Salp Swarm Algorithm: Theory, Literature Review, and Application in Extreme Learning Machines

Chapter *in* Studies in Computational Intelligence · January 2020

CITATIONS
87

READS
6,337

4 authors:

Hossam Faris
University of Jordan
220 PUBLICATIONS **13,683** CITATIONS
SEE PROFILE

Ibrahim Aljarah
University of Jordan
157 PUBLICATIONS **10,213** CITATIONS
SEE PROFILE

Majdi Mafarja
Birzeit University
102 PUBLICATIONS **8,099** CITATIONS
SEE PROFILE

Ali Asghar Heidari
National University of Singapore
204 PUBLICATIONS **13,800** CITATIONS
SEE PROFILE

Salp Swarm Algorithm: Theory, Literature Review, and Application in Extreme Learning Machines

Hossam Faris, Seyedali Mirjalili, Ibrahim Aljarah, Majdi Mafarja and Ali Asghar Heidari

Abstract Salp Swarm Algorithm (SSA) is a recent metaheuristic inspired by the swarming behavior of salps in oceans. SSA has demonstrated its efficiency in various applications since its proposal. In this chapter, the algorithm, its operators, and some of the remarkable works that utilized this algorithm are presented. Moreover, the application of SSA in optimizing the Extreme Learning Machine (ELM) is investigated to improve its accuracy and overcome the shortcomings of its conventional training method. For verification, the algorithm is tested on 10 benchmark datasets and compared to two other well-known training methods. Comparison results show that SSA based training methods outperforms other methods in terms of accuracy and is very competitive in terms of prediction stability.

1 Introduction

Artificial Neural Networks (ANNs) are mathematical models that are widely applied in machine learning for supervised and unsupervised tasks [3, 23, 26, 27, 38]. ANNs can be distinguished based on their architecture and the learning algorithm.

H. Faris · I. Aljarah
King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan
e-mail: hossam.faris@ju.edu.jo

I. Aljarah
e-mail: i.aljarah@ju.edu.jo

S. Mirjalili (✉)
Institute of Integrated and Intelligent Systems, Griffith University, Nathan, Brisbane, QLD 4111, Australia
e-mail: seyedali.mirjalili@griffithuni.edu.au

M. Mafarja
Department of Computer Science, Faculty of Engineering and Technology, Birzeit University, PoBox 14, Birzeit, Palestine
e-mail: mmafarja@birzeit.edu

A. A. Heidari
School of Surveying and Geospatial Engineering, University of Tehran, Tehran, Iran
e-mail: as_heidari@ut.ac.ir

S. Mirjalili et al. (eds.), *Nature-Inspired Optimizers*, Studies in Computational Intelligence 811, https://doi.org/10.1007/978-3-030-12127-3_11

The architecture of the network refers to the way that their basic processing elements are connected and distributed over a number of layers. While the learning algorithm is responsible for optimizing the connection weights of the network in order to minimize or maximize some predefines criteria [4, 25, 28].

The most popular type of ANN in the literature is the Single Hidden Layer Feedforward Networks (SLFN). SLFN is typically trained by the Back-propagation algorithm, which is a gradient descent training method. In spite of its popularity, this algorithm has some drawbacks such as the high sensitivity to the initial weights, high probability to be trapped in a local minima, slow convergence, and the need to carefully set its parameters like the learning rate and momentum [17, 32, 33, 62].

In an attempt to overcome the aforementioned problems, Extreme Learning Machine (ELM) was proposed by Huang et al. in [42]. ELM is extremely fast training method for SLFN which starts by initializing the input weights and hidden biases randomly, then it calculates the output weights analytically using Moore-Penrose (MP) generalized inverse. ELMs have several advantages over the traditional gradient descent based neural networks. ELM advantages include: (1) they are extremely fast to train, (2) no need for human intervention and extra effort for tuning some parameters like the learning rate and momentum such as gradient descent algorithms. Due to these advantages, ELM became very popular in the last decade with wide range of successful applications [21].

Although ELM has promising generalization performance and it eliminates the need for tuning some parameters like in the Back-propagation case, it showed tendency to toward the need for higher number of processing elements in the hidden layer [16]. In addition, the random initialization of the input weights in ELM could affect its generalization performance [20, 57].

In the last few years, different approaches were proposed to improve the performance of ELM networks. One noticeable research line is the deployment of evolutionary and swarm based intelligence algorithms to optimize the input weights and biases of ELM. Swarm intelligence techniques are a hot subject of studies to be used in many trending applications such as machine learning [2, 30, 52], global mathematical optimization [9, 37–39, 41], spatial problems [35, 36, 36, 40], spam and intrusion detection systems [6, 8, 24, 28], feature selection [30, 47–54], clustering analysis [1, 5, 7, 63], optimizing neural networks [3, 4, 23, 26], link prediction [12], software effort estimation [31], and bio-informatics [10, 15]. Moreover, swarm intelligence techniques are used in training ELM such as Differential Evolution (DE) and Particle Swarm Optimization (PSO) [34, 60, 65, 66].

