

Time-Varying Hierarchical Chains of Salps with Random Weight Networks for Feature Selection ¹

Hossam Faris^a, Ali Asghar Heidari^b, Ala' M. Al-Zoubi^a, Majdi Mafarja^c, Ibrahim Aljarah^a, Mohammed Eshtay^a, Seyedali Mirjalili^d

^aKing Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
{hossam.faris,i.aljarah}@ju.edu.jo, alaah14@gmail.com, m.eshtay@fqs.ju.edu.jo

^bSchool of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran
as_heidari@ut.ac.ir

^cDepartment of Computer Science, Birzeit University, Birzeit, Palestine
mmafarja@birzeit.edu, mmafarjeh@gmail.com

^dInstitute of Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia
seyedal.mirjalili@griffithuni.edu.au

Abstract

Feature selection (FS) is considered as one of the most common and challenging tasks in Machine Learning. FS can be considered as an optimization problem that requires an efficient optimization algorithm to find its optimal set of features. This paper proposes a wrapper FS method that combines a time-varying number of leaders and followers binary Salp Swarm Algorithm (called TVBSSA) with Random Weight Network (RWN). In this approach, the TVBSSA is used as a search strategy, while RWN is utilized as an induction algorithm. The objective function is formulated in a manner to aggregate three objectives: maximizing the classification accuracy, maximizing the reduction rate of the selected features, and minimizing the complexity of generated RWN models. To assess the performance of the proposed approach, 20 well-known UCI datasets and a number of existing FS methods are employed. The comparative results show the ability of the proposed approach in outperforming similar algorithms in the literature and its merits to be used in systems that require FS.

Keywords: Feature Selection, Salp Swarm Algorithm, Optimization, Evolutionary Algorithms

1. Introduction

1 Data dimensionality has a great impact on the performance of data mining and machine
2 learning tasks. With the advanced data collection tools, a huge amount of data becomes
3 available in many fields such as pattern recognition, image processing, and disease diagnosis
4 systems. Dealing with such data requires massive time and space resources. Also, not all
5 features included in those datasets have the same degree of importance for the field of study.
6 Some of those features can be excluded without affecting the amount of information that can

¹Find the published version here: <https://www.sciencedirect.com/science/article/pii/S0957417419306141>

7 be extracted from the original datasets since they may be redundant or irrelevant. Feature
8 selection (FS) has proved to be an effective tool for tackling this problem (Jain and Zongker,
9 1997; Guyon and Elisseeff, 2003; Liu and Motoda, 2012).

10 FS process tries to select the most informative features that can represent the whole
11 feature set without information loss. FS methods can be classified to three different categories
12 according to the degree of dependency on the class label in the dataset: supervised methods
13 (correspond to labeled datasets) (Nie et al., 2008; Faris et al., 2018), semi-supervised methods
14 (correspond to partially labeled datasets) (Han et al., 2015), and unsupervised methods
15 (correspond to unlabeled datasets) (Nie et al., 2016).

16 According to the involvement level of a learning algorithm in evaluating a feature subset,
17 FS methods can be divided into filters, wrappers, and embedded methods (Zhao et al., 2010).
18 In the filter methods, learning algorithms are not used, and features are ranked based on
19 the internal relations between the data itself. Filter methods are known as fast methods,
20 but the selection of features is not based on the performance of a specific learning task such
21 as classification. Typical filter models include information gain (IG) (Quinlan, 1986), Chi-
22 Square (Liu and Setiono, 1995), Gain Ratio (Quinlan, 1993), and ReliefF (Robnik-ikonja
23 and Kononenko, 2003).

24 As opposed to the filter methods, wrappers determine a predictor, and the feature subset
25 is selected if it enhances the performance of that predictor (e.g., classification accuracy)
26 (Kohavi and John, 1997; Oh et al., 2004). Since the wrapper approaches tend to evaluate each
27 feature using a specific learning algorithm (e.g., classifier), they can optimize the objective
28 (e.g., classification accuracy), but the optimization process is usually slow. The embedded
29 methods involve FS into the training process, where a trade-off between learning speed and
30 model performance is desirable (Tibshirani, 1996; Nie et al., 2010).

31 Besides the evaluation method in wrapper models (e.g., classifier), searching for the best
32 feature subset is a challenging problem in FS methods. FS is known to be as an NP-hard
33 problem with an exponential growth where 2^N solutions have to be handled when dealing
34 with a dataset with N features. Three search strategies can be mainly utilized with FS
35 methods:

- 36 • Complete (brute-force) search that tends to generate all possible solutions to select the
37 best one
- 38 • Random searches that select the subsets randomly in the hope to find the best one
- 39 • Heuristic approaches that use the heuristic information to guide the random search
40 process.

41 Using complete and random methods with FS is impractical with medium and large-scale
42 datasets and a random search becomes a complete search in the worst case. Heuristic search
43 methods provide general ways that tradeoff between local search and global search to find
44 good solutions (not necessarily the best) within a reasonable running time (Talbi, 2009).

45 Nature-inspired metaheuristic algorithms are regarded as powerful algorithms with su-
46 perior performances in treating different optimization scenarios including machine learn-
47 ing and FS (Shrivastava et al., 2017). In previous works, different categories of heuristics
48 such as evolutionary methods (e.g., genetic algorithms (GA) (Holland, 1992)) and swarm

49 intelligence-based techniques (e.g., particle swarm optimization (PSO) (Kennedy and Eber-
50 hart, 1997)) have been applied to the FS problem (Mafarja and Sabar, 2018; Mafarja et al.,
51 2018c). Swarm intelligence (SI) is mainly inspired by the intelligent, collective behavior of
52 decentralized and self-organized swarms. SI methods include but not limited to Ant Colony
53 Optimization (ACO) (Dorigo et al., 2006), Grey Wolf Optimizer (GWO) (Mirjalili et al.,
54 2014), Dragonfly Algorithm (DA) (Mirjalili, 2016), Water Cycle Algorithm (WCA) (Heidari
55 et al., 2017), Krill Herd (KH) (Gandomi and Alavi, 2012) algorithm, Whale Optimization
56 Algorithm (WOA) (Mirjalili and Lewis, 2016; Al-Zoubi et al., 2018), and Firefly Algorithm
57 (FA) (Yang, 2009). Salp Swarm Algorithm (SSA) (Mirjalili et al., 2017) is a well-established,
58 recent SI algorithm that mimics the collective foraging behavior of salps in oceans and seas.
59 The SSA showed its efficiency in realizing the optimal solutions of many optimization prob-
60 lems including FS (Faris et al., 2018; Sayed et al., 2018) and parameters tuning of PEM fuel
61 cells (El-Fergany, 2018).

62 With a large number of algorithms published, one can ask if there is a need to propose
63 new methods to solve FS problem. Referring to No-Free-Lunch (NFL) theorem Wolpert and
64 Macready (1997), there is no universally best method that can solve all possible classes of
65 FS problems. Therefore, the door is still open for proposing new optimization methods or
66 improving the existing ones for tackling FS problems more efficiently.

67 In addition, the majority of the previous works in the area of unsupervised, wrapper based
68 approaches have focused on utilizing k-NN as an induction algorithm due to its simplicity
69 and speed. This is to reduce the computational overhead of the wrapper. However, the
70 simplicity of the k-NN comes at the cost of the prediction power. It is well-shown that k-NN
71 is not competitive in terms of prediction accuracy with more advanced algorithms in machine
72 learning like neural networks and support vector machines.

73 The above-mentioned gaps at the search and induction levels in feature selection moti-
74 vated our attempts to propose an improved version of a recent swarm intelligent algorithm
75 called Salp Swarm Algorithm (SSA) in combination with Random Weight Networks (RWN)
76 as a wrapper based approach for feature selection. SSA was recently proposed by Mirjalili
77 et al. (2017). SSA mimics the swimming behavior as a group in a form of chain in oceans.
78 The algorithm with its model of salp chains has shown promising results in a wide range
79 of challenging real-world complex problems which motivated us to utilize it as the search
80 level for feature selection. On the other side, RWN benefits from several advantages such as
81 the rapid training process, high generalization performance, and the need for little human
82 intervention and involvement. These advantages motivated us to investigate its efficiency
83 as an induction algorithm in the wrapper to be a good candidate to replace the classical
84 k-NN algorithm. As a potential drawback, however, there is no rule of thumb to set the
85 number of hidden nodes in its structure. This is perhaps one of the reasons that prevent
86 other researchers from utilizing RWN as an induction algorithm.

87 The contributions in the proposed wrapper-based approach can be summarized as follows:

- 88 • A time-varying hierarchal based version of BSSA is proposed (TVBSSA) for the feature
89 selection task. TVBSSA will dynamically change the number of leaders over the course
90 of iterations, as it starts with few leaders to emphasize the exploration process but with
91 the advancement of the iterations it increases them to intensify the exploitation process.
- 92 • Unlike most of the previous works, RWN is utilized as an induction algorithm in the

93 proposed wrapper-based approach. To the best of our knowledge, there is a lack work
94 that investigates and experiments the efficiency of RWN for this task on a large number
95 of benchmark datasets.

- 96 • Another contribution in the proposed approach, is that the number of neurons in the
97 hidden layer of RWN is automatically tuned by means of TVBSSA operators which
98 eliminate the need for any additional effort to tune this important parameter.

99 The rest of this paper is organized as follows: Section 2 reviews previous works on wrapper
100 based FS. The preliminaries of the main algorithms used in this work are given in Section 3.
101 The proposed wrapper approach is described in details in Section 4. The experiments and
102 results are discussed in Section 5. Finally, a summary of findings, conclusions, and future
103 work are given in Section 6.

104 2. Previous works

105 Nature-inspired metaheuristic algorithms have been used as beneficial and efficient tools
106 for searching the feature space because of their global (exploration) and local search (ex-
107 ploitation) capabilities. Recently, they were extensively used by the community of machine
108 learning to tackle a wide spectrum of FS problems. Different types of metaheuristic al-
109 gorithms including evolutionary algorithms (e.g. GA), swarm intelligence algorithms (e.g.
110 PSO) and ACO) and others (e.g. Memetic algorithms) were utilized to tackle these problems.

111 GA is one of the first evolutionary algorithms that were employed in various wrapper
112 FS methods with different classifiers. GA was used with SVM classifier as an evaluator to
113 design wrapper FS approaches as in (Shoorehdeli et al., 2006; Chen et al., 2008; Tan et al.,
114 2008; Seo et al., 2014). While other wrapper-based approaches used KNN (Cho et al., 2008;
115 Derrac et al., 2009; Da Silva et al., 2011; Winkler et al., 2011; Seo et al., 2014; Jeong et al.,
116 2015; Emary et al., 2016a; Mafarja and Mirjalili, 2018; Faris et al., 2018). ANN also has
117 been combined with GA for FS approaches in (Yang and Honavar, 1998; Hong and Cho,
118 2006; Tan et al., 2008; Yusta, 2009).

119 Derrac et al. (2009) introduced a co-evolutionary algorithm based on GA for FS. The
120 population is divided into three parts, the first part was used for FS, the second was used
121 for instance selection, while the third was used for both instance and FS. Rather than
122 the wrapper approaches, two hybrid filter-wrapper approaches that used GA as a search
123 strategy were proposed in (Tan et al., 2008; Oreski and Oreski, 2014), two variants of GA
124 were employed as wrapper-FS methods Li et al. (2010). In those approaches, KNN was used
125 as a base classifier in a dynamic Adaboost learning model. Another approach, that used GA
126 for both FS and optimizing the structure of the MLP classifier, was proposed in In (Souza
127 et al., 2011).

