MUŞ ALPARSLAN ÜNİVERSİTESİ
HENDESE TEKNİK BİLİMLER VE MÜHENDİSLİK DERGİSİ
MUŞ ALPARSLAN UNIVERSITY
JOURNAL OF HENDESE TECHNICAL SCIENCES AND ENGINEERING
Muş Alparslan Üniversitesi Külliyesi Teknik Bilimler Meslek Yüksekokulu, 49250, Muş, Türkiye
https://dergipark.org.tr/en/pub/hendese/board
e-ISSN: 3023-7777
HENDESE Ekim 2025, 2 (2): 44-102
Cilt / Volume: 2 **Sayı / Issue: 2** **Yıl / Year: 2025**

Classification of IRIS Data Set with Arithmetic Optimization Algorithm and Statistical Results

Dilan AKCALI [1], Beytullah YAGIZ [2], Erdal EKER [1]

[1]*Muş Alparslan University, Muş, TÜRKİYE*
dilanakcali0@gmail.com —ORCID 0009-0005-4226-7381
e.eker@alparslan.edu.tr —ORCID 0000-0002-5470-8384
[2]*Department of Statistics, Institute of Science, Van Yuzuncu Yil University, Van, TÜRKİYE*
beytmat@gmail.com—ORCID 0000-0001-5453-185X

Article Information

Article Types	Research Article
Received	18 June 2025
Accepted	11 September 2025

Year: 2025 | **Volume:** 2 | **Issue:** 2 | **Pages:** 89-95

Cite as: D. Akcalı, B. Yagiz, and E. Eker, "Classification of IRIS Data Set with Arithmetic Optimization Algorithm and Statistical Results," Hendese Journal of Technical Sciences and Engineering, Vol. 2, No. 2, pp. 89–95, 2025, doi: 10.5281/zenodo.17474578.

Corresponding Author: Beytullah YAGIZ

https://doi.org/10.5281/zenodo.17474578

HENDESE TEKNİK BİLİMLER ve MÜHENDİSLİK DERGİSİ
JOURNAL of HENDESE TECHNICAL SCIENCES and ENGINEERING
RESEARCH ARTICLE

Classification of IRIS Data Set with Arithmetic Optimization Algorithm and Statistical Results

Dilan AKCALI[1], **Beytullah YAGIZ***[2], **Erdal EKER**[1]

[1]*Muş Alparslan University, Muş, TÜRKİYE*

[2]*Department of Statistics, Institute of Science, Van Yuzuncu Yil University, Van, TÜRKİYE*

(**Received:** 18.06.2025, **Accepted:** 11.09.2025, **Published Online:** 30.10.2025)

Keywords
Arithmetic optimization algorithm,
Artificial neural network,
Multilayer perceptrons,
Optimization

ABSTRACT

The aim of this paper is to optimize the classification performance with the arithmetic optimization algorithm, one of the swarm-based intelligent algorithms, by using the multilayer perceptron model, which is an artificial neural network architecture. Model training is provided by IRIS flower data, which is widely used. arithmetic optimization algorithm is a metaheuristic optimization method inspired by basic arithmetic functions consisting of discovery and exploitation phases. The multilayer perceptron model is structured to consist of input, hidden, and output layers and is trained to classify the types of flowers in the IRIS dataset. The model's performance was evaluated using statistical metrics such as accuracy, recall, and F1 score. Simulations were carried out using the MATLAB package program. When the results were examined, the average accuracy rate of the model was measured as 96.7%. The recall rate was 96.0% and the F1 score was 96.3%. These results show that hybridizing metaheuristic algorithms to artificial intelligent network models can produce effective and efficient results in complex datasets.

Corresponding Author: beytmat@gmail.com

Volume: 2 **Issue:** 2

Cite as: D. Akcalı, B. Yagiz, and E. Eker, "Classification of IRIS Data Set with Arithmetic Optimization Algorithm and Statistical Results," *Hendese Journal of Technical Sciences and Engineering*, Vol. 2, No. 2, pp. 89–95, 2025, doi: 10.5281/zenodo.17474578.