In this work, a new training method for ELM is proposed based on a recent nature-inspired optimization algorithm called Salp Swarm Algorithm (SSA) [55]. SSA is deployed to evolve the input connection weights and hidden biases of the single hidden layer feedforward network of ELM. We will refer to this algorithm as SSA-ELM. SSA has shown recently very promising results when applied for complex engineering machine learning problems [55]. This has ignited a motivation for exploring the efficiency of its operator in evolving ELM networks to overcome their shortcomings.

2 Extreme Learning Machines

ELM is a learning framework for single hidden layer feedforward neural networks (SLFN). ELM was introduced by Haung in [43]. The general idea of ELM is to randomly generate the connection weights between the input layer and the hidden layer then the weights that connect the hidden layer to the output layer are analytically computed. The architecture of ELM is shown in Figure.

For N arbitrary distinct samples shown by (x_i,t_i), where $x_i = [x_{i1}, x_{i2}, , x_{iN}]^T \in \mathbb{R}^n$ and $t_i = [t_{i1}, t_{i2}, , t_{iN}]^T \in \mathbb{R}^m$, a standard SLFN with an activation function $g(x)$ and $\tilde{N}$ neurons in hidden layer can be mathematically represented as in Eq. (1) [42]:

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

Where b_i is the threshold of the ith hidden neuron, $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector that connects the ith hidden neuron with the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the vector of weights that connects the ith hidden neuron with output neurons [42].

The conventional SLFNs with $\tilde{N}$ hidden neurons and activation function $g(x)$ are capable of approximating the initial N samples with zero error means that $\sum_{i=1}^{\tilde{N}} \left\| o_j - t_j \right\| = 0$, i.e., we have w_i, β_i, and b_i such that [42]:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\omega_i \cdot x_j + b_i) = t_j,\ j = 1, \ldots, N \tag{2}$$

The aforementioned N rules can be expressed as given in Eq. (3):

$$H\beta = T \tag{3}$$

where

$$H(w_1, \ldots, w_N, b_1, \ldots, b_N, x_1, \ldots, x_N) = \begin{bmatrix} g(w_1.x_1 + b_1) & \ldots & g(w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ \vdots & & \vdots \\ g(w_1.x_N + b_1) & \ldots & g(w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{4}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m}, T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{N \times m} \tag{5}$$

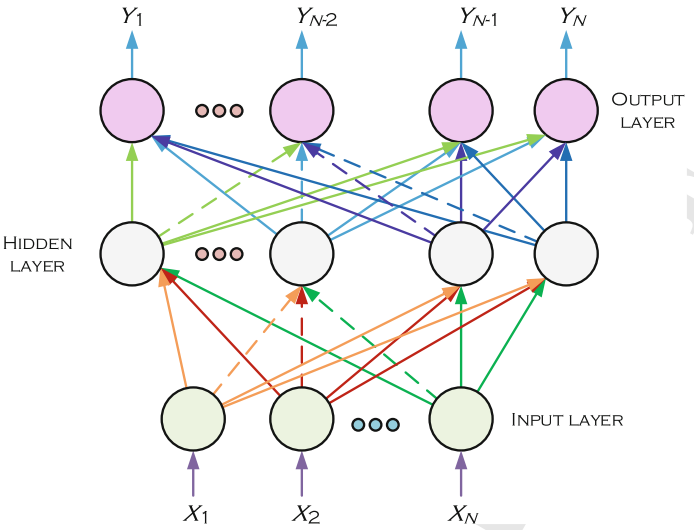


Fig. 1 Overall structure of ELM

where H is the hidden layer output matrix, the ith column of H is the ith hidden output vector of neuron with regard to $x_1, x_2, \ldots, x_N$ [42]. Figure 1 shows the overall structure of the ELM.

3 Salp Swarm Algorithm

The SSA is a recent nature-inspired optimizer proposed by Mirjalili et al. [55] in 2017. The purpose of SSA is to develop a population-based optimizer by mimicking the swarm behavior of salps in nature. The performance of the original SSA as an ELM trainer has not been investigated to date. SSA algorithm reveals satisfactory diversification and intensification propensities that make it appealing for evolving ELM training tasks. The unique advantages of SSA cannot be obtained by using some traditional optimizers such as PSO, GWO, and GSA techniques. The SSA can be considered as a capable, flexible, simple, and easy to be understood and utilized in parallel and serial modes. Furthermore, it has only one adaptively decreasing parameter to make a fine balance between the diversification and intensification inclinations. In order to avoid immature convergence to local optima (LO), the position vectors of salps are gradually updated considering other salps in a dynamic crowd of agents. The dynamic movements of salps enhance the searching capabilities of the SSA in escaping from LO and immature convergence drawbacks. It also keeps the elite salp found so far to guide other members of swarm towards better areas of the feature space.