128 In addition, various SI algorithms were employed to search the feature space in different
129 FS methods. For instance, the PSO algorithm was combined with SVM classifier in many
130 wrapper FS methods as in (Unler et al., 2011; Alba et al., 2007; Tang et al., 2005). Other
131 approaches that adopted KNN classifier as an evaluator were proposed as in (Xue et al.,
132 2012, 2013). Moreover, some approaches used the ANN classifier (Agrafiotis and Cedeno,
133 2002; Huang and He, 2007)

134 An ACO variant was used with SVM for a wrapper FS approach in a face recognition
135 application in (Yan and Yuan, 2004). ACO was combined with DE in the work presented
136 by Khushaba et al. (2008), in which the former algorithm was used to find initial solutions
137 and the latter algorithm improved the quality of solution obtained. Another work based on
138 ACO that is named (ABACO) was proposed by Kashef and Nezamabadi-pour (2013). A
139 recent FS approach that utilized a hybrid ACO-ABC model (called AC-ABC) was proposed
140 by Shunmugapriya and Kanmani (2017) to benefit from advantages of both ABC and ACO.

141 A Grey Wolf Optimizer (GWO) based wrapper approach was proposed by (Emary et al.,
142 2016b) and applied to select an optimal subset of features for classification problems. Two
143 binary versions of GWO were tested and compared to GA and PSO. Another wrapper
144 approach based on KNN used Whale Optimization Algorithm (WOA) was introduced by
145 Mafarja and Mirjalili (Mafarja and Mirjalili, 2018), in which several variants of binary WOA
146 were considered to search for the optimal feature subset for classification problems. The same
147 authors proposed another wrapper based algorithm that combines WOA with Simulated
148 Annealing (SA) to solve the problem of feature selection (Mafarja and Mirjalili, 2017). The
149 idea was to require SA to exploit the promising regions found by WOA.

150 A recent binary Ant Lion Optimizer (ALO) was proposed in (Emary et al., 2016a) to
151 handle the feature selection problem. Recently, Faris *et al.* proposed a FS wrapper ap-
152 proach based on a newly proposed metaheuristic called Salp Swarm Algorithm (SSA) and
153 KNN. Many other FS wrapper approaches based on SI algorithms have been proposed such
154 as Binary Cuckoo Search (Rodrigues et al., 2013), Gravitational Search Algorithm (GSA)
155 (Rashedi and Nezamabadi-pour, 2014), Binary Bat Algorithm (BBA) (Nakamura et al.,
156 2013), Harmony Search (HS) (Ramos et al., 2011; Zainuddin et al., 2016), Competitive Swarm
157 Optimizer (Gu et al., 2018) and Binary Grasshopper Optimization Algorithm (BGOA) (Ma-
158 farja et al., 2018b).

159 As previously mentioned, various metaheuristic methods adopted as the wrapper ap-
160 proach for FS. Many of them are using one of the popular classifiers in wrapper approach,
161 that is KNN. However, few research works employed RWN instead of KNN. In (Chyzyk
162 et al., 2014), the authors have used Extreme Learning Machine (ELM) that can be imple-
163 mented a special type of RWN to evaluate the quality of the selected features. They utilized
164 GA to explore the search space for the best subset of features and applied their method on
165 Alzheimer’s disease dataset. Another work was done by Yang *et al.* (Yang et al., 2016) that
166 used both DE and ELM to deal with a tool wear estimation application. The developed
167 algorithm is divided into two phases. In the first phase, a discrete DE was employed with
168 ELM to select the input features. In the second phase, a continuous DE was used to optimize
169 the kernel function parameters of ELM. The results showed the superiority of the proposed
170 method in tool wear estimation.

171 3. Preliminaries

172 3.1. Salp Swarm Algorithm

173 SSA is a new meta-heuristic technique recently proposed by (Mirjalili et al., 2017) to
174 efficiently find the optimal solutions for different classes of constrained and unconstrained
175 problems. Salp is known as a type of Salpidae family. Swarming behaviors of salps is a very
176 noticeable phenomenon because salps can build cooperative semi-parallel chains throughout

177 foraging events in the deep oceans. Based on the collaborative strategies of salps, they can
 178 preserve more energy when foraging. The SSA method is developed based on the behavior
 179 of salps in forming chains and their foraging tactics. From optimization point of view, these
 180 chains have a significant impact on balancing the exploration and exploitation inclinations
 181 of SSA by assisting it in escaping from local optima (LO) and avoiding stagnation problems.

182 In SSA, the population is formed using some chains of salps, which consist of two types
 183 of salps: leader and followers. When the agent is front-runner, we label it as leader, whereas
 184 other salps will be classified as followers. The role of leader salp is to guide and lead the
 185 direction and next steps of population, while the follower salps pay attention to other peers.
 186 A schematic view of salp chain is shown in Fig. 1.



Figure 1: Illustration of a salp chain

187 In SSA, position of salps during the exploration and exploitation phases is defined as an
 188 n -dimensional space, where n is total number of variables. For a set of salps X made up of
 189 N salps with d dimensions, population of SSA is recorded in a $(N \times d)$ -dimensional matrix,
 190 as described in Eq. (1):

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \quad (1)$$

191 In SSA, all salps are trying to forage and use the food source using the salp chains. Therefore,
 192 food source is the target location of swarm and the position of leader salp is determined based
 193 on situation of this source. This rule is expressed in Eq. (2):

$$x_j^1 = \begin{cases} F_j + c_1 ((ub_j - lb_j) c_2 + lb_j) & c_3 \geq 0.5 \\ F_j - c_1 ((ub_j - lb_j) c_2 + lb_j) & c_3 < 0.5 \end{cases} \quad (2)$$

where x_j^1 denotes the location of leader, and F_j is the state of food source in the j^{th} dimension, ub_j is the superior restriction of j^{th} dimension, and lb_j shows the inferior limit of j^{th} dimension, c_2 and c_3 are two random values inside $[0, 1]$, and c_1 can be described as a significant parameter in SSA, which is expressed as in Eq. (3):

$$c_1 = 2e^{-\left(\frac{4t}{T_{max}}\right)^2} \quad (3)$$

194 where t is the iteration, and T_{max} defines the maximum number of iterations. The parameter
 195 c_1 plays a noticeable role in harmonizing the exploration and exploitation propensities of SSA.
 196 The position vector of salps are defined and updated in each iteration as in Eq. (4):

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \quad (4)$$

197 where $i \geq 2$ and x_j^i represents the situation of i^{th} salp at the j^{th} dimension. The pseudo-code
 of conventional SSA is shown in Algorithm 1.

Algorithm 1 Pseudo-code of SSA

Input: Swarm size and total number of iterations (T_{max}).

Output: The best salp and its fitness value.

Initialize the random salps $x_i(i = 1, 2, \dots, n)$

while (Termination condition is not true) **do**

 Evaluate the fitness of salps (search agents)

 Determine the fittest agent and set it as **F**

 Update c_1 by Eq. (3)

for (each salp (x_i)) **do**

if ($i == 1$) **then**

 Update the position vector of leader by Eq. (2)

else

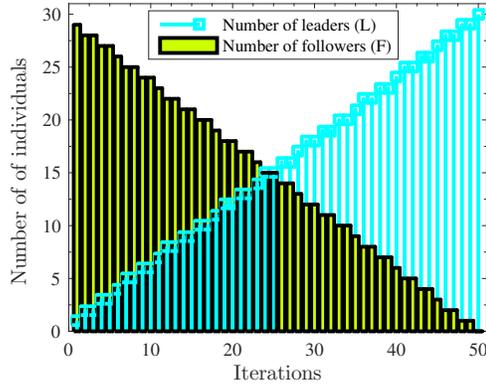
 Update the location of follower salps by Eq. (4)

 Update all agents with regard to the superior and inferior bounds of decision variables

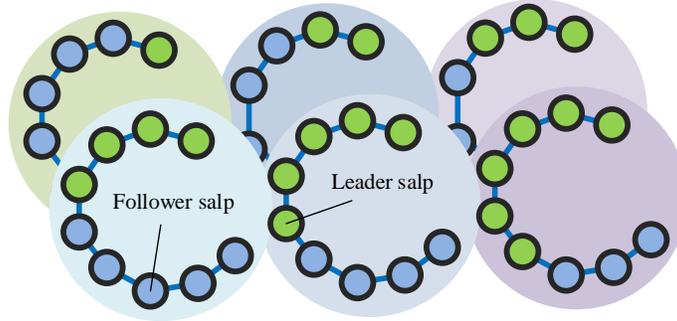
 Return those salps that violated the bounds.

Return **F**

198
 199 Considering Algorithm 1, after defining the population size and possible number of it-
 200 erations, the SSA algorithm first distributes random salps inside the defined search space.
 201 It then evaluates the objective value (fitness) of all salps to sort and detect the best search
 202 agent F . All salps are interested to track and follow the prime candidate salp (leader) as
 203 shown in Fig. 2. During searching process, the variable c_1 is calculated in each step by Eq.
 204 (3). The formula in Eq. (2) can assist SSA in finding and updating the location of leader,
 205 whereas Eq. (4) is utilized to inform other salps. Pending the termination state, all phases
 206 of SSA, apart from the initialization stage, should be repeated to rise the excellence of agents
 207 as much as possible and find a suboptimal or optimum solution for the intended problem.



(a) Number of leaders and followers



(b) Time-varying hierarchy in some iterations (order from upper left to lower right)

Figure 2: Monitoring of the number of leaders and followers and structure of hierarchy over the course of iterations in TVBSSA

208 *3.2. Random weight networks (RWN)*

209 In 1992, Schmidt et al. (1992) introduced RWNs for tackling the training phases of
 210 single-hidden layer feed-forward neural networks (SLFNs) (Cao et al., 2018). Later, Pao
 211 et al. (1994) also established a similar model termed random vector functional-link networks
 212 (RVFLNs). The RWN was developed to generalize the SLFN. Also, it aims to generalize
 213 multi hidden-layer feed forward networks where a node is considered a subnetwork connecting
 214 additional hidden nodes. In regards to gradient descent approaches used for training of
 215 SLFN, we can observe that learning ratio of RWN is particularly fast. It also can provide
 216 a better generalization routine. What is more, not like methods such as back propagation,
 217 which the user should manually set the parameters such as learning rate, number of epochs,
 218 etc., the RWN does not necessitate thoroughgoing human involvement.

219 However, RWN does not optimize in an iterative manner to handle the parameters of
 220 SLFN. As an alternative, input weights and biases will be randomly initialized, and then it
 221 concludes the output weighting vector systematically based on Moore-Penrose generalized
 222 inverse. There are more learning and generalization characteristics for the RVFLNs. Inter-
 223 ested readers are referred to Pao et al. (1994) and Igelnik and Pao (1995). In RVFLNs, direct

224 links are established and initialized from input layer and lead to output layer. This fact is the
 225 only dissimilarity among RWNs and RVFLNs. Considering the satisfactory learning rate,
 226 generalization and approximation proficiencies of RWNs and RVFLNs, they have involved
 227 researchers in diverse areas of engineering and science, significantly.

Suppose N distinct samples represented by (x_i, t_i) , where $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T \in \mathbb{R}^n$ and $t_i = [t_{i1}, t_{i2}, \dots, t_{in}]^T \in \mathbb{R}^m$, SLFNs with activation function $g(x)$ and \tilde{N} hidden neurons are formulated as in Eq. (5) (Huang et al. (2004)):

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot x_j + b_i) = o_j, j = 1, \dots, N \quad (5)$$

228 where b_i is the threshold of the i th hidden neuron, $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ is the vector
 229 consists of weighting coefficients for linking the i th hidden neuron to input elements, $\beta_i =$
 230 $[\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ denotes the weighting coefficients to link the i th hidden element to the
 231 output elements (Huang et al. (2004)).

