BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 9, No. 4, October 2021

A Novel Approach Based to Neural Network and Flower Pollination Algorithm to Predict Number of COVID-19 Cases

Ceren Baştemur Kaya and Ebubekir Kaya*

Abstract— **Flower Pollination Algorithm (FPA) is one of the popular heuristic algorithms that model pollination in the natural environment. Since 2012, it has been used in the solution of many difficult real world problems and successful results have been achieved. In this study, FPA is utilized for the training of neural network to predict number of COVID-19 cases. Namely, a model based on FPA and neural network (FPA_NN) is proposed. Within the scope of application, the data belonging to Turkey are estimated using the proposed model. A data set is created with the data between 1 April 2020 and 15 September 2020. A time series is created with these data and the nonlinear dynamic systems are obtained to model the problem. In order to determine the performance of the proposed model, RMSE (root mean square error) are used. The output graphs of the results are also examined in detail. The results are compared with neural network approaches based on PSO and HS. The Wilcoxon signed rank test is utilized to determine the significance of the results. The results show that FPA is generally more effective than PSO and HS to predict number of COVID-19 cases based on neural network.**

Index Terms— **COVID-19, flower pollination algorithm, neural network, swarm intelligence.**

I. Introduction

THE BIGGEST epidemic of 2020 is undoubtedly COVID-19. Millions of people have been exposed to this epidemic so far and hundreds of thousands of people have died. Scientists conduct studies to recognize, analyze, model and predict this epidemic. Younes and Hasan [1] proposed a model based on extended Kalman Filter (EKF) algorithm and a stochastic Lotka–Volterra model to evaluate the spread of COVID-19. Duran-Lopez et al. [2] worked on COVID-19 diagnosis from chest X-ray images and proposed the novel deep learning-based system called COVID-XNet. They used a convolutional neural network (CNN) to extract features and classify on normal and COVID-19 cases. Pal et al. [3] proposed an approach based on long short-term memory (LSTM) and neural network to estimate country-based risk of COVID-19. They compared the performance of their proposed method with methods such as linear regression, Lasso linear regression, Ridge regression, Elastic Net, LSTM-FCNS, Recidual RNN, GRU and GRU + Baysian. They reported that the proposed method is effective. Ezzat et al. [4] proposed a new approach based on DenseNet121, gravitational search optimization and CNN for the diagnosis of COVID-19 disease. They compared the performance of the proposed method with different approaches such as MobileNet, DarkCovidNet, CNN-SA, CoroNet and Deep Bayes-SqueezeNet. They reported an accuracy rate of 98.38% for the proposed method. Ismael and Şengür [5] presented novel study using deep learning approaches and local texture descriptors for COVID-19 detection with X-ray chest image. Zhan et al. [6] proposed a pseudocoevolutionary simulated annealing (SA) algorithm for identifying epidemic spreading dynamics of COVID-19. Al-Qaness et al. [7] used a method based on ANFIS, the modified FPA algorithm and the salp swarm algorithm for the prediction of COVID-19 cases. They introduced the FPASSA hybrid method by adapting SSA to the local search mechanism of FPA. This hybrid method was used to determine the parameters of ANFIS. The performance of FPASSA-ANFIS was compared with classical ANFIS, GA-ANFIS, PSO-ANFIS, ABC-ANFIS and FPA-ANFIS. They used RMSE, MAE, MAPE, RMSRE, and R2 error types for comparisons and reported that the performance of their proposed method was effective. Melin et al. [8] presented a new approach with multiple ensemble neural network models and fuzzy response aggregation for predicting COVID-19 data in Mexico. Al-Qaness et al. [9] proposed a method based on ANFIS and the marine predators algorithm (MPA) to estimate the number of people affected in Italy, Iran, Korea, and the USA. In this method, ANFIS's parameters were determined using MPA. The performance of MPA was compared with different ANFIS-based approaches. Saba and Elsheikh [10] proposed a method based on artificial intelligence techniques

CEREN BAŞTEMUR KAYA, is with Department of Computer Technologies, Nevsehir Haci Bektas Veli University, Nevsehir, Turkey, (e-mail: cerenbastemurkaya@gmail.com).

https://orcid.org/0000-0002-0091-3606

EBUBEKİR KAYA, is with Department of Computer Engineering, Nevsehir Haci Bektas Veli University, Nevsehir, Turkey, (e-mail: ebubekirkaya@yandex.com, ebubekir@nevsehir.edu.tr).

https://orcid.org/0000-0001-8576-7750

Manuscript received May 4, 2021; accepted August 9, 2021.
DOI: 10.17694/bajece.932391

to predict the spread of the epidemic in Egypt and autoregressive integrated moving average (ARIMA) and nonlinear autoregressive artificial neural networks (NARANN). Here, only a very small part is given to shed light on the studies on COVID-19. When the literature is examined, it is seen that there are many studies on COVID-19 [11-15].

Most of the studies on COVID-19 are based on artificial intelligence techniques [11-15]. One of the important artificial intelligence techniques is artificial neural networks (ANNs). ANN is used effectively in many real world problems [16-20]. A good training process is required to achieve effective results with ANN. A good training algorithm is required for a good training process. When the literature is examined, it is seen that many heuristic algorithms have been proposed [21-23]. Due to the advantages that heuristic algorithms have, it has been used extensively in neural network training lately. Genetic algorithm (GA) [24-25], artificial bee colony (ABC) algorithm [26-27], particle swarm optimization (PSO) [28-29], harmony search (HS) [30-31], differential evolution (DE) [32-33], firefly algorithm (FA) [34] and cuckoo search (CS) [35] are some of the algorithms used extensively in ANNs training. One of the other algorithms used in ANN training is FPA. Liang et al. [36] used an improved FPA for optimizing backward propagation network. They analyzed the results of BP algorithm, FPA-BP algorithm and IFPA-BP algorithm. Kowalski and Wadas [37] introduced a probabilistic neural network with FPA. Dutta and Kumar [38] used an ANN based FPA for modeling and optimization of a liquid flow process. Apart from these studies, there are different studies based on FPA and ANN [39-43].

FPA has been used in solving many real world problems other than ANN network training [44-45]. This shows that FPA is a powerful and effective algorithm. Computer science, bioinformatics, operation research, imaging science, food industry, meteorology, medicine, education and engineering are some of the areas where FPA is used [44]. As part of this study, ANN is trained with FPA for the prediction of COVID-19 cases belongs to Turkey. There are three reasons for choosing FPA as an ANN training algorithm. First, FPA has been used to solve many real-world problems in different fields and has achieved successful results. This shows that FPA is an effective algorithm. Second, as seen in the literature, FPA has been used in ANN training and the successful results have been obtained. This means that FPA is effective in ANN training. Third, FPA-based ANN model has been used for the first time to predict COVID-19 data. This shows that the study is innovative.

