

Article

AOSMA-MLP: A Novel Method for Hybrid Metaheuristics Artificial Neural Networks and a New Approach for Prediction of Geothermal Reservoir Temperature

Ezgi Gurgenc ^{1,*} , Osman Altay ²  and Elif Varol Altay ² ¹ Department of Mechanical Engineering, Firat University, Elazig 23119, Turkey² Department of Software Engineering, Manisa Celal Bayar University, Manisa 45400, Turkey; osman.altay@cbu.edu.tr (O.A.); elif.altay@cbu.edu.tr (E.V.A.)

* Correspondence: egurgenc@firat.edu.tr

Featured Application: The proposed models can help uncover the usage areas of geothermal waters by determining the reservoir temperatures in advance. Thus, they can be used as a decision support system to make the most appropriate selection.

Abstract: To ascertain the optimal and most efficient reservoir temperature of a geothermal source, long-term field studies and analyses utilizing specialized devices are essential. Although these requirements increase project costs and induce delays, utilizing machine learning techniques based on hydrogeochemical data can minimize losses by accurately predicting reservoir temperatures. In recent years, applying hybrid methods to real-world challenges has become increasingly prevalent over traditional machine learning methodologies. This study introduces a novel machine learning approach, named AOSMA-MLP, integrating the adaptive opposition slime mould algorithm (AOSMA) and multilayer perceptron (MLP) techniques, specifically designed for predicting the reservoir temperature of geothermal resources. Additionally, this work compares the basic artificial neural network and widely recognized algorithms in the literature, such as the whale optimization algorithm, ant lion algorithm, and SMA, under equal conditions using various evaluation regression metrics. The results demonstrated that AOSMA-MLP outperforms basic MLP and other metaheuristic-based MLPs, with the AOSMA-trained MLP achieving the highest performance, indicated by an R^2 value of 0.8514. The proposed AOSMA-MLP approach shows significant potential for yielding effective outcomes in various regression problems.

Keywords: hydrogeochemistry; reservoir temperature; geothermal energy; adaptive opposition slime mould multilayer perceptron; artificial neural network; metaheuristic optimization



Citation: Gurgenc, E.; Altay, O.; Altay, E.V. AOSMA-MLP: A Novel Method for Hybrid Metaheuristics Artificial Neural Networks and a New Approach for Prediction of Geothermal Reservoir Temperature. *Appl. Sci.* **2024**, *14*, 3534. <https://doi.org/10.3390/app14083534>

Academic Editors: Jan Blachowski, Jörg Benndorf, Damian Kasza and Isabel Fernandes

Received: 14 March 2024

Revised: 10 April 2024

Accepted: 20 April 2024

Published: 22 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, driven by increased urbanization and improvements in living standards, the energy consumed in homes and living spaces is on the rise. Buildings are now responsible for about 40% of energy consumption and nearly 36% of greenhouse gas emissions in the European Union (EU). A significant portion of this energy is used for heating and cooling [1,2]. Consequently, there has been a notable increase in the demand for fossil fuels, raising concerns about energy supply security, environmental pollution, and greenhouse gas emissions in recent years [3]. These concerns can be alleviated by employing renewable energy sources with low carbon emissions, such as biomass, solar, and geothermal energy [2,4,5]. Among these energy sources, geothermal energy, independent of weather conditions, has significant potential. Due to its positive attributes, geothermal energy has become increasingly important for heating, cooling, and other energy applications [2,6–9]. Moreover, geothermal energy is recognized as a renewable energy source crucial for eco-

conomic development, reducing environmental pollution and greenhouse gases, and ensuring energy security [3,10,11].

Geothermal energy, stored beneath the Earth's crust and at its depths, originates from the natural internal heat of the planet. It has various applications, ranging from direct use to electricity production, depending on the reservoir temperature (RT). These applications include the heating and cooling of buildings, electricity generation in power plants, agricultural applications, and uses in balneology. While some applications require low- to medium-temperature fluids, others necessitate medium- to high-temperature fluids for heating, cooling, and electric power generation [3,12]. As such, the value of RTs is crucial for determining the usability of geothermal water [12].

Hydrogeochemical analyses of geothermal waters provide insights into their formation mechanisms, enabling predictions about RTs [13]. Extensive research is vital prior to drilling in geothermal resource areas to determine the most suitable locations, which can help reduce the costs associated with geothermal drilling and enhance the utilization of geothermal resources. Identifying RTs of geothermal waters necessitates complex geological surveys and analyses, which are time-consuming, costly, and complex [14–22].

Machine learning techniques, successfully applied in various fields, offer a promising approach to address these challenges by predicting RTs. RTs have been predicted using methods like linear regression, linear support vector machine, and deep neural networks [12]. Additionally, there are studies on predicting geothermal heat flow and deep RTs using methods like gradient-assisted R-regression trees and artificial neural networks (ANNs) [23,24].

ANNs are extensively used in regression problems, with various models such as feed-forward networks (FNNs) [25] and radial basic function (RBF) [26] networks found in the literature. The multilayer perceptron (MLP), a type of FNN, is particularly popular [27]. While classic optimization algorithms are commonly used, they may encounter issues like becoming stuck at local minima, premature convergence, and suboptimal performance. To overcome these issues, using metaheuristic optimization algorithms during the MLP training phase is suggested [28–30].

Among the proposed methods, whale optimization (WOA-MLP) [31], the ant lion optimizer (ALO-MLP) [32], and the slime mould algorithm (SMA-MLP) [33] are widely cited in the literature. However, because SMA has limitations in the exploration and exploitation phase due to its reliance on two random search agents, the adaptive opposition SMA (AOSMA) was introduced to mitigate these disadvantages [34].

This study seeks to accurately estimate the reservoir temperatures (RTs) of geothermal resources by leveraging hydrogeochemical data through a novel methodology. The primary goal is to mitigate the financial, temporal, and labor-intensive demands traditionally associated with geothermal projects. This is achieved by devising a new expert system designed to supplant the necessity for protracted fieldwork and the reliance on specialized personnel and equipment. Central to this research is the innovative application of the adaptive opposition-based slime mould algorithm (AOSMA) during the multilayer perceptron (MLP) training phase—a first in this context. This newly proposed method aims to adeptly predict RTs. Furthermore, it undergoes a comparative analysis against the one-step secant backpropagation artificial neural network (ANN), a first-order optimization technique, as well as other methodologies trained with distinct metaheuristic optimization algorithms, including whale optimization algorithm-MLP (WOA-MLP), ant lion optimizer-MLP (ALO-MLP), and slime mould algorithm-MLP (SMA-MLP).

2. Material and Methods

2.1. Data Acquisition

In this study, we utilized a dataset comprising 161 data points, previously assembled by researchers within our team. This dataset, initially used to predict the application areas of reservoir temperatures (RTs) through classification approaches, is the foundation for our current investigation [17]. Herein, we propose a hybrid machine learning approach, the

adaptive opposition-based slime mould algorithm–multilayer perceptron (AOSMA-MLP), applied to tackle this real-world problem by predicting the numerical values of RTs—a task commonly addressed with regression techniques in machine learning literature.

To facilitate a comprehensive understanding of the present study, it is pertinent to briefly revisit some critical aspects of the dataset, detailed in our prior research [17]. Figure 1 illustrates the modeling’s input and output parameters, encompassing seven inputs and one output. The selection of ions as input parameters is primarily influenced by their availability in the fluid, given that geothermal fluids, deriving their components from either a degassing magma heat source or the surrounding rock, often contain high concentrations of these ions. Notably, the Na⁺ and K⁺ ion concentrations are essential for understanding water–rock interactions at elevated temperatures. Moreover, an increased B content typically signifies deep feeding and a high-temperature deep reservoir. Cl[−] is a major anion that enhances the salinity of geothermal fluids. The silica concentration, affected by temperature, is another crucial parameter. In a geothermal system, SiO₂’s solubility decreases with falling temperature. Additionally, silica is vital for estimating reservoir temperatures in thermal springs, and its precipitation can influence the operational process. The dissolved ion content in geothermal fluids, indicative of the temperature and reservoir geology of an area, varies with temperature. Low-temperature fluids contain fewer dissolved solids than their high-temperature counterparts, making electrical conductivity (EC) an important measure for assessing dissolved particles in geothermal fluids [12].