Doi: 10.5281/zenodo.17474578

1. INTRODUCTION

Machine learning and optimization techniques have become powerful tools for tackling complex challenges. In this setting, optimization involves the process by which a system meets a goal within certain rules and limitations. It includes the entire process of setting up the necessary input protocols to ensure the desired output operations within the established framework of rules and boundaries. Artificial neural networks (ANNs), a type of machine learning algorithm, offer numerous architectural possibilities for optimizing problems. Multi-layer perceptron (MLP), one such architecture, are used for data classification. Optimization algorithms for training MLP have been developed to guide the supervision process [1]. The MLP architecture provides several benefits, such as a wide range of applications and a structure that is easy to adapt. This architecture can address various types of problems, including classification and regression. It offers an effective framework for learning non-linear, complex problems, reveals hidden patterns in data clusters, and can be smoothly integrated with other deep learning layers, like evolutionary neural networks, allowing for the creation of more intricate and comprehensive model structures [2]. Due to these benefits, the MLP is used as a versatile and powerful tool. However, the MLP architecture also has certain drawbacks. Primarily, it does not ensure an optimal solution, which is a fundamental limitation of this structure. Especially in complex problem domains, there is a risk of straying from the global optimum. Additionally, the model's performance heavily relies on the correct selection of algorithm settings, a process that requires experience and expertise. Furthermore, operating MLP with multilayered structures and large data clusters incurs a high computational cost, posing a challenge, particularly in resource-limited applications. These drawbacks can directly affect the applicability and efficiency of a model [3]. Nonetheless, there has been a relative increase in MLP-related applications in recent years [4].

Metaheuristic algorithms are a type of optimization method used in the framework of MLP. These algorithms are known for their broad search strategies, which are not limited to particular problems, allowing them to tackle complex optimization issues. Unlike traditional heuristics, metaheuristic algorithms are crafted to thoroughly explore vast solution spaces. They are favoured over conventional mathematical programming because they can pinpoint optimal areas within the search space, handle large-scale problems, and function with greater efficiency. These techniques are beneficial as they employ multiple initial solutions, do not require derivative information, and do not demand continuity. By approaching the problem as a hidden structure, they prove effective in finding solutions in unexplored search regions [5].

In recent times, there has been a significant rise in the exploration of metaheuristic algorithms. Noteworthy instances of these algorithms include the genetic algorithm (GA) [6], ant colony optimization (ACO) [1], artificial bee colony (ABC) [7], grey wolf optimization (GWO) [8], harris hawks optimization (HHO) [9], particle swarm optimization (PSO) [10], and the hunger games algorithm (HGS) [11]. In the realm of engineering, the application of metaheuristic algorithms, which provide more adaptable and creative solutions, has become increasingly common. While these algorithms do not promise a conclusive solution, they are adapting at examining a wider array of possibilities and achieving better outcomes. Particularly, nature-inspired optimization algorithms can deliver quicker and more accurate solutions than conventional methods. Despite the considerable benefits of metaheuristic algorithms, they also have certain drawbacks. Firstly, these algorithms are not guaranteed to discover the optimal solution; they often produce only near-optimal results [12]. Moreover, the effectiveness of these algorithms heavily depends on the parameters used, which can be difficult to fine-tune. The substantial computational cost is another significant limitation, especially for large-scale issues. Additionally, in some cases, algorithms might get stuck in local optima within the solution space and fail to achieve the global optimum. These aspects are important drawbacks to consider when using metaheuristic algorithms. Metaheuristic algorithms are commonly employed in MLP architecture [13].

In this paper, the IRIS dataset from the UCI repository was processed using the AOA and trained through the MLP architecture, achieving high performance [14], [15]. AOA is renowned for its simplicity and effectiveness, utilizing mathematical operations to optimize solutions. It aims to find the global optimum by employing swarm intelligence during problem-solving, which is particularly advantageous for complex and high-dimensional datasets [16]. AOA has been applied to real-world challenges, such as maximum power point tracking, pattern recognition, and image segmentation [17], as well as in post-earthquake analysis for detecting structural damage in reinforced concrete structures. It has also been used to analyse energy production variations in solar panels [18], model biological components, and classify and segment medical parts [19]. Despite its benefits, AOA's limited discovery ability and tendency to become trapped in local optima have been identified as areas needing improvement. Consequently, various hybridization studies have been conducted based on the No Free Lunch theorem [20]. One such study found that an AOA-based ANN algorithm performed well in classification and prediction tasks. Particularly in classification studies, hybrid methods demonstrated high performance criteria. The arithmetic optimization algorithm was combined with Lévy random steps (LRS) to enhance linear search capabilities and avoid local search traps, leading to more optimal results and improved performance of traditional engineering solutions [21].

