algorithms

MDPI

Article

Multi-Objective Unsupervised Feature Selection and Cluster Based on Symbiotic Organism Search

Abbas Fadhil Jasim AL-Gburi [1,*], Mohd Zakree Ahmad Nazri [1], Mohd Ridzwan Bin Yaakub [1] and Zaid Abdi Alkareem Alyasseri [2,3]

[1] Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; zakree@ukm.edu.my (M.Z.A.N.); ridzwanyaakub@ukm.edu.my (M.R.B.Y.)
[2] Information Technology Research and Development Center (ITRDC), University of Kufa, Najaf 54001, Iraq; zaid.alyasseri@uokufa.edu.iq
[3] College of Engineering, University of Warith Al-Anbiyaa, Karbala 56001, Iraq
* Correspondence: p104340@siswa.ukm.edu.my

Abstract: Unsupervised learning is a type of machine learning that learns from data without human supervision. Unsupervised feature selection (UFS) is crucial in data analytics, which plays a vital role in enhancing the quality of results and reducing computational complexity in huge feature spaces. The UFS problem has been addressed in several research efforts. Recent studies have witnessed a surge in innovative techniques like nature-inspired algorithms for clustering and UFS problems. However, very few studies consider the UFS problem as a multi-objective problem to find the optimal trade-off between the number of selected features and model accuracy. This paper proposes a multi-objective symbiotic organism search algorithm for unsupervised feature selection (SOSUFS) and a symbiotic organism search-based clustering (SOSC) algorithm to generate the optimal feature subset for more accurate clustering. The efficiency and robustness of the proposed algorithm are investigated on benchmark datasets. The SOSUFS method, combined with SOSC, demonstrated the highest f-measure, whereas the KHCluster method resulted in the lowest f-measure. SOSFS effectively reduced the number of features by more than half. The proposed symbiotic organisms search-based optimal unsupervised feature-selection (SOSUFS) method, along with search-based optimal clustering (SOSC), was identified as the top-performing clustering approach. Following this, the SOSUFS method demonstrated strong performance. In summary, this empirical study indicates that the proposed algorithm significantly surpasses state-of-the-art algorithms in both efficiency and effectiveness. Unsupervised learning in artificial intelligence involves machine-learning techniques that learn from data without human supervision. Unlike supervised learning, unsupervised machine-learning models work with unlabeled data to uncover patterns and insights independently, without explicit guidance or instruction.

Keywords: unsupervised learning; symbiotic organisms search algorithm; clustering; unsupervised feature selection; multi-objective

check for updates

Citation: AL-Gburi, A.F.J.; Nazri, M.Z.A.; Yaakub, M.R.B.; Alyasseri, Z.A.A. Multi-Objective Unsupervised Feature Selection and Cluster Based on Symbiotic Organism Search. *Algorithms* **2024**, *17*, 355. https://doi.org/10.3390/a17080355

Academic Editor: Mohammad Majid al-Rifaie

Received: 20 July 2024
Revised: 8 August 2024
Accepted: 9 August 2024
Published: 14 August 2024

1. Introduction

In the ever-expanding landscape of data-driven applications, unsupervised learning techniques play a pivotal role in extracting meaningful patterns from raw data. Clustering, as one of the fundamental tasks in unsupervised learning, seeks to group similar data points together while maintaining separation between distinct clusters [1]. Several research works have been carried out, and various clustering approaches have been proposed, including kernel methods such as support vector machine (SVM) [2], self-organizing maps (SOM) [3], and k-means clustering [4]. However, achieving optimal clustering results remains a challenging endeavor due to various factors such as noisy features, high dimensionality, and the need for robust initialization.

Unsupervised feature selection (UFS), on the other hand, aims to identify the most relevant subset of features from the original feature space [5]. By selecting informative features, computational complexity would be reduced, and the quality of clustering results could also be enhanced. Based on evaluation criteria, the current unsupervised feature-selection studies can also be categorized into two primary groups: wrapper and filter-based studies [6]. The evaluation criterion for wrapper-based techniques is the chosen features' classification performance. On the other hand, the assessment criterion in filter techniques remains unaffected by the machine-learning technology. The filter approaches employ a variety of metrics, including distance measurements [7], consistency measures [8], correlation measures [9], and information theory-based measures [10]. Wrapper methods generally outperform filter methods because they evaluate the performance of the unsupervised selected features on a classification algorithm, even though filter methods are usually less computationally expensive [11]. However, these selection techniques continue to face issues with high computational time and convergence to local optima [12]. In addition, traditional unsupervised feature-selection methods often operate independently of the clustering algorithm, overlooking the inherent synergy between feature selection and the clustering process [13]. Metaheuristic techniques have been frequently adopted recently due to their robust global-search capabilities, which help overcome these shortcomings, especially when the number of features increases. Some of these classic metaheuristic algorithms, most widely applied to the unsupervised feature-selection and clustering problems, include the genetic algorithm (GA) [14–16], particle swarm optimization (PSO) [17–19], and harmony search algorithms, among others [20,21].

The SOS technique, first introduced by [22], is a stochastic metaheuristic approach using randomization to determine a collection of solutions. Based on the interactions between species in an ecosystem, the SOS algorithm was designed with a faster convergence time and greater robustness than these classic metaheuristic algorithms [23]. When compared to other population-based metaheuristic algorithms that searched for near-optimal solutions by training a set of candidate solutions using population characteristics to iteratively guide the searching, like the ant colony optimization (ACO) algorithm, the SOS algorithm is better for three key reasons. One benefit of the mutualism and commensalism stages of the SOS algorithm is that it concentrates on creating new creatures, which makes it possible for the algorithm to find a variety of solutions. It follows that the algorithm becomes more adept at exploring. For a second reason, the parasitism phase makes the algorithm more exploitation-capable by keeping it from being stuck in local optima. The final benefit is that there are just two general parameters for the SOS algorithm: the maximum number of iterations and the population size. Because of all these benefits, the SOS algorithm is widely used and has been modified to address a variety of optimization issues across a number of industries. Recently, to enhance its performance, modified [24] and hybrid [25] versions of the SOS algorithm have been developed as an alternative to the initial SOS algorithm proposed by [22]. Ref. [20] addressed the supervised feature-selection issue for 19 datasets from the UCI repository using the binary version of the SOS algorithm. The results indicated that, for the majority of datasets, the binary SOS algorithm may achieve a high classification accuracy with the fewest characteristics. The SOS algorithm has also been used to solve multi-objective problems in optimization. A multi-objective symbiotic organism search technique based on the weighted-sum method was proposed by [26] as a supervised learning method for economic/emission dispatch problems in power systems. The proposed method has been found to outperform other optimization algorithms such as the genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO), the bees algorithm (BA), the mine blast algorithm (MBA), ant colony optimization (ACO), and cuckoo search (CS).

The application of SOS algorithms has since increased, particularly in the engineering field [27]. Though unsupervised learning has the capability to improve computational efficiency and retrieval recall, very few studies has been carried out in the literature specifically addressing unsupervised learning problems such as feature selection and clustering [27,28]. Previous studies concentrated on identifying optimal feature selection for brain–computer interfaces [26] and satellite image classification issues [29]. Within the literature, Refs. [30,31] explored text clustering and feature selection utilizing the SOS method. In their empirical research, Ref. [32] addressed text classification problems, and [31] proposed an SOS-based approach for feature extraction issues. Though the literature provides a larger proportion of works on single-objective approaches than on multi-objective optimization methods, it is observed that multi-objective optimization methods for FS problems based on metaheuristics techniques are not sufficiently examined [11]. Non-dominated sorting GA II (NSGA-II) or its variants form the basis of the majority of multi-objective techniques [33–36]. Other evolutionary computation (EC) approaches used in multi-objective feature selection include DE [37], ACO [38], and PSO [39]. According to the results of all these studies, multi-objective optimization algorithms outperform single-objective techniques in terms of both the quantity of features needed for supervised learning and classification performance. However, the existing literature has predominantly focused on datasets of modest to intermediate scale, indicating that the multi-objective feature-selection problem remains an unexplored field of study for unsupervised learning, much like high-dimensional data clustering. In addition, given that multi-objective evolutionary computation algorithms used for the FS problem are based on conventional algorithms like ACO, PSO, and GA, which typically have significant drawbacks such as slow convergence rate, high computational complexity, and trapping into local optima, there is also a need to investigate a novel multi-objective algorithm's capability to handle the feature-selection problems [33,40].