The SSA has an iterative nature. Hence, it iteratively generates and evolves some random individuals (i.e., salps) inside the bounding box of the problem. Then, all

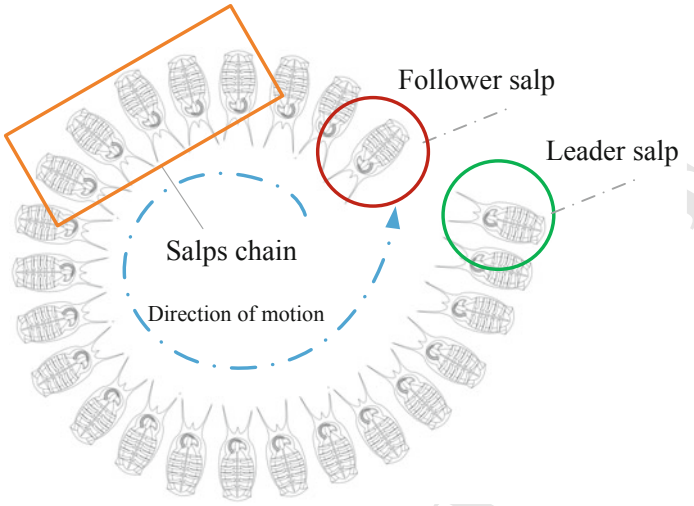


Fig. 2 Illustration of salp's chain and the concept of leader and follower

salps should update their location vectors (leader of chain and followers). The leader salp will attack in the direction of a food source (F), while all followers can move towards the rest of salps (and leader directly or indirectly) [55]. An illustration of salp chain is shown in Fig. 2.

The population of salps X consists of N agents with d-dimensions. Hence, it can be exposed by a $N \times d$-dimensional matrix, as described in Eq. (6):

$$X_i = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \tag{6}$$

In SSA, the position of leader is calculated by Eq. (7):

$$x_j^1 = \begin{cases} F_j + c_1 \left(\left(ub_j - lb_j \right) c_2 + lb_j \right) & c_3 \geq 0.5 \\ F_j - c_1 \left(\left(ub_j - lb_j \right) c_2 + lb_j \right) & c_3 < 0.5 \end{cases} \tag{7}$$

where x_j^1 denotes the leaderś location and F_j reveals the position vector of food source in the jth dimension, ub_j shows the superior limit of jth dimension, and lb_j represents the inferior limit of jth dimension, c_2 and c_3 are random values inside [0, 1], c_1 is the main parameter of algorithm expressed as in Eq. (8):

$$c_1 = 2e^{-(\frac{4t}{T_{max}})^2} \tag{8}$$

where t is the iteration, while T_{max} shows the maximum number of iterations. By increasing iteration count, this parameter decreases. As a result, it can manage to put more emphasize on the diversification inclination on initial stages and put more emphasize on intensification tendency in last steps of optimization. The location of followers are adjusted by Eq. (9):

$$x_j^i = \frac{x_j^i + x_j^{i-1}}{2} \tag{9}$$

where $i \geq 2$ and x_j^i is the location of the ith follower salp at the jth dimension. The pseudo-code of SSA is expressed in Algorithm 1.

Algorithm 1 Pseudo-code of the SSA

```
Initialize the swarm x_i (i = 1, 2, ..., n)
while (end condition is not met) do
    Obtain the fitness of all salps
    Set F as the leader salp
    Update c_1 by Eq. (11.8)
    for (every salp (x_i)) do
        if (i == 1) then
            Update the position of leader by Eq. (11.7)
        else
            Update the position of followers by Eq. (11.9)
    Update the population using the upper and lower limits of variables
    Return back salps that violated the bounding restrictions.
Return F
```

Based on Algorithm 1, we see that SSA first initiates all salps, randomly. Then, it evaluates all salps to select the fittest salp of swarm F. The leader will be followed by the chain of salps as shown in Fig. 2. Meanwhile, the variable c_1 is adjusted by Eq. (8). Equation (7) assists SSA in updating the location of leader, while Eq. (9) can update the position vector of the follower salps. Until the last iteration, all these steps except the initialization step have to be repeated.

4 Literature Review

Since its release, SSA has been applied in various applications in which it showed its efficiency and competitiveness. In this section, noticeable recent works on SSA and their main results are reviewed.

One of the key works on SSA was presented in [22] by Faris et. al to deal with feature selection (FS) tasks. The work proposed two wrapper feature selection approaches based on binary versions of SSA (BSSA). In the first approach, eight transfer functions are used to transform the continuous version of SSA to binary.

In the second approach, a crossover operator replaced the average operator to deepen the exploration tendency in the original algorithm. The proposed approaches were tested using 22 benchmark datasets and the results were compared with other FS methods. The results showed that the proposed SSA-based approaches can significantly outperform other well-established approaches on the majority of the datasets. Another work on FS was conducted by Sayed et al. in [61]. The authors presented a new chaos-based SSA (CSSA) to deal with FS datasets. Simulation results revealed that the CSSA can be regarded as a good optimizer compared to some previous methods. As a specific FS application, SSA was presented in combination with k-nearest neighbors (k-NN) to choose a small number of features and achieve higher classification accuracies in predicting chemical compound activities [44]. Comparison results showed the efficiency of SSA compared to other well-known optimizers.

Ismael et al. [45] used basic SSA to tackle the task of choosing the best conductor in a real-life radial distribution system in Egypt. SSA was applied also for tuning of power system stabilizer in a multi-machine power system [18]. The presented experiments showed that SSA outperformed other intelligent techniques. In [58], SSA is used to optimize the gains and the parameters of the fractional order proportional-integral derivative controller based on an integral time absolute error as objective function. Another application in electric engineering, the authors in [11] proposed the application of SSA for optimizing the sizing of a CMOS differential amplifier and the comparator circuit. Their conducted experiments showed that the performance of the proposed SSA with CMOS analog IC designs are better than other reported studies.

