



ARTICLE

An Improved Animated Oat Optimization Algorithm with Particle Swarm Optimization for Dry Eye Disease Classification

Essam H. Houssein^{1,*}, Eman Saber¹ and Nagwan Abdel Samee²

¹Faculty of Computers and Information, Minia University, Minia, 61519, Egypt

²Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

*Corresponding Author: Essam H. Houssein. Email: essam.halim@mu.edu.eg

Received: 17 June 2025; Accepted: 13 August 2025; Published: 31 August 2025

ABSTRACT: The diagnosis of Dry Eye Disease (DED), however, usually depends on clinical information and complex, high-dimensional datasets. To improve the performance of classification models, this paper proposes a Computer Aided Design (CAD) system that presents a new method for DED classification called (*IAOO-PSO*), which is a powerful Feature Selection technique (FS) that integrates with Opposition-Based Learning (OBL) and Particle Swarm Optimization (PSO). We improve the speed of convergence with the PSO algorithm and the exploration with the *IAOO* algorithm. The *IAOO* is demonstrated to possess superior global optimization capabilities, as validated on the IEEE Congress on Evolutionary Computation 2022 (CEC'22) benchmark suite and compared with seven Metaheuristic (MH) algorithms. Additionally, an *IAOO-PSO* model based on Support Vector Machines (SVMs) classifier is proposed for FS and classification, where the *IAOO-PSO* is used to identify the most relevant features. This model was applied to the DED dataset comprising 20,000 cases and 26 features, achieving a high classification accuracy of 99.8%, which significantly outperforms other optimization algorithms. The experimental results demonstrate the reliability, success, and efficiency of the *IAOO-PSO* technique for both FS and classification in the detection of DED.

KEYWORDS: Feature selection (FS); machine learning (ML); animated oat optimization algorithm (AOO); dry eye disease (DED); oppositional-based learning (OBL); particle swarm optimization (PSO)

1 Introduction

A common disease known as Dry Eye Disease (DED) occurs when your tears are unable to adequately lubricate your eyes. Tears can be insufficient and erratic for various reasons [1]. DED has been shown to successfully prevent the transmission of the COVID-19 virus, which has been linked in clinics [2,3]. There are notable differences in the severity of DED [4]. Wearing masks has been shown to successfully prevent the transmission of the COVID-19 virus, which has been linked to an increase in symptoms of DED [5]. DED is a long-term condition that develops gradually over time. High-risk individuals with DED include those who use video terminals, wear makeup and contact lenses daily, use eye drops over time, stay up late, use their eyes for extended periods, and/or are older than 30 years. Irritation, dryness, burning, photophobia, eye weariness, increased secretions, redness, feeling like a foreign thing, and ocular pain are the early symptoms of DED [6]. In the latter stages of DED, corneal ulcers and even perforations may develop, ultimately resulting in blindness. Early detection and diagnosis of DED are clinically significant for eye care professionals to develop a thorough and precise treatment plan, which is significant for society and the economy to maintain people's normal lives and stabilize the social economy and development.



To enable early detection, this demonstrates the necessity of a computer-aided diagnosis (CAD) system. By increasing diagnostic accuracy, these systems can facilitate earlier intervention and more successful treatment [7]. AI-driven technologies can greatly improve the capabilities of CAD systems, increasing the accuracy and accessibility of early detection of DED [8].

The diagnosis of DED often relies on complex and high-dimensional datasets that include clinical parameters and ocular biomarkers. Although big datasets provide useful information, noise and extraneous features introduced by their high dimensionality can impair ML models' performance. To address these problems, FS is crucial for locating important biomarkers and simplifying data, which helps ML models classify patients with DED more precisely, improving clinical results and diagnostic precision.

The two main categories of FS techniques are the wrapper and filter approaches [9]. Regardless of the particular ML model, filter methods assess the significance of features. These techniques are computationally efficient, but could miss crucial feature interactions that are necessary for some learning tasks. Wrapper techniques, on the other hand, iteratively evaluate feature subsets, applying the performance of a particular ML algorithm to identify the optimal subset [10,11]. The high computational cost of these approaches is a major drawback, although they are excellent at identifying complicated feature connections. This is because they need to repeatedly train the model and evaluate it across various feature subsets to determine the best combination. Researchers are increasingly using Metaheuristic (MH) algorithms for more efficient FS because of the high processing costs of exhaustive search methods and the inefficiency of random search approaches. These algorithms are commonly used in wrapper-based approaches, where the selected ML classifier and the MH technique have a significant impact on classification performance [12].

Several MH algorithms have been adapted for FS and have received recognition for their promising results in numerous research studies. These include Evolutionary Algorithms (EAs) [13], Ant Colony Optimization (ACO) [14], Genetic Algorithm (GA) [15], and Artificial Bee Colony (ABC) [16]. Even while these algorithms show excellent search capabilities and frequently produce results that are close to optimal, their generalizability across different datasets is still restricted. Additionally, as the dimensionality of the data increases, FS's accuracy tends to decrease. Therefore, there is still much room to improve these approaches to better handle these difficulties.

The Animated Oat Optimization Algorithm (AOO) [17] introduces a novel bio-inspired approach by emulating the natural behaviors of animated oats, thereby contributing significantly to the development of optimization techniques. AOO models three core biological behaviors: (i) Seed dispersal by natural forces like wind, water, and animals; (ii) hygroscopic motion, where the primary awn of the seed undergoes deformation and rotation, enabling self-propelled rolling and spatial distribution; and (iii) an energy-storage mechanism activated during rolling, which allows the seed to overcome obstacles and trigger further motion under specific conditions for enhanced dispersion.

Furthermore, Tizhoosh [18] has validated the notion of Opposition-Based Learning (OBL), which holds that the opposite number is closer to the ideal value than a random number and can improve search capabilities and speed up convergence. Numerous scholars used this mathematical technique extensively in various MH algorithms [19–21]. In their paper, Kang et al. [19] presented the OBL strategy as a solution to the issues of limited population diversity and early convergence in conventional PSO. As demonstrated by the experiment results, Zhang et al. [22] used the OBL to enhance the performance of the original gray wolf optimizer. Their suggested algorithm outperformed the original Grey Wolf Optimization (GWO) Algorithm and other MH algorithms in terms of search capability and convergence rate.

1.1 Contribution

The following are this paper's primary contributions:

- This paper presents the Improved Animated Oat Optimization (IAOO-PSO) algorithm, which integrates the Animated Oat Optimization (AOO) with the Particle Swarm Optimization (PSO) algorithm and the Oppositional-Based Learning (OBL) technique. Through increased exploration capabilities, these improvements overcome the drawbacks of current FS techniques, boosting feature subset selection and overall performance.
- We propose the IAOO-PSO-based eye disease classification model, a novel approach to classify DED. This model illustrates how well *IAOO-PSO* works for FS when combined with an SVM classifier.
- The *IAOO-PSO* algorithm outperforms seven other MH algorithms, including (PSO [23], weighted mean of vectors (INFO) [24], Whale Optimization Algorithm (WOA) [25], Beluga Whale Optimization (BWO) [26], Sine Cosine Algorithm (SCA) [27], Moth-Flame Optimization (MFO) [28], and the original AOO algorithm) tested on the DED dataset and the CEC'22 test suite, according to extensive studies. The results demonstrate how well *IAOO-PSO* performs when handling limited optimization problems related to FS and classification.
- This study provides a comprehensive evaluation of the proposed model using multiple performance metrics, including Accuracy (ACC), sensitivity (SE), F1-score (F1), specificity (SP), and positive predictive value (PPV). The detailed analysis highlights the robustness and reliability of the *IAOO-PSO* algorithm in both FS and classification tasks.
- The DED classification model based on *IAOO-PSO-SVM* achieved a remarkable classification ACC of 99.8%. The remarkable results suggest that the suggested *IAOO-PSO-SVM* method has great potential for diagnosing DED, especially in terms of increasing classification accuracy through optimized FS.

1.2 Paper Structure

The remainder of the paper is structured as follows: Recent studies on two major subjects are reviewed in [Section 2](#): [Section 2.1](#) discusses the creation of CAD systems for DED, while [Section 2.2](#) discusses the use of MH algorithms for FS in eye disease datasets. The suggested approach and methods used are described in full in [Section 3](#). First performance tests utilizing the CEC2022 benchmark functions are shown in [Section 4](#). Moreover, the comprehensive experimental results of the classification model on the DED dataset are shown in [Section 5](#). [Section 7](#) concludes with a summary of the paper's conclusions.

2 Literature Review

The growth in DED has led to the development of CAD systems that use ML and optimization techniques to increase the precision of diagnoses, especially through efficient FS and classification systems. After highlighting recent developments in CAD systems for DED classification, this part discusses the expanding use of MH algorithms in medical FS, emphasizing existing techniques and their drawbacks.

2.1 Applications of CAD Systems in DED Classification

In detecting and diagnosing DED, CAD systems are making significant strides. The enhancement of the precision, effectiveness, and applicability of these systems has been the main focus of recent research. The methods, constraints, and results of the most recent CAD systems created for DED classification are thoroughly reviewed in this section. Therefore, DED therapy depends heavily on early identification. Normal DED detection using tear break-up time, meibomian pictures, and other methods is labor-intensive, time-consuming, and inefficient. Artificial intelligence (AI) has been evolving, producing several notable

advancements in the medical domain [29]. The critical value of this approach for detecting ocular eye diseases is rising [30]. The diagnosis and prediction of eye diseases, including high myopia [31], diabetic retinopathy [32], and cataracts [33], have made extensive use of it.

Yabusaki et al. [34] developed an ML model to diagnose DED that automatically defines 11 features for Interfering Fringe Color (IFC) images, such as brightness, color saturation, color ratio, and local complexity. This methodology demonstrated better inter-rater agreement (Kappa coefficient = 0.820) with eye-care practitioners and higher F1-scores (0.815, 0.095) when compared to a Bag of Visual Words (BVW) ML model. According to the researchers' results, ML is a resource-efficient and successful method for detecting DED that reduces diagnostic biases. In a different study, Cartes et al. [35] used ML methods to classify DED patients based on their mean osmolarity records (20 with DED and 20 normal). Because the logistic classifier algorithm achieved an accuracy of 85%, the results indicate that tear osmolarity in DED can be effectively discriminated. According to the research, ML is a successful method for detecting DED utilizing variables related to tear osmolarity [35].

Yang et al. [36] studied using ML models to predict DED in patients who had undergone a vitrectomy. Using a dataset of 217 cases with features such as OSDI, TBUT, osmolarity, interocular difference, ocular surface staining corneal spot number, conjunctival spot number, length and width of lid margin, and demographic data, they used supervised ML models, specifically Logistic Regression (LR) and Artificial Neural Network (ANN). An Area Under the Curve (AUC) of 0.741 for LR and 0.786 for ANN is revealed by the Receiver Operating Characteristic (ROC) curve analysis. The study concludes that ML models are effective in detecting DED, with ANN outperforming LR in this regard [36]. Most of the research has been published in the last few years, particularly after 2014. Table 1 displays their relevant references, datasets used, objectives, algorithms used, and results.

