Training Neural Networks Using Salp Swarm Algorithm for Pattern Classification

Ahmed A. Abusnaina
Department of Computer Science, Birzeit University
Birzeit
aabusnaina@birzeit.edu

Sobhi Ahmad
Department of Computer Science, Birzeit University
Birzeit
sahmed@birzeit.edu

Radi Jarrar
Department of Computer Science, Birzeit University
Birzeit
rjarrar@birzeit.edu

Majdi Mafarja
Department of Computer Science, Birzeit University
Birzeit
mmafarja@birzeit.edu

ABSTRACT

Pattern classification is one of the popular applications of neural networks. However, training the neural networks is the most essential phase. Traditional training algorithms (e.g. Back-propagation algorithm) have some drawbacks such as falling into the local minima and slow convergence rate. Therefore, optimization algorithms are employed to overcome these issues. Salp Swarm Algorithm (SSA) is a recent and novel nature-inspired optimization algorithm that proved a good performance in solving many optimization problems. This paper proposes the use of SSA to optimize the weights coefficients for the neural networks in order to perform pattern classification. The merits of the proposed method are validated using a set of well-known classification problems and compared against rival optimization algorithms. The obtained results show that the proposed method performs better than or on par with other methods in terms of classification accuracy and sum squared errors.

CCS CONCEPTS

• **Computing methodologies** → **Metaheuristic**; **Pattern classification**; **Neural networks**;

KEYWORDS

Salp Swarm Algorithm, Neural Networks, Optimization, Pattern Classification

ACM Reference Format:
Ahmed A. Abusnaina, Sobhi Ahmad, Radi Jarrar, and Majdi Mafarja. 2018. Training Neural Networks Using Salp Swarm Algorithm for Pattern Classification. In *ICFNDS'18: International Conference on Future Networks and Distributed Systems, June 26–27, 2018, Amman, Jordan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3231053.3231070

ICFNDS'18, June 26–27, 2018, Amman, Jordan

ACM ISBN 978-1-4503-6428-7/18/06…$15.00
https://doi.org/10.1145/3231053.3231070

1 INTRODUCTION

Neural Networks (NN) are simplified mathematical approximation of biological neural system in terms of structure and function. NN can perform variety of functions such as identification of objects and patterns (i.e., pattern classification), making decisions based on prior experiences and knowledge and prediction of future events based on past experience [9]. The learning process (i.e., the training) of NN is the most important aspect, where the neurons connections weights are tuned based on a training algorithm in order to minimize the error between the actual and the desired outputs [26].

Gradient Descent (GD) paradigm has been widely used to tune the weights coefficients (i.e. train) of NN. Back-propagation (BP) Levenberg Marquardt (LM) algorithms are a popular examples of GD techniques. GD techniques in general, computes the derivatives of the error function (e.g. mean square error, sum squared error, etc.) to tune the neurons weights. However, GD techniques suffer from the slow convergence rate, instability and the possibility of falling into local minima [14, 23, 27].

Heuristic search paradigm, that aim to provide a sufficiently good solution to an optimization problem in a reasonable time, is considered as an alternative to GD in adjusting the NN weights. In literature, metaheuristic (MH) algorithms -a higher level heuristic- proved their ability to solve several optimization problems like feature selection [17, 19, 20], function optimization [25], supervised training of artificial neural networks [2, 4] and adjusting time-delay components in spiking neural networks [3, 5].

MH algorithms can be classified into two main families; single solution based (S-Metaheuristic) algorithms, that use to manipulate one solution during the optimization process, and population based (P-Metaheuristic) algorithms, that manipulate a pool of solutions (called population) during the optimization process [29].

The strength or the potential of P-Metaheuristic algorithms resides in searching various and multiple zones of the solution space for the same problem simultaneously via a set of individuals (solutions). These characteristics of the P-Metaheuristic matches the nature of the search space of NN weights which is high dimensional, multimodal and probable to be corrupted or to have missing data [4]. According to Mirjalili et al. [22], the stochastic nature of P-Metaheuristic algorithms allows them to avoid local minima and converge to the global minima faster than GD techniques in optimizing the weight coefficients of NN.