The sections continuing in this study are organized as: In Section 2, flower pollination algorithm, artificial neural network and ANN training with FPA are explained. In Section 3, results and experiments are presented. Conclusions are given in the last section.

II. METHODOLOGY

A. *Flower Pollination Algorithm*

FPA imitates the pollination process of flowering plants and was developed by Yang in 2012 [46]. It carries out pollination in two ways as biotic and abiotic in flowering plants. The determinant of this is the pollinators. Biotic pollination mostly occurs by insects such as honey bees and butterflies. Abiotic is mostly caused by wind and diffusion. Pollination in flowering plants is divided into two according to the type of source: Self-pollination and cross-pollination. Self-pollination is between flowers on the same plant. Cross-pollination is between the flowers of different plants. Biotic, abiotic, self-pollination and cross-pollination are visualized as in Fig. 1 by Abdel-Basset and Shawky [44]. As seen in Fig. 2, the basis of the FPA is based on local and global pollination. While global pollination occurs in larger areas with biotic effect, abiotic factors allow local pollination to occur in more limited area.

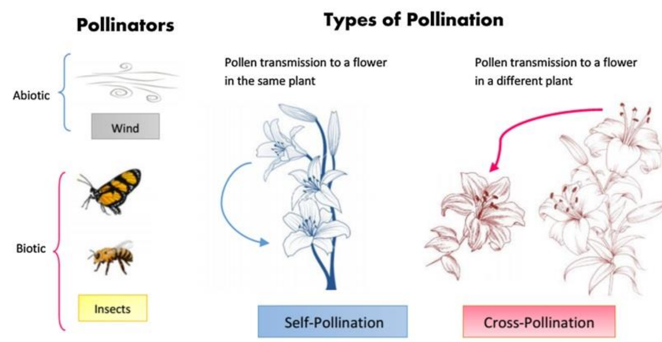

Fig.1. The pollinators and pollination types [44]

Flower Pollination Algorithm (or simply Flower Algorithm)

Objective min *or* max $f(\mathbf{x})$, $\mathbf{x} = (x_1, x_2, ..., x_d)$
Initialize a population of n flowers/pollen gametes with random solutions
Find the best solution $\mathbf{g}_*$ *in the initial population*
Define a switch probability $p \in [0, 1]$
while $(t < MaxGeneration)$
  for $i = 1 : n$ *(all n flowers in the population)*
    if $rand < p$,
      Draw a (d-dimensional) step vector L which obeys a Lévy distribution
      Global pollination via $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + L(\mathbf{g}_* - \mathbf{x}_i^t)$
    else
      Draw ϵ *from a uniform distribution in [0,1]*
      Randomly choose j and k among all the solutions
      Do local pollination via $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \epsilon(\mathbf{x}_j^t - \mathbf{x}_k^t)$
    end if
    Evaluate new solutions
    If new solutions are better, update them in the population
  end for
    Find the current best solution $\mathbf{g}_*$
end while

Fig.2. Pseudo code of Flower Pollination Algorithm (FPA) [46]

In the FPA algorithm, the pollination process takes place with some assumptions. These are:

• Biotic and cross pollination are processes of global pollination. In this process, pollinators can move to large areas with Levy flight.

• Abiotic and self pollination are processes of local pollination.

• Flower constancy is a reproduction probability associated with the similarity of the two flowers and is effectively used by some pollinators.

• The transition between local and global pollination process is controlled by switch probability (p).

Global pollination process is carried out as biotic with the help of pollinators such as insects and birds. In the global pollination process, the pollinators act according to the Levy distribution. This mode of action is one of the most important differences between global and local pollination. Global pollination and flower constancy are formulated as given (1).

$$x_i^{t+1} = x_i^t + \gamma L(g_* - x_i^t) \tag{1}$$

Here, x_i^t is the pollen i. Namely, it is solution vector at tth iteration. g_* is current best solution found among all solutions at the currentiteration. γ represents a scaling factor to control the step size. L is a Lévy-flights-based step size.

Local pollination process occurs in the form of abiotic and self pollination. Global pollination and flower constancy are formulated as given (2). Here, x_j^t and x_k^t are pollen from different flowers of the same plant species. $\in$ is a local random walk in the range of [0, 1].

$$x_i^{t+1} = x_i^t + \in \left(x_j^t - x_k^t\right) \tag{2}$$

In FPA, the local and global optimization process is controlled by switch probability (p). For global pollination to be more effective, the *p* value should be greater than 0.5. As this value approaches 1, the effect of biotic pollinators increases. Otherwise, local pollination is more effective in the process. The range in which the *p* value is effective can vary depending on the problem type. For this, analyzing the *p* control parameter according to different problems can provide better quality solutions.

B. Neural Network and Training Process

One of the types of ANNs is feed forward artificial neural networks (FFNNs). In a FFNN, there is a one-way movement from inputs to outputs. It may or may not have the hidden layer. If it has a hidden layer, it consists of three layers as seen in Fig. 3. An example of a FFNN is presented in Fig. 3. It consists of 3 inputs and 5 outputs and there are 4 artificial neurons in the hidden layer. Neurons in different layers had a connection between each other. Each connection has a weight. There is no connection between neurons in the same layer [47-48].

In order to create a suitable model with ANN the following steps should be taken into consideration [47]:

- The inputs and outputs of the problem to be used in modeling should be determined.
- Input and output values should be normalized between *a* and *b* such as [0, 1] or [-1, 1].
- The number of neurons in the hidden layer must be determined. The activation functions to be used must be selected.
- Weights and bias values must be produced.
- The parameters should be updated with a training algorithm.
- The training process should continue until the stop criterion.
- The trained neural network should be tested.

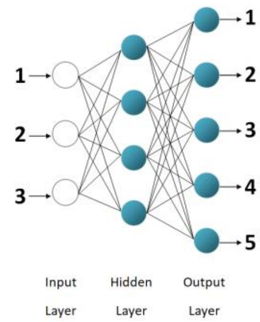

Fig.3. An example for FFNN

Within the scope of this study, the training of the FFNN is carried out with the FPA algorithm. The studies have been carried out on neural networks consisting of 2 inputs and 3 inputs. The model has only 1 output, corresponding to the number of COVID-19 patients/cases. It has been applied on different network structures containing 3, 5, 10 and 15 neurons in the hidden layer. In this context, 2-3-1, 2-5-1, 2-10-1 and 2-15-1 network structures are created for 2 inputs. For 3 inputs, 3-3-1, 3-5-1, 3-10-1 and 3-15-1 network structures are used. Sigmoid is preferred as the activation function. Within the scope of neural network training, weights and bias values are optimized within limits by FPA. The number of parameters to be determined in ANN indicates the dimension for FPA. These parameters correspond to solution vector. The stopping criterion in FPA is the maximum number of iterations. The training process continues until it reaches this value.

