Metaheuristic Algorithms in Artificial Neural Network Training

Münevver BAHAR
Necmettin Erbakan University

Halil İbrahim AYAZ
Necmettin Erbakan University

Murat KARAKOYUN
Necmettin Erbakan University

To Cite This Chapter

Bahar, M., Ayaz, H. İ., & Karakoyun, M. (2024). Metaheuristic Algorithms in Artificial Neural Network Training. In S. Koçer & Ö. Dündar (Eds.), *Intelligent Systems and Optimization in Engineering* (pp. 33-52). ISRES Publishing.

Introduction

Various methods are used today for problems in different sectors. Artificial Neural Networks (ANN) technology is one of the newest among these methods and has been successfully applied in many areas. ANN aims to imitate and develop the working principles of the human brain and to perform the basic functions that the human brain performs biologically with a suitable software (Çörekcioğlu et al., 2021). ANN architecture consists of the connections of simple processing units called nodes or neurons. Between the input and output layers, there could be one or several hidden layers, and the connection between each node has a certain weight (Kaytan et al., 2020).

In ANN, operations such as classification, modeling, optimization or prediction are performed by updating the connection settings in the network; this process is called learning. This process, which is carried out in order to increase the performance of the network, is carried out in the form of iterations in computer programs (Çalışkan & Deniz, 2015). In the learning process, the most widely used algorithm for multilayer feedforward networks is the "Back Propagation" (BP) training algorithm, known as backpropagation (Ticknor, 2013). The backpropagation algorithm focuses on minimizing the difference between the target output values and the output values produced by the network as the iterations progress in the training process, and it relies on adjusting the network's connection weights to reduce this difference (Özkan, 2012).

In the ANN model, the weights are usually updated during the backpropagation process with derivative-based techniques (Jiadong et al., 2024). However, derivative-based techniques may not be able to successfully update the weight values in challenging problems, which may cause the weights to get stuck at local minimum points (Emambocus et al., 2023; Karakoyun, 2024; F. N. Özdemir & Özkış, 2024). To overcome this problem, researchers have used metaheuristic algorithms in the backpropagation phase of the network in many studies. Metaheuristic algorithms are strategies designed to solve various problems by mathematically simulating the behaviors of natural entities such as humans, animals, and plants. These algorithms aim to reach the best solution in the solution space faster by using effective search techniques in a high-level working environment (Çelik, 2013).

Given that training artificial neural networks is both a critical and complex task, it has been the focus of extensive research. In recent years, particularly, various metaheuristic algorithms have been applied to address this issue (Karakoyun, 2024). It is virtually impossible to thoroughly review all related studies in the literature due to time constraints. Therefore, this study's literature review section highlights a selection of recent and significant works in the field.

Özdemir (F. N. Özdemir & Özkış, 2024) developed a hybrid model with the Snow Ablation Optimizer (SAO) algorithm to update the weights of the artificial neural network. The developed hybrid model was compared with hybrid models created with gray wolf, reptile search, cuckoo and sine cosine algorithms on five different data sets and achieved the best result with the SAO model in terms of average success order with a value of 1.2 in all metrics. Aksu et al. (Aksu et al., 2022) used two different estimation methods based on multilayer neural network to provide reliable estimation of solar radiation. The network coefficients and bias values of the neural network were trained using Imperialist Competitive Algorithm (ICA) and Particle Swarm Optimization algorithm (PSO). Ateş (Ateş, 2022) created a hybrid approach that combines a multilayer ann model with PSO and the Cultural Algorithm (CA) to achieve minimal error in short-term PV panel output power predictions. Özmen et al. (Özmen et al., 2023) worked on early detection of diabetes by reducing the number of features with metaheuristic methods. They performed feature selection using Salp Swarm Algorithm (SSA), Artificial Bee Colony Algorithm, Whale Optimization Algorithm (WOA) and Ant Colony Algorithm (ACO) with examples from UCI (UCI Machine Learning Repository) data repository. For the evaluation of selected features, K-Nearest Neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM) and ANN methods were used to calculate accuracy, sensitivity and specificity parameters. Ayaz (Ayaz & Kamisli Ozturk, 2021) addressed optimization challenges in train seat planning by applying heuristic approaches and parallel machine scheduling to minimize waiting times and maximize resource utilization. Köprü (Köprü, 2020) used artificial neural network to estimate the amount of liquid crude iron produced with monthly raw material information of blast furnace enterprise. Zaimoğlu (Zaimoğlu, 2023) developed a new approach called Binary Chaotic Horse Herd Optimization Algorithm (BCHOAFS) by augmenting the proposed binary version of HOA with five different well-known chaotic maps in order to increase the success and stability of the algorithm. Jama (Jama, 2021) presented a modified version of bio-inspired Ant Lion Optimization Algorithm (ALO) to solve the region growing segmentation problem. Farahani et al (Shahvaroughi Farahani & Razavi Hajiagha, 2021) sought to forecast stock price indices using an ANN and trained it with recent metaheuristic algorithms like Social Spider Optimization (SSO) and the Bat Algorithm (BA). They employed the Genetic Algorithm (GA), a heuristic method, for feature selection and identifying the most relevant indicators. Mu'azu (Abdullahi Mu'azu, 2023) aimed to optimize the hybrid configuration of ANN with Cuttlefish Optimization Algorithm (CFOA), Electrostatic Discharge Algorithm (ESDA) and Henry Gas Solubility Optimization Algorithm (HGSOA) and Sine Cosine Algorithm (SCA) algorithms for soil BC analysis.

Other examples of studies in this area include the enhanced SSA for training multilayer sensors (MLS) (Atlı, 2022), the application of the PSO algorithm in photovoltaic (PV) energy systems (A. Özdemir & Pamuk, 2021), the use of the Vibration Particle System Algorithm for fine-tuning weight matrices (Özkaya et al., 2021), and the development of a hybrid INFO-Simulated Annealing Algorithm to optimize the carrier arm of drones in unmanned aerial vehicles (Yildiz, 2023), among others.

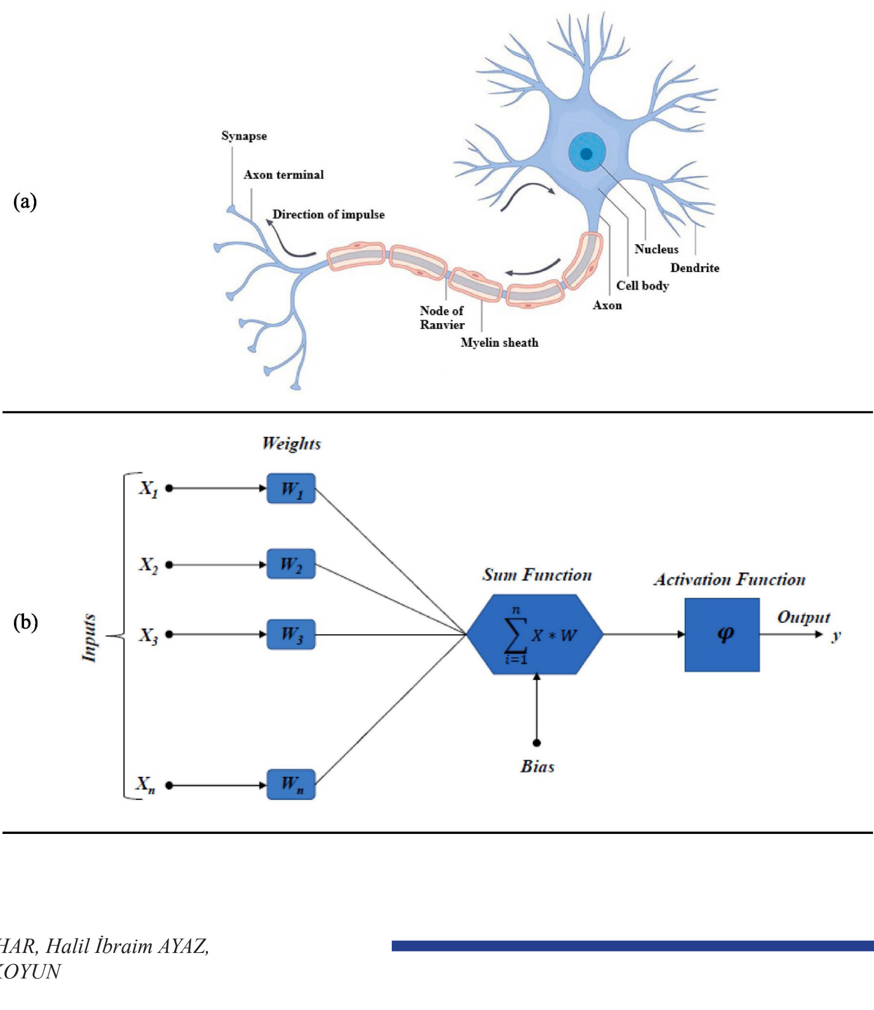
In this study, the performances of the metaheuristic algorithms Grasshopper Optimization Algorithm (GOA) (Saremi et al., 2017), Artificial Hummingbird Algorithm (AHA) (Zhao et al., 2022), Arithmetic Optimization Algorithm (AOA) (Abualigah et al., 2021), Crayfish Optimization Algorithm (COA) (Jia et al., 2023), Artificial Bee Colony (ABC) (Karaboga, 2005) and Tree-seed algorithm (TSA) (Sahman et al., 2019) were evaluated on 21 different datasets. The performance evaluation was performed using various metrics (precision, specificity, F1-score and sensitivity).

