[45] S. Gupta, K. Deep, A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Syst. Appl. 119 (2019) 210–230.
[46] J.D. Rose, An efficient association rule based hierarchical algorithm for text clustering, Int. J. Adv. Eng. Technol. VII (I) (2016).
[47] R. Rajakumar, J. Amudhavel, P. Dhavachelvan, T. Vengattaraman, GWO-LPWSN: Grey wolf optimization algorithm for node localization problem in wireless sensor networks, J. Comput. Netw. Commun. 2017 (2017).