

RESEARCH

Metaheuristic-Driven Optimization of a Neural Model Using Tuna Swarm Intelligence for Cognitive Classification of Wheat Species

Beytullah Yağız¹ · Erdal Eker² · Yener Altun¹ · Şeyma Nur Atar² · Davut İzci^{3,7} · Serdar Ekinci⁴ · Mohit Bajaj^{5,8} · Mohammad Shabaz⁶

Received: 17 May 2025 / Revised: 19 September 2025 / Accepted: 15 October 2025

© The Author(s) 2025

Abstract

Accurate classification of wheat varieties is vital for enhancing agricultural productivity and maintaining food quality standards. However, conventional methods based on manual inspection are time-consuming, labor-intensive, and prone to human error. This paper introduces an innovative methodology that integrates the tuna swarm optimization algorithm with a multi-layer perceptron neural network to achieve high-accuracy classification of wheat species. The SEEDS dataset, containing three wheat species (Kama, Rosa, and Canadian), was used to evaluate the effectiveness of the proposed model. Initially, the tuna swarm optimization algorithm's robustness was validated using the CEC 2019 benchmark suite, where it demonstrated superior performance against alternative metaheuristics such as the Archimedes optimization algorithm, prairie dog optimization, and Harris hawks optimization. Tuna swarm optimization achieved a 79.29% classification accuracy (Macro and Weighted F1-score = 0.7923) while significantly reducing the mean squared error to $7.52\text{E} - 02$, outperforming other optimization algorithms. The proposed tuna swarm optimization-based multi-layer perceptron exhibited strong generalization capability, as demonstrated by the ROC-AUC values exceeding 0.77 for all classes, with the highest classification success observed in class "0" (AUC = 0.9851). Compared to conventional gradient-based training methods, the tuna swarm optimization-enhanced multi-layer perceptron model consistently outperformed alternative techniques in convergence stability, classification precision, and robustness across multiple iterations. These results highlight the potential of metaheuristic-driven neural network optimization in agricultural applications, providing a highly efficient and scalable solution for automated wheat classification.

Keywords Agricultural machine learning · Metaheuristic algorithms · Multi-layer perceptron · Optimization techniques · Swarm intelligence · Tuna swarm optimization · Wheat classification · Wheat species recognition

1 Introduction

1.1 Background

Wheat is one of the most critical cereals in the global food sector, fulfilling a substantial portion of humanity's basic nutritional needs. Ensuring sustainable and sufficient wheat production is essential for global food security. As the world population grows, demand for wheat is increasing, while factors such as climate change and urbanization reduce agricultural land, impacting production negatively. Under these conditions, quality classification offers one of the most effective ways to boost production efficiency. By accurately classifying wheat types, processing can be optimized to



produce various sub-products. However, in many regions, classification still relies on human observation, which is labor-intensive and time-consuming. More efficient, data-driven methods are needed to improve wheat quality classification.

1.2 Related Works

Machine learning (ML) offers promising solutions, enabling decision-making and classification through pre-processed datasets, advancing the field of artificial learning. Artificial learning encompasses the learning capability of computer systems, with machine learning as a sub-layer where algorithms refine solutions iteratively to achieve more optimal results. Each iteration in ML builds on prior knowledge, improving the model over time. This approach provides specific solutions by training models on data [1–3].

In optimization contexts, determining the best parameter values to enhance output is termed optimization [4]. Given the pervasive need for optimization across research fields, continuous development and adaptation of optimization methods are essential [5, 6]. Modeling and analysis are the core pillars of optimization: modeling involves formulating the problem mathematically, while solving entails finding the optimal solution to this model [7–11].

Metaheuristic algorithms, known for their effectiveness in optimization, have become prominent in recent literature. Unlike deterministic methods, metaheuristic algorithms use stochastic processes, directly seeking solutions through exploration and exploitation phases [12, 13]. Some prominent metaheuristics include simulated annealing (SA) [14], which begins optimization from a single solution, and swarm-based methods that start with multiple candidates, iteratively refining solutions in global and local spaces [15]. Notable metaheuristics include genetic algorithm (GA) [16], particle swarm optimization (PSO) [17], firefly algorithm (FA) [18, 19], cuckoo search algorithm (CS) [20], flower pollination algorithm (FPA) [21], and more recent techniques such as grey wolf optimizer (GWO) [22] and tuna swarm optimization (TSO) [23].

Among the various metaheuristic algorithms, PSO and GWO are commonly used for training neural networks due to their simplicity and effectiveness. However, these algorithms sometimes face premature convergence and may struggle to maintain an optimal balance between exploration and exploitation, particularly in complex, high-dimensional search spaces. Two different algorithms are needed to train an artificial neural network. The first one is the machine learning algorithm that forms the architecture. In this study, MLP architecture is utilized. The second algorithm serves as a supervisory algorithm that trains the MLP to achieve optimal performance in data training. In this study, the TSO algorithm fulfills this role. Artificial neural networks (ANNs), specifically multilayer perceptron (MLP), are widely used in solving nonlinear problems due to their ability to model complex data relationships [24]. The MLP architecture, particularly effective among feed-forward neural networks, functions as a “black box” with hidden layers that enhance classification accuracy and minimize error. MLPs trained with metaheuristic algorithms have been found to overcome challenges of local minima, slow convergence, and lengthy initialization [24–30]. Metaheuristic-trained MLPs have also outperformed traditional gradient-based methods across diverse applications, including image recognition, natural language processing, and energy management [31–36]. The initial application of the GWO to train the MLP marked a pivotal point in improving training effectiveness [37]. Other research has demonstrated the effectiveness of metaheuristics like the lion optimization algorithm (LOA) [38], vortex search algorithm [32], and biogeography-based optimizer (BBO) [39] in optimizing MLPs. Such metaheuristic-based approaches have consistently shown higher global optimality compared to gradient methods [40].

In different research, various probability distributions were utilized to enhance the BBO (PD-BBO) [41] and to train the MLP, aiming to boost the precision of medical data classification. The distributions employed include gamma, beta, Gaussian, exponential, Poisson, geometric, Rayleigh, and Weber. These were integrated into the BBO algorithm's migration process to substitute the random distribution, and the migration probability was established. Recent research has applied swarm intelligence algorithms, including gray wolf optimizer, battle royale optimizer, sine cosine algorithm, and their improved versions, as well. These algorithms address the optimization task of determining network parameters to reduce errors during data processing. However, these algorithms often face challenges like local optima stagnation and inefficient search processes, limiting their effectiveness. In one of the studies, the competing leaders GWO (CGWO) were proposed to overcome these challenges [42].

The CGWO introduces competing leaders to create a dynamic wolf pack leadership structure, avoiding local optima and improving convergence. A population diversity-enhanced initiation method is developed to boost the competing leaders mechanism.

1.3 Novelty and Contributions

In recent years, the application of metaheuristic algorithms in agricultural data classification has witnessed a notable increase. These algorithms present considerable advantages over traditional methods due to their enhanced accuracy and stability when handling complex and multidimensional datasets. Nevertheless, recent literature reviews have identified persistent deficiencies in current metaheuristics, particularly concerning generalization, parameter adjustment, and explainability across diverse agricultural products and datasets [43]. Issues related to stability and accuracy remain, especially in large and unstable datasets, and research on real-time and scalable applications is still limited. The SEEDS dataset is frequently employed in numerous studies within this domain, serving as a standard reference for algorithm comparison. However, many existing studies inadequately address the generalizability and transferability of results to real-world applications, primarily due to the relatively small and balanced nature of the SEEDS dataset [44]. Furthermore, most studies are confined to basic machine learning or classical metaheuristics and insufficiently focus on critical issues such as algorithm stability, hyperparameter optimization, and explainability [45–47]. This study aims to achieve superior performance in terms of both accuracy and stability in agricultural data classification through a TSO-based approach and to discuss the generalizability of the results obtained, thereby addressing both the general research gaps in the literature and the deficiencies associated with the SEEDS dataset.

In the context of the above discussion, in this study, wheat types are classified using TSO [23]-optimized MLPs based on the SEEDS dataset [36] from the UCI machine learning repository. The foraging behaviors observed in tuna (spiral and parabolic movements) enable the TSO to balance exploration and exploitation, delivering high problem-solving efficiency with minimal control parameters. Previous studies in the food sector have shown that machine learning can enhance classification tasks, with notable applications using bagging and hard voting techniques for classifier optimization [44–46]. When MLPs were tested on this dataset using fivefold cross-validation, they achieved a remarkable classification rate of 99.5% [45]. Comparative studies have also explored K-nearest neighbors (KNN), classification and regression trees (CART), and Gaussian Naïve Bayes (NB) algorithms, finding that ensemble classifiers often achieve optimal accuracy through forced voting strategies [46]. Additional research using the SEEDS dataset demonstrated a deep neural network (DNN) model reaching 100% accuracy in classifying wheat types into groups like Rosa, Canadian, and Kama, underscoring the potential of ML-driven classification in agriculture [48].

This paper addresses the challenge of enhancing agricultural productivity in response to the increasing demand for food amidst diminishing arable land. A critical aspect of this endeavor is the rapid and precise classification of wheat varieties. Traditional methods are often time-consuming and prone to errors. Consequently, there is a growing adoption of machine learning and artificial intelligence in agriculture. However, the training of these AI systems is complex, requiring optimal configurations, and conventional approaches may not always yield satisfactory results. Innovative techniques, such as swarm intelligence algorithms like the TSO, have demonstrated superior efficacy in tackling complex problems. This paper employs the TSO algorithm in conjunction with MLP neural network to accurately classify wheat types. This approach contributes to the automation and reliability of agricultural practices. The findings indicate that this method holds promise for both research applications and practical farming operations.

The paper underscores the distinct benefits of combining the TSO algorithm with MLP neural networks for classifying wheat, distinguishing this method from traditional approaches. This combination leads to significantly improved classification accuracy and facilitates efficient, data-driven analysis, reducing errors often linked to manual observation. The TSO algorithm is particularly adept at navigating complex search spaces, optimizing the training process, and providing greater convergence stability than conventional methods. Due to its computational efficiency, the TSO-MLP framework is highly scalable and ideal for large-scale or real-time agricultural applications. Additionally, its adaptability indicates potential applicability to other crops, expanding its influence within the agricultural sector. By automating classification with high precision, the model is poised to enhance

productivity and elevate food quality standards, while also cutting labor costs and boosting operational efficiency for agricultural processors. The research exemplifies how metaheuristic-driven neural network optimization can propel AI-driven agriculture forward, surpassing other metaheuristic algorithms in both benchmark tests and practical classification tasks. Crucially, the study also considers ethical aspects, stressing responsible implementation and data privacy. By setting the stage for future adaptations and real-time deployment, this research introduces a promising path for improving agricultural productivity and the responsible use of artificial intelligence in the field.