Artificial Neural Network

ANNs are computer software that perform basic functions such as learning, remembering, generalizing and producing new information from the obtained data by imitating the learning processes of the human brain. ANNs are used for various purposes such as pattern recognition, classification, modeling, optimization and prediction (Rençber, 2018).

The development of artificial neural networks began with research on the working principles of the human brain, and an important step was taken in 1943 when McCulloch and Pitts developed the first artificial neural network model. In the 1950s, studies in the field of artificial neural networks gained momentum with Hebb's learning theory and Rosentblatt's "Perceptron" model, but these studies entered a period of stagnation in the 1960s due to artificial intelligence research. From the 1980s onwards, artificial neural networks began to attract attention again, and their popularity increased with Hopfield's creation of the mathematical foundations of networks and Rummelhart's parallel programming studies. During this process, developments in computer hardware also contributed to the integration of artificial neural networks into practical applications (Keskenler & Keskenler, 2017). Figure 1.a shows a biological neuron, and Figure 1.b shows an artificial neuron model.

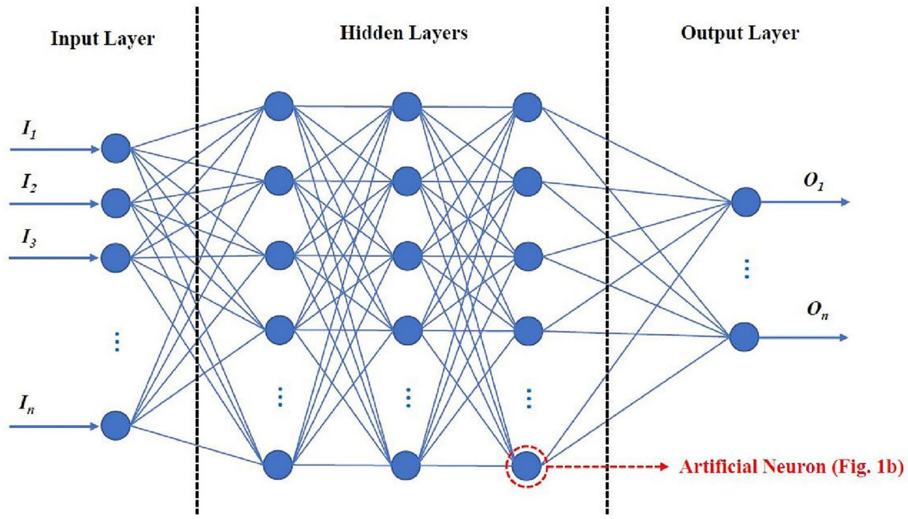
Figure 1 a-b)
Biological Neuron And Artificial Neuron Model (Karakoyun, 2024)



ANN models, which are based on the principle of learning based on experience, aim to produce a single output from many inputs. The basic component of this technique is the processing elements known as neurons (Çınaroğlu & Avcı, 2020). Artificial nerve cells, or neurons, in the network have the ability to make predictions about similar examples that they have not encountered before by comprehending an event based on data, with or without supervision. Neurons are organized in logical groups called layers (Çalışkan & Deniz, 2015).

The structure of ANN is divided into two main groups as single layer perceptron (SLP) and multilayer perceptron (MLP). The first studies on ANN focused on the SLP architecture. However, when it was understood that SLPs could only solve linear problems, the MLP architecture that can also learn nonlinear problems was developed. MLPs have another layer called the intermediate (hidden) layer in addition to the input and output layers (F. N. Özdemir & Özkış, 2024). Hidden layers are responsible for transmitting signals from the input layer to the output layer and do not have direct connections with the external environment. The general structure of the MLP is shown in Figure 2.

Figure 2
General Structure Of The Mlps (Karakoyun, 2024)



Each unit in the network calculates the weighted sum of the input data coming to it. This sum is obtained by multiplying the input data by the connection weights. The data progresses with this process in each layer of the network. First, the weights are randomly assigned. In the first hidden layer, the multiplication results are collected and these results are transferred to the next hidden layer or output layer by being subjected to the activation function. The activation function undertakes the function of processing the input value and converting it into an output value. The most commonly used activation function in MNEs (Multilayer Artificial Neural Networks) is the sigmoid function, which produces outputs ranging from 0 to 1. Thanks to the activation function, the network is transformed into a non-linear structure as a result of the operations performed in the hidden layers, which provides an advantage in solving complex problems (Akel & Karacameydan, 2012).

In the MLP architecture, the learning process is divided into two stages: forward computation and backward computation. Forward computation creates the output of the network, while backward computation deals with updating the weights. This stage of the learning process is performed by the backpropagation algorithm (Ticknor, 2013). The backpropagation algorithm focuses on minimizing the difference between the targeted output and the output produced by the network as iterations progress in the training process and updates the network connection weights to minimize this difference.

Although various methods are used to evaluate error, the most commonly preferred method is the mean square error (MSE). MSE is a technique often used to measure the difference (error rate) between expected and predicted values in machine learning systems (Gölcük et al., 2023). Equation 1. Provides the mathematical notation of MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(p_i - a_i\right)^2 \quad (1)$$

In this equation, p_i is the predicted output value, a_i is the actual output value, and n is the number of samples used in the training process. In summary, the goal of MLPs is to find the biases and weights that minimize the MSE value. In other words, a lower MSE value indicates a more effective training process, while a higher MSE value indicates a more inefficient training process (Karakoyun, 2024).

ANN generalize over the examples presented to the system during the training process. When a change occurs in any dimension of the problem, the network may need to be retrained. The ability of the network to quickly adapt to new situations increases the probability of reaching the desired output faster (Çınaroğlu & Avcı, 2020). The performance of the trained network is evaluated on the test dataset that was not used during training, a process called 'testing the network'. The success in the testing phase is measured by metrics such as precision, specificity, F1-score and sensitivity. These metrics are discussed in detail in Section 5.

Metaheuristic Algorithms

The term metaheuristic refers to high-level heuristic strategies designed to solve a broad range of optimization problems. In recent years, many metaheuristic algorithms have been successfully applied to tackle complex and difficult problems. The attractiveness of these algorithms lies in their ability to find the best or optimal solutions for even very large problem instances within a relatively short time frame (Dokeroglu et al., 2019).