Other engineering problems include optimizing load frequency control using SSA in managing the active power of an isolated renewable microgrid [13], extracting the parameters of polarization curves of polymer exchange membrane fuel cells model [19], design of PID-Fuzzy control for seismic exited structural system against earthquake [14], and solving economic load dispatch problem [59].

SSA was also deployed in environmental applications. In [67], SSA was utilized to optimize the hyper-parameters of the least squares support sector machine for forecasting energy-related CO_2 emissions. The results showed the proposed model is superior and has the potential to improve the forecasting accuracy and reliability of CO_2 emissions.

All these works show that the SSA has a high capability in managing the fine exploration and exploitation tendencies especially for FS tasks. This method can show satisfactory convergence trends, relatively deep exploration and exploitation of the feature space, and flexibility in detecting near-optimal solutions.

5 Proposed SSA-ELM Approach

In this section, we describe the design and procedure of the proposed SSA-ELM training algorithm. In SSA-ELM, the operators of SSA are utilized to evolve ELM networks, where each salp represents a candidate ELM network. To capture this

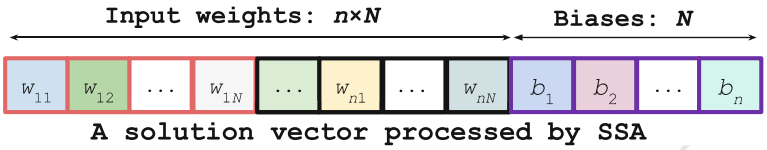


Fig. 3 Structure design of the salp used in the proposed SSA-ELM

representation, the salp is designed to hold the parameters of the network that we want to optimize. In this case, these parameters are the synaptic weights that connect the input layer to the output layer and the biases on the hidden neurons. Therefore, the length (L) of each salp is $I \times N + N$, where I is the number of input variables. The structure design of the salp is represented as shown in Fig. 3.

In order to assess the quality of the generated solutions, a fitness function should be used. In this work we use one of the most common fitness functions which minimization of misclassification rate. This rate can be expressed as given in Eq. 10.

$$f = min(1 - Accuracy) \tag{10}$$

Where the $Accuracy$ is a total number of correctly classified instances over the total number of instances in the training set and it can be calculated as given in Eq. 11.

$$Accuracy = \frac{\sum_{i=1}^{n} \sum_{j=1}^{c} f(i, j) C(i, j)}{n} \tag{11}$$

Where n is the number of instances in the training set, and c is the number of classes. $f(i, j)$ is an indicator function that returns 0 or 1. $f(i, j)$ is 1 if the instance i is of class j. $C(i, j)$ is 1 iff the predicted class of instance i is j, otherwise 0.

6 Experiments and Results

The proposed SSA-ELM approach is evaluated based on 10 benchmark datasets drawn from the University of California at Irvine (UCI) Machine Learning Repository [46]. The datasets are described in terms of number of instances and features in Table 1. The datasets are varied in their specifications to form different levels of complexity for the experimented algorithms. All datasets are normalized using Min-Max normalization method in the range [0, 1] [2, 29, 52, 63].

For verification, SSA-ELM is compared with other two metaheuristic based ELM models commonly used in the literature which are PSO-ELM and DE-ELM. In addition SSA-ELM is compared with the basic ELM network. The parameters of

Table 1 Datasets specifications

Dataset	Instances	Features
Australian	690	14
Blood	748	4
Breast cancer	699	8
Chess	3196	36
German	1000	24
Habitit	155	10
Ring	7400	20
Tictactoe	958	9
Titanic	2201	3
Twonorm	7400	20

Table 2 Parameters of algorithms

Algorithm	Parameter	Value
PSO	Inertia factor	0.2–0.9
	c_1	2
	c_2	2
DE	CR Crossover	0.8
	F Scaling	1

PSO, DE and SSA are set as listed in Table 2. The number of hidden neurons in all algorithms is set as $2 \times F + 1$ where F is the number input features in the dataset. This rule is commonly used in the literature for SLFNs [3, 25, 56, 64]. The number of iterations and population/swarm size in all algorithms is set to 50.

In order to obtain credible results, each algorithm is trained and tested using 10-folds cross validation. Then, the average of the accuracy rates and the standard deviations are calculated and reported along with the best accuracy rates.

The convergence curves of PSO-ELM, DE-ELM and SSA-ELM for all datasets are shown in Figs. 4 and 5. It can be noticed that SSA-ELM has a fast convergence than the other two algorithms in most of the datasets. In Table 3, we list the average accuracy results, standard deviations and the best obtained accuracy rates. Inspecting the results, it can be seen that the SSA-ELM obtained the highest average of accuracies in 9 datasets with a noticeable difference in some datasets like Australian, German, Habitit and Tictactoe datasets. Considering the values of standard deviations, SSA-ELM shows the smallest values or very competitive to the other algorithms. This indicates the stability and robustness of the algorithm. In addition, SSA-ELM hit the highest accuracy rates in 6 datasets.