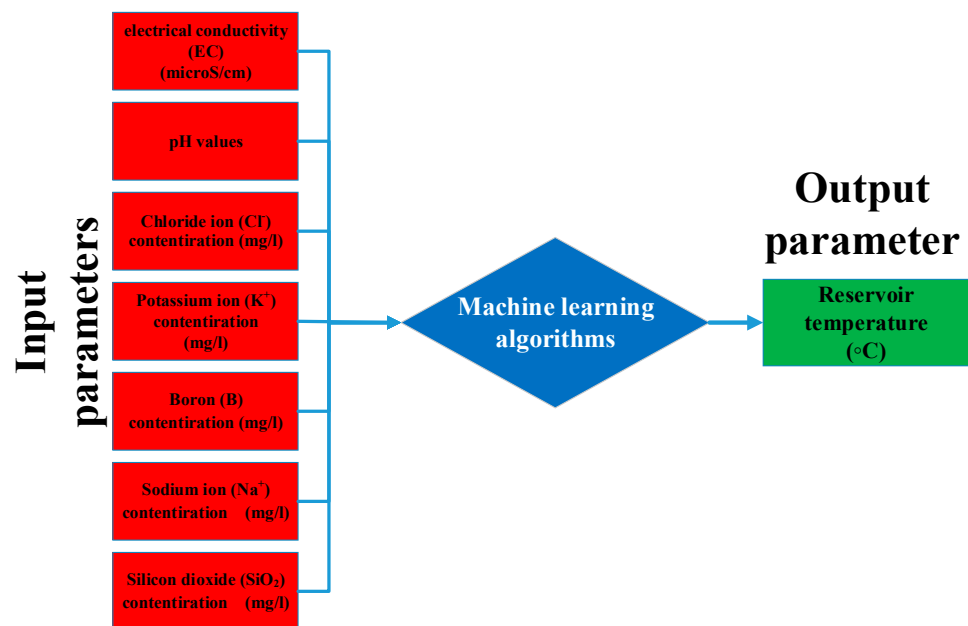


Figure 1. Input and output parameters used in modeling.

This section meticulously details the compilation of a comprehensive dataset utilized in the current study, drawing from various sources focusing on the hydrogeochemical properties and reservoir temperatures (RTs) of geothermal resources across Turkey, particularly within Anatolia. Following is a concise rephrasing to clarify and streamline the information:

The dataset used in this study amalgamates hydrogeochemical data and RTs from several geothermal resources across Western, Central, and Eastern Anatolia, Turkey. It integrates the following.

Western Anatolia: a collection of 83 datasets from [12], detailing hydrogeochemical properties and RTs of geothermal resources. Central Anatolia (Çamlıdere): hydrogeochemical data and RTs sourced from 12 different locations, as reported by [35]. Eastern Anatolia: a selection of 42 datasets from [36], specifically focusing on hot water sources from various

cities and locations. Central Anatolia (Nevşehir, Kozaklı): nine datasets from research documented in [37], excluding data from point 4 for consistency. Central Anatolia (Terme and Karakurt, Kırşehir): this study contributed seven datasets, excluding cold spring data but including hydrogeochemical and RT data [38]. Central Anatolia (Seydişehir, Kavak, Konya): eight datasets derived from literature [39], omitting cold spring data but incorporating measurements from April. Each dataset has been meticulously selected to ensure a robust analysis, excluding cold springs, to maintain focus on the geothermal energy potential. The range and specifics of the hydrogeochemical and RT data are tabulated in Table 1, providing a foundation for the comprehensive analysis undertaken in this study.

Table 1. Range of hydrogeochemical and RT dataset.

	Parameter	Unit	Value
Input Data	pH	-	2.4–9.7
	EC	microS/cm	203–10,434
	K ⁺	mg/L	0.7–191
	Na ⁺		2.6–2600
	Boron		0–38
	SiO ₂		11–650
	Cl ⁻		2.8–2500
Output Data	RT	°C	11–245

2.2. Artificial Neural Network

The artificial neural network (ANN) is a computational model inspired by the human brain's neural network. It typically consists of three primary layers: the input layer, one or more hidden layers, and the output layer [40,41]. The primary goal in training an ANN is continuous optimization, which involves mapping input to output to find optimal bias and weight values within the fewest possible iterations. The multilayer perceptron (MLP) is the most frequently utilized ANN structure in the literature. This section delves into MLP and the metaheuristic methods increasingly adopted for MLP training in recent years. Additionally, it covers evaluation metrics for regression problem assessments and data-normalization techniques.

Multi-Layer Perceptron Neural Networks

Similar to ANNs, the MLP functions by mapping a set of input values to a corresponding set of output values. This mapping is achieved through a transformation process designed to derive the output. An MLP consists of three layers: the input layer contains n input values; the hidden layer, positioned between the input and output layers, varies in size depending on the problem type; and the output layer, which aggregates the results of the MLP network [42]. Figure 2a,b illustrate the fundamental MLP structure and a single neuron's architecture, respectively. The input layer hosts n neurons, the output layer includes k neurons, and the hidden layer comprises m neurons. Each neuron in the hidden layer performs two critical operations: summation and activation. The sum obtained is subsequently passed through an activation function, as depicted in Equation (1).

$$Sum_j = \sum_{i=1}^n w_{ij} \cdot in_i + b_j \quad (1)$$

where w_{ij} is the connection weight between the hidden neuron j and the input neuron i . b_j is the bias value.

$$y_j = f(Sum_j) \quad (2)$$

where y_j is the output value of neuron j , and f is the sigmoid function. Its calculation is shown in Equation (3).

$$f(\text{Sum}_j) = \frac{1}{1 + e^{-\text{Sum}_j}} \tag{3}$$

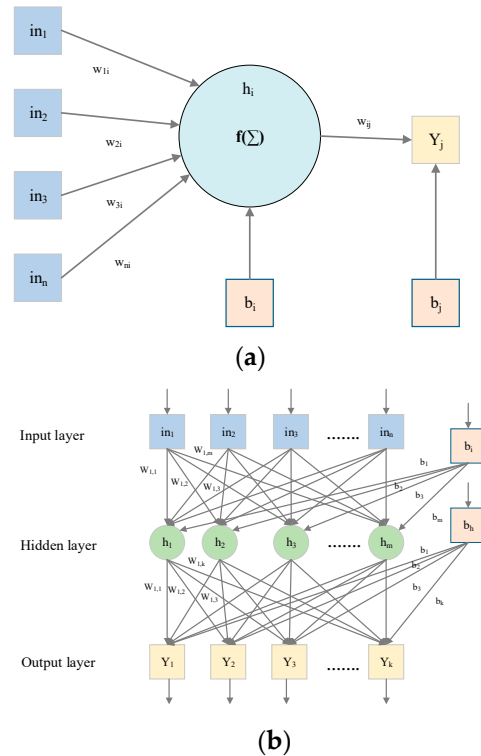


Figure 2. (a) One single neuron and (b) Single hidden layer MLP network.

At the end of these operations, the final output value \hat{Y}_i is calculated using the sum and activation operations. These operations are given in Equations (5) and (6).

$$\text{Sum}_j = \sum_{i=1}^m w_{ij} \cdot y_j + b_j \tag{4}$$

$$\hat{Y}_j = f(\text{Sum}_j) \tag{5}$$

Here Y_j is the final output of j .

2.3. Whale Optimization Algorithm

Mirjalili and Lewis (2016) introduced the whale optimization algorithm (WOA), a metaheuristic optimization technique inspired by the social behavior of humpback whales specifically designed to tackle challenging optimization problems [43]. Humpback whales are known for their unique hunting strategy, wherein they can pinpoint and entirely engulf their prey. In the context of WOA, this behavior is simulated by allowing the simulated whales to adjust their positions relative to a designated optimal search agent, analogous to the target prey in their natural hunting process. This adaptation of humpback whale behavior into a computational model is encapsulated in the mathematical formulations presented in Equations (6) and (7), detailing the algorithm’s mechanics of search and optimization.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \tag{6}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \tag{7}$$

The location vector of the optimal solution that has been found up to this point is denoted by \vec{X}^* . \vec{A} and \vec{C} coefficient vectors are calculated in Equations (8)–(10). The position vector is denoted by \vec{X} . t represents the most recent iteration.

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{8}$$

$$\vec{a} = 2 - 2 \frac{t}{t_{max}} \tag{9}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{10}$$

In the whale optimization algorithm (WOA), a coefficient vector “a” decreases linearly from 2 to 0 over the course of iterations, influencing both the exploration and exploitation phases. The vector “r” is a randomly generated vector within the range of [0, 1], with “t_max” representing the maximum number of iterations and “t” denoting the current iteration number. Equations (8) and (9) are designed to maintain a balance between exploration and exploitation during the optimization process. In these equations, “r” generates a random number, introducing stochastic elements into the population’s location updates. Exploration is initiated when $A \geq 1$, while exploitation occurs when $A < 1$, enabling the algorithm to transition between exploration and exploitation at any optimization stage.