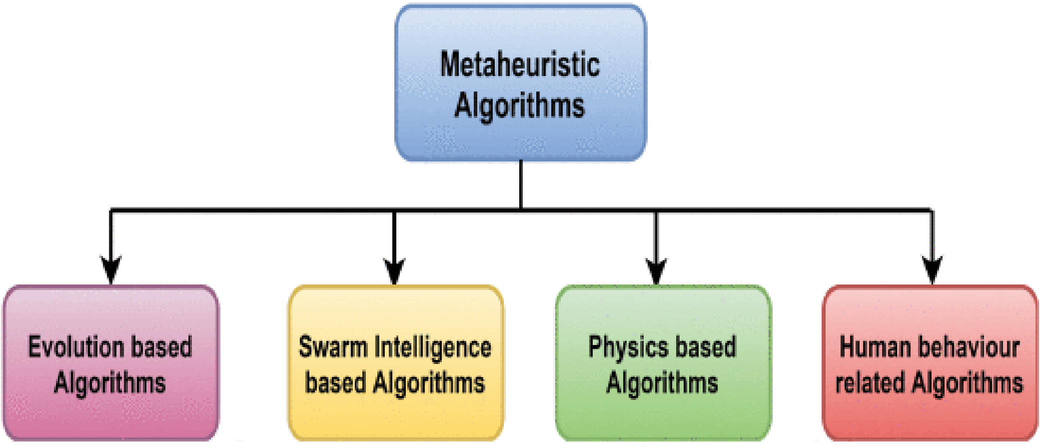
Metaheuristic methods serve three primary purposes: rapidly solving problems, handling large-scale problems, and creating more robust algorithms. These approaches are not only straightforward to design but also flexible and easy to implement. Typically, metaheuristic algorithms utilize a mix of rules and randomization to replicate natural phenomena (Rere et al., 2016).

Metaheuristic algorithms exhibit stochastic behavior, initiating their optimization process by producing random solutions. Unlike gradient-based search methods, they do not require the calculation of search space derivatives. These algorithms are valued for their flexibility and simplicity, owing to their straightforward concepts and ease of implementation. They can be easily adjusted to suit specific problems. A key characteristic of metaheuristic algorithms is their exceptional ability to avoid premature convergence (Agrawal, 2021). Because of their stochastic nature, these techniques function like a hidden mechanism, efficiently avoiding local optima and thoroughly exploring the search space.

Metaheuristic algorithms can be divided into four main categories based on their behavior: evolution-based, swarm intelligence-based, physics-based, and human behavior-based algorithms (Mohamed et al., 2020). These categories are shown in Figure 3.

Figure 3

Classification Of Metaheuristic Algorithms (Agrawal, 2021)



Evolution-based algorithms are inspired by the natural evolution process and start with a randomly generated population of solutions. These algorithms combine the best solutions to create new individuals; in this process, methods such as mutation, crossover and selection of the most suitable solution are used (Mohamed et al., 2020). The most well-known example of this category is the GA based on Darwin's theory of evolution (Holland, 1992).

Swarm intelligence-based algorithms are inspired by the social behavior of living things such as insects, animals, fish and birds (Agrawal, 2021). One of the most popular techniques in this field is PSO, developed by Kennedy and Eberhart (Kennedy & Eberhart, 1995), which is based on the behavior of a flock of birds flying through the search space and finding their best positions.

Physics-based algorithms are inspired by the physical laws that exist in the universe. Algorithms in this category, such as Simulated Annealing (Kirkpatrick et al., 1983) and Harmony Search (Geem et al., 2001), perform optimization by imitating physical processes. Finally, algorithms inspired by human behavior are inspired by the performance and methods that people exhibit when performing different activities. Popular methods include Teaching-Learning Based Optimization (TLBO) (Rao et al., 2012) and League Championship Algorithm (Kashan, 2009).

In this research article, the new generation metaheuristic algorithms such as Grasshopper Optimization Algorithm (Saremi et al., 2017), Artificial Hummingbird Algorithm (Zhao et al., 2022), Arithmetic Optimization Algorithm (Abualigah et al., 2021), Crayfish Optimization Algorithm (Jia et al., 2023), Artificial Bee Colony (Karaboga, 2005) and Tree-Seed Algorithm (Sahman et al., 2019) were used.

Grasshopper Optimization Algorithm

GOA is a new Swarm Intelligence method inspired by the swarm behavior of locusts in nature. This algorithm was proposed by Saremi et al. in 2017 (Saremi et al., 2017). Literature shows that this algorithm is used to solve various optimization problems such as feature selection, scheduling, load frequency control, economic dispatch, engineering, etc. (Meraihi et al., 2021).

The algorithm simulates the repulsion and attraction forces between locusts. Repulsion forces enable the locusts to explore the search space, while attraction forces guide them towards promising areas. GOA includes a factor that progressively decreases the locusts' comfort zone, ensuring a balance between the exploration phase (global search) and exploitation phase (local search) during the optimization. This mechanism assists GOA in accurately approximating the global optimum, reducing the risk of getting trapped in a local optimum. As the best solution found by the swarm so far becomes the target for the swarm to follow, the locusts significantly increase their chances of locating the global optimum by enhancing the target over the optimization process.

Artificial Hummingbird Algorithm

AHA is a meta-heuristic optimization method modeled after the feeding strategies of hummingbirds in the wild. Hummingbirds possess three distinct flight abilities: axial, diagonal, and omnidirectional. Additionally, their memory capacity to choose the optimal food source plays a crucial role in the AHA algorithm. AHA replicates three types of foraging behaviors during the optimization process: directed foraging, territorial foraging, and migratory foraging (Aslanov et al., 2023; Bakır, 2024; Zhao et al., 2022).

In the AHA algorithm, foraging mimics the flight abilities of hummingbirds when searching for food, and this step represents the process of exploring the solution space. Hummingbirds' ability to recall the best food sources through their memories and return to these sources helps the algorithm discover potential solutions. Territorial foraging reflects hummingbirds' behavior in protecting food sources, and represents the stage in which the algorithm focuses on promising solutions and improves these areas. Migratory foraging is the diversification stage, where the algorithm explores new solution spaces, such as when birds migrate to new areas when food sources are scarce, and aims to find better global solutions by avoiding local optima (Khodadadi et al., 2023; Zhao et al., 2022).

Arithmetic Optimization Algorithm

AOA is a metaheuristic optimization algorithm that uses arithmetic operations to solve global optimization problems. Suitable for both discrete and continuous problems, AOA aims to find the best solution in the search space and searches for solutions using arithmetic operations in this process (Abualigah et al., 2021).

The algorithm works in two main stages: In the first stage, a wide search space is scanned to discover potential solutions, and in the second stage, these solutions are optimized in a narrower area. Initially, a random population of solutions is created and new solutions are obtained by applying arithmetic operators to these solutions. In this process, the quality of each candidate solution is evaluated and the best solutions are constantly updated (Abualigah et al., 2021; Gölcük et al., 2023).

The simplicity of AOA both facilitates its implementation and enables it to be used in a wide range of optimization problems. It can be used effectively in many areas such as engineering design, artificial intelligence and energy system optimization. AOA was developed inspired by arithmetic operations in nature and has achieved successful results in different optimization problems (Dhal et al., 2023).

Crayfish Optimization Algorithm

COA is a metaheuristic optimization algorithm developed by modeling the behavior of crayfish in aquatic ecosystems. This algorithm mimics biological and environmental interactions in nature to solve various optimization problems. COA has a population-based structure; a population consisting of a certain number of crayfish individuals represents a solution in the search space (Jia et al., 2023).

The movements of crayfish under water occur for various reasons such as searching for food, avoiding dangers, and mating, and these movements are used to find new solutions in the solution space during the optimization process. The behavior of crayfish scanning their surroundings and finding the best food while searching for food is used as local and global search strategies to find the best solution in COA. While the instinct to avoid dangers aims to avoid bad solutions and find better solutions, social interactions accelerate the sharing of information within the community and the optimization process (Jia et al., 2023, 2024).

Its advantages include the ability to focus on the best solutions while maintaining the diversity among solutions and adapting to various optimization problems with its nature-inspired structure. However, the effectiveness of the algorithm depends on the parameters used, and these parameters need to be adjusted correctly. Also, COA, like other metaheuristic algorithms, has the risk of getting stuck in local optima. COA is an important tool for researchers who are interested in optimization algorithms, especially those inspired by nature and modeling biological processes (Jia et al., 2023, 2024).