In literature, several P-Metaheuristic algorithms have been proposed to optimize the weights coefficients of NN. Those methods were used to overcome the drawbacks of GD techniques. However, the effectiveness of P-Metaheuristic algorithms rely on algorithm settings and problem characteristics [4]. Furthermore, several of these training algorithms were validated using low dimension problems such as datasets with relatively small number of features and patterns (e.g. Iris datasets) or logical operations (e.g. AND, OR, and XOR problems). Consequently, most of them were unable to generalize their superiority against others [14].

Abusnaina and Abdullah [1] proposed the use of Mussels Wandering Optimization (MWO) algorithm to reduce the processing training time with acceptable accuracy. An enhanced version of MWO algorithm to self-adapt the MWO parameters in order to improve the accuracy of NN proposed in [2, 4]. The use of MWO algorithm and its variants has been proposed to adjust the NN connections weights while the NN structure is fixed. The Flower Pollination Algorithm (FPA) has been also proposed to tune the NN parameters [26].

The Genetic Algorithm (GA) proposed by Dorsey et al. [11] optimize the connection weights of NN, but a simple logical operation were used for validation. Schiffmann et al. [24] proposed the use of GA to optimize the NN structure with fixed connection weights. Other P-Metaheuristic algorithms that have been used to optimize the NN connections weights, the NN structure or both include, Evolutionary Strategies (ES) [31], Artificial Bee Colony (ABC) [7] [13], Particle Swarm Optimization (PSO) [28], Ant Colony Optimization (ACO) [10] and the Harmony Search (HS) algorithm [14] [15].

Salp Swarm Algorithm (SSA) is a recent and novel P-Metaheuristic optimization algorithm proposed by Mirjalili et al. [21]. SSA is mainly inspired from the swarming and navigation behavior of salps in oceans. SSA has been applied for solving a large set of mathematical optimization problems. Moreover, the Multi-objective version of SSA was adapted to solve many challenging engineering design problems [21]. The SSA has also proven its high performance in feature selection problems in comparison to other algorithms such as PSO and DE [12]. A chaotic version of SSA was also applied to tackle the feature selection and proved a good performance [6].

This paper proposes using the SSA algorithm for training NN for the first time in literature. NN is used here to perform pattern classification. A set of well-known health datasets were used to assess the performance of the proposed approach. The selection of those datasets is due to their diversity, variety and high dimensionality. The reported results demonstrates that the SSA algorithm is a competitive alternative to the existing rival optimization algorithms.

The rest of this paper is organized as follows: Section 2 presents the fundamentals of SSA algorithm. Section 3 demonstrates the proposed method and its application for NN training problem. The experimental setup and results are discussed in Section 4. Finally, Section 5 concludes the findings of this paper.

2 SALP SWARM ALGORITHM

Salp Swarm Algorithm (SSA) is a swarm based metaheuristic algorithm that was recently proposed by Mirjalili et al. [21]. SSA mimics the swarming behavior of salps in oceans especially the navigation and searching for food sources. SSA is a stochastic algorithm that starts the optimization process by creating a set of initial random solutions to initiate the initial population, and then improve these solutions over the time in two phases; exploration (diversification) and exploitation (intensification). In exploration, the search space is explored to discover the promising regions, while in exploitation the neighborhood of specific solutions are to be searched hoping to find better solutions than the current ones.

Nature inspired algorithms are very popular techniques because of their ability to avoid the local optimum, their performance in finding a better solution for real world problems, and their flexibility as well as their simplicity [18].This is because they are inspired from real world behaviors, such as swarms, human, physics, etc.

SSA is the first algorithm to mimic the behavior of Salps in nature. The salps are creatures living in seas and oceans. They are similar to Jellyfishes in their tissues and movement towards the food sources [16]. Salps are usually found in groups (Swarms) called Salp chains; each salp chain contains a leader and a set of followers as shown in Fig 1.

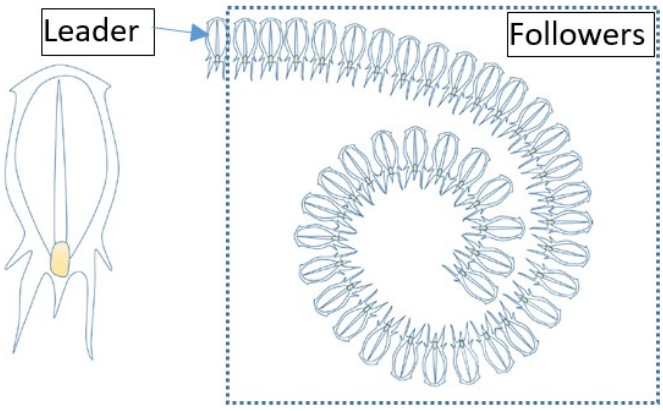

Figure 1: a Single Salp (at left) and a Salp chain (at right) [21], modified.