The unique bubble-net attacking strategy of humpback whales is mimicked within WOA through two main mechanisms: the spiral update position mechanism and the shrinking encircling mechanism. The coefficient vector A , which influences these mechanisms, is determined by setting the value of “a” within the range of $[-1, 1]$ $[-1, 1]$, while the shrinking encircling mechanism reduces the value of “a” linearly through iterations. This approach positions the new location midway between its current location and that of the optimal search agent. To emulate the humpback whales’ spiral movements toward their prey, the spiral equation defining the path between the whale and the prey’s location is established. This mathematical representation captures the essence of the humpback whales’ hunting strategy, applying it to the optimization process in WOA.

$$\vec{X}(t + 1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \tag{11}$$

$$\vec{D}' = \left| \vec{X}^*(t) - \vec{X}(t) \right| \tag{12}$$

where \vec{D}' is the distance of the i whale from its target (the best solution so far), l is a random value within the range $[-1, 1]$, and b is a constant used to define the logarithmic spiral’s form. Because humpback whales swim in a circle that gets smaller and smaller around their prey while at the same time moving along the path in a spiraling fashion, the shrinking containment method and the spiral approach are used simultaneously. In order to model this behavior, it is presumed that each mechanism has a probability of occurring of fifty percent [44]. The method based on the variation of vector A can be utilized for hunting (exploration). Random searches are conducted by humpback whales based on one another’s locations. As a result, A is used in conjunction with random values less than or greater than 1 to force the search agent to depart from the reference whale. During the exploration phase, in contrast to the exploitation phase, they do not update the position of a search agent according to the best search agent discovered up to this point. Instead, they choose a search agent at random and base the update on that. This mechanism highlights the $A > 1$ discovery phase and allows WOA to conduct a global search. Equations (13) and (14) are a mathematical model of the process of hunting prey.

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X} \right| \tag{13}$$

$$\vec{X}(t + 1) = \vec{X}_{rand} - A \cdot D \tag{14}$$

where \vec{X}_{rand} is a randomly selected whale from the current population whose position vector is random.

2.4. Ant Lion Algorithm

The ant lion optimizer (ALO) algorithm models the interaction between ant lions and ants within the context of a trap [45]. In this algorithm, ant lions set traps to capture ants, thereby enhancing their fitness through successful predation. Ants, symbolizing potential solutions within the search space, navigate through this space to mirror these natural interactions. Reflecting the stochastic movement of ants in their quest for food, the algorithm employs a specific random walk strategy to simulate the ants' movement patterns. This approach captures the essence of the dynamic and unpredictable interactions that occur in nature, translating them into a computational method for solving optimization problems.

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)], \tag{15}$$

where *cumsum* is a function that calculates the cumulative sum, *n* is the maximum number of iterations that may occur, *t* represents the step of a random walk (referred to as an iteration in this research), and *r(t)* is a stochastic function defined as follows:

$$r(t) = \begin{cases} 1, & \text{if } rand > 0.5 \\ 0, & \text{if } rand \leq 0.5 \end{cases} \tag{16}$$

where *rand* is a uniformly distributed random number in the range [0, 1]. In the following matrix, the ant's location is noted and used for optimization:

$$M_{ant} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & \dots & A_{2,d} \\ & & & & \vdots \\ & & & & \vdots \\ A_{n,1} & A_{n,2} & \dots & \dots & A_{n,d} \end{bmatrix} \tag{17}$$

where M_{ant} stores the location of each ant, $A_{i,j}$ indicates the *i*th ant's *j*th variable (dimension), *d* is the number of variables, and *n* is the number of ants. During optimization, a fitness (objective) function is used to assess each ant, and the fitness value of each ant is stored in the following matrix:

$$M_{OA} = \begin{bmatrix} f([A_{1,1} & A_{1,2} & \dots & \dots & A_{1,d}]) \\ f([A_{2,1} & A_{2,2} & \dots & \dots & A_{2,d}]) \\ & & & & \vdots \\ & & & & \vdots \\ f([A_{n,1} & A_{n,2} & \dots & \dots & A_{n,d}]) \end{bmatrix} \tag{18}$$

where M_{OA} is the matrix used for preserving each ant's fitness, and *f* is the fitness (objective) function. In addition, the ant lions lurk in the search area. Their locations are saved using the following matrices:

$$M_{Antlion} = \begin{bmatrix} AL_{1,1} & AL_{1,2} & \dots & \dots & AL_{1,d} \\ AL_{2,1} & AL_{2,2} & \dots & \dots & AL_{2,d} \\ & & & & \vdots \\ & & & & \vdots \\ AL_{n,1} & AL_{n,2} & \dots & \dots & AL_{n,d} \end{bmatrix} \tag{19}$$

$M_{Antlion}$ is the matrix used to store each ant lion's location.

$$M_{OAL} = \begin{bmatrix} f([AL_{1,1} \ AL_{1,2} \ \dots \ AL_{1,d}]) \\ f([AL_{2,1} \ AL_{2,2} \ \dots \ AL_{2,d}]) \\ \vdots \\ f([AL_{n,1} \ AL_{n,2} \ \dots \ AL_{n,d}]) \end{bmatrix} \tag{20}$$

M_{OAL} is the matrix used to save each ant lion's fitness.

2.4.1. Random Walks of Ants

Equation (21) is used to normalize ants' positions and restrict them from wandering outside of the search area.

$$X_i^t = \frac{(X_i^t - a_i) - (d_i - c_i^t)}{d_i^t - a_i} + c_i \tag{21}$$

where a_i denotes the smallest random walk for the i th variable, c_i^t denotes the smallest variable at the t th iteration, and d_i^t represents the largest variable at the t th iteration.

2.4.2. Trapping in Antlion's Pits

Ant lion traps affect random ant walks. The following equations are proposed to describe this supposition:

$$c_i^t = Antlion_j^t + c^t \tag{22}$$

$$d_i^t = Antlion_j^t + d^t \tag{23}$$

where c^t stands for the vector containing all variables' minimum values at iteration t ; d^t stands for the vector containing all variables' maximum values at iteration t ; c_i^t and d_i^t represent the minimum and maximum values of all variables for the i -th ant, respectively; and $Antlion_j^t$ represents the position of the selected j -th.

2.4.3. Building Trap

The capacity of ant lions to hunt should be modeled using a selection method. The more fit the ant lion, the greater its chance of catching an ant. The roulette wheel selection method was used to choose ant lions based on their fitness value.

2.4.4. Sliding Ants towards Antlion

Ants must walk in a random pattern, while ant lions can construct traps according to their fitness. Antlions spray sand from the pit once an ant is inside the trap. This behavior hinders an ant's escape. In order to show this behavior, the size of the ants' random walk's hypersphere is made smaller. In this respect, the following equations are suggested:

$$c^t = \frac{c^t}{l} \tag{24}$$

$$d^t = \frac{d^t}{l} \tag{25}$$

where c^t represents the lowest value for all variables at iteration t , d^t is the vector containing t 's maximum value for all variables, and l is a ratio, as described by Equation (26).

$$l = 10^w \frac{t}{t_{max}} \tag{26}$$

where w is a constant determined depending on the current iteration ($w = 2$ when $t > 0.1t_{max}$, $w = 3$ when $t > 0.5t_{max}$, $w = 4$ when $t > 0.75t_{max}$, $w = 5$ when $t > 0.9t_{max}$,

and $w = 6$ when $t > 0.95t_{max}$), where t is the current iteration, and t_{max} is the maximum number of iterations. In essence, the constant w allows for exploitation accuracy level adjustment.

2.4.5. Sliding Ants towards Antlion

The prey is captured by the ant lion’s jaws as it descends to the bottom of the pit during the last phase of the hunt. The ant lion then drags the ant into the sand, where it eats the insect’s body. When an ant becomes more physically fit than its comparable ant lion, it is considered that prey is captured. In order to expand its capacity for hunting fresh ants, the ant lion must thus change its posture to that of the hunted ant. Equation (27) represents this process:

$$Antlion_j^t = Ant_i^t \text{ if } f(Ant_i^t) > f(Antlion_j^t) \tag{27}$$

where $Antlion_j^t$ displays the position of chosen j -th antlion at the t -th iteration, and Ant_i^t shows the position of the i -th ant at the t -th iteration.

2.4.6. Elitism

The ant lion that is fittest in each iteration is labeled an elite. The chosen ant lion and the elite ant lion direct the random walk of an ant using the selection process, and so the repositioning of a particular ant follows Equation (28).