Table 1: An overview of AI for DED classification

| Ref. | Dataset | Objective | Used algorithm | Result |
|------|-------------------------|---|--|--|
| [37] | Cornea dataset | Mechanism of DED and therapeutic effect | GLM, MLR | MLR: p -values < 0.05 for DC density and CFS while GLM: p -values < 0.05 for DC density and number of DCs% |
| [38] | Tear meniscus dataset | Calculate the height of the tear meniscus | CNN | Acc = 82.5%, SE = 0.899, PPV = 0.911, F1-score = 0.901 |
| [39] | Meibomian gland dataset | Meibomian Gland Dysfunction (MGD) diagnosis | GAN | F1-score = 0.825 |
| [40] | Tear film dataset | Tear film patterns classification | SVM, RF, RT, Naive Bayes, DNN, simple NN | RF: ACC = 97.54% |
| [41] | Meibomian gland dataset | MGD diagnosis | Nine CNN combinations | Single CNN: SE = 0.942, SP = 0.821, ensemble CNNs: SE = 0.921, SP = 0.988 |
| [42] | Cornea dataset | DED mechanism and therapeutic impact | CNN (U-net) | SE = 96% |

(Continued)

Table 1 (continued)

| Ref. | Dataset | Objective | Used algorithm | Result |
|------|--|-----------------------------------|--|---|
| [34] | Tear film dataset | DED classification | SVM | F1-score = 0.981. |
| [43] | Meibography dataset | MGD diagnosis | Riesz pyramid | ACC = 99.08%, SE = 1, SP = 0.98 |
| [44] | Tear film dataset | Tear film patterns classification | SVM, Decision tree, Naive Bayes, simple NN | NN: ACC = 96%, SE = 92%, SP = 97%, PPV = 92%, F1-score = 0.93 |
| [45] | Tearscope-plus dataset | Tear film patterns classification | SVM | ACC = 96.09%, PPV = 92.00%, SE = 89.66%, SP = 97.98%, F1-score = 91.23% |
| [46] | IR meibography dataset | MGD detection | PA, SA, FFA, SVM | SP = 96.1%, SE = 97.9% |
| [35] | Tear osmolarity dataset | DED detection | LR, Naive Bayes, SVM, RF | LR: ACC = 85% |
| [47] | Peptides and proteins in tears dataset | DED detection | NN | DED: SE = 0.83, SP = 0.93 |
| [48] | Peptides and proteins in tears dataset | Protein analysis | DNN,discriminant analysis | DNN: ACC = 89%, discriminant analysis: ACC = 71% |
| [49] | Proteins in tears dataset | Protein analysis | K-means, discriminant analysis | K-means: ACC = 71%, discriminant analysis: ACC = 72%. |

2.2 Feature Selection Using a Metaheuristic in DED Datasets

The success of MH algorithms in addressing issues such as large dimensionality, complexity, and noise has led to their increased prominence in the literature for FS in medical datasets. This section examines the main contributions and uses of these algorithms, highlighting how they can enhance diagnostic precision and prediction performance.

An approach for detecting optic discs has been presented that combines the Ant Colony Optimization algorithm with anisotropic diffusion [50]. Another automated method for the detection of optic discs has been described using the firefly algorithm [51]. Various swarm intelligence-based methods have also been used to detect the optic disc [52]. Using their method, the authors have demonstrated that the firefly algorithm produces the best results. To efficiently detect optic discs, the gravitational law has also been applied [53], but there is still room for improvement in performance. The bat optimization approach [51] has also been used recently for optical disc detection, and the method has improved the disc segmentation results in a relatively short calculation time.

Nam et al. [54] proposed an Explainable Artificial Intelligence EAI-based ML prediction system for the detection of DED and the identification of risk and protective factors. Of the 4391 cases collected for the study, 80% were used for training from the KHNHANES dataset. Several packages, such as rpart, psych, glmnet, survey, WeightedROC, ggplot2, weights, and qgraph, were used together with the R programming

language (version 3.6.1). To minimize overfitting and improve robust performance, ML models were used for FS and prediction model simplification.

The study found that several important features, such as age, rhinitis, lipid-lowering medicine, sex, cataract surgery, current depression, rhinoplasty, stress and omega-3 fatty acid intake, were associated with DED [54]. Moreover, DED is a serious postoperative consequence and an important factor to take into account while developing a laser surgical strategy. EAI is a useful tool for expert-level knowledge of the criteria utilized in the selection of refractive surgery techniques, according to Yoo et al. [55]. They used ANN ML, SVM, and XGboost models to classify patients into four categories of surgeries according to the features of DED, lifestyle, and financial constraints. Accuracy was assessed using both internal and external datasets and the 10-fold cross-validation approach. Table 2 displays an overview of the MHs of FS on DED and their relevant references, datasets used, objectives, algorithms used, and results.

Table 2: An overview of MHs for FS on DED

| Ref. | Dataset | Objective | Used algorithm | Result |
|------|--|--|--|---|
| [56] | DED dataset | DED and MGD diagnosis | Enhanced Quantum Bacterial Foraging Optimisation Algorithm (EQBFOA). | ACC = 96.34% |
| [57] | Digital Retinal Images for Vessel Extraction (DRIVE) and Structured Analysis of the Retina (STARE) dataset | Optic disc detection | Supervised Gradient Vector Flow snake (SGVF snake) model | ACC = 91% |
| [58] | Eye diseases dataset | DED and MGD diagnosis | Ant Colony Optimization algorithm, Bacterial Foraging Optimization, Firefly algorithm, Cuckoo Search algorithm, and Krill Herd algorithm | Bacterial Foraging Algorithm: ACC = 99.55% with DiaRetDB1, ACC = 100% with HEI-MED and DRIVE datasets, and ACC = 98% with STAR dataset. |
| [59] | Retinal fundus dataset | Eye disease detection | Flower Pollination Optimization Algorithm + CNN (FPOA-CNN) | PPV = 98.30%, ACC = 95.27%, SP = 95.21%, and F1-score = 93.3% |
| [60] | Cataract dataset | Cataract diagnosis | DCNN + RF | ACC = 90.69% |
| [61] | AMD disease dataset | DED and Age-related Macular Degeneration (AMD) diagnosis | ACO + SVM | ACC = 99% |

(Continued)

Table 2 (continued)

| Ref. | Dataset | Objective | Used algorithm | Result |
|------|---------------------|-----------------------|----------------|--|
| [62] | Eye disease dataset | Eye disease diagnosis | FPOA-CNN | The best-ranked feature with a mean difference of 1029.7 |
| [63] | Eye disease dataset | Glaucoma diagnosis | MC-LS-SVM | ACC = 98.11%, SP = 94.5% |
| [64] | Eye disease dataset | DR diagnosis | ADL-CNN | ACC = 98%, SP = 95.1% |
| [65] | Eye disease dataset | Eye disease diagnosis | DL-CNN | ACC = 96.5%, SP = 98.6% |

Although CAD systems and MHs-based FS techniques have advanced, many current methods still have issues with generalization, efficiency, and scalability. Following the no-free-lunch (NFL) [66], which states that no single optimization algorithm excels at solving every optimization problem, this study suggests a CAD system that uses the Improved Animated Oat Optimization Algorithm (IAOO) to improve the classification of DED. Using AOO with OBL and PSO to overcome the drawbacks of the original techniques and enhance the exploration and exploitation capabilities.

3 Methodology

This section describes the Improved Animated Oat Optimization (IAOO) algorithm, the PSO algorithm, the original algorithm (AOO), and the combination of IAOO and PSO called (IAOO-PSO). The PSO algorithm speeds up convergence, while IAOO uses Oppositional-Based Learning (OBL) to balance exploration and exploitation. By doing this, the likelihood of results globally optimal solutions will increase, and local optima will be efficiently avoided. The following subsections present the specific enhancements and methodological advancements introduced in the *IAOO-PSO* algorithm. These sections elaborate on the implementation strategies and analyze the impact of each improvement on the algorithm's overall performance.

3.1 Animated Oat Optimization Algorithm (AOO)

AOO is a recently released MH algorithm that draws inspiration from the animated oat's natural behavior in the surroundings [17]. The three distinct behaviors of animated oat are simulated by AOO:

1. Seed dispersal by natural forces like wind, water, and animals.
2. Hygroscopic movement induces deformation and rotation of the primary awn in animated oat seeds, enabling the seed to roll and disperse effectively across the ground.
3. During the rolling propagation phase, energy is accumulated when the seed encounters obstacles, activating a propulsion mechanism under certain conditions to facilitate further dispersal.

3.1.1 Initialization

As demonstrated by Eq. (1), AOO begins with a collection of random solutions.

$$X = \begin{bmatrix} X_{1,1} & \dots & X_{1,j} & \dots & X_{1,\text{Dim}-1} & X_{1,\text{Dim}} \\ X_{2,1} & \dots & X_{2,j} & \dots & X_{2,\text{Dim}-1} & X_{2,\text{Dim}} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \ddots & X_{i,j} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ X_{N-1,1} & \dots & X_{N-1,j} & \dots & X_{N-1,\text{Dim}-1} & X_{N-1,\text{Dim}} \\ X_{N,1} & \dots & X_{N,j} & \dots & X_{N,\text{Dim}-1} & X_{N,\text{Dim}} \end{bmatrix} \quad (1)$$

The location information associated with each member of the i^{th} subgroup is represented by x_i . N is the number of individuals that make up the subpopulation, and Dim is the problem's dimensional size. Eq. (2) allows for the calculation of each $x_{i,j}$ in the matrix X .

$$x_{i,j} = r \cdot (UB_j - LB_j) + LB_j, \quad i = 1, 2, \dots, N; \quad j = 1, 2, \dots, \text{Dim} \quad (2)$$

where r is a random number between 0 and 1.

3.1.2 Parameters Calculation

Features of the animated oat seed dispersal process in motion are related to the mass, rolling coefficient of eccentricity, and length of the seed's main awn. The formula below can be used to compute these:

$$\begin{cases} m = 0.5 \cdot \frac{r}{\text{Dim}} \\ L = N \cdot \frac{r}{\text{Dim}} \\ e = 0.5 \cdot \frac{r}{\text{Dim}} \\ c = 1 - \left(\frac{t}{T}\right)^3 \end{cases} \quad (3)$$

where m stands for the animated oat seed's mass, N for its population size, L for its main awn length, e for the eccentric rotation coefficient during seed rolling, t for the number of iterations that are currently occurring, and c for the dynamic adjustment factor.