Although feature selection has been studied extensively, the literature review indicates that multi-objective unsupervised learning for two problems—unsupervised feature selection and clustering—has received relatively less attention. Furthermore, existing multi-objective research faces many of the aforementioned issues and has not addressed large-scale datasets such as TDT TREC data. This study proposes a multi-objective algorithm with a wrapper-based approach for data clustering, taking into account the shortcomings of the existing literature and the benefits of the SOS method. To the best of our knowledge, this study is the first to employ a multi-objective SOS algorithm to find the best possible unsupervised feature combination that maximizes clustering performance while decreasing the total number of selected features for a given set of data. To show the suggested method's robustness and dependability, it is evaluated using popular datasets from benchmark datasets. The results obtained are compared with the current approaches for both datasets, and the contribution related to the solution quality is given. The results of the study demonstrated that the proposed method performed better in terms of its capacity to provide acceptable outcomes, which included both an improvement in clustering performance and a reduction in the number of selected features. The robustness of this method is demonstrated by the better results it yields for both datasets. This work also examines and applies many SOS algorithm variants. The findings of these algorithms are compared with one another, and their benefits and drawbacks are identified.

The rest of the paper is organized as follows: Section 2 presents a review of related works, covering the background of the SOS algorithm, global-search unsupervised feature-selection algorithms based on SOS methods, and the clustering algorithm utilizing SOS algorithms. Section 3 outlines the proposed methods for this study. Sections 4 and 5 detail the experimental settings and results, respectively. Finally, the conclusion of the work is provided in Section 6.

2. Review of Related Works

This section includes the background of the SOS algorithm, global-search unsupervised feature-selection algorithms based on SOS methods, and clustering algorithms utilizing SOS approaches

2.1. Background of the SOS Algorithm

In this section, we introduce the original SOS algorithm, which is inspired by the three coexistence relations of mutualism, commensalism, and parasitism among organisms in the ecosystem. What follows explains these three phases of the SOS algorithm in detail.

2.1.1. Mutualism Phase

In this phase, interactions between two organisms, X_i and X_j, happen at random. Note that X_i and X_j, are two different organisms (where $X_i \neq X_j$). A mutual advantageous relationship between the two entities is formed by these interactions. Improved reciprocal survival rates of the two entities in the ecosystem are the goal of the correlation between X_i and X_j. By using Equations (1) and (2), the potential solutions X_{inew} and X_{jnew}, respectively, are obtained.

$$X_{inew} = X_i + \text{rand}(0,1) \times (X_{best} - X_{mutual} \times B_{F1}) \tag{1}$$

$$X_{jnew} = X_j + \text{rand}(0,1) \times (X_{best} - X_{mutual} \times B_{F2}) \tag{2}$$

where X_{mutual} is represented in Equation (3).

$$X_{mutual} = \frac{X_i + X_j}{2} \tag{3}$$

$$B_{F1} = 1 + \text{round}(\text{rand}(0,1)) \tag{4}$$

$$B_{F2} = 1 + \text{round}(\text{rand}(0,1)) \tag{5}$$

Within the range of 0 to 1, the $\text{rand}(0,1)$ function produces a vector of random numbers with a uniform distribution. According to [41], the organism designated as X_{best} exhibits the highest fitness function values in relation to their environmental adaptation. The term 'X_{mutual}', on the other hand, suggests that the two species exhibit mutualistic traits that promote their mutual survival. Equations (4) and (5) specify the random selection procedure used to determine the values of the benefit factors B_{F1} and B_{F2}. Those parameters show the degree of benefit resulting from interacting with each organism. Subsequently, the newly computed value of the fitness function is expressed as $f(X_{inew})$ and $f(X_{jnew})$. These values show better performance than the previous fitness functions, $f(X_i)$ and $f(X_j)$ [42]. Therefore, Equations (1) and (2) can be further transformed as follows:

$$X_{inew} = X_i + \text{rand}(0,1) \times (X_{best} - X_{mutual} \times B_{F1})\text{if } f(X_{inew}) > f(X_i) \tag{6}$$

$$X_{jnew} = X_j + \text{rand}(0,1) \times (X_{best} - X_{mutual} \times B_{F2})\text{if } f(X_{jnew}) > f(X_j) \tag{7}$$

2.1.2. Commensalism Phase

In the commensalism phase, two organisms, X_i and X_j, are randomly chosen from the ecosystem, and the organism X_i is updated according to Equation (8) [43].

$$X_{inew} = X_i + \text{rand}(-1,1) * (X_{best} - X_j),\ \text{if } f(X_{inew}) > f(X_i) \tag{8}$$

A vector with randomly distributed values, which are evenly spaced over the range of -1 to 1, is produced by the $\text{rand}(-1,1)$ function. If organism X_{inew} shows a superior fitness value, it may replace organism X_i. ($X_{best} - X_j$) refers to the benefits that X_i gains with respect to X_j. In addition, there is no new solution for X_j, since X_j gains nothing from the interaction. In this phase, solution X_j serves to update X_i which is contrary to the mutualism phase, and therefore, the vector X_{inew} is updated according to Equation (8).

2.1.3. Parasitism Phase

In a symbiotic relationship where one species exclusively benefits at the expense of another, parasitism is exemplified by the interaction between humans, the Anopheles mosquito, and the Plasmodium parasite. The Anopheles mosquito, acting as a vector for the parasite, remains unaffected, while the human host experiences negative effects. According to [42], the Plasmodium parasite reproduces within the human body. Therefore, in the solution search space, organism X_i generates an artificial vector $X_{parasite}$ to mimic the parasitic behaviors previously reported for the Anopheles mosquito. This is accomplished by modifying organism X_i's randomly selected dimension through a process of adjustment [42]. Following that, an organism called X_j is randomly selected from the environment to serve as the host for the parasite $X_{parasite}$. The $X_{parasite}$ will then try to replace X_j within the ecosystem. If $X_{parasite}$ proves to be more fit than X_j, X_j will be replaced with $X_{parasite}$. This implies that X_j develops immunity to $X_{parasite}$, which ultimately causes $X_{parasite}$ to become extinct in the environment. This can be expressed as follows:

$$X_{parasite} = \text{rand}(0,1) \times (\text{UB} - \text{LB}) + \text{LB} \tag{9}$$

where the two boundary points that require addressing are denoted by LB (lower bound) and UB (upper bound). The majority of improvements to the traditional SOS algorithm have been achieved by the alterations to either the commensalism phase, the mutualism phase, or a combination of the two approaches. The addition of a fourth phase to the existing phases occurs only in exceptional and rare situations. This work presents a thorough analysis of several recent advances and hybridization approaches used in SOS algorithms, as reported in the previous studies.

2.2. Global-Search Unsupervised Feature-Selection Algorithms Based on SOS Methods

In the context of data mining and machine learning, unsupervised feature-selection methods have attracted a number of research interests as a result of their capability to identify and select relevant features without relying on class-label information. Feature selection aims to identify the relevant characteristics and remove the irrelevant features from the dataset in order to achieve benefits such as a decrease in the dimensionality of the data, an improvement in the performance of classification methods, and helps in the learning process [26]. However, it is noteworthy that there are two types of feature extraction approaches: wrapper-/coating-based and filter-based methods [21]. Coating-based methods use classification algorithms as a criterion for selecting the best possible solutions. Conversely, filter-based methods show a reasonable level of computing efficiency and are not dependent on any particular algorithm.

Researchers argue that since coating-based systems incorporate classification algorithms in the evaluation criteria, they perform better than filter-based methods [44,45]. In light of this background, the authors [21] offer different wrapper-based binary techniques that use the SOS method to solve feature-selection issues. BSOSST was first utilized, while BSOSVT was used as the second model. The binary SOS (BSOS) is developed using these two coating-based approaches. In the third method, EEBCSOS demonstrated its efficacy and exploratory capabilities. In order to address the problem of feature selection, Ref. [46] presented a novel technique called the improved binary symbiotic organism search (IBSOS), which makes use of the wrapper method. To preserve the delicate balance between exploration and exploitation, the authors also included the same biological symbiosis approaches that are used in the continuous SOS method in the proposed IBSOS technique. The NSMOSOS algorithm, a wrapper-based multi-objective algorithm, was introduced by [26] to generate the ideal feature subset. The study evaluated a brain–computer interface (BCI) system's effectiveness and robustness for motor-imaging feature selection across two datasets, achieving the highest accuracy results for both. In a recent study, Ref. [31] introduced an innovative feature-selection technique to extract the most relevant features from extensive input data, using the BSOS metaheuristic to enhance email spam detection. Research by [20] demonstrated that the BSOS algorithm could identify the minimal set of features across various datasets while maintaining high classification accuracy. However, the Bayesian structural optimization method (BSOS) has limitations with low-dimensional datasets and reduced sensitivity in high-dimensional datasets. Additionally, Ref. [47] proposed a new feature-selection approach employing the SOS method to enhance the accuracy and effectiveness of sleep staging by using physiological data to classify sleep stages. To cope with high-dimensional feature-selection problems, parallel multi-objective optimization approaches were also proposed by [48]. The proposed multi-objective evolutionary alternative (MOEA) approaches were implemented on EEG signals for brain–computer interface (BCI) benchmarks, which show the superior results in terms of hypervolume and speedup.