This paper employs the CEC 2019 benchmark suite to evaluate the effectiveness of the TSO. The CEC 2019 suite encompasses a variety of test functions, including unimodal, multimodal, hybrid, and composition problems. These functions create complex landscapes with numerous local optima and variable interactions, complicating direct convergence to the global optimum. The suite effectively assesses the stability and robustness of optimization algorithms by testing their ability to avoid local minima and achieve global solutions. Despite the existence of newer benchmark sets, the CEC 2019 suite remains a respected standard for highlighting the strengths and weaknesses of metaheuristic algorithms. The choice to use CEC 2019 was influenced by its widespread use, the availability of comparative results, and its capacity to challenge the global search capabilities of optimization algorithms.

1.4 Structure of the Paper

The remainder of this paper is structured to comprehensively guide the reader through the methodology, analysis, and findings. In Sect. 2, the TSO is introduced in detail, with its mathematical foundations and behavioral modeling explained. This section also presents the algorithm's performance on the CEC 2019 benchmark suite, establishing its robustness in solving diverse optimization problems. Section 3 describes the experimental framework for applying the TSO algorithm to train an MLP model for the classification of wheat species using the SEEDS dataset. This section includes architectural details of the neural network, the rationale behind design choices, and a thorough comparison with other metaheuristic algorithms. Additionally, performance metrics such as accuracy, F1-score, ROC-AUC, and convergence behavior are reported and analyzed. Finally, Sect. 4 concludes the paper by summarizing the key findings, discussing practical implications in agricultural applications, highlighting limitations, and proposing directions for future research and algorithmic improvements.

2 Mathematical Model of TSO

As with all swarm-based metaheuristic algorithms, the TSO algorithm starts the optimization process by creating an initial set of candidate solutions. In Eq. (1), X_i denotes the initial random population and $rand$ denotes the random number between $[0, 1]$. UB and LB denote the upper and lower bounds, and NoP denotes the swarm size.

$$X_i = (rand[0, 1]) \times (UB - LB) + LB \quad i = 1, 2, \dots, NoP \quad (1)$$

The TSO algorithm creates a spiral structure when tuna, which adopts two behavioral patterns, spiraling and parabolic, search for prey. A small group of fish in the leader position, traveling consistently in a certain direction, ensures that the whole swarm joins them, and a spiral foraging strategy is formed with a sequential flow of information, the formulation of which is given in Eqs. (2)–(6).

$$X_i^{t+1} = \begin{cases} \alpha_1 \times \left(X_{best}^t + \beta \left| X_{best}^t - X_i^t \right| \right) + \alpha_2 \times X_i^t, & i = 1 \\ \alpha_1 \times \left(X_{best}^t + \beta \left| X_{best}^t - X_i^t \right| \right) + \alpha_2 \times X_{i-1}^t, & i = 2, 3, \dots, NoP \end{cases} \quad (2)$$

$$\alpha_1 = a + (1 - a) \times \frac{t}{t_{max}} \quad (3)$$

$$\alpha_2 = (1 - a) - (1 - a) \times \frac{t}{t_{max}} \quad (4)$$

$$\beta = e^{b \cdot l} \times \cos(2\pi b) \quad (5)$$

$$l = e^{3\cos(((t_{max}+1/t)-1)\pi)} \quad (6)$$

where X_i^{t+1} is the i -th individual of iteration $t+1$, X_{best}^t is the best available individual, a constant a is the parameter that gives a measure of the randomly selected tuna's tracking of the previous individual and the individual in the best position at the initial stage, α_1 and α_2 are weight coefficients and control parameters that evaluate the trend of this tracking. t is the current iteration, t_{max} is the maximum iteration and b is a random number in the interval $[0,1]$. Thanks to the exploration capability of the TSO, by placing the starting point (X_{rand}^t) for spiral search on a random coordinate in the search space, the best individual of the swarm, which forms a spiral structure around the food, creates the ability to overcome the problem in case of failure in finding food and enables it to search in a wider area. Equation (7) modeled the exploration ability mathematically.

$$X_i^{t+1} = \begin{cases} \alpha_1 \times \left(X_{rand}^t + \beta |X_{rand}^t - X_i^t| \right) + \alpha_2 \times X_i^t, & i = 1 \\ \alpha_1 \times \left(X_{rand}^t + \beta |X_{rand}^t - X_i^t| \right) + \alpha_2 \times X_{i-1}^t, & i = 2, 3, \dots, NP \end{cases} \quad (7)$$

Metaheuristic algorithms typically commence with an extensive global exploration phase, which gradually transitions into a focus on local areas, known as the exploitation phase. As iterations progress in TSO, this principle emphasizes the most optimal individuals by leveraging the foraging capacity of random agents during the exploration phase. This is achieved by moving in a spiral and capitalizing on the diversity of solutions generated in this phase. This phase is represented in Eq. (8). In Eq. (8), where $i = 2, 3, \dots, NP$ and $i = 1$.

$$X_i^{t+1} = \begin{cases} \begin{cases} \alpha_1 \times \left(X_{rand}^t + \beta |X_{rand}^t - X_i^t| \right) + \alpha_2 \times X_i^t, & \text{if } rand < \frac{t}{t_{max}} \\ \alpha_1 \times \left(X_{rand}^t + \beta |X_{rand}^t - X_i^t| \right) + \alpha_2 \times X_{i-1}^t, & \end{cases} \\ \begin{cases} \alpha_1 \times \left(X_{best}^t + \beta |X_{best}^t - X_i^t| \right) + \alpha_2 \times X_i^t, & \text{if } rand \geq \frac{t}{t_{max}} \\ \alpha_1 \times \left(X_{best}^t + \beta |X_{best}^t - X_i^t| \right) + \alpha_2 \times X_{i-1}^t, & \end{cases} \end{cases} \quad (8)$$

Tuna adopts a spiral architecture as well as a parabolic feeding pattern. Both types of feeding are simultaneous and have equal possibilities. In parabolic feeding, they form a parabola with food as the starting point, and each fish forages by searching around itself. Equation (9) and (10) provide a mathematical model of these two movements. Here TF is a random number with value 1 or -1 .

$$X_i^{t+1} = \begin{cases} X_{best}^t + rand \times (X_{best}^t - X_i^t) + TF \times p^2 \times (X_{best}^t - X_i^t), & \text{if } rand < 0.5 \\ TF \times p^2 \times X_i^t, & \text{if } rand \geq 0.5 \end{cases} \quad (9)$$

$$p = \left(1 - \frac{t}{t_{max}} \right)^{(t/t_{max})} \quad (10)$$

In the optimization process of TSO, an initial swarm of a specified number of agents is randomly generated. Each agent in the swarm randomly chooses one of the two search strategies and reevaluates and updates its position at each iteration. This process continues until the best satisfactory result is reached [23]. The pseudocode of the TSO is provided in Algorithm 1.

Algorithm 1 Pseudocode of TSO algorithm

1. Initialize

Input:

Initialize the swarm size (NoP) and maximum iteration (t_{max}).

Entry free parameters a and z

Output:

Position of the optimal solution (the best individual) and its corresponding fitness value.

2. Main optimization loop

$t \leftarrow 0$ 'set current iteration'

WHILE ($t < t_{max}$) 'calculate the fitness values of all tunas.'

DO

3. Fitness evaluation

FOR X_i^t

DO

X_{best}^t 'calculate fitness value',

END FOR

4. Update best position

FOR X_i^t 'each tuna'

DO

Update α_1, α_2, p coefficients 'adaptive behavior: α_1, α_2, p adjust through optimization'

$rand[0,1]$ 'Create a random value'

5. Position Up-to-date rules

IF ($rand_1 < z$)

THEN

**spiral foraging behavior //Controlled by parameter z and Eq. (1)

Update position X_i^{t+1} using Eq. (1)

ELSE IF ($rand_1 \geq z$)

THEN

Create a random value: $rand_2[0,1]$

IF ($rand_2 < 0.5$)

THEN

Create a random value: $rand_3[0,1]$

IF ($\frac{t}{t_{max}} < rand_3$)

THEN

IF ($t/t_{max} < rand_3$)

THEN

** Early exploration phase

Update position X_i^{t+1} using Eq. (7)

ELSE

** Late exploitation phase

Update position X_i^{t+1} using Eq. (2)

END IF

ELSE IF ($rand_2 \geq 0.5$)

THEN

** Random search behavior //Provides diversity through Eq. (9)

Update position X_i^{t+1} using Eq. (9)

END IF

END IF

END FOR

6. Increase iteration

$t = t + 1$

END WHILE

Return the best individual X_{best} and the best fitness value $F(X_{best})$.

3 Experimental Observations

3.1 Simulation of TSO via CEC 2019 Benchmark Test Set

The effectiveness of the TSO algorithm is evaluated through a series of simulations on the CEC 2019 benchmark function set, a widely used standard for assessing optimization algorithms due to its complex landscape. Table 1 details each function used in the evaluation, specifying function type, dimensionality, and search ranges. These benchmark functions, such as Rastrigin's and Ackley's, introduce challenging features like non-linearity and multimodality, ideal for testing the robustness and adaptability of metaheuristic algorithms like TSO.

Table 2 presents a comparative statistical analysis of TSO against alternative algorithms, including Archimedes optimization algorithm (AOA), prairie dog optimization (PDO), and Harris hawks optimization (HHO). For each function, key performance metrics (mean, standard deviation, best, and worst values) are recorded. As seen, TSO demonstrates competitive performance, notably achieving the lowest mean values for Function 01 (Storn's Chebyshev Polynomial) and Function 05 (Griewangk's Function), indicating strong convergence toward optimal solutions. The standard deviation values for TSO in these functions also remain relatively low, suggesting stability across multiple runs. For functions with high-dimensional landscapes, such as Function 07 (Modified Schwefel's Function), TSO outperforms the alternatives by balancing exploration and exploitation effectively. In Function 07 (Modified Schwefel's Function), TSO showed high standard deviation ($\sigma \approx 2.07 \times 10^2$), indicating significant solution variability between runs. This suggests that for highly multimodal landscapes, TSO's stochastic nature can lead to inconsistent convergence, potentially becoming trapped in local optima depending on initial conditions. Through repeated experiments on CEC 2019 functions with different random seeds and populations, results showed that while TSO achieves competitive mean performance, solution quality varies significantly for certain functions. This emphasizes the importance of considering both mean and variance when evaluating metaheuristic algorithms. As demonstrated in Table 2, the performance of the TSO algorithm on the CEC 2019 dataset indicates that the problem remains stable, yielding high-quality solutions within the complex and multidimensional search spaces it encompasses. This experiment underscores the TSO algorithm's efficacy as a robust optimizer, thereby establishing a strong foundation for subsequent theoretical and practical investigations.

Figure 1 provides a boxplot representation of the results across the benchmark functions, visually comparing the distribution of outcomes for TSO and other algorithms. TSO exhibits tighter, more compact distributions, particularly in functions with rugged landscapes, implying consistent performance and a reduced likelihood of extreme outliers. This consistency suggests that TSO maintains a stable solution quality across various optimization challenges, which is crucial for practical applications where reliability is essential.

