ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20211227132418
Original scientific paper

Website Phishing Technique Classification Detection with HSSJAYA Based MLP Training

Erkan ERDEMIR*, Adem Alpaslan ALTUN

Abstract: Website phishing technique is the process of stealing personal information (ID number, social media account information, credit card information etc.) of target users through fake websites that are similar to reality by users who do not have good intentions. There are multiple methods in detecting website phishing technique and one of them is multilayer perceptron (MLP), a type of artificial neural networks. The MLP occurs with at least three layers, the input, at least one hidden layer and the output. Data on the network must be trained by passing over neurons. There are multiple techniques in training the network, one of which is training with metaheuristic algorithms. Metaheuristic algorithms that aim to develop more effective hybrid algorithms by combining the good and successful aspects of more than one algorithm are algorithms inspired by nature. In this study, MLP was trained with Hybrid Salp Swarm Jaya (HSSJAYA) and used to determine whether websites are suspicious, phishing or legal. In order to compare the success of MLP trained with hybrid algorithm, Salp Swarm Algorithm (SSA) and Jaya (JAYA) were compared with MLPs trained with Cuckoo Algorithm (CS), Genetic Algorithm (GA) and Firefly Algorithm (FFA). As a result of the experimental and statistical analysis, it was determined that the MLP trained with HSSJAYA was successful in detecting the website phishing technique according to the results of other algorithms.

Keywords: hybrid salp swarm jaya algorithm; metaheuristic; MLP; website phishing

1 INTRODUCTION

The Internet is a system that allows large-scale networks such as private, public and academic to connect and transfer data between each other [1, 2]. The most common usage areas of the Internet are search, news, social networking, gaming, e-commerce, education, file and data transfer, communication, online services and remote work [3]. Internet use has many advantages as well as risks. Some of these include cyberbullying, physical and mental health problems, inappropriate content, wasted time and internet fraud [4-6].

Internet fraud, which is growing day by day and causing economic damage, is a cybercrime that is carried out by using tools such as the internet, e-mail, sms and malware etc. to seize and defraud people's information [7-9]. Trojans, identity theft, data breach, ransomware, denial of service, scareware, malware and phishing are some techniques used in internet fraud [7, 10, 11].

Phishing is a fraud method, which is a kind of social engineering attack that allows malicious users (cyber criminals) to deceive the private information of target users (such as credit card, debit card, social media accounts, etc.) [12-14]. Imitation of an e-mail sent by a corporate firm, fake e-mails that seem to have been sent from social media, banking sites prepared like the real one are just a few of the phishing techniques used to deceive people [15].

In this research, a study on website phishing detection was conducted. In the website phishing technique, the victim is directed to the fake of the corporate site in various ways in order to obtain their information by malicious users [15]. The victim, who does not know about website phishing, very quickly believes the fake website and can transfer their information to cybercriminals. This is because, thanks to today's web programming language technologies, it has become very difficult to distinguish the fake website from the real website [16].

One method used to detect website phishing techniques is the blacklist method [17, 18]. The blacklist includes information about websites containing phishing threats [19]. Cybercriminals also liken URLs to legitimate-looking sites so that victims believe in legitimate-looking sites. The fake web page, which is a copy of the real thing visually, is likened to a legal site as a URL, making it easier for the victims to believe [19, 20]. Another method used in detecting phishing is artificial neural network (ANN), which is a type of machine learning used to make sense or detect trends from data that is too complex or imprecise to be noticed by humans or other computer techniques [21-23]. There are multiple models of ANN, one of which is a single-direction feed forward multilayer perceptron (MLP). If the MLP is well trained, the weight set that minimizes the estimation and classification errors will be obtained in the most efficient way [24-29].

Different algorithms may be required as a learning technique in the training of the network. One of them is deterministic algorithms. Deterministic algorithms are a mathematical optimization method used to optimize ANN's performance, in which no hesitant results are obtained [30-32]. It can be expressed as a disadvantage of deterministic approaches that adding extra hidden layers in the training of the network and being stuck in the local optimum depending on the first solution slows down the training of the network [33-35].Another technique used in training the network is stochastic algorithms [36]. Stochastic algorithms use randomness, which reduces the probability of getting stuck in the local minimum and makes it less dependent on the initial solution [37]. There are multiple types of stochastic algorithms, one of which is metaheuristic algorithms inspired by nature [38]. Some of the advantages of metaheuristic algorithms are that they can be designed easily, can solve real problems and functions efficiently, can be hybridized with more than one algorithm, and are not stuck on local optimal values [38-40].

Salp swarm algorithm (SSA) [41], jaya algorithm (JAYA) [42, 43], cuckoo search algorithm (CS) [44], genetic algorithm (GA) [45] and firefly algorithm (FFA) [46] are some of the popular metaheuristic algorithms.

There are different methods used to develop more powerful and efficient metaheuristic methods or to minimize the weaknesses of metaheuristic algorithms; one of these methods is to develop hybrid metaheuristic algorithms by combining the strengths of more than one algorithm [47-49]. In the literature, there are studies used in detecting phishing techniques by using metaheuristic algorithms. Some of these are listed below:

Ali & Malebary [50] proposed a particle swarm optimization (PSO) based method to better detect the website phishing technique. According to this method, PSO is used to better weight different website features in order to achieve higher accuracy values when finding phishing websites. This intelligent method, which was developed according to the results obtained, achieved better results in detecting phishing compared to other machine learning techniques.

Suleman & Awan [51] stated that they used machine learning techniques for phishing detection. Researchers using URL-based phishing detection in their studies have tried to detect fake or legal websites with machine learning techniques. Researchers using GA for feature selection in machine learning found that the accuracy increased compared to the results without GA.

Krithiga & Ilavarasan [52], proposed a hybrid algorithm consisting of Whale Optimization Algorithm (WOA) and SSA to detect spam profiles, which are a threat to social networks; created a data set with the data they obtained from Twitter by them. The hybrid algorithm was compared with other methods using this data set and its performance was measured. Researchers have determined that the method they developed gives effective results compared to other methods.

Khurma et al. [53] proposed a new phishing detection system based on SSA. The researchers aimed to maximize the classification performance of their proposed phishing system and to minimize the number of features. They obtained more than one binary SSA using different transfer functions. The researchers found that the binary SSA with X-TF obtained better results among the algorithms analyzed according to a certain evaluation criteria.

Anupam & Kar [54] used a machine learning method to detect whether sites are legitimate or fake using different properties of a website's URL. With the Support Vector Machine binary classifier, an optimum hyperplane was determined with the help of metaheuristic algorithms to predict whether a website is legitimate or fake. Among these metaheuristic algorithms, they found that the Grey Wolf algorithm is better. They also compared the metaheuristic algorithms with the grid-search optimized Random Forest technique. As a result, it was determined that all metaheuristic algorithms used in the research were successful according to the Random Forest technique.

In this study, a new MLP training technique has been proposed for the detection of website phishing technique. MLP training was carried out using the Hybrid Salp Swarm Jaya algorithm (HSSJAYA) developed by Erdemir & Altun [55]. The reason for choosing this algorithm in MLP training is that it is a new hybrid algorithm, and its success in solving benchmark problems. The developed method has been analyzed by comparing it with many leading metaheuristic algorithms, including SSA and JAYA, which create the hybrid algorithm.