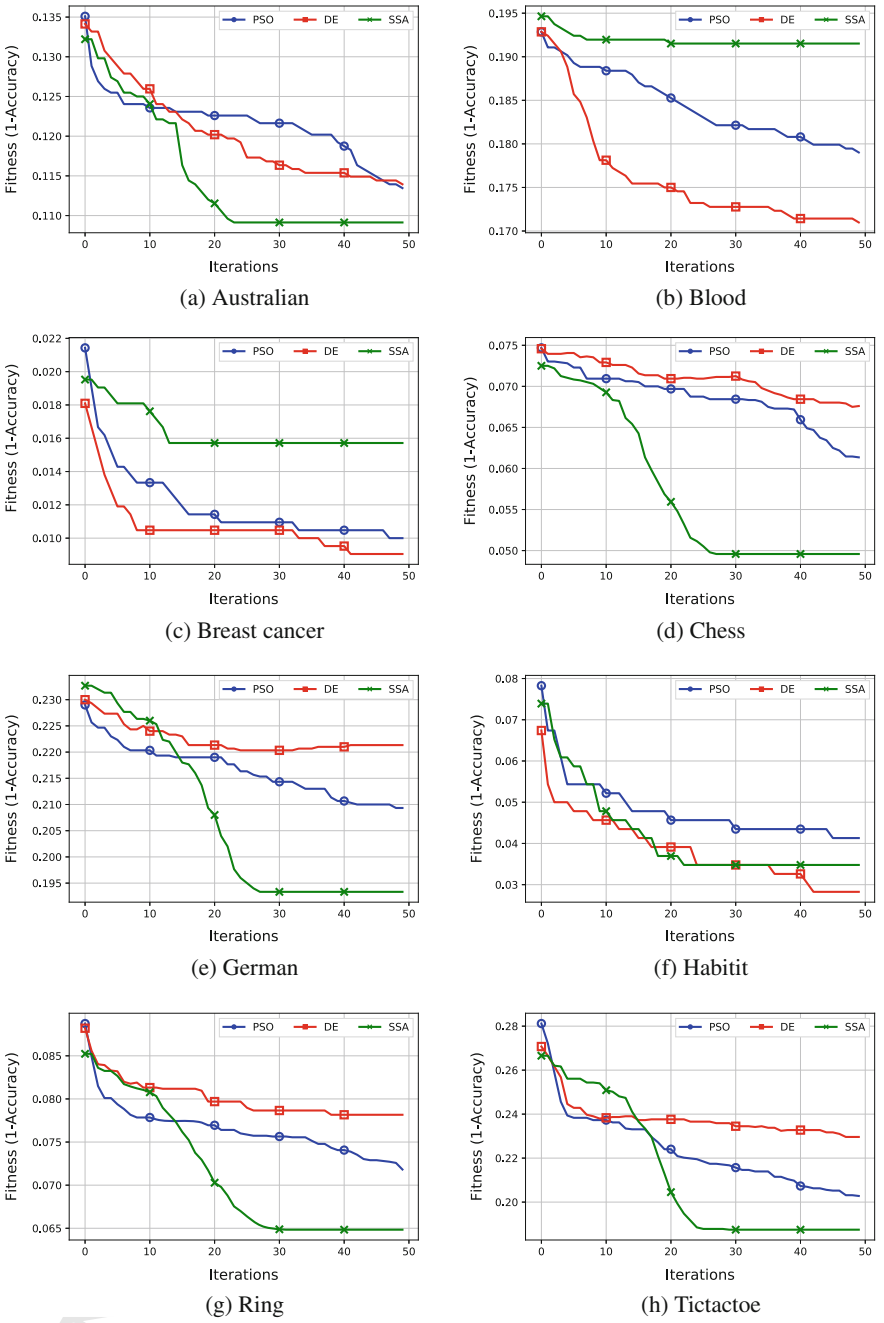
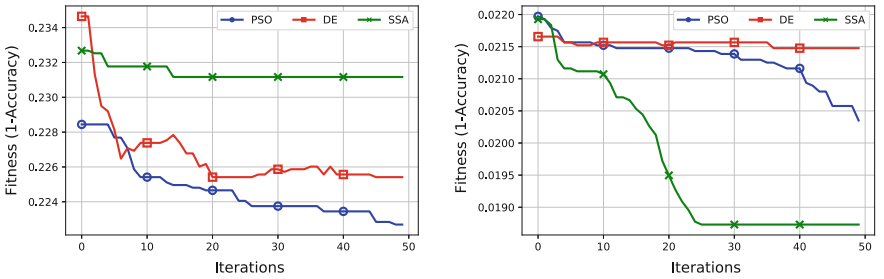