The mathematical model of swarming behavior proposed by [21] divides the salp chain in two groups: the leader and the followers. The first salp in the chain is denoted as the leader while the other salps are considered the followers.Accordingly, in the salp chain, the followers follow each other and the task of the leader is to lead the chain.

In SSA, the population represents the salp chain and each solution in the population represent the position of a salp in the chain. Each solution has n-dimensions, where n is the number of problems variables. Thus, two-dimensional matrix is used to store the positions of all salps. The best solution in the population is denoted as the food source or target and called F.

In the optimization process, each solution is updated depending on its role in the salp chain. The following equations are used to update the positions of slaps (leader and followers). For the position of the leader, the following equation (Eq. 1) is used:

$$x_j^1 = \begin{cases} F_j + c_1 \left(\left(ub_j - lb_j \right) c_2 + lb_j \right) & c_3 \geq 0 \\ F_j - c_1 \left(\left(ub_j - lb_j \right) c_2 + lb_j \right) & c_3 < 0 \end{cases} \quad (1)$$

where x_j^1 represents the position of the leader's j^{th} dimension. F_j represents the best solution (i.e., food source) in the is the j^{th}, ub_j and lb_j are the upper and lower bounds in the j^{th} dimension respectively. Variables c_1, c_2 and c_3 are random numbers. c_1 plays a vital role in the performance of SSA, since it is the only parameter that controls the balance between exploration and exploitation. As can be seen in Eq. 2, c_1 is a time varying parameter (depends on the iteration number) that allows high exploration rates at the early stages of the optimization process, while high exploitation rates are allowed in the last stages.

$$c_1 = 2e^{-(\frac{4l}{L})^2} \tag{2}$$

where l and L are the current iteration and the maximum number of iterations respectively. c_2 and c_3 are uniform random numbers generated in the period $[0, 1]$. These variables indicate whether the next position in the j^{th} dimension will be moved towards the $+\infty$ or $-\infty$ in addition to the step size.

Newton's law of motion (Eq.3), was used to simulate updating the positions of the followers.

$$x_j^i = \frac{1}{2}\left(x_j^i + x_j^{i-1}\right) \tag{3}$$

where $i \geq 2$ and x_j^i depicts the position of the i^{th} follower at the j^{th} dimension.

In SSA, the leader salp moves towards the food source, whereas the followers move towards the leader. The food source position can be changed during the process and then the leader will continue moving towards the new food source position. The steps of this optimization algorithm and its pseudocode are shown in Algorithm 1 below.

Algorithm 1 Pseudocode of the SSA algorithm

```
Generate an initial population of salps randomly x_i(i = 1, 2, ..., n) considering ub and lb
while Not satisfying a stopping condition do
    Evaluate each salp in the population using the fitness function
    Denote the best salp as F
    Update c1 by Eq. 2
    for (each salp (x_i) in the population) do
        if leader then
            Update the position by Eq. 1
        else
            Update the position by Eq. 3
    Amend the salps based on the upper and lower bounds of variables
    update F
Return F
```

The main structure of SSA is represented in Algorithm 1. As for any P- Metaheuristic, SSA starts the optimization process by generating a set of solutions (salps) randomly to establish the initial population. Each solution in the population is then evaluated using a fitness function and the best solution in the population is denoted as the leader (F). The value of c_1 is then updated using Eq. 2 to control the balance between exploration and exploitation. The position of each salp is then updated according to its role in the chain (i.e., the leader's position is updated using Eq. 1) and the followers' position are updated using Eq.3. This process is repeated for a predefined number of iterations.

3 THE PROPOSED METHOD

This paper proposes an approach to adjust and train the neuron weights (including biases) of NN using SSA algorithm. Experimentally, this approach is validated using health datasets as illustrated in Figure 2. The SSA algorithm is repeatedly applied until the training termination condition of the NN is met.