2 OVERVIEW

2.1 Salp Swarm Algorithm (SSA)

Developed by Mirjalili et al. [41], SSA was designed by imitating the eco system of the salp creature, which lives in the depths of the oceans and seas as a swarm. The leader leads the salp swarm. The leader updates its position according to the food it wants to reach. Follower salps update their position by following the leader salp, because the best way to reach the food is to follow the leader. In this algorithm, random solutions are initially generated. In the exploration phase, the places that are thought to be food are searched in the search space, while the existing solutions are compared with their neighbors in order to find the best solution in the exploitation phase. The updated Eq. (1) [56] is used to update the leader position. Eq. (2) is used to update the position of the followers.

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

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

According to the equations above, while x_j^i is $i == 1$, it shows the leader salp's position and in other cases the position of follower salps in the j-th dimension. The food source's position in the j-th dimension shows F_j. The terms ub_j and lb_j represent the lower and upper bounds of the j-th dimension, respectively. The terms c_1, c_2 and c_3 are auxiliary coefficients. c_1 is important for food source and leader salp position update. The equation for c_1 is shown in Eq. (3).

$$c_1 = 2e^{-\left(\frac{4it}{Max\_it}\right)^2} \quad (3)$$

The term e in Eq. (3) denotes the euler number, the term *it* denotes the current iteration number, and the term *Max_it* denotes the maximum iteration number. The terms c_2 and c_3 are random values between 0 and 1.

In different studies on SSA, it has been stated that the multiple leader salp method increases the randomness of the algorithm positively compared to the single leader, but it drags the algorithm to instability, and to overcome this problem, the leader and follower salps should be balanced by half (N/2) [57-60]. Fig. 1 contains the flowchart prepared according to the situation specified in this paragraph. In this study, analyses were made for the SSA and HSSJAYA algorithms, taking into account the flowchart in Fig. 1.

2.2 Jaya Algorithm (JAYA)

JAYA, developed by Rao [42], is an algorithm that performs the steps according to the best and worst solution with its simple, easy and powerful aspects. In this algorithm, the aim is to try to be closer to the best solution and further to the worst solution. There are no special parameters for the algorithm. According to Rao [42], the updates of the solution are in Eq. (4).

$$\begin{aligned} s'_{j,k,i} &= s_{j,k,i} + r_{1,j,i}\left(s_{j,best,i} - \left|s_{j,k,i}\right|\right) \\ &- r_{2,j,i}\left(s_{j,worst,i} - \left|s_{j,k,i}\right|\right) \end{aligned} \quad (4)$$

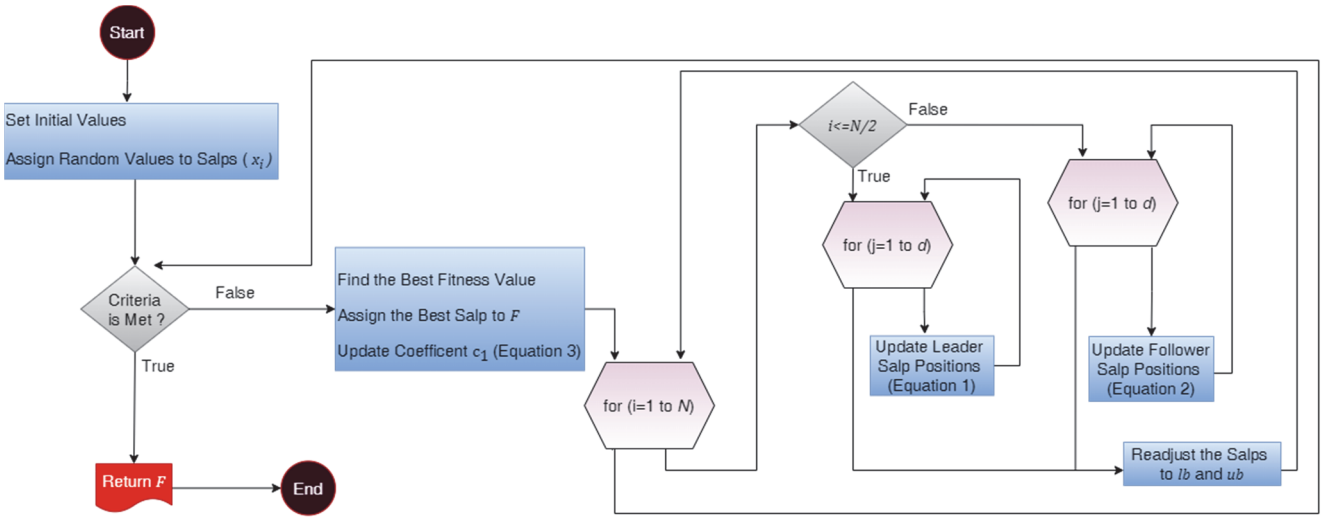

Figure 1 Flowchart of SSA

According to this equation, $s'_{j,k,i}$ is the new (updated) solution of the j-th variable for the k-th candidate solution during the i-th iteration, $s_{j,k,i}$ is solution of the j-th variable for the k-th candidate solution during the i-th iteration, $r_{1,j,i}$ and $r_{2,j,i}$ are random values 0 to 1, $s_{j,best,i}$ is expressed as the best candidate solution of the j-th variable, $s_{j,worst,i}$ is expressed as the worst candidate solution of the j-th variable. In addition, the $r_{1,j,i}\left(s_{j,best,i}-\left|s_{j,k,i}\right|\right)$ part of the equation is the case of the solution approaching the best solution. On the other hand, $-r_{2,j,i}\left(s_{j,worst,i}-\left|s_{j,k,i}\right|\right)$ explains the situation where the solution moves away from the worst solution. Flowchart of JAYA is shown in Fig. 2.