III. Experiments and Results

A. Data Preparation

For the estimation of COVID-19 data belongs to Turkey, the number of daily cases between 1 April 2020 and 15 September 2020 is investigated. The numbers of cases are taken from the website of the WHO. A time series is created with the 168 days of data. Namely, the time series can be thought as a one-dimensional array with 168 elements. The relevant time series has large scale values. However, FFNN structures are used in modeling this time series. In order to create the model properly, the time series data is scaled in [0, 1] interval by using (3). Here, x is the data set. x_{min} refers to the minimum value of x. x_{max} corresponds to the maximum value of x. x_n is the scaled state of x in the interval [0, 1].

$$x_n = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{3}$$

By using time series, systems consisting of 2, 3 and 4 inputs are created. The input and output of these systems are shown in Table I. In S_1, y(t-1) and y(t-2) are the inputs of the system. y(t-1), y(t-2), y(t-3) are the inputs of S_2. In S_3, y(t-1), y(t-2), y(t-3) and y(t-4) are the inputs of the system. All systems contain one output as y(t). As can be seen from these systems, three different nonlinear dynamic systems are created. For the modeling of the systems, 135 data are used as train data. In addition, 33 data are randomly selected as test data.

The input and output data used in the systems are exemplified for better understanding. Let's assume y is the normalized case data for the last four days. y is given in (4). According to S_1, if y(t)=0.8, the input values y(t-1) and y(t-2) are 0.6 and 0.5, respectively.

$$y = \{0.1, 0.5, 0.6, 0.8\} \tag{4}$$

TABLE I
THE SYSTEMS USED IN APPLICATIONS

System	Inputs	Output	Number of Train/Test Data
S_1	y(t-1), y(t-2)	y(t)	135/33
S_2	y(t-1), y(t-2), y(t-3)	y(t)	135/33
S_3	y(t-1), y(t-2), y(t-3), y(t-4)	y(t)	135/33

TABLE II
THE CONTROL PARAMETERS VALUES USED

Algorithms	Control Parameters	Values
PSO	Population Size	10
	Intertia Weights	[0.9,0.6]
	Maximum Number of Iterations	10000
HS	Memory Size	10
	Consideration Rate	0.95
	PAR	0.3
	Maximum Number of Iterations	10000
FPA	Population Size	10
	Switch Probability	0.6
	Number of Cycles	10000

B. *Model Evaluation Parameters*

FPA is used in ANN training. The control parameters of FPA directly affect its performance. Therefore, a detailed study has been carried out on control parameters. The results are obtained for different population size values (n=10, n=20 and n=50). At the same time, the analysis are performed within different switch probability values (p=0.5, p=0.6, p=0.7, p=0.8 and p=0.9). The different ANN structures have been used to achieve effective results in S_1, S_2 and S_3 systems. Each application starts with randomly selected starting population and is run 30 times. Mean error value is obtained by taking the average of the results obtained. RMSE is used as the error type and is calculated using (5). Here, n refers to the number of samples. y_i is real output and $\overline{y_i}$ is predicted output. In the following section, the performance of PSO and HS in related problems is also analyzed. The control parameters used in these analyzes are also presented in Table II.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_i - \overline{y_i})^2} \tag{5}$$

C. *Results and Discussion*

Three nonlinear dynamic systems (S_1, S_2 and S_3) have been created for predicting COVID cases. These systems are modeled with different ANN structures and the results obtained are presented in Table III. According to different population size and different network structure, training error value, test error value, training standard deviation and test standard deviation are reached in this table. In S_1 model, 2-3-1, 2-5-1, 2-10-1, 2-15-1 network structure is used. In 2-3-1 network structure, the best train error value is found for n = 10. The best test error value is found at n=20. As the population size increased, the train error value also increases. This is not valid in test error value. During the training process, effective standard deviation values are achieved for all population values. In the test process, the increase in the population size increases the standard deviation. In 2-5-1, the best training error value is found when n=10, while the best test error value is obtained with n=20. The best test error value is 0.0401. The train standard deviation for n=20 and n=50 is 0.0012. In 2-10-1, the train error values obtained for n=10, n=20 and n=50 are 0.0490, 0.0519 and 0.0534 respectively. Close test error values are found at n=10 and n=20. In 2-15-1, an effective training error value is reached by using n = 10. In this network structure, more unsuccessful test error values are obtained compared to other network structures. While the standard deviation of the train is successful, it is observed that the standard deviation of the test is high.

For S_2, 3-3-1, 3-5-1, 3-10-1 and 3-15-1 network structures are utilized. In 3-3-1, the best train error value is found as 0.0476 by using n=10. In this network structure, the best test error value is 0.0568. As the population size increases, the test error value also increases. Train standard deviation values are successful according to test standard deviation. In 3-5-1, the effective train error value is found with n=10 as in 3-3-1. The best test error value is obtained as 0.0524 with n=20. The effective train standard deviation values are observed for all population sizes. The same success is not achieved in test standard deviation values. In 3-10-1, the best train error value is 0.0486. The best test error value is also 0.05647. The most effective train standard deviation values are reached with n= 20 and n=50. At the same time, it is observed that the train standard deviation values are better than the test standard deviation values. In 3-15-1, the effective train error value is found as 0.0467. Test error value is higher than train error value. A similar approach is also found in standard deviations.