3.1.3 Exploration Phase

Once some of the animated oat seeds have fallen from the plant, wind, water, or animals are the main factors that disperse them. Because of its high degree of unpredictability, this dispersal method permits investigation of a large solution space. These are the position updates:

$$W = \frac{c}{\pi} \cdot (2 \cdot r_{\text{Dim}} - 1) \otimes UB \quad (4)$$

$$X_{t+1}(i) = \begin{cases} \frac{1}{N} \cdot \sum_{j=1}^N X_t(j) + W, & \text{if } \text{mod}(i, N/10) = 0, \\ X_{\text{best}} + W, & \text{if } \text{mod}(i, N/10) = 1, \\ X_t(i) + W, & \text{otherwise,} \end{cases} \quad (5)$$

where the positions of the i th and best individuals in the population are denoted by $X_{t+1}(i)$ and X_{best} , respectively, in the $t + 1$ generation.

3.1.4 Exploitation Phase

The remaining animated oat seeds will now be divided into two groups according to whether or not they run into problems during dissemination. Furthermore, we assume that the two outcomes have an equal chance. When there are no obstructions, the hygroscopic rolling, driven by moisture-induced stress gradients, causes the seed location to vary. We use crucial curvature for snap buckling to represent this motion, which was motivated by Lindtner et al. [67], who showed that anisotropic swelling is controlled by the orientation of cellulose microfibrils. The rolling mechanism can be represented theoretically using torque formulas and eccentric rotation:

$$A = UB - \left| \frac{UB \cdot t \cdot \sin(2\pi r)}{T} \right| \quad (6)$$

$$R = (m \cdot e + L^2) \cdot \frac{r_{Dim} \cdot (-A, A)}{Dim} \quad (7)$$

$$Levy(dim) = 0.01 \cdot \frac{\mu \cdot \sigma}{|v|^{1/\beta}} \quad (8)$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1 + \beta}{2}\right) \cdot \pi \cdot 2^{\frac{\beta-1}{2}}}\right)^{1/\rho} \quad (9)$$

$$X_t(i) = X_{best} + R + c \cdot Levy(Dim) \otimes X_{best} \quad (10)$$

where the size of the random matrix $r_{Dim}(-A, A)$ corresponds to the goal problem and its values range from $-A$ to A . The predicted position or mean value, represented by μ in Levy flight, is a random number between 0 and 1 that is usually used to modify the step size. During flying, this unpredictability aids in regulating movement direction and distance. The step length's scale parameter, σ , establishes the range of step length changes and regulates the step length distribution's width. The particle or person's motion condition is reflected by the current velocity vector, denoted by v . The stable distribution's parameter, β , controls the step lengths' diversity and randomness as well as the distribution's shape. Lastly, Γ stands for the gamma function, which is a continuous extension of the factorial function and is used to determine the factorial values of parameters that are not integers. All we have to do to use the Levy flight is to predefine the problem's dimension ($\beta = 1.5$). We assume that the main awn ejects itself using stored energy when the seed runs into difficulties during the dissemination phase. Projectile motion characterizes the entire seed dispersal process. We use a condensed projectile motion model for position updates to model this procedure, and the following equation describes how the position is updated.

$$B = UB - \left| \frac{UB \cdot t \cdot \cos(2\pi r)}{T} \right| \quad (11)$$

$$\left\{ \begin{array}{l} k = 0.5 + 0.5 \cdot r \\ x = 3 \cdot \frac{r}{Dim} \\ \theta = \pi \cdot r \\ \alpha = \frac{1}{\pi} \cdot e^{\frac{r'}{T}} \end{array} \right\} \quad (12)$$

$$J = \frac{2k \cdot x^2 \cdot \sin(2\theta)}{mg} \cdot \frac{r_{\text{Dim}} \cdot (-B, B)}{\text{Dim}} \cdot (1 - \alpha) \quad (13)$$

$$X_t(i) = X_{\text{best}} + J + c \cdot \text{Levy}(\text{Dim}) \otimes X_{\text{best}} \quad (14)$$

where θ describes the angle between the ground and the seed's launch direction, α corresponds to the air drag coefficient influencing its motion through the air, r' is a randomly chosen value within the interval $[0, T]$, k represents the elasticity of the seed's primary awn, and x signifies the variation in the awn's length as it stores elastic energy before release

3.2 Particle Swarm Optimization Algorithm (PSO)

The PSO [68] requires fewer parameters to adjust and is straightforward to use. The field of swarm intelligence studies computer systems that are motivated by the collective intelligence that emerges from the collaboration of homogeneous agents in the environment, such as fish schools and bird flocks. PSO is modeled after the structured behavior of schools of fish and flocks of birds [69].

PSO is a population-based optimization tool in which an initial population of random particles is used to guide the algorithm's search for optima through generational updates. Let's say the colony is made up of M particles and the search space is D -dimensional. With $i = 1, 2, \dots, M$, each particle has a position $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ and a velocity $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. The best position discovered by any particle in the swarm is known as the global best ($gbest$), and when a particle travels, it recalls its best position prior, known as the personal best ($pbest$).

By monitoring their own best, the swarm will progress towards the top spot globally as the number of iterations increases. $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ represents the $gbest$ of the entire swarm, while $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ represents the $pbest$ of the i th particle. Three basic phases are iteratively carried out by the PSO algorithm until a stopping criterion is met. The following are [70]:

1. Calculate each particle's fitness within the population by applying the optimized objective function.
2. By comparing the recently assessed fitness with the previous best individual and global fitness, the best fitness and positions can be updated in the appropriate places.
3. Each particle's position and velocity within the population should be updated. The PSO algorithm's optimization capability is the result of this updating stage, according to the following equations:

$$v_{i,d} = \omega \cdot v_{i,d} + c_1 \cdot r_1 \cdot (p_{i,d} - x_{i,d}) + c_2 \cdot r_2 \cdot (p_{g,d} - x_{i,d}) \quad (15)$$

$$x_{i,d} = x_{i,d} + v_{i,d} \quad (16)$$

where the learning variables c_1 and c_2 determine the relative influence of the cognitive and social components, respectively, and ω is the inertia weight.

3.3 Opposition-Based Learning (OBL)

Conventional MH algorithms begin the search with an initial solution consisting of a set of randomly generated numbers. The algorithm's convergence rate is unstable and typically slow. OBL is implemented to get over these issues, and both randomly produced and reverse solutions are taken into consideration.

In 2005, the field of machine learning was exposed to the main opponent concept [18]. Various OBL techniques have since been effectively added to MH algorithms to boost performance. A thorough analysis of the application and development of OBL techniques may be found in [71]. The interval $[LB_i, UB_i]$ defines

x_i for every point $X = (x_1, x_2, \dots, x_D)$ in the D dimensional space. A distinct opposite point, $\overline{x_i}$, is defined as follows:

$$\overline{x_i} = LB_i + UB_i - x_i \quad (17)$$

3.4 Support Vector Machines (SVMs)

SVM technique, commonly used for regression and classification applications, were used as a classifier in the classification phase. Vapnik was the first to develop SVM in the late 20th century [72]. Since then, it has been used in several fields, including bioinformatics and biometrics. The fundamental idea is to use hyperplanes to divide the data into distinct classes. When the data is linearly separable, SVM performs exceptionally well; when the data is not linearly separable, it struggles. Kernel functions allow for linear separation by mapping the data to a higher-dimensional region. The performance of SVM depends on choosing the right kernel function and parameters [73]. Two important parameters are C and γ . C balances the trade-off between smooth decision boundaries and accurate training data classification, while γ shows the impact of specific training instances. Eq. (18) defines the SVM approach used for handling binary classification problems.

$$f(x) = [y, \phi(x)] + z \quad (18)$$

Eq. (19) computes y , Eq. (20) determines z , and $\phi(x)$ displays the mapping of sample x to the high-dimensional space.

$$g(y, \epsilon) = \frac{1}{2} \|y\|^2 + C \sum_{i=1}^N \xi_i \quad (19)$$

$$y_i([y, \phi(x_i)] + z) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad (20)$$

where the slack variable ξ_i is used to deal with misclassifications. Eq. (21) illustrates how the Radial Basis Function (RBF) kernel nonlinearly translates input vectors to the appropriate feature space.

$$k(x_i, x_j) = [\phi(x_i), \phi(x_j)] \quad (21)$$

3.5 The Proposed IAPO Algorithm

This section presents the IAPO-PSO, a suggested algorithm that improves on the original APO by combining the OBL technique and the PSO algorithm. The OBL technique is integrated into the initialization phase to increase the probability of escaping local optima by enhancing population diversity and guiding the search process closer to the global optimum. To produce more useful candidate solutions during exploitation, we also use the PSO method to accelerate convergence. This improves convergence toward the optimal solution while expanding the search in potential areas. The main drawbacks of the original APO, such as its vulnerability to early convergence, inadequate exploitation in intricate environments, and a poor balance between exploration and exploitation, were intended to be addressed by these changes. To guarantee robustness and convergence stability, effective optimization necessitates resolving these problems, particularly the propensity to become stuck in local minima. The following describes the primary stages of the proposed IAPO-PSO algorithm:

3.5.1 IAPO-PSO Initialization Phase

To create a varied and useful initial population, the initialization step of the APO algorithm is crucial. Within the search region delineated by the lower and upper limits, LB and UB , each solution is produced at

random. The algorithm uses OBL to increase diversity. We integrate the OBL into the fundamental algorithm for AOO. OBL, which Tizhoosh first presented, suggests that opposing numbers have a higher chance of solving arbitrary ones. OBL has been successfully applied to many algorithms and aims to improve the effectiveness of MH algorithms. This study incorporates OBL late in the initial phase to fully explore optimal solutions and avoid local stagnation. The OBL technique is calculated using Eq. (17).

After evaluating the \bar{x}_i fitness values, the solution with the highest fitness is added to the original population. By guaranteeing a varied and evenly distributed initial population, this method enhances the algorithm's ability to explore.

3.5.2 Termination Criteria

Two stopping criteria are used to determine when the IA OO-PSO algorithm stops. In particular, the search is stopped when either the maximum number of iterations ($T_{Maxiter} = 100$) is reached or the best solution does not significantly improve over a predetermined number of consecutive iterations. When the algorithm reaches computational limits or when further investigation is unlikely to yield better results, this hybrid condition ensures that the algorithm stops. The best solution found thus far is continuously monitored and updated during execution. This solution is given back as the optimization process's result at termination. This approach avoids undue computational strain in later, stagnant phases of the search while striking a balance between exploration and exploitation. To ensure that the algorithm terminates either when it finds the best answer or when it is unlikely that any further improvements can be made, the best solution found during the search phase is returned as the final result. The following provides the best solution:

$$= \arg \min_{\substack{1 \leq t \leq T_{Maxiter} \\ 1 \leq i \leq N}} f(\mathbf{x}_i^{(t)}) \quad (22)$$

where the best solution X_{best} is determined as the particle with the minimum objective value among all N individuals across all T_{max} iterations. This guarantees that the algorithm retains and returns the globally optimal solution encountered during the entire search process.

Algorithm 1 shows the pseudo-code of the proposed IA OO-PSO approach.