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{28}$$

where Ant_i^t denotes the location of the i -th ant at the t -th iteration, R_A^t denotes the t -th iteration’s random walk around the ant lion chosen by the roulette wheel, and R_E^t denotes the t -th iteration’s random walk around the elite.

2.5. Slime Mould Algorithm and Adaptive Opposition Slime Mould Algorithm

A stochastic optimizer called SMA has been proposed by Li et al. [46]. Inspired by the oscillation of the mode of slime mould in nature, the proposed SMA method has been successfully applied in many studies in the literature [47,48]. Suppose there are N slime moulds with a lower boundary (LB) and upper boundary (UB) in the search space. The position of the i th slime mould in d dimensions is represented as $X_i = (x_i^1, x_i^2, \dots, x_i^d)$, $\forall i \in [1, N]$, and the fitness of i -th slime is expressed as $f(X_i), \forall i = [1, N]$. Thus, the position of N slime mould and its fitness at the present moment (iteration) t may be stated according to Equations (29) and (30):

$$X(t) = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^d \\ x_2^1 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \dots & x_N^d \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \tag{29}$$

$$f(X) = [f(X_1), f(X_2), \dots, f(X_N)] \tag{30}$$

For the next iteration ($t + 1$) in the SMA, the position of the slime mould is updated using Equation (31).

$$X_i(t+1) = \begin{cases} X_{LB}(t) + V_b(W \cdot X_A(t) - X_B(t)) & r_1 \geq \delta \text{ and } r_2 < p_i \\ V_c \cdot X_i(t) & r_1 \geq \delta \text{ and } r_2 \geq p_i \\ rand \cdot (UB - LB) + LB & r_1 < \delta \end{cases}, \forall i \in [1, N] \tag{31}$$

where V_b and V_c represent random velocity, X_{LB} represents the best local individual for the current iteration, W represents the weight vector, and X_A and X_B represent randomly pooled slime mould samples from the available populations. r_1 and r_2 are random values

between 0 and 1. The probability that the slime mould will appear at a random search site is set at δ 0.03. p_i is the i -th slime mould's threshold value, which assists in picking the slime mould position using the best individual or itself for the following iteration, and is computed as shown in Equation (32).

$$p_i = \tanh|f(X_i) - f_{GB}|, \forall i \in [1, N], \tag{32}$$

where $f(X_i)$ is the fitness value of the i -th slime mould X_i , and $f(X_{GB})$ is the global best fitness value f_{GB} derived by Equation (33) of the global best position X_{GB} .

$$f_{GB} = f(X_{GB}) \tag{33}$$

Equation (34) may be used to calculate the weight W of N different types of slime mould in the current iteration:

$$W(\text{sortInd}_f(i)) = \begin{cases} 1 + \text{rand} \cdot \log\left(\frac{f_{LB} - f(X_i)}{f_{LB} - f_{LW}} + 1\right) & 1 \leq i \leq \frac{N}{2} \\ 1 - \text{rand} \cdot \log\left(\frac{f_{LB} - f(X_i)}{f_{LB} - f_{LW}} + 1\right) & \frac{N}{2} < i \leq N \end{cases} \tag{34}$$

where “rand” represents a random number between [0, 1]. The local worst fitness value is represented by f_{Lw} , and the local best fitness value is represented by f_{LB} . f_{LB} and f_{Lw} are calculated based on the fitness value f given in Equation (30). We can sort the fitness value in ascending order for a minimization problem, as shown in Equation (35):

$$[\text{Sort}_f, \text{SortInd}_f] = \text{sort}(f) \tag{35}$$

The local worst fitness value Equation (36), the local best fitness value Equation (37), and the corresponding local best individual value Equation (38) are subtracted as follows:

$$f_{Lw} = f(\text{Sort}_f(N)) \tag{36}$$

$$f_{LB} = f(\text{Sort}_f(1)) \tag{37}$$

$$X_{LB} = X(\text{SortInd}_f(1)) \tag{38}$$

where V_b and V_c represent random velocities selected from a continuous uniform distribution in the $[-b, b]$ range and in the $[-c, c]$ range. The b and c values for the t iteration are calculated as shown in Equations (39) and (40):

$$b = \text{arctanh}\left(-\left(\frac{t}{T}\right) + 1\right) \tag{39}$$

$$c = 1 - \frac{t}{T} \tag{40}$$

The ideal feeding path for slime mould may be improved in the search process, as explained in Equation (31) [34]. Based on δ and p_i , the next iteration position update rule of slime mould in SMA is dependent on the following three cases:

Case 1: When $r_1 \geq \delta$ and $r_2 < p_i$, the slime mould search trajectory is directed by the best local slime X_{LB} and two randomly pooled slimes X_A and X_B with the velocity V_b . It balances exploration and exploitation. Case 1 indicates that X_A and X_B are arbitrarily combined slime mould; therefore, the solutions we acquire are not well explored and exploited [34]. This limitation may be solved by substituting X_A with the best local X_{LB} . In this case, the equation can be updated, as shown in Equation (41).

$$\begin{aligned}
 Xn_i(t) &= X_{LB}(t) + V_b(W \cdot X_{LB}(t) - X_B(t)), \text{ if } r_1 \geq \delta \text{ and } r_2 < p_i \\
 Xn_i(t) &= V_c \cdot X_i(t), \text{ if } r_1 \geq \delta \text{ and } r_2 \geq p_i \\
 Xn_i(t) &= rand \cdot (UB - LB) + LB, \text{ if } r_1 < \delta
 \end{aligned}
 \tag{41}$$

Case 2: The slime mould trajectory is directed by its own location with a velocity V_c when $r_1 \geq \delta$ and $r_2 \geq p_i$. This helps with exploitation. According to Case 2, the slime mould utilizes a neighboring location; thus, it may take a different route with a lower fitness value. An adaptive decision (AD) strategy might be a better option to overcome this limitation [34].

Case 3: When $r_1 < \delta$, in the search space, the slime mould re-initializes, assisting with exploration. Based on Case 3, the SMA provides a provision for devoted exploration; nevertheless, the exploration is limited since δ has a small value. We must add more exploration to SMA in order to assist it in circumventing this limitation and exceeding the local minima. To address the limitations of Cases 2 and 3, we employ an AD technique to determine if it is necessary to investigate further using OBL [34].

2.5.1. Opposition-Based Learning

The OBL updated the location of the subsequent iterations by comparing an estimate Xo_i in the search space that is the exact opposite of the position Xn_i for each slime mould ($i = 1, 2, \dots, N$). With increased convergence, this step reduces the likelihood of being stranded in the local minima. Thus, it is calculated that the Xo_i for the i -th slime mould in the j -th dimension is,

$$Xo_i^j(t) = \min(Xn_i(t)) + \max(Xn_i(t)) - Xn_i^j(t) \tag{42}$$

where $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, d$. i is chosen for the Xs_i minimization problem. The slime mould position is as follows:

$$Xs_i(t) = \begin{cases} Xo_i(t) & \text{if } f(Xo_i(t)) < f(Xn_i(t)) \\ Xn_i(t) & \text{if } f(Xo_i(t)) \geq f(Xn_i(t)) \end{cases} \tag{43}$$

2.5.2. Adaptive Decision Strategy

An AD is made based on the present fitness value of $f(Xn_i(t))$ and the old fitness value of $f(X_i(t))$ while the slime mould is pursuing a decedent nutrition route. When necessary, the AD uses OBL to augment the investigation. The AD method of AOSMA is then used to adjust the location for the next iteration and is depicted as shown in Equation (44).

$$X_i(t + 1) = \begin{cases} Xn_i(t) & \text{if } f(Xn_i(t)) \leq f(X_i(t)) \\ Xs_i(t) & \text{if } f(Xn_i(t)) > f(X_i(t)) \end{cases}, \forall i \in [1, N] \tag{44}$$

Surprisingly, the suggested AOSMA improves the efficiency of SMA by using an AD technique to determine if OBL is required along the search trajectory.