An artificial neural networks is composed of three layers, namely, input, hidden and output layers. Each layer contains a number of neurons, which obtain their inputs from previous neurons-layer and forward the output to their following neurons-layer as depicted in Figure 3.

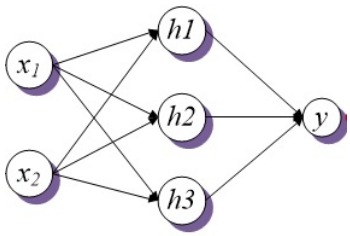

Figure 3: Feed-Forward Neural Networks.

Each NN is represented by a vector, which forms the complete set of NN structure with their corresponding weights and biases as described in Eq. 4.

$$s_i = \vec{w}_i = [W_{X_x - H_h}, B_h, W_{H_h - Y_y}, B_y] \tag{4}$$

where s_i is an individual salp in the population of salps, $W_{X_x - H_h}$ is the weight of input neuron x to hidden neuron h, B_h is the bias value of hidden neuron h, $W_{H_h - Y_y}$ represents the weight of hidden neuron h to output neuron y, and B_y is the bias value of output neuron y.

Each individual in the population (i.e., a salp being a leader or a follower) represents a neural network. The sum squared of errors (SSE) is the objective function to be evaluated. The SSE is illustrated in Eq. 5, which should be minimized as a result of the SSA optimization algorithm.

$$f(s_i) = SSE = \sum_{p=1}^{P}\sum_{y=1}^{Y}\left(D_p^y - A_p^y\right)^2 \tag{5}$$

where $f(s_i)$ is the fitness value of an individual salp s_i. P represents the number of patterns in the classification problem. Y is the number of output neurons at the output layer. D_p^y is the desired y^{th} output of p^{th} pattern and A_p^y is the actual y^{th} output of p^{th} pattern.

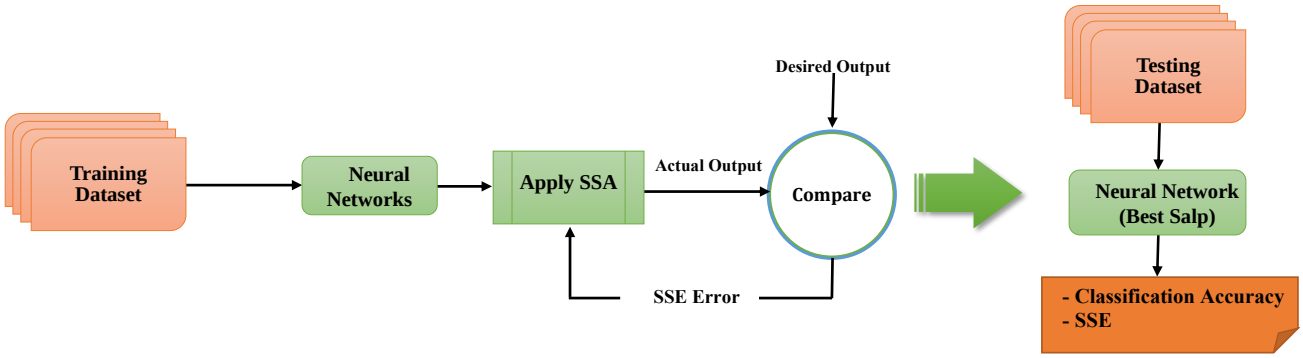

Figure 2: Basic diagram of the proposed method.

4 DATASETS, SETTINGS AND RESULTS

The merits of the SSA are validated using five benchmark classification datasets related to health data. The choice of testing this approach based on health datasets is derived from the importance of health-related data and the complexity of the nature in this domain features. The diversity, variety and the high number of features are common characteristics of health data. Moreover, using datasets with many features may provide little or redundant information, which may lead to slow down or to produce an inaccurate classifier [30].