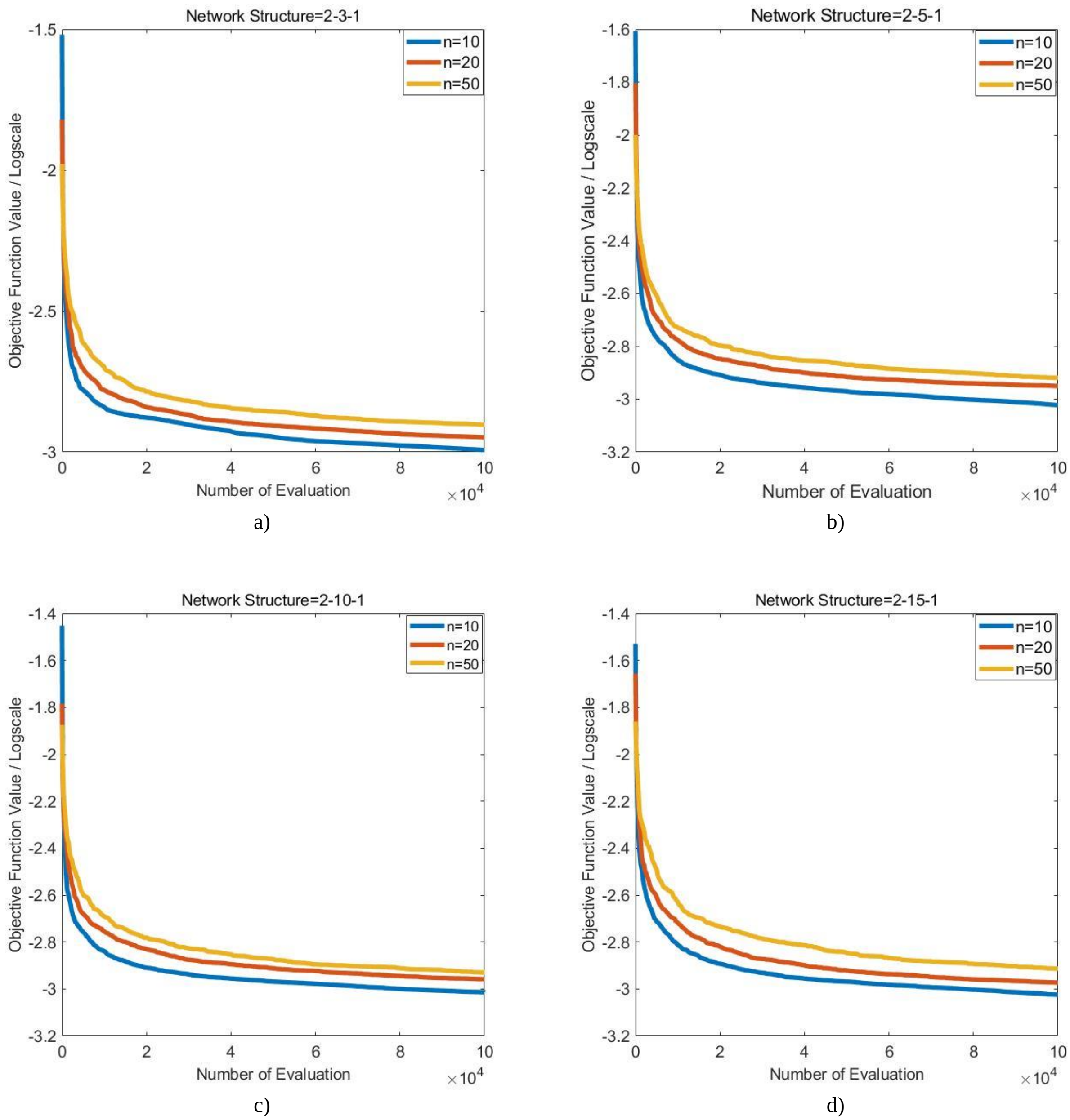

Fig.4. Comparison of effect of population size on convergence for 2 inputs

For S_3, 3, 5, 10 and 15 artificial neurons are used respectively in the hidden layer and 4-3-1, 4-5-1, 4-10-1 and 4-15-1 network structures are obtained. In 4-3-1, the train error values obtained for n = 10, n = 20 and n = 50 are 0.0479, 0.0529 and 0.0586 respectively. For n = 10 and n = 20, close test error values are found. An ineffective error value is found for n = 50. Effective standard deviation values are obtained for train. The test standard deviation values are greater than 0.01. In 4-5-1, the best train and test error values are 0.0472 and 0.0518 respectively. The best standard deviation value for train is reached through n = 50. In 4-10-1, the best train error value is 0.0461. Test error values are 0.0569 and above.

Especially for train, effective standard deviation has been reached. The test standard deviation value is 0.0133 and above. In 4-15-1, the effective training error value is found for n = 10. Test error values are between 0.0569 and 0.0674. Train and test standard deviation values are similar to other network structures.

Population size has affected the solution quality in all systems. At the same time, population size is a factor affecting convergence. The effect of population size on convergence is given in Fig. 4. The convergence speed decreases as colony size increases in all network structures. A good convergence has been obtained for n=10. For n=50, the speed of convergence has decreased.