The SLFNs with \tilde{N} hidden neurons and activation function $g(x)$ should approximate the original N samples with zero error, that is $\sum_{i=1}^{\tilde{N}} \|o_j - t_j\| = 0$, i.e., we know w_i , β_i , and b_i , such that (Huang et al. (2004)):

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot x_j + b_i) = t_j, j = 1, \dots, N \quad (6)$$

The above-stated N rules are described as in Eq. (7):

$$H\beta = T \quad (7)$$

where

$$H(w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & & \vdots \\ g(w_1 \cdot x_N + b_1) & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (8)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m}, T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (9)$$

232 where H shows the hidden layer output matrix, the i th column of H is the i th hidden output
 233 vector with respect to x_1, x_2, \dots, x_N Huang et al. (2004). The general structure of RWN is
 234 demonstrated in Fig. 3.

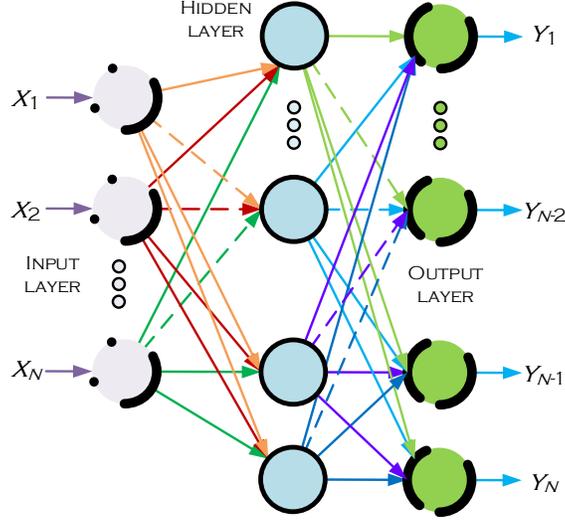


Figure 3: Overall structure of RWN

235 4. Proposed TVBSSA-RWN wrapper

236 As any wrapper-based feature selection method, our proposed approach consists of three
 237 main components as follows:

- 238 • The search algorithm: which will be implemented as a dynamic version of the SSA
 239 optimizer.
- 240 • The induction algorithm: in our case the random weight network will be utilized.
- 241 • The evaluation metric: Our proposed evaluation metric will consist of three main parts
 242 that combine the classification accuracy, the feature reduction rate and complexity of
 243 the induction algorithm.

244 4.1. Binary SSA

245 Based on Mirjalili and Lewis (2013), one of the efficient techniques in converting contin-
 246 uous algorithms to binary versions is to utilize transfer functions (TFs). TFs are proposed
 247 based on definition of a probability value for updating each individual's element represent-
 248 ing the selected subset of features. In this work, we utilize one of the most commonly used
 249 S-shaped TF to covert the continuous SSA to a binary variant, which is called BSSA. The
 250 S-shaped TF can be calculated by Eq. (10).

$$T(x_j^i(t)) = \frac{1}{1 + e^{-x_j^i(t)}} \quad (10)$$

251 where $X_i^d(t+1)$ is the i -th element at d^{th} dimension in X solution, $T(x_j^i(t))$ is the probability
 252 value, which can be obtained via Eq. (10). Then, the valued of this function is used to
 253 generate ones and zeros in the individual as given in Eq. (11).

$$x_i^k(t+1) = \begin{cases} 0 & \text{If } rand < T(v_i^k(t+1)) \\ 1 & \text{If } rand \geq T(v_i^k(t+1)) \end{cases} \quad (11)$$

254 where x_j^i is the j -th element in x solution in the j -th dimension, and t is the current
 255 iteration. The utilized TF is shown in Fig. 4.

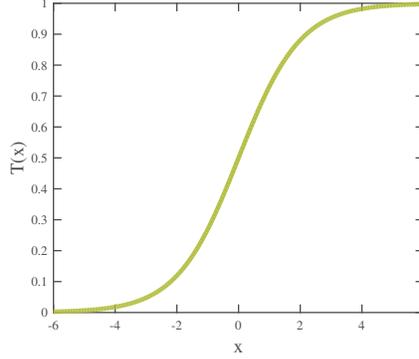


Figure 4: Utilized S-shaped TF

256 4.2. Time-Varying Binary Salp Swarm Algorithm (TVBSSA)

257 The basic binary SSA (BSSA) has a satisfactory performance in balancing the exploration
 258 and exploitation. However, there is room for further improvements in the searching leanings
 259 of binary SSA because it may stagnate in local optima (LO), and premature convergence
 260 can be happen in dealing with some feature spaces. In order to alleviate the stagnation
 261 disadvantages of BSSA, a modified leadership structure is proposed in this work to develop an
 262 improved binary variant called TVBSSA. One important phenomenon that has a significant
 263 impact on the performance of SSA is the way the follower salps will follow the leader salp
 264 during the searching phases. In SSA, there is one leader in each iteration, which is followed
 265 by $(N-1)$ salps, where N is the number of all agents. Because we need a high exploration in
 266 beginning of the search and more exploitative steps in the last stages, a dynamic time-varying
 267 structure for the leadership hierarchy of SSA is utilized here. During the course of iterations,
 268 this strategy linearly increases the number of leaders and decreases the number of follower
 269 salps. In each iteration, several leaders are determined and then, the rest of population is
 270 employed to play the role of follower salps in the exploration and exploitation phases.

271 Hence, unlike the BSSA, in the time-varying version of BSSA (TVBSSA), the number of
 272 leaders and followers change of the course of iterations. In this regard, the number of leaders
 273 is determined based on the following rule:

$$L = \text{ceil}(N \times \frac{i}{\text{MaxIter} + 1}) \quad (12)$$

274 where N is the population size, i is the index of each salp, and MaxIter is the upper limit
 275 for iterations. According to Eq. (12), the number of follower salps (F) in each iteration can
 276 be obtained via Eq. (13)

$$F = N - L \quad (13)$$

The pseudo-code of proposed TVBSSA is shown in Algorithm 2.

Algorithm 2 Pseudo-code of TVBSSA

```

Initialize the salp population  $x_i (i = 1, 2, \dots, n)$  considering  $ub$  and  $lb$ 
while (end condition is not satisfied) do
    Calculate the fitness of each search agent (salp)
    Set  $\mathbf{F}$  as the best search agent
    Update  $c_1$  by Eq. 3
    Update number of leaders  $L$  by Eq. (12)
    Update number of followers  $F$  by Eq. (13)
    for (each salp ( $x_i$ )) do
        if  $x_i$  is a leaser then
            Update the position of the leading salp by Eq. (2)
            Calculate the probabilities using a TF which takes the output
            of Eq. (2) as its input as in Eq. 10 (S-Shaped)
        else
            Update the position of the follower salp by Eq. (4)
    Update all salps based on the upper and lower bounds of variables
Return the computed  $\mathbf{F}$ 

```

278 *4.3. Solution representation*

279 In TVBSSA-RWN, each individual (salp) represents a candidate solution for the targeted
280 problem. In our case, the solution is to find the near-optimal subset of features and to tune
281 the number of neurons in the hidden layer of the network, simultaneously. Therefore, the
282 individual in our proposed algorithm is designed to capture these two parts of the solution.
283 As shown in Fig. 5, each individual is represented as a one-dimensional binary array of two
284 parts: the features part (f) and the neurons part (n). The length of the features part equals
285 to the number of features in the dataset D , while the neurons part consists of a number of
286 bits K required to represent the maximum number of neurons. Therefore, the length of the
287 individual of TVBSSA-RWN is $D + K$.

288 The process of decoding the individual representation is illustrated in Fig. 5. The features
289 part used to select the features in the dataset that corresponds to the bits that has value of
290 1, while the other features are eliminated and the training dataset is reduced. The reduced
291 dataset is used to train a RWN that has a number of neurons determined by the neurons
292 part n . The resulted network is evaluated as described next.

293 *4.4. Fitness evaluation*

294 This component of the model is required to evaluate the quality (also known as fitness of
295 in metaheuristic algorithms) of the generated solutions by the swarm optimizer. For this, the
296 fitness function is designed to maximize the accuracy of the induction algorithm (i.e RWN
297 in our case), minimize the number of selected features, and to minimize the complexity of
298 the generated RWN by reducing the number of selected hidden neurons in the network.

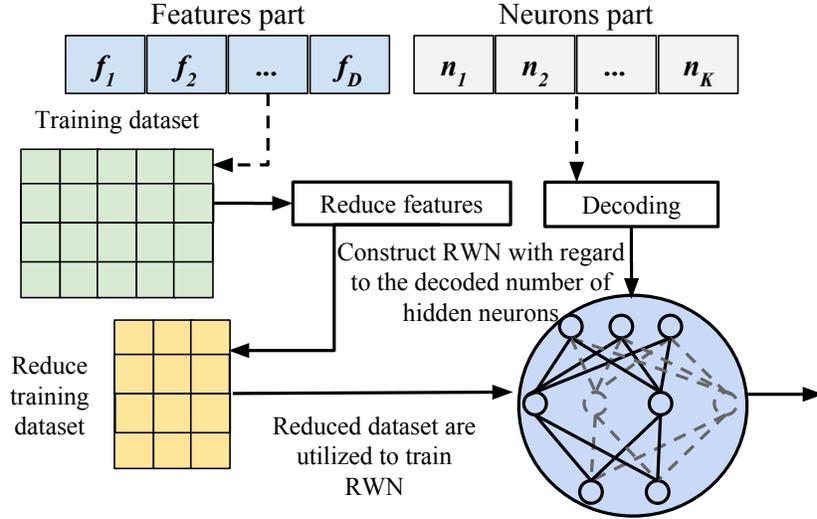


Figure 5: Solution representation in TVBSSA-RWN

$$Fitness = \alpha AccErr + \beta \frac{f}{F} + \gamma \frac{n}{N} \quad (14)$$

where $AccErr$ is the misclassification rate of the generated RWN network, f is the number of selected features by the BSSA, F is the number of features in the dataset, n represents the number of neurons determined by the optimizer, and N is the maximum number of possible neurons in the RWN network. The parameters α , β , and γ are three factors to control the weight of the contribution of their corresponding terms which are the misclassification rate, the reduction rate, and the complexity of the network, respectively. The maximum value of α , β , γ is 1 while minimum is 0. Typically, $\beta = (1 - \alpha)$ as in Emary et al. (2016a).

4.5. Procedure of TVBSSA-RWN

Combining all the parts described earlier, the procedure of the proposed TVBSSA-RWN can be described in the following steps:

- *Initialization*: A predefined number of individuals (i.e. salps) are randomly initialized. Each of these individuals is a candidate solution for the FS problem, and it consists of the feature selection part and the number of hidden neurons in the RWN part.
- *Fitness evaluation*: each individual in the swarm is evaluated using the following steps:
 - Individual split: each individual is divided into the features and neurons parts.
 - Feature selection and reduction of training dataset: The features part is utilized to select its corresponding features and eliminate non-selected features from the training dataset.
 - Network construction: The RWN is built using the specified number of hidden nodes by the individual.
 - Network training: RWN is trained based on the reduced training dataset.

- 320 – Network evaluation: The developed RWN network is evaluated based on a vali-
321 dation set
- 322 – Fitness calculation: the final fitness value is calculated as given in Eq. (14).
- 323 • *Hierarchy update*: In this process, the number of leaders and followers are updated
324 according to Eq. (12) and Eq. (13), respectively.
- 325 • *Exploit and Explore*: The updating mechanisms described in Eq. (2) and (4) are applied
326 to the leaders and followers to perform the exploitation and exploration processes.
- 327 • *Termination*: The previous steps are performed iteratively until the maximum number
328 of iterations is reached.

329 5. Experimental results and discussion

330 In this section summarizes the results of the proposed SSA with dynamic updating strate-
331 gies for different FS datasets are presented and analyzed.

332 5.1. Experimental setup

333 All experiments are conducted in similar conditions to provide fair comparisons. All
334 methods are coded in MATLAB 2013, and we used a same system with Intel Core(TM)
335 i5-5200U 2.2GHz of CPU and 4.0GB of RAM.