2.6. Metaheuristic Optimization Algorithms for Learning Mlp

In the literature, many metaheuristic methods have been proposed to develop the MLP network in the training phase. SMA, ALO, WOA and AOSMA metaheuristic methods are among those that are widely used with success. The successful application of these methods to benchmark functions and real-world problems in the literature has motivated the selection of the methods. In the training phase of MLP, metaheuristic optimization methods are used to determine the bias and weight values that make up the network connections. Four methods are used to find the optimal set of bias and weight values. There is no accepted equation in the literature for selecting the number of hidden nodes. In this study, it was obtained using the following equation:

$$m = 2 \cdot d + 1 \tag{45}$$

Here, d is the number of data features, and m is the number of neurons. The n value, that is, the total weight and bias value, was calculated using the equation below.

$$n = (d \cdot m) + (2 \cdot m) + 1 \tag{46}$$

Search agents commonly found in metaheuristic optimization methods represent a vector with n floating-point numbers. The flow chart of the proposed methods is demonstrated in Figure 3. The use of metaheuristic methods in the training phase of MLP generally covers four processes. First, initialization is initiated by specifying the MLP structure, such as the number of neurons (m) and the total weight and bias values (n). Then, the MLP network set—that is, the bias and weight values—are randomly generated. Then, the second step, the fitness evaluation step, is passed. In this step, the fitness value is calculated. For the fitness value, the fitness function must be selected. In this study, the MSE value was chosen, and its equation is given below.

$$MSE = \frac{1}{k} \sum_{i=1}^k (Y_i - \hat{Y}_i)^2 \tag{47}$$

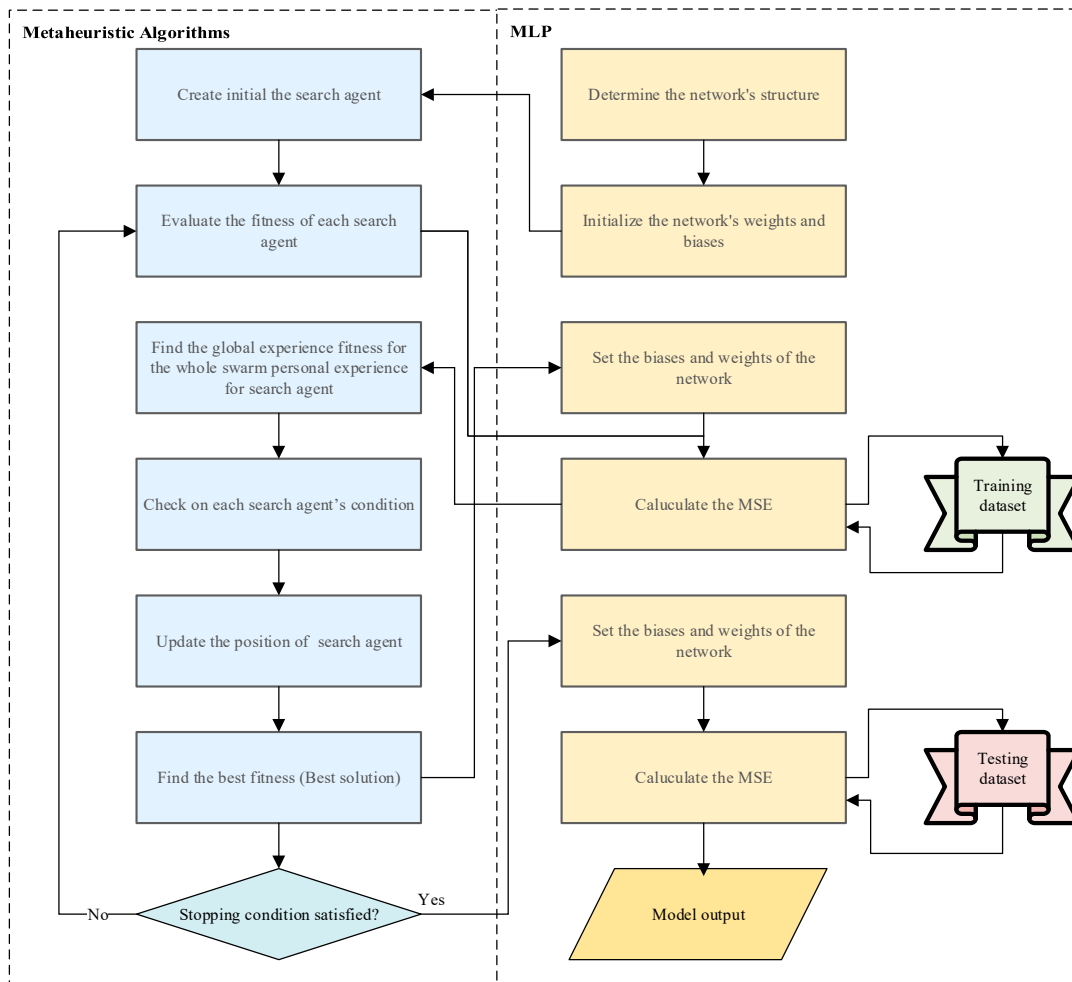


Figure 3. Flow chart of the metaheuristic learning algorithm.

Here, \hat{y}_i is the predictive value, y_i is the actual value, and k is the number of samples in the training set. In the third step, the update step, the best global fitness value, and personal fitness values are updated for each search agent. Finally, there is the termination step. The termination step is used to continue training the MLP until the maximum number of iterations is achieved.

2.7. Evaluation Metrics

Four different evaluation metrics, namely, R^2 , R , RMSE, and MAE, were used in this study. These metrics are widely used in the literature to compare the performances of the models proposed for predicting regression problems [40,49]. Equations of evaluation metrics are given in Equations (48)–(51).

$$R^2 = 1 - \left(\frac{\sum_j (Y_j - \hat{Y}_j)^2}{\sum_j (Y_j - \bar{Y})^2} \right) \tag{48}$$

$$p(Y, \hat{Y}) = \frac{1}{n-1} \sum_{j=1}^n \left(\frac{Y_j - \mu_Y}{\sigma_Y} \right) \left(\frac{\hat{Y}_j - \mu_{\hat{Y}}}{\sigma_{\hat{Y}}} \right) \tag{49}$$

$$RMSE = \sqrt{\left(\frac{1}{n} \right) \times \sum_j |Y_j - \hat{Y}_j|^2} \tag{50}$$

$$MAE = \frac{1}{n} \sum_{j=1}^n |Y_j - \hat{Y}_j| \tag{51}$$

2.8. Normalization

In this study, min-max [0, 1] normalization was applied to the dataset. The min-max normalization equation is given in Equation (52).

$$min - max; x'_{i,n} = \frac{x_{i,n} - min(x_i)}{max(x_i) - min(x_i)} \tag{52}$$

3. Results

In this study, the reservoir temperature (RT) was predicted using several approaches: an MLP trained with a classical optimization method, MLPs trained with three distinct metaheuristic methods, and the AOSMA-MLP, which was proposed for the first time. The dataset comprised 161 samples, encompassing hydrogeochemical data and RTs from geothermal resources across various regions in Anatolia, Turkey. Understanding RTs is essential for determining the most effective use of geothermal resources, whether for direct heating applications or electricity generation, depending on the RT. The dataset was divided into training and test sets, constituting 80% (128 samples) and 20% (33 samples) of the data, respectively. This division ensured consistency across the four models proposed in the study. The metaheuristic methods employed were the whale optimization algorithm (WOA), ant lion optimizer (ALO), slime mould algorithm (SMA), and the newly proposed adaptive opposition-based slime mould algorithm (AOSMA). Characteristics of the MLPs trained using these metaheuristic optimization methods are detailed in Table 2, including the number of hidden neurons (HNO), which was kept consistent across all models. For the classical optimization-based training of MLP, the one-step secant backpropagation method was utilized. Table 2 outlines various parameters, including the number of attributes (NA), samples (NS), training samples (NTRS), test samples (NTS), hidden neurons (HNO), dimension (Dim), weight (W), bias (B), and neural network structure (NNS). Additionally, Table 3 lists the parameters of the employed metaheuristic optimization methods, with the maximum number of iterations and the number of search agents standardized at 200 and 30, respectively, to facilitate equal evaluation conditions across all methods.

Table 2. Properties of the MLP.

NA	NS	NTRS	NTS	HNO	Dim	W	B	NNS
7	161	128	33	15	136	120	16	7-15-1

Table 3. The parameters of the metaheuristic optimization methods.

Algorithm	Parameters	Values
WOA	\vec{a} Shape of the logarithmic spiral (b)	Linearly decreased from 2 to 0 1
SMA	δ	0.03
AOSMA	δ	0.03

The training durations for the metaheuristic optimization methods, namely, ALO-MLP, WOA-MLP, SMA-MLP, and AOSMA-MLP, were recorded as 13.07, 8.06, 8.17, and 8.26 min, respectively, highlighting the efficiency and potential applicability of these advanced computational techniques in geothermal resource evaluation.

4. Discussion

To assess the effectiveness of the models proposed in this study, four widely recognized evaluation metrics were employed: R-squared (R^2), correlation coefficient (R), root mean square error (RMSE), and mean absolute error (MAE). The comparative results of the four models are systematically presented in Table 4, with the most favorable outcomes highlighted for clarity.