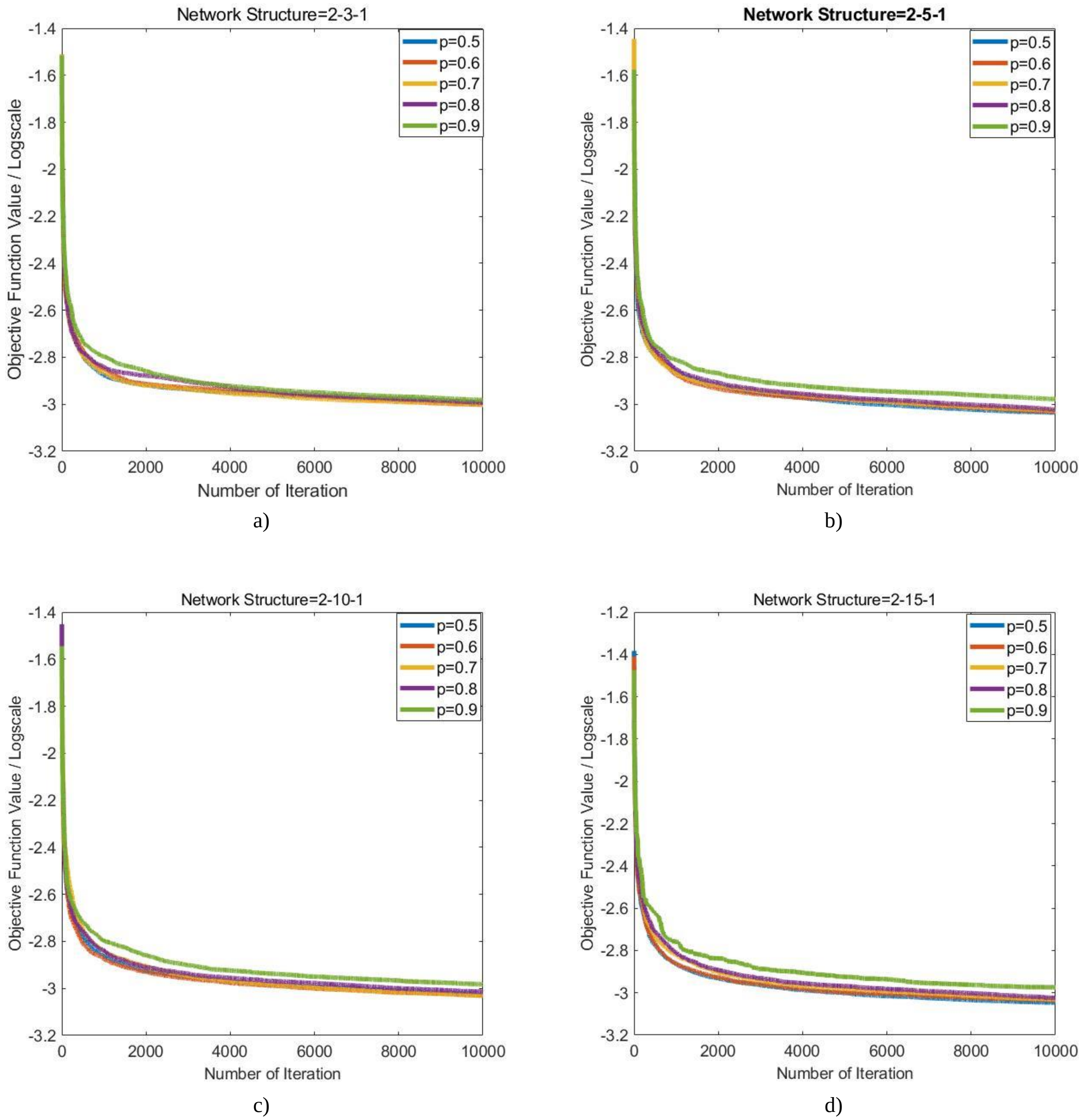

Fig.5. Comparison of effect of switch probability on convergence for 2 inputs

One of the important control parameters of FPA is switch probability (p). The effect of switch probability on solution quality is presented in Table IV. The results are analyzed for different values of p (0.5, 0.6, 0.7, 0.8 and 0.9). Table I shows that the best results are obtained for n = 10. For this reason, switch probability analysis is performed for n = 10 and S_1. In 2-3-1, the best train error value is found with p = 0.6 and p = 0.7. However, the difference between the best result and the worst result is around 2%. The best test error value is obtained as 0.0433 by p=0.8. In 2-5-1, the best train and test error values are found as 0.0480 and 0.0434 respectively. For train, p = 0.5 is more effective. For the test, p is 0.9. In 2-10-1, the effective education error values are obtained with p = 0.6 and p = 0.7. If p = 0.8 and p = 0.9, it is more effective for the test. In 2-15-1, the best train error value is reached with p = 0.5. It is seen that p = 0.8 and p = 0.9 is more successful for the best test error value.

Switch probability affects convergence as well as solution quality. The effect of switch probability on convergence is presented in Fig. 5. In 2-3-1, except for p = 0.9, other p values have similar effects on convergence. In 2-5-1, 2-10-1 and 2-15-1, the worst convergence is reached with p = 0.9. Convergences of other p values are similar to each other as seen in Fig. 5.

TABLE III
COMPARISON OF RESULTS OBTAINED FOR DIFFERENT POLULATION SIZE, NUMBER OF INPUTS AND NETWORK STRUCTURES (p=0.8)

Population Size	Number of Input	Inputs of System	Network Structure	Number of Parameters	The Results			
					Train		Test	
					$Mean_{RMSE}$	Std.	$Mean_{RMSE}$	Std.
n=10	2	S_1	2-3-1	13	0.0501	0.0025	**0.0433**	0.0054
			2-5-1	21	0.0486	0.0020	0.0455	0.0069
			2-10-1	41	0.0490	0.0022	0.0442	0.0077
			2-15-1	61	**0.0485**	0.0022	0.0484	0.0081
	3	S_2	3-3-1	16	0.0476	0.0020	0.0568	0.0109
			3-5-1	26	0.0471	0.0028	**0.0545**	0.0122
			3-10-1	51	0.0486	0.0034	0.0581	0.0127
			3-15-1	76	**0.0467**	0.0030	0.0592	0.0140
	4	S_3	4-3-1	19	0.0479	0.0026	**0.0542**	0.0120
			4-5-1	31	0.0472	0.0030	0.0548	0.0151
			4-10-1	61	**0.0461**	0.0037	0.0569	0.0133
			4-15-1	91	0.0472	0.0039	0.0596	0.0131
n=20	2	S_1	2-3-1	13	0.0524	0.0025	0.0411	0.0076
			2-5-1	21	0.0523	0.0012	**0.0401**	0.0060
			2-10-1	41	0.0519	0.0015	0.0441	0.0063
			2-15-1	61	**0.0511**	0.0020	0.0464	0.0118
	3	S_2	3-3-1	16	**0.0520**	0.0024	0.0571	0.0139
			3-5-1	26	0.0526	0.0018	**0.0524**	0.0106
			3-10-1	51	0.0521	0.0014	0.05647	0.0144
			3-15-1	76	0.0525	0.0022	0.0583	0.0188
	4	S_3	4-3-1	19	**0.0529**	0.0035	0.0545	0.0151
			4-5-1	31	0.0532	0.0034	**0.0518**	0.0137
			4-10-1	61	0.0532	0.0021	0.0601	0.0137
			4-15-1	91	0.0540	0.0025	0.0666	0.0175
n=50	2	S_1	2-3-1	13	0.0548	0.0013	**0.0430**	0.0093
			2-5-1	21	0.0540	0.0012	0.0443	0.0095
			2-10-1	41	0.0534	0.0012	0.0479	0.0134
			2-15-1	61	**0.0506**	0.0012	0.0497	0.0119
	3	S_2	3-3-1	16	0.0560	0.0015	0.0591	0.0144
			3-5-1	26	0.0559	0.0012	**0.0564**	0.0138
			3-10-1	51	**0.0556**	0.0015	0.0651	0.0164
			3-15-1	76	0.0559	0.0017	0.0689	0.0143
	4	S_3	4-3-1	19	0.0586	0.0019	**0.0611**	0.0159
			4-5-1	31	**0.0580**	0.0019	0.0620	0.014
			4-10-1	61	0.0584	0.0017	0.0674	0.0193
			4-15-1	91	0.0599	0.0023	0.0694	0.0172

The network structures used also affect the solution quality. Increasing the number of neurons in the network structure does not mean that the solution quality will be better. Table II shows the effect of network structures on solution quality. The 2-3-1 network structure is insufficient for effective train solutions. It is observed that train achievements are close in other network structures. The effect of network structures on test process is different from train process. The best test error values of p = 0.5 and p = 0.8 are obtained with a network of 2-3-1. The effective results are found with 2-5-1 and 2-10-1 at p=0.6, p=0.7 and p=0.9. In terms of testing, the 2-15-1 network structure has failed compared to the others. The effects of different network structures on convergence are compared and presented in Fig. 6. First of all, this graphic reflects the train process. 2-3-1 network structure has the least impact on convergence. Others have similar performance.