336 The parameters of the algorithms are set as follows. In PSO, acceleration constants c_1
337 and c_2 are set to 2, while the inertia weight is set to 0.9 to 0.4 as in Pacifico and Ludermir
338 (2013). The probabilities of crossover and mutation operators in GA are set to 0.8 and 0.1,
339 respectively, as in Yang et al. (2013). For all algorithms, the population/swarm size is set
340 to 50 with same iteration. The maximum number of hidden neurons in TVBSSA is set to
341 1024, which is represented by 10 elements in the chromosome. For training and testing, we
342 applied 10-folds cross-validation. In this method, each of the training and testing processes
343 is repeated 10 times, and then, the average (Avg) and standard deviation (Std) of the results
344 are calculated.

345 Table 1 shows the main characteristics of 20 well-studied datasets used in this research
346 to compare and investigate the efficacy of optimizers. These test cases are obtained from
347 the well-regarded UCI repository (Lichman, 2013), which utilized in many works and covers
348 varied characteristics and datasets with various features and instances.

349 5.2. Selection of $(\alpha \beta \gamma)$

350 In this subsection, the impact of α , β , and γ values on the performance of TVBSSA is
351 investigated based on the results for Leukemia dataset. These parameters play the role of
352 weighting factors that can change the contribution of each component in the final fitness
353 function.

354 Based on previous works, researchers always set α and β to 0.99 and 0.01, respectively
355 (Emary et al., 2016a, 2018; Mafarja et al., 2018a; Faris et al., 2018). Here, we investigate
356 the impact of different combinations of α , β , and γ values as tabulated in Table 2. Table 2
357 shows the impact of changing α , β , and γ values under six cases (S_1 to S_6) on the accuracy,

Table 1: List of datasets

No.	Dataset	No. of Features	No. of instances
1.	BreastEW	30	596
2.	Exactly	13	1000
3.	Exactly2	13	1000
4.	HeartEW	13	270
5.	Lymphography	18	148
6.	M-of-n	13	1000
7.	PenglungEW	325	73
8.	SonarEW	60	208
9.	SpectEW	22	267
10.	CongressEW	16	435
11.	KrvskpEW	36	3196
12.	Tic-tac-toe	9	958
13.	Vote	16	300
14.	WaveformEW	40	5000
15.	WineEW	13	178
16.	Clean1	166	476
17.	Semeion	265	1593
18.	Colon	2000	62
19.	Leukemia	7129	72
20.	TOX-171	5749	171

Table 2: Impact of α , β and γ on the performance of TVBSSA in terms of accuracy, number of features, and number of neurons for the Leukemia dataset.

Case	α	β	γ	Accuracy		Number of features		Number of neurons	
				Avg	Std	Avg	Std	Avg	Std
S_1	0.999	0.001	0.001	0.9238	0.0808	3958.8000	776.5847	493.4000	138.3267
S_2	0.995	0.005	0.005	0.9048	0.0670	3786.0000	1011.5861	505.9000	167.5340
S_3	0.99	0.01	0.01	0.9156	0.0366	4171.2600	405.7624	470.9400	25.7681
S_4	0.95	0.05	0.05	0.8321	0.2200	3570.2000	1268.9346	448.3000	173.2693
S_5	0.8	0.2	0.2	0.7655	0.1443	2096.3000	956.3636	192.4000	133.5808
S_6	0.5	0.5	0.5	0.6435	0.1551	1359.4000	374.0280	33.8000	42.7000

number of features, and number of neurons obtained by TVBSSA. As per accuracy results in Table 2, we see that the TVBSSA with S_1 case outperforms TVBSSA with other cases in terms of accuracy rate. According to the number of features, we see the S_6 case comes to the first place. Based on number of neurons, S_6 case is the most preferable one. Because the most important criteria for the majority of FS scenarios is the accuracy of classification, however, we validated all methods based on the S_1 cases. Note that if we set any other value for these parameters, the performance of the proposed BDSSA can be repeated again.

5.3. Effect of transfer function

In this subsection, we study the effect of v-shape and s-shape transfer functions Mirjalili and Lewis (2013) on the performance of the proposed TVBSSA. Table 3 shows the results in terms of accuracy rate of two versions of TVBSSA one used s-shape and the other used v-shape. The results show that both versions of TVBSSA have very competitive results with a slight advantage for the version that incorporated s-shape transfer function, as it obtained

371 the best average results in 13 datasets out of 20 datasets. Therefore, TVBSSA with s-shape
 372 transfer function will be used in the rest of experiments and comparisons.

Table 3: Classification accuracy rates of the TVBSSA-based on s-shape and v-shape transfer functions.

Dataset	TVBSSA-RWN			
	S-Shape		V-Shape	
	Avg	Std	Avg	Std
BreastEW	0.9616	0.0053	0.9497	0.0114
Exactly	0.9986	0.0013	0.9958	0.0045
Exactly2	0.7828	0.0124	0.7780	0.0095
HeartEW	0.8652	0.0067	0.8807	0.0169
Lymphography	0.8706	0.0199	0.8587	0.0290
M-of-n	0.9922	0.0030	0.9928	0.0042
penglungEW	0.7578	0.0281	0.6979	0.0578
SonarEW	0.8176	0.0256	0.7958	0.0281
SpectEW	0.8750	0.0097	0.8965	0.0135
CongressEW	0.9637	0.0086	0.9647	0.0129
KrvskpEW	0.9816	0.0025	0.9735	0.0024
Tic-tac-toe	0.9818	0.0031	0.9534	0.0165
Vote	0.9611	0.0098	0.9686	0.0051
WaveformEW	0.8307	0.0051	0.8222	0.0039
WineEW	0.9546	0.0243	0.9548	0.0200
clean1	0.8631	0.0112	0.8593	0.0072
semeion	0.9704	0.0039	0.9652	0.0024
Colon	0.7938	0.0433	0.6119	0.0204
Leukemia	0.9156	0.0366	0.6493	0.0775
TOX-171	0.7625	0.0324	0.7821	0.0295

373 5.4. Results and discussion

374 Table 4 shows the results of BSSA-RWN in terms of accuracy measure against other
 375 competitors BSSA, GA, and PSO using k-NN classifier. As per results in Table 4, we see that
 376 the BSSA with RWN can outperform all competitors on approximately 70 % of datasets with
 377 satisfying STD results. After BSSA-RWN, the GA-k-NN has reached to the best classification
 378 accuracy rates on the M-of-n, SonarEW, and clean1 datasets. For penglungEW case, the
 379 PSO-k-NN shows the best rate. As it can be observed in the results of BSSA-k-NN, it cannot
 380 show superior efficacy on any dataset, especially compared to BSSA-RWN. However, when
 381 comparing BSSA-k-NN with GA-k-NN and PSO-k-NN, we observe that it cannot outperform
 382 them on majority of datasets. This observation indicants that the binary SSA outperforms
 383 GA and binary PSO when it is joined with a RWN classifier, and a significant improvement
 384 in the accuracy rates is seen due to the unique advantages of RWN over k-NN classifier. But
 385 when we integrate it with k-NN, the results cannot be better than BSSA-RWN variant. This
 386 also shows the significant role of RWN classifier on the accuracy rates.

387 Table 5 tabulates the results of BSSA-RWN in terms of number of selected features
 388 compared to other competitors BSSA, GA, and PSO with k-NN classifier. As per results in
 389 Table 5, we see that the GA-k-NN has attained the best rank.

Table 4: Accuracy results obtained by BSSA-RWN against results of other methods using k-NN

Dataset	BSSA-RWN		BSSA-k-NN		GA-k-NN		PSO-k-NN	
	Acc	Std	Acc	std	Acc	std	Acc	std
BreastEW	0.9504	0.0133	0.9455	0.0085	0.9409	0.0027	0.9409	0.0051
Exactly	0.9980	0.0014	0.9908	0.0054	1.0000	0.0000	0.9904	0.0039
Exactly2	0.7744	0.0087	0.7502	0.0126	0.7580	0.0000	0.7580	0.0000
HeartEW	0.8600	0.0173	0.8052	0.0127	0.7926	0.0166	0.7852	0.0172
Lymphography	0.8557	0.0178	0.7796	0.0188	0.8060	0.0166	0.7654	0.0283
M-of-n	0.9898	0.0041	0.9978	0.0015	1.0000	0.0000	0.9974	0.0018
penglungEW	0.7600	0.0278	0.8597	0.0268	0.8786	0.0186	0.8972	0.0233
SonarEW	0.7988	0.0070	0.8226	0.0081	0.8239	0.0071	0.8228	0.0301
SpectEW	0.8688	0.0134	0.7887	0.0097	0.7919	0.0212	0.7758	0.0137
CongressEW	0.9614	0.0064	0.9394	0.0153	0.9445	0.0064	0.9398	0.0079
KrvskpEW	0.9804	0.0024	0.9693	0.0026	0.9758	0.0023	0.9674	0.0024
Tic-tac-toe	0.9787	0.0046	0.8109	0.0116	0.8194	0.0038	0.8204	0.0104
Vote	0.9601	0.0090	0.9215	0.0068	0.9238	0.0107	0.9322	0.0052
WaveformEW	0.8324	0.0039	0.7832	0.0057	0.7970	0.0060	0.7874	0.0074
WineEW	0.9619	0.0203	0.9371	0.0106	0.9428	0.0105	0.9437	0.0108
clean1	0.8567	0.0156	0.8545	0.0102	0.8583	0.0119	0.8528	0.0084
semeion	0.9705	0.0035	0.9702	0.0017	0.9688	0.0039	0.9700	0.0021
Colon	0.7805	0.0301	0.7505	0.0456	0.7395	0.0431	0.7548	0.0424
Leukemia	0.8994	0.0412	0.8714	0.0126	0.8907	0.0142	0.8868	0.0101
TOX-171	0.7516	0.0343	0.7528	0.0467	0.7491	0.0204	0.7308	0.0288

Table 5: Number of features for BSSA-RWN against other methods using k-NN

Dataset	BSSA-RWN		BSSA-k-NN		GA-k-NN		PSO-k-NN	
	Avg	Std	Avg	std	Avg	std	Avg	std
BreastEW	16.3600	0.7403	17.9800	1.3217	12.6600	0.8019	14.6400	0.7162
Exactly	12.8800	0.3271	6.6000	0.1225	6.0000	0.0000	6.6200	0.1304
Exactly2	13.9200	0.6419	1.9600	0.3286	1.0000	0.0000	1.0000	0.0000
HeartEW	10.8800	1.3773	8.7200	0.2168	7.5200	0.4025	7.0200	0.4438
Lymphography	13.3800	1.8580	12.0400	0.7335	10.5000	0.6325	9.8400	0.6348
M-of-n	12.8000	0.6519	6.8200	0.0837	6.0000	0.0000	6.9200	0.2049
penglungEW	124.0200	12.6458	191.3800	1.2398	150.2800	2.1476	155.2200	0.9935
SonarEW	40.9800	2.0705	38.5800	1.4096	29.6800	1.1189	30.5400	0.9633
SpectEW	16.3000	1.7776	15.0600	0.6731	9.8000	0.4690	10.5200	0.4207
CongressEW	11.0800	1.2071	7.0600	0.5899	3.6400	0.3362	4.9200	0.6573
KrvskpEW	34.1600	1.0455	23.5400	0.5941	18.1600	0.7232	20.0800	0.5263
Tic-tac-toe	14.6800	0.2490	6.9800	0.0447	6.9600	0.0894	6.9600	0.0548
Vote	10.7000	1.0173	7.0600	0.6731	3.2000	0.3674	4.3800	0.5167
WaveformEW	32.5800	0.9149	25.8200	1.1212	19.6000	0.5050	20.8200	0.8556
WineEW	7.6200	0.3701	8.6200	0.5263	6.8000	0.1732	7.1400	0.4037
clean1	115.4200	3.3214	107.0000	3.3830	81.6600	1.6832	82.6800	0.6140
semeion	162.1800	8.5377	168.2000	5.1841	132.4200	1.1946	131.7800	2.5956
Colon	1188.9000	87.6461	1272.9400	72.7751	986.2800	7.0871	988.1000	6.1053
Leukemia	4022.1600	211.6794	4744.3600	200.2358	3538.5400	8.8141	3542.4200	8.7833
TOX-171	3392.6400	142.5020	3851.4000	141.6246	2863.8800	6.0521	2868.9600	6.0521

390 Table 6 demonstrates the results of BSSA-RWN in terms of fitness values compared to
 391 other peers with k-NN classifier. The variation of fitness results are also visualized in Fig.
 392 6 Referring to the results in Table 6, it is observed that BSSA with RWN has attained the
 393 fittest results on 90 % of datasets, which is followed by GA-k-NN, PSO-k-NN, and BSSA-k-
 394 NN (see Fig. 6). These results indicate that the BSSA-based feature selection using RWN
 395 has increased the quality of found features compared to previous k-NN methods.