3.2 Training of TSO-based MLP

The input and output properties of the dataset used in the selection of the MLP architecture are crucial. In this context, it is necessary to look at the characteristics of the SEEDS dataset. The SEEDS dataset from the UCI repository contains 70 samples each of the Kama, Rosa, and Canadian wheat varieties (<http://archive.ics.uci.edu/ml/datasets/seeds>). The dataset contains the features of images recorded on X-ray codec plates and is used for classification purposes in MLP training. Seed properties such as circumference, kernel length, kernel width, area, compactness, and asymmetry coefficient were analyzed. The input layer consists of seven sensors, and the features loaded on each sensor can be expressed as follows. For each grain of wheat, they are categorized as area (area, A), perimeter (perimeter, P), compactness (compactness, C), length, width, asymmetry coefficient, and length of a kernel groove [35].

In the MLP architecture, there are 7 sensors in the input layer, 22 sensors in the hidden layer, and 3 sensors in the output layer. The Sigmoid function is chosen as the transfer function since it is a non-linear structure,

Table 1 Details of CEC 2019 benchmark function problems

Function (CEC2019)	Definition	Dimension	Search range	F _{min}
01	Storn's Chebyshev Polynomial Fitting Problem	9	[− 8192, 8192]	1
02	Inverse Hilbert Matrix Problem	16	[− 16384, 16384]	1
03	Lennard–Jones Minimum Energy Cluster	18	[− 4, 4]	1
04	Rastrigin's Function	10	[− 100, 100]	1
05	Griewangk's Function	10	[− 100, 100]	1
06	Weierstrass Function	10	[− 100, 100]	1
07	Modified Schwefel's Function	10	[− 100, 100]	1
08	Expanded Schaffer's F6 Function	10	[− 100, 100]	1
09	Happy Cat Function	10	[− 100, 100]	1
10	Ackley Function	10	[− 100, 100]	1

and the error terms in the classification are evaluated with the MSE (mean-square error) criterion; the smaller the MSE value, the better the model. While the mathematical complexities of the TSO algorithm are detailed, it is acknowledged that the original manuscript did not adequately elucidate the rationale for specific decisions in the MLP architecture, particularly the choice of 22 hidden neurons and the employment of the sigmoid activation function. The number of hidden neurons in the MLP was determined empirically through experimentation. A range of hidden neuron counts, varying from 10 to 30, was examined, and the performance of each configuration was assessed on the SEEDS dataset using cross-validation. The configuration with 22 hidden neurons achieved an optimal balance between classification accuracy and model complexity, minimizing both underfitting and overfitting. This choice aligns with studies on similar datasets, where a moderate number of hidden units has proven effective for small to medium-sized classification problems. The sigmoid activation function was chosen due to its widespread use in classical MLP architectures, especially for problems with normalized input features, as in the SEEDS dataset. The sigmoid function provides non-linear transformations suitable for binary and multi-class classification tasks, facilitating gradient-based optimization and stable convergence. The use of sigmoid activation aligns with benchmark studies, enabling direct comparison of results.

An artificial neuron operating system is shown in Fig. 2 where S_j is the sigmoid transfer function, w_m is the weights, l_m is the sensors in each layer, O_i is the output. The difference between the desired output value (O_d) and the actual output value (O_a) is calculated by MSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (O_d - O_a)^2 \quad (11)$$

The process of training the SEEDS dataset under the supervision of the TSO algorithm and optimization of the data is also given in Fig. 3. As seen in the model presented in Fig. 3, the TSO algorithm, selected as the optimizer, produces an output for each data point in the dataset. The error between this output and the expected data obtained from training the MLP architecture represents the performance of the TSO-based MLP structure. As a result of this architecture set up, training was carried out under the supervision of metaheuristic algorithms.

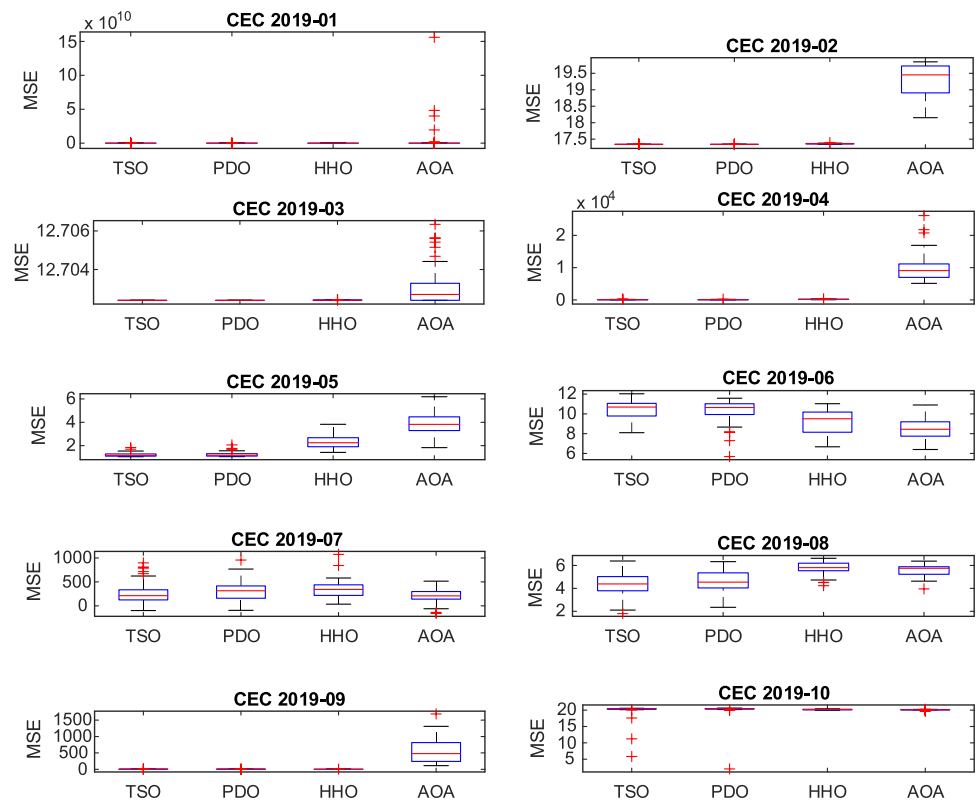
Permutation importance analysis has elucidated the model's capacity to identify features integral to the decision-making process. This approach ensures transparency and interpretability, thereby facilitating the identification of characteristics that are pivotal in the wheat classification process within agricultural applications. A permutation importance analysis was conducted on the test split (30%, with 30 shuffles per feature) to interpret the TSO-optimized MLP. Kernel Groove Length ($\Delta\text{Acc} \approx 0.086$) and Asymmetry Coefficient (≈ 0.054) emerged as the most influential predictors, whereas Compactness and Kernel Width contributed minimally ($\Delta\text{Acc} < 0$). Table 3 summarizes these results, offering practitioners insights into which grain characteristics drive the automated classification.

Table 2 Comparative statistical results on CEC2019 benchmark functions ($NoP = 40$, $t_{max} = 500$, $run = 50$)

Function (CEC 2019)	Measure	TSO	AOA	PDO	HHO
01	Mean	3.8580E + 04	3.1478E + 08	6.3740E + 04	5.1407E + 04
	Standard Deviation	1.4829E + 03	1.1821E + 09	2.2096E + 04	5.8520E + 03
	Best	3.6078E + 04	4.3721E + 05	4.3795E + 04	4.1794E + 04
	Worst	4.4900E + 04	6.1268E + 09	1.4828E + 05	7.1181E + 04
02	Mean	1.7343E + 01	1.9143E + 01	1.7736E + 01	1.7358E + 01
	Standard Deviation	3.6807E - 14	5.2253E - 01	4.5639E - 01	7.0000E - 03
	Best	1.7343E + 01	1.7986E + 01	1.7343E + 01	1.7344E + 01
	Worst	1.7343E + 01	1.9846E + 01	1.8998E + 01	1.7373E + 01
03	Mean	1.2702E + 01	1.2703E + 01	1.2702E + 01	1.2702E + 01
	Standard Deviation	8.9738E - 15	6.4469E - 04	8.9771E - 07	7.9164E - 06
	Best	1.2702E + 01	1.2702E + 01	1.2702E + 01	1.2702E + 01
	Worst	1.2702E + 01	1.2705E + 01	1.2702E + 01	1.2702E + 01
04	Mean	7.5662E + 01	1.0248E + 04	1.0849E + 04	1.6806E + 02
	Standard Deviation	4.1444E + 01	3.8387E + 03	3.9600E + 03	5.8746E + 01
	Best	1.3929E + 01	3.8533E + 03	5.2315E + 03	7.7682E + 01
	Worst	2.3480E + 02	2.2019E + 04	2.0013E + 04	3.3183E + 02
05	Mean	1.2171E + 00	3.7857E + 00	3.8741E + 00	2.3662E + 00
	Standard Deviation	1.6550E - 01	8.2346E - 01	6.4646E - 01	5.4070E - 01
	Best	1.0541E + 00	2.4242E + 00	2.6134E + 00	1.3986E + 00
	Worst	1.8405E + 00	5.4804E + 00	5.3203E + 00	3.4785E + 00
06	Mean	1.0419E + 01	8.9059E + 00	9.7044E + 00	9.0058E + 00
	Standard Deviation	8.9970E - 01	8.9017E - 01	1.6061E + 00	1.2324E + 00
	Best	8.0948E + 00	6.6080E + 00	3.8058E + 00	6.2786E + 00
	Worst	1.2031E + 01	1.0423E + 01	1.2242E + 01	1.1632E + 01
07	Mean	2.4558E + 02	1.8822E + 02	7.1688E + 02	3.8293E + 02
	Standard Deviation	2.0696E + 02	1.6137E + 02	1.9282E + 02	1.7316E + 02
	Best	-9.8666E + 01	-1.6368E + 02	2.3122E + 02	-2.8936E + 01
	Worst	8.1035E + 02	5.6504E + 02	1.0899E + 03	8.0038E + 02
08	Mean	4.4415E + 00	5.5320E + 00	6.1467E + 00	5.7542E + 00
	Standard Deviation	1.0655E + 00	6.8783E - 01	4.0442E - 01	5.7070E - 01
	Best	1.8019E + 00	3.0193E + 00	5.2092E + 00	4.3637E + 00
	Worst	6.3854E + 00	6.5104E + 00	6.8602E + 00	7.0500E + 00
09	Mean	2.4562E + 00	5.8276E + 02	1.2701E + 03	3.1015E + 00
	Standard Deviation	9.3500E - 02	3.4677E + 02	5.2851E + 02	3.7530E - 01
	Best	2.3491E + 00	6.9184E + 01	5.0165E + 02	2.4841E + 00
	Worst	2.7555E + 00	1.4416E + 03	2.8805E + 03	4.5670E + 00
10	Mean	1.9847E + 01	2.0113E + 01	2.0302E + 01	2.0208E + 01
	Standard Deviation	2.4590E + 00	4.5529E - 02	1.6466E - 01	1.0620E - 01
	Best	5.8248E + 00	1.9878E + 01	2.0014E + 01	2.0051E + 01
	Worst	2.0576E + 01	2.0175E + 01	2.0600E + 01	2.0468E + 01