When comparing performance, wrapper methods generally surpass filter methods. However, for high-dimensional datasets like microarray datasets, where sample sizes are smaller than the feature dimensions, the wrapper method can become computationally intensive. To address this, researchers have developed hybrid strategies that integrate both wrapper and filter techniques [48,49]. The authors tested five distinct discrete SOS algorithms [48]. These techniques aim to improve classification accuracy by optimizing the neighborhood sizes of the k-nearest neighbor method and feature subsets through a two-step process. First, a large number of features are removed using a filter approach. Subsequently, a wrapper approach is applied to select the best subset among the other features that remains. Different approaches to SOS and feature selection are compiled in Table 1.

Table 1. A list of the various feature-selection techniques and SOS algorithms.

S/N	SOS Variations	Feature-Selection Approaches	References	Supervised or Unsupervised Learning
1	Modified SOS	Wrapped based	[20]	Unsupervised learning
2	Hybrid method	Wrapper/coating-based	[21]	Unsupervised learning
3	Multi-objective SOS	Wrapper based	[26]	Unsupervised learning
4	Modified SOS	Wrapper based	[31]	Unsupervised learning
5	Improved SOS	Wrapper based	[46]	Supervised learning
6	Modified SOS	Filter based	[47]	Unsupervised learning
7	MOEA	Wrapper based	[48]	Unsupervised learning
8	Five distinct SOS algorithms that combine modified and hybridized techniques	Wrapper based	[49]	Supervised learning
9	Hybrid approach	Filter and wrapper-based	[50]	Supervised learning
10	Hybrid method	-	[51]	Unsupervised learning
11	Hybrid method		[52]	Supervised learning

2.3. Clustering Algorithms Based on SOS Methods

According to [26,49,53], SOS algorithms have been widely used for classification problems but not yet evaluated for the unsupervised feature-selection and clustering problems. An unsupervised technique for analyzing data referred to as clustering is used to find sets of objects that are homogeneous based on the values of their properties. Clustering can be divided into two primary categories: partitional clustering methods and hierarchical clustering methods. The recurrent hierarchical grouping of data objects is an essential component of the hierarchical clustering technique. In addition, a single partition of a dataset is created using the partitional clustering technique to find the underlying categories in the data. This method is non-hierarchical and is based on the application of a predetermined objective function. k-means is an example of this.

In an effort to further solve problems associated with automatic data clustering, an automatic k-means clustering-based SOS (CSOS) was proposed by [54]. By integrating the global and local search strategies in a sub-ecosystem built on the automated k-means clustering technique, the CSOS algorithm essentially performs a hybrid search approach. Mutualism and commensalism are the two distinct phases that are covered by the local search. Furthermore, in CSOS global search, this is referred to as the parasitism phase, in which only the best solution within a cluster can communicate with the best solutions in other clusters. Additionally, a data categorization methodology combining an SOS algorithm and a regularized extreme learning machine was developed by [55]. The newly developed algorithm, SOS-RELM, consists of two separate SVM phases: backpropagation and least-squares support vector machines. As a result, the SOS method is both an effective and efficient optimization approach. In the second phase, it optimizes the regularization parameters, invisible biases, and input weights. Moreover, [49] presented DSOS algorithms in a different study for the optimization of neighborhood size and feature subsets as supervised learning.

Furthermore, [53] presented a method for automatic data clustering that makes use of the SOS algorithm. In this article, we have discussed the ease of trapping in a local optimum and the strong dependence of the k-means clustering algorithm on the beginning solution. To create clusters, the SOS framework used an automated k-means clustering algorithm. The most effective solutions interact with one another within each cluster, combining local and global search techniques [56]. However, such an approach leads to a rise in computational cost.

Similarly, by including a multi-agent system (MAS) and self-adaptive benefit factors in the SOS algorithm, [57] created a multi-agent SOS (MASOS) to enhance the performance of the original SOS algorithm. Using this method, every organism acts as an agent interacting locally to choose the best course of action. When an agent chooses another agent from its immediate area, the SOS algorithm would execute in three separate steps. Despite that, there was a lot of computational complexity in the proposed task. The adopted clustering approaches that were described are listed in Table 2.

Table 2. The clustering approach adopted.

Authors	Adopted Clustering Approach
[54]	The number of clusters that form at the beginning is half of the ecosize, which is divided into smaller ecologies. Subsequently, CSOS optimization is then applied.
[58]	Enhancing the BDI and BIC through the optimization process.
[56]	The CVI functions as the objective function in the optimization problems of the Davies–Boulding index and the compact separated index, both of which must be minimized.
[53]	To optimize the clustering problem, the SOS algorithm randomly initializes the cluster within the ecosystem.
[59]	The benefit factors are found by a non-linear method, and their weights are used to effectively explore and exploit the search region.
[60]	The number of initial clusters is determined by the eco-size, which is the number of sub-ecosystems that the self-organizing system (SOS) forms and then optimizes.

3. Proposed Method

Unsupervised feature selection plays a crucial role in mitigating the curse of dimensionality, particularly in tasks like document clustering where high-dimensional text data are involved. The role of feature selections is multifaceted. They serve purposes such as enhancing performance (e.g., accuracy), aiding data visualization, simplifying model selection, and minimizing dimensionality by eliminating noise and unnecessary attributes [61,62].

In this study, a symbiotic organism search algorithm (SOS) was developed to solve numerical optimization over a continuous search space. The proposed SOS algorithm, like other population-based methods, iteratively employs a population of candidate solutions within the search space to find the optimal global solution. SOS starts with an initial population (referred to as the ecosystem), where organisms are randomly generated. Each organism represents a candidate solution, and its associated fitness value reflects its adaptation to the desired objective. This approach models the ecological interaction between two organisms in the ecosystem to control the production of new solutions. The three phases, including parasitism, commensalism, and mutualism, which resemble the real-world biological interaction framework, are shown.

The nature of interactions determines the primary principle for each phase. In the mutualism phase, interactions benefit both sides. In the commensalism phase, one side benefits without affecting the other. In the parasitism phase, one side benefits while actively harming the other. Throughout all stages, interactions between the organisms are random and continue until the termination conditions are met. The SOS algorithm processes are described in full in the following algorithm, and further information about the three phases as provided in the next section include

```
Initialization
REPEAT
    1. Mutualism phase.
    2. Commensalism phase.
    3. Parasitism phase.
UNTIL (the termination criterion is met).
```

3.1. Mutualism Phase

An illustrative example of mutualism, which benefits both participating organisms, is the symbiotic relationship between bees and flowers. Bees actively fly among flowers, collecting nectar that they transform into honey—a process beneficial to the bees themselves. Simultaneously, this activity also benefits the flowers, as bees inadvertently distribute pollen during their foraging, facilitating pollination. In the context of the SOS phase, this mutualistic interaction serves as a model.

In SOS, X_i is an organism matched to the *i*th member of the ecosystem. Another organism X_j is then selected randomly from the ecosystem to interact with X_i. Both organisms engage in a mutualistic relationship to increase the mutual survival advantage in the ecosystem. New candidate solutions for X_i and X_j are calculated based on the mutualistic symbiosis between organism X_i and X_j, which is modeled in Equations (10) and (11).

$$X_{inew} = X_i + \text{rand}(0,1) \times \left(X_{best}\text{-mutual vector} \times BF_1\right), \tag{10}$$

$$X_{jnew} = X_i + \text{rand}(0,1) \times \left(X_{best}\text{-mutual vector} \times BF_2\right), \tag{11}$$

$$\text{Mutual Vector} = \frac{X_i + X_j}{2} \tag{12}$$

rand(0,1) in Equations (10) and (11) is a vector of random numbers.