In this research, the datasets are obtained from UCI Machine Learning Repository [8]. The selected datasets are Haberman, Wisconsin Breast Cancer, Diabetes, Thyroid and Liver. The specifications of each dataset are demonstrated in Table 1 and the description of each dataset is as follows:

- Haberman: this dataset contains cases (i.e., survived or died) from a study that was conducted during 12 years at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
- Diabetes: The Diabetes dataset was produced by National Institute of Diabetes, Digestive and Kidney Diseases (USA). The classification problem of Diabetes diagnostic is whether the females confirm positive or negative signs of diabetes (Diabetic, not Diabetic) according to World Health Organization criteria.
- Cancer: The Wisconsin Breast Cancer dataset contains medical records for cancer cases, where these patterns belong to two types (Benign, Malignant). The dataset was obtained from the University of Wisconsin Hospitals.
- Thyroid: Thyroid dataset contains biological records for different people. The dataset contains instances belong to three classes of whether the people are normal or they have thyroid problems (Normal, Sub-function, Hyper-function).
- Liver: The Liver dataset created by BUPA Medical Research Ltd., which contains a set of blood tests for people who suffers from excessive alcohol consumption.

Each dataset is partitioned into two parts 80% of patterns for training and the rest of 20% for testing. Each experimental session was conducted 20 times. Two criteria are used for evaluation; classification accuracy and the sum of squared error (SSE). The mean value and the best-out of 20 runs are reported as described in Eq. 6, Eq. 7 and Eq. 8.

$$Acc. = \frac{NCCP}{P} * 100\% \tag{6}$$

$$mean \quad of \quad Acc. \quad = \frac{1}{R}\sum_{r=1}^{R} Acc_r \tag{7}$$

$$best \quad Acc. = \max\{Acc_1, Acc_2, ..., Acc_R\} \tag{8}$$

where $Acc.$: is Classification accuracy, $NCCP$: is number of correctly classified patterns, P: is total number of patterns, and R: is number of runs.

The reported results of SSA are compared against other recent and rival P-Metahurestic algorithms; specifically, MWO, HS, FPA, and GA which taken from [26] and [4]. The SSA parameters are set as follows, maximum number of iterations is set to 1500, number of salps (population size) N=50; whereas half of the population individuals are leaders while the remaining are followers.

The classification accuracy of NN trained by different P-Metaheuristic algorithms are shown in Table 2. Results show that SSA is competitive to rival algorithms. The SSA algorithm achieves the highest best accuracy in 3 datasets out of 5. In addition, the SSA records the highest mean accuracy to classify Thyroid dataset, which it is the most difficult to classify because of its complexity and large dimensional space. Furthermore, the SSA algorithm is ranked second mean accuracy in 3 datasets.

Whilst the solution space is medium or small (in a relative to Thyroid problem), the SSA fail to record the highest mean accuracy. The results declare that the SSA seems to be more intensive to

Table 1: Datasets Specifications.

Dataset	No. of Patterns	NN Structure	Dimension
Haberman	306	3-4-2	26
Liver	345	6-5-2	48
Diabetes	768	8-7-2	79
Cancer	683	9-8-2	98
Thyroid	7,200	21-15-3	378

Table 2: Classification accuracy of NN trained by SSA algorithm and others for different datasets.

Dataset		MWO	GA	HS	FPA	SSA
Haberman	Best	79.03	77.42	75.8	77.4	79.03
	Mean	77.1	74.11	69.3	71.7	74.8
Cancer	Best	98.54	99.29	99.29	98.5	97.8
	Mean	96.72	97.46	98.4	96.9	96.7
Diabetes	Best	78.57	79.22	78.5	81.1	79.8
	Mean	75.1	73.89	75.2	75.5	75.8
Thyroid	Best	92.57	92.57	92.6	92.5	94.86
	Mean	92.55	92.57	92.5	92.5	93.06
Liver	Best	-	-	-	72.4	73.9
	Mean	-	-	-	67.9	63.84

the best solutions (the group of leaders in the population) than to be exploitative to new solutions. Nevertheless, the merits of the SSA needs more investigations and more experiments should be conducted which left as future works.

Table 3 presents the SSE values of NN trained by SSA and other algorithms. The NN trained by SSA algorithm achieves the lowest mean SSE in 2 out of 5 datasets (i.e. Haberman and Thyroid problems), which are considered the difficult to classify. In addition, the NN trained by SSA algorithm ranked the second.

Despite obtaining the lowest SSE value using SSA for classifying the Haberman, NN trained by SSA fails to gain the highest accuracy. This could be explained by the fact of over training, when the NN loss its generality to classify new patterns (testing patterns) and becomes more dedicated for the training set of patterns.

Table 3: SSE values of NN trained by SSA algorithm and others for different datasets.