The primary aim of this paper is to develop a MLP model using a metaheuristic algorithm. Specifically, it seeks to optimize the MLP model with the AOA to improve classification performance and conduct a statistical evaluation of the results. The outcomes will be presented through an analysis of the proposed method using metrics such as accuracy, recall rate, and F1 scores. The Iris dataset was employed in this study, as it is a balanced and well-structured dataset frequently used in classification problems, providing an appropriate platform for assessing the performance of AOA.

The paper is organized into four main sections. Initially, the Introduction discusses the use and advantages of metaheuristic algorithms in MLP architecture. The Literature Review section examines the types of these algorithms and the applications of the AOA in previous studies. The third section, Method and Application, outlines the process of optimizing the MLP with the Iris dataset using AOA. Finally, in the Results and Discussion section, the performance of AOA on the MLP is evaluated using statistical metrics, and the results are summarized.

2. ARITHMETIC OPTIMIZATION ALGORITHM

Developed by Abualigah *et al.* [22], AOA is a swarm-based metaheuristic optimization method inspired by mathematical arithmetic concepts. AOA effectively executes the stages of exploration and exploitation by emulating the distribution properties of the fundamental arithmetic operations: addition, subtraction, multiplication, and division. The optimization process of AOA comprises the initialization, exploration, and exploitation phases. In the initial phase, the algorithm's basic parameters are introduced alongside the candidate solution set. During the exploration phase, the objective is to comprehensively scan the search space and avoid entrapment in local optima. At this stage, the positions of the search agents are updated using the multiplication ("×") and division ("÷") operators; multiplication enhances the diversity of solutions, while division increases solution precision. In the exploitation phase, the aim is to refine the solutions obtained during exploration. At this stage, existing solutions are made more precise using the addition ("+") and subtraction ("−") operators, thereby narrowing the search focus. Similar to other metaheuristic algorithms, AOA initiates the optimization process with a randomly generated set of candidate solutions. These candidate solutions are iteratively evaluated and refined through specific optimization rules and an objective function. AOA is capable of solving optimization problems without requiring derivative calculations, thus offering a broad range of applications. This systematic structure enables AOA to balance global and local search, enhancing its effectiveness in the problem-solving process.

At the initial stage, the x matrix is created. The purpose of x is to indicate the population that the AOA algorithm initially randomly generated (1). In the matrix N, when indicating the candidate solutions or population size n, can express how many dimensions each solution has.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{1}$$

The optimization process commences with an initial set, and the algorithm endeavours to generate superior solutions by enhancing the population in each iteration [22]. During each iteration, the most optimal candidate solution within the population is regarded as the best or near-optimal solution identified thus far, and the algorithm continues by preserving this solution. Through this process, the population's information is enriched, leading to the discovery of more effective solutions. Prior to the initiation of the AOA, decisions regarding exploration and exploitation were informed by the outcome of the Mathematical Optimizer Accelerated (MOA) function, as determined by the Eq. 2 [22].

$$MOA\,(C\_Iter) = \min + C\_Iter \times \left(\frac{\max - \min}{iter_{max}}\right) \tag{2}$$

MOA (C_Iter), represents the functional outcome in the iteration. C_Iter refers to the current iteration between 1 and the maximum number of iterations ($iter_{max}$). The least value of MOA is represented by min notation, and the maximum value is indicated by max notation.

$$MOP(C\_Iter) = 1 - \frac{C\_Iter^{\frac{1}{\alpha}}}{iter_{max}^{\frac{1}{\alpha}}} \tag{3}$$

where Math Optimizer Probobaply (MOP) is a coefficient, $iter_{current}$ pecifies the number of iterations. $iter_{max}$ epresents the maximum number of iterations of the algorithm. α is a fixed parameter that determines the sensitivity of exploitation [22].

$$x_{i,j}(C_{Iter} + 1) \begin{cases} best(x_j) \div (MOP) \times \left((ub_j - lb_j) \times \mu + L_j\right), & rand2 < 0,5 \\ best(x_j) \times (MOP) \times \left((U_j - L_j) \times \mu + L_j\right), & otherwise \end{cases} \tag{4}$$

The exploitation phase constitutes a pivotal component of the AOA, facilitating an in-depth exploration of the local regions within the solution space. During this phase, the sensitivity of the solution is heightened by steering candidate solutions towards the most optimal solution available. This process involves updates executed through mathematical formulas predicated on arithmetic operations, thereby ensuring a concentrated focus on the local area. Concurrently, a degree of randomness is preserved to avert convergence to a local optimum. Consequently, the exploitation phase enables the algorithm to further refine the optimal solution [23].