The most important success criterion in artificial neural network training is that the actual output and the estimated output are close to each other. The error decreases as the outputs gets closer together. In order to analyze this situation, the real output-estimated output graph is compared in Fig. 7. As seen in Fig. 7, the outputs are very similar to each other. This shows that the artificial neural network training process using FPA is successful. At the same time, it also shows the FPA is effective in predicting the COVID-19 cases belonging to Turkey.

TABLE IV
COMPARISON OF RESULTS OBTAINED FOR DIFFERENT SWITCH PROBABILITY VALUES (N=10)

Switch Probability	Number of Input	Network Structure	The results	
			$Train_{RMSE}$	$Test_{RMSE}$
p=0.5	2 y(t-1), y(t-2)	2-3-1	0.0502	**0.0459**
		2-5-1	0.0480	0.0477
		2-10-1	0.0484	0.0463
		2-15-1	**0.0475**	0.0479
p=0.6		2-3-1	0.0496	0.0468
		2-5-1	0.0482	**0.0453**
		2-10-1	0.0482	0.0457
		2-15-1	**0.0480**	0.0475
p=0.7		2-3-1	0.0496	0.0476
		2-5-1	0.0484	**0.0457**
		2-10-1	**0.0482**	**0.0457**
		2-15-1	0.0483	0.0465
p=0.8		2-3-1	0.0501	**0.0433**
		2-5-1	0.0486	0.0455
		2-10-1	0.0490	0.0442
		2-15-1	**0.0485**	0.0484
p=0.9		2-3-1	0.0506	0.0456
		2-5-1	0.0508	**0.0434**
		2-10-1	0.0506	0.0443
		2-15-1	**0.0503**	0.0484

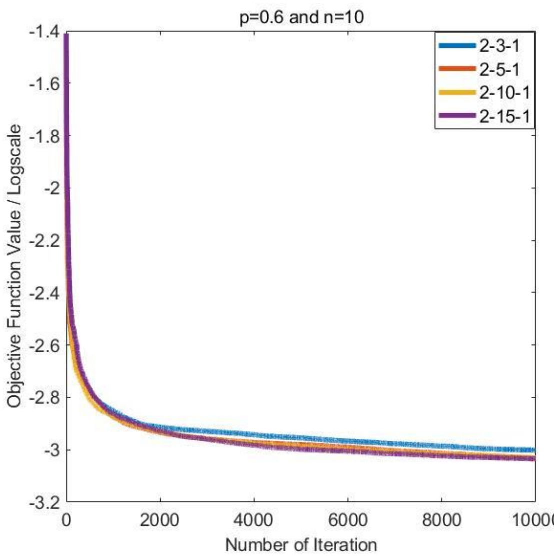

Fig.6. Comparison of effect of network structures on convergence

3 systems (S_1, S_2, S_3) have been used to solve this prediction problem. The effects of the relevant systems on train and test are different. The best train error value is found as 0.0461 with S_3. The test error value corresponding to this value is 0.0569. This is quite high. When the test error is considered, the best test error value was found as 0.0401 with n = 20 and S_1. When Table III and Table IV are evaluated together, it is understood that the results obtained for S_1 are more effective in modeling the problem.

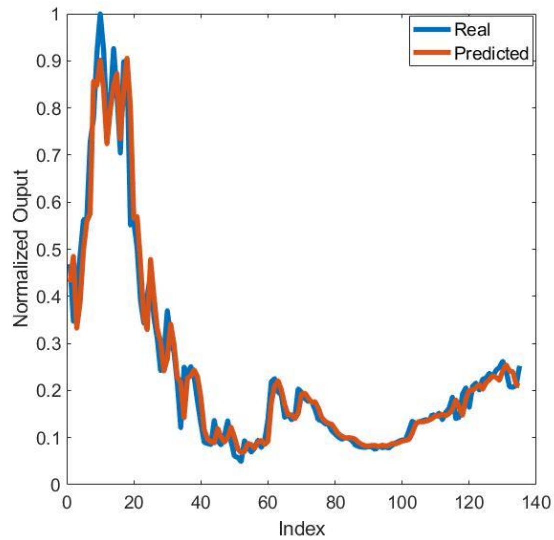

Fig.7. Comparison of real and predicted outputs

It is observed that FPA-based artificial neural network training is successful in solving the related problem. To better analyze the success of FPA, it should be compared with different heuristic algorithms (PSO and HS). Comparison results are given in Table V. For Table V, the control parameters given in Table II are used. When Table V is examined, the best train results in all network structures are obtained with FPA. After FPA, the most effective results are found by utilizing PSO. At the same time, effective standard deviation values have been reached with PSO and FPA.

TABLE V
COMPARISON OF PERFORMANCE OF PSO, HS AND FPA

Algorithm	Network Structure	Mean (RMSE)	Std.
PSO	2-3-1	0.0513	0.0023
	2-5-1	0.0506	**0.0016**
	2-10-1	0.0517	**0.0020**
	2-15-1	0.0531	0.0024
HS	2-3-1	0.0554	0.0057
	2-5-1	0.0520	0.0022
	2-10-1	0.0538	0.0048
	2-15-1	0.0660	0.0126
FPA (Proposed)	2-3-1	**0.0496**	**0.0022**
	2-5-1	**0.0482**	0.0021
	2-10-1	**0.0482**	0.0024
	2-15-1	**0.0480**	**0.0022**

The solutions obtained with FPA appear to be more successful than PSO. To determine this exactly, its significance must be examined. The Wilcoxon signed rank test is used for this and the results are presented in Table VI. The significance analysis is performed according to p = 0.05 level. P values between PSO and FPA are less than 0.05. In the same way, all p values except 2-3-1 are found to be 0.000. p value for 2-3-1 is 0.006. All p values between HS and FPA are obtained as 0.000. This shows that all results obtained with FPA are significant.

TABLE VI
WILCOXON SIGNED RANK TEST RESULT.