Dataset		MWO	GA	HS	FPA	SSA
Haberman	Best	356.9	456	400.8	425.3	356.4
	Mean	352.4	511.2	384.1	417.2	346.0
Cancer	Best	146.5	128.0	97.5	79.7	102.5
	Mean	134.1	176.5	98.2	71.9	82.4
Diabetes	Best	806.8	992.0	2489.0	718.0	760.7
	Mean	832.0	1025.4	11308.0	724.5	758.7
Thyroid	Best	3237.1	3416.0	3225.6	3245.5	2024.8
	Mean	3257.7	3416.0	3244.6	3219.4	2741.0
Liver	Best	-	-	-	376.1	418.8
	Mean	-	-	-	389.7	430.8

5 CONCLUSIONS

In this paper, an approach to adjust the parameters of NN connections weights using SSA algorithm was proposed. The results showed that SSA algorithm has been successfully applied to train neural networks. The results showed that there is no superiority of one algorithm over another, however, results of the SSA is competitive alternative to other P-Metahuerstic algorithms. The performance of SSA is acceptable and has promising results which nominate it for other optimization applications such as timetabling.

Training the NN using the SSA for classifying many other datasets is an going research. Future work includes further analysis for the SSA algorithm and proposing an enhanced version of SSA to overcome its shortcoming, e.g. high pressure on intensification/exploitation phase rather diversification/exploration phase.

REFERENCES

[1] Ahmed A Abusnaina and Rosni Abdullah. 2013. Mussels wandering optimization algorithmn based training of artificial neural networks for pattern classification. Proceedings of the 4th International Conference on Computing and Informatics, ICOCI, 78–85.

[2] Ahmed A Abusnaina, Rosni Abdullah, and Ali Kattan. 2014. Enhanced MWO training algorithm to improve classification accuracy of artificial neural networks. In *Recent advances on soft computing and data mining*. Springer, 183–194.

[3] Ahmed A. Abusnaina, Rosni Abdullah, and Ali Kattan. 2015. The Application of Mussels Wandering Optimization Algorithm for Spiking Neural Networks Training. In *1st International Engineering Conference (IEC2014) On Developments in Civil and Computer Engineering Applications*. 197–204.

[4] Ahmed A Abusnaina, Rosni Abdullah, and Ali Kattan. 2018. Self-Adaptive Mussels Wandering Optimization Algorithm with Application for Artificial Neural Network Training. *Journal of Intelligent Systems* (2018).

[5] Ahmed A Abusnaina, Rosni Abdullah, and Ali Kattan. 2018. Supervised Training of Spiking Neural Network by Adapting the E-MWO Algorithm for Pattern Classification. *Neural Processing Letters* (2018), 1–22.

[6] Sobhi Ahmed, Majdi Mafarja, Hossam Faris, and Ibrahim Aljarah. 2018. Feature selection using salp swarm algorithm with chaos.. In *2018 The Second International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*. ACM.

[7] Zakaria Noor Aldeen Mahmood Al Nuaimi and Rosni Abdullah. 2017. NEURAL NETWORK TRAINING USING HYBRID PARTICLE-MOVE ARTIFICIAL BEE COLONY ALGORITHM FOR PATTERN CLASSIFICATION. *Journal of ICT* 16, 2 (2017), 314–334.

[8] Kevin Bache and Moshe Lichman. 2013. UCI machine learning repository. (2013).

[9] Christopher Bennett, Rodney A Stewart, and Cara D Beal. 2013. ANN-based residential water end-use demand forecasting model. *Expert systems with applications* 40, 4 (2013), 1014–1023.

[10] Christian Blum and Krzysztof Socha. 2005. Training feed-forward neural networks with ant colony optimization: An application to pattern classification. In *Hybrid Intelligent Systems, 2005. HIS'05. Fifth International Conference on*. IEEE, 6–pp.

[11] Robert E Dorsey, John D Johnson, and Walter J Mayer. 1994. A genetic algorithm for the training of feedforward neural networks. *Advances in artificial intelligence in economics, finance and management* 1 (1994), 93–111.

[12] Hadeel Ibrahim, Mazher Wamidh, Osman Ucan, and Bayat Oguz. 2017. Feature Selection using Salp Swarm Algorithm for Real Biomedical Datasets. *IJCSNS International Journal of Computer Science and Network Security* (2017).