3. MULTI LAYER PERCEPTRON

MLP is a type of feedforward artificial neural network comprising three layers: the input, hidden, and output layers. The input layer serves as the initial stage of the model, responsible for receiving the data properties into the network. Following the input layer is the hidden layer, which processes the information and performs nonlinear transformations. The output layer, succeeding the hidden layer, generates the network's output and prediction values. In this study, the iris dataset was structured according to the MLP architecture. The input layer consists of sensors that define the characteristics of the iris dataset, the hidden layer comprises sensors weighted to optimize the nonlinear problem structure, and the output layer is designed to classify iris flower species. In MLP, each neuron is connected to all neurons in the preceding layer, with these connections referred to as synapses. These synapses transmit information between layers, assigning specific weights to inputs. The output of a neuron is calculated by multiplying the inputs it receives by weights, with a bias added to this total. The resulting value is conveyed to the next layer in hidden layers, typically through a nonlinear activation function such as Sigmoid. In the output layer, activation functions such as sigmoid, softmax, or linear are employed depending on the problem [24]. The sigmoid function converts input values into outputs in the range (0, 1).

$$f(s_j) = \mathrm{sigmoid}(s_j) = \frac{1}{[1+e^{-s_j}]}, j = 1,2,\ldots,h \quad (5)$$

$$o_k = \sum_{j=1}^{h} w_{jk} f(s_j) - \theta'_k, k = 1,2,\ldots,m \quad (6)$$

Mean Squared Error (MSE) is a metric used to evaluate the performance of statistical models. MSE is calculated by averaging the squares of the differences between the observed values and the values predicted by the model. In this way, it quantitatively expresses how close the model's predictions are to the actual values [25].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (7)$$

where n,number of data points, Y_i, observed value and, $\hat{Y}_i$: estimated value.

Conversely, cross-validation statistical metrics offer insights into a model's accuracy, consistency, and ability to generalize. These metrics are derived from the confusion matrix, which is a table used to evaluate the performance of a classification algorithm. A confusion matrix provides a visual and summary representation of how well a classification algorithm performs [26].

Table I. Confusion matrix.

	Predicted Class	
Actual Class	TP	FN
	FP	TN

Table I is called confusion matrix. In the matrix, true positive (TP) is when the model successfully predicts a positive outcome, which matches the actual positive result. A true Negative (TN) occurs when the model correctly forecasts a negative outcome, consistent with the actual negative result. A false positive (FP), also known as a Type I error, arises when the model incorrectly predicts a positive outcome, while the actual result is negative. Conversely, a false negative (FN), or Type II error, happens when the model inaccurately predicts a negative outcome, despite the actual result being positive. Accuracy is the simplest performance measure, indicating the ratio of correct predictions to the total number of observations, calculated as;

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (8)$$

Precision is the ratio of correctly predicted positive outcomes to all predicted positive outcomes, expressed as

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (9)$$

Recall represents the ratio of correctly predicted positive outcomes to all actual positive outcomes, given by

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (10)$$

The F1 Score, which is the harmonic mean of Precision and Recall, accounts for both false positives and false negatives. Although it may not be as straightforward as accuracy, the F1 Score is often more advantageous, especially when there is an imbalance in class distribution.

$$\text{F1 Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (11)$$

MLP is trained using supervised learning, with the backpropagation algorithm frequently employed in this process. The model's prediction error is propagated backward from the output layer to the input layer, and the weights are adjusted based on this error signal. During this process, hyperparameters such as the learning rate and momentum significantly influence the learning speed and stability of the model. The performance of the MLP is contingent upon factors such as the number of layers, the number of neurons within each layer, the quality of the training data, and the training parameters. While an increased number of hidden layers enhances the model's capacity to learn complex relationships, it also poses a risk of overfitting. Consequently, the MLP is utilized as a robust modelling tool, particularly in complex data analysis requiring supervised learning, due to its ability to learn nonlinear functions and its feedforward architecture.