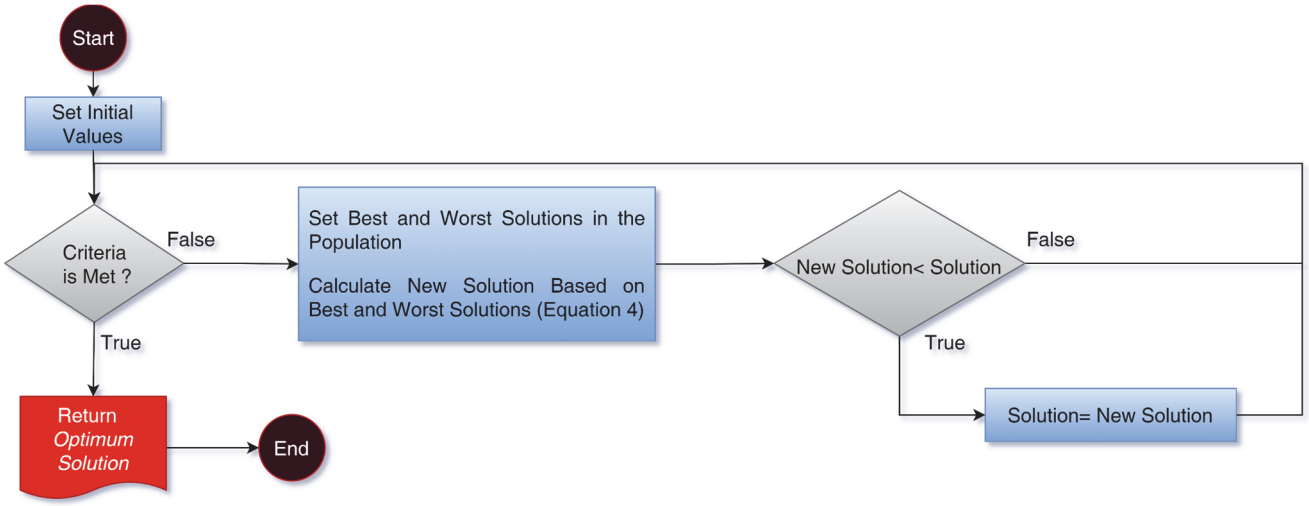

Figure 2 Flowchart of JAYA

2.3 Hybrid Salp Swarm Jaya Algorithm (HSSJAYA)

Developed by Erdemir & Altun [55], HSSJAYA covers the leader and follower position update sections of SSA, it was created by taking into account the best and worst approach of JAYA. In this algorithm, the targeted position for the salps to get closer to the food is expressed as the best food source. The position of the salps away from the food is expressed as the worst food source position. In HSSJAYA, the approach to the best food position that salp chain should reach and the worst food position that it is not wanted to reach is given in Fig. 3.

In order for the leader and follower salps to reach the best food source, the position status consisting of the worst food source is included in the position updates, so that the salps can reach the best food source more easily.

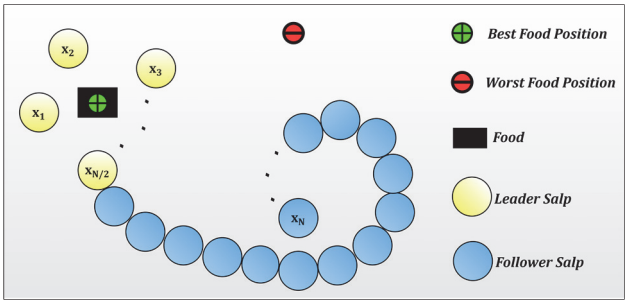

Figure 3 According to half multi leader salp in HSSJAYA, best food/worst food position sampling of salp chain

Initial values must be set at the beginning of the algorithm. The salp population is randomly determined by the determined amount of search agent (N) and size (d). Salps whose fitness values are calculated are sorted. d -

dimensional best food position, worst food position and new candidate solution are created. Sorted salps are compared among themselves with their fitness values and assigned as the best food position, best food position fitness value, worst food position and worst food position fitness value. This part of the flowchart to be included in Fig. 4 will be indicated as "Set initial values". The equation used to find a new candidate solution in JAYA is used as in Eq. (5) in HSSJAYA;

$$s'_j = x_j^i + bwfr\left(r_{1,i,j}\left(bfp_j - \left|x_j^i\right|\right) - r_{2,i,j}\left(wfp_j - \left|x_j^i\right|\right)\right) \quad (5)$$

According to Eq. (5); s'_j, new solution in j-th dimension, x_j^i, i-th follower salp position in the j-th dimension, $r_{1,i,j}$ and $r_{2,i,j}$ are random values between 0 and 1, bfp_j is the best food position in the j-th dimension, wfp_j is the worst food position in the j-th dimension, $bwfr$ is the best worst fit ratio. $bwfr$ is calculated according to Eq. (6).

$$bwfr = bfpfv \,/\left(bfpfv + wfpfv\right) \quad (6)$$

According to Eq. (6), the terms $bfpfv$ and w$fpfv$ represent the best food positon fitness value and the worst food positon fitness value respectively. Unlike SSA, in HSSJAYA an update is made in the position of the salps. This update is included in Eq. (7).

$$x_j^i = x_j^i - s'_j \quad (7)$$

In the HSSJAYA algorithm, the leader salps update is as in Eq. (8), provided that it is $i \le N/2$:

$$x_j^i = \begin{cases} bfp_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 \ge 0.5 \\ bfp_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right) & c_3 < 0.5 \end{cases} \quad (8)$$

The terms x_j^i and bfp_j in Eq. (8) have been explained before and x_j^i represents the leader salp (multiple). The terms ub_j and lb_j represent the lower and upper bounds of the j-th dimension respectively. The auxiliary coefficient c_1 is calculated according to Eq. (9).

$$c_1 = iv2e^{-\left(\frac{4it}{Max\_it}\right)^2} \quad (9)$$

Unlike c_1 in SSA, iv parameter has been added to c_1. The iv value is a non-random value between 0 and 1, it is the value of improving the c_1 coefficient. The researchers who developed this algorithm stated that it would be more useful to use the parameter as a non-random parameter so that it can be changed according to the problem. The terms c_2 and c_3 are random values between 0 and 1. Provided that $i > N/2$ updating the follower salper in HSSJAYA is the same as in SSA. This update is shown in Eq. (2) and also the flowchart of HSSJAYA is in Fig. 4.

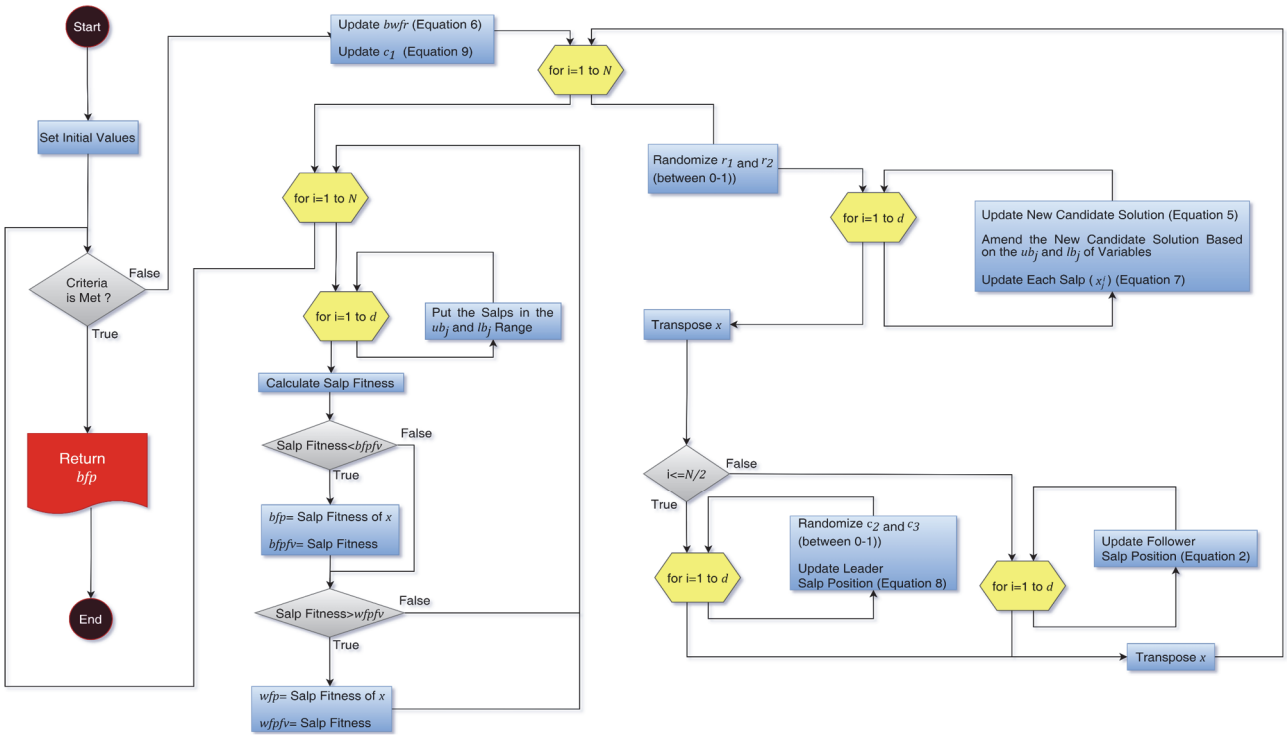

Figure 4 Flowchart of HSSJAYA

3 HSSJAYA BASED MLP TRAINING

MLP is a feedforward neural network in which neurons are mostly directly connected to each other, and data moves from the input layer to the output layer by passing one or more hidden layers. The input data of the problem is given to the input layer. In the output layer, the results of the network are obtained against the expected results of the problem [61].