Fig. 4 Convergence curves for SSA, DE and PSO



(a) Titanic (b) Twonorm

Fig. 5 Convergence curves for SSA, DE and PSO

Table 3 Average and best accuracy rates of ELM networks

Dataset	PSO	DE	SSA	ELM
	avg±stdv	avg±stdv	avg±stdv	avg±stdv
	[best]	[best]	[best]	[best]
Australian	85.74 ± 3.54	83.68 ± 3.83	**88.24 ± 3.1**	86.03 ± 2.96
	[92.65]	**[92.65]**	**[92.65]**	[89.71]
Blood	77.97 ± 4.03	79.32 ± 3.06	**79.86 ± 3.02**	78.65 ± 3.24
	[83.78]	[82.43]	**[83.78]**	**[83.78]**
Breast	96.09 ± 2.98	95.94 ± 1.65	**96.52 ± 2.83**	95.94± 2.44
	[100.0]	[98.55]	**[100.0]**	[98.55]
Chess	93.02 ± 1.91	93.18 ± 1.63	**93.27 ± 1.19**	92.61± 1.43
	[95.91]	**[95.91]**	[94.97]	[94.34]
German	74.3 ± 4.4	73.5 ± 3.21	**76.6 ± 2.12**	74.0± 3.62
	[82.0]	[77.0]	[81.0]	[81.0]
Habitit	79.33 ± 8.58	82.0 ± 6.32	**82.0 ± 9.45**	76.67± 9.56
	[93.33]	[93.33]	**[100.0]**	[86.67]
Ring	91.69 ± 1.11	91.6 ± 1.05	**92.02 ± 1.27**	88.57 ± 2.36
	[93.23]	[92.96]	**[93.64]**	[91.07]
Tictactoe	73.37 ± 3.06	72.42 ± 4.98	**75.47 ± 3.44**	68.32 ± 4.72
	[77.9]	[78.95]	**[80.0]**	[74.74]
Titanic	78.18 ± 1.87	78.55 ± 2.39	**78.73 ± 1.78**	77.23 ± 2.54
	[81.36]	**[84.09]**	[81.36]	[79.09]
Twonorm	97.4 ± 0.65	**97.71 ± 0.43**	97.62 ± 0.71	97.4 ± 0.55
	[98.51]	[98.38]	[98.38]	[98.24]

7 Conclusion and Future Directions

In this chapter we reviewed one of the recent promising metaheuristic algorithms called Salp Swarm Algorithm (SSA), which is inspired by the swarming behavior of salps in seas. Moreover, we investigated the efficiency of SSA in training extreme learning machine (ELM). Specifically, SSA is used to optimize the input weights and biases of the hidden layer in ELM. The SSA based training method is experimented using 10 benchmark datasets and compared to two other well-knows methods which are Particle Swarm Optimization and Differential Evolution. Evaluation results show that the proposed methods outperform others in terms of classification accuracy. For future work, it is planned to study the potential of SSA in training other types of neural networks like Radial Basis Function networks (RBFN), Kernel ELM (KELM) and Regularized ELM.

References

1. Al-Madi, N., Aljarah, I., & Ludwig, S. (2014). Parallel glowworm swarm optimization clustering algorithm based on mapreduce. In *IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2014)*. IEEE Xplore Digital Library.
2. Aljarah, I., AlaM, A. Z., Faris, H., Hassonah, M. A., Mirjalili, S., & Saadeh, H. (2018). Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. *Cognitive Computation* pp. 1–18.
3. Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, *22*(1), 1–15.
4. Aljarah, I., Faris, H., Mirjalili, S., & Al-Madi, N. (2018). Training radial basis function networks using biogeography-based optimizer. *Neural Computing and Applications*, *29*(7), 529–553.
5. Aljarah, I., & Ludwig, S. A.: (2012). Parallel particle swarm optimization clustering algorithm based on mapreduce methodology. In *Proceedings of the Fourth World Congress on Nature and Biologically Inspired Computing (IEEE NaBIC12)*. IEEE Explore.
6. Aljarah, I., & Ludwig, S. A. (2013). A mapreduce based glowworm swarm optimization approach for multimodal functions. In *IEEE Symposium Series on Computational Intelligence, IEEE SSCI 2013*. IEEE Xplore.
7. Aljarah, I., & Ludwig, S. A.: A new clustering approach based on glowworm swarm optimization. In *Proceedings of 2013 IEEE Congress on Evolutionary Computation Conference (IEEE CEC13)*, Cancun, Mexico. IEEE Xplore.
8. Aljarah, I., & Ludwig, S. A. (2013). Towards a scalable intrusion detection system based on parallel pso clustering using mapreduce. In *Proceedings of Genetic and Evolutionary Computation Conference (ACM GECCO13) Amsterdam*, July 2013. ACM.
9. Aljarah, I., & Ludwig, S. A. (2016). A scalable mapreduce-enabled glowworm swarm optimization approach for high dimensional multimodal functions. *International Journal of Swarm Intelligence Research (IJSIR)*, *7*(1), 32–54.
10. Aminisharifabad, M., Yang, Q., & Wu, X. (2018). A Penalized Autologistic regression with application for modeling the microstructure of dual-phase high strength steel. *Journal of Quality Technology*, in-press.
11. Asaithambi, S., & Rajappa, M. (2018). Swarm intelligence-based approach for optimal design of cmos differential amplifier and comparator circuit using a hybrid salp swarm algorithm. *Review of Scientific Instruments*, *89*(5), 054702.