Artificial Bee Colony

ABC Algorithm was developed by Karaboğa in 2005 (Karaboga, 2005) and was inspired by the natural food-gathering behavior of honeybees. This algorithm solves optimization problems by simulating the processes of bee colonies in nature to find efficient food sources. ABC is used in various engineering and scientific problems as a metaheuristic optimization algorithm (Karaboga & Basturk, 2007).

The ABC algorithm mimics the behavior of three different types of bees: worker bees, observer bees, and scout bees. Worker bees evaluate existing food sources (solutions) and try to improve these sources. Observer bees focus on the most efficient sources based on the information received from worker bees. If worker bees cannot develop a solution, they turn into scout bees and wander randomly to search for new food sources. This process continues until the specified number of iterations and solutions are developed at each stage (Karaboga, 2005; Karaboga & Basturk, 2007).

The basic operation of the ABC algorithm is similar to the food source-finding behavior of bees in nature. Bees move towards more efficient sources and away from inefficient ones. In this way, the algorithm tries to find the best solutions for optimization problems. Especially in solving nonlinear and complex problems, the ABC algorithm attracts attention with its flexibility and efficiency (Karaboga, 2005; Karaboga & Akay, 2009).

Tree-Seed Algorithm

TSA is an optimization algorithm inspired by the seed propagation and sprouting processes of trees in nature. This algorithm symbolizes the formation of new individuals by spreading the seeds of trees in various ways. TSA is applied to solve complex optimization problems by imitating these natural processes (Sahman et al., 2019).

At the heart of TSA is seed dispersal, where each seed is generated based on either the optimal or a randomly chosen tree position within the population. After creating three potential positions, they are evaluated against the objective function of the problem. For every tree, two different strategies exist for generating seeds, and this selection is guided by the algorithm's primary control parameter, known as Search Bias (ST). Once the seeds are assessed using the objective function, those with superior fitness compared to the current tree positions are selected to form the next generation. This process of seed production and growth is repeated until the algorithm reaches the maximum number of fitness evaluations (Kiran, 2015; Sahman et al., 2019).

Parameter Settings

Some parameters used in the implementation of the algorithms are common. These common parameters and their values are as follows: number of runs 25, population size 50, search space limits [-10, 10] and maximum fitness evaluation (maxFEs) 20,000. In addition, some algorithms have some particular parameters. These params and their associated values utilized in this study are presented in Table 1.

Table 1
Particular Parameters Of The Comparison Algorithms

Algorithms	Params
GOA	cMax = 1, cMin = 0.00004
AHA	No particular parameter
AOA	C1 = 2, C2 = 6, C3 = 1, C4 = 2, u = 0.9, l = 0.1
COA	No particular parameter
ABC	limit = 100
TSA	ST = 0.1, least_seed = 0.1, most_seed = 0.25

The Implementation of the Metaheuristic Algorithms in Training of ANN

The primary goal of training an ANN is to optimize the network's biases and weights. For this reason, the optimization algorithms used in the training process focus on solution sets consisting of biases and weights (Karakoyun, 2024). In this article, the architecture of MLPs was generated dynamically by considering the features and classes in the dataset used. In determining the MLP structure, a set of rules stated below were followed.

- The count of hidden neurons = (2 * number of attribute) + 1
- The count of biases = the count of hidden neurons + the count of classes
- The count of weights = (the count of attributes * the count of hidden neurons) + (the count of classes * the count of hidden neurons)
- Dimension of the problem (solution) = the count of biases + the count of weights

In a MLP, the count of hidden neurons refers to the total count of neurons present in the hidden layer. The bias count represents the total biases associated with neurons in both the hidden and output layers, as each neuron in the MLP requires a bias term. The count of weights represents the total connections between the neurons in the hidden layer and the input/output layers. Lastly, the size of the problem is determined by summing the weight and bias numbers.

A key challenge in training ANNs is designing the MLP architecture and defining the solution vector representation. After this issue is solved, metaheuristic algorithms were applied to ANN training by combining them with other steps. In this process, first, the data to be used for training and testing the model is read from a file. If the data is not separated as training and test, it is separated as training and test data in this step. Then, the MLP model is created according to the characteristics of the data and a solution vector suitable for the MLP structure is designed. Weights and biases are optimized using optimization algorithms. It is checked whether the data is separated as training and test; if not, the results obtained using the training data are returned as output, if separated, the model trained with the training data is applied on the test data and the results are produced. Finally, both training and test results are evaluated with various metrics and presented.

Experimental Results

In this section, the datasets and comparison metrics used in the study are introduced. Subsequently, comprehensive comparative findings using various metrics and approaches are provided.

Data Sets

In this study, a large dataset collected from various sources was used (Karakoyun, 2024; Qaddoura et al., 2020). A total of 21 different datasets were examined. The number of features in these datasets varied between 2 and 34, while the number of classes varied between 2 and 8. While the appendicitis dataset had the fewest samples with 106 samples, the Aniso, Blobs and Varied datasets had the largest samples with 1500 samples. The other datasets were between these two limits (between 106 and 1500). In order for the datasets used in the study to be a reference for future research, care was taken to ensure that they had diversity in terms of the sample size, the count of classes and the count of features.

Based on the working principle of ANN, determining the weights requires a training phase. Additionally, a testing phase is essential to evaluate the effectiveness of the training. Therefore, datasets need to be separated into train and test sets.

In this research, datasets containing over 150 instances were allocated as 75% for training and 25% for testing. On the other side, for datasets with 150 instances or fewer, all available data was fully used in the training phase to guarantee a qualified training process. Table 2 provides comprehensive details about the datasets used.

Table 2
Comprehensive Details Regarding The Datasets Utilized

#	Dataset	Feature	Label	Instances	Training Data	Test Data
1	Aniso	2	3	1500	1125	375
2	Blobs	2	3	1500	1125	375
3	Varied	2	3	1500	1125	375
4	Aggregation	2	7	788	594	194
5	Balance	4	3	625	469	156
6	Smiley	2	4	500	375	125
7	Mouse	2	3	490	368	122
8	Ionosphere	34	2	351	264	87
9	Liver	6	3	345	259	86
10	Ecoli	7	8	336	252	84
11	Vertebral3	6	3	310	233	77
12	Pathbased	2	3	300	225	75
13	Heart	13	2	270	203	67
14	Glass	9	6	214	162	52
15	Seeds	7	3	210	158	52
16	Wine	13	3	178	133	45
17	Iris	4	3	150	150	N/A
18	Iris2D	2	3	150	150	N/A
19	Vary-density	2	3	150	150	N/A
20	Diagnosis_II	6	3	120	120	N/A
21	Appendicitis	7	2	106	106	N/A

Comparison Metrics

To evaluate and compare the performance of the algorithms used for ANN training, four distinct metrics were employed: precision, specificity, F1-score and sensitivity.

These metrics are metrics obtained from the confusion matrix. The confusion matrix is a performance measurement method frequently used in the literature to measure the accuracy of the model and the classification success of data sets. Figure 4 shows the confusion matrix.

Figure 4
Confusion Matrix

		Predicted condition	
	Population (P+N)	**Predicted Positive (PP)**	**Predicted Negative (PN)**
Actual Condition	**Positive (P)**	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Some basic concepts are essential for a clearer understanding of the metric calculation process. Positive (P) represents true positive situations in the dataset, while negative (N) represents true negative situations. True positive (TP) indicates that a condition is correctly detected, while true negative (TN) indicates that lack of a condition is correctly detected. False positive (FP) indicates that a condition is present when it is not present, and false negative (FN) indicates that a condition is present with an incorrect result that it is not present.