The bold values represent the best values obtained

Figure 4 presents a two-dimensional t-SNE visualization of the hidden layer activations from a neural network trained on the SEEDS dataset. Each point on the plot represents a sample, with blue points indicating Class-1 (Kama), green points indicating Class-0 (Rosa), and red points indicating Class-1 (Canadian). Figure 4 reveals that the Class-1 (Canadian) samples are divided into two distinct subclusters. One of these subclusters is clearly separated from the other classes, while the other is mixed with the blue cluster where Class-1 (Kama) samples are located, resulting in significant overlap in this region. This indicates that some Class-1 samples overlap

Fig. 1 Boxplot representation of the CEC2019 test suite

considerably with Class-1 in the feature space, complicating the model's ability to distinguish these samples. Class-0 (Rosa) samples are clustered in a more distinct region compared to both Class-1 groups and do not exhibit significant overlap with the other classes. This explains the model's higher accuracy in identifying Class-0. Class-0 samples are better separated from the others. The overlap between Class-1 and Class-1 is a key factor contributing to the low precision and recall values observed for Class-1. The model's difficulty in clearly defining the boundary between these two classes leads to some Class-1 examples being misclassified as Class-1, and vice versa. Utilizing the two-dimensional t-SNE visualization, the activation of the hidden layer within the MLP network trained on the dataset is observable. This visualization facilitates a more detailed examination of the model by identifying the classes in which the model demonstrates greater success and potential points of confusion.

Visualizing the calibration process of the MLP is instrumental in enhancing comprehension of the model's training dynamics and parameter configuration. Graphical representation of the variations in training and validation losses across iterations facilitates the monitoring of the model's learning trajectory and the identification of phenomena such as overfitting or underfitting. Additionally, it aids in visually substantiating the impact of various hyperparameters, including learning rate, number of neurons, and batch size, on model performance. This approach fosters a deeper understanding of the model's calibration process and the rationale underlying the final parameter selections [49].

Figure 5 displays a heatmap that illustrates the validation accuracy of the MLP model across various combinations of learning rates and numbers of hidden neurons. The findings reveal that the highest validation accuracy occurs when the number of hidden neurons is set to 22 and the learning rate is 0.15. Deviations from these values, either lower or higher, tend to diminish the model's performance, indicating potential underfitting or overfitting. The optimal region, highlighted in yellow, underscores the necessity of carefully tuning these parameters to maximize classification accuracy. This analysis supports the empirical selection of hyperparameters in the study and underscores the importance of systematic optimization in neural network training.

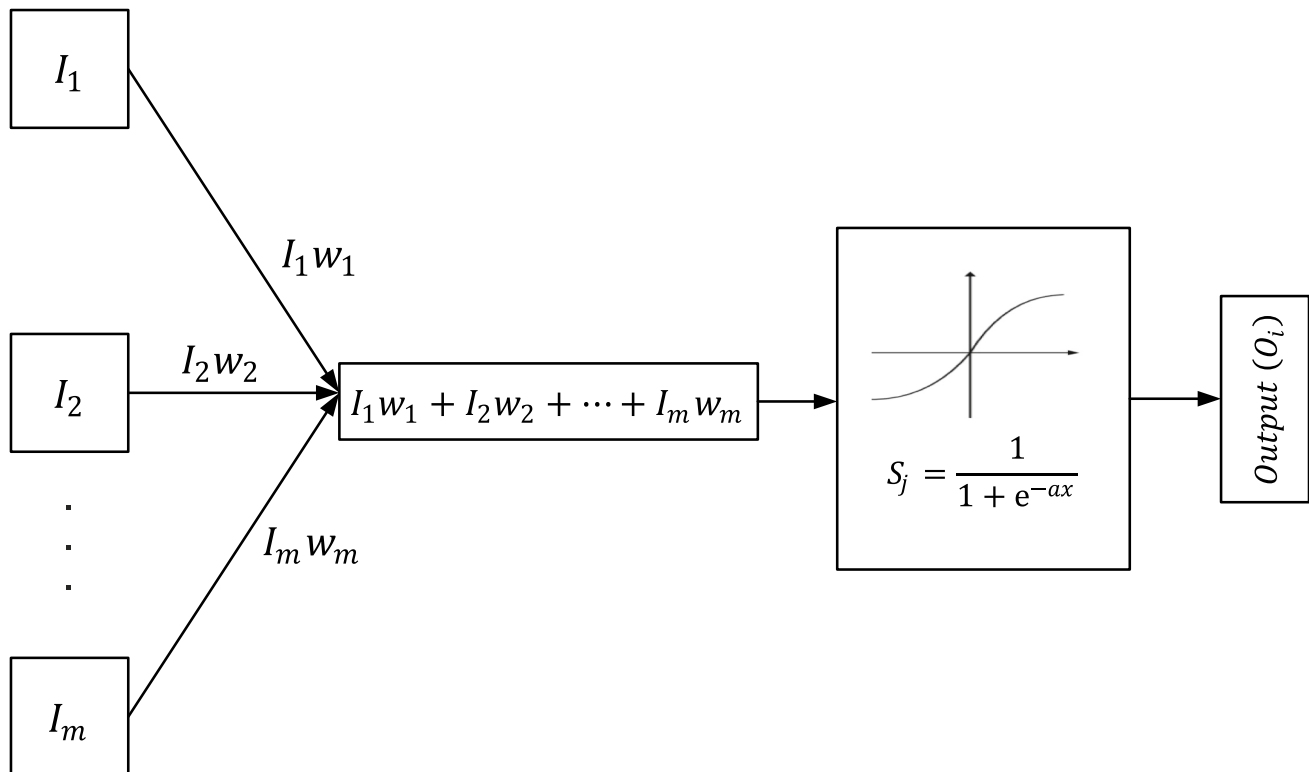


Fig. 2 Working system of artificial sensor

Archimedes optimization algorithms (AOA), capuchin search algorithm (CapSA), artificial gorilla troops optimizer (GTO), Harris hawks optimization (HHO), prairie dog optimization algorithm (PDO), and sine–cosine search algorithm (SCA) were selected as alternative algorithms to the TSO algorithm [49–54]. Mean value, standard deviation, and classification rate were chosen as statistical measures. The statistical measures were performed with 10 runs, 300 iterations, and 200 search agents. All experiments were conducted using MATLAB 2024 on a workstation equipped with an Intel Core i7, 16 GB RAM, and Windows 11. As can be seen in Table 4, the classification rate and average MSE values of the TSO algorithm were found to be better than those of the alternative algorithms. This result shows that the TSO algorithm has an effective strategy to increase efficiency and accuracy in training the network.

Convergence curves provide the reader with information about the local area, early convergence, and approaching the optimum point in the optimization of a problem. In this framework, Fig. 6, the iterative realization stages of the outcomes and the results are visually shown on the converge curve plot, and it is observed that the TSO algorithm gradually reaches the solution as a result of 300 iterations and achieves the most optimal value among the alternatives. Upon examining the performance of the CapSA and TSO algorithms over 0 to 300 iterations, it is observed that the TSO algorithm exhibits a smooth convergence, attaining the optimal value without any interruptions or delayed convergence. Similarly, the CapSA algorithm demonstrates effective convergence; however, it encounters premature convergence after the 150th iteration, stabilizing at a suboptimal point.

As shown in Fig. 7, the confusion matrix was constructed to evaluate the performance of the TSO classification algorithm utilized in this study. There are four terms in the matrix. These give the number of correctly and incorrectly classified data. To find *Accuracy*, *Precision*, *Recall* results, the data are obtained from the confusion matrix. These data are True Positive (TP), i.e., both the desired and actual value is positive, False Positive (FP), i.e., the result is positive while the desired value is negative, True Negative (TN), i.e., both

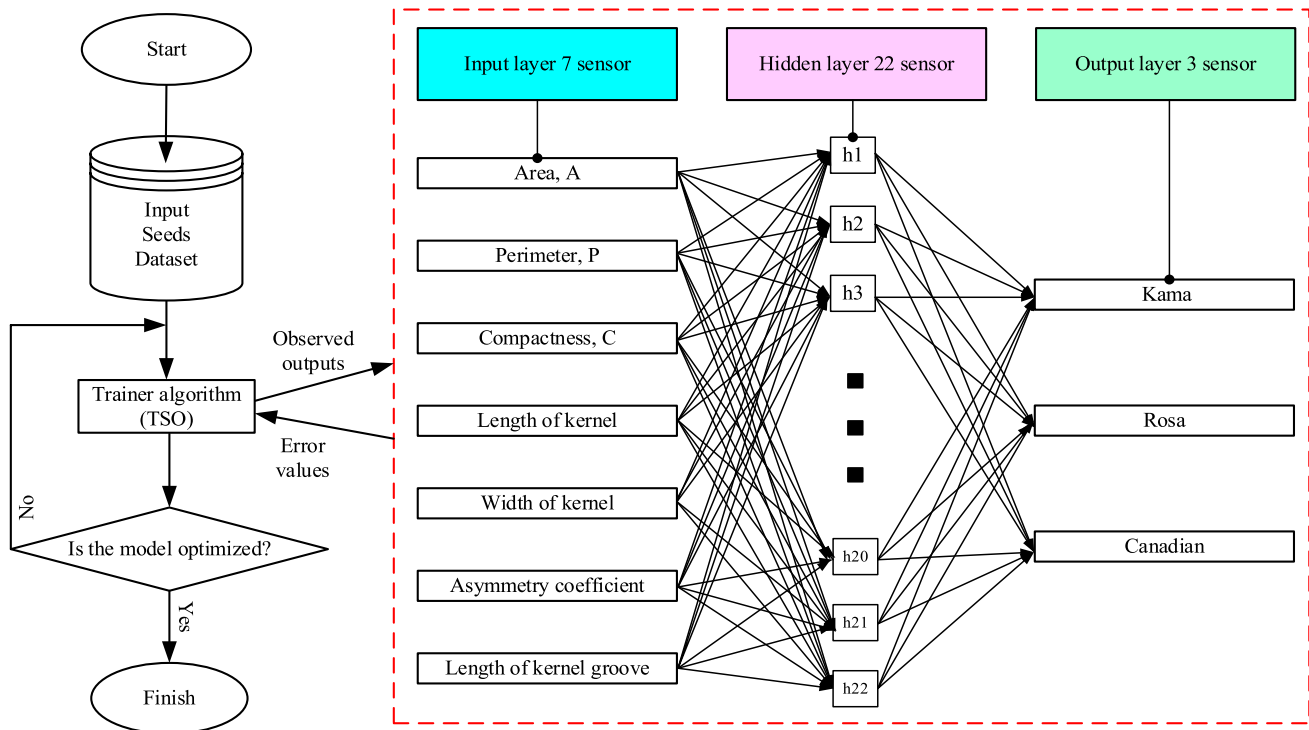


Fig. 3 Training of TSO-based MLP

Table 3 Permutation importance analysis

Rank	Feature	Mean Δ -accuracy	Std. Dev
1	Kernel Groove Length	0.0862	0.0366
2	Asymmetry Coefficient	0.0540	0.0347
3	Kernel Length	0.0429	0.0304
4	Perimeter	0.0132	0.0284
5	Area	0.0101	0.0267
6	Kernel Width	−0.0016	0.0250
7	Compactness	−0.0175	0.0237

the desired and actual value is negative, False Negative (FN), the result is negative while the desired value is positive. Here the FN value is also referred to as error II [48]. *Accuracy* gives the proportion of correctly classified positives out of the whole sample, *Precision* gives the ratio of the number of correctly classified positives to the total number of positives, and *Recall*, as a measure of *Precision*, gives the ratio of the number of correctly classified positives to the sum of the number of correctly classified positives and the number of incorrectly classified negatives. The *F1 – score* measures the overall performance of the classification model by taking all these measures into account. The F1-score is the harmonic mean of *Accuracy* and *Recall*. It is a significant measure to see the strength of the balance between these two values [55].