The structure of the MLP created in this study consists of three layers: input, one hidden layer and output.

The network needs to be trained by transmitting data from neurons in one layer to neurons in other layers. At this point, the summation and activation functions are of great importance. Summation functions are used to calculate the net input in a cell. The summation function used in this study is summed by adding the bias (b) to the multiplication of the input (x) data by the weights (w). The summation function used in this study is as in Eq. (10). The activation function produces output values for the perceptron against the net input. In this study, the tangent-sigmoid function in Eq. (11) was used as the activation function [62, 63].

$$net = \sum_{i=1}^{n}(x_i w_i) + b \tag{10}$$

$$net \Rightarrow a \qquad f(a) = \text{tansig}(a) = \frac{2}{1+e^{-2a}} - 1 \tag{11}$$

In the training of the network, mean squared error (*MSE*) was chosen as the function for evaluating the measurements between algorithms and calculating the fitness value. *MSE* function is shown in Eq. (12) [21].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(o_i - o_i')^2 \tag{12}$$

In the equation, n represents the number of samples, o the actual output value, and o' the expected output value. In this study, the confusion matrix was used as another performance measurement criterion. It classifies the amount of true and false according to the estimated and expected (existing) values. There are four types of them: true positive (TP), false positive (FP), true negative (TN) and false negative (FN) [64]. In the literature, evaluation measures have been developed according to the classification results obtained from the confusion matrix, the percentile performance of the algorithms has been compared by using a few of these measures in the network training and testing processes. These assessment measures are accuracy, precision, recall, f1_score and error [65].

The diagram of the training of the MLP network using HSSJAYA is given in Fig. 5. According to this diagram, first of all, initial values (assigning random values to salps, best/worst food position, fitness values) are set. Salps are assigned as weights and biases for training the network. The network is trained using the specified number of layers and the number of neurons in the training dataset. The best food position fitness value and the worst food position fitness value are calculated with the function used to calculate the *MSE* according to the stages in HSSJAYA. The best food position fitness value is assigned to the *MSE*. If the maximum iteration is reached, the confusion matrix results for the training and testing phases are calculated. Finally, the results of the confusion matrix and the convergence curve obtained from the *MSE* are output.

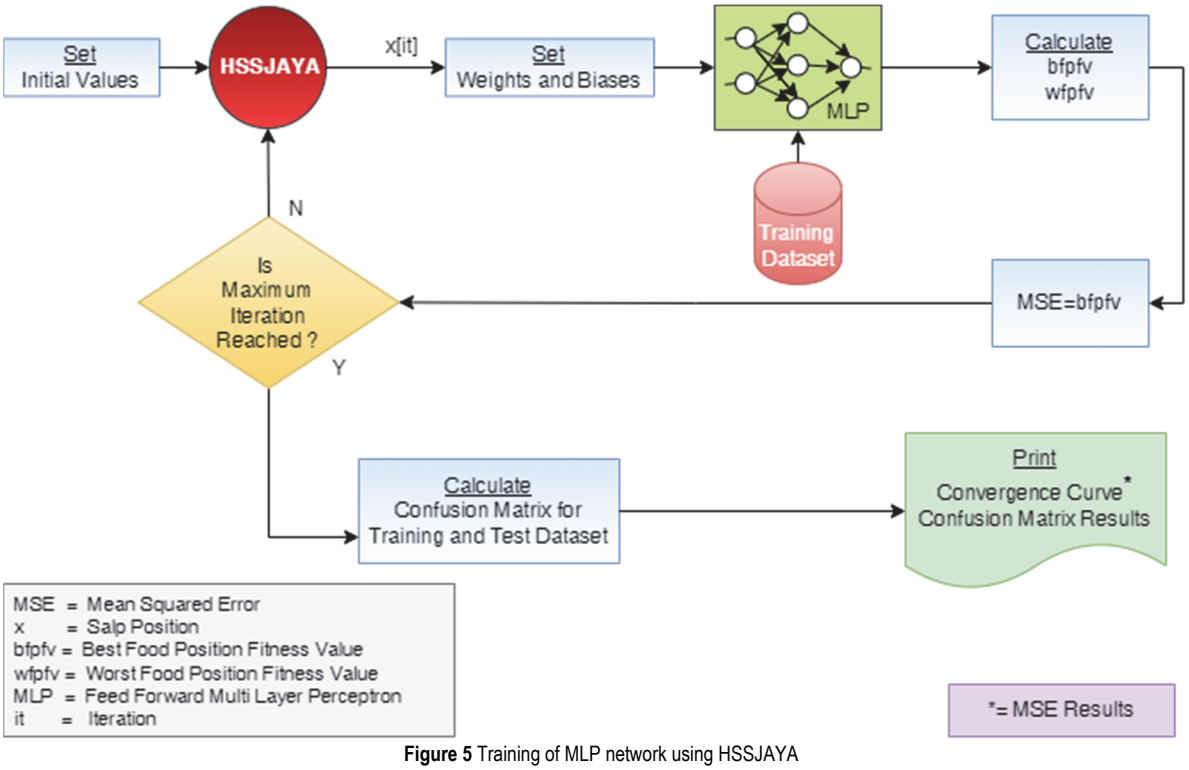

Figure 5 Training of MLP network using HSSJAYA

4 EXPERIEMENTAL RESULTS

In this study, the MLP network trained with HSSJAYA was used to detect the website phishing technique. The results were compared with MLP trained with CS, FFA, and GA, including SSA and JAYA, which make up the hybrid algorithm. In network training, search agents in all algorithms were formed among [−1, 1], these values were used to create weights and biases. The data set used in the classification detection of website phishing technique is the data set named "Website Phishing" developed by Abdelhamid et al. [17] and it was obtained from the UCI machine learning repository [66]. The values in the data set were normalized between 0-1 [67]. The

number of neurons in the hidden layer was calculated with the $2I + 1$ equation. I denotes the input [68]. Tab. 1 contains information about the data set.

Table 1 Website phishing dataset information

Dataset	Attributes	Number of			
		Hidden Layers	Samples		Output
			Train	Test	
Website Phishing	10 (9 Inputs) (1 Output)	19	902	451	3

In this study, Evolopy-NN framework was used to analyze the experimental results. Evolopy frameworks are an easy-to-use toolbox for MLP training, feature selection, and solving global optimization problems [26-29, 57, 69, 70-72] Algorithms can have their own fixed or variable parameters. In this study, the *iv* parameter of HSSJAYA was set as 0.1. Parameter values of other algorithms are the same as in the framework called Evolopy used in this research. [57, 69].