Table 4. Performance of ANN, ALO-MLP, WOA-MLP, SMA-MLP, and AOSMA-MLP on hydrogeochemical and RT data.

	R^2	R	RMSE	MAE
ANN	0.7169	0.8701	36.94	29.28
WOA-MLP	0.7765	0.8969	32.82	25.26
ALO-MLP	0.8096	0.9049	30.29	25.67
SMA-MLP	0.7955	0.8974	31.39	26.81
AOSMA-MLP	<u>0.8514</u>	<u>0.9233</u>	<u>26.76</u>	<u>21.45</u>

A detailed analysis revealed the following insights based on the evaluation metrics:

- **R^2 Value:** The AOSMA-MLP algorithm outperformed the ANN, WOA-MLP, ALO-MLP, and SMA-MLP methods by 18.76%, 9.65%, 5.16%, and 7.03%, respectively, indicating a more accurate fit to the data.
- **R Value:** In terms of correlation, the AOSMA-MLP method exhibited superior performance by 6.11%, 2.94%, 2.03%, and 2.89% compared to the ANN, WOA-MLP, ALO-MLP, and SMA-MLP algorithms, respectively, suggesting stronger linear relationships between predicted and observed values.
- **RMSE Value:** The AOSMA-MLP model demonstrated a significant reduction in prediction error, showing improvements of 27.56%, 18.46%, 11.65%, and 14.75% over the ANN, WOA-MLP, ALO-MLP, and SMA-MLP algorithms, respectively.
- **MAE:** In terms of absolute errors, the AOSMA-MLP approach was found to be more precise, reducing errors by 26.74%, 15.08%, 16.44%, and 19.99% compared to the ANN, WOA-MLP, ALO-MLP, and SMA-MLP algorithms, respectively.

Upon a comprehensive review of Table 4, the AOSMA-MLP model surpasses its counterparts across all evaluated metrics, establishing its superiority and effectiveness in predicting the reservoir temperatures (RTs) of geothermal resources. This comparative analysis underscores the potential of the AOSMA-MLP approach as a more accurate and reliable method for RT prediction in geothermal studies. In the study conducted by Tut Hakkidir, the RT value was predicted using linear regression, a linear support vector machine, and a deep neural network (DNN). A total of 83 different datasets were used in the study, and the best result was obtained in DNN. DNN achieved an MAE value of 6.45 and an RMSE value of 8.29 [12]. In the study conducted by Bemah et al., the RT value was estimated using natural gradient boosting (NGB), extreme learning machine, group method

of data handling, generalized regression neural network, and back propagation neural network methods. A total of 83 different datasets were used in the study, and the best result was obtained with the NGB method. The NGB MAE was calculated at 3.97, RMSE at 4.59, and R^2 at 0.9959 [50]. In another study by Quan et al., the RT value was estimated using ANN, support vector regression genetic algorithm SVR, and improved support vector machine (M-GASVR) methods. The M-GASVR method yielded the best results in the study. The M-GASVR was calculated as MAE 0.45, RMSE 0.556, and R^2 0.903 [51]. Perez-Zarate and colleagues conducted a study using different ANN models to predict the RT value. In the study, the ANN model that gave the best results was calculated as MAE 18.32, RMSE 26.4971, and R 0.7165 [24].

Figure 4 compares predicted and actual reservoir temperatures (RTs) using four distinct models. Figure 4 shows that the multilayer perceptron (MLP) trained via the adaptive opposition-based slime mould algorithm (AOSMA) yields predictions that are significantly closer to actual values compared to other models. Notably, at index numbers 1, 15, 17, and 28, AOSMA-MLP demonstrates remarkable accuracy, predicting values that closely mirror the true RTs.

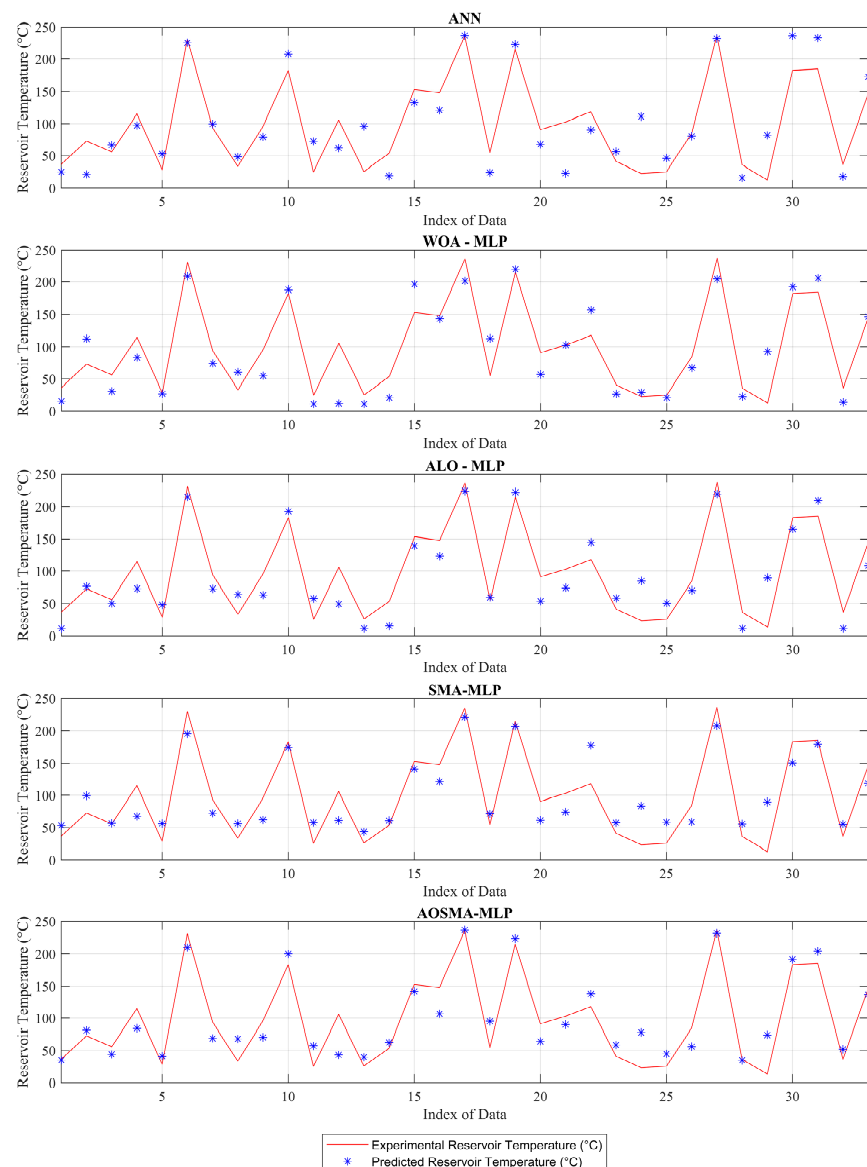


Figure 4. Experimental and predictive values of ANN, ALO-MLP, WOA-MLP, SMA-MLP, and AOSMA-MLP in hydrogeochemical and RT data.

Further insight is provided by scatter plots depicted in Figure 5. This graphical representation reinforces the accuracy of AOSMA-MLP, showcasing its predictions in close alignment with real RT values. The scatter plots vividly illustrate the precision of AOSMA-MLP, underscoring its superior performance in forecasting RTs based on the hydrogeochemical characteristics of geothermal resources.

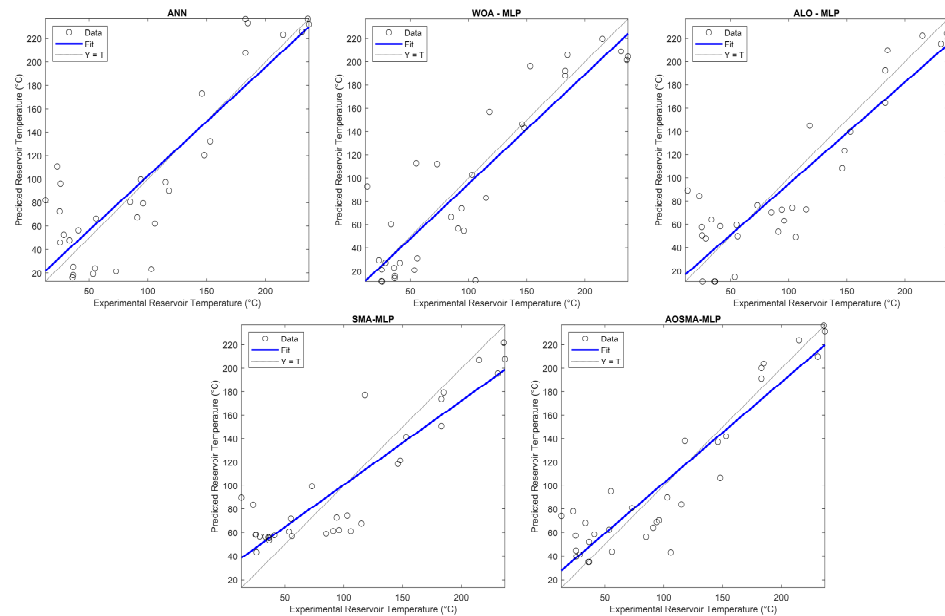


Figure 5. Scatterplots of the ANN, ALO-MLP, WOA-MLP, SMA-MLP, and AOSMA-MLP.