Table 6: Fitness for BSSA-RWN against other methods using k-NN

Data	BSSA-RWN		BSSA-k-NN		GA-k-NN		PSO-k-NN	
	Avg	Std	Avg	std	Avg	std	Avg	std
BreastEW	0.0200	0.0014	0.0520	0.0014	0.0474	0.0008	0.0572	0.0023
Exactly	0.0094	0.0014	0.0066	0.0011	0.0046	0.0000	0.0073	0.0012
Exactly2	0.1980	0.0038	0.2410	0.0002	0.2404	0.0000	0.2404	0.0000
HeartEW	0.0740	0.0070	0.1823	0.0013	0.1930	0.0025	0.1992	0.0047
Lymphography	0.0218	0.0103	0.1481	0.0049	0.1542	0.0025	0.1700	0.0018
M-of-n	0.0075	0.0004	0.0058	0.0003	0.0046	0.0000	0.0073	0.0012
penglungEW	0.0128	0.0109	0.0903	0.0020	0.0884	0.0025	0.1048	0.0049
SonarEW	0.0251	0.0071	0.0951	0.0024	0.0921	0.0034	0.1134	0.0034
SpectEW	0.0734	0.0046	0.1807	0.0017	0.1806	0.0037	0.1965	0.0031
CongressEW	0.0130	0.0023	0.0460	0.0015	0.0387	0.0015	0.0471	0.0024
KrvskpEW	0.0235	0.0007	0.0369	0.0012	0.0247	0.0007	0.0379	0.0021
Tic-tac-toe	0.0178	0.0021	0.1806	0.0013	0.1960	0.0023	0.1969	0.0023
Vote	0.0105	0.0034	0.0530	0.0009	0.0462	0.0025	0.0543	0.0025
WaveformEW	0.1587	0.0009	0.2657	0.0013	0.2455	0.0012	0.2692	0.0015
WineEW	0.0039	0.0003	0.0269	0.0015	0.0281	0.0017	0.0370	0.0030
clean1	0.0416	0.0049	0.1063	0.0012	0.1009	0.0027	0.1177	0.0023
semeion	0.0152	0.0006	0.0252	0.0003	0.0233	0.0006	0.0276	0.0006
Colon	0.0511	0.0021	0.1866	0.0019	0.2029	0.0041	0.2179	0.0045
Leukemia	0.0105	0.0004	0.0457	0.0027	0.0447	0.0011	0.0636	0.0051
TOX-171	0.0012	0.0001	0.1273	0.0032	0.1200	0.0084	0.1460	0.0065

396 The accuracy results of proposed TVBSSA versus BSSA with RWN are compared in
 397 Table 7. As per results in Table 7, it is observed that the TVBSSA outperforms the BSSA
 398 in dealing with 80 % of datasets. Table 8 compares TVBSSA to BSSA based on the number
 399 of features. As per results in Table 8, it is seen that the TVBSSA is superior to BSSA in
 400 terms of obtained set of features in dealing with Exactly, HeartEW, Lymphography, M-of-n,
 401 penglungEW, SonarEW, SpectEW, CongressEW, Vote, WineEW, semeion, and TOX-171.

402 The convergence curves of TVBSSA-RWN are compared to BSSA-RWN, BSSA-KNN,
 403 GA-KNN, and PSO-KNN methods in Figs. 7 and 8. The convergence trends for BreastEW,
 404 Exactly2, HeartEW, Lymphography, penglungEW, SonarEW, SpectEW, CongressEW, Tic-
 405 tac-toe, Vote, WaveformEW, WineEW, Semeion, Colon, Leukemia, and TOX-171 show
 406 that there is a significant gap between the curves of BSSA-KNN, GA-KNN, PSO-KNN and
 407 those for BSSA-RWN and TVBSSA-RWN methods, which are equipped with RWN learning
 408 schemes. This also indicates the significant impact of RWN' structure on convergence speed
 409 of the developed BSSA-based FS methods. The trends also show the unstable balance of
 410 exploratory and exploitative trends in PSO-KNN and GA-KNN for majority of datasets such

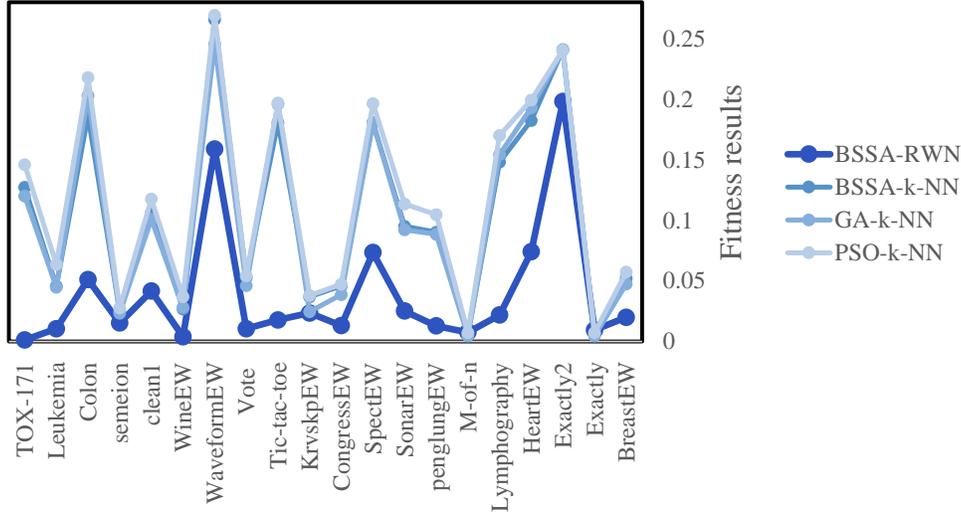


Figure 6: Variation of fitness results

411 as BreastEW, HeartEW, Lymphography, penglungEW, SonarEW, SpectEW, CongressEW,
 412 and TOX-171, while the proposed TVBSSA-RWN shows an stable and superior performance
 413 for majority of cases. The convergence curves of TVBSSA-RWN and BSSA-RWN are very
 414 competitive and both superior to other peers.

415 As mentioned in the previous sections, one the main features of the proposed TVBSSA-
 416 RWN is that it automatically tunes the number of neurons in the hidden layer of its RWN.
 417 This feature makes the algorithm needs little human intervention/involvement and saves time
 418 required for tuning this important parameter. Figures 9 and 10 demonstrate the change of
 419 average number of neurons over the course of iterations in TVBSSA-RWN for all benchmark
 420 datasets. Although the algorithm was given a large space to search for this number, which
 421 is [1,1024], TVBSSA-RWN was capable in narrowing down this range over the course of
 422 iterations, until it reaches a number that minimizes the fitness value.

423 Results of Wilcoxon statistical rank-sum test are presented in Table 9. According to
 424 the p-values in this table, we can see that the difference between the accuracy rates of the
 425 proposed TVBSSA-RWN versus other peers are significantly meaningful in the majority of
 426 cases.

427 5.5. Comparison with other meta-heuristics in literature

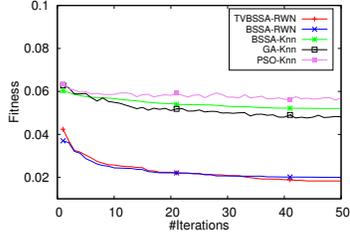
428 In this section, the classification accuracy rates obtained by the proposed TVBSSA com-
 429 bined with RWN is compared with those reported in previous literature. The results are
 430 obtained from three recognized papers, which utilized same datasets for comparison pur-
 431 poses. In Table 10, the efficacy of TVBSSA in terms of classification accuracy is compared
 432 with the performance of Genetic Algorithm (GA1) and Particle Swarm Optimization (PSO1)
 433 in Kashef and Nezamabadi-pour (2015) performed based on the source code developed by
 434 the same researchers. In addition, the accuracy rates are compared with rates of the bi-
 435 nary Grey Wolf Optimization - approach 1 (bGWO1), binary Grey Wolf Optimization -
 436 approach 2 (bGWO2), GA2 and PSO2 from the work in Emary et al. (2016a), Binary But-
 437 tterfly Optimization Algorithm (S-bBOA) Arora and Anand (2019), Binary Gravitational

Table 7: Accuracy for BSSA-RWN versus TVBSSA-RWN

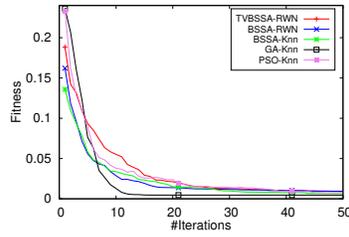
Dataset	BSSA-RWN		TVBSSA-RWN	
	Acc	Std	Acc	Std
BreastEW	0.9504	0.0133	0.9616	0.0053
Exactly	0.9980	0.0014	0.9986	0.0013
Exactly2	0.7744	0.0087	0.7828	0.0124
HeartEW	0.8600	0.0173	0.8652	0.0067
Lymphography	0.8557	0.0178	0.8706	0.0199
M-of-n	0.9898	0.0041	0.9922	0.0030
penglungEW	0.7600	0.0278	0.7578	0.0281
SonarEW	0.7988	0.0070	0.8176	0.0256
SpectEW	0.8688	0.0134	0.8750	0.0097
CongressEW	0.9614	0.0064	0.9637	0.0086
KrvskpEW	0.9804	0.0024	0.9816	0.0025
Tic-tac-toe	0.9787	0.0046	0.9818	0.0031
Vote	0.9601	0.0090	0.9611	0.0098
WaveformEW	0.8324	0.0039	0.8307	0.0051
WineEW	0.9619	0.0203	0.9546	0.0243
clean1	0.8567	0.0156	0.8631	0.0112
semeion	0.9705	0.0035	0.9704	0.0039
Colon	0.7805	0.0301	0.7938	0.0433
Leukemia	0.8994	0.0412	0.9156	0.0366
TOX-171	0.7516	0.0343	0.7625	0.0324

Table 8: Number of features for BSSA-RWN versus TVBSSA-RWN

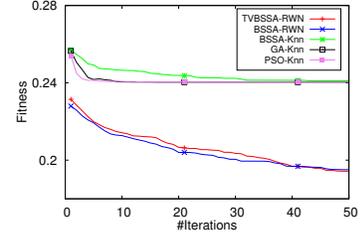
Dataset	BSSA-RWN		TVBSSA-RWN	
	Avg	Std	Avg	Std
BreastEW	16.3600	0.7403	16.7600	1.7401
Exactly	12.8800	0.3271	12.8200	0.2864
Exactly2	13.9200	0.6419	14.5400	0.8849
HeartEW	10.8800	1.3773	10.5200	0.9365
Lymphography	13.3800	1.8580	13.1000	1.1336
M-of-n	12.8000	0.6519	12.6400	0.5320
penglungEW	124.0200	12.6458	107.7200	15.7311
SonarEW	40.9800	2.0705	39.4800	1.9202
SpectEW	16.3000	1.7776	15.9800	0.4764
CongressEW	11.0800	1.2071	10.6400	1.5662
KrvskpEW	34.1600	1.0455	34.8000	1.6248
Tic-tac-toe	14.6800	0.2490	14.8200	0.4324
Vote	10.7000	1.0173	10.0400	1.6009
WaveformEW	32.5800	0.9149	33.5600	1.1781
WineEW	7.6200	0.3701	7.0800	0.4658
clean1	115.4200	3.3214	127.5200	3.9442
semeion	162.1800	8.5377	160.5800	2.8648
Colon	1188.9000	87.6461	1218.5000	35.8552
Leukemia	4022.1600	211.6794	4171.2600	405.7624
TOX-171	3392.6400	142.5020	3353.4400	124.2197