In the training of the network, the algorithms were tried independently for 30 times with different search agents and iteration numbers. Those with the same number of search agents and iteration values were grouped together and evaluated.

Studies with similar criteria (such as the same population, the same iteration, the same number of trials, the same network structure) in the literature compare with each other. Since we could not find any studies with similar criteria for the algorithms we used for comparison in our research, we carried out our analyses with the criteria we determined.

Analyses made according to different search agents and iterations are indicated by group names in figures and tables. The number of search agents/iterations used in the study are 30/100 (Group 1), 30/200 (Group 2), 50/100 (Group 3), and 50/200 (Group 4), respectively.

Results are the average results obtained as a result of the number of independent tries indicated. Average *MSE* and standard deviation results of MLP trained with algorithms that have worked 30 times with different search agents and iteration numbers are given in Tab. 2.

Table 2 Average (*MSE*) and standard deviation results*

Algorithms		Group 1	Group 2	Group 3	Group 4
HSSJAYA	Ave.	**0.0832**	**0.0781**	**0.0814**	**0.0760**
	St. Dv.	**0.0033**	**0.0026**	**0.0041**	**0.0022**
SSA	Ave.	0.0904	0.0826	0.0866	0.0814
	St. Dv.	0.0070	0.0043	0.0061	0.0044
JAYA	Ave.	0.1404	0.1250	0.1375	0.1254
	St. Dv.	0.0191	0.0184	0.0198	0.0192
CS	Ave.	0.1164	0.0951	0.1113	0.0950
	St. Dv.	0.0070	0.0032	0.0074	0.0029
FFA	Ave.	0.0901	0.0836	0.0880	0.0839
	St. Dv.	0.0060	0.0063	0.0067	0.0063
GA	Ave.	0.1204	0.1094	0.1116	0.1023
	St. Dv.	0.0126	0.0116	0.0120	0.0098
*Results of 30 independent trials					

When Tab. 2 is examined, low error rate and standard deviation rate in all groups were performed by MLP trained with HSSJAYA. According to this result, it is seen that the MLP network trained with HSSJAYA detects the website phishing technique better than the MLP networks trained with other algorithms.

Fig. 6 shows the convergence curves of MLPs trained with algorithms.When the convergence curves are examined, it is seen that the MLP network trained with HSSJAYA has a faster convergence speed than other algorithms in detecting website phishing.

As in the "HSSJAYA Based MLP Training" title, another measurement criterion in this study is the measurement results obtained from the confusion matrix. Confusion matrix results of MLP networks trained with algorithms are given in Tabs. 3 to 6. Results are shown as percentages.

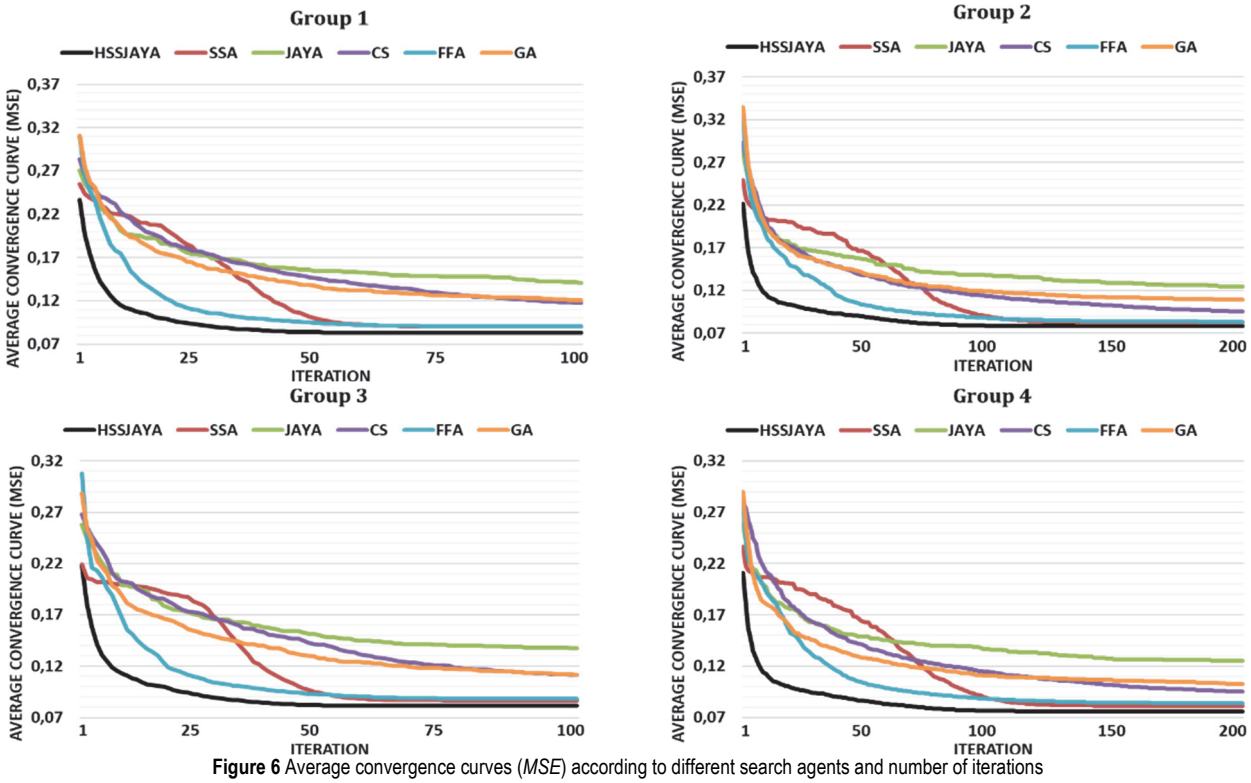

Figure 6 Average convergence curves (*MSE*) according to different search agents and number of iterations

Table 3 Results of confusion matrix for group 1

Algorithms	Train					Test				
	Accuracy	Precision	Recall	F1 Score	Error	Accuracy	Precision	Recall	F1 Score	Error
HSSJAYA	**85.8081**	92.0577	**83.1881**	**87.3887**	**14.1919**	**85.0145**	92.2560	**82.0217**	**86.8248**	**14.9855**
SSA	84.7854	91.7940	81.5846	86.3658	15.2146	84.1159	92.0741	80.6137	85.9326	15.8841
JAYA	78.3987	89.2132	72.5568	79.6344	21.6013	77.7246	89.4527	71.7690	79.2209	22.2754
CS	81.9074	90.6395	77.5063	83.4738	18.0926	81.0870	90.4454	76.8351	82.9708	18.9130
FFA	84.9309	**92.2041**	81.4331	86.4628	15.0691	84.2971	**92.6561**	80.3730	86.0237	15.7029
GA	80.9742	91.3392	75.0947	82.2426	19.0258	79.8188	90.9797	73.9832	81.3905	20.1812