From a statistical perspective, specificity and sensitivity quantify the accuracy in detecting the existence or non-existence of a condition. When conditions with the presence of a condition are classified as positive and those without as negative, sensitivity measures how effectively a test identifies true positives, while specificity measures its ability to identify true negatives. Precision, on the other hand, is calculated by dividing the number of true positive cases by the total number of cases classified as positive, whether correctly or incorrectly. This metric reflects how precisely a class is identified. The F1 score combines sensitivity and precision using their harmonic mean, providing a balanced view of the results from the confusion matrix (Karakoyun, 2024; Tharwat, 2021). The equations for calculating these metrics depend on the confusion matrix and the aforementioned definitions are provided below:

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$F1\ Score = \frac{2TP}{2TP + FP + FN} \tag{5}$$

Results

To facilitate the comparison of the experimental outcomes, we computed the precision, specificity, F1-score and sensitivity for each algorithms depend on the mean worths achieved after conducting 25 runs. Every metric is presented with its mean and standard deviation derived from the algorithms' results over these 25 runs. Additionally, the success ranking was established using the mean worth of the 25 runs and included in the result tables. The performance metrics for precision, specificity, F1-score and sensitivity for the algorithms are displayed in Table 3, Table 4, Table 5, and Table 6, respectively. In the tables, A represents the mean, S represents the standard deviation, and R represents the success ranking.

Table 3

Experimental Results Of The Algorithms For Sensitivity Metric

	GOA			AHA			AOA			COA			ABC			TSA		
Datasets	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**
Aniso	0.86	0.16	3	0.87	0.11	2	0.89	0.14	1	0.86	0.14	4	0.78	0.15	6	0.8	0.13	5
Blobs	0.91	0.15	5	0.95	0.11	2	0.96	0.09	1	0.9	0.17	6	0.92	0.12	4	0.92	0.11	3
Varied	0.87	0.13	1	0.86	0.08	2	0.82	0.11	4	0.84	0.1	3	0.76	0.11	6	0.79	0.09	5
Aggregation	0.23	0.07	2	0.23	0.06	1	0.23	0.07	3	0.21	0.06	4	0.18	0.06	6	0.19	0.06	5
Balance	0.59	0.07	1	0.5	0.08	5	0.57	0.09	2	0.51	0.08	4	0.47	0.1	6	0.55	0.06	3
Smiley	0.26	0.04	1	0.25	0	3	0.25	0	3	0.25	0	3	0.25	0.01	2	0.25	0	4
Mouse	0.73	0.17	1	0.55	0.16	5	0.65	0.16	2	0.48	0.18	6	0.61	0.13	3	0.59	0.12	4
Ionosphere	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1
Liver	0.35	0.04	2	0.35	0.03	4	0.34	0.03	5	0.35	0.03	3	0.34	0.03	6	0.35	0.03	1
Ecoli	0.21	0.06	1	0.16	0.04	3	0.19	0.05	2	0.15	0.05	4	0.12	0.06	6	0.14	0.05	5
Vertebral3	0.55	0.13	3	0.57	0.1	2	0.61	0.05	1	0.5	0.14	5	0.49	0.11	6	0.55	0.12	4
Pathbased	0.47	0.12	1	0.43	0.17	4	0.46	0.13	2	0.37	0.18	5	0.45	0.13	3	0.36	0.17	6
Heart	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1
Glass	0.19	0.06	1	0.18	0.04	3	0.18	0.04	4	0.19	0.04	2	0.17	0.05	6	0.17	0.03	5
Seeds	0.6	0.13	1	0.54	0.08	3	0.59	0.08	2	0.47	0.12	6	0.48	0.13	5	0.51	0.09	4
Wine	0.47	0.15	5	0.56	0.15	1	0.51	0.14	4	0.54	0.14	3	0.38	0.11	6	0.56	0.14	2
Iris	0.72	0.2	2	0.73	0.1	1	0.72	0.11	3	0.62	0.15	6	0.63	0.15	5	0.7	0.1	4
Iris2D	0.79	0.17	1	0.7	0.07	2	0.69	0.11	3	0.62	0.12	6	0.64	0.1	5	0.65	0.03	4
Vary-density	0.83	0.15	1	0.69	0.06	6	0.76	0.14	2	0.7	0.15	5	0.7	0.13	4	0.7	0.1	3
Diagnosis_II	0.65	0.05	1	0.55	0.08	5	0.61	0.08	2	0.58	0.1	3	0.45	0.11	6	0.57	0.08	4
Appendicitis	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1	0.5	0	1
Avg. Rank	**1.71**			2.71			2.33			3.86			4.48			3.52		

Table 3 shows that GOA achieves the best results in 14 out of 21 datasets in the precision metric and ranks first in average ranking success with a score of 1.71. The algorithm that closely follows GOA in average ranking success is AOA. AOA ranks first in 6 out of 21 datasets with a score of 2.33. However, it shares the first place with GOA in 3 out of these 6 datasets.

Table 4

Experimental Results Of The Algorithms For Specificity Metric

	GOA			AHA			AOA			COA			ABC			TSA		
Datasets	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**
Aniso	0.93	0.08	3	0.94	0.06	2	0.94	0.07	1	0.93	0.07	4	0.89	0.07	6	0.90	0.06	5
Blobs	0.95	0.08	5	0.97	0.05	2	0.98	0.05	1	0.95	0.08	6	0.96	0.06	4	0.96	0.06	3
Varied	0.94	0.06	1	0.93	0.04	2	0.91	0.05	4	0.92	0.05	3	0.88	0.05	6	0.89	0.05	5
Aggregation	0.88	0.02	3	0.89	0.02	1	0.88	0.02	2	0.88	0.02	4	0.87	0.02	6	0.87	0.02	5
Balance	0.88	0.06	1	0.81	0.07	5	0.87	0.07	2	0.82	0.07	4	0.78	0.08	6	0.86	0.05	3
Smiley	0.75	0.02	1	0.75	0	3	0.75	0	3	0.75	0	3	0.75	0.00	2	0.75	0.00	4
Mouse	0.87	0.08	1	0.80	0.08	4	0.83	0.08	2	0.74	0.09	6	0.82	0.07	3	0.80	0.06	5
Ionosphere	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1
Liver	0.68	0.04	2	0.68	0.03	4	0.68	0.03	5	0.68	0.03	3	0.67	0.03	6	0.69	0.03	1
Ecoli	0.91	0.02	1	0.89	0.02	3	0.90	0.02	2	0.88	0.02	5	0.88	0.02	6	0.89	0.02	4
Vertebral3	0.81	0.09	4	0.83	0.07	2	0.85	0.04	1	0.78	0.10	5	0.77	0.08	6	0.82	0.08	3
Pathbased	0.73	0.06	1	0.71	0.08	4	0.72	0.07	2	0.68	0.09	5	0.72	0.06	3	0.67	0.09	6
Heart	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1
Glass	0.84	0.01	1	0.84	0.01	3	0.84	0.01	4	0.84	0.01	2	0.83	0.01	6	0.84	0.01	5
Seeds	0.80	0.06	1	0.77	0.04	3	0.80	0.04	2	0.74	0.06	6	0.74	0.07	5	0.75	0.05	4
Wine	0.75	0.08	5	0.79	0.08	2	0.77	0.08	4	0.78	0.07	3	0.69	0.06	6	0.79	0.08	1
Iris	0.86	0.10	2	0.87	0.05	1	0.86	0.06	3	0.81	0.07	6	0.82	0.07	5	0.85	0.05	4
Iris2D	0.90	0.09	1	0.85	0.04	2	0.85	0.05	3	0.81	0.06	6	0.82	0.05	5	0.82	0.02	4
Vary-density	0.91	0.07	1	0.84	0.03	6	0.88	0.07	2	0.85	0.08	5	0.85	0.06	4	0.85	0.05	3
Diagnosis_II	0.98	0.05	1	0.88	0.08	5	0.94	0.07	2	0.91	0.10	3	0.79	0.11	6	0.90	0.08	4
Appendicitis	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1	0.50	0	1
Avg. Rank		**1.81**			2.71			2.29			3.90			4.48			3.43	

In Table 4, which details the results of the specificity metric, GOA again stands out and achieved the best results in 14 out of 21 datasets with an average score of 1.81. AOA ranked second again in this metric with an average score of 2.29.