What follows explains the function of BF1 and BF2. In the natural world, certain mutualistic connections may benefit one organism more than another. In another context, interactions with organism B may be extremely advantageous for organism A. When interacting with organism A, organism B may only receive minimal or insignificant benefits. In this case, benefit factors (BF_1 and BF_2) are arbitrarily assigned to 1 or 2. These variables indicate the extent to which each organism benefits from the contact—that is, whether one organism gains all or some benefit from it.

The relationship feature between organisms X_i and X_j is represented by a vector named 'Mutual_Vector', as shown in Equation (12). The mutualistic effort to accomplish their objective of enhancing survival advantage is reflected in the (X_{best}-Mutual Vector × BF_1) component of the equation. Moreover, all species are compelled to increase their degree of adaptation to their ecosystem, based on Darwin's theory of evolution, which states that 'only the fittest organisms will prevail'. Some of them enhance their adaption to survival by forming symbiotic relationships with other organisms. Since X_{best} represents the maximum level of adaptation, it is required in this scenario. Consequently, we model the highest degree of adaptation as the objective point for the fitness increment of both organisms using (X_{best}/global solution). In the end, organisms are only updated if their current fitness exceeds their fitness before the interaction.

3.2. Commensalism Phase

A common example of the interaction between remora fish and sharks can be used to define commensalism. In such a scenario, the remora gains the advantage when it clings to the shark and consumes its remaining food. The behaviors of remora fish do not affect the shark, and their relationship offers very little benefit to it.

As in the mutualism phase, an organism denoted X_j is chosen at random from the environment to engage in interactions with X_i. In this situation, organism X_i makes an effort to gain something from the exchange. However, the relationship does not benefit or harm organism X_j itself. The commensal symbiosis between organisms X_i and X_j, which is described in Equation (13), is used to determine the new candidate solution of X_i. By the rules, organism X_i is modified only if its current fitness exceeds its fitness before the interaction.

$$X_{inew} = X_i + \text{rand}(-1, 1) \times X_{best} - X_j \tag{13}$$

The portion of the equation denoted by ($X_{best} - X_j$) reflects a positive advantage that X_j offers by helping X_i maximize its survival advantage within ecosystems in the current organism (represented by X_{best}).

3.3. Parasitism Phase

The Plasmodium parasite, which spreads between human hosts using its connection with the Anopheles mosquito, is a good example of parasitism. Whereas the parasites grow and replicate in the human body, their human host may suffer malaria and die as a result. By creating a synthetic parasite known as 'Parasite Vector', SOS gives organism X_i a function like that of the Anopheles mosquito. Using a random value to adjust the randomly chosen dimensions, organism X_i is duplicated in the search space to form a parasite vector. The host for the parasite vector is an organism denoted by X_j, which is chosen at random from the ecosystem. X_j is being replaced by the parasite vector in the ecosystem. Consequently, the fitness of both organisms is determined.

If the parasite vector has higher fitness values than organism X_j, it will be eliminated and its place in the ecosystem will be taken over. In other words, if the fitness value of X_j is higher, the parasite will not be able to survive in that ecosystem since X_j will be resistant to it.

3.4. Development of Initial Features

The values of selected features appear to be organized as an array. In optimization terminology, particle position corresponds to this array in particle swarm optimization (PSO), while genetic algorithms refer to it as a 'Chromosome'. As a result, the proposed approach labels each individual feature as a 'Raindrop' feature. In the problem selection of N_{var} dimensional features, a raindrop represents an array of $1 \times N_{var}$. Such an array is explained as follows:

$$\text{Feature of symbiotic} = [X_1, X_2, X_3 \ldots X_N] \tag{14}$$

At the beginning of the feature selection, a candidate representative of a matrix of size raindrops $N_{pop} \times N_{var}$ is created (i.e., features raindrops). Then, the matrix X is randomly created and provided as follows (columns and rows are the quantity of the variable of design and the quantity of unsupervised feature selections):

$$\text{Feature Ecosystem} = \begin{bmatrix} \text{eco}_1 \\ \text{eco}_2 \\ \text{eco}_3 \\ \vdots \\ \text{eco}_{\text{Neco\_size}} \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} x_1^1 x_2^1 x_3^1 & \cdots & x_{\text{Nfeature}}^1 \\ \vdots & \ddots & \vdots \\ x_1^{\text{Neco\_size}} x_2^{\text{Neco\_size}} x_3^{\text{Neco\_size}} & \cdots & x_{\text{Nfeature}}^{\text{Neco\_size}} \end{bmatrix}$$

Every value of the decision variable (X_1, X_2, $X_3 \ldots X_{Nvar}$) can be described as the following numbers (0 or 1), where N_{vars} and N_{pop} are the number of design variables and the number of raindrops (preliminary unsupervised features selection), respectively. Moreover, N_{pop} raindrops are generated, and subsequently, the raindrop cost is obtained by the assessment of the function of cost (*Cost*) as follows.

$$\text{Cost}_i = \text{f}\left(x_1^i, x_2^i, \ldots x_{\text{Nfeature}}^i\right)\ i = 1,\ 2,\ 3, \ldots,\ \text{Neco\_size}. \tag{16}$$

3.5. Cost of Solutions

As previously established, each row in eco is associated with many features in the document. In the context of eco a row's set of features is represented is denoted by f = (f_1, $f_2 \ldots f_k$). The objective for each row in eco is to assess the mean absolute difference (MAD), as detailed in [61]. MAD is used to determine the most relevant features for text classification by correlating the scores with the importance of each feature. The aim is to assign scores that accurately reflect the significance of each feature. One way to obtain such a score is to take the difference between the mean values and the sample. It can be depicted as per the following equation:

$$MAD_i = \frac{1}{n_i} \sum\nolimits_{j=1}^{n} \left| X_{ij} - \overline{X_i} \right| \tag{17}$$

where X_{ij} is the value of feature i in accordance with the document j and X_i is the mean of the feature i, which is computed according to the equation as follows:

$$\overline{X_i} = \left(\frac{1}{n}\right) \sum\nolimits_{j=1}^{n} X_{ij} \tag{18}$$

Every element in the solution indicates the cluster number as C = (c_1, $c_2 \ldots c_k$), and the computation of each solution in eco corresponds to a document cluster. The set of K centroids that correspond to a row in eco is represented by the C. The centroid of the kth cluster can be calculated as follows: $c_k = (c_{k1} \ldots,)$.

$$c_{kj} = \frac{\sum_{i=1}^{n} a_{ki} d_{ij}}{\sum_{i=1}^{n} a_{ki}} \tag{19}$$

The goal is to verify that by minimizing distances within and between clusters, cluster centroids optimize similarity both within and between clusters. The row corresponds to the average distance of documents from the cluster centroid and the associated fitness value. Following this, a suitable solution is derived based on this information. The condition is commonly known as attention deficit disorder with hyperactivity (ADDC).

$$Cos_i = \left[\sum\nolimits_{i=1}^{K} \frac{1}{ni} \sum\nolimits_{j=1}^{n} D(Ccent,\ dj)\right] / K \tag{20}$$

The cosine similarity is denoted by the D (.,.), where d_{ij} is the jth document in cluster i, K represents the number of clusters, and n_i is the number of documents contained in cluster i (e.g., ($n_i = \sum_{j=1}^{n} a_{ij}$)). If the cost value of the locally optimized vector is higher than that of the eco solutions, the newly solution generated can be replaced within a row in eco.

4. Experimental Settings

4.1. Parameter Setting of Symbiotic Organisms Search as Unsupervised Feature Selection

This section examines how the solutions of the algorithms evolved over generations under various configurations of just one important parameter. This is eco, where eco is the organism count (the initial feature population). In this case, this section clarifies the effects of changing certain parameters and specifically looks at three different scenarios, as shown in Table 3. Furthermore, utilizing the internal evaluation known as MAD, the experimental investigation demonstrated that the best results were obtained by a clear relationship between ecoNeco_size and the number of features. We looked at every situation and determined that the maximum repetition count for each run should be 100. Section 3.5 discusses the use of MAD to determine the fitness function value, which is the solution cost value. Additionally, the unsupervised symbiotic organism search algorithm used for the assessment is based on the feature selection covered in Section 2.2, at $d_{max} = 1 \times 10^3$. The scenario with ecoNeco_size = 40 is the best one, related to the fifth.

Table 3. Some scenarios of the parameters' symbiotic organisms search as the feature selection.