[13] Dervis Karaboga, Bahriye Akay, and Celal Ozturk. 2007. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In *International conference on modeling decisions for artificial intelligence*. Springer, 318–329.

[14] Ali Kattan and Rosni Abdullah. 2011. Training of feed-forward neural networks for pattern-classification applications using music inspired algorithm. *International Journal of Computer Science and Information Security* 9, 11 (2011), 44.

[15] Ali Kattan and Rosni Abdullah. 2013. Training feed-forward artificial neural networks for pattern-classification using the harmony search algorithm. In *The Second International Conference on Digital Enterprise and Information Systems (DEIS2013)*. The Society of Digital Information and Wireless Communication, 84–97.

[16] Larry Madin. 1990. Aspects of jet propulsion in salps. *Canadian Journal of Zoology* (1990).

[17] Majdi Mafarja, Ibrahim Aljarah, Ali Asghar Heidari, Abdelaziz I. Hammouri, Hossam Faris, AlaâĂŹM. Al-Zoubi, and Seyedali Mirjalili. 2018. Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems. *Knowledge-Based Systems* 145 (2018), 25 – 45. https://doi.org/10.1016/j.knosys.2017.12.037

[18] Majdi Mafarja, Derar Eleyan, Salwani Abdullah, and Seyedali Mirjalili. 2017. S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem. In *Proceedings of the International Conference on Future Networks and Distributed Systems*. ACM, 14.

[19] Majdi Mafarja and Seyedali Mirjalili. 2018. Whale optimization approaches for wrapper feature selection. *Applied Soft Computing* 62 (2018), 441–453.

[20] Majdi M Mafarja and Seyedali Mirjalili. 2017. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* 260 (2017), 302–312.

[21] Seyedali Mirjalili, Amir Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. 2017. Salp swarm algorithm: a

bio-inspired optimizer for engineering design problems. *Advances in Engineering Software* (2017).

[22] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. 2014. Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences* 269 (2014), 188–209.

[23] Nazri Mohd Nawi, Abdullah Khan, M. Z. Rehman, Tutut Herawan, and Mustafa Mat Deris. 2014. Comparing Performances of Cuckoo Search Based Neural Networks. In *Recent Advances on Soft Computing and Data Mining*, Tutut Herawan, Rozaida Ghazali, and Mustafa Mat Deris (Eds.). Springer International Publishing, Malaysia, 163–172.

[24] Wolfram Schiffmann, Merten Joost, and Randolf Werner. 1993. Application of genetic algorithms to the construction of topologies for multilayer perceptrons. In *Artificial Neural Nets and Genetic Algorithms*. Springer, 675–682.

[25] Moh'd Khaled Shambour, Ahamad Tajudin Khader, Ahmed A. Abusnaina, and Qusai Shambour. 2014. Modified Tournament Harmony Search for Unconstrained Optimisation Problems. In *Recent Advances on Soft Computing and Data Mining*, Tutut Herawan, Rozaida Ghazali, and Mustafa Mat Deris (Eds.). Springer International Publishing, Malaysia, 283–292.

[26] Moh'd Khaled Yousef Shambour, Ahmed A. Abusnaina, and Ahmed I. Alsalibi. 2018. Modified Global Flower Pollination Algorithm and its Application for Optimization Problems. *Interdisciplinary Sciences: Computational Life Sciences* (2018).

[27] Danielle NG Silva, Luciano DS Pacifico, and Teresa Bernarda Ludermir. 2011. An evolutionary extreme learning machine based on group search optimization. In *Evolutionary computation (CEC), 2011 IEEE congress on*. IEEE, 574–580.

[28] Tejen Su, Jyunwei Jhang, and Chengchih Hou. 2008. A hybrid artificial neural networks and particle swarm optimization for function approximation. *International Journal of Innovative Computing, Information and Control* 4, 9 (2008), 2363–2374.

[29] El-Ghazali Talbi. 2009. *Metaheuristics: from design to implementation*. Vol. 74. John Wiley & Sons.

[30] K. Wan and V. Alagar. 2016. Characteristics and classification of big data in health care sector. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 1439–1446. https://doi.org/10.1109/FSKD.2016.7603389

[31] Willfried Wienholt. 1993. Minimizing the system error in feedforward neural networks with evolution strategy. In *ICANNâĂŹ93*. Springer, 490–493.