Algorithm 1: Pseudocode of IA OO-PSO

```

1: Start IA OO-PSO
2: Input: Population size  $N$ , Maximum iterations  $T_{Maxiter}$ , and dimensions  $Dim$ .
3: Output: The best location  $X_{best}$ 
4: Initialize the population using OBL  $\bar{X}_i$  of  $X_i$  Eq. (17)
5: for each individual  $i$  in the population do
6:   Calculate fitness of the opposite point  $FitOpposite(i)$  and  $Fit(i)$  using Eq. (25).
7:   if  $FitOpposite(i) <= Fit(i)$  then
8:      $X_i = \bar{X}_i$ 
9:   end if
10: Calculate AOO parameters using Eq. (3).
11: Calculate particle's position and velocity using Eqs. (15) and (16).
12: end for
13: for  $t : T_{Maxiter}$  do
14:   for  $i : n$  do

```

(Continued)

Algorithm 1 (continued)

```

15:      Use Eq. (3) to calculate the parameters.
16:      if  $r_1 > 0.5$  then
17:          The new position is determined based on the rules defined in Eqs. (4) and (5).
18:      else
19:          if  $r_2 > 0.5$  then
20:              The new position is determined based on the rules defined in Eqs. (6)–(10).
21:          else
22:              The new position is determined based on the rules defined in Eqs. (11)–(14).
23:          end if
24:      end if
25:      Assess the performance (fitness) of all individuals.
26:  end for
27:  Record population fitness values and optimal positions;
28: end for
29: Output: The best location  $X_{best}$ 
30: End IA OO-PSO

```

3.5.3 Time Complexity of IA OO-PSO

Three main stages can be used to evaluate the time complexity of IA OO: initialization, solution updates, and fitness assessments. The time complexity during the initialization phase, which includes determining initial fitness values and setting up population placements, is $O(N.Dim)$, where Dim is the number of dimensions and N is the population size. Each individual's opposite points are computed and evaluated by the algorithm during the OBL phase. The sum of these operations' time complexity is $O(N.Dim)$. This phase's time complexity with Max_{iter} iterations is $Max_{iter}.N.Dim$.

When combining these stages, the total time complexity of IA OO-PSO is:

$$O(IA OO - PSO) = O(N.Dim) + O(n.dim) + O(Max_{iter}.N.Dim) = O(Max_{iter} \times n \times dim)$$

Consequently, the total time complexity of the IA OO-PSO algorithm is $O(Max_{iter}.N.Dim)$.

3.6 The Proposed IA OO-PSO-SVM Classification Model

The primary objective of developing an automated diagnostic system using machine learning (ML) techniques is to construct a predictive model that not only fits the training data effectively but also generalizes well to unseen test data—demonstrating high testing accuracy. To improve diagnostic performance, particularly disease classification accuracy, various data mining algorithms are used during the feature preprocessing stage. This preprocessing generally falls into two main categories: feature extraction and FS.

FS aims to identify a subset of features from the original dataset that are most relevant to the target labels, using either statistical or search-based techniques. This process plays a vital role in enhancing model training, controlling overfitting, and promoting generalization. Binary variants of MH algorithms offer an effective approach for exploring the feature space and identifying optimal feature subsets, even in the presence of large-scale feature dimensions. Integrating MHs into FS helps improve both model performance and classification accuracy.

Within FS and ML frameworks, the fitness function in MHs plays a central role by evaluating the quality of each candidate feature subset. Recent research has enhanced these systems by using hybrid MH

approaches to reduce search space complexity while improving classification accuracy [74]. In this study, the integration of a novel DED classification method is demonstrated using a publicly available dataset of DEDs. The classification pipeline consists of four main stages: data preprocessing, feature selection using the proposed *IAOO-PSO* algorithm, classification, and evaluation. The overall framework of the proposed *IAOO-PSO*-based SVM classification model is illustrated in Fig. 1.

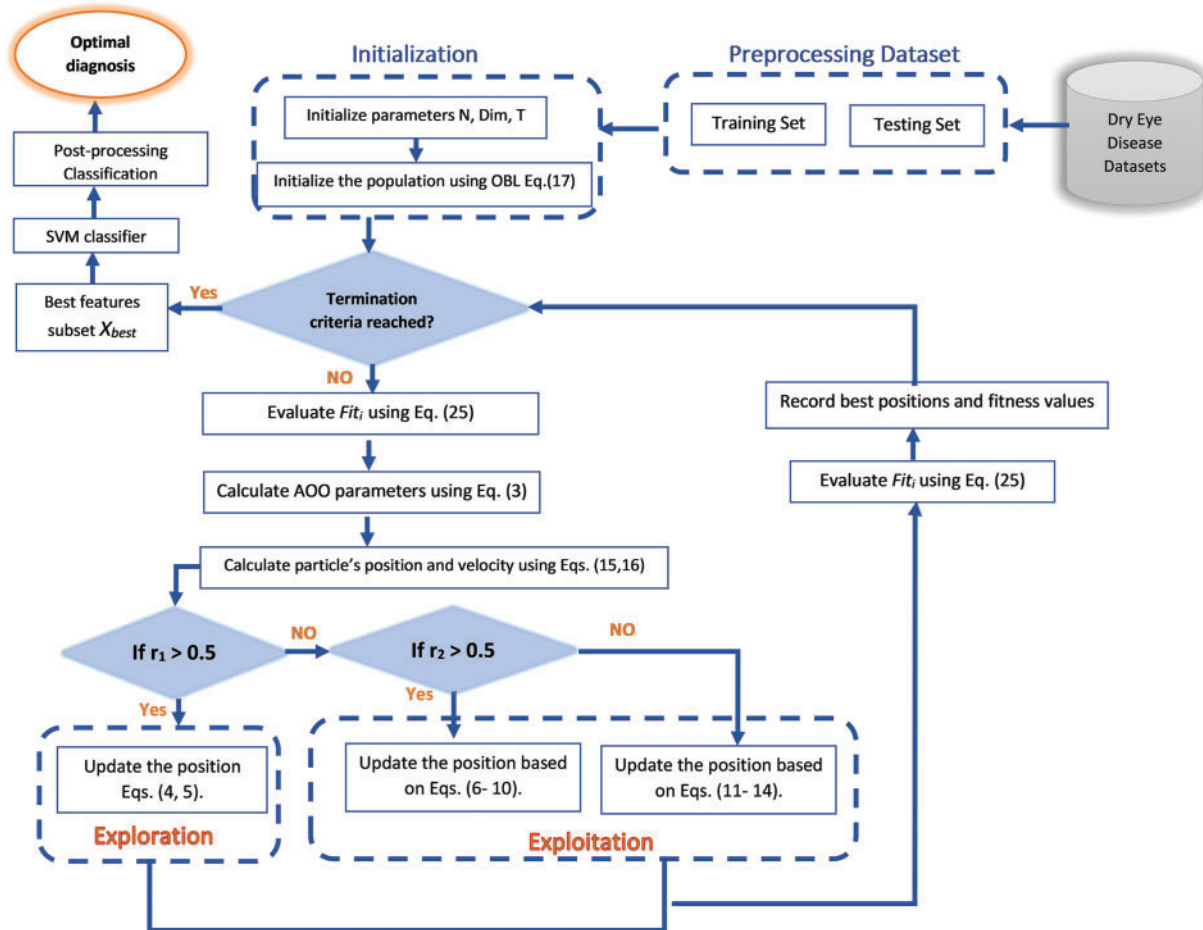


Figure 1: The general structure of the proposed *IAOO-PSO*-based SVM classification model

3.6.1 Phase 1: Collection and Preprocessing of Datasets

In this study, a structured dataset was used to classify DED based on key features such as screen time, blue light filter usage, eye strain, redness, itching, and sleep duration. This dataset enables researchers and healthcare professionals to explore associations between lifestyle factors and ocular health. The data, comprising individuals aged 18 to 45, was obtained from the Kaggle Machine Learning Repository [75].

The dataset contains information from over 20,000 participants, including teenagers, young adults, and middle-aged individuals of both sexes. It consists of 26 features along with a binary target variable, where a value of 1 indicates the presence of dry eye (65%) and 0 denotes the absence (35%). A detailed description of the features of the dataset is provided in Table 3.

Table 3: An overview of the DED dataset's features

| No. | Feature | Description | Range | Type |
|-----|-----------------------------|--|--------------|-------------|
| 1 | Gender | Consisting of both genders | M/F | Categorical |
| 2 | Age | Age of individuals | 18–45 | Numerical |
| 3 | Sleep duration | Measure of time of sleep in hours | 4–10 | Numerical |
| 4 | Sleep quality | Quality of sleep | 1–5 | Numerical |
| 5 | Stress level | Stress levels of individuals | 1–5 | Numerical |
| 6 | Blood pressure | Systolic blood pressure | 90–140 | Numerical |
| 7 | Heart rate | Measure of pulse in beats per minute | 60–100 | Numerical |
| 8 | Daily steps | The daily stepping count of the topic in 1000 units | 1000–20 k | Numerical |
| 9 | Physical activity | The daily physical activity count per minute | 0–180 | Numerical |
| 10 | Height | Height measured in cm | 150–200 | Numerical |
| 11 | Weight | Weight measured in kg | 50–100 | Numerical |
| 12 | Sleep disorder | Find any sleep disorders | Y/N | Categorical |
| 13 | Wake up during night | Verify if the individual wakes up at night | Y/N | Categorical |
| 14 | Feel sleepy during day | Verify drowsiness during day | Y/N | Categorical |
| 15 | Caffeine consumption | Determines the individual's caffeine consumption | Y/N | Categorical |
| 16 | Alcohol consumption | Determines the individual's alcohol consumption | Y/N | Categorical |
| 17 | Smoking | Determines the individual smoking habits | Y/N | Categorical |
| 18 | Medical issue | Whatever medical issue of individual | Y/N | Categorical |
| 19 | Ongoing medication | The current prescribed drug or treatment for any disease | Y/N | Categorical |
| 20 | Smart device before bed | Check the user's recent activity before going to bed | Y/N | Categorical |
| 21 | Average screen time | The individual's average daily screen time | 1–10 | Numerical |
| 22 | Blue-light filter | Shield the eyes when using a screen | Y/N | Categorical |
| 23 | Discomfort Eye-strain | If if the individual experiences any eye strain or pain | Y/N | Categorical |
| 24 | Redness in eye | Identify redness in eye | Y/N | Categorical |
| 25 | Itchiness/Irritation in eye | Categorize prevalence of itchiness in eyes | Y/N | Categorical |
| 26 | DED | Label | Class | N, Y |

Before analysis, extensive data preprocessing was conducted to ensure accuracy and consistency. This included handling missing values and outliers. Furthermore, normalization techniques were applied to standardize feature scales, reducing the potential bias of variable magnitude and facilitating more effective and reliable model training and evaluation.