Algorithm	Network Structure	p Value	significance
PSO – FPA	2-3-1	0.006	+
	2-5-1	0.000	+
	2-10-1	0.000	+
	2-15-1	0.000	+
HS – FPA	2-3-1	0.000	+
	2-5-1	0.000	+
	2-10-1	0.000	+
	2-15-1	0.000	+

IV. Conclusions

In this study, a hybrid approach based on FPA and neural network to predict number of COVID-19 cases belonging to Turkey is proposed. In the proposed approach, the parameters of the feed forward neural network are optimized by FPA. Namely, the ANN training is carried out using FPA. The data between 1 April 2020 and 15 September 2020 is used to solve the problem. A time series is created with these data and time series analysis is performed.

In order to obtain effective results, different population sizes, different switch probabilities, different network structures and different models are examined. It has been observed that these parameters affect the solution quality and speed of convergence in the solution of the related problem. When the application results are examined, it is seen that the proposed method is effective to predict number of COVID-19 cases.

In order to solve the related problem, ANN has been trained with different heuristic algorithms such as PSO and HS. The performance of FPA is compared to PSO and HS. The Wilcoxon signed rank test is used for the significance analysis of the results. The results show that FPA is generally more successful than PSO and HS to predict number of COVID-19 cases based on neural network.

When the literature is examined, it is seen that the use of FPA in ANN training is limited. This study also reveals the success of FPA in ANN training. At the same time, it shows that FPA can be used in different studies based on neural networks in the future. In particular, a variant of FPA can be proposed to achieve more effective results in estimating the number of COVID-19 cases. Hybrid training algorithms based on FPA can be suggested.

References

[1] A. B. Younes, Z. Hasan. “COVID-19: modeling, prediction, and control.” Applied Sciences, vol. 10, no. 11, 2020, pp. 3666.

[2] L. Duran-Lopez, J. P. Dominguez-Morales, J. Corral-Jaime, S. Vicente-Diaz, A. Linares-Barranco. “COVID-XNet: A custom deep learning system to diagnose and locate COVID-19 in Chest X-ray images.” Applied Sciences, vol. 10, no. 16, 2020, pp. 5683.

[3] R. Pal, A.A. Sekh, S. Kar, D.K. Prasad. “Neural network based country wise risk prediction of COVID-19.” Applied Sciences, vol. 10, no. 18, 2020, pp. 6448.

[4] D. Ezzat, A.E. Hassanien, H.A. Ella. “An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization.” Applied Soft Computing, 2020, pp. 106742.

[5] A.M. Ismael, A. Şengür. “Deep learning approaches for COVID-19 detection based on chest X-ray images.” Expert Systems with Applications, 2020, pp. 114054.

[6] C. Zhan, Y. Zheng, Z. Lai, T. Hao, B. Li. “Identifying epidemic spreading dynamics of COVID-19 by pseudocoevolutionary simulated annealing optimizers.” Neural Computing and Applications, 2020, pp. 1-14.

[7] M.A. Al-Qaness, A.A. Ewees, H. Fan, M. Abd El Aziz. “Optimization method for forecasting confirmed cases of COVID-19 in China.” Journal of Clinical Medicine, vol. 9, no. 3, 2020, pp. 674.

[8] P. Melin, J.C. Monica, D. Sanchez, O. Castillo, “Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: The case of Mexico.” In Healthcare, vol. 8, no. 2, 2020, June, pp. 181, Multidisciplinary Digital Publishing Institute.

[9] M.A. Al-Qaness, A.A. Ewees, H. Fan, L. Abualigah, M. Abd Elaziz. “Marine predators algorithm for forecasting confirmed cases of COVID-19 in Italy, USA, Iran and Korea.” International Journal of Environmental Research and Public Health, vol. 17, no. 10, 2020, pp. 3520.

[10] A.I. Saba, A.H. Elsheikh. “Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks.” Process Safety and Environmental Protection, 2020.

[11] S. Lalmuanawma, J. Hussain, L. Chhakchhuak. “Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review.” Chaos, Solitons & Fractals, 2020, pp. 110059.

[12] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, ... , D. Shen. “Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for Covid-19.” IEEE Reviews in Biomedical Engineering, 2020.

[13] R. Vaishya, M. Javaid, I. H. Khan, A. Haleem. “Artificial Intelligence (AI) applications for COVID-19 pandemic.” Diabetes & Metabolic Syndrome: Clinical Research & Reviews, vol. 14, no. 4, 2020.

[14] Y. Mohamadou, A. Halidou, P.T. Kapen. “A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19.” Applied Intelligence, vol. 1, no. 13, 2020.

[15] I.E. Agbehadji, B.O. Awuzie, A.B. Ngowi, R.C. Millham. “Review of big data analytics, artificial intelligence and nature-inspired computing models towards accurate detection of COVID-19 pandemic cases and contact tracing.” International Journal of Environmental Research and Public Health, vol. 17, no. 15, 2020, pp. 5330.

[16] S. Dreiseitl, L. Ohno-Machado. “Logistic regression and artificial neural network classification models: A methodology review.” Journal of Biomedical Informatics, vol. 35, no. 5-6, 2002, pp. 352-359.

[17] P. J. Lisboa, A.F. Taktak. “The use of artificial neural networks in decision support in cancer: A systematic review.” Neural Networks, vol. 19, no. 4, 2006, pp. 408-415.

[18] A. Tealab. “Time series forecasting using artificial neural networks methodologies: A systematic review.” Future Computing and Informatics Journal, vol. 3, no. 2, 2018, pp. 334-340.

[19] R. K. Dase, D.D. Pawar.” Application of artificial neural network for stock market predictions: A review of literature.” International Journal of Machine Intelligence, vol. 2, no. 2, 2010, pp. 14-17.

[20] A. Dhillon, G.K. Verma. “Convolutional neural network: A review of models, methodologies and applications to object detection.” Progress in Artificial Intelligence, vol. 9, no. 2, 2020, pp. 85-112.

[21] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi. “Metaheuristic research: a comprehensive survey.” Artificial Intelligence Review, vol. 52 no.4, 2009, pp. 2191-2233.

[22] S. Aslan, S. Demirci. “Plazma tedavisi temelli yeni bir optimizasyon algoritması.” IEEE Signal Processing and Communications Applications Conference (SIU), 2021, pp. 1-4.

[23] S. Aslan, S. Demirci. “Immune plasma algorithm: a novel meta-heuristic for optimization problems.” IEEE Access, vol. 8, 2020, pp. 220227-220245.

[24] C. Harpham, C.W. Dawson, M.R. Brown. "A review of genetic algorithms applied to training radial basis function networks." Neural Computing & Applications, vol. 13, no. 3, 2004, pp. 193-201.