12. Barham, R., & Aljarah, I. (2017). Link prediction based on whale optimization algorithm. In *The International Conference on new Trends in Computing Sciences (ICTCS2017), Amman, Jordan*.
13. Barik, A. K., & Das, D. C. (2018). Active power management of isolated renewable microgrid generating power from rooftop solar arrays, sewage waters and solid urban wastes of a smart city using salp swarm algorithm. In *Technologies for Smart-City Energy Security and Power (ICSESP), 2018*. (pp. 1–6). IEEE.
14. Baygi, S. M. H., Karsaz, A., & Elahi, A. (2018). A hybrid optimal pid-fuzzy control design for seismic exited structural system against earthquake: A salp swarm algorithm. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)* (pp. 220–225). IEEE.
15. Chitsaz, H., & Aminisharifabad, M. (2015). Exact learning of rna energy parameters from structure. *Journal of Computational Biology*, *22*(6), 463–473.
16. Cho, J. H., Lee, D. J., & Chun, M. G. (2007). Parameter optimization of extreme learning machine using bacterial foraging algorithm. *Journal of Korean Institute of Intelligent Systems*, *17*(6), 807–812.
17. Ding, S., Su, C., & Yu, J. (2011). An optimizing bp neural network algorithm based on genetic algorithm. *Artificial Intelligence Review*, *36*(2), 153–162.
18. Ekinci, S., & Hekimoglu, B. (2018). Parameter optimization of power system stabilizer via salp swarm algorithm. In *2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)* (pp. 143–147). IEEE.
19. El-Fergany, A. A. (2018). Extracting optimal parameters of pem fuel cells using salp swarm optimizer. *Renewable Energy*, *119*, 641–648.
20. Eshtay, M., Faris, H., & Obeid, N. (2018). Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. *Expert Systems with Applications*, *104*, 134–152.
21. Eshtay, M., Faris, H., & Obeid, N. (2018). Metaheuristic-based extreme learning machines: a review of design formulations and applications. *International Journal of Machine Learning and Cybernetics* (pp. 1–19).
22. Faris, H., Mafarja, M., Heidari, A., Aljarah, I., Al-Zoubi, A., Mirjalili, S., et al. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, *154*, 43–67.
23. Faris, H., Aljarah, I., Al-Madi, N., & Mirjalili, S. (2016). Optimizing the learning process of feedforward neural networks using lightning search algorithm. *International Journal on Artificial Intelligence Tools*, *25*(06), 1650033.
24. Faris, H., Aljarah, I., Al-Shboul, B. (2016). A hybrid approach based on particle swarm optimization and random forests for e-mail spam filtering. In *International Conference on Computational Collective Intelligence* (pp. 498–508). Springer, Cham.
25. Faris, H., Aljarah, I., & Mirjalili, S. (2016). Training feedforward neural networks using multi-verse optimizer for binary classification problems. *Applied Intelligence*, *45*(2), 322–332.
26. Faris, H., Aljarah, I., & Mirjalili, S. (2017). Evolving radial basis function networks using moth–flame optimizer. In *Handbook of Neural Computation* (pp. 537–550).
27. Faris, H., Aljarah, I., & Mirjalili, S. (2017). Improved monarch butterfly optimization for unconstrained global search and neural network training. *Applied Intelligence* pp. 1–20.
28. Faris, H., & Aljarah, I., et al. (2015). Optimizing feedforward neural networks using krill herd algorithm for e-mail spam detection. In *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)* (pp. 1–5). IEEE.
29. Faris, H., Hassonah, M. A., AlaM, A. Z., Mirjalili, S., & Aljarah, I. (2017). A multi-verse optimizer approach for feature selection and optimizing svm parameters based on a robust system architecture. *Neural Computing and Applications* pp. 1–15.
30. Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., AlaM, A. Z., Mirjalili, S., et al. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems*, *154*, 43–67.
31. Ghatasheh, N., Faris, H., Aljarah, I., & Al-Sayyed, R. M. (2015). Optimizing software effort estimation models using firefly algorithm. *Journal of Software Engineering and Applications*, *8*(03), 133.