4. CONSEQUENCES OF AOA BASED MLP

AOA-MLP pseudocode is importmant point the assessment of optimization procees.

ALGORITHM: AOA-based MLP Training for Classification
INPUT:
- *Dataset (ınput, Output)*
- *Network architecture (input_size, hidden_size, output_size)*
- *AOA parameters (N: population size, M_Iter)*
- *Search bounds (LB, UB) for weights/biases*

OUTPUT:
- *Optimized MLP weights/biases, Classification accuracy, Convergence curve*

BEGIN
1. *DATA PREPROCESSING:*
   - *Load and normalize the dataset*
   - *Encode labels*
   - *Split into training and validation sets*
2. *NETWORK INITIALIZATION:*
   - *Define the MLP architecture*
   - *Calculate the total dimension*
   - *Set search bounds [-10, 10]*
3. *AOA INITIALIZATION:*

- Initialize population X of size N
- Set parameters: MOP_Max=1, MOP_Min=0.2, Alpha=5, Mu=0.499
- Initialize Best_Position and Best_Fitness
- Initialize the convergence curve
4. FITNESS FUNCTION:
FUNCTION MLP_Fitness(solution):
- Extract weights and biases
- Construct the network
- Forward propagate data
- Calculate Mean Squared Error (MSE)
- Return MSE
END FUNCTION
5. MAIN OPTIMIZATION LOOP:
FOR iteration = 1 to M_Iter:
a. Calculate MOP and MOA
b. FOR each agent i:
FOR each dimension j:
- Generate r1, r2, r3
IF r1 < MOA:
IF r2 > 0.5:
X_new[i,j] = Best_Position[j] / (MOP + eps) × ((UB - LB) × Mu + LB)
ELSE:
X_new[i,j] = Best_Position[j] × MOP × ((UB - LB) × Mu + LB)
END IF
ELSE:
IF r3 > 0.5:
X_new[i,j] = Best_Position[j] - MOP × ((UB - LB) × Mu + LB)
ELSE:
X_new[i,j] = Best_Position[j] + MOP × ((UB - LB) × Mu + LB)
END IF
END IF
END FOR
- Apply constraints
- Evaluate fitness
- Update positions and fitness
END FOR
c. Update the convergence curve
END FOR
6. EVALUATION:
- Extract optimal parameters
- Test the network
- Calculate accuracy, precision, recall, and F1-score
7. OUTPUT:
- Display accuracy
- Plot convergence
- Show metrics
- Return parameters
END ALGORITHM

In this paper, the MLP model was optimized using the AOA and applied to the IRIS dataset for classification purposes. The intuitive superstructure of the AOA facilitated a more effective determination of MLP weights, resulting in significant improvements in the model's classification performance. The network architecture comprises an input layer, a hidden layer, and an output layer. Given the characteristics of the iris flower dataset, the input layer is configured with four nodes, while the output layer contains three nodes. The hidden layer is composed of nine nodes, determined using the 4*2+1 formula. This configuration was identified as the most appropriate architecture following a process of trial and error. The findings indicate that the MLP model optimized with AOA demonstrates high performance in metrics such as accuracy, recall rate, and F1 score as seen from Fig. 1 to Fig. 4. This suggests that AOA provides a more robust and flexible alternative to traditional optimization methods. Consequently, the integration of metaheuristic algorithms like AOA with artificial neural networks holds the potential to enhance success in classification problems.

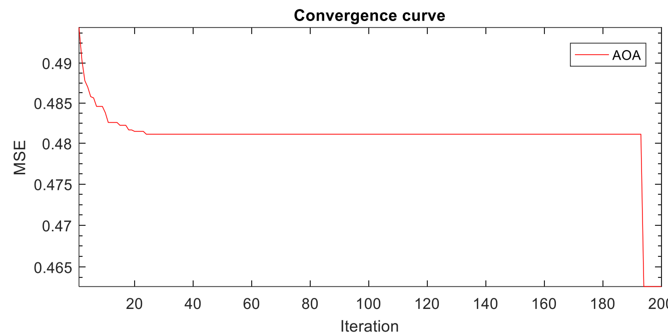

Fig.1. Converge curve of AOA.