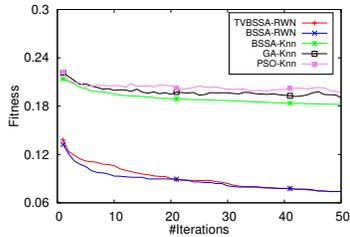
(a) BreastEW



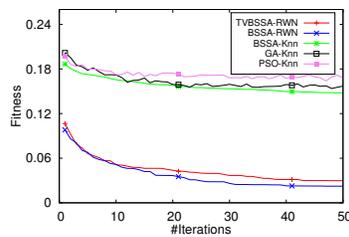
(b) Exactly



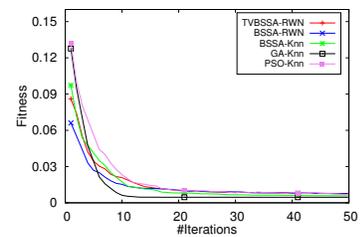
(c) Exactly2



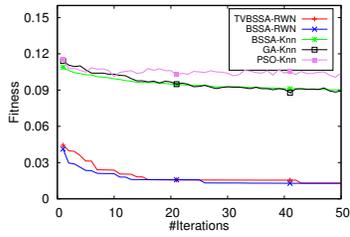
(d) HeartEW



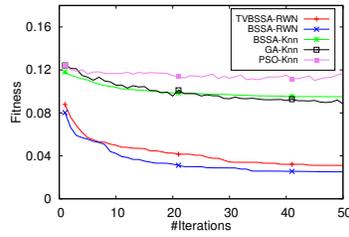
(e) Lymphography



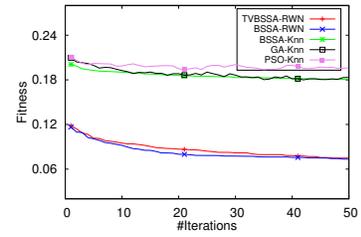
(f) M-of-n



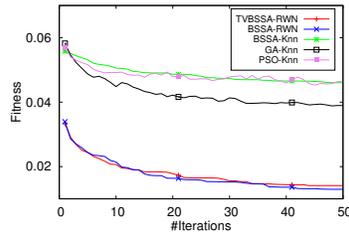
(g) penglungEW



(h) SonarEW

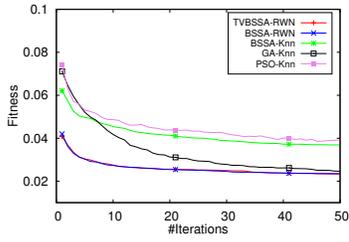


(i) SpectEW

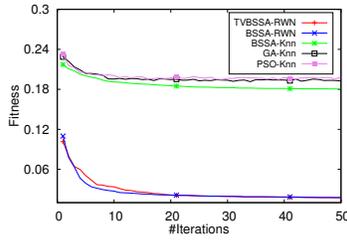


(j) CongressEW

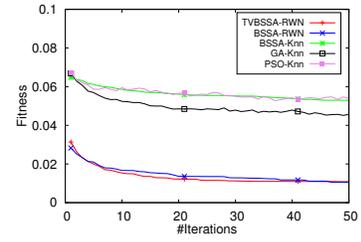
Figure 7: Convergence curves for TVBSSA-RWN and other methods for BreastEW, Exactly, Exactly2, HeartEW, Lymphography, M-of-n, PenglungEW, SonarEW, SpectEW, and CongressEW benchmark datasets.



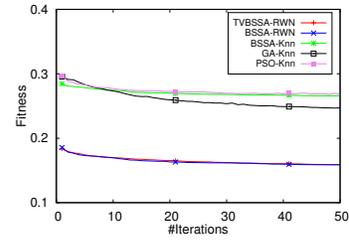
(a) KrvskepEW



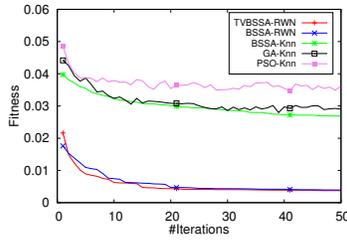
(b) Tic-tac-toe



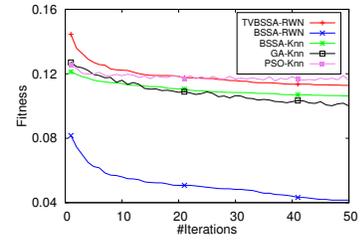
(c) Vote



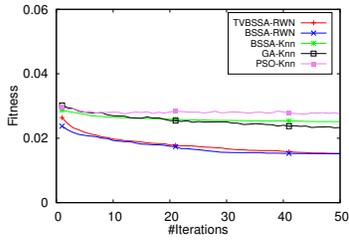
(d) WaveformEW



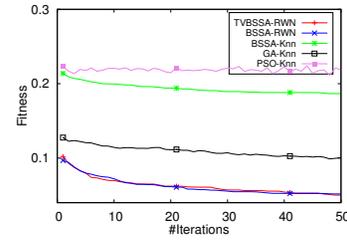
(e) WineEW



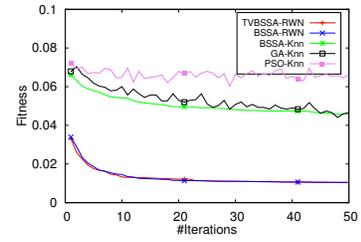
(f) Clean1



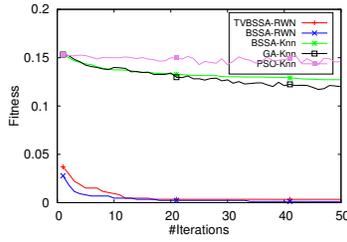
(g) Semeion



(h) Colon



(i) Leukemia



(j) TOX-171

Figure 8: Convergence curves for TVBSSA-RWN and other methods for KrvskepEW, Tic-tac-toe, Vote, WaveformEW, WineEW, Clean1, Semeion, Colon, Leukemia, and TOX-171 datasets.

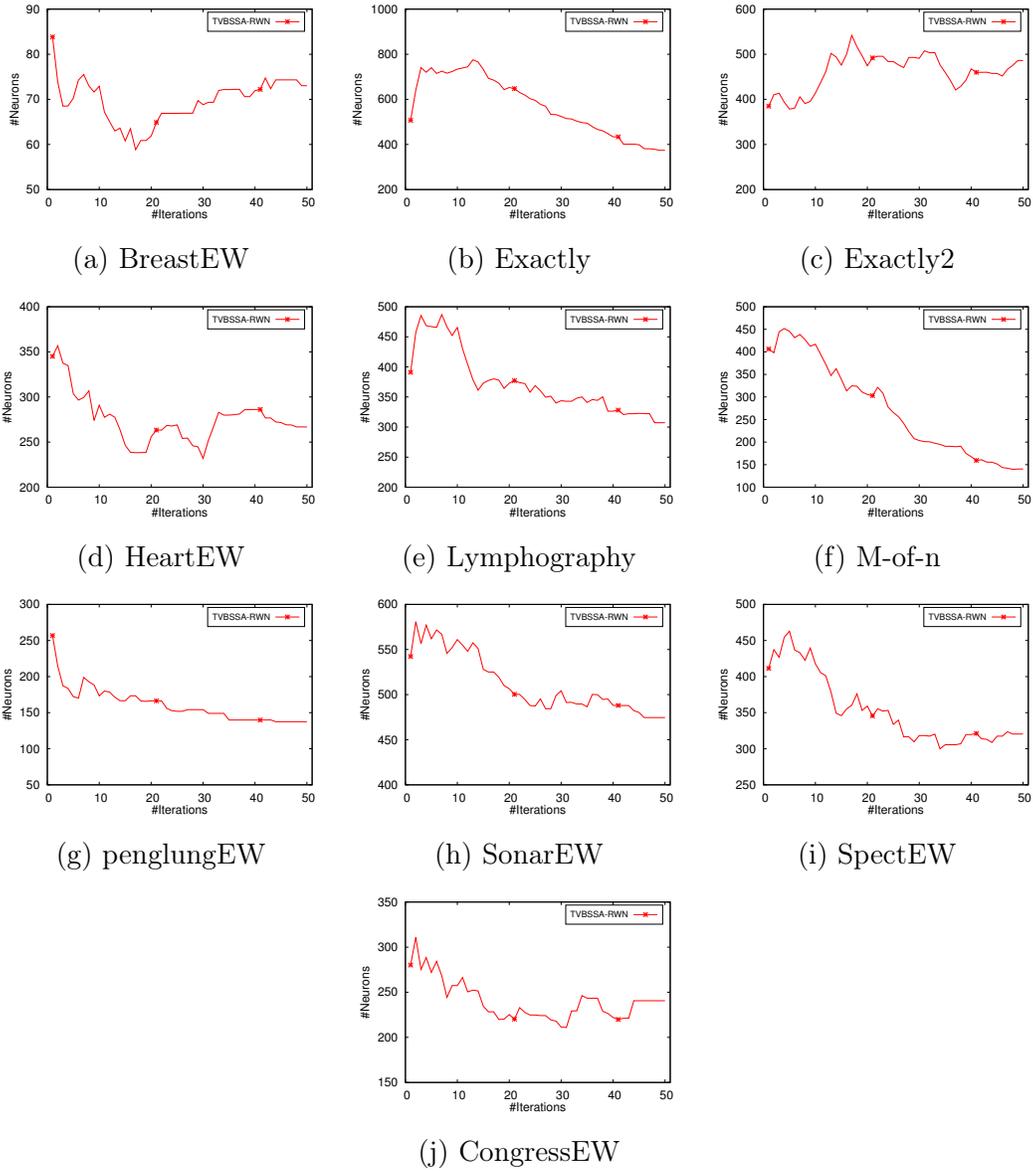


Figure 9: Average number of neurons over the course of iterations in TVBSSA-RWN.