Table 5

Experimental Results Of The Algorithms For Precision Metric

	GOA			AHA			AOA			COA			ABC			TSA		
Datasets	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**
Aniso	0.79	0.25	4	0.88	0.13	1	0.87	0.20	2	0.87	0.16	3	0.75	0.23	6	0.78	0.20	5
Blobs	0.86	0.23	6	0.93	0.16	2	0.95	0.14	1	0.86	0.24	5	0.91	0.16	4	0.91	0.16	3
Varied	0.83	0.21	2	0.90	0.04	1	0.82	0.17	4	0.83	0.16	3	0.75	0.19	6	0.78	0.17	5
Aggregation	0.15	0.08	2	0.14	0.07	3	0.16	0.06	1	0.13	0.08	4	0.08	0.07	6	0.10	0.07	5

Balance	0.57	0.12	1	0.52	0.09	4	0.54	0.09	2	0.49	0.11	5	0.44	0.14	6	0.54	0.06	3
Smiley	0.20	0.03	1	0.19	5.67E-17	3	0.19	5.67E-17	3	0.19	5.67E-17	3	0.20	0.03	2	0.19	9.35E-05	4
Mouse	0.65	0.23	1	0.46	0.17	5	0.56	0.21	2	0.36	0.21	6	0.55	0.18	3	0.55	0.16	4
Ionosphere	0.32	1.70E-16	1	0.32	1.70E-16	1	0.32	1.70E-16	1	0.32	1.70E-16	1	0.32	1.70E-16	1	0.32	1.70E-16	1
Liver	0.32	0.10	2	0.31	0.11	3	0.29	0.10	6	0.29	0.12	5	0.30	0.11	4	0.35	0.08	1
Ecoli	0.13	0.07	1	0.10	0.04	3	0.11	0.06	2	0.10	0.06	5	0.06	0.05	6	0.10	0.08	4
Vertebral3	0.42	0.17	4	0.45	0.14	2	0.53	0.10	1	0.40	0.18	6	0.40	0.16	5	0.45	0.17	3
Pathbased	0.39	0.17	3	0.39	0.23	2	0.41	0.20	1	0.34	0.23	5	0.36	0.13	4	0.25	0.14	6
Heart	0.22	1.42E-16	1	0.22	1.42E-16	1	0.22	1.42E-16	1	0.22	1.42E-16	1	0.22	1.42E-16	1	0.22	1.42E-16	1
Glass	0.10	0.07	2	0.09	0.06	3	0.07	0.04	4	0.11	0.07	1	0.06	0.06	6	0.07	0.04	5
Seeds	0.50	0.23	1	0.44	0.11	3	0.50	0.16	2	0.39	0.16	5	0.34	0.20	6	0.41	0.14	4
Wine	0.31	0.19	5	0.45	0.18	2	0.37	0.19	4	0.43	0.19	3	0.19	0.14	6	0.47	0.15	1
Iris	0.62	0.29	2	0.67	0.19	1	0.59	0.19	3	0.49	0.17	6	0.53	0.20	5	0.56	0.18	4
Iris2D	0.71	0.26	1	0.58	0.16	3	0.60	0.20	2	0.53	0.19	4	0.52	0.17	5	0.50	0.07	6
Vary-density	0.76	0.24	1	0.61	0.17	4	0.67	0.24	2	0.61	0.23	5	0.62	0.20	3	0.59	0.19	6
Diagnosis_II	0.66	0.03	1	0.57	0.10	5	0.62	0.09	2	0.57	0.15	4	0.43	0.17	6	0.60	0.04	3
Appendicitis	0.40	0	1	0.40	0	1	0.40	0	1	0.40	0	1	0.40	0	1	0.40	0	1
Avg. Rank		**2.05**			2.52			2.24			3.86			4.38			3.57	

According to the data presented in Table 5, concerning the sensitivity metric, GOA achieved the highest success in 11 out of the 21 data sets and ranks first in average success ranking with a score of 2.05. AOA, coming in second for average success ranking, has a score of 2.24.

Table 6

Experimental Results Of The Algorithms For F1-Score Metric

	GOA			AHA			AOA			COA			ABC			TSA		
Datasets	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**	**A**	**S**	**R**
Aniso	0.82	0.22	4	0.86	0.14	1	0.86	0.19	2	0.83	0.18	3	0.73	0.20	6	0.76	0.18	5
Blobs	0.88	0.20	5	0.93	0.14	2	0.95	0.13	1	0.87	0.22	6	0.90	0.16	4	0.91	0.15	3
Varied	0.84	0.18	2	0.85	0.10	1	0.79	0.16	4	0.81	0.14	3	0.71	0.15	6	0.75	0.14	5
Aggregation	0.15	0.07	3	0.16	0.07	2	0.16	0.07	1	0.14	0.07	4	0.09	0.06	6	0.11	0.07	5
Balance	0.56	0.10	1	0.47	0.11	5	0.54	0.10	2	0.47	0.11	4	0.41	0.14	6	0.53	0.07	3
Smiley	0.22	0.04	1	0.22	8.50E-17	3	0.22	8.50E-17	3	0.22	8.50E-17	3	0.22	0.01	2	0.22	2.57E-04	4
Mouse	0.68	0.21	1	0.49	0.17	5	0.59	0.19	2	0.40	0.20	6	0.56	0.14	3	0.54	0.13	4
Ionosphere	0.39	5.67E-17	1	0.39	5.67E-17	1	0.39	5.67E-17	1	0.39	5.67E-17	1	0.39	5.67E-17	1	0.39	5.67E-17	1
Liver	0.30	0.07	2	0.29	0.05	3	0.29	0.06	4	0.28	0.06	5	0.28	0.05	6	0.31	0.05	1

Ecoli	0.15	0.06	1	0.12	0.04	3	0.14	0.06	2	0.10	0.05	4	0.07	0.05	6	0.10	0.05	5
Vertebral3	0.47	0.15	4	0.49	0.11	2	0.55	0.06	1	0.42	0.17	5	0.41	0.13	6	0.48	0.15	3
Pathbased	0.38	0.10	1	0.34	0.16	4	0.37	0.12	2	0.30	0.16	5	0.34	0.12	3	0.27	0.14	6
Heart	0.31	1.70E-16	1	0.31	1.70E-16	1	0.31	1.70E-16	1	0.31	1.70E-16	1	0.31	1.70E-16	1	0.31	1.70E-16	1
Glass	0.12	0.06	2	0.11	0.05	3	0.10	0.04	4	0.12	0.05	1	0.07	0.06	6	0.09	0.04	5
Seeds	0.51	0.18	2	0.45	0.10	3	0.51	0.12	1	0.37	0.15	5	0.36	0.16	6	0.41	0.11	4
Wine	0.36	0.18	5	0.47	0.18	1	0.41	0.17	4	0.45	0.17	3	0.24	0.13	6	0.46	0.17	2
Iris	0.65	0.26	2	0.67	0.15	1	0.64	0.16	3	0.52	0.17	6	0.54	0.16	5	0.61	0.14	4
Iris2D	0.74	0.23	1	0.60	0.11	3	0.60	0.14	2	0.53	0.14	6	0.54	0.13	5	0.54	0.02	4
Vary-density	0.78	0.20	1	0.60	0.09	6	0.69	0.19	2	0.60	0.20	5	0.62	0.16	3	0.61	0.14	4
Diagnosis_II	0.65	0.05	1	0.54	0.10	5	0.60	0.09	2	0.57	0.13	3	0.42	0.14	6	0.56	0.08	4
Appendicitis	0.45	0	1	0.45	0	1	0.45	0	1	0.45	0	1	0.45	0	1	0.45	0	1
Avg. Rank		**2.00**			2.67			2.14			3.81			4.48			3.52	

Lastly, Table 6 shows that GOA achieved the best results in 11 out of 21 datasets in the f1-score metric and ranked first in average ranking success with a score of 2.00. The algorithm that closely follows GOA in average ranking success is AOA. AOA ranked first in 7 out of 21 datasets with a score of 2.14.