Scenarios	eco_{Neco_size}
1	8
2	16
3	24
4	32
5	40
6	48
7	56
8	64
9	72

4.2. Investigating the Impact of Different SOS Parameters on Cluster SOS

The aim of this section is to examine how the algorithms' solutions have evolved over generations under various configurations for a single parameter. This is ecoNeco_size, where ecoNeco_size is the quantity of documents and the number of groups, both of which are user-specified.

Given these conditions, this section focuses on highlighting the effect of a single parameter changes. Specifically, the following three scenarios have been tested and are displayed in Table 4. Empirical research has also demonstrated that the best results can be obtained with a linear relationship between ecoNeco_size and the number of clusters. All scenarios were tested over ten runs, with a maximum number of iterations fixed at 100 for all runs. The fitness function value is the ADDC value of the solution. The SOS algorithm, which is covered in Section 5, is the evaluation algorithm. $d_{max} = 1 \times 10^3$ was obtained using the Routers dataset.

The evolution of the solution for various SOS parameter values as clusters shows that when ecoNeco_size is decreased, the solution is found more slowly than when ecoNeco_size is increased, which causes the solution to be found faster than when ecoNeco_size is small. Nevertheless, with the benefits of reducing the required amount of space and converging to the optimal result, using the appropriate ecoNeco_size seems to be a reasonable and logical choice. In addition, doubling the number of clusters (8 × 2 in this dataset) can yield the best results with a specific selection of ecoNeco_size.

Table 4. Scenarios for analysis of SOS convergence behavior.

Scenario	Npop
1	2
2	4
3	6
4	8
5	16
6	18
7	20
8	24
9	28

Performance Measurement and Datasets

The universal F-measure from [63] was used in this study to evaluate the external condition. The F-measure, which takes into account recall and accuracy of information retrieval, is high if clusters are ideal. Each class is expected to have its own set of essential documents, and each class will continue to have its required document compilation. Higher F-measure values indicate better clustering performance, with a metric comprising a range of 0 to 1. Equation (21) shows the mathematical expression of F-measures for cluster *j* and class *i*:

$$\mathrm{F}(i,j) = \frac{2\mathrm{Recall}(i,j) \times \mathrm{Precision}(i,j)}{\mathrm{Recall}(i,j) + \mathrm{Precision}(i,j)}. \tag{21}$$

Equation (22) is used to determine the overall value for the F-measure, which is taken as in the weighted average of all.

$$\mathrm{F} = \sum\nolimits_i \frac{n_i}{N} maxF(i,j) \tag{22}$$

Thus, the F-measure values are distributed across the interval (0,1), and the values are proportional to higher levels of clustering quality.

This study used four independent and separate datasets to conduct a thorough evaluation of algorithm performance. This approach guaranteed an exhaustive and objective evaluation and comparison of the algorithms. The main dataset, known as classic 3, was used as a benchmark for comparison during the text-mining process. There are 3892 documents in the collection, which have been classified into three groups. In particular, there are 1460 documents centered on information retrieval (CISI), 1033 documents referring to medical problems (MEDs), and 1399 documents related to aviation systems (CRANs) [64].

The second dataset was made up of 1445 CNN news articles that were selected from the TDT2 and TDT3 corpora. Replicating the dataset, the i-Event experiment uses information from the TDT2 and TDT3 corpora [65]. A significant number of chosen documents are usually required to experiment and create clusters on the user interface. The concision of CNN's reporting and the significance of the events they covered also had a role in the choice of these sources.

The 20 newsgroups data [66] are the data used in this study and include 10,000 messages in total. Each Usenet newsgroup had 1000 messages, and these messages were collected from ten different newsgroups. The current study utilizes this dataset as its third dataset, which had 3831 documents in total after pre-processing. Thus, to assess how effectively algorithms handle large-scale datasets, a dataset with 20 newsgroups was utilized.

Another popular dataset used extensively in earlier academic studies is Reuters-21578 [67], which is a test set for the classification of text. However, there are several limitations related to the procedure of data collections in this setting. Many documents belong to more than one class, and most of the papers do not have labels for the class annotations. Moreover, the dataset distribution shows consistency across different groups. Some classes, like 'earn' and 'acquisition', have a huge number of documents, while other classes, like 'reserve' and 'veg-oil', have very few documents. To overcome these constraints, this study used a data with eight main groups and 1100 documents in each group. The summary description of document set is given in Table 5 with number of documents and number of clusters respectively.

Table 5. Summary description of document set.

Document	Source	#of Document	#of Cluster
DS1	Classic 3	3892	3
DS2	TDT2 and TDT3 of TREC 2001	1445	53
DS3	20 NEWSGROUP	3831	10
DS4	routers	4195	8

# represent the number of documents and clusters in each category.

5. Results and Discussion

5.1. Evaluation of the SOS Cluster Using All Features

In this section, we evaluate different algorithms, including harmony search as clustering (HSCLUST), k-means, one-step k-means with harmony search (KHS), and SOS, using standard datasets. For all these algorithms, the similarity metric was the cosine correlation measure. Notably, the results presented in this section represent the average performance over 20 runs (to ensure unbiased comparisons). Additionally, each algorithm is executed with 1000 iterations per run. It is worth noting that no specific parameters need to be set up for the k-means algorithm. For SOS, the ecoNeco_size is set to twice the number of classes in the dataset. This paper adapts the same settings as [61] Bsoul et al. (2021) for the other HSCLUST algorithm. Specifically, the authors set the HMS to be twice the number of clusters in the dataset, and PARmax was set to 0.9, PARmin was 0.45, and HMCR was set to 0.6.

In Table 6, the algorithmic performances in the document collections are evaluated based on the F-measure. Among the different algorithms, HS + k-means stands out with the highest F-measure. On the other hand, HS as clustering performs poorly. Interestingly, the proposed SOS algorithm is comparable to the k-means algorithm. Specifically, in four datasets, SOS outperforms k-means. In terms of clustering, SOS outperforms HS when compared to the SOS algorithm. As a result, SOS outperforms HS in locating the optimum centroids, while k-means is not very good at locating the global optimal initial centroids. In other words, the SOS is not as powerful in local optima as k-means. Building upon this observation, the next section explores the combination of SOS-based unsupervised feature selection and SOS in clustering. The goal in the subsequent sub-section is to leverage SOS's strength in finding global optima most important features.

Table 6. The result of three cluster algorithms using the external evaluation F-measure.

Datasets	k-Means	HS	SOS	KHS
Classic 3	0.929	0.909	**0.933**	<u>**0.935**</u>
TREC 2001	0.804	0.829	**0.831**	<u>**0.853**</u>
Newsgroup	0.582	0.611	**0.649**	<u>**0.659**</u>
Routers	0.636	0.682	**0.702**	<u>**0.712**</u>

Best result: underline and bold; second-best result: bold.

Evaluation of SOS-Based Unsupervised Feature Selection with SOS Cluster

As shown in Table 7, the summary comprises the number of features derived from the k-means and bag-of-words (BOW) models, as well as the results of the SOSFS and the SOSC algorithms for cluster performance evaluation. Based on our findings, DS1 had the best F-measure (92.9%) obtained by using the k-means clustering algorithm, whereas DS3 had the lowest F-measure (58.2%). For dataset DS1, the optimization of the PSOC method yielded the highest F-measure of 89.1%. In contrast, for dataset DS3, PSOC resulted in the lowest F-measure of 60.6%. When utilizing the HSC method, dataset DS1 achieved the highest F-measure of 90.9%, whereas dataset DS3 had the lowest F-measure of 61.4%. For the SOSC technique, dataset DS1 produced the highest F-measure of 93.3%, while dataset DS3 recorded the lowest F-measure of 64.9%. Finally, the KHCluster method achieved the maximum F-measure of 93.5% for dataset DS1, but for dataset DS3, KHCluster resulted in the lowest F-measure of 65.9%.

Table 7. The f-measurement of the cluster using all features.

Comparison	DS1 Classic 3		DS2 TREC 2001		DS3 Newsgroup		DS4 Routers	
	Features	**F-Measure**	**Features**	**F-Measure**	**Features**	**F-Measure**	**Features**	**F-Measure**
k-means	13,310	0.929	6737	0.804	27,211	0.582	12,152	0.636
PSOC	13,310	0.891	6737	0.841	27,211	0.606	12,152	0.688
HSC	13,310	0.909	6737	0.829	27,211	0.611	12,152	0.682
WDOC	13,310	**0.933**	6737	**0.83**	27,211	**0.64**	12,152	**0.69**
KHCluster	13,310	**0.935**	6737	**0.853**	27,211	**0.659**	12,152	**0.712**

Best result: underline and bold; second-best result: bold.