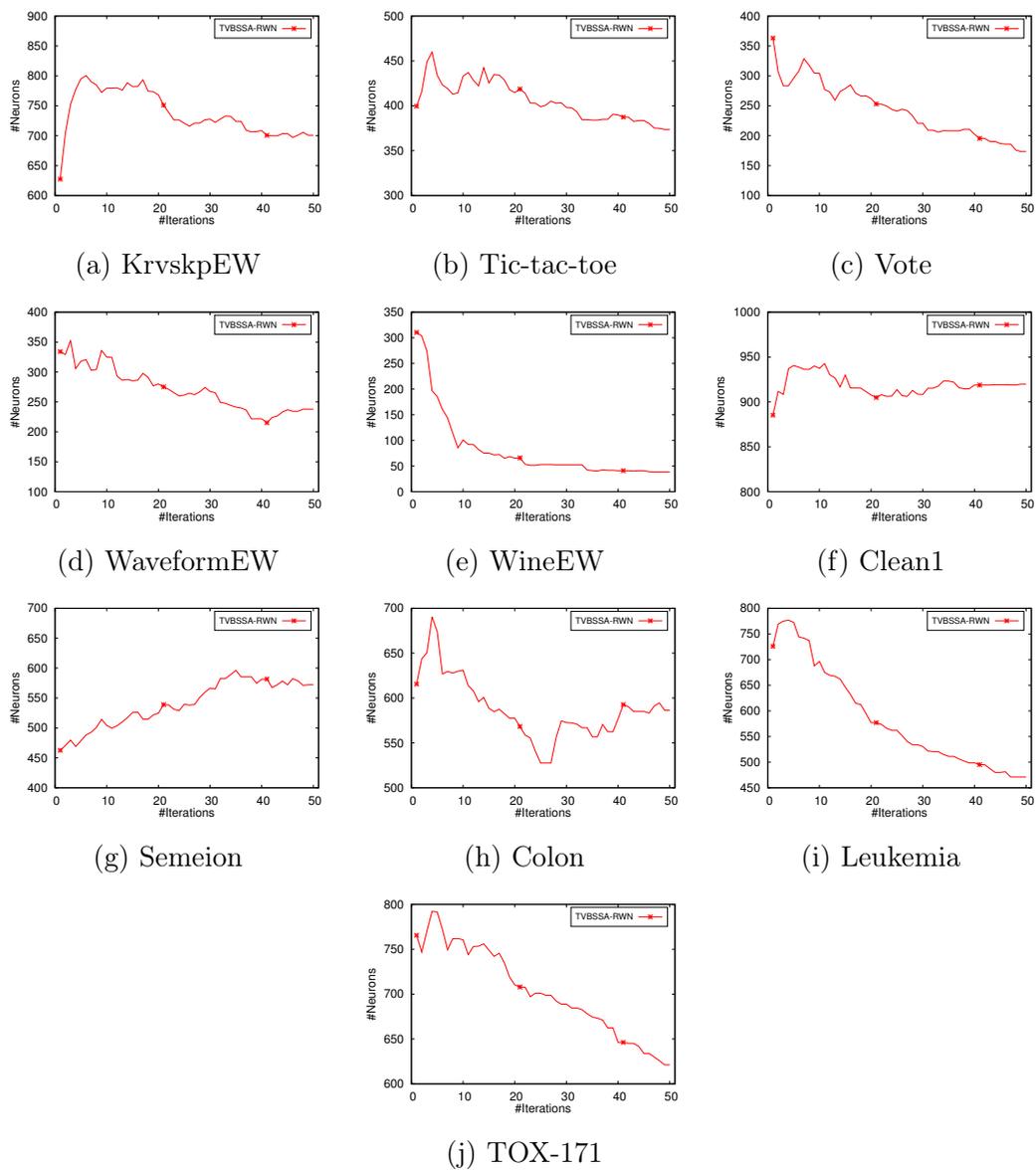


Figure 10: Average number of neurons over the course of iterations in TVBSSA-RWN.

Table 9: P-values of the Wilcoxon test for the classification accuracy results of the TVBSSA-RWN versus other approaches ($p \geq 0.05$ are underlined).

Dataset	BSSA-kNN	GA-kNN	PSO-kNN
BreastEW	1.32E-5	1.15E-7	2.92E-6
Exactly	1.59E-5	4.04E-3	1.75E-6
Exactly2	2.48E-34	8.80E-36	1.90E-34
HeartEW	3.05E-7	2.17E-9	1.49E-9
Lymphography	2.08E-8	2.73E-6	3.60E-11
M-of-n	6.09E-4	1.68E-7	2.64E-3
penglungEW	9.86E-6	1.05E-8	4.00E-11
SonarEW	<u>9.97E-1</u>	<u>8.72E-1</u>	<u>7.77E-1</u>
SpectEW	2.24E-14	1.21E-12	2.13E-15
CongressEW	1.35E-5	4.72E-6	4.60E-6
KrvskpEW	3.71E-15	5.24E-7	6.96E-18
Tic-tac-toe	1.92E-34	1.92E-34	1.88E-34
Vote	4.14E-7	2.07E-9	3.97E-6
WaveformEW	1.68E-30	3.07E-23	7.02E-29
WineEW	9.52E-3	8.65E-3	1.32E-2
Clean1	<u>2.90E-1</u>	<u>3.16E-1</u>	<u>1.93E-1</u>
semeion	<u>8.76E-1</u>	<u>2.77E-1</u>	<u>6.04E-1</u>
Colon	<u>2.87E-1</u>	5.86E-3	4.65E-2
Leukemia	2.41E-3	<u>7.28E-2</u>	3.45E-2
TOX-171	<u>8.07E-1</u>	<u>6.25E-1</u>	4.17E-2

438 Search Algorithm (BGSA), and Binary Bat Algorithm (BBA) Mafarja et al. (2018b).

439 As can be seen in Table 10, the proposed TVBSSA-RWN can reach to highest accuracy
440 rates compared to previous methods on around 78% of datasets. The proposed TVBSSA-
441 RWN have shown a significant superiority over GWO, GA, and PSO algorithms on several
442 datasets. The accuracy rates of the TVBSSA-RWN are better than the rates of GA, binary
443 PSO, and binary GWO in Emary et al. (2016b) on all datasets. It also provides superior
444 results compared to GA1-based FS method in Kashef and Nezamabadi-pour (2015) for 92%
445 of datasets, while it outperforms the PSO1-based technique for 85% of datasets. These re-
446 sults also indicate the significant role of time-varying hierarchy and merits of RWN instead
447 of K-NN in enriching the exploitation and exploitation inclinations and alleviating the stag-
448 nation drawbacks of the BSSA. The comparative study with previous works show that the
449 proposed mechanisms have enhanced the quality of the resulted feature sets in dealing with
450 different cases.

451 All in all, we can summarize the advantages of the proposed TVBSSA-RWN algorithm
452 upon the well-known wrapper based feature selection algorithms as follows: first, it utilizes
453 the model of salp chains in SSA for the search process in the wrapper which proved to
454 be very efficient different complex optimization problems. Second, and unlike most of the
455 well-known feature selection algorithms, TVBSSA-RWN incorporates RWN as a powerful
456 induction algorithm. The experiments reveal that the combination of time varying chain of
457 salps and RWN showed very promising results for the feature selection tasks.

Table 10: Classification accuracies of the TVBSSA-based approach compared to other optimizers from literature.

Dataset	TVBSSA	GA1	PSO1	bGWO1	bGWO2	GA2	PSO2	S-bBOA	BBA	BGSA
Exactly	0.999	0.822	0.973	0.708	0.776	0.674	0.688	0.972	0.610	0.697
Exactly2	0.783	0.677	0.666	0.745	0.750	0.746	0.730	0.760	0.628	0.706
HeartEW	0.865	0.732	0.745	0.776	0.776	0.780	0.787	0.824	0.754	0.777
Lymphography	0.871	0.758	0.759	0.744	0.700	0.696	0.744	0.868	0.701	0.781
M-of-n	0.992	0.916	0.996	0.908	0.963	0.861	0.921	0.972	0.722	0.835
penglungEW	0.758	0.672	0.879	0.600	0.584	0.584	0.584	0.878	0.795	0.919
SonarEW	0.818	0.833	0.804	0.731	0.729	0.754	0.737	0.936	0.844	0.888
SpectEW	0.875	0.756	0.738	0.820	0.822	0.793	0.822	0.846	0.800	0.783
CongressEW	0.964	0.898	0.937	0.935	0.938	0.932	0.928	0.959	0.872	0.951
KrvskpEW	0.982	0.940	0.949	0.944	0.956	0.920	0.941	0.966	0.816	0.908
Tic-tac-toe	0.982	0.764	0.750	0.728	0.727	0.719	0.735	0.798	0.665	0.753
Vote	0.961	0.808	0.888	0.912	0.920	0.904	0.904	0.965	0.851	0.931
WaveformEW	0.831	0.712	0.732	0.786	0.789	0.773	0.762	0.743	0.669	0.695
WineEW	0.955	0.947	0.937	0.930	0.920	0.937	0.933	0.984	0.919	0.951
W T L	9 0 5	0 0 14	1 0 13	0 0 14	0 0 14	0 0 14	0 0 14	3 0 11	0 0 14	1 0 13

6. Conclusion and future directions

In this work, an improved binary SSA was utilized as a search strategy combined with RWN classifier as an induction algorithm to design an improved wrapper FS method (called TVBSSA). In TVBSSA, a dynamic strategy was adopted to control the number of leaders and followers and replace the static strategy in the original SSA. Three objectives were utilized in the fitness function; maximizing the classification accuracy, maximizing the reduction rates, and minimizing the complexity of RWN classifier. The extensive experiments using 14 well-regarded benchmark datasets revealed that the developed approach beat other peers from the literature in terms of prediction power. Based on the Wilcoxon statistical test, TVBSSA significantly outperformed other approaches in terms of the classification accuracy.

The main conclusion drawn from the results and findings is the merits of the proposed method in handling the challenges that an optimization algorithm faces when solving feature selection problems. The proposed method is worthy of integration in different machine learning systems to find an optimal set of features.

However, it is important to note that this work comes with limitations. First, all datasets that were used in this work range from small to large datasets, however, nowadays there are many applications with much larger scale worth of investigation based on frameworks specialized for this type of datasets with extremely high-dimensional feature space. Moreover, we investigated only binary classification problems. Other types of machine learning tasks like multiclass classification problems and regression problems have additional issues at the level of design and formulation that need to be carefully addressed.

As a future direction, researchers can adopt other dynamic strategies to control other parameters in SSA, especially c_3 parameter which controls the balance between exploration and exploitation phases. Moreover, it would be very interesting to handle the problem of optimization in this work as a multi-objective problem by separating the three objectives with the use of the dominance relation to finding the best feature subsets. Also, the findings of this work motivate us to investigate a parallel implementation of the salp chains model to