The ability to accurately predict RTs from the hydrogeochemical properties of geothermal resources signifies a substantial advancement. This approach mitigates the drawbacks associated with traditional geological studies, which are often time-consuming, expensive, and reliant on specialized equipment. Consequently, the financial burden on geothermal energy projects is lessened, offering a considerable advantage to investors. Moreover, this predictive capability enhances flexibility in feasibility assessments for engineers and experts engaged in geothermal energy projects. The proposed models not only aid in determining the potential applications of geothermal waters by preemptively identifying RTs but also serve as a decision-support system, enabling the selection of the most suitable options for exploiting geothermal resources.

5. Conclusions

In this study, the adaptive opposition-based slime mould algorithm (AOSMA) was innovatively applied during the training phase of the multilayer perceptron (MLP) for the first time. This novel approach was specifically tailored for predicting reservoir temperatures (RTs) and was benchmarked against a conventional first-order optimization technique, one-stage secant backpropagation, and three distinct stochastic metaheuristic optimization methods. The metaheuristic techniques compared included the whale optimization algorithm (WOA), ant lion optimizer (ALO), and slime mould algorithm (SMA), all of which are extensively recognized in the literature. Key findings from this comparative analysis are as follows:

- The AOSMA-MLP demonstrated superior performance relative to the other metaheuristic optimization algorithms tested. By leveraging AOSMA for ANN training with hydrogeochemical and RT data derived from geothermal sources, it effectively addressed common limitations of alternative methods, such as susceptibility to local minima and constraints on global exploration capabilities.

- Across the board, AOSMA-MLP showcased a distinct advantage over competing methods across the four different evaluation metrics employed in this study. This underscores its efficacy and robustness in predicting RTs.
- In terms of accuracy of fit to the data, the AOSMA-MLP algorithm performed better than the ANN, WOA-MLP, ALO-MLP, and SMA-MLP approaches by 18.76%, 9.65%, 5.16%, and 7.03%, respectively.
- The promising outcomes achieved with AOSMA-MLP indicate its potential applicability across a broad spectrum of regression problems, extending beyond the scope of this study.
- The AOSMA-MLP model demonstrated a significant reduction in prediction error, showing improvements of 27.56%, 18.46%, 11.65%, and 14.75% over the ANN, WOA-MLP, ALO-MLP, and SMA-MLP algorithms, respectively.
- The application of the AOSMA-MLP model for predicting RTs in geothermal resources is projected to significantly aid engineers and project planners in identifying optimal drilling locations. Given the typically time-intensive, expensive, and complex nature of such determinations, this model can substantially reduce project costs and enhance flexibility within the investment and design phases of geothermal energy projects.
- Overall, the introduction and application of AOSMA-MLP represent a significant advancement in geothermal energy research, offering practical benefits for the planning and execution of geothermal projects.

Author Contributions: Conceptualization, E.G. and O.A.; methodology, E.G. and O.A.; software, O.A. and E.V.A.; validation, O.A. and E.V.A.; formal analysis, E.G. and O.A.; investigation, E.G. and O.A.; resources, E.G., O.A. and E.V.A.; data curation, E.G., O.A. and E.V.A.; writing—original draft preparation, E.G., O.A. and E.V.A.; writing—review and editing, E.G., O.A. and E.V.A.; visualization, E.G., O.A. and E.V.A.; supervision, E.G., O.A. and E.V.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

EU	European Union
RT	Reservoir temperature
ANN	Artificial Neural Network
FNNS	Feed-Forward Networks
MLP	Multilayer Perceptron
WOA-MLP	Whale Optimization
ALO-MLP	Ant Lion Optimizer
SMA-MLP	Slime Mould Algorithm
AOSMA	Adaptive Opposition SMA
EC	Electrical Conductivity
Cl ⁻	Chloride ion
K ⁺	Potassium ion
B	Boron
Na ⁺	Sodium ion
SiO ₂	Silicon dioxide

References

1. Luo, Z.; Wang, Y.; Zhou, S.; Wu, X. Simulation and prediction of conditions for effective development of shallow geothermal energy. *Appl. Therm. Eng.* **2015**, *91*, 370–376. [[CrossRef](#)]
2. Çetin, T.H.; Zhu, J.; Ekici, E.; Kanoglu, M. Thermodynamic assessment of a geothermal power and cooling cogeneration system with cryogenic energy storage. *Energy Convers. Manag.* **2022**, *260*, 115616. [[CrossRef](#)]
3. Ambriz-Díaz, V.M.; Rubio-Maya, C.; Chávez, O.; Ruiz-Casanova, E.; Pastor-Martínez, E. Thermodynamic performance and economic feasibility of Kalina, Goswami and Organic Rankine Cycles coupled to a polygeneration plant using geothermal energy of low-grade temperature. *Energy Convers. Manag.* **2021**, *243*, 114362. [[CrossRef](#)]
4. Werner, S. International review of district heating and cooling. *Energy* **2017**, *137*, 617–631. [[CrossRef](#)]
5. Gang, W.; Wang, S.; Xiao, F. District cooling systems and individual cooling systems: Comparative analysis and impacts of key factors. *Sci. Technol. Built Environ.* **2017**, *23*, 241–250. [[CrossRef](#)]
6. Moya, D.; Aldás, C.; Kaparaju, P. Geothermal energy: Power plant technology and direct heat applications. *Renew. Sustain. Energy Rev.* **2018**, *94*, 889–901. [[CrossRef](#)]
7. Inayat, A.; Raza, M. District cooling system via renewable energy sources: A review. *Renew. Sustain. Energy Rev.* **2019**, *107*, 360–373. [[CrossRef](#)]
8. Kanoğlu, M.; Çengel, Y.A.; Cimbala, J.M. *Fundamentals and Applications of Renewable Energy*; McGraw-Hill Education: New York, NY, USA, 2020.
9. Gang, W.; Wang, S.; Xiao, F.; Gao, D.-c. District cooling systems: Technology integration, system optimization, challenges and opportunities for applications. *Renew. Sustain. Energy Rev.* **2016**, *53*, 253–264. [[CrossRef](#)]
10. Rostamzadeh, H.; Gargari, S.G.; Namin, A.S.; Ghaebi, H. A novel multigeneration system driven by a hybrid biogas-geothermal heat source, Part II: Multi-criteria optimization. *Energy Convers. Manag.* **2019**, *180*, 859–888. [[CrossRef](#)]
11. Michaelides, E.E.S. Future directions and cycles for electricity production from geothermal resources. *Energy Convers. Manag.* **2016**, *107*, 3–9. [[CrossRef](#)]
12. Tut Haklıdır, F.S.; Haklıdır, M. Prediction of reservoir temperatures using hydrogeochemical data, Western Anatolia geothermal systems (Turkey): A machine learning approach. *Nat. Resour. Res.* **2020**, *29*, 2333–2346. [[CrossRef](#)]
13. Okan, Ö.Ö.; Kalender, L.; Çetindağ, B. Trace-element hydrogeochemistry of thermal waters of Karakoçan (Elazığ) and Mazgirt (Tunceli), Eastern Anatolia, Turkey. *J. Geochem. Explor.* **2018**, *194*, 29–43. [[CrossRef](#)]
14. Haklıdır, F.S.T.; Haklıdır, M. The fluid temperature prediction with hydro-geochemical indicators using a deep learning model: A case study Western Anatolia (Turkey). In Proceedings of the 43rd Workshop on Geothermal Reservoir Engineering, Stanford, CA, USA, 11–13 February 2019.
15. Porkhial, S.; Salehpour, M.; Ashraf, H.; Jamali, A. Modeling and prediction of geothermal reservoir temperature behavior using evolutionary design of neural networks. *Geoth* **2015**, *53*, 320–327. [[CrossRef](#)]
16. Alacalı, M. Hydrogeochemical investigation of geothermal springs in Erzurum, East Anatolia (Turkey). *Environ. Earth Sci.* **2018**, *77*, 802. [[CrossRef](#)]
17. Altay, E.V.; Gurgenc, E.; Altay, O.; Dikici, A. Hybrid artificial neural network based on a metaheuristic optimization algorithm for the prediction of reservoir temperature using hydrogeochemical data of different geothermal areas in Anatolia (Turkey). *Geoth* **2022**, *104*, 102476. [[CrossRef](#)]
18. Bassam, A.; Santoyo, E.; Andaverde, J.; Hernández, J.; Espinoza-Ojeda, O. Estimation of static formation temperatures in geothermal wells by using an artificial neural network approach. *Comput. Geosci.* **2010**, *36*, 1191–1199. [[CrossRef](#)]
19. Fannou, J.-L.C.; Rousseau, C.; Lamarche, L.; Kajl, S. Modeling of a direct expansion geothermal heat pump using artificial neural networks. *Energy Build.* **2014**, *81*, 381–390. [[CrossRef](#)]
20. Kalogirou, S.A.; Florides, G.A.; Pouloupatis, P.D.; Panayides, I.; Joseph-Stylianou, J.; Zomeni, Z. Artificial neural networks for the generation of geothermal maps of ground temperature at various depths by considering land configuration. *Energy* **2012**, *48*, 233–240. [[CrossRef](#)]
21. Bourhis, P.; Cousin, B.; Loria, A.F.R.; Laloui, L. Machine learning enhancement of thermal response tests for geothermal potential evaluations at site and regional scales. *Geoth* **2021**, *95*, 102132. [[CrossRef](#)]
22. Zhang, X.; Wang, E.; Liu, L.; Qi, C. Machine learning-based performance prediction for ground source heat pump systems. *Geoth* **2022**, *105*, 102509. [[CrossRef](#)]
23. Rezvanbehbahani, S.; Stearns, L.A.; Kadivar, A.; Walker, J.D.; van der Veen, C.J. Predicting the geothermal heat flux in Greenland: A machine learning approach. *GeoRL* **2017**, *44*, 12271–12279. [[CrossRef](#)]
24. Pérez-Zárate, D.; Santoyo, E.; Acevedo-Anicasio, A.; Díaz-González, L.; García-López, C. Evaluation of artificial neural networks for the prediction of deep reservoir temperatures using the gas-phase composition of geothermal fluids. *Comput. Geosci.* **2019**, *129*, 49–68. [[CrossRef](#)]
25. Yang, J.; Ma, J. Feed-forward neural network training using sparse representation. *Expert Syst. Appl.* **2019**, *116*, 255–264. [[CrossRef](#)]
26. Orr, M.J. *Introduction to Radial Basis Function Networks*; University of Edinburgh: Edinburgh, Scotland, 1996.
27. Fekri-Ershad, S. Bark texture classification using improved local ternary patterns and multilayer neural network. *Expert Syst. Appl.* **2020**, *158*, 113509. [[CrossRef](#)]
28. Altay, O.; Ulas, M.; Alyamac, K.E. DCS-ELM: A novel method for extreme learning machine for regression problems and a new approach for the SFRSCC. *PeerJ Comput. Sci.* **2021**, *7*, e411. [[CrossRef](#)] [[PubMed](#)]