Table 8 illustrates the performance of the SOSFS algorithm with various clustering techniques:

Table 8. The f-measurement of the proposed SOS-based unsupervised feature selection.

Comparison	DS1 Classic 3		DS2 TREC 2001		DS3 Newsgroup		DS4 Routers	
	Features	**F-Measure**	**Features**	**F-Measure**	**Features**	**F-Measure**	**Features**	**F-Measure**
k-means	**8186**	0.93	5573	0.824	20,854	0.602	9561	0.636
PSOC	9927	0.928	**4826**	0.847	**13,824**	0.636	**5084**	0.688
HSC	10,843	0.929	**5128**	0.831	**11,843**	0.621	10,854	0.682
WDOC	**9824**	**0.939**	5834	**0.83**	19,283	**0.64**	6891	**0.69**
KHCluster	10,289	**0.949**	6057	**0.861**	20,851	**0.668**	**5732**	**0.748**

Best result: underline and bold; second-best result: bold.

For dataset DS1, the integration of SOSFS with k-means achieved the highest F-measure of 93%. Conversely, for dataset DS3, the same integration produced the lowest F-measure of 60.2%. The SOSFS algorithm effectively reduced the number of features in DS1 by 8186 out of 13,310, in DS2 by 5573 out of 6737, in DS3 by 20,854 out of 27,211, and in DS4 by 9561 out of 1215.When SOSFS was combined with PSO, the lowest F-measure of 63.6% was observed for dataset DS3. The feature reduction for SOSFS with PSO was significant: 9927 out of 13,310 features in DS1, 4826 out of 6737 in DS2, 13,824 out of 27,211 in DS3, and 5084 out of 12,152 in DS4.With SOSFS and HSC, dataset DS3 had the lowest F-measure of 62.1%. Feature reduction was also substantial: 10,843 out of 13,310 in DS1, 5128 out of 6737 in DS2, 11,843 out of 27,211 in DS3, and 108,54 out of 12,152 in DS4.

When the SOSFS algorithm was combined with WDOC, dataset DS3 yielded the lowest F-measure of 65%. In DS1, DS2, DS3, and DS4, SOSFS effectively decreased the number of features by 9824 out of 13,310, 5834 out of 6737, 19,283 out of 27,211, and 6891 out of 12,152. The lowest F-measure of 66.8% was achieved for dataset DS3 when the SOSFS algorithm was combined with the KHCluster. In DS1, DS2, DS3, and DS4, SOSFS effectively decreased the number of features by 10,289 out of 13,310, 6057 out of 6737, 20,851 out of 27,211, and 5732 out of 12,152. For SOSUF reduction in DS1, k-means was the best. In DS2, SOSUFS was best in PSOC; in DS3, it was best with HSC; and in DS4, it was best with PSOC. Comparing Tables 7 and 8, the KHCluster showed the best performance and was more powerful than used all the features. From Tables 7 and 8, we can observe that the SOSUFS proved that some of the features mis-cluster and reduce the performance of cluster algorithms. In all cluster algorithms, the SOSUFS enhances the performance when looking to the F-measure.

As seen in Table 9, when combined with SOSC, the SOSFS approach produced the highest F-measure for dataset DS1: 95.3%. In DS1, DS2 and DS3, 10,346 and 12,152 features, 7096 and 4731 features, and 5651 and 12,152 features, respectively, were successfully reduced by SOSFS. The proposed symbiotic organisms search-based optimal unsupervised feature selection (SOSUFS) in conjunction with symbiotic organism search-based optimal clustering (SOSC) was found to be the best text clustering strategy. Following the evaluation, the SOSFS algorithm combined with the k-means algorithm demonstrated a relatively strong performance. The SOSC and particle swarm optimization clustering (PSOC) algorithms also showed moderate performance. In contrast, the KHCluster method ranked third in terms of performance. Out of all the approaches that were assessed, the k-means algorithm performed the worst. Text clustering performance has been demonstrated to be enhanced by using the UFS with the k-means algorithm.

Table 9. The f-measurement of the proposed hybrid multi-objective clustering.

Comparison	DS1 Classic 3		DS2 TREC 2001		DS3 Newsgroup		DS4 Routers	
	Features	F-Measure	Features	F-Measure	Features	F-Measure	Features	F-Measure
k-means	**8186**	0.93	5573	0.824	20,854	0.602	9561	0.636
PSOC	9927	0.928	**4826**	0.847	13,824	0.636	**5084**	0.688
HSC	10,843	0.929	5128	0.831	**11,843**	0.621	10,854	0.682
WDOC	9824	0.939	5834	0.83	19,283	0.64	6891	0.69
KHCluster	10,289	**0.949**	6057	**0.861**	20,851	**0.668**	5732	**0.748**
SOSFS with SOSC	**7096**	**0.953**	**4731**	**0.865**	**10,346**	**0.686**	**5651**	**0.734**

Best result: underline and bold; second-best result: bold.

A statistical evaluation of the proposed hybridized SOS multi-objective methods for clustering, unsupervised feature selection was conducted. The optimal performance of text clustering can be assessed, as can whether significant differences can be obtained, using this multi-objective method for clustering and unsupervised feature selection. Based on the Friedman criterion, Table 10 shows the ranks of the multi-objectives proposed for the clustering, unsupervised feature selection and other methods. The criteria establish the rating, with a lower value signifying a higher rank.

Table 10. The ranking of the proposed algorithms using the Friedman test.

Algorithms	Ranking
k-means	10.18
PSOC	10.15
HSC	10.01
WDOC	9.61
SOSC	9.39
KHCluster	9.3
SOSC and SOSFS	**8.53**
Friedman test (*p*-value)	0.00
man-Davenport (*p*-value)	0.00

The best is in bold font.

Table 10 demonstrates that our unsupervised feature selection and grouping technique, which employs SOS, had the lowest value and was ranked highest. The last two rows of Table 10 display the Friedman and Iman–Davenport *p*-values. Hybridize SOS in unsupervised feature selection with SOSC, KHCluster, SOSCe, WDOC, HSC, PSOC, and k-means is in the second, third, fourth, fifth, sixth, and seventh ranks, respectively.

5.2. Discussion

This study focused on the problems of finding a near-optimal partition cluster concerning the ADDC criterion for a given set of texts, to split them into a specified number of clusters and find near-optimal features. To this point, this work studied each of the existing clustering methods and detected the behavior of k-means, HSC, KHSC, and PSO. Although each algorithm exhibits considerable performance, some of the weaknesses of each algorithm are found and discussed. The k-means algorithm, for instance, performed depending on the initial centroid selected. However, the memory requirements for large datasets are its major limitation. In addition, the k-means algorithm is also found to perform better than the PSO algorithm. Given the limitations of existing clustering methods, metaheuristics approaches, such as harmony search clustering, filled the gaps. The HS method shows an amazing performance compared to other approaches based on the ADDC evaluation, whereas HSCLUST was found to have the worst performance compared with other traditional clusters. The proposed algorithms using SOS in unsupervised feature selection and in clustering in this article aimed to address the limitations of the traditional methods by demonstrating the partitioning problems as optimization issues. Primarily, the symbiotic organism search algorithm, the SOS, was employed as an unsupervised feature-selection and cluster method to optimize the objective functions related with the optimal clustering method. The effect of the ecoNeco_size parameter was verified, and the experimental works show that with linear relations between ecoNeco_size and the number of features and clusters, better results would be possible. As shown from the results, ecoNeco_size has a better performance with two times the number of classes in the dataset. The first experimental findings revealed that the SOS method has equivalent performance to the k-means algorithm but performed better than the HS method for the clustering task. Based on the experimental analysis of benchmark datasets, the result shows that the hybridization of SOS-based unsupervised feature selection with SOS in clustering was better than other hybrid approaches, due to the effective nature of the SOS, which focuses on the global optima features and centroids. However, the results show that the performance was better achieved by combining the SOS-based unsupervised feature-selection method with an SOS-based cluster method compared to KHS, SOS, k-means, and HS, respectively.