When we analyze the average results and rankings across precision, specificity, F1-score and sensitivity metrics, GOA consistently emerges as the best performing algorithm. GOA proves its effectiveness by achieving the best results on 14 out of 21 datasets in sensitivity and specificity metrics, with average ranking scores of 1.71 and 1.81, respectively. In precision, GOA leads on 11 datasets, maintaining an average ranking of 2.05. Similarly, in f1-score metric, GOA stands out with the best results on 11 datasets and an average ranking score of 2.00. AOA follows GOA closely. It achieves the best rankings on 6 datasets for sensitivity and specificity, 7 datasets for precision and f1-score, and in some cases, it shares the first place with GOA. These findings highlight the consistent performance superiority of GOA in training ANNs across multiple evaluation metrics.

Discussion and Conclusion

In this article, the performances of six different meta-heuristic algorithms for training artificial neural networks were compared. ANN training was performed on 21 different classification datasets using Grasshopper Optimization Algorithm, Artificial Hummingbird Algorithm, Arithmetic Optimization Algorithm, Crayfish Optimization Algorithm, Artificial Bee Colony and Tree-Seed Algorithm. The performances of the algorithms were evaluated using four basic metrics such as precision, specificity, F1-score and sensitivity. The experimental results obtained show that GOA in particular provides high success compared to other algorithms for ANN training. GOA achieved the best results in 14 out of 21 datasets and ranked first in the average success ranking. Although AOA and other algorithms also achieved good results, GOA was the algorithm with the highest overall success. These results show that metaheuristic algorithms provide an effective solution to solve complex weight update processes in ANN training.

It was observed that as the number of features of the datasets used in the study increases, the problem size also increases significantly. From this perspective, it is seen that metaheuristic algorithms, especially GOA, are successful on large-scale continuous problems. Comparisons between the algorithms were examined in detail in terms of both average metric values and success rankings. As a result of the comparisons made, it is seen that metaheuristic algorithms have significant potential for ANN training and provide effective solutions for such complex problems.

In further studies, it is recommended that these algorithms can be tested on different problems and spread to a wider application area. Additionally, developing hybrid approaches by combining different meta-heuristic algorithms may be an important research topic for future studies.

References

Abdullahi Mu'azu, M. (2023). Hybridized artificial neural network with metaheuristic algorithms for bearing capacity prediction. *Ain Shams Engineering Journal*, *14*(5), 101980. https://doi.org/10.1016/j.asej.2022.101980

Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113609. https://doi.org/10.1016/j.cma.2020.113609

Agrawal, P. (2021, February 2). *Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)*. https://ieeexplore.ieee.org/abstract/document/9344597

Akel, V., & Karacameydan, F. (2012). *Yatırım Fonları Net Varlık Değerlerinin Yapay Sinir Ağları Yöntemiyle Tahmin Edilmesi*. https://earsiv.anadolu.edu.tr/xmlui/handle/11421/161

Aksu, İ. Ö., Esenboğa, B., Yavuzdeğer, A., Nazligül, H., TekiN, P., & DemiRdelen, T. (2022). *YSA Tabanlı Metasezgisel Yöntemlerle Kısa Vadeli Solar Güç Tahmini*.

Aslanov, T., Çavdar, B., Öztürk, Ö., Başlık, Ş., & Akyazi, Ö. (2023, September 22). *İki Alanlı Termal Güç Sisteminde AHA Tabanlı PID, FOPID ve FOPI-FOPD Denetleyicilerin Siber Saldırı Analizi Cyber Attack Analysis of AHA Based PID, FOPID and FOPI- FOPD Controllers in Two Area Thermal Power System*.

Ateş, K. T. (2022). Çok Katmanlı Yapay Sinir Ağı Modeli ve Kültürel Algoritma Modeli Kullanılarak Geliştirilen Melez Yöntem ile Kısa Vadeli Fotovoltaik Enerji Santrali Çıkış Gücü Tahmini. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *5*(1), Article 1. https://doi.org/10.47495/okufbed.1028813

Atlı, F. D. (2022). *Yapay Sinir Ağlarının Eğitimi Için Salp Sürü Optimizasyonu Algoritmasının İYileştirilmesi* [M.Sc., Necmettin Erbakan University (Turkey)]. https://www.proquest.com/docview/2784407806/abstract/D25DEFE4EE024945PQ/1

Ayaz, H. I., & Kamisli Ozturk, Z. (2021). A mathematical model and a heuristic approach for train seat scheduling to minimize dwell time. *Computers & Industrial Engineering*, *160*, 107590. https://doi.org/10.1016/j.cie.2021.107590

Bakır, H. (2024). A novel artificial hummingbird algorithm improved by natural survivor method. *Neural Computing and Applications*, *36*(27), 16873–16897. https://doi.org/10.1007/s00521-024-09928-z

Çalışkan, M. M. T., & Deniz, D. (2015). Yapay Sinir Ağlarıyla Hisse Senedi Fiyatları ve Yönlerinin Tahmini. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, *10*(3), Article 3.

Çelik, Y. (2013). *Optimizasyon problemlerinde bal arıları evlilik optımızasyonu algoritmasının (marriage in honey bee optimization-MBO) performansının geliştirilmesi* [Doktora, Selçuk Üniversitesi]. https://tez.yok.gov.tr/UlusalTezMerkezi/tezDetay.jsp?id=oDMHpvC1u3eqlJ8Sf0Fwrg&no=6_0T-

032nAUXAYvxRNfRNQ

Çınaroğlu, E., & Avcı, T. (2020). THY Hisse Senedi Değerinin Yapay Sinir Ağları İle Tahmini. *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi*, *34*(1), Article 1. https://doi.org/10.16951/atauniiibd.530322

Çörekcioğlu, M., Ercan, E., & Elibüyük, S. A. (2021). Yapay Sinir Ağı Yöntemlerinin Tekstil Sektöründe Kullanım Uygulamaları. *Teknik Bilimler Dergisi*, *11*(2), Article 2. https://doi.org/10.35354/tbed.884531

Dhal, K. G., Sasmal, B., Das, A., Ray, S., & Rai, R. (2023). A Comprehensive Survey on Arithmetic Optimization Algorithm. *Archives of Computational Methods in Engineering*, *30*(5), 3379–3404. https://doi.org/10.1007/s11831-023-09902-3

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., & Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering*, *137*, 106040. https://doi.org/10.1016/j.cie.2019.106040

Emambocus, B. A. S., Jasser, M. B., & Amphawan, A. (2023). A Survey on the Optimization of Artificial Neural Networks Using Swarm Intelligence Algorithms. *IEEE Access*. https://ieeexplore.ieee.org/abstract/document/10004960

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, *76*(2), 60–68. https://doi.org/10.1177/003754970107600201

Gölcük, İ., Ozsoydan, F. B., & Durmaz, E. D. (2023). An improved arithmetic optimization algorithm for training feedforward neural networks under dynamic environments. *Knowledge-Based Systems*, *263*, 110274. https://doi.org/10.1016/j.knosys.2023.110274

Holland, J. H. (1992). Genetic Algorithms. *Scientific American*, *267*(1), 66–73.

Jama, B. S. A. (2021). *Modifiye edilmiş karınca aslanı optimizasyon algoritması kullanılarak bölge büyütme yöntemi ile gri seviye görüntü segmentasyonu* [Master Thesis, Konya Teknik Üniversitesi]. https://gcris.ktun.edu.tr/handle/20.500.13091/2250

Jia, H., Rao, H., Wen, C., & Mirjalili, S. (2023). Crayfish optimization algorithm. *Artificial Intelligence Review*, *56*(2), 1919–1979. https://doi.org/10.1007/s10462-023-10567-4

Jia, H., Zhou, X., Zhang, J., Abualigah, L., Yildiz, A. R., & Hussien, A. G. (2024). Modified crayfish optimization algorithm for solving multiple engineering application problems. *Artificial Intelligence Review*, *57*(5), 127. https://doi.org/10.1007/s10462-024-10738-x

Jiadong, Q., Ohl, J. P., & Tran, T.-T. (2024). Predicting clay compressibility for foundation design with high reliability and safety: A geotechnical engineering perspective using artificial neural network and five metaheuristic algorithms. *Reliability Engineering & System Safety*, *243*, 109827. https://doi.org/10.1016/j.ress.2023.109827

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report-Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department*.