$$Accuracy = \frac{TP + TN}{Number\ of\ Instance} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

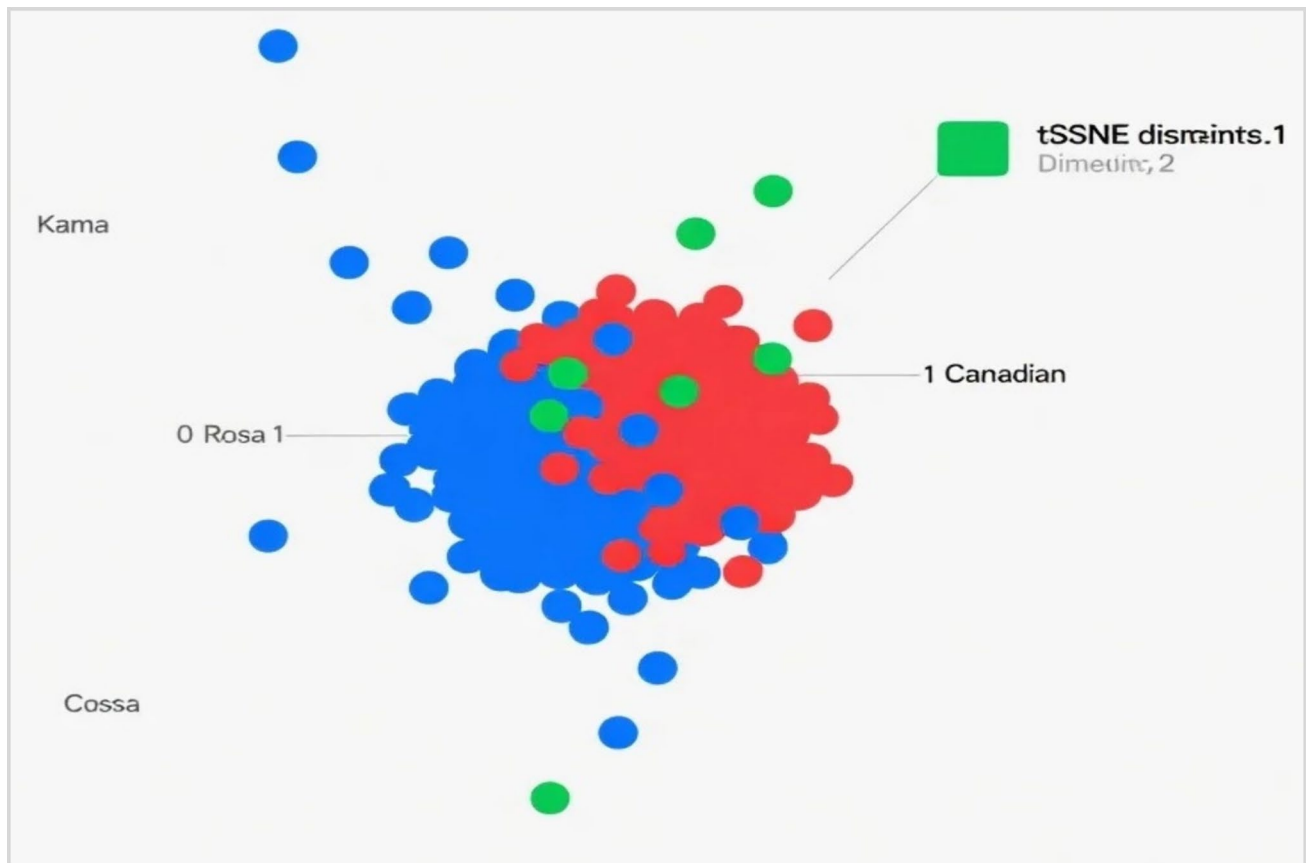


Fig. 4 t-SNE visualization of MLP latent space

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

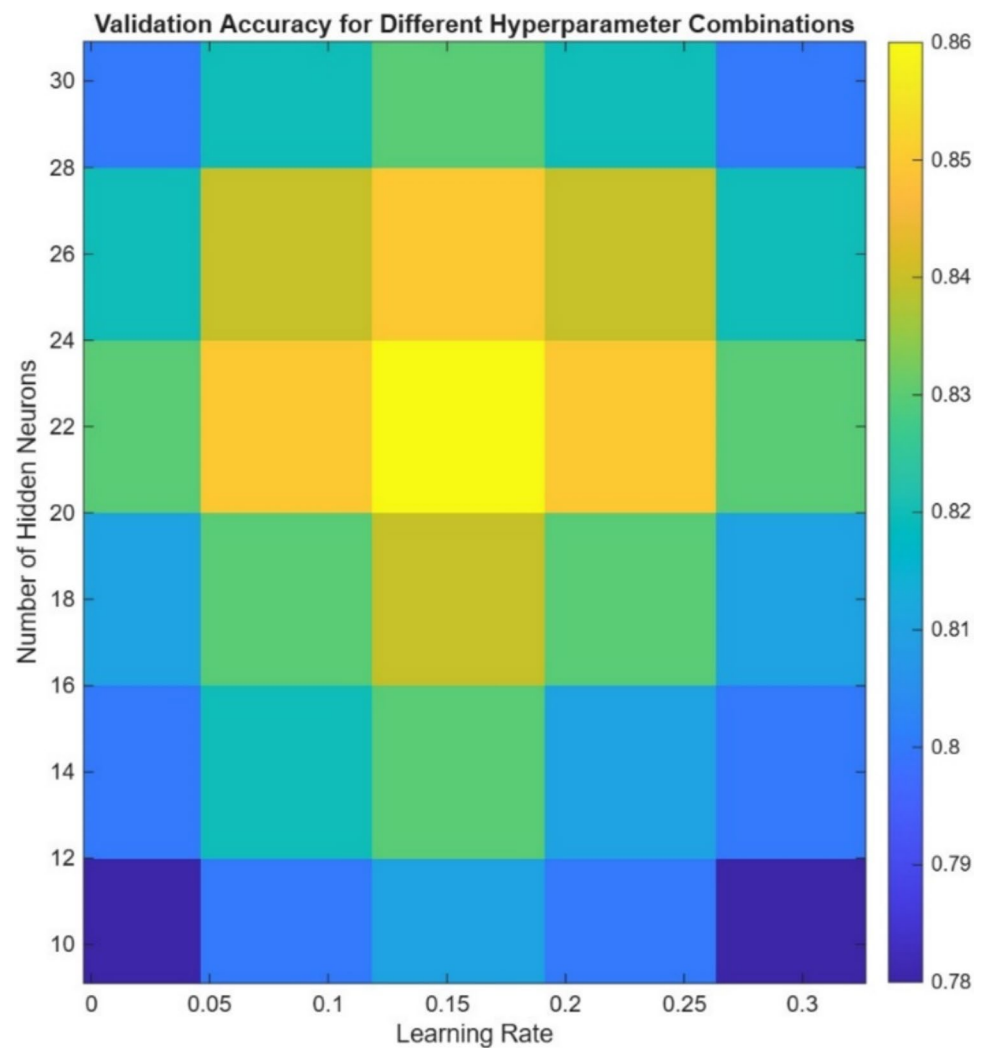
$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (15)$$

Statistical result of TSO based MLP model is shown in Table 5. When evaluated in the light of these data, it will be seen that a good balance is achieved between Precision and Recall criteria in class 0. It can be assumed that the balance is not bad for the other two classes. In the F1-score value, since the samples are equally distributed for all classes, Macro F1-score and Weighted F1-score are equal, and the result is 0.7923. This F1-score value indicates that the classification success of the model presented in the study is 79.29%.

ROC analysis is a plot that reveals the classification ability between correct and incorrect predicted values. It is an important criterion to reveal the discrimination power of the test and the experimental quality of the applied model [55]. Area Under the Curve (AUC) refers to the area under the ROC curve. The size of the area is directly proportional to the quality of the test. In other words, the larger the area, the higher the test success [56]. In Fig. 8, three classes were evaluated in this study, and it is observed that the rates of the three classes are higher than the threshold value (0.5), which is read as classification success, but especially the classification success of the 0 class is high. It is observed in Fig. 8 that the test success is 0.7727 for the -1 class, 0.9851 for the 0 class, and 0.7864 for the 1 class. According to these results, it can be observed that the model is highly successful.

Time complexity measures a computer's code execution time through operation count. Big O notation expresses an algorithm's worst-case time complexity algebraically. The algorithm maintains "tunas"

Fig. 5 Validation accuracy for different hyperparameter combinations of MLP



representing neural network weights, evaluating fitness by calculating MLP forward loss over iterations. Key parameters are population size (N), search space dimensionality (D), and iterations (I). The main computational cost comes from candidate fitness evaluation via MLP forward propagation, scaling with $N \times D$. Position updates and optimal solution identification scale with $N \times D$ and N . The overall computational complexity is $O(N \cdot D \cdot I)$, increasing linearly with population size, network weights, and optimization iterations. This linear scaling makes the algorithm predictable, though larger networks increase computational demands. Figure 9 shows TSO's theoretical time complexity across population sizes (50, 100, 150, 200) as iterations increase, with $D = 168$. Operations increase linearly with iterations per population size. The evenly spaced lines show population size proportionally affects computational cost. The plot confirms TSO's linear scaling behavior regarding population size and iteration count, important for large-scale implementations.