Data Cleaning

To ensure that the data used in the research was accurate and pertinent, data cleaning was done for this paper. This process entailed removing superfluous or erroneous data that could impair the performance of algorithms and models. Furthermore, the data cleaning procedure comprised fixing dataset mistakes and minimizing duplicates [76]. Three primary steps comprised the data-cleaning process:

- Removing instances in which more than 80% of the data had missing values.
- Features with more than 80% missing values are eliminated.
- We will use median imputation to fill in the remaining missing data.

Data Normalization

Normalization is a technique used to adjust the scale of raw data while retaining its essential characteristics. By converting the original values to a common scale, normalization helps preserve the relative differences between data points within the dataset [77]. The transformed values typically fall within the range of $[(0, 1)$ or $(-1, 1)]$. The normalized weight for a specific attribute W and its value x is calculated using Eq. (23).

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (23)$$

where the mean is represented by μ , the standard deviation by σ , and the original value of attribute W by x .

3.6.2 Phase 2: Feature Selection Using the IA00-PSO Algorithm

To find the most informative subset of features that optimizes classification performance, the *IA00-PSO* is used in this phase. By exploring the search space, *IA00-PSO* minimizes the size of features that are selected while reducing redundancy, removing features that aren't relevant, and improving classification accuracy. A population of solutions is initialized at the start of the optimization process, with each agent standing in for a subset of candidate features that are represented as continuous-valued vectors. The transformation rule specified in Eq. (24) is then used to binarize these vectors to signal feature inclusion or exclusion:

$$X_{bin}^i = \begin{cases} 1, & \text{if } x_i > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

Each solution is assessed by training an SVM classifier on the selected subset. The classification error and the length of the subset are used to provide a fitness function that measures the classification performance. In Eq. (25), the fitness function is defined; a lower value denotes a better solution:

$$fit = \alpha \cdot E + (1 - \alpha) \cdot \frac{|S|}{|T|} \quad (25)$$

In this case, E stands for the classification error, $|S|$ for the number of features that were selected, and $|T|$ for the total number of features. The trade-off between classification performance and compactness of the subset is balanced by the parameter $\alpha \in [0, 1]$. To balance classification performance with a small penalty for using more features, we chose $\alpha = 0.9$ for our experiments. After some initial testing, this value was

selected to maintain the feature subset's compactness without materially lowering classification accuracy. The objective of *IAOO-PSO* is to reduce this fitness value as much as possible during the optimization procedure. Through its improved exploration and exploitation processes, *IAOO-PSO* iteratively refines candidate solutions to identify the most relevant features, resulting in lower dimensionality and improved classification results.

3.6.3 Phase 3: Classification

We overcame these difficulties and improved the classification accuracy by modifying the SVM settings using the recommended *IAOO-PSO* for FS. Numerous studies and empirical data demonstrate SVM's higher performance over other classifiers such as DT, kNN, and NB, which is why it was selected. SVM has been widely used as a benchmark in FS research and integrates well with feature selection frameworks. The effect of *IAOO-PSO* in combination with SVM on handling problems of DED classification and FS is covered in detail in this section. A wrapper algorithm finds the optimum combination of feature subsets to train the SVM, and then uses those subsets to increase performance.

Algorithm 2 explains the precise steps in the *IAOO-PSO-SVM* model and how to use Eq. (25) to calculate the fitness function, which indicates the optimal solution.

Algorithm 2: Procedural steps of the *IAOO-PSO-SVM* model applied to DED data for FS and classification

Require: Dataset D_s , number of features f_n , number of iterations n , population size pop

Ensure: Classification results

- 1: **Data Preprocessing**
 - 2: Applying Imputation Techniques to Handle Missing Values.
 - 3: Apply normalization and standardization to the dataset D_s .
 - 4: **Feature Selection**
 - 5: Apply the *IAOO-PSO* algorithm 1
 - 6: Initialize the *IAOO-PSO* algorithm parameters
 - 7: $D_s \leftarrow \text{IAOO-PSO}(fit, dim, N, T, lb, ub, f_n)$
 - 8: Generate particles using *IAOO-PSO* within the range of 1 to $2^{49} - 1$
 - 9: Represent the generated particles in binary format.
 - 10: Apply SVM Algorithm for FS using the generated binary particles.
 - 11: **Classification Process**
 - 12: Apply SVM training using the relevant features derived from D_s .
 - 13: Evaluate the classification performance of the SVM.
-

3.6.4 Phase 4: Evaluation of the Classification Model

To evaluate the classification performance of the proposed *IAOO-PSO-SVM* model, several metrics are used, including Accuracy, F1-score, sensitivity, specificity, and precision. These metrics collectively provide a well-rounded assessment by addressing different facets of prediction quality, such as accuracy, consistency, and the model's ability to balance FP and FN, as illustrated in Table 4.

Table 4: Performance evaluation metrics

| Equation | Description |
|--|--|
| $ACC = \frac{TP + TN}{(TP + TN + FP + FN)}$ | Calculating the proportion of true positives and true negatives relative to the total sample size. |
| $\mu_{ACC} = \frac{1}{M} \sum_{j=1}^M ACC_*^j$ | Accuracy rate. |
| $\mu_{Fit} = \frac{1}{M} \sum_{j=1}^M Fit_*^j$ | The correlation between decreases in the FS ratio and a decrease in classification error rates. |
| $\mu_{FS} = \frac{1}{M} \sum_{j=1}^M f_*^j$ | Select the most relevant features. |
| $SE = \frac{TP}{(TP + FN)}$ | The proportion of true positive cases correctly detected. |
| $\mu_{SE} = \frac{1}{M} \sum_{j=1}^M SE_*^j$ | TP rate. |
| $SP = \frac{TN}{(TN + FP)}$ | The proportion of TN cases correctly detected. |
| $\mu_{SP} = \frac{1}{M} \sum_{j=1}^M SP_*^j$ | TN rate. |
| $PPV = \frac{TP}{(TP + FP)}$ | The proportion of correctly predicted positive instances out of all instances predicted as positive. |
| $\mu_{PPV} = \frac{1}{M} \sum_{j=1}^M PPV_*^j$ | Positive predictive value |
| $F1 = 2 * \frac{PPV * SE}{PPV + SE}$ | Providing a single metric that balances both FP and FN. |
| $\mu_{F1} = \frac{1}{M} \sum_{j=1}^M F1_*^j$ | A harmonic mean of PPV and SE. |
| $\sigma_x = \sqrt{\frac{1}{M} \sum_{j=1}^M (S_*^j - \mu_x)^2}$ | Quantifies the amount of variation or dispersion in a set of values. |
| $\mu_{Time} = \frac{1}{M} \sum_{j=1}^M Time_*^j$ | Average consumption time (Sec.). |

Note: Mean μ , Standard Deviation (STD) σ . Accuracy (ACC), Best Fitness (Fit), Sensitivity (SE), Specificity (SP), positive predictive value (PPV), F1-score (F1). False-Positive (FP), True-Positive (TP), True-Negative (TN), and False-Negative (FN).

4 Experiment 1: Evaluating the Performance of the Proposed IA00-PSO Algorithm on CEC'22 Benchmark Suite

The proposed *IA00-PSO* algorithm is rigorously tested using the CEC'22 benchmark suite [78], which comprises 12 widely adopted optimization functions. These functions are grouped as follows: *F1* (unimodal), *F2–F4* (multimodal), *F5–F8* (hybrid), and *F9–F12* (composite). Unimodal functions, featuring a single global optimum, are well-suited for evaluating convergence speed and exploitation capabilities. Multimodal functions, with multiple local optima, are used to assess the algorithm's exploration proficiency. Hybrid and composite functions are more complex and challenge the algorithm's ability to escape local optima and maintain a balance between exploration and exploitation. All experiments are conducted with a problem dimension of 10.

To further validate the effectiveness of IA OO-PSO, it is compared against a selection of both classical and recently developed MH algorithms, including Particle Swarm Optimization (PSO), weighted mean of vectors (INFO), Whale Optimization Algorithm (WOA), Beluga Whale Optimization (BWO), Sine Cosine Algorithm (SCA), Moth-Flame Optimization (MFO), and the original AOO algorithm. Each algorithm is evaluated using its standard parameter settings as specified in the original literature. A summary of these parameter configurations is provided in Table 5. Standard parameters were used to ensure a consistent, fair, and reproducible experimental setup across all methods. This also allowed us to focus our analysis on the feature selection capabilities of each algorithm, without introducing bias from problem-specific hyperparameter tuning.

Table 5: Configuration details for IA OO-PSO and the comparative optimization algorithms

| Optimizers | Configuration |
|-------------------|--|
| Common parameters | $N_p = 30$ $Dim = 10$ $runs = 30$ $T_{Maxiter} = 100$ |
| INFO | $c = 2; d = 4$ |
| WOA | α reduces from 2 to 0 $a2$ reduces from -1 to -2 |
| SCA | $a = 2$ |
| BWO | $B_F = (0, 0.5)$ |
| PSO | $c_1 = c_2 = 2; V_{max} = 6; \omega = 0.7$ |
| MFO | P reduces from -1 to -2. |
| AOO | $\beta = 1.5$ Communication radius = 35 m |
| IA OO-PSO | $m = 1.5$ |

A number of performance indicators, such as the mean (μ) and Standard Deviation (σ) of the fitness values, were used to guarantee a thorough and equitable assessment. These metrics provided information on accuracy and consistency.

Furthermore, the overall performance ranking of the algorithms was ascertained using the Friedman rank test [79]. Every experiment was conducted using MATLAB R2022a on a system that had an Intel(R) Core(TM) i7-13620H CPU running at 2.40 GHz and 16 GB of Memory.

4.1 Statistical Analysis of CEC'22 Benchmark Functions

In this subsection, the IA OO-PSO algorithm is compared with AOO and six other MH techniques. For 12 benchmark functions with a problem dimension of 10, the mean (μ) and STD (σ) of the fitness scores achieved by IA OO-PSO and the competing algorithms are presented in Table 6.