29. Aljarah, I.; Faris, H.; Mirjalili, S.; Al-Madi, N.; Sheta, A.; Mafarja, M. Evolving neural networks using bird swarm algorithm for data classification and regression applications. *Clust. Comput.* **2019**, *22*, 1317–1345. [[CrossRef](#)]
30. Liao, J.; Asteris, P.G.; Cavaleri, L.; Mohammed, A.S.; Lemonis, M.E.; Tsoukalas, M.Z.; Skentou, A.D.; Maraveas, C.; Koopialipour, M.; Armaghani, D.J. Novel fuzzy-based optimization approaches for the prediction of ultimate axial load of circular concrete-filled steel tubes. *Buildings* **2021**, *11*, 629. [[CrossRef](#)]
31. Aljarah, I.; Faris, H.; Mirjalili, S. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput.* **2018**, *22*, 1–15. [[CrossRef](#)]
32. Heidari, A.A.; Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M. Ant lion optimizer: Theory, literature review, and application in multi-layer perceptron neural networks. In *Nature-Inspired Optimizers*; Springer: Cham, Switzerland, 2020; pp. 23–46.
33. Zubaidi, S.L.; Abdulkareem, I.H.; Hashim, K.S.; Al-Bugharbee, H.; Ridha, H.M.; Gharghan, S.K.; Al-Qaim, F.F.; Muradov, M.; Kot, P.; Al-Khaddar, R. Hybridised artificial neural network model with slime mould algorithm: A novel methodology for prediction of urban stochastic water demand. *Water* **2020**, *12*, 2692. [[CrossRef](#)]
34. Naik, M.K.; Panda, R.; Abraham, A. Adaptive opposition slime mould algorithm. *Soft Comput.* **2021**, *25*, 14297–14313. [[CrossRef](#)]
35. Pasvanoğlu, S.; Çelik, M. Hydrogeochemical characteristics and conceptual model of Çamlidere low temperature geothermal prospect, northern Central Anatolia. *Geoth* **2019**, *79*, 82–104. [[CrossRef](#)]
36. Aydın, H.; Karakuş, H.; Mutlu, H. Hydrogeochemistry of geothermal waters in eastern Turkey: Geochemical and isotopic constraints on water-rock interaction. *J. Volcanol. Geotherm. Res.* **2020**, *390*, 106708. [[CrossRef](#)]
37. Pasvanoğlu, S.; Chandrasekharam, D. Hydrogeochemical and isotopic study of thermal and mineralized waters from the Nevşehir (Kozakli) area, Central Turkey. *J. Volcanol. Geotherm. Res.* **2011**, *202*, 241–250. [[CrossRef](#)]
38. Pasvanoğlu, S.; Gültekin, F. Hydrogeochemical study of the Terme and Karakurt thermal and mineralized waters from Kirşehir Area, central Turkey. *Environ. Earth Sci.* **2012**, *66*, 169–182. [[CrossRef](#)]
39. Bozdağ, A. Hydrogeochemical and isotopic characteristics of Kavak (Seydişehir-Konya) geothermal field, Turkey. *J. Afr. Earth Sci.* **2016**, *121*, 72–83. [[CrossRef](#)]
40. Ulas, M.; Altay, O.; Gurgenc, T.; Özel, C. A new approach for prediction of the wear loss of PTA surface coatings using artificial neural network and basic, kernel-based, and weighted extreme learning machine. *Friction* **2020**, *8*, 1102–1116. [[CrossRef](#)]
41. Asteris, P.G.; Maraveas, C.; Chountalas, A.T.; Sophianopoulos, D.S.; Alam, N. Fire resistance prediction of slim-floor asymmetric steel beams using single hidden layer ANN models that employ multiple activation functions. *Steel Compos. Struct.* **2022**, *44*, 769–788.
42. Ren, H.; Ma, Z.; Lin, W.; Wang, S.; Li, W. Optimal design and size of a desiccant cooling system with onsite energy generation and thermal storage using a multilayer perceptron neural network and a genetic algorithm. *Energy Convers. Manag.* **2019**, *180*, 598–608. [[CrossRef](#)]
43. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
44. Altay, E.V. Gerçek Dünya Mühendislik Tasarım Problemlerinin Çözümünde Kullanılan Metasezgisel Optimizasyon Algoritmalarının Performanslarının İncelenmesi. *Int. J. Innov. Eng. Appl.* **2022**, *6*, 65–74.
45. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
46. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
47. Altay, O. Chaotic slime mould optimization algorithm for global optimization. *Artif. Intell. Rev.* **2022**, *55*, 3979–4040. [[CrossRef](#)]
48. Mostafa, M.; Rezk, H.; Aly, M.; Ahmed, E.M. A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel. *Sustain. Energy Technol. Assess.* **2020**, *42*, 100849. [[CrossRef](#)]
49. Altay, O.; Gurgenc, T.; Ulas, M.; Özel, C. Prediction of wear loss quantities of ferro-alloy coating using different machine learning algorithms. *Friction* **2020**, *8*, 107–114. [[CrossRef](#)]
50. Ibrahim, B.; Konduah, J.O.; Ahenkorah, I. Predicting reservoir temperature of geothermal systems in Western Anatolia, Turkey: A focus on predictive performance and explainability of machine learning models. *Geoth* **2023**, *112*, 102727. [[CrossRef](#)]
51. Quan, Q.; Hao, Z.; Xifeng, H.; Jingchun, L. Research on water temperature prediction based on improved support vector regression. *Neural Comput. Appl.* **2022**, *34*, 8501–8510. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.