Fig. 1 shows the change in the MSE value, which measures the performance of the AOA algorithm, according to the number of iterations. The convergence curve shows that the algorithm converges rapidly in the first 40 iterations or so and remains at a stably low error value until the 200th iteration.

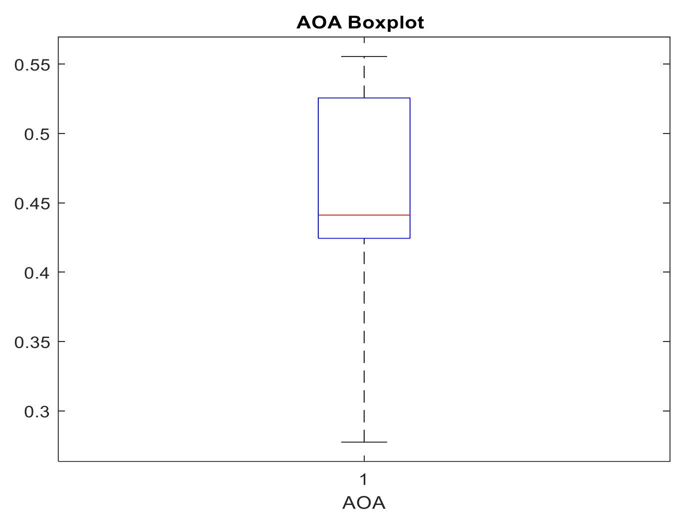

Fig. 2. Boxplot of AOA.

Box plot of MSE values obtained from different runs of the AOA method in Fig. 2 the box shows the median, interquartile distribution, and outliers. This graph is intended to evaluate the reliability and stability of the algorithm's error distribution. As can be seen in the graph, although there are no extreme values, the median value is closer to the first card, the proximity of the values to the optimal point can be considered as an advantage.

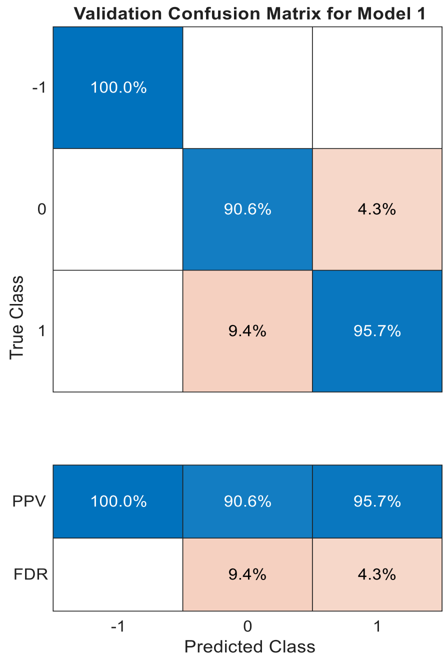

Fig. 3. Validation confusion matrix.

When Fig. 3 is examined, the complexity matrix of the validation dataset obtained for Model 1. It shows the comparison between the actual classes (-1, 0, 1) and the predicted classes. High percentages on the diagonal reflect the correct classification performance of the model, while cells outside the diagonal reveal confusion between classes. In addition, positive predictive value (PPV) and false discovery rate (FDR) metrics are presented at the bottom.

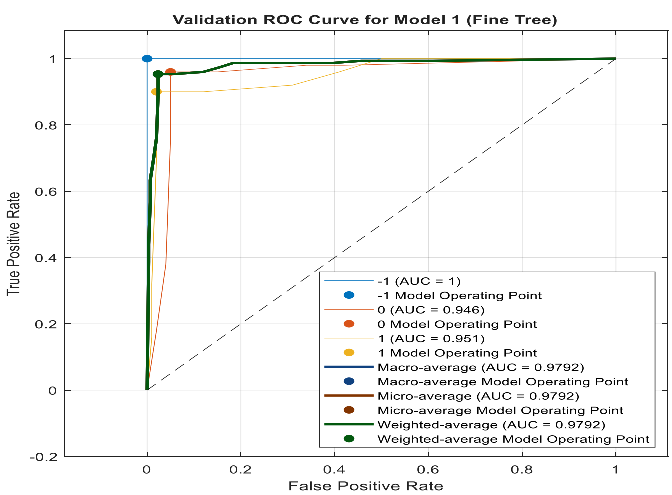

Fig. 4. Validation ROC curve.