[25] Y. Azimi, S.H. Khoshrou, M. Osanloo. "Prediction of blast induced ground vibration (BIGV) of quarry mining using hybrid genetic algorithm optimized artificial neural network." Measurement, vol. 147, 2019, pp. 106874.

[26] K. Taheri, M. Hasanipanah, S.B. Golzar, M.Z. Abd Majid. "A hybrid artificial bee colony algorithm-artificial neural network for forecasting the blast-produced ground vibration." Engineering with Computers, vol. 33, no. 3, 2017, pp. 689-700.

[27] D. Karaboga, B. Akay, C. Ozturk. "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks." International conference on modeling decisions for artificial intelligence, Springer, Berlin, Heidelberg, August 2007, pp. 318-329.

[28] J. Yu, S. Wang, L. Xi. "Evolving artificial neural networks using an improved PSO and DPSO." Neurocomputing, vol. 71, no. 4-6, 2008, pp. 1054-1060.

[29] S.K. Satapathy, S. Dehuri, A.K. Jagadev. "EEG signal classification using PSO trained RBF neural network for epilepsy identification." Informatics in Medicine Unlocked, vol. 6, 2017, pp. 1-11.

[30] J. Saadat, P. Moallem, H. Koofigar. "Training echo state neural network using harmony search algorithm." Int. J. Artif. Intell, vol. 15, no. 1, 2017, pp. 163-179.

[31] S. Kulluk, L. Ozbakir, A. Baykasoglu. "Training neural networks with harmony search algorithms for classification problems." Engineering Applications of Artificial Intelligence, vol. 25, no.1, 2012, pp. 11-19.

[32] J. Ilonen, J.K. Kamarainen, J. Lampinen. "Differential evolution training algorithm for feed-forward neural networks." Neural Processing Letters, vol. 17, no. 1, 2003, pp. 93-105.

[33] N. Chauhan, V. Ravi, D.K. Chandra. "Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks." Expert Systems with Applications, vol. 36, no. 4, 2009, pp. 7659-7665.

[34] J. Nayak, B. Naik, D. Pelusi, A.V. Krishna. "A comprehensive review and performance analysis of firefly algorithm for artificial neural networks." In Nature-Inspired Computation in Data Mining and Machine Learning, 2020, Springer, Cham, pp. 137-159.

[35] E. Valian, S. Mohanna, S. Tavakoli. "Improved cuckoo search algorithm for feedforward neural network training." International Journal of Artificial Intelligence & Applications, vol. 2, no.3, 2011, pp. 36-43.

[36] X. Liang, W. Liang, J. Xiong. "Intelligent diagnosis of natural gas pipeline defects using improved flower pollination algorithm and artificial neural network." Journal of Cleaner Production, 2020, 121655.

[37] P.A. Kowalski, K. Wadas. "Triggering probabilistic neural networks with flower pollination algorithm." Computational Intelligence and Mathematics for Tackling Complex Problems, 2020, Springer, Cham, pp. 107-113.

[38] P. Dutta, A. Kumar. "Modeling and optimization of a liquid flow process using an artificial neural network-based flower pollination algorithm." Journal of Intelligent Systems, vol. 29, no.1, 2018, pp. 787-798.

[39] G.S. Shehu, N. Çetinkaya. "Flower pollination–feedforward neural network for load flow forecasting in smart distribution grid." Neural Computing and Applications, vol. 31, no. 10, 2019, pp. 6001-6012.

[40] M.H.B.A. Yazid, M.S. Talib, M.H. Satria. "Flower pollination neural network for heart disease classification." IOP Conference Series: Materials Science and Engineering, IOP Publishing, vol. 551, no. 1, August 2019, pp. 012072.

[41] L. Pan, X. Feng, F. Sang, L. Li, M. Leng, X. Chen. "An improved back propagation neural network based on complexity decomposition technology and modified flower pollination optimization for short-term load forecasting." Neural Computing and Applications, vol. 31, no. 7, 2019, pp. 2679-2697.

[42] Y. Ren, H. Li, H.C. Lin. "Optimization of feedforward neural networks using an improved flower pollination algorithm for short-term wind speed prediction." Energies, vol. 12, no. 21, 2019, pp. 4126.

[43] S. Chatterjee, B. Datta, N. Dey. "Hybrid neural network based rainfall prediction supported by flower pollination algorithm." Neural Network World, vol. 28, no. 6, 2018, pp. 497-510.

[44] M. Abdel-Basset, L.A. Shawky. "Flower pollination algorithm: A comprehensive review." Artificial Intelligence Review, vol. 52, no. 4, 2019, pp. 2533-2557.

[45] A.E. Kayabekir, G. Bekdaş, S.M. Nigdeli, X.S. Yang. "A comprehensive review of the flower pollination algorithm for solving engineering problems." Nature-Inspired Algorithms and Applied Optimization, Springer, Cham, 2018, pp. 171-188.

[46] X.S. Yang. "Flower pollination algorithm for global optimization." Unconventional Computation and Natural Computation, edited by J. Durand-Lose and N. Jonoska, Berlin: Springer, vol. 7445 of Lecture Notes in Computer Science, 2012, pp. 240–249.

[47] K. Lachhwani. "Application of neural network models for mathematical programming problems: A state of art review." Archives of Computational Methods in Engineering, vol. 27, no. 1, 2020, pp. 171-182.

[48] T. K. Gupta, K. Raza. "Optimization of ANN architecture: A review on nature-inspired techniques." Machine Learning in Bio-Signal Analysis and Diagnostic Imaging, Academic Press, 2019, pp. 159-182.

BIOGRAPHIES

CEREN BAŞTEMUR KAYA received the Ph.D. degree in Department of Computer and Instructional Technologies Education from Gazi University, in 2018. She has been working in Department of Computer Technologies, Nevsehir Vocational College, Nevsehir Haci Bektas Veli University since 2011. She is an assistant professor and also the head of the department now. Her research fields consist education in engineering, virtual reality and artificial intelligence techniques such as heuristics, artificial neural networks and neuro-fuzzy.

EBUBEKİR KAYA received the Ph.D. degree in Department of Computer Engineering from Erciyes University, in 2017. He has been working in Department of Computer Engineering, Nevsehir Haci Bektas Veli University. He is an assistant professor and also the head of the department now. He also engages in entrepreneurship activities. His research fields consist smart city, smart home, embedded systems, image processing and artificial intelligence techniques such as heuristics, artificial neural networks and neuro-fuzzy.