Karaboga, D., & Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, *214*(1), 108–132. https://doi.org/10.1016/j.amc.2009.03.090

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global*

Optimization, *39*(3), 459–471. https://doi.org/10.1007/s10898-007-9149-x

Karakoyun, M. (2024). Artificial neural network training using a multi selection artificial algae algorithm. *Engineering Science and Technology, an International Journal*, *53*, 101684. https://doi.org/10.1016/j.jestch.2024.101684

Kashan, A. H. (2009). League Championship Algorithm: A New Algorithm for Numerical Function Optimization. *2009 International Conference of Soft Computing and Pattern Recognition*, 43–48. https://doi.org/10.1109/SoCPaR.2009.21

Kaytan, M., Yeroğlu, C., & Aydilek, İ. B. (2020). Yapay Sinir Ağları Eğitiminde Kullanılan Optimizasyon Yöntemlerinin İncelenmesi ve Kan Nakli Hizmet Merkezi Veri Seti Üzerinden Değerlendirilmesi. *Computer Science*, *5*(2), Article 2.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *4*, 1942–1948 vol.4. https://doi.org/10.1109/ICNN.1995.488968

Keskenler, M. F., & Keskenler, E. F. (2017). *Geçmişten Günümüze Yapay Sinir Ağları ve Tarihçesi*.

Khodadadi, N., Mirjalili, S. M., Zhao, W., Zhang, Z., Wang, L., & Mirjalili, S. (2023). Multi-Objective Artificial Hummingbird Algorithm. In A. Biswas, C. B. Kalayci, & S. Mirjalili (Eds.), *Advances in Swarm Intelligence: Variations and Adaptations for Optimization Problems* (pp. 407–419). Springer International Publishing. https://doi.org/10.1007/978-3-031-09835-2_22

Kiran, M. S. (2015). TSA: Tree-seed algorithm for continuous optimization. *Expert Systems with Applications*, *42*(19), 6686–6698. https://doi.org/10.1016/j.eswa.2015.04.055

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

Köprü, E. Y. (2020). *Yapay Sinir Ağları İle Sıvı Ham Demir Tamini Ve 5. Yüksek Fırın Uygulaması* [Thesis]. http://acikerisim.karabuk.edu.tr:8080/xmlui/handle/123456789/962

Meraihi, Y., Gabis, A. B., Mirjalili, S., & Ramdane-Cherif, A. (2021). Grasshopper Optimization Algorithm: Theory, Variants, and Applications. *IEEE Access*, *9*, 50001–50024. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3067597

Mohamed, A. W., Hadi, A. A., & Mohamed, A. K. (2020). Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *International Journal of Machine Learning and Cybernetics*, *11*(7), 1501–1529. https://doi.org/10.1007/s13042-019-01053-x

Özdemir, A., & Pamuk, N. (2021). Kısmi Gölgelenme Şartları Altındaki Kompleks Yapılı Fotovoltaik Enerji Sistemlerinde Maksimum Güç Noktası Takibinin Metasezgisel Algoritmalar Kullanılarak İncelenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, *31*, Article 31. https://doi.org/10.31590/ejosat.1006248

Özdemir, F. N., & Özkış, A. (2024). Kar Erime Optimizasyonu Algoritması ile Çok Katmanlı Yapay Sinir Ağının Eğitimi. *Çukurova Üniversitesi Mühendislik Fakültesi Dergisi*, *39*(2), Article 2. https://doi.org/10.21605/cukurovaumfd.1514409

Özkan, F. (2012). Döviz Kuru Tahmininde Parasal Model ve Yapay Sinir Ağları Karşılaştırması. *Business and Economics Research Journal*.

Özkaya, S., Conker, Ç., & Bilgiç, H. H. (2021). Esnek Robot Kol Sistemi İçin Lqr Denetleyici Parametrelerinin Metasezgisel Algoritmalar Kullanılarak Belirlenmesi. *Konya Journal of Engineering Sciences*, *9*(3), Article 3. https://doi.org/10.36306/konjes.896087

Özmen, T., Kuzu, Ü., Koçyiğit, Y., & Sarnel, H. (2023). Metasezgisel yöntemlerle öznitelik sayısını azaltarak diyabetin erken dönemde tespiti. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *29*(6), Article 6.

Qaddoura, R., Faris, H., Aljarah, I., & Castillo, P. A. (2020). EvoCluster: An Open-Source Nature-Inspired Optimization Clustering Framework in Python. In P. A. Castillo, J. L. Jiménez Laredo, & F. Fernández de Vega (Eds.), *Applications of Evolutionary Computation* (pp. 20–36). Springer International Publishing. https://doi.org/10.1007/978-3-030-43722-0_2

Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2012). Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems. *Information Sciences*, *183*(1), 1–15. https://doi.org/10.1016/j.ins.2011.08.006

Rençber, Ö. F. (2018). *Basamak Korelasyon, Kohonen ve ANFIS Yapay Sinir Ağ Modellerinin Sınıflandırma Performanslarının Karşılaştırılması: Lojistik Performans Endeksi Üzerine Uygulama*.

Rere, L. M. R., Fanany, M. I., & Arymurthy, A. M. (2016). Metaheuristic Algorithms for Convolution Neural Network. *Computational Intelligence and Neuroscience*, *2016*(1), 1537325. https://doi.org/10.1155/2016/1537325

Sahman, M. A., Cinar, A. C., Saritas, I., & Yasar, A. (2019). Tree-seed algorithm in solving real-life optimization problems. *IOP Conference Series: Materials Science and Engineering*, *675*(1), 012030. https://doi.org/10.1088/1757-899X/675/1/012030

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper Optimisation Algorithm: Theory and application. *Advances in Engineering Software*, *105*, 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004

Shahvaroughi Farahani, M., & Razavi Hajiagha, S. H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. *Soft Computing*, *25*(13), 8483–8513. https://doi.org/10.1007/s00500-021-05775-5

Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics*, *17*(1), 168–192. https://doi.org/10.1016/j.aci.2018.08.003

Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, *40*(14), 5501–5506. https://doi.org/10.1016/j.eswa.2013.04.013

Yildiz, B. (2023). Yeni Bir Hibrid Metasezgisel Algoritma İle Drone Kolunun Yapısal Optimizasyonu. *Makina Tasarım ve İmalat Dergisi*, *21*(2), Article 2. https://doi.org/10.56193/matim.1302774

Zaimoğlu, E. A. (2023). *Büyük boyutlu veriler için metasezgisel yöntemler ile öznitelik indirgemede yeni bir yaklaşım geliştirilmesi = Developing a new approach to feature selection with metaheuristic methods for large scale data* [doctoralThesis, Sakarya Üniversitesi]. https://acikerisim.sakarya.edu.tr/handle/20.500.12619/101468

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, 114194. https://doi.org/10.1016/j.cma.2021.114194

About the Authors

Münevver BAHAR has a bachelor's degree in Computer Engineering from Necmettin Erbakan University. She is currently doing her master's degree at Necmettin Erbakan University, Department of Computer Engineering. She works in the fields of Artificial Intelligence, Machine Learning and Optimization.
E-mail: munevverbahar1@gmail.com, **ORCID:** 0009-0009-0489-0884

Halil İbrahim AYAZ, PhD, He earned a BS and an MS degree in Industrial Engineering from Hacettepe University. He earned a PhD degree in Industrial Engineering from Eskişehir Technical University. His research interests include machine learning, data mining and analysis, scheduling, and optimization. He has two journal articles and seven studies presented at international meetings. Currently, he works as an assistant professor doctor in the Industrial Engineering department at Necmettin Erbakan University.
E-mail: hiayaz@erbakan.edu.tr, **ORCID:** 0000-0001-5547-6485

Murat KARAKOYUN, received the master and Ph.D. degrees from the Department of Computer Engineering of Selçuk University and Konya Technical University, respectively. He is working as an Assistant Professor at Computer Engineering Department of Necmettin Erbakan University. Areas of interest: artificial intelligence, image processing, data mining, machine learning.
E-mail: mkarakoyun@erbakan.edu.tr, **ORCID:** 0000-0002-0677-9313

Similarity Index
The similarity index obtained from the plagiarism software for this book chapter is 19%.