6. Conclusions

This study investigates a multi-objective symbiotic organism search algorithm for unsupervised feature selection that simultaneously takes into account the number of selected features and the accuracy of the clustering. The effectiveness of the proposed approach is examined using reputable benchmark datasets. Based on the results, the proposed feature-selection model offered a feature subset of high quality for feature selection, and the final model outperformed the other techniques in terms of clustering accuracy for both datasets. The fact that the proposed feature-selection technique produced satisfactory results for both datasets indicates that it may be able to provide a potential solution to this problem. In addition, the robustness of the model across datasets is also demonstrated by this finding. However, in comparison to the traditional unsupervised feature-selection algorithms, the computing cost of the proposed algorithm is significant. Although unsupervised feature selection is typically an offline procedure, this is not a total disadvantage. Additionally, hybrid techniques are presented in this paper and applied to all datasets for unsupervised feature selection. These algorithms were shown to have both strengths and limitations. Out of all the algorithms discussed in this study, the combination of SOSFS and SOSC produced the best clustering accuracy results while reducing features that are not relevant. This approach presents an optimal solution set instead of a single solution and demonstrates the significance of handling feature-selection problems as a multi-objective process. As a result, this study concludes that the proposed SOSFS is a realistic and efficient method for solving unsupervised feature-selection problems. Additionally, the proposed SOSC model outperformed the other algorithms for both datasets in terms of clustering accuracy. Furthermore, various datasets for feature selection and clustering may also be used to test the proposed hybrid approach. In addition, future research needs to examine the nature of features chosen by the SOSFS algorithm when combine with k-means and SOSC, which have not yet been evaluated.

Author Contributions: Conceptualization, A.F.J.A.-G., M.Z.A.N. and Z.A.A.A.; methodology, A.F.J.A.-G., M.Z.A.N. and Z.A.A.A.; software, A.F.J.A.-G.; validation, A.F.J.A.-G., M.Z.A.N. and M.R.B.Y.; formal analysis, A.F.J.A.-G., M.Z.A.N. and M.R.B.Y.; investigation, A.F.J.A.-G. and M.Z.A.N.; resources, A.F.J.A.-G., M.Z.A.N. and Z.A.A.A.; data curation, A.F.J.A.-G.; writing—original draft preparation, A.F.J.A.-G., M.Z.A.N. and M.R.B.Y.; writing—review and editing, A.F.J.A.-G., M.Z.A.N. and Z.A.A.A.; visualization, A.F.J.A.-G.; supervision, M.Z.A.N. and M.R.B.Y.; project administration, Z.A.A.A.; funding acquisition, A.F.J.A.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Malaysia Ministry of Higher Education and the Universiti Kebangsaan Malaysia under the Q2678Q25 FRGS/1/2022/ICT02/UKM/02/3 with machines and materials.

Data Availability Statement: The dataset used in this work based on [64,65].

Acknowledgments: We thank the anonymous reviewers for their comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Oyewole, G.J.; Thopil, G.A. Data clustering: Application and trends. *Artif. Intell. Rev.* **2022**, *56*, 6439–6475. [CrossRef] [PubMed]
2. Gedam, A.G.; Shikalpure, S.G. Direct kernel method for machine learning with support vector machine. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, India, 6–7 July 2017; pp. 1772–1775.
3. da Silva, L.E.B.; Wunsch, D.C. An Information-Theoretic-Cluster Visualization for Self-Organizing Maps. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 2595–2613. [CrossRef] [PubMed]
4. Sinaga, K.P.; Yang, M.-S. Unsupervised K-Means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
5. Wang, P.; Xue, B.; Liang, J.; Zhang, M. Feature clustering-Assisted feature selection with differential evolution. *Pattern Recognit.* **2023**, *140*, 109523. [CrossRef]
6. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [CrossRef]
7. Jiao, L.; Liu, Y.; Zou, B. Self-organizing dual clustering considering spatial analysis and hybrid distance measures. *Sci. China Earth Sci.* **2011**, *54*, 1268–1278. [CrossRef]
8. Chakraborty, B.; Chakraborty, G. Fuzzy Consistency Measure with Particle Swarm Optimization for Feature Selection. In Proceedings of the 2013 IEEE International Conference on Systems, Man and Cybernetics (SMC 2013), Manchester, UK, 13–16 October 2013; pp. 4311–4315.
9. Li, G.; Li, Y.; Tsai, C.-L. Quantile Correlations and Quantile Autoregressive Modeling. *J. Am. Stat. Assoc.* **2015**, *110*, 246–261. [CrossRef]
10. Pardo, L. New Developments in Statistical Information Theory Based on Entropy and Divergence Measures. *Entropy* **2019**, *21*, 391. [CrossRef]
11. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **2015**, *20*, 606–626. [CrossRef]
12. Liu, Q.; Chen, C.; Zhang, Y.; Hu, Z. Feature selection for support vector machines with RBF kernel. *Artif. Intell. Rev.* **2011**, *36*, 99–115. [CrossRef]
13. Rong, M.; Gong, D.; Gao, X. Feature Selection and Its Use in Big Data: Challenges, Methods, and Trends. *IEEE Access* **2019**, *7*, 19709–19725. [CrossRef]
14. Abualigah, L.M.; Khader, A.T.; Al-Betar, M.A. Unsupervised feature selection technique based on genetic algorithm for improving the Text Clustering. In Proceedings of the 2016 7th International Conference on Computer Science and Information Technology (CSIT), Amman, Jordan, 13–14 July 2016; pp. 1–6.
15. Shamsinejadbabki, P.; Saraee, M. A new unsupervised feature selection method for text clustering based on genetic algorithms. *J. Intell. Inf. Syst.* **2011**, *38*, 669–684. [CrossRef]
16. Bennaceur, H.; Almutairy, M.; Alhussain, N. Genetic Algorithm Combined with the K-Means Algorithm: A Hybrid Technique for Unsupervised Feature Selection. *Intell. Autom. Soft Comput.* **2023**, *37*, 2687–2706. [CrossRef]
17. Zhang, Y.; Wang, S.; Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Math. Probl. Eng.* **2015**, *2015*, 931256. [CrossRef]
18. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]