4 Discussion

In this paper, we aimed to automatically classify wheat species using a MLP architecture, which was trained with the TSO algorithm. The results reveal that the TSO-based MLP model achieves superior accuracy compared to other metaheuristic algorithms, as demonstrated by its performance on both the CEC 2019 benchmark functions and the SEEDS dataset. The model's classification effectiveness, measured by F1-score, ROC, and AUC, indicates

Table 4 Statistical results for training of MLP

Algorithm	Metric	Results	Rank
TSO	Mean	7.5201E−02	1
	Standard Deviation	7.4820E−02	1
	Classification Rate	7.9524E+01	1
AOA	Mean	3.0480E−01	7
	Standard Deviation	5.4900E−02	6
	Classification Rate	2.3333E+01	7
CapSA	Mean	1.8810E−01	6
	Standard Deviation	1.6270E−01	5
	Classification Rate	3.3333E+01	7
GTO	Mean	1.3100E−01	3
	Standard Deviation	1.1980E−01	4
	Classification Rate	7.3333E+01	3
HHO	Mean	1.8410E−01	5
	Standard Deviation	1.0820E−01	2
	Classification Rate	4.7619E+01	6
PDO	Mean	1.4350E−01	2
	Standard Deviation	9.5500E−02	7
	Classification Rate	5.8571E+01	4
SCA	Mean	2.2090E−01	6
	Standard Deviation	2.7100E−02	2
	Classification Rate	6.0952E+01	3

The bold values represent the best values obtained

that this approach is effective for classifying agricultural data. The findings show that the TSO algorithm can generate solutions in complex search spaces and address traditional challenges in MLP training. Permutation significance analysis was used to identify the most influential features, thereby enhancing model transparency. The model's generalizability was further strengthened through hyperparameter optimization and cross-validation. However, certain limitations exist. The SEEDS dataset is relatively small and balanced, and the model's performance on larger, real-world datasets remains untested. The TSO algorithm's inherent randomness and tendency to become trapped in local minima can affect solution quality in complex problems, as indicated by the high standard deviations observed in the CEC 2019 tests. Testing under diverse conditions and with different wheat species is crucial for ensuring generalizability. Future research could involve applying the TSO-MLP model to various crops and larger datasets, refining the model with hybrid TSO variants, and integrating it into real-time applications. Enhancing the transparency of the model's decision-making processes, while considering ethical implications, is essential to benefit small-scale farmers. In conclusion, this study underscores the potential of meta-heuristic algorithms in agricultural machine learning and demonstrates that the TSO-based MLP model offers a promising approach for both academic and practical applications.

5 Conclusion

In this paper, the optimization of an MLP model using the TSO algorithm was first evaluated to enhance wheat species classification. Our findings demonstrate that TSO offers robust performance in navigating complex search spaces, as evidenced by its competitive results on the CEC 2019 benchmark functions. Compared to alternative metaheuristic algorithms, TSO achieved lower mean values and demonstrated stability across multiple runs, particularly in high-dimensional and multimodal optimization problems. The TSO-based MLP model also outperformed other configurations in classification accuracy, as assessed by metrics such as MSE, F1-score, ROC,

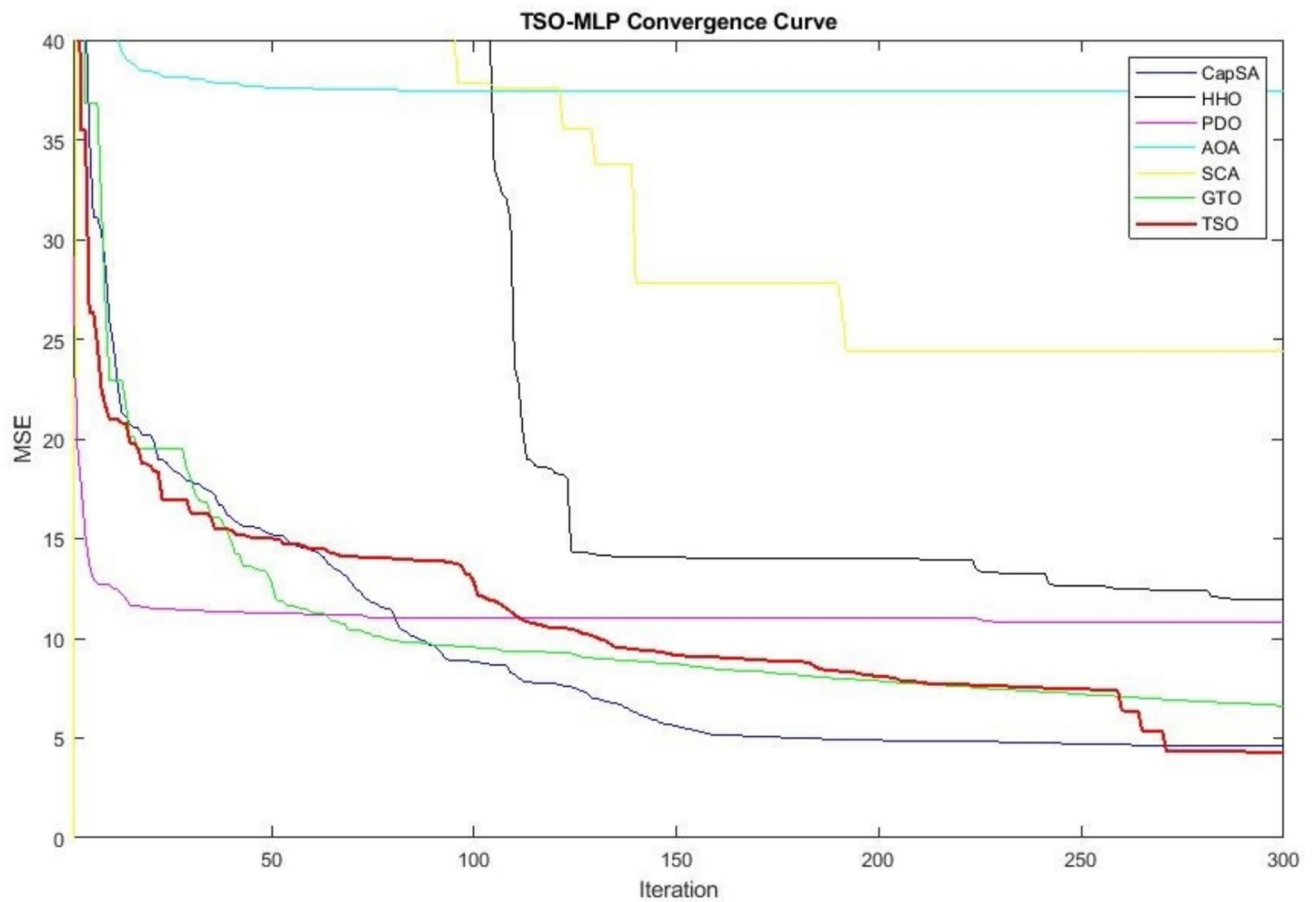


Fig. 6 Converge curve of MLP training

Fig. 7 Confusion matrix

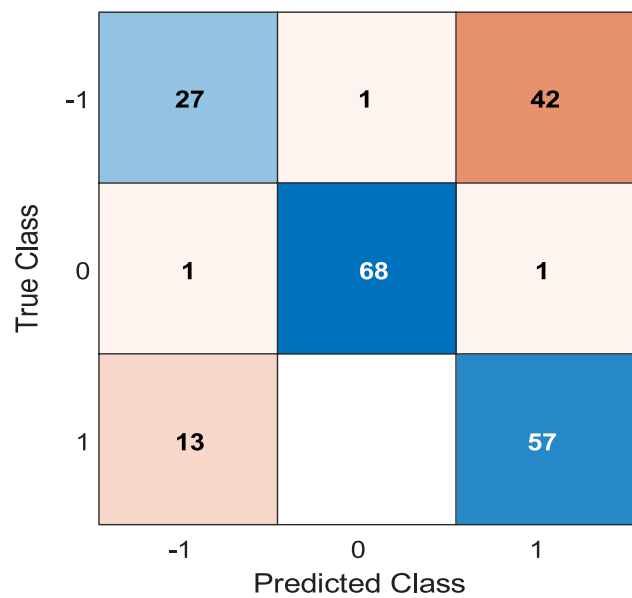
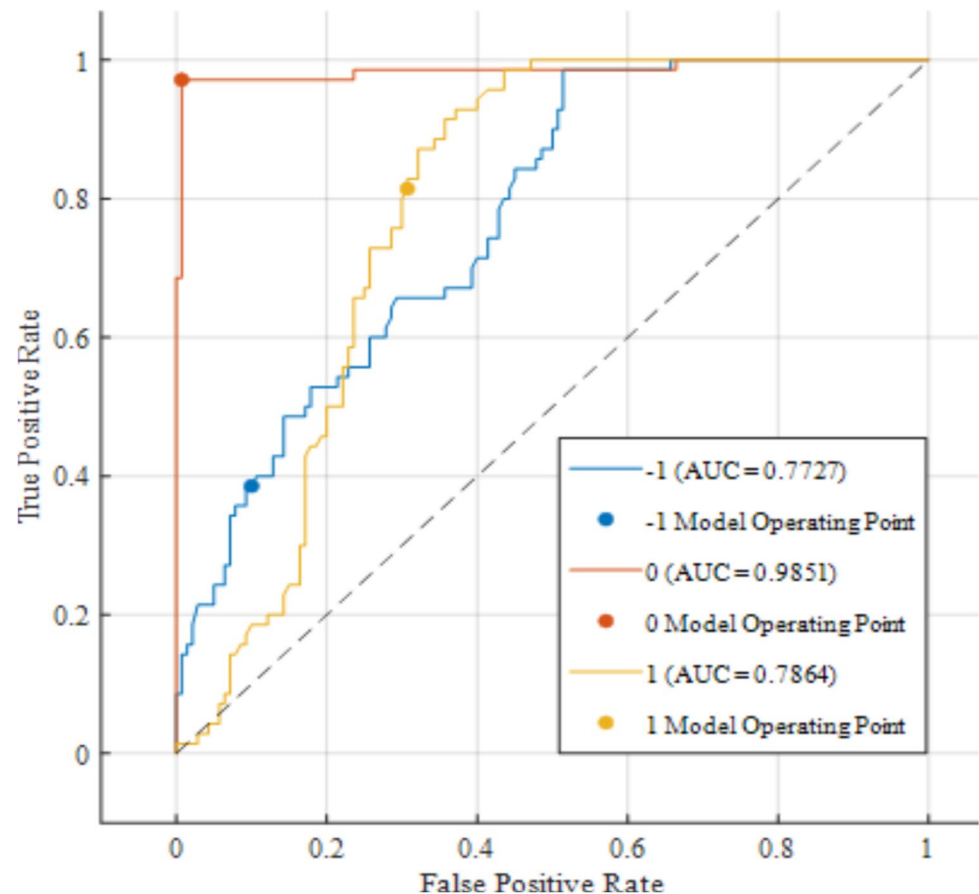


Table 5 Statistical result of TSO based MLP model

Statistical results	Classes		
	-1	0	1
<i>Accuracy</i>	0.7285	0.9857	0.7333
<i>Precision</i>	0.6585	0.9855	0.5700
<i>Recall</i>	0.3857	0.9714	0.8143
<i>F1 – score</i>	0.7280	0.9784	0.6706
Macro F1-score	Weighted F1-score		
0.7923	0.7923		

Fig. 8 ROC curve and AUC


and AUC, underlining its effectiveness in achieving high-quality classification outcomes. These values were optimized with cross-validation and comply with the recommended intervals in the literature. The implications of this work extend beyond model accuracy. Employing TSO to optimize MLP architectures for the SEEDS dataset has introduced a potentially valuable approach for enhancing classification in agricultural contexts. Accurate wheat classification can enhance productivity and streamline quality control processes, contributing to food security considering growing global demand and limited agricultural land.