Table 6: Comparison of IA OO-PSO with seven other algorithms on the CEC'22 benchmark suite

| Function | Metric | IA OO-PSO | AOO | PSO | BWO | INFO | SCA | WOA | MFO |
|----------|----------|-----------|----------|----------|----------|----------|----------|----------|----------|
| F1 | μ | 1.21E+04 | 2.66E+04 | 2.00E+04 | 1.92E+05 | 2.45E+04 | 2.24E+10 | 4.61E+10 | 9.93E+09 |
| | σ | 1.09E+01 | 1.78E+04 | 1.05E+02 | 1.36E+06 | 1.85E+04 | 2.26E+09 | 7.15E+10 | 1.15E+02 |
| F2 | μ | 2.63E+02 | 7.61E+02 | 5.02E+02 | 1.15E+03 | 6.04E+02 | 7.39E+03 | 7.44E+03 | 7.46E+03 |

(Continued)

Table 6 (continued)

| Function | Metric | IAOO-PSO | AOO | PSO | BWO | INFO | SCA | WOA | MFO |
|--------------------|----------|------------------|----------|----------|----------|----------|----------|----------|----------|
| F3 | σ | 7.51E+01 | 6.58E+02 | 2.43E+02 | 5.43E+02 | 3.01E+02 | 7.46E+02 | 2.81E+02 | 5.96E+02 |
| | μ | 5.539E+02 | 6.53E+02 | 6.44E+02 | 6.68E+02 | 6.28E+02 | 7.53E+02 | 7.61E+02 | 7.77E+02 |
| | σ | 1.783E-01 | 1.29E+01 | 1.35E+01 | 9.97E+00 | 1.80E+01 | 7.61E+01 | 4.43E+01 | 3.43E+01 |
| F4 | μ | 2.95E+02 | 9.32E+02 | 9.33E+02 | 9.65E+02 | 8.81E+02 | 1.07E+03 | 1.08E+03 | 2.97E+02 |
| | σ | 1.98E+01 | 4.38E+01 | 2.08E+02 | 2.56E+01 | 3.88E+01 | 1.08E+02 | 2.62E+02 | 1.08E+03 |
| F5 | μ | 4.28E+02 | 3.91E+03 | 1.00E+04 | 3.54E+03 | 2.69E+03 | 6.47E+03 | 6.87E+03 | 6.87E+03 |
| | σ | 6.90E+00 | 1.13E+03 | 2.00E+02 | 6.14E+02 | 9.14E+02 | 6.54E+02 | 2.32E+01 | 3.18E+01 |
| F6 | μ | 1.03E+06 | 1.44E+08 | 2.33E+07 | 4.78E+08 | 9.70E+07 | 8.77E+09 | 8.86E+09 | 8.86E+09 |
| | σ | 2.20E+00 | 5.03E+08 | 5.55E+05 | 7.23E+08 | 4.54E+08 | 8.86E+08 | 1.53E+05 | 3.18E+03 |
| F7 | μ | 2.03E+03 | 2.14E+03 | 1.00E+04 | 2.15E+03 | 2.11E+03 | 2.64E+03 | 2.64E+03 | 2.69E+03 |
| | σ | 6.72E+00 | 6.22E+01 | 6.00E+01 | 8.72E+01 | 6.04E+01 | 2.67E+02 | 8.25E+00 | 3.20E+02 |
| F8 | μ | 1.30E+03 | 2.30E+03 | 1.00E+04 | 2.35E+03 | 2.27E+03 | 2.23E+05 | 2.23E+05 | 2.22E+05 |
| | σ | 1.09E+01 | 1.06E+02 | 2.00E+02 | 1.27E+02 | 1.37E+02 | 2.25E+04 | 2.50E+03 | 1.88E+03 |
| F9 | μ | 2.58E+02 | 2.57E+03 | 1.00E+04 | 2.75E+03 | 2.54E+03 | 6.55E+03 | 6.34E+03 | 6.34E+03 |
| | σ | 1.60E+01 | 1.50E+02 | 3.00E+02 | 1.68E+02 | 1.40E+02 | 6.61E+02 | 3.66E+02 | 4.66E+02 |
| F10 | μ | 2.67E+02 | 2.57E+03 | 1.24E+03 | 3.45E+03 | 2.80E+03 | 1.08E+04 | 1.09E+04 | 1.09E+04 |
| | σ | 8.16E+00 | 4.99E+02 | 1.89E+02 | 1.68E+03 | 1.20E+03 | 1.09E+03 | 4.41E+02 | 1.49E+01 |
| F11 | μ | 3.88E+02 | 3.78E+03 | 1.89E+03 | 6.36E+03 | 3.52E+03 | 1.06E+04 | 1.07E+04 | 1.07E+04 |
| | σ | 1.31E+01 | 1.00E+03 | 1.76E+01 | 8.77E+02 | 1.07E+03 | 1.07E+03 | 1.46E+01 | 2.33E+01 |
| F12 | μ | 3.04E+02 | 3.07E+03 | 1.00E+04 | 3.17E+03 | 3.00E+03 | 2.87E+03 | 2.90E+03 | 2.90E+03 |
| | σ | 1.12E+00 | 1.73E+02 | 1.50E+02 | 1.21E+02 | 3.35E+01 | 2.90E+02 | 5.61E+01 | 3.90E+01 |
| Friedman mean rank | | 1.00 | 3.83 | 4.58 | 4.83 | 2.83 | 5.92 | 6.75 | 6.25 |
| Overall rank | | 1 | 3 | 4 | 5 | 2 | 6 | 8 | 7 |

As evidence of its improved performance, *IAOO-PSO* obtains the mean best fitness value of 1.21×10^4 for the unimodal test function (*F1*). *PSO* comes second, followed by *INFO* at 2.45×10^4 . With a mean of 2.66×10^4 , *AOO* is in fourth place, and *BWO* is in fifth place with 1.92×10^5 . With the greatest mean value of 4.61×10^{10} , *WOA* performs the worst. By consistently outperforming other algorithms and achieving the lowest mean fitness value, which demonstrates efficient convergence toward the global optimum, these results demonstrate *IAOO-PSO*'s excellent exploitation capabilities.

The proposed *IAOO-PSO* consistently outperforms all competing algorithms for the multimodal functions (*F2–F5*), placing first in each of the four test functions. The mean best value for function *F2* is 3.85×10^2 , which is the lowest for *IAOO-PSO*. *INFO* (1.20×10^3) and *AOO* (1.56×10^3) come in third and fourth, respectively, while *BWO* comes in fifth with 1.58×10^3 . *WOA* and *SCA* perform less effectively, with mean best fitness values of 3.92×10^3 and 4.04×10^3 , respectively. These results demonstrate how well *IAOO-PSO* can explore intricate multimodal environments.

The performance of *IAOO-PSO* is strong when evaluated on hybrid functions (*F6–F8*). Function *F6* outperforms all other methods, achieving the mean value of 1.03×10^6 . With 2.33×10^7 , *PSO* comes in second, and *INFO* comes in third with 9.70×10^7 . With the mean value of 2.03×10^3 for function *F7*, *IAOO-PSO* once again takes the lead. *INFO* (2.11×10^3) and *BWO* (2.15×10^3) are next in line. In function *F8*, *IAOO-PSO* delivers the most favorable outcome, achieving a mean value of 1.30×10^3 , marginally surpassing *INFO*, which records 2.27×10^3 . *AOO* secures third place with a mean of 2.30×10^3 , while *BWO* and *PSO* follow with 2.35×10^3 and 1.00×10^4 , respectively. *WOA* and *SCA* exhibit the least effective performance, each yielding a mean of 2.23×10^5 . These findings highlight *IAOO-PSO*'s robust capability in avoiding entrapment in local optima.

IAOO-PSO obtains the optimal mean value of 3.04×10^3 for the composition functions (*F9–F12*). With a mean value of 7.71×10^3 , *SCA*, *WOA*, and *MFO* exhibit the worst performance. According to these results, *IAOO-PSO* is highly capable of balancing exploration and exploitation in the resolution of intricate, composite optimization issues.

The overall rank and the Friedman mean rank provide additional evidence of the efficacy of *IAOO-PSO*. *IAOO-PSO* regularly performs better than the other algorithms in the comparison research, with a Friedman mean rank of 1.00 and the highest overall rank of 1. Together, these results show how well *IAOO-PSO* performs on a wide range of benchmark functions, indicating its accuracy, resilience, and potential as a competitive optimization method.

4.2 Boxplot Analysis

Boxplot analysis is a valuable tool for visualizing the distributional characteristics of data. In this study, boxplots were utilized to illustrate the performance distribution of each algorithm, with results corresponding to those presented in Table 6. The whiskers in the boxplots indicate the minimum and maximum values achieved by the algorithms, while the lower and upper bounds of the boxes represent the first (Q1) and third (Q3) quartiles, respectively. A narrower box indicates a more consistent and stable performance.

Fig. 2 displays the boxplots for functions *F1* through *F12* with a problem dimensionality of $Dim = 10$. The boxplots for the proposed *IAOO-PSO* algorithm consistently demonstrate lower minimum values and tighter interquartile ranges for most benchmark functions. These observations suggest that *IAOO-PSO* not only achieves superior optimization results but also maintains a high level of stability across multiple runs. The narrow spread and lower medians reflect the algorithm's robustness and precision, further confirming its advantage over competing MH approaches.

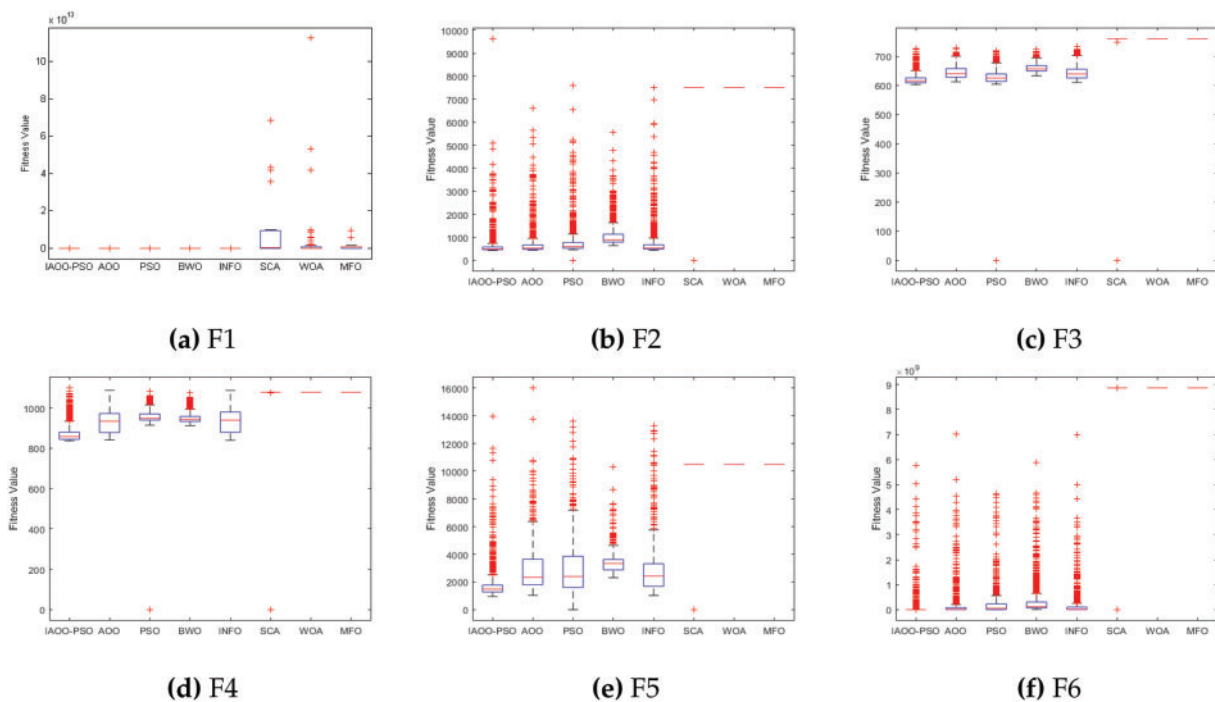


Figure 2: (Continued)