Table 4 Results of confusion matrix for group 2

Algorithms	Train					Test				
	Accuracy	Precision	Recall	F1 Score	Error	Accuracy	Precision	Recall	F1 Score	Error
HSSJAYA	**86.4278**	**92.3622**	**84.0025**	**87.9747**	**13.5722**	**85.9855**	**92.7617**	**83.2371**	**87.7318**	**14.0145**
SSA	85.9239	92.2821	83.1629	87.4759	14.0761	85.0435	92.5355	81.7810	86.8084	14.9565
JAYA	80.2912	90.6585	74.7096	81.6116	19.7088	79.7246	90.5938	74.3803	81.3533	20.2754
CS	84.3934	91.7719	80.9154	85.9673	15.6066	83.5652	91.5311	80.2286	85.4566	16.4348
FFA	85.8380	92.0995	83.2008	87.4086	14.1620	84.8478	91.8846	82.1059	86.7012	15.7029
GA	82.2546	91.4481	77.3043	83.6961	17.7454	81.3478	91.4750	76.2575	83.0644	20.1812

Table 5 Results of confusion matrix for group 3

Algorithms	Train					Test				
	Accuracy	Precision	Recall	F1 Score	Error	Accuracy	Precision	Recall	F1 Score	Error
HSSJAYA	**85.7969**	**92.2836**	**82.9230**	**87.3441**	**14.2031**	**85.2174**	92.4682	**82.1661**	**86.9987**	**14.7826**
SSA	85.4013	91.7718	82.7399	87.0145	14.5987	84.7319	92.1331	81.6366	86.5521	15.2681
JAYA	79.5819	91.1540	72.6641	80.5096	20.4181	78.6667	90.1798	72.5271	80.0541	21.3333
CS	82.2508	90.2267	78.6427	83.9440	17.7492	81.6159	90.1206	78.2310	83.6372	18.3841
FFA	85.2594	92.2804	81.9381	86.7881	14.7406	84.6087	92.2717	81.2635	86.4067	15.3913
GA	82.1053	92.2826	76.2121	83.3969	17.8947	81.8478	**92.5691**	76.0529	83.4149	18.1522

Table 6 Results of confusion matrix for group 4

Algorithms	Train					Test				
	Accuracy	Precision	Recall	F1 Score	Error	Accuracy	Precision	Recall	F1 Score	Error
HSSJAYA	**86.5995**	**92.4334**	**84.2361**	**88.1389**	**13.4005**	**86.0072**	**92.4639**	**83.5860**	**87.7946**	**13.9928**
SSA	85.8903	92.2231	83.1566	87.4456	14.1097	85.0797	91.9470	82.4549	86.9330	14.9203
JAYA	80.2053	90.6924	74.5202	81.4482	19.7947	79.3406	90.5082	73.7425	80.8828	20.6594
CS	84.2516	91.2260	81.2563	85.9046	15.7484	83.7029	91.3670	80.6258	85.6109	16.2971
FFA	86.0284	92.2057	83.4407	87.5883	13.9716	85.4058	92.2972	82.7196	87.2161	14.5942
GA	83.3109	91.5859	79.0909	84.8286	16.6891	82.7391	91.4422	78.7605	84.5747	17.2609

When Tabs. 3 to 6 are examined it is seen that the MLP trained with HSSJAYA achieved successful results in all of the accuracy, recall, precision, f1 score and error values according to the training and test error matrix results in Group 2 and Group 4. In addition, HSSJAYA showed success in all other measurements except training and test sensitivity results in Group 1 and test sensitivity results in Group 3. The hybrid algorithm was found to be successful in all measurements compared to the confusion matrix results of MLPs trained with the algorithms (SSA and JAYA) from which it was derived.

Table 7 Wilcoxon rank sum test *p*-value results (N/A = not applicable)

Algorithms	Groups			
	Group 1	Group 2	Group 3	Group 4
HSSJAYA	N/A	N/A	N/A	N/A
SSA	$2.204e^{-05}$	$1.208e^{-05}$	$1.286e^{-04}$	$6.802e^{-08}$
JAYA	$1.970e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$
CS	$2.872e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$
FFA	$6.975e^{-06}$	$3.479e^{-05}$	$2.063e^{-05}$	$2.290e^{-08}$
GA	$2.872e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$	$2.872e^{-11}$

Although the MLP trained with HSSJAYA is detected as successful in detecting website phishing according to the experimental results, it should be determined whether the algorithm also creates a statistically significant difference. As a statistical test, wilcoxon rank sum test [73] was applied as one of the data analysis tests and the results are listed in Tab. 7. In order for the results to be statistically significant, $p < 0.05$ ($5e^{-02}$) was accepted. The most successful algorithm in each group shown in Tab. 2 was compared with the other algorithm [41, 74].

When the results in Tab. 7 are examined, MLP trained with HSSJAYA shows a statistically significant difference compared to other algorithms.

5 CONCLUSIONS AND FUTURE STUDIES

MLP trained with HSSJAYA and other algorithms were analyzed for website phishing classification detection. Accordingly, the HSSJAYA algorithm was successful in optimizing benchmark functions compared to the algorithms it was compared to [55] and was also successful in training MLP compared to the algorithms it was compared to. In other words, the MLP trained with HSSJAYA was determined according to the experimental results, which it detected better than other algorithms compared to determine whether websites are phishing, suspicious or legal, according to the attributes in the website phishing dataset used in the study. This shows that the philosophy of reaching the best food of the HSSJAYA algorithm works. The reason why HSSJAYA is successful is that it has the strengths of SSA and JAYA in its structure.

JAYA is a successful and powerful algorithm. However, JAYA success appears to be low in these analyses. In the literature, it is stated that according to the NFL theorem, any metaheuristic may not always produce good results as a result of analyses such as all optimization

problems or training the network [75]. Therefore, we cannot say that JAYA is a failure for all studies.

We predict that the researchers using MLP trained with the hybrid algorithm in this study will successfully conclude classification estimation processes by using datasets belonging to different problems. In future studies, it is thought that HSSJAYA will be more successful if it is re-hybridized with one or more algorithms that are successful in MLP training.

6 REFERENCES

[1] Dennis, M. A. & Kahn, R. (2021, June 8). *Internet*. Encyclopedia Britannica. Retrieved November 22, 2021, from https://www.britannica.com/technology/Internet.

[2] Technopedia. (2020). *What is the internet? - definition from Techopedia*. Retrieved November 22, 2021, from https://www.techopedia.com/definition/2419/internet.

[3] Angelov, M. (2020). *Technology: Top 15 uses of the internet (in our daily life).* The Digital Chain. Retrieved November 22, 2021, from https://thedigitalchain.com/uses-of-the-internet/.

[4] Machimbarrena, J. M., Calvete, E., Fernández-González, L., Álvarez-Bardón, A., Álvarez-Fernández, L., & González-Cabrera, J. (2018). Internet risks: an overview of victimization in cyberbullying, cyber dating abuse, sexting, online grooming and problematic internet use. *International Journal Of Environmental Research And Public Health*, *15*(11), 2471, 15. https://doi.org/10.3390/ijerph15112471

[5] Nkotagu, G. H. (2011). Internet fraud: Information for teachers and students. *Journal of International Students*, *1*(2), 72-75. https://doi.org/10.32674/jis.v1i2.557

[6] Singh, R. (2018, September 20). *10 lines on disadvantages of internet*. Teachingbanyan.com. Retrieved November 23, 2021, from https://www.teachingbanyan.com/10-lines/10-lines-on-disadvantages-of-internet/ .

[7] Fortinet. (n.d.). *Internet fraud: What is internet fraud?* Retrieved November 23, 2021, from https://www.fortinet.com/resources/cyberglossary/internet-fraud.