Second, the study presented in this paper offers significant insights into the effects and costs for farmers, as well as issues of interpretability and ethics, which will aid in addressing practical challenges. To succinctly examine these topics, the implementation of automation and precision in wheat classification can assist farmers in optimizing their product selections, enhancing quality control, and potentially increasing market value, thereby improving economic outcomes. By reducing the necessity for manual inspection, the model can decrease labor

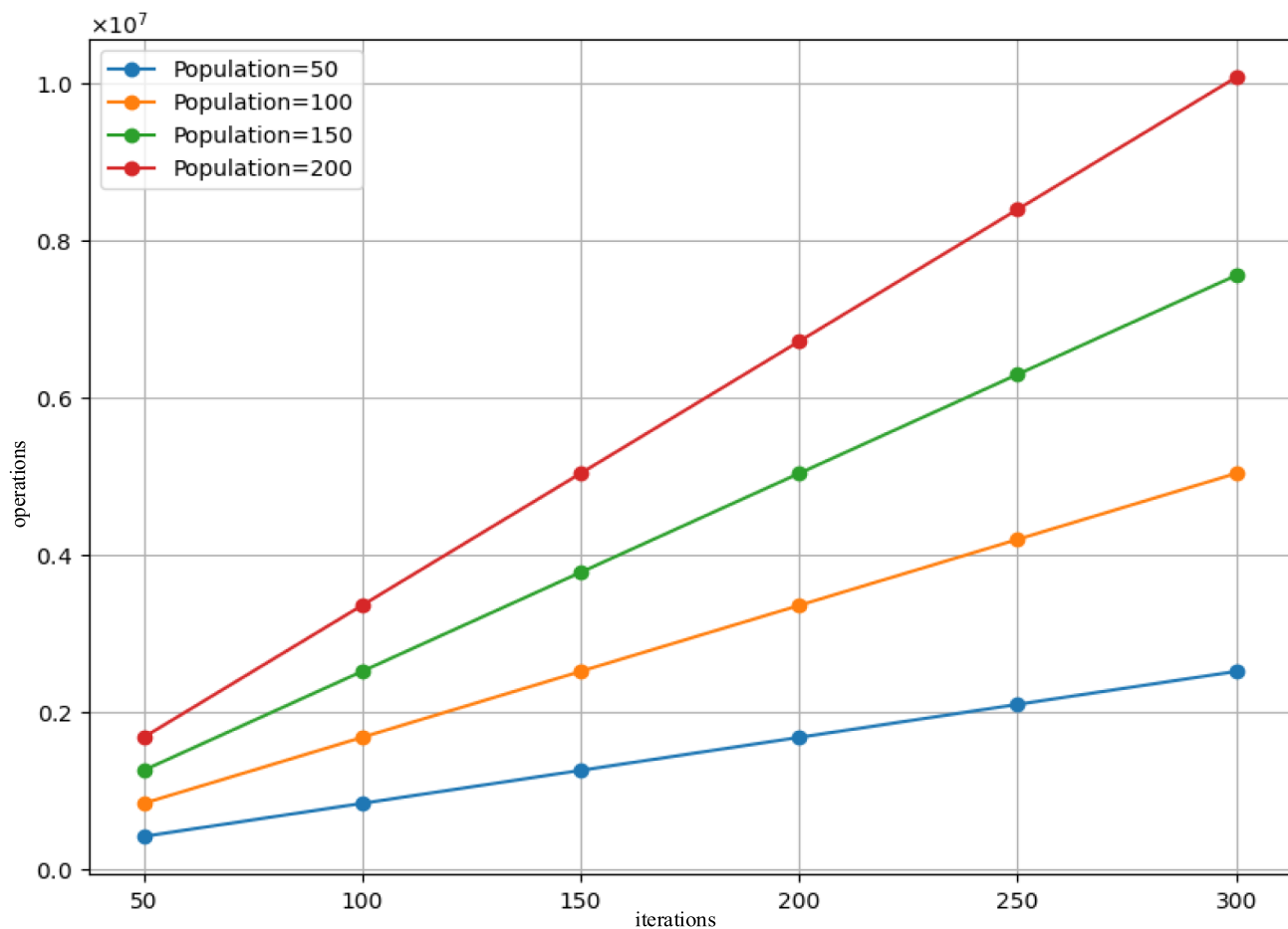


Fig. 9 Big-O notations

costs and minimize human error, making the classification process more scalable and accessible, particularly in resource-limited regions.

Cost and Efficiency: The TSO-MLP model's high accuracy and computational efficiency, as evidenced by its rapid convergence and low MSE, can facilitate real-time application, reduce operational expenses, and encourage its widespread adoption in agricultural cooperatives or processing facilities. **Interpretability:** Although neural networks are frequently perceived as "black boxes," employing confusion matrices and ROC curves can provide stakeholders with insights into model decisions, fostering transparency and confidence in AI-driven solutions.

Ethical considerations: Emphasizing the importance of data privacy, fairness, and transparency in deploying AI models in agriculture ensures that technological advancements benefit all stakeholders equally. It also underscores the necessity for continuous monitoring to prevent unintended bias and ensure that smallholder farmers are not disadvantaged by the implementation of automated systems.

Although the paper does not explicitly discuss the limitations of the TSO-MLP model in agricultural applications, certain issues can be identified. Enhancing the model's efficacy could be achieved by acquiring comprehensive, high-quality training data for various wheat species across different regions. Due to the framework's linear time complexity, it can be effectively utilized in large-scale applications, provided the necessary infrastructure is available. Further research to ensure the model's robust performance under diverse environmental conditions or with untested wheat species will improve its adaptability. Integrating the model into existing agricultural workflows, along with advancements in innovative technologies, may lead to increased efficiency and automation of processes. Efforts to enhance transparency will make the model's decisions more understandable, thereby

strengthening stakeholders' trust in the process. Alongside the adoption of AI-based solutions, addressing ethical considerations such as data privacy and support for small-scale farmers will ensure the technology's fairness and benefit to all stakeholders. Finally, regular monitoring and updates to maintain the model's accuracy and mitigate potential biases will sustain the system's reliability and effectiveness over the long term.

While the TSO algorithm has proven to be effective, it continues to face certain challenges, particularly due to its excessive randomness and propensity to become trapped in local optima. This issue is evident in the CEC 2019 benchmark results, where TSO exhibited a relatively high standard deviation in Function 07 (Modified Schwefel's function), indicating inconsistent performance across different runs. Such variability suggests that the algorithm's stochastic nature can hinder stable convergence, especially in complex, high-dimensional environments. To address these challenges, several enhanced TSO variants have been developed. For instance, Q-learning-based TSO (QLTSO) integrates reinforcement learning to guide the search process, thereby reducing randomness and improving the algorithm's ability to avoid local optima. Similarly, the circular chaos map and levy flight TSO (CLTSO) enhance the balance between exploration and exploitation through chaotic mapping and nonlinear adaptation, resulting in more consistent and precise solutions. Comparative analyses have shown that these advanced variants outperform the standard TSO in terms of both convergence stability and solution quality, particularly for challenging benchmark functions. Thus, while the original TSO provides a solid foundation, adopting hybrid or improved versions may further mitigate its inherent limitations.

Future research could focus on adapting this hybrid TSO-MLP framework to other crops or agricultural products, investigating the versatility of this approach across different classification datasets. Additionally, improvements to the TSO algorithm itself, such as integrating reinforcement learning or further hybridization with other optimization techniques, could yield even higher accuracy rates and efficiency in real-time applications. In functions where the TSO algorithm shows high variance, it is recommended to try hybrid or enhanced TSO variants for more stable results. Overall, this work sets the stage for further advancements in metaheuristic-based machine learning solutions within agriculture, supporting a shift toward more efficient, cost-effective, and scalable classification models. Despite the SEEDS dataset being relatively small and balanced, its use in this study provides a clear and controlled environment for assessing the proposed method. This arrangement allows for a concentrated evaluation of the method's effectiveness under specific conditions. As a well-known benchmark in the literature, the SEEDS dataset enables direct comparisons with past research and establishes a strong basis for future investigations. The encouraging results from the SEEDS dataset suggest that the proposed method has significant potential for application to more complex, larger, and imbalanced agricultural datasets. Future research will build on these results by extending the evaluation to such datasets and investigating advanced techniques—such as class weighting, resampling, and data augmentation—to further improve the method's robustness and scalability in real-world situations.

Author Contributions Beytullah Yağız, Erdal EKER, Yener Altun, Şeyma Nur Atar: Conceptualization, Methodology, Software, Visualization, Investigation, Writing- Original draft preparation. Davut Izci, Serdar Ekinici: Data curation, Validation, Supervision, Resources, Writing—Review & Editing. Mohit Bajaj, Mohammad Shabaz: Project administration, Supervision, Resources, Writing—Review & Editing.

Funding This work received no funding.