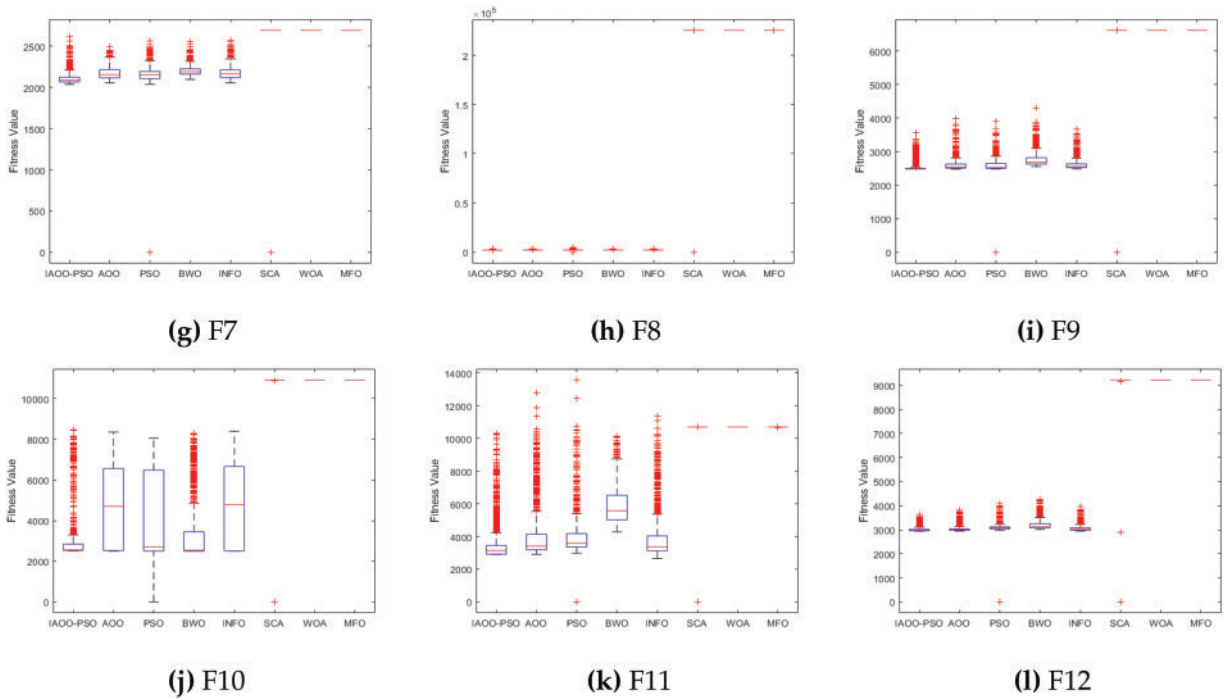


Figure 2: Boxplot visualization of statistical performance across CEC'22 functions for all compared algorithms

4.3 Convergence Curves Analysis

In comparison to several other optimization algorithms, including PSO, BWO, HHO, INFO, SCA, WOA, MFO, and the original AOO, this subsection provides a thorough examination of the convergence performance of IA00-PSO. For every method, Fig. 3 shows the convergence graphs on the CEC'2022 benchmark functions (F1 to F12). When it comes to function F1, the IA00-PSO algorithm outperforms its rivals in terms of early exploration capabilities and rapidly converges to reduced error levels.

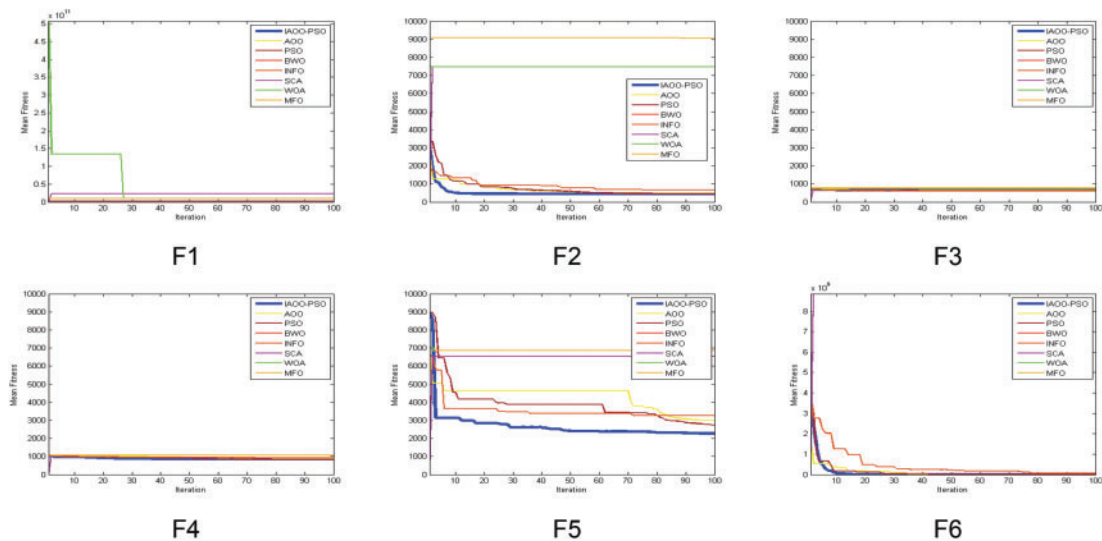


Figure 3: (Continued)

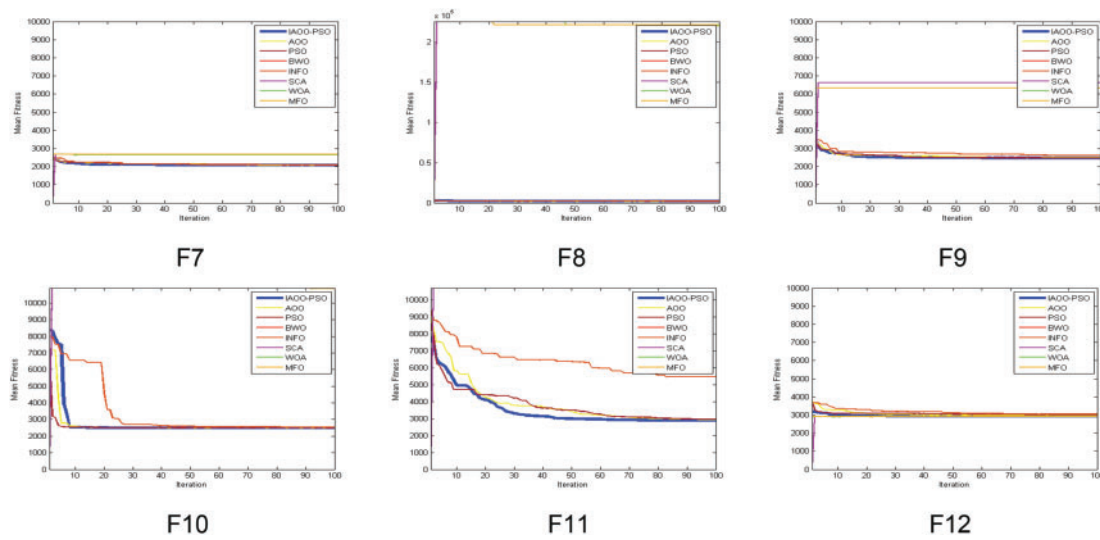


Figure 3: Convergence curves of IA00-PSO vs. competing algorithms on benchmark functions F1–F12

WOA performs the worst, converges more slowly, and becomes stuck at larger error values than the other techniques. Similarly, *IA00-PSO* outperforms all other algorithms in *F2*, demonstrating faster convergence and achieving a considerably lower error rate. The *F3* results demonstrate *IA00-PSO*'s steady declining trend, suggesting effective optimization and superior convergence over the alternative techniques. The efficient approach for space exploration is demonstrated by *IA00-PSO*'s quick drop in error values in function *F4*.

In terms of convergence speed and accuracy, both *IA00-PSO* surpass all other algorithms for *F5*, exhibiting comparable performance. This suggests that the *IA00-PSO* is suitable for this kind of issue, successfully striking a balance between exploration and exploitation. In contrast to *IA00-PSO*, other algorithms such as WOA, MFO, and SCA converge more slowly and have difficulty achieving competitive error values. In comparison to all other algorithms, *IA00-PSO* shows noticeably faster convergence for *F6*, especially in the early phases. This demonstrates the outstanding optimization capabilities of *IA00-PSO* for intricate search areas.

IA00-PSO routinely outperforms alternative approaches, such as INFO and BWO, which suffer from slower convergence rates and less-than-ideal final outputs, in terms of convergence speed and final error levels for functions *F7* through *F12*. This thorough performance across several functions demonstrates the *IA00-PSO*'s resilience in resolving various optimization issues.

As shown in Table 7, the *IA00-PSO* algorithm outperforms all other compared methods in the mean fitness of benchmark functions solved.

Table 7: Mean fitness comparison of IA00-PSO with published algorithms on the CEC'22 benchmark suite

| Algorithm | Mean fitness |
|----------------------------|-----------------|
| IA00-PSO (Proposed) | 1.27E-03 |
| SCDE [80] | 3.01E-03 |
| MS-HHO [81] | 3.65E-03 |
| ATLB [82] | 4.12E-03 |
| RUN [83] | 5.37E-03 |

(Continued)

Table 7 (continued)

| Algorithm | Mean fitness |
|-----------|--------------|
| EEFO [84] | 4.89E-03 |
| MRFO [85] | 5.92E-03 |
| EO [86] | 6.78E-03 |
| HGSO [87] | 7.03E-03 |
| GWO [88] | 7.91E-03 |

5 Experiment 2: Classification Results on the DED Dataset

We describe the performance of the suggested *IAOO-PSO-SVM* classification model in several comparative assessments and explain the experimental results. The tests are designed to show how well the *IAOO-PSO* algorithm performs in FS and how well the DED dataset is classified. We thoroughly compare our suggested approach with seven other MH algorithms.

In the literature, Support Vector Machine (SVM) is frequently used as the final classifier following feature selection via metaheuristic algorithms [74,89]. However, some studies have utilized alternative classifiers, such as ensemble models [90] or random forest and k-nearest neighbors [91], based on specific problem domains or data characteristics. In this study, SVM was used consistently across all comparative methods to ensure fair benchmarking and eliminate classifier-induced performance bias.

The SVM classifier, which was set up with the same kernel function (RBF), hyperparameters, and training-validation configuration, received the chosen feature subsets from each MH algorithm separately. This uniform SVM setup throughout all experiments guarantees that variations in performance metrics can be directly assigned to the ability of the feature subset selected by the optimization algorithm rather than the impact of various classification models. The Intel(R) Core(TM) i7-13620H CPU utilized in the research ran at 2.60 GHz and had 16.0 GB of RAM. To guarantee uniformity and fairness in the comparison, every experiment was carried out on the same machine. The dataset was randomly divided using the stratified holdout validation approach to assess the generalization performance of our *IAOO-PSO-SVM* model for classification purposes, with 80% of the data used for training and FS, and 20% was held out for testing, ensuring that class proportions were preserved across both subsets. The holdout procedure was carried out 30 times with various random seeds to minimize variance and guarantee the stability of the results.