[8] Mishra, S. & Soni, D. (2019). SMS phishing and mitigation approaches. *2019 Twelfth International Conference on Contemporary Computing*, 1-5. https://doi.org/10.1109/IC3.2019.8844920

[9] Shulzhenko, N. & Romashkin, S. (2020). Internet fraud andtransnational organized crime. *Juridical Tribune (Tribuna Juridica)*, *10*(1), 162-172.

[10] Puram, P. K., Kaparthi, M., & Rayaprolu, A. K. H. (2011). Online scams: Taking the fun out of the internet. *Indian Journal of Computer Science And Engineering*, *2*(4), 559-565.

[11] Sönmez, Ü. (2017). Bilişim sistemleri aracılığıyla dolandırıcılık suçu. *Dicle Üniversitesi Adalet Meslek Yüksekokulu Dicle Adalet Dergisi*, *1*(2), 47-59.

[12] Elbahadır, H. (2017). *Hacking interface.* İstanbul: Kodlab.

[13] Georgescu, E. (2021, January 21). *Did you know that there are various types of online financial frauds lurking in the cyberspace?*. Heimdal Security. Retrieved November 23, 2021, from https://heimdalsecurity.com/blog/types-of-online-financial-frauds/.

[14] Tahir, A. (2021). *Sanal gerçeklik*. İstanbul: Payidar Yayınları.

[15] Hekim, H. (2015). Oltalama (phishing) saldırıları. *Siber suçlar, tehditler, farkındalık ve mücadele içinde*, 57-83.

[16] Berghel, H., Carpinter, J., & Jo, J.-Y. (2007). Phish phactors: offensive and defensive strategies. *Advances in Computers*, *70*, 223-268. https://doi.org/10.1016/S0065-2458(06)70005-5

[17] Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection based associative classification data mining. *Expert Systems with Applications*, *41*(13), 5948-5959. https://doi.org/10.1016/j.eswa.2014.03.019

[18] Manoj, P., Bhuvan Kumar, Y., Rakshitha, D., & Megha, G. (2021). Detection and classification of phishing websites. *Trends in Computer Science and Information Technology*, *6*(2), 053-059. https://doi.org/10.17352/tcsit.000040

[19] Rao, R. S. & Pais, A. R. (2017). An enhanced blacklist method to detect phishing websites. *Information systems security*, *10717*, 323-333. https://doi.org/10.1007/978-3-319-72598-7_20

[20] Büber, E. (2018, February 8). Phishing url detection with ml. *Towards Data Science*. Retrieved November 23, 2021, from https://towardsdatascience.com/phishing-domain-detection-with-ml-5be9c99293e5.

[21] Ferreira, R. P., Martiniano, A., Napolitano, D., Romero, M., de Oliveira Gatto, D. D., Farias, E. B. P., & Sassi, R. J. (2018). Artificial neural network for websites classification with phishing characteristics. *Social Networking*, *7*(2), 97-109. https://doi.org/10.4236/sn.2018.72008

[22] Maind, S. B. & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, *2*(1), 96-100.

[23] Eluyode, O. S. & Akomolafe, D. T. (2013). Comparative study of biological and artificial neural networks. *European Journal of Applied Engineering and Scientific Research*, *2*(1), 36-46.

[24] Ataseven, B. (2013). Yapay sinir ağları ile öngörü modellemesi. *Öneri Dergisi*, *10*(39), 101-115.

[25] Rao, M. B. (2000). Feedforward neural network methodology. *Technometrics*, *42*(4), 432-433. https://doi.org/10.1080/00401706.2000.10485725

[26] Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, *22*, 1-15. https://doi.org/10.1007/s00500-016-2442-1

[27] Aljarah, I., Faris, H., Mirjalili, S., & Al-Madi, N. (2018). Training radial basis function networks using biogeography-based optimizer. *Neural Computing and Applications*, *29*, 529-553. https://doi.org/10.1007/s00521-016-2559-2

[28] 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, 1-32. https://doi.org/10.1142/S0218213016500330

[29] 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. https://doi.org/10.1007/s10489-016-0767-1

[30] Keskintürk, T. (2006). Diferansiyel gelişim algoritması. *Istanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, *5*(9), 85-99.

[31] Sel, Ç. (2013). *Genel atama problemlerinin çözümünde deterministik, olasılık temelli ve sezgisel yöntemlerin uygulanması*. Master's thesis, University of Ankara.

[32] Yamany, W., Fawzy, M., Tharwat, A., & Hassanien, A. E. (2015). Moth-flame optimization for training multi-layer perceptrons. *11th International Computer Engineering Conference(ICENCO)*, 267-272. https://doi.org/10.1109/ICENCO.2015.7416360

[33] Agrawal, U., Arora, J., Singh, R., Gupta, D., Khanna, A., & Khamparia, A. (2020). Hybrid wolf-bat algorithm for optimization of connection weights in multi-layer perceptron. *ACM Transactions on Multimedia Computing, Communications, and Applications*, *16*(1s), 1-20. https://doi.org/10.1145/3350532

[34] Bi, W., Wang, X., Tang, Z., & Tamura, H. (2005). Avoiding the local minima problem in backpropagation algorithm with modified error function. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, *E88-A*(12), 3645-3653. https://doi.org/10.1093/ietfec/e88-a.12.3645

[35] Yaghini, M., Khoshraftar, M. M., & Fallahi, M. (2013). A hybrid algorithm for artificial neural network training. *Engineering Applications of Artificial Intelligence*, *26*(1), 293-301. https://doi.org/10.1016/j.engappai.2012.01.023

[36] Tang, Z. & Koehler, G. J. (1994). Deterministic global optimal FNN training algorithms. *Neural Networks*, *7*(2), 301-311. https://doi.org/10.1016/0893-6080(94)90024-8

[37] Bairathi D. & Gopalani D. (2019). Salp swarm algorithm (SSA) for training feed-forward neural networks. *Advances in intelligent systems and computing*, *816*, 521-534. https://doi.org/10.1007/978-981-13-1592-3_41

[38] Gupta, D. & Gupta, V. (2017). Test suite prioritization using nature inspired meta-heuristic algorithms. *Advances in Intelligent Systems and Computing*, *557*, 216-226. https://doi.org/10.1007/978-3-319-53480-0_22

[39] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[40] Wang, G. & Guo, L. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics*, *2013*, 1-21. https://doi.org/10.1155/2013/696491

[41] 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. https://doi.org/10.1016/j.advengsoft.2017.07.002

[42] Rao, R. V. (2016). Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, *7*(1), 19-34. http://doi.org/10.5267/j.ijiec.2015.8.004

[43] Dede, T., Grzywiński, M., & Rao, R. V. (2020). Jaya: a new meta-heuristic algorithm for the optimization of braced dome structures. *Advances in intelligent systems and computing*, *949*, 13-20. https://doi.org/10.1007/978-981-13-8196-6_2

[44] Yang, X. S. & Deb, S. (2009). Cuckoo search via Lévy flights. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210-214. https://doi.org/10.1109/NABIC.2009.5393690

[45] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, *267*(1), 66-72. https://doi.org/10.1038/scientificamerican0792-66

[46] Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2), 78-84. https://doi.org/10.1504/IJBIC.2010.032124