32. Gori, M., & Tesi, A. (1992). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, *1*, 76–86.
33. Gupta, J. N., & Sexton, R. S. (1999). Comparing backpropagation with a genetic algorithm for neural network training. *Omega*, *27*(6), 679–684.
34. Han, F., Yao, H. F., & Ling, Q. H. (2013). An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing*, *116*, 87–93.
35. Heidari, A. A., Kazemizade, O., & Hakimpour, F. (2017). A new hybrid yin-yang-pair swarm optimization algorithm for uncapacitated warehouse location problems. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-4/W4* (pp. 373–379).
36. Heidari, A. A., Mirvahabi, S. S., & Homayouni, S. (2015). An effective hybrid support vector regression with chaos-embedded biogeography-based optimization strategy for prediction of earthquake-triggered slope deformations. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-1/W5* (pp. 301–305).
37. Heidari, A. A., & Abbaspour, R. A. (2018). Enhanced chaotic grey wolf optimizer for real-world optimization problems: A comparative study. In *Handbook of Research on Emergent Applications of Optimization Algorithms* (pp. 693–727). IGI Global.
38. 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.
39. Heidari, A. A., Abbaspour, R. A., & Jordehi, A. R. (2017). Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems. *Applied Soft Computing*, *57*, 657–671.
40. Heidari, A. A., & Delavar, M. R. (2016). A modified genetic algorithm for finding fuzzy shortest paths in uncertain networks. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLI-B2* (pp. 299–304).
41. Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with lévy flight for optimization tasks. *Applied Soft Computing*, *60*, 115–134.
42. Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks, 2004. Proceedings*. vol. 2 (pp. 985–990). IEEE.
43. Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, *70*(1), 489–501.
44. Hussien, A. G., Hassanien, A. E., & Houssein, E. H. (2017). Swarming behaviour of salps algorithm for predicting chemical compound activities. In *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)* (pp. 315–320). IEEE.
45. Ismael, S., Aleem, S., Abdelaziz, A., & Zobaa, A. (2018). Practical considerations for optimal conductor reinforcement and hosting capacity enhancement in radial distribution systems. IEEE Access.
46. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml.
47. Mafarja, M., & Abdullah, S. (2011). Modified great deluge for attribute reduction in rough set theory. In *2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (vol. 3, pp. 1464–1469). IEEE.
48. Mafarja, M., & Abdullah, S. (2013). Investigating memetic algorithm in solving rough set attribute reduction. *International Journal of Computer Applications in Technology*, *48*(3), 195–202.
49. Mafarja, M., & Abdullah, S. (2013). Record-to-record travel algorithm for attribute reduction in rough set theory. *Journal of Theoretical and Applied Information Technology*, *49*(2), 507–513.
50. Mafarja, M., & Abdullah, S. (2014). Fuzzy modified great deluge algorithm for attribute reduction. In *Recent Advances on Soft Computing and Data Mining* (pp. 195–203). Springer, Cham.
51. Mafarja, M., & Abdullah, S. (2015). A fuzzy record-to-record travel algorithm for solving rough set attribute reduction. *International Journal of Systems Science*, *46*(3), 503–512.
52. Mafarja, M., Aljarah, I., Heidari, A. A., Hammouri, A. I., Faris, H., AlaM, A. Z., et al. (2018). Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. *Knowledge-Based Systems*, *145*, 25–45.

53. Mafarja, M., Jaber, I., Eleyan, D., Hammouri, A., & Mirjalili, S. (2017). Binary dragonfly algorithm for feature selection. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)* (pp. 12–17).
54. Mafarja, M., & Mirjalili, S. (2017). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, *62*, 441–453.
55. 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*.
56. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences*, *269*, 188–209.
57. Mohapatra, P., Chakravarty, S., & Dash, P. K. (2015). An improved cuckoo search based extreme learning machine for medical data classification. *Swarm and Evolutionary Computation*, *24*, 25–49.
58. Mohapatra, T. K., & Sahu, B. K. (2018). Design and implementation of ssa based fractional order pid controller for automatic generation control of a multi-area, multi-source interconnected power system. In *Technologies for Smart-City Energy Security and Power (ICSESP), 2018* (pp. 1–6). IEEE.
59. Reddy, Y. V. K., & Reddy, M. D. Solving economic load dispatch problem with multiple fuels using teaching learning based optimization and salp swarm algorithm. *Zeki Sistemler Teori ve Uygulamaları Dergisi, 1*(1), 5–15.
60. Sánchez-Monedero, J., Hervas-Martinez, C., Gutiérrez, P., Ruz, M. C., Moreno, M. R., & Cruz-Ramirez, M. (2010). Evaluating the performance of evolutionary extreme learning machines by a combination of sensitivity and accuracy measures. *Neural Network World*, *20*(7), 899.
61. Sayed, G. I., Khoriba, G., & Haggag, M. H. (2018). A novel chaotic salp swarm algorithm for global optimization and feature selection. *Applied Intelligence* pp. 1–20.
62. Sexton, R. S., Dorsey, R. E., & Johnson, J. D. (1999). Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *European Journal of Operational Research*, *114*(3), 589–601.
63. Shukri, S., Faris, H., Aljarah, I., Mirjalili, S., & Abraham, A. (2018). Evolutionary static and dynamic clustering algorithms based on multi-verse optimizer. *Engineering Applications of Artificial Intelligence*, *72*, 54–66.
64. Wdaa, A. S. I. (2008). Differential evolution for neural networks learning enhancement. Ph.D. thesis, Universiti Teknologi Malaysia.
65. Xu, Y., & Shu, Y. (2006). Evolutionary extreme learning machine-based on particle swarm optimization. *Advances in Neural Networks-ISNN*, *2006*, 644–652.
66. Yang, Z., Wen, X., Wang, Z. (2015). Qpso-elm: An evolutionary extreme learning machine based on quantum-behaved particle swarm optimization. In *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)* (pp. 69–72). IEEE.
67. Zhao, H., Huang, G., & Yan, N. (2018). Forecasting energy-related co2 emissions employing a novel ssa-lssvm model: Considering structural factors in china. *Energies*, *11*(4), 781.

477504_1_En_11_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:28/12/2018 Pages: 16 Layout: T1-Standard