References

- Agrafiotis, D. K., Cedeno, W., 2002. Feature selection for structure- activity correlation using binary particle swarms. *Journal of medicinal chemistry* 45 (5), 1098–1107.
- Al-Zoubi, A. M., Faris, H., Hassonah, M. A., et al., 2018. Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts. *Knowledge-Based Systems* 153, 91–104.
- Alba, E., Garcia-Nieto, J., Jourdan, L., Talbi, E.-G., 2007. Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE*, pp. 284–290.
- Arora, S., Anand, P., 2019. Binary butterfly optimization approaches for feature selection. *Expert Systems with Applications* 116, 147–160.
- Cao, W., Wang, X., Ming, Z., Gao, J., 2018. A review on neural networks with random weights. *Neurocomputing* 275, 278–287.
- Chen, D., Chan, K. C., Wu, X., 2008. Gene expression analyses using genetic algorithm based hybrid approaches. In: *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE*, pp. 963–969.
- Cho, H.-W., Kim, S. B., Jeong, M. K., Park, Y., Ziegler, T. R., Jones, D. P., 2008. Genetic algorithm-based feature selection in high-resolution nmr spectra. *Expert systems with applications* 35 (3), 967–975.
- Chyzyk, D., Savio, A., Graña, M., 2014. Evolutionary elm wrapper feature selection for alzheimer’s disease cad on anatomical brain mri. *Neurocomputing* 128, 73–80.
- Da Silva, S. F., Ribeiro, M. X., Neto, J. d. E. B., Traina-Jr, C., Traina, A. J., 2011. Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision support systems* 51 (4), 810–820.
- Derrac, J., Garcia, S., Herrera, F., 2009. A first study on the use of coevolutionary algorithms for instance and feature selection. In: *International Conference on Hybrid Artificial Intelligence Systems*. Springer, pp. 557–564.
- Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. *IEEE computational intelligence magazine* 1 (4), 28–39.
- El-Fergany, A. A., 2018. Extracting optimal parameters of pem fuel cells using salp swarm optimizer. *Renewable Energy* 119, 641–648.
- Emary, E., Zawbaa, H. M., Grosan, C., 2018. Experienced gray wolf optimization through reinforcement learning and neural networks. *IEEE transactions on neural networks and learning systems* 29 (3), 681–694.
- Emary, E., Zawbaa, H. M., Hassanien, A. E., 2016a. Binary ant lion approaches for feature selection. *Neurocomputing* 213, 54–65.
- Emary, E., Zawbaa, H. M., Hassanien, A. E., 2016b. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* 172, 371–381.
- Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Al-Zoubi, A. M., Mirjalili, S., Fujita, H., 2018. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems* 154, 43 – 67. URL <http://www.sciencedirect.com/science/article/pii/S0950705118302132>
- Gandomi, A. H., Alavi, A. H., 2012. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation* 17 (12), 4831–4845. URL <http://www.sciencedirect.com/science/article/pii/S1007570412002171>
- Gu, S., Cheng, R., Jin, Y., 2018. Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing* 22 (3), 811–822.
- Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3 (Mar), 1157–1182.
- Han, Y., Yang, Y., Yan, Y., Ma, Z., Sebe, N., Zhou, X., 2015. Semisupervised feature selection via spline regression for video semantic recognition. *IEEE Transactions on Neural Networks and Learning Systems* 26 (2), 252–264.
- Heidari, A. A., Abbaspour, R. A., Jordehi, A. R., 2017. An efficient chaotic water cycle algorithm for optimization tasks. *Neural Computing and Applications* 28 (1), 57–85.
- Holland, J. H., 1992. Genetic algorithms. *Scientific american* 267 (1), 66–73.
- Hong, J.-H., Cho, S.-B., 2006. Efficient huge-scale feature selection with speciated genetic algorithm. *Pattern Recognition Letters* 27 (2), 143–150.
- Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. Vol. 2. IEEE*, pp. 985–990.
- Huang, R., He, M., 2007. Feature selection using double parallel feedforward neural networks and particle swarm optimization. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE*, pp. 692–696.
- Igelnik, B., Pao, Y.-H., 1995. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks* 6 (6), 1320–1329.
- Jain, A., Zongker, D., 1997. Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence* 19 (2), 153–158.
- Jeong, Y.-S., Shin, K. S., Jeong, M. K., 2015. An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems. *Journal of The Operational research society* 66 (4), 529–538.
- Kashef, S., Nezamabadi-pour, H., 2013. A new feature selection algorithm based on binary ant colony optimization. In: *Information and Knowledge Technology (IKT), 2013 5th Conference on. IEEE*, pp. 50–54.
- Kashef, S., Nezamabadi-pour, H., 2015. An advanced aco algorithm for feature subset selection. *Neurocomputing* 147, 271–279.
- Kennedy, J., Eberhart, R. C., 1997. A discrete binary version of the particle swarm algorithm. In: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. Vol. 5. IEEE*, pp. 4104–4108.

- Khushaba, R. N., Al-Ani, A., AlSukker, A., Al-Jumaily, A., 2008. A combined ant colony and differential evolution feature selection algorithm. In: International Conference on Ant Colony Optimization and Swarm Intelligence. Springer, pp. 1–12.
- Kohavi, R., John, G. H., 1997. Wrappers for feature subset selection. *Artificial intelligence* 97 (1-2), 273–324.
- Li, R., Lu, J., Zhang, Y., Zhao, T., 2010. Dynamic adaboost learning with feature selection based on parallel genetic algorithm for image annotation. *Knowledge-Based Systems* 23 (3), 195–201.
- Lichman, M., 2013. UCI machine learning repository.
URL <http://archive.ics.uci.edu/ml>
- Liu, H., Motoda, H., 2012. Feature selection for knowledge discovery and data mining. Vol. 454. Springer Science & Business Media.
- Liu, H., Setiono, R., 1995. Chi2: Feature selection and discretization of numeric attributes.
- Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., Al-Zoubi, A. M., Mirjalili, S., 2018a. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems* 145, 25 – 45.
URL <http://www.sciencedirect.com/science/article/pii/S0950705117306159>
- Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Ala'M, A.-Z., Mirjalili, S., 2018b. Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems* 145, 25–45.
- Mafarja, M., Jarrar, R., Ahmad, S., Abusnaina, A., 2018c. Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In: The 2nd International Conference on Future Networks & Distributed Systems , Amman, Jordan. Vol. 2. ACM.
- Mafarja, M., Mirjalili, S., 2018. Whale optimization approaches for wrapper feature selection. *Applied Soft Computing* 62, 441–453.
- Mafarja, M., Sabar, N. R., 2018. Rank based binary particle swarm optimisation for feature selection in classification. In: The 2nd International Conference on Future Networks & Distributed Systems , Amman, Jordan. Vol. 2. ACM.
- Mafarja, M. M., Mirjalili, S., 2017. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neuro-computing* 260, 302–312.
- Mirjalili, S., 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications* 27 (4), 1053–1073.
URL <http://dx.doi.org/10.1007/s00521-015-1920-1>
- Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., Mirjalili, S. M., 2017. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* 114, 163–191.
- Mirjalili, S., Lewis, A., 2013. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation* 9, 1–14.
- Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. *Advances in Engineering Software* 95, 51–67.
- Mirjalili, S., Mirjalili, S. M., Lewis, A., 2014. Grey wolf optimizer. *Advances in Engineering Software* 69, 46–61.
- Nakamura, R. Y. M., Pereira, L. A. M., Rodrigues, D., Costa, K. A. P., Papa, J. P., Yang, X.-S., 2013. Binary bat algorithm for feature selection. In: *Swarm Intelligence and Bio-Inspired Computation*. Elsevier, pp. 225–237.
- Nie, F., Huang, H., Cai, X., Ding, C. H., 2010. Efficient and robust feature selection via joint l_2, l_1 -norms minimization. In: *Advances in neural information processing systems*. pp. 1813–1821.
- Nie, F., Xiang, S., Jia, Y., Zhang, C., Yan, S., 2008. Trace ratio criterion for feature selection. In: *AAAI*. Vol. 2. pp. 671–676.
- Nie, F., Zhu, W., Li, X., et al., 2016. Unsupervised feature selection with structured graph optimization. In: *AAAI*. pp. 1302–1308.
- Oh, I.-S., Lee, J.-S., Moon, B.-R., 2004. Hybrid genetic algorithms for feature selection. *IEEE Transactions on pattern analysis and machine intelligence* 26 (11), 1424–1437.
- Oreski, S., Oreski, G., 2014. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert systems with applications* 41 (4), 2052–2064.
- Pacifico, L. D., Ludermit, T. B., 2013. Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies. In: *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, pp. 1–6.
- Pao, Y.-H., Park, G.-H., Sobajic, D. J., 1994. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6 (2), 163–180.
- Quinlan, J., 1986. Induction of decision trees. *Machine learning* 1 (1), 81–106.
- Quinlan, J., 1993. *C4.5: programs for machine learning*. Morgan kaufmann.
- Ramos, C. C., Souza, A. N., Chiachia, G., Falcão, A. X., Papa, J. P., 2011. A novel algorithm for feature selection using harmony search and its application for non-technical losses detection. *Computers & Electrical Engineering* 37 (6), 886–894.
- Rashedi, E., Nezamabadi-pour, H., 2014. Feature subset selection using improved binary gravitational search algorithm. *Journal of Intelligent & Fuzzy Systems* 26 (3), 1211–1221.
- Robnik-ikonja, M., Kononenko, I., 2003. Theoretical and empirical analysis of relief and rrelief. *Machine learning* 53 (1), 23–69.
- Rodrigues, D., Pereira, L. A., Almeida, T., Papa, J. P., Souza, A., Ramos, C. C., Yang, X.-S., 2013. Bcs: A binary cuckoo search algorithm for feature selection. In: *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE, pp. 465–468.
- Sayed, G. I., Khoriba, G., Haggag, M. H., 2018. A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence*, 1–20.
- Schmidt, W. F., Kraaijveld, M. A., Duin, R. P., 1992. Feedforward neural networks with random weights. In: *Pattern Recognition, 1992. Vol. II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on*. IEEE, pp. 1–4.
- Seo, J.-H., Lee, Y. H., Kim, Y.-H., 2014. Feature selection for very short-term heavy rainfall prediction using evolutionary computation. *Advances in Meteorology* 2014.

- Shoorehdeli, M. A., Teshnehlab, M., Moghaddam, H. A., 2006. Feature subset selection for face detection using genetic algorithms and particle swarm optimization. In: Networking, Sensing and Control, 2006. ICNSC'06. Proceedings of the 2006 IEEE International Conference on. IEEE, pp. 686–690.
- Shrivastava, P., Shukla, A., Vepakomma, P., Bhansali, N., Verma, K., 2017. A survey of nature-inspired algorithms for feature selection to identify parkinson's disease. *Computer methods and programs in biomedicine* 139, 171–179.
- Shunmugapriya, P., Kanmani, S., 2017. A hybrid algorithm using ant and bee colony optimization for feature selection and classification (ac-abc hybrid). *Swarm and Evolutionary Computation* 36, 27–36.
- Souza, F., Matias, T., Araújo, R., 2011. Co-evolutionary genetic multilayer perceptron for feature selection and model design. In: Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on. IEEE, pp. 1–7.
- Talbi, E.-G., 2009. *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons.
- Tan, F., Fu, X., Zhang, Y., Bourgeois, A. G., 2008. A genetic algorithm-based method for feature subset selection. *Soft Computing* 12 (2), 111–120.
- Tang, E. K., Suganthan, P. N., Yao, X., 2005. Feature selection for microarray data using least squares svm and particle swarm optimization. In: Computational Intelligence in Bioinformatics and Computational Biology, 2005. CIBCB'05. Proceedings of the 2005 IEEE Symposium on. IEEE, pp. 1–8.
- Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.
- Uner, A., Murat, A., Chinnam, R. B., 2011. mr2pso: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences* 181 (20), 4625–4641.
- Winkler, S. M., Affenzeller, M., Jacak, W., Stekel, H., 2011. Identification of cancer diagnosis estimation models using evolutionary algorithms: a case study for breast cancer, melanoma, and cancer in the respiratory system. In: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation. ACM, pp. 503–510.
- Wolpert, D. H., Macready, W. G., 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1 (1), 67–82.
- Xue, B., Zhang, M., Browne, W. N., 2012. New fitness functions in binary particle swarm optimisation for feature selection. In: Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, pp. 1–8.
- Xue, B., Zhang, M., Browne, W. N., 2013. Novel initialisation and updating mechanisms in pso for feature selection in classification. In: European Conference on the Applications of Evolutionary Computation. Springer, pp. 428–438.
- Yan, Z., Yuan, C., 2004. Ant colony optimization for feature selection in face recognition. In: Biometric Authentication. Springer, pp. 221–226.
- Yang, H., Yi, J., Zhao, J., Dong, Z., 2013. Extreme learning machine based genetic algorithm and its application in power system economic dispatch. *Neurocomputing* 102, 154–162.
- Yang, J., Honavar, V., 1998. Feature subset selection using a genetic algorithm. In: Feature extraction, construction and selection. Springer, pp. 117–136.
- Yang, W.-A., Zhou, Q., Tsui, K.-L., 2016. Differential evolution-based feature selection and parameter optimisation for extreme learning machine in tool wear estimation. *International Journal of Production Research* 54 (15), 4703–4721.
- Yang, X.-S., 2009. *Firefly Algorithms for Multimodal Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169–178. URL https://doi.org/10.1007/978-3-642-04944-6_14
- Yusta, S. C., 2009. Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters* 30 (5), 525–534.
- Zainuddin, Z., Lai, K. H., Ong, P., 2016. An enhanced harmony search based algorithm for feature selection: Applications in epileptic seizure detection and prediction. *Computers & Electrical Engineering* 53, 143–162.
- Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A., Liu, H., 2010. Advancing feature selection research. ASU feature selection repository, 1–28.