Data Availability The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Declarations

Conflicts of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as

long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

1. Chang, V., Bhavani, V.R., Xu, A.Q., Hossain, M.: An artificial intelligence model for heart disease detection using machine learning algorithms. *Healthc. Anal.* **2**, 100016 (2022)
2. Ma, L., Liu, X., Gao, F., Lee, K.Y.: Data-driven iterative learning model predictive control with self-modified prior knowledge. *IEEE Trans. Neural. Netw. Learn. Syst.* (2024). <https://doi.org/10.1109/TNNLS.2024.3453380>
3. Liu, X., Huang, H., Xiang, J.: A personalized diagnosis method to detect faults in gears using numerical simulation and extreme learning machine. *Knowl.-Based Syst.* **195**, 105653 (2020)
4. Izci, D.: An enhanced slime mould algorithm for function optimization. In: 2021 3rd international congress on human-computer interaction, optimization and robotic applications (HORA), pp. 1–5. IEEE (2021)
5. Malekshahi Rad, M., Rahmani, A.M., Sahafi, A., Nasih Qader, N.: Social internet of things: vision, challenges, and trends. *Hum.-centric Comput. Inf. Sci.* **10**, 52 (2020)
6. Fan, M., Zhang, X., Hu, J., Gu, N., Tao, D.: Adaptive data structure regularized multiclass discriminative feature selection. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 5859–5872 (2022)
7. Abualigah, L., et al.: A survey of the manta ray foraging optimization algorithm. In: *Metaheuristic optimization algorithms*, pp. 193–203. Elsevier, Amsterdam (2024). <https://doi.org/10.1016/B978-0-443-13925-3.00005-4>
8. Abualigah, L., et al.: A review of mothflame optimization algorithm: analysis and applications. In: *Metaheuristic optimization algorithms*, pp. 205–219. Elsevier, Amsterdam (2024). <https://doi.org/10.1016/B978-0-443-13925-3.00006-6>
9. Rizk-Allah, R.M., Snášel, V., Izci, D., Ekinci, S.: Spiral-refraction mutation prairie dog algorithm: optimization framework for engineering design of interconnected multimachine power system. *Appl. Soft Comput.* **165**, 112036 (2024)
10. Ekinci, S., Izci, D.: Pattern search ameliorated arithmetic optimization algorithm for engineering optimization and infinite impulse response system identification. *ELECTRICA* **24**, 119–130 (2024)
11. Bacanin, N., et al.: Training multi-layer perceptron with enhanced brain storm optimization metaheuristics. *Comput. Mater. Contin.* **70**, 4199–4215 (2022)
12. Kumari, M.: A review on metaheuristic algorithms: recent and future trends. In: *Metaheuristics-based materials optimization*, pp. 103–128. Elsevier, Amsterdam (2025). <https://doi.org/10.1016/B978-0-443-29162-3.00004-6>
13. Zamani, H., Nadimi-Shahraki, M.H., Gandomi, A.H.: QANA: quantum-based avian navigation optimizer algorithm. *Eng. Appl. Artif. Intell.* **104**, 104314 (2021)
14. Suman, B., Kumar, P.: A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Oper. Res. Soc.* **57**, 1143–1160 (2006)
15. Eker, E., Kayri, M., Ekinci, S., İzci, D.: Comparison of swarm-based metaheuristic and gradient descent-based algorithms in artificial neural network training. *Adv. Distribut. Comput. Artif. Intell. J.* **12**, e29969 (2023)
16. Holland, J.H.: Genetic algorithms. *Sci. Am.* **267**, 66–73 (1992)
17. Shami, T.M., et al.: Particle swarm optimization: a comprehensive survey. *IEEE Access* **10**, 10031–10061 (2022)
18. Yang, X.-S.: Firefly algorithms for multimodal optimization, pp. 169–178. Springer, Cham (2009). https://doi.org/10.1007/978-3-642-04944-6_14
19. Bacanin, N., et al.: A novel multiswarm firefly algorithm: an application for plant classification. *Int. Conf. Intell. Fuzzy Syst.* (2022). https://doi.org/10.1007/978-3-031-09173-5_115
20. Joshi, A.S., Kulkarni, O., Kakandikar, G.M., Nandedkar, V.M.: Cuckoo search optimization- a review. *Mater. Today Proc.* **4**, 7262–7269 (2017)
21. Yang, X.-S., Karamanoglu, M., He, X.: Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng. Optim.* **46**, 1222–1237 (2014)
22. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
23. Xie, L., et al.: Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. *Comput. Intell. Neurosci.* **2021**(1), 9210050 (2021)
24. Suratgar, A.A., Tavakoli, M.B., Hoseinabadi, A.: Modified Levenberg-Marquardt method for neural networks training. *World Acad. Sci. Eng. Technol.* **6**, 46–48 (2005)
25. Sheeja, R., Iqbal, M.M., Sivasankar, C.: Multi-objective-derived energy efficient routing in wireless sensor network using adaptive black hole-tuna swarm optimization strategy. *Ad Hoc Netw.* **144**, 103140 (2023)

26. Lei, X., et al.: Beyond visual range maneuver intention recognition based on attention enhanced tuna swarm optimization parallel BiGRU. *Complex Intell. Syst.* **10**, 2151–2172 (2024)
27. Wang, W., Tian, J.: Modified tuna swarm optimization algorithm using levy flight. In: Imane, H. (ed.) *International conference on computer vision, application, and algorithm (CVAA 2022)*, vol. 17. SPIE (2023)
28. Brindha, S., Bhargavi, K.V., Kamali, T.: Optimal cluster minimization for vanets using modified tuna swarm optimization. In: *2023 7th international conference on trends in electronics and informatics (ICOEI)*, pp. 1389–1393. IEEE (2023). <https://doi.org/10.1109/ICOEI56765.2023.10126028>
29. Zhou, W., Wang, L., Han, X., Wang, Y.: An adaptive density spatial clustering algorithm based on tuna swarm optimization for image segmentation. In: Yue, Y. (ed.) *International conference on electronic information engineering and computer science (EIECS 2022)*, vol. 26. SPIE (2023)
30. Chen, M., Du, W.: Dynamic relationship network and international management of enterprise supply chain by particle swarm optimization algorithm under deep learning. *Expert. Syst.* **41**, e13081 (2024)
31. Ekinici, S., et al.: Hunger games pattern search with elite opposite-based solution for solving complex engineering design problems. *Evol. Syst.* **15**, 939–964 (2024)
32. Wu, D., Foong, L.K., Lyu, Z.: Two neural-metaheuristic techniques based on vortex search and backtracking search algorithms for predicting the heating load of residential buildings. *Eng. Comput.* **38**, 647–660 (2022)
33. Guo, Z., Moayedi, H., Foong, L.K., Bahiraei, M.: Optimal modification of heating, ventilation, and air conditioning system performances in residential buildings using the integration of metaheuristic optimization and neural computing. *Energy Build.* **214**, 109866 (2020)
34. Chong, H.Y., Yap, H.J., Tan, S.C., Yap, K.S., Wong, S.Y.: Advances of metaheuristic algorithms in training neural networks for industrial applications. *Soft. Comput.* **25**, 11209–11233 (2021)
35. 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: Mirjalili, S., Song Dong, J., Lewis, A. (eds.) *Studies in computational intelligence*, vol. 811, pp. 23–46. Springer, Cham (2020)
36. Blake, C. L. & Merz, C. J.: UCI Repository of machine learning databases. University of California <http://archive.ics.uci.edu/ml/> <http://archive.ics.uci.edu/ml/> (1998).
37. Mirjalili, S.: How effective is the grey wolf optimizer in training multi-layer perceptrons. *Appl. Intell.* **43**, 150–161 (2015)
38. Bansal, P., Gupta, S., Kumar, S., Sharma, S., Sharma, S.: MLP-LOA: a metaheuristic approach to design an optimal multilayer perceptron. *Soft comput* **23**, 12331–12345 (2019)
39. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Let a biogeography-based optimizer train your multi-layer perceptron. *Inf. Sci.* **269**, 188–209 (2014)
40. Ahmed, A.N., et al.: A comprehensive comparison of recent developed meta-heuristic algorithms for streamflow time series forecasting problem. *Appl. Soft Comput.* **105**, 107282 (2021)
41. Li, X.-D., Wang, J.-S., Hao, W.-K., Wang, M., Zhang, M.: Multi-layer perceptron classification method of medical data based on biogeography-based optimization algorithm with probability distributions. *Appl. Soft Comput.* **121**, 108766 (2022)
42. Yang, Z.: Competing leaders grey wolf optimizer and its application for training multi-layer perceptron classifier. *Expert Syst. Appl.* **239**, 122349 (2024)
43. Nayak, G.H.H., et al.: Meta-transformer: leveraging metaheuristic algorithms for agricultural commodity price forecasting. *J. Big Data* **12**, 138 (2025)
44. Çelik, A.: Determination of the classification success of KNN algorithm distance metric methods on wheat seeds dataset. *Afyon Kocatepe Univ. J. Sci. Eng.* **23**, 1142–1149 (2023)
45. Ajaz, R.H., Hussain, L.: Seed classification using machine learning techniques. *Seed* **2**, 1098–1102 (2015)
46. Khatri, A., Agrawal, S., Chatterjee, J.M.: Wheat seed classification: utilizing ensemble machine learning approach. *Sci. Program.* **2022**, 1–9 (2022)
47. Hernández-Orallo, J., Flach, P., Ferri, C.: A unified view of performance metrics: translating threshold choice into expected classification loss. *J. Mach. Learn. Res.* **13**, 2813–2869 (2012)
48. Eldem, A.: An application of deep neural network for classification of wheat seeds. *Eur. J. Sci. Technol.* (2020). <https://doi.org/10.31590/ejosat.719048>
49. Hashim, F.A., Hussain, K., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W.: Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **51**, 1531–1551 (2021)
50. Braik, M., Sheta, A., Al-Hiary, H.: A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. *Neural Comput. Appl.* **33**, 2515–2547 (2021)
51. Abdollahzadeh, B., Soleimanian Gharehchopogh, F., Mirjalili, S.: Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **36**, 5887–5958 (2021)
52. Heidari, A.A., et al.: Harris hawks optimization: algorithm and applications. *Future Gener. Comput. Syst.* **97**, 849–872 (2019)
53. Ezugwu, A.E., Agushaka, J.O., Abualigah, L., Mirjalili, S., Gandomi, A.H.: Prairie dog optimization algorithm. *Neural Comput. Appl.* **34**, 20017–20065 (2022)

- 54. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016)
- 55. Singh, K.K., Elhoseny, M., Singh, A., Elngar, A.A. (eds.): *Machine learning and the internet of medical things in health-care*. Elsevier, Amsterdam (2021). <https://doi.org/10.1016/C2019-0-03077-4>
- 56. Myerson, J., Green, L., Warusawitharana, M.: Area under the curve as a measure of discounting. *J. Exp. Anal. Behav.* **76**, 235–243 (2001)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

**Beytullah Yağız¹ · Erdal Eker² · Yener Altun¹ · Şeyma Nur Atar² · Davut Izci^{3,7} ·
Serdar Ekinci⁴ · Mohit Bajaj^{5,8} · Mohammad Shabaz⁶**

✉ Mohammad Shabaz
bhatsab4@gmail.com

Beytullah Yağız
beytmat@gmail.com

Erdal Eker
e.eker@alparslan.edu.tr

Yener Altun
yeneraltun@yyu.edu.tr

Şeyma Nur Atar
seymatar97@gmail.com

Davut Izci
davutizci@gmail.com; davutizci@uludag.edu.tr

Serdar Ekinci
sekinci@beu.edu.tr

Mohit Bajaj
mohitbajaj.ee@geu.ac.in

¹ Van Yüzüncü Yıl University, Van, Turkey

² Social Science Vocational School, Muş Alparslan University, Muş, Turkey

³ Department of Electrical and Electronic Engineering, Bursa Uludag University, Bursa 16059, Turkey

⁴ Department of Computer Engineering, Bitlis Eren University, Bitlis 13100, Turkey

⁵ Department of Electrical Engineering, Graphic Era (Deemed to be University), Dehradun 248002, India

⁶ Marwadi University Research Centre, Department of Computer Engineering, Faculty of Engineering and Technology, Marwadi University, Rajkot 360003, Gujarat, India

⁷ Applied Science Research Center, Applied Science Private University, 11931, Amman, Jordan

⁸ Graphic Era Hill University, Dehradun 248002, India