The model's performance was assessed using metrics such as accuracy, recall, specificity, precision, and F1-score, along with the MSE rate and cross-validation. These metrics were derived from the confusion matrix. In experiments conducted on the IRIS dataset, the model's MSE was found to be 0.1194, with an average accuracy rate of 96.7%. The recall rate stood at 96.0%, and the F1 score was 96.3%. These results indicate that the model excels in accurately identifying positive classes and overall classification. Upon reviewing these figures, a low MSE value, derived from statistical measures, indicates a minimal error rate. The F1 score, which represents the harmonic mean, is notably high, suggesting that the model exhibits strong generalizability in cross-validation. This is due to the high accuracy rate observed between the MSE ratio and the F1 score. Moreover, MLP model, optimized with AOA, achieved a higher accuracy rate with fewer iterations compared to traditional training algorithms. This suggests that the training process has become more efficient, thereby reducing computational costs.

5. CONCLUSION

In this paper, the efficacy of artificial intelligence-based methodologies in addressing classification problems is demonstrated through the integration of MLP model, a machine learning algorithm, with the AOA. By leveraging physics-based metaheuristic algorithms, the model exhibits a more flexible and adaptable structure compared to conventional methods. The advantages of this study include the flexibility of the proposed method and its ease of adaptation to various problem types. Furthermore, the model's overall structure indicates its potential applicability to diverse datasets and real-world challenges. Among the study's limitations, it is posited that the model's success is contingent upon the algorithm's parameter settings, which may increase computational time. There is a potential issue of convergence to local optima; however, the algorithm's flexibility allows for further enhancement. For future research, it is recommended to evaluate the proposed model on different, more complex, and larger datasets and to develop hybrid models with other optimization algorithms. Additionally, assessing the model's performance in real-world problem-solving scenarios could enhance the method's generalizability and effectiveness.

REFERENCES

[1] D. Izci, S. Ekinci, E. Eker, and M. Kayri, "Improved Manta Ray foraging optimization using opposition-based learning for optimization problems," *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–6, Jun. 2020, doi: 10.1109/hora49412.2020.9152925.

[2] P. Bansal, S. Gupta, S. Kumar, S. Sharma, and S. Sharma, "MLP-LOA: A metaheuristic approach to design an optimal multilayer perceptron," *Soft Computing*, vol. 23, no. 23, pp. 12331–12345, Jan. 2019, doi: 10.1007/s00500-019-03773-2.

[3] R. Yu, W. Yu, and X. Wang, "KAN or MLP: A fairer comparison," *arXiv preprint*, arXiv:2407.16674, Aug. 17, 2024, doi: 10.48550/arXiv.2407.16674.

[4] T. Hoya, "Review of the two existing artificial neural network models—Multilayer perceptron and probabilistic neural networks," in *Syntactic Networks—Kernel Memory Approach*, vol. 1157, *Studies in Computational Intelligence*, Cham, Switzerland: Springer, 2024, pp. 17–34, doi: 10.1007/978-3-031-57312-5_2.

[5] P. Bansal, S. Kumar, S. Pasrija, and S. Sharma, "A hybrid grasshopper and new cat swarm optimization algorithm for feature selection and optimization of multi-layer perceptron," *Soft Computing*, vol. 24, pp. 15463–15489, Oct. 2020, doi: 10.1007/s00500-020-04877-w.

[6] M. Gerşil and T. Palamutçuoğlu, "Ders çizelgeleme probleminin melez genetik algoritmalar ile performans analizi," *Niğde*

Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, vol. 6, no. 1, pp. 242–262, Apr. 2013.

[7] E. Küçüksille and M. Tokmak, "Yapay Arı Kolonisi Algoritması Kullanarak Otomatik Ders Çizelgeleme," *Süleyman Demirel University Journal of Natural and Applied Sciences*, vol. 15, no. 3, pp. 203–210, Jul. 2011.

[8] F. A. Şenel, A. S. Yüksel, F. Gökçe, and T. Yiğit, "Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi," *Journal of Balıkesir University Institute of Science and Technology*, vol. 20, no. 2, pp. 293–306, Dec. 2018, doi: 10.25092/baunfbed.433321.

[9] S. Ekinci, B. Hekimoğlu, A. Demirören, and S. Kaya, "Harris Hawks Optimization Approach for Tuning of FOPID Controller in DC-DC Buck Converter," in *Proc. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP)*, Malatya, Turkey, 2019, pp. 1–9, doi: 10.1109/IDAP.2019.8875992.