[47] Şenel, F. A., Gökçe, F., Yüksel, A. S., & Yiğit, T. (2019). A novel hybrid PSO-GWO algorithm for optimization problems. *Engineering with Computers*, *35*, 1359-1373. https://doi.org/10.1007/s00366-018-0668-5

[48] Thangaraj, R., Pant, M., Abraham, A., & Bouvry, P. (2011). Particle swarm optimization: Hybridization perspectives and experimental illustrations. *Applied Mathematics and Computation*, *217*(12), 5208-5226. https://doi.org/10.1016/j.amc.2010.12.053

[49] Ting, T. O., Yang, X. S., Cheng, S., & Huang, K. (2015). Hybrid metaheuristic algorithms: Past, present, and future. *Studies in computational intelligence*, *585*, 71-83. https://doi.org/10.1007/978-3-319-13826-8_4

[50] Ali, W. & Malebary, S. (2020). Particle swarm optimization-based feature weighting for improving intelligent phishing website detection. *IEEE Access*, *8*, 116766-116780. https://doi.org/10.1109/ACCESS.2020.3003569

[51] Suleman, M. T. & Awan, S. M. (2019). Optimization of url-based phishing websites detection through genetic algorithms. *Automatic Control and Computer Sciences*, *53*, 333-341. https://doi.org/10.3103/S0146411619040102

[52] Krithiga, R. & Ilavarasan, E. (2020). A novel hybrid algorithm to classify spam profiles in twitter. *Webology*, *17*(1), 260-279. http://doi.org/10.14704/WEB/V17I1/WEB17003

[53] Khurma, R. A., Sabri, K. E., Castillo, P. A., & Aljarah, I. (2021). Salp swarm optimization search based feature selection for enhanced phishing websites detection. *Lecture notes in computer science, 12694*, 146-161. https://doi.org/10.1007/978-3-030-72699-7_10

[54] Anupam, S. & Kar, A.K. (2021). Phishing website detection using support vector machines and nature-inspired optimization algorithms. *Telecommunication Systems*, *76*, 17-32. https://doi.org/10.1007/s11235-020-00739-w

[55] Erdemir, E. & Altun, A. A. (2022). A new metaheuristic approach to solving benchmark problems: Hybrid salp swarm jaya algorithm. *CMC-Computers, Materials & Continua*, *71*(2), 2923-2941. https://doi.org/10.32604/cmc.2022.022797

[56] Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., & Heidari, A. A. (2020). Salp swarm algorithm: theory, literature review, and application in extreme learning machines. *Studies in Computational Intelligence*, *811*, 185-199. https://doi.org/10.1007/978-3-030-12127-3_11

[57] Faris, H., Qaddoura, R., Aljarah, I., Bae, J. W., Fouad, M. M., Floden, L., Wolz, D., Bawagan, P., & Merelo-Guervós, J. J. (2016). *EvoloPy.* Github. Retrieved November 25, 2021, from https://github.com/7ossam81/EvoloPy/blob/master/optimizers/.

[58] Mirjalili, S. (2018). *SSA: Salp Swarm Algorithm.* Mathworks. Retrieved November 25, 2021, from https://www.mathworks.com/matlabcentral/fileexchange/63745-ssa-salp-swarm-algorithm.

[59] Wang, D., Zhou, Y., Jiang, S., & Liu, X. (2018). A simplex method-based salp swarm algorithm for numerical and engineering optimization. *IFIP Advances in Information and Communication Technology*, *538*, 150-159. https://doi.org/10.1007/978-3-030-00828-4_16

[60] Zhang, Q., Chen, H., Heidari, A. A., Zhao, X., Xu, Y., Wang, P., Li, Y., & Li, C. (2019). Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers. *IEEE Access*, *7*, 31243-31261. https://doi.org/10.1109/ACCESS.2019.2902306

[61] Turkoglu, B. & Kaya, E. (2020). Training multi-layer perceptron with artificial algae algorithm. *Engineering Science and Technology, an International Journal*, *23*(6), 1342-1350. https://doi.org/10.1016/j.jestch.2020.07.001

[62] Priddy, K. L. & Keller, P. E. (2005). *Artificial neural networks: an introduction*. Bellingham: SPIE Press. https://doi.org/10.1117/3.633187

[63] Yonaba, H., Anctil, F., & Fortin, V. (2010). Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. *Journal of Hydrologic Engineering*, *15*(4), 275-283. https://doi.org/10.1061/(ASCE)HE.1943-5584.0000188

[64] Le, T.-T.-H., Kim, J., & Kim, H. (2017). An effective intrusion detection classifier using long short-term memory with gradient descent optimization. *2017 International Conference on Platform Technology and Service (PlatCon)*, 1-6. https://doi.org/10.1109/PlatCon.2017.7883684

[65] Saito, T. & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *Plos One*, *10*(3), 1-21. https://doi.org/10.1371/journal.pone.0118432

[66] Dua, D. & Graff, C. (2019). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science.

[67] Patel, V. R. & Mehta, R. G. (2011). Impact of outlier removal and normalization approach in modified k-means clustering algorithm. *International Journal of Computer Science Issues*, *8*(5-2), 331-336.

[68] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences*, *269*, 188-209. https://doi.org/10.1016/j.ins.2014.01.038

[69] Aljarah, I., Faris, H., & Fouad, M. M. (2016). *Evolopy-NN,* Github. Retrieved October 15, 2021, from https://github.com/7ossam81/EvoloPy-NN.

[70] Faris, H., Aljarah, I., Mirjalili, S., Castillo, P., & Merelo-Guervós, J. J. (2016). Evolopy: An open-source nature-inspired optimization framework in python. *Proceedings of the 8th International Joint Conference on Computational Intelligence - ECTA, (IJCCI 2016)*, *3*, 171-177.

https://doi.org/10.5220/0006048201710177
[71] Qaddoura, R., Faris, H., Aljarah, I., & Castillo, P. A. (2020). EvoCluster: An open-source nature-inspired optimization clustering framework in python. *Lecture Notes in Computer Science*, *12104*, 20-36. https://doi.org/10.1007/978-3-030-43722-0_2
[72] Khurma, R. A., Aljarah, I., Sharieh, A., & Mirjalili, S. (2020). Evolopy-fs: An open-source nature-inspired optimization framework in python for feature selection. *Algorithms for intelligent systems: Evolutionary machine learning techniques*, 131-173. https://doi.org/10.1007/978-981-32-9990-0_8
[73] Pratt, W. E. (2010). *Wilcoxon Rank Sum Test*. *Encyclopedia of research design*. Thousand Oaks, CA, Sage.
[74] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3-18. https://doi.org/10.1016/j.swevo.2011.02.002
[75] Wolpert, D. H. & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67-82. https://doi.org/10.1109/4235.585893

Contact information:

Erkan ERDEMIR, PhD, Information Technologies Teacher
(Corresponding author)
Department of Information Technologies,
Tokat Vocational and Technical Anatolian High School
(Tokat Mesleki ve Teknik Anadolu Lisesi),
Gazi Osman Pasa Bulvari No: 48,
60030 Merkez/Tokat/Turkey
E-mail: erdemirerkan@gmail.com

Adem Alpaslan ALTUN, PhD, Professor
Department of Computer Engineering,
Faculty of Technology,
Konya Selcuk University,
42130 Selcuklu/Konya/Turkey
E-mail: altun@selcuk.edu.tr