19. Lalwani, S.; Sharma, H.; Satapathy, S.C.; Deep, K.; Bansal, J.C. A Survey on Parallel Particle Swarm Optimization Algorithms. *Arab. J. Sci. Eng.* **2019**, *44*, 2899–2923. [CrossRef]
20. Han, C.; Zhou, G.; Zhou, Y. Binary Symbiotic Organism Search Algorithm for Feature Selection and Analysis. *IEEE Access* **2019**, *7*, 166833–166859. [CrossRef]
21. Mohmmadzadeh, H.; Gharehchopogh, F.S. An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems. *J. Supercomput.* **2021**, *77*, 9102–9144. [CrossRef]
22. Cheng, M.-Y.; Prayogo, D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [CrossRef]
23. Abdullahi, M.; Ngadi, A.; Dishing, S.I.; Abdulhamid, S.M.; Ahmad, B.I. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment. *J. Netw. Comput. Appl.* **2019**, *133*, 60–74. [CrossRef]
24. Miao, F.; Zhou, Y.; Luo, Q. A modified symbiotic organisms search algorithm for unmanned combat aerial vehicle route planning problem. *J. Oper. Res. Soc.* **2018**, *70*, 21–52. [CrossRef]
25. Wu, H.; Zhou, Y.; Luo, Q. Hybrid symbiotic organisms search algorithm for solving 0–1 knapsack problem. *Int. J. Bio-Inspired Comput.* **2018**, *12*, 23–53. [CrossRef]
26. Baysal, Y.A.; Ketenci, S.; Altas, I.H.; Kayikcioglu, T. Multi-objective symbiotic organism search algorithm for optimal feature selection in brain computer interfaces. *Expert Syst. Appl.* **2020**, *165*, 113907. [CrossRef]
27. Gharehchopogh, F.S.; Shayanfar, H.; Gholizadeh, H. A comprehensive survey on symbiotic organisms search algorithms. *Artif. Intell. Rev.* **2019**, *53*, 2265–2312. [CrossRef]
28. Ganesh, N.; Shankar, R.; Čep, R.; Chakraborty, S.; Kalita, K. Efficient Feature Selection Using Weighted Superposition Attraction Optimization Algorithm. *Appl. Sci.* **2023**, *13*, 3223. [CrossRef]
29. Jaffel, Z.; Farah, M. A symbiotic organisms search algorithm for feature selection in satellite image classification. In Proceedings of the 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 21–24 March 2018; pp. 1–5.
30. Cheng, M.-Y.; Cao, M.-T.; Herianto, J.G. Symbiotic organisms search-optimized deep learning technique for mapping construction cash flow considering complexity of project. *Chaos Solitons Fractals* **2020**, *138*, 109869. [CrossRef]
31. Mohammadzadeh, H.; Gharehchopogh, F.S. Feature Selection with Binary Symbiotic Organisms Search Algorithm for Email Spam Detection. *Int. J. Inf. Technol. Decis. Mak.* **2021**, *20*, 469–515. [CrossRef]
32. Cheng, M.-Y.; Kusoemo, D.; Gosno, R.A. Text mining-based construction site accident classification using hybrid supervised machine learning. *Autom. Constr.* **2020**, *118*, 103265. [CrossRef]
33. Al-Tashi, Q.; Abdulkadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Approaches to Multi-Objective Feature Selection: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 125076–125096. [CrossRef]
34. Abdollahzadeh, B.; Gharehchopogh, F.S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **2021**, *38* (Suppl. S3), 1845–1863. [CrossRef]
35. Zhang, M.; Wang, J.-S.; Liu, Y.; Song, H.-M.; Hou, J.-N.; Wang, Y.-C.; Wang, M. Multi-objective optimization algorithm based on clustering guided binary equilibrium optimizer and NSGA-III to solve high-dimensional feature selection problem. *Inf. Sci.* **2023**, *648*, 119638. [CrossRef]
36. Al-Tashi, Q.; Abdulkadir, S.J.; Rais, H.M.; Mirjalili, S.; Alhussian, H.; Ragab, M.G.; Alqushaibi, A. Binary Multi-Objective Grey Wolf Optimizer for Feature Selection in Classification. *IEEE Access* **2020**, *8*, 106247–106263. [CrossRef]
37. Xue, B.; Fu, W.; Zhang, M. Differential evolution (DE) for multi-objective feature selection in classification. In Proceedings of the GECCO' 14: Genetic and Evolutionary Computation Conference, Dunedin, New Zealand, 15–18 December; pp. 83–84.
38. Vieira, S.M.; Sousa, J.M.C.; Runkler, T.A. Multi-criteria ant feature selection using fuzzy classifiers. In *Swarm Intelligence for Multi-objective Problems in Data Mining*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 19–36. [CrossRef]
39. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276. [CrossRef]
40. Abdullahi, M.; Ngadi, A.; Dishing, S.I.; Abdulhamid, S.M.; Usman, M.J. A survey of symbiotic organisms search algorithms and applications. *Neural Comput. Appl.* **2019**, *32*, 547–566. [CrossRef]
41. Ezugwu, A.E.; Adewumi, A.O. Soft sets based symbiotic organisms search algorithm for resource discovery in cloud computing environment. *Future Gener. Comput. Syst.* **2017**, *76*, 33–50. [CrossRef]
42. Ezugwu, A.E.-S.; Adewumi, A.O. Discrete symbiotic organisms search algorithm for travelling salesman problem. *Expert Syst. Appl.* **2017**, *87*, 70–78. [CrossRef]
43. Ezugwu, A.E.-S.; Adewumi, A.O.; Frîncu, M.E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. *Expert Syst. Appl.* **2017**, *77*, 189–210. [CrossRef]
44. Mohammadzadeh, H.; Gharehchopogh, F.S. A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection. *Int. J. Commun. Syst.* **2020**, *34*. [CrossRef]
45. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2018**, *116*, 147–160. [CrossRef]
46. Du, Z.-G.; Pan, J.-S.; Chu, S.-C.; Chiu, Y.-J. Improved Binary Symbiotic Organism Search Algorithm with Transfer Functions for Feature Selection. *IEEE Access* **2020**, *8*, 225730–225744. [CrossRef]

47. Miao, F.; Yao, L.; Zhao, X. Symbiotic organisms search algorithm using random walk and adaptive Cauchy mutation on the feature selection of sleep staging. *Expert Syst. Appl.* **2021**, *176*, 114887. [CrossRef]
48. Kimovski, D.; Ortega, J.; Ortiz, A.; Baños, R. Parallel alternatives for evolutionary multi-objective optimization in unsupervised feature selection. *Expert Syst. Appl.* **2015**, *42*, 4239–4252. [CrossRef]
49. Liao, T.; Kuo, R. Five discrete symbiotic organisms search algorithms for simultaneous optimization of feature subset and neighborhood size of KNN classification models. *Appl. Soft Comput.* **2018**, *64*, 581–595. [CrossRef]
50. Apolloni, J.; Leguizamón, G.; Alba, E. Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Appl. Soft Comput.* **2016**, *38*, 922–932. [CrossRef]
51. Zare-Noghabi, A.; Shabanzadeh, M.; Sangrody, H. Medium-Term Load Forecasting Using Support Vector Regression, Feature Selection, and Symbiotic Organism Search Optimization. In Proceedings of the 2019 IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA, 4–8 August 2019; pp. 1–5.
52. Gana, N.N.; Abdulhamid, S.M.; Misra, S.; Garg, L.; Ayeni, F.; Azeta, A. Optimization of Support Vector Machine for Classification of Spyware Using Symbiotic Organism Search for Features Selection. In *Lecture Notes in Networks and Systems*; Springer: Cham, Switzerland, 2022. [CrossRef]
53. Zhou, Y.; Wu, H.; Luo, Q.; Abdel-Baset, M. Automatic data clustering using nature-inspired symbiotic organism search algorithm. *Knowl.-Based Syst.* **2019**, *163*, 546–557. [CrossRef]
54. Yang, C.-L.; Sutrisno, H. A clustering-based symbiotic organisms search algorithm for high-dimensional optimization problems. *Appl. Soft Comput.* **2020**, *97*, 106722. [CrossRef]
55. Zhang, B.; Sun, L.; Yuan, H.; Lv, J.; Ma, Z. An improved regularized extreme learning machine based on symbiotic organisms search. In Proceedings of the 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), Hefei, China, 5–7 June 2016; pp. 1645–1648.
56. Ikotun, A.M.; Ezugwu, A.E. Boosting k-means clustering with symbiotic organisms search for automatic clustering problems. *PLoS ONE* **2022**, *17*, e0272861. [CrossRef] [PubMed]
57. Acharya, D.S.; Mishra, S.K. A multi-agent based symbiotic organisms search algorithm for tuning fractional order PID controller. *Measurement* **2020**, *155*, 107559. [CrossRef]
58. Rajah, V.; Ezugwu, A.E. Hybrid Symbiotic Organism Search algorithms for Automatic Data Clustering. In Proceedings of the 2020 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 11–12 March 2020; pp. 1–9.
59. Chakraborty, S.; Nama, S.; Saha, A.K. An improved symbiotic organisms search algorithm for higher dimensional optimization problems. *Knowl.-Based Syst.* **2021**, *236*, 107779. [CrossRef]
60. Sherin, B.M.; Supriya, M.H. SOS based selection and parameter optimization for underwater target classification. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–4.
61. Bsoul, Q.; Salam, R.A.; Atwan, J.; Jawarneh, M. Arabic Text Clustering Methods and Suggested Solutions for Theme-Based Quran Clustering: Analysis of Literature. *J. Inf. Sci. Theory Pract.* **2021**, *9*, 15–34. [CrossRef]
62. Mehdi, S.; Smith, Z.; Herron, L.; Zou, Z.; Tiwary, P. Enhanced Sampling with Machine Learning. *Annu. Rev. Phys. Chem.* **2024**, *75*, 347–370. [CrossRef] [PubMed]
63. Larsen, B.; Aone, C. Fast and effective text mining using linear-time document clustering. In Proceedings of the KDD99: The First Annual International Conference on Knowledge Discovery in Data, San Diego, CA, USA, 15–18 August 1999; pp. 16–22.
64. Sanderson, M. Test Collection Based Evaluation of Information Retrieval Systems. *Found. Trends Inf. Retr.* **2010**, *4*, 247–375. [CrossRef]
65. Mohd, M.; Crestani, F.; Ruthven, I. Evaluation of an interactive topic detection and tracking interface. *J. Inf. Sci.* **2012**, *38*, 383–398. [CrossRef]
66. Zobeidi, S.; Naderan, M.; Alavi, S.E. Effective text classification using multi-level fuzzy neural network. In Proceedings of the 2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), Qazvin, Iran, 7–9 March 2017; pp. 91–96.
67. Lewis, D.D.; Yang, Y.; Rose, T.G.; Li, F. RCV1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.* **2004**, *5*, 361–397.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.