[10] S. Ekinci, B. Hekimoğlu, E. Eker, and D. Sevim, "Hybrid Firefly and Particle Swarm Optimization Algorithm for PID Controller Design of Buck Converter," in *Proc. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, 2019, pp. 1–6, doi: 10.1109/ISMSIT.2019.8932733.

[11] Ö. Can, "Otomatik Gerilim Regülatörü Sistemi için Farklı Optimizasyon Teknikleri Ayarlı Denetleyici Tasarımı," *The Black Sea Journal of Sciences*, vol. 13, no. 3, pp. 858–870, Sep. 2023, doi: 10.31466/kfbd.1252029.

[12] "Sezgisel algoritma," *Vikipedi*, Jun. 23, 2024. Accessed: Dec. 24, 2024. [Online]. Available: https://tr.wikipedia.org/w/index.php?title=Sezgisel_algoritma&oldid=33358727

[13] S. Mirjalili, "How effective is the Grey Wolf optimizer in training multi-layer perceptrons," *Applied Intelligence*, vol. 43, no. 1, pp. 150–161, Jul. 2015, doi: 10.1007/s10489-014-0645-7.

[14] A. Kaveh and K. B. Hamedani, "Improved arithmetic optimization algorithm and its application to discrete structural optimization," *Structures*, vol. 35, pp. 748–764, Jan. 2022, doi: 10.1016/j.istruc.2021.11.012.

[15] R. A. Fisher, "Iris," *UCI Machine Learning Repository*, 1936, doi: 10.24432/C56C76.

[16] O. Cihan, "Çok Etmenli Sistemlerde Bir Dağıtık Denklem Çözüm Algoritmasının Yakınsama Hızı En İyilemesi," *Avrupa Bilim ve Teknoloji Dergisi (European Journal of Science and Technology)*, no. 26 (HORA Special Issue), pp. 262–269, Jul. 2021, doi: 10.31590/ejosat.952456.

[17] A. G. Yılmaz and S. Alsamoua, "Meta sezgisel arama algoritmalarının çok seviyeli görüntü eşikleme performanslarının karşılaştırılması," *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, Nov. 2024, doi: 10.28948/ngumuh.1529614.

[18] A. R. Dal, "Güneş enerji panellerindeki optimum eğim açısının verime etkisinin incelenmesi," *Bilecik Şeyh Edebali University Journal of Science*, vol. 8, no. 1, pp. 241–250, Jun. 2021, doi: 10.35193/bseufbd.878795.

[19] R. Terzi and M. U. Demirezen, "Açık kaynak veri seti ile eğitilen yapay zeka modellerinin klinik ortamdaki performans analizi," *Journal of Ankara University Faculty of Medicine (Ankara Üniversitesi Tıp Fakültesi Mecmuası)*, vol. 75, pp. 25, 2022, doi: 10.4274/atfm.galenos.2022.97830.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[21] S. Barua and A. Merabet, "Lévy Arithmetic Algorithm: An enhanced metaheuristic algorithm and its application to engineering optimization," *Expert Systems with Applications*, vol. 241, p. 122335, May 2024, doi: 10.1016/j.eswa.2023.122335.

[22] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The Arithmetic Optimization Algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113609, Apr. 2021, doi: 10.1016/j.cma.2020.113609.

[23] J. O. Agushaka and A. E. Ezugwu, "Advanced arithmetic optimization algorithm for solving mechanical engineering design problems," *PLOS ONE*, Aug. 2021, doi: 10.1371/journal.pone.0255703.

[24] E. Eker, M. Kayri, S. Ekinci, and D. Izci, "Training Multi-Layer Perceptron Using Harris Hawks Optimization," in *Proc. 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey, Jun. 2020, pp. 1–5, doi: 10.1109/HORA49412.2020.9152874.

[25] T. O. Hodson, T. M. Over, and S. S. Foks, "Mean Squared Error, Deconstructed," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 12, e2021MS002681, Dec. 2021, doi: 10.1029/2021MS002681.

[26] E. Beauxis-Aussalet and L. Hardman, "Simplifying the visualization of confusion matrix," in *Proc. 26th Benelux Conference on Artificial Intelligence (BNAIC)*, Nov. 2014.