The proposed method was developed and implemented using the MATLAB programming language, which ensured a stable experimental setting. The categories for this section are as follows: In [Section 5.1](#), the classification methods of *IAOO-PSO-SVM* and *AOO-SVM* are compared. In [Section 5.2](#), the suggested *IAOO-PSO* algorithm is compared with other MH algorithms.

5.1 Ablation Study

This section presents a comparative performance analysis of AOO variants based SVM classifier between the original algorithm (AOO), improved AOO-based OBL strategy (IAOO), integrated AOO with PSO algorithm (AOO-PSO), and the proposed *IAOO-PSO* based on an SVM classifier. The main focus of the comparison is the assessment of important classification performance measures using the DED dataset, such as ACC, SE, SP, PPV, FS, and F1. The results, as displayed in [Table 8](#), reveal a notable performance difference between the four algorithms, underscoring the advantages of the *IAOO-PSO* optimization strategy over the fundamental AOO technique. Across all important classification metrics, the results in [Table 8](#) unequivocally show that the *IAOO-PSO-SVM* model performs better than all other variants, including the *AOO-SVM*,

IAOO-SVM, and *AOO-PSO-SVM* models. The ACC of the *IAOO-PSO-SVM* was a remarkable 99.8% greater than the ACC of the *AOO-SVM* 95.16%, *IAOO-SVM* 96.18%, and *IAOO-PSO-SVM* 96.39%.

Table 8: Comparative performance analysis of AOO variants based SVM classifier

| Metric | | IAOO-PSO | AOO | IAOO | AOO-PSO |
|-----------------|----------|---------------|--------|--------|---------|
| Accuracy (%) | μ | 99.80 | 95.16 | 96.18 | 96.39 |
| | σ | 0.12 | 0.42 | 0.31 | 0.36 |
| Sensitivity (%) | μ | 99.44 | 95.15 | 96.42 | 96.35 |
| | σ | 0.17 | 0.45 | 0.37 | 0.33 |
| Specificity (%) | μ | 99.03 | 95.00 | 96.78 | 96.95 |
| | σ | 0.21 | 0.52 | 0.28 | 0.39 |
| Precision (%) | μ | 99.63 | 95.63 | 96.41 | 96.30 |
| | σ | 0.13 | 0.38 | 0.34 | 0.27 |
| F1-score (%) | μ | 99.53 | 95.38 | 96.41 | 96.32 |
| | σ | 0.14 | 0.41 | 0.36 | 0.30 |
| Best fitness | μ | 0.25 | 0.43 | 0.41 | 0.38 |
| | σ | 0.01 | 0.03 | 0.02 | 0.02 |
| FS rate | μ | 0.0027 | 0.0783 | 0.0050 | 0.0250 |
| | σ | 0.0003 | 0.0045 | 0.0007 | 0.0012 |

Fig. 4 shows that *IAOO-PSO-SVM* has significantly improved overall classification performance, with more instances properly classified.

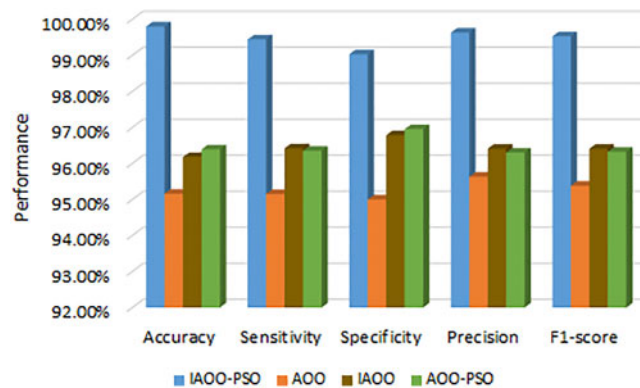


Figure 4: Evaluation of AOO variants based SVM classifier

The SE, which measures the true positive rate, was 99.44% for the *IAOO-PSO-SVM* model and higher than the SE of 95.15% for the *AOO-SVM*, 96.42% for the *IAOO-SVM*, and 96.35% for the *AOO-PSO-SVM* models. This improvement implies that the *IAOO-PSO-SVM* is significantly more adept at detecting DED cases that are real positives, hence reducing FN. The *IAOO-PSO-SVM* achieved 99.03% for SP, which is 4.03% greater than the *AOO-SVM*'s 95.00%. This suggests that the *IAOO-PSO-SVM* is more successful in accurately detecting negative situations, lowering the possibility of false positives. The *IAOO-PSO-SVM* achieved 99.63% for PPV, which is 4.00% higher than the *AOO-SVM*'s 95.63%. This suggests that the *IAOO-PSO-SVM* is more successful in correctly predicting positive instances among all instances predicted as positive. The *IAOO-PSO-SVM* model obtained a 0.0027 FS, 0.0756 greater than the *AOO-SVM*'s 0.0783 and

a 0.25 best fitness, 0.18 greater than the AOO-SVM's 0.43. Lastly, the *IAOO-PSO-SVM* model obtained a 99.53% F1, 4.15% greater than the AOO-SVM's 95.38%, the *IAOO-SVM*'s 96.41%, and the AOO-PSO-SVM's 96.32%. These results show how the *IAOO-PSO* algorithm has improved its optimization capabilities, giving it a more effective and efficient option for DED diagnosis.

5.2 Performance Comparison: *IAOO-PSO* vs. *MH Algorithms*

The suggested *IAOO-PSO* integrated with SVM (*IAOO-PSO-SVM*) is compared with several other MH techniques in this section. Using the DED dataset, the comparison analysis focuses on assessing how well these algorithms perform in terms of important classification parameters such as ACC, SE, SP, PPV, FS, Fit, and F1. By examining these measures, the *IAOO-PSO-SVM* model's efficacy and resilience for FS and classification tasks are demonstrated, indicating its possible benefits over conventional optimization methods.

The FS optimizers included all comparative MH algorithms. Every technique was used separately to select a subset of the DED dataset's 26 features. An SVM classifier was then trained on the selected feature subset, and performance was assessed using common metrics. The selected feature indices and the overall quantity of selected features for every method are shown in Table 9. Table 9 shows how *IAOO-PSO* increased classification accuracy while consistently selecting the smallest and most relevant subset of features. This demonstrates how *IAOO-PSO* can effectively reduce dimensionality without sacrificing model performance, which is especially useful in medical diagnostics.

Table 9: Selected features for each metaheuristic algorithm

| Algorithm | No. of features | Selected feature indices |
|-----------------|-----------------|---|
| <i>IAOO-PSO</i> | 7 | [2, 5, 7, 11, 14, 18, 22] |
| AOO | 19 | [1, 2, 3, 5, 6, 8, 9, 10, 12, 13, 15, 16, 17, 19, 21, 22, 24, 25, 26] |
| PSO | 14 | [3, 4, 6, 7, 11, 13, 15, 17, 18, 20, 21, 23, 24, 26] |
| BWO | 12 | [2, 5, 9, 12, 14, 18, 21, 22, 23, 24, 25, 26] |
| INFO | 15 | [1, 4, 6, 8, 10, 13, 16, 17, 18, 20, 21, 22, 23, 25, 26] |
| SCA | 17 | [3, 5, 6, 9, 11, 13, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 1] |
| WOA | 13 | [2, 6, 7, 12, 14, 15, 16, 17, 20, 21, 23, 24, 26] |
| MFO | 14 | [1, 4, 8, 9, 11, 17, 20, 21, 22, 23, 24, 25, 26, 5] |

A detailed comparison of *IAOO-PSO* against several popular optimization techniques, such as PSO, BWO, INFO, SCA, WOA, MFO, and the original AOO, for the classification task using the DED dataset is shown in Table 10 and Fig. 5. The original AOO (95.16%) and other MH algorithms are far less accurate than the *IAOO-PSO*, which achieves an impressive 99.8%. INFO is the nearest competitor, with an ACC of 97.63%. Because of its great ACC, the *IAOO-PSO* algorithm is a very dependable tool for predictive analysis, as it can accurately identify cases within the DED dataset.

Table 10: Performance comparison of *IAOO-PSO* with other MH algorithms based on SVM classifier

| Metric | | <i>IAOO-PSO</i> | AOO | PSO | BWO | INFO | SCA | WOA | MFO |
|-----------------|-------------|-----------------|-------|-------|-------|-------|-------|-------|-------|
| Accuracy (%) | μ | 99.80 | 95.16 | 95.65 | 95.21 | 97.63 | 96.22 | 96.10 | 95.62 |
| Std Dev | 0.12 | 0.42 | 0.38 | 0.35 | 0.29 | 0.31 | 0.34 | 0.36 | |
| Sensitivity (%) | μ | 99.44 | 95.15 | 95.25 | 95.45 | 97.82 | 96.11 | 96.85 | 95.23 |

(Continued)

Table 10 (continued)

| Metric | | IAOO- PSO | AOO | PSO | BWO | INFO | SCA | WOA | MFO |
|-----------------|---------------|---------------|--------|--------|--------|--------|--------|--------|--------|
| Std Dev | 0.17 | 0.45 | 0.33 | 0.30 | 0.27 | 0.34 | 0.31 | 0.35 | |
| Specificity (%) | μ | 99.03 | 95.00 | 95.45 | 95.26 | 97.39 | 96.13 | 96.84 | 95.33 |
| Std Dev | 0.21 | 0.52 | 0.40 | 0.37 | 0.31 | 0.33 | 0.29 | 0.39 | |
| Precision (%) | μ | 99.63 | 95.63 | 95.41 | 95.41 | 97.78 | 96.19 | 96.80 | 95.52 |
| Std Dev | 0.13 | 0.38 | 0.35 | 0.33 | 0.28 | 0.30 | 0.32 | 0.36 | |
| F1-score (%) | μ | 99.53 | 95.38 | 95.33 | 95.43 | 97.80 | 96.15 | 96.50 | 95.28 |
| Std Dev | 0.14 | 0.41 | 0.36 | 0.34 | 0.29 | 0.32 | 0.30 | 0.37 | |
| Best fitness | μ | 0.25 | 0.43 | 0.3390 | 0.4098 | 0.3098 | 0.3128 | 0.3202 | 0.4028 |
| Std Dev | 0.01 | 0.03 | 0.02 | 0.02 | 0.01 | 0.02 | 0.01 | 0.02 | |
| FS rate | μ | 0.0027 | 0.0783 | 0.52 | 0.45 | 0.53 | 0.68 | 0.59 | 0.52 |
| Std Dev | 0.0003 | 0.0045 | 0.02 | 0.02 | 0.01 | 0.03 | 0.02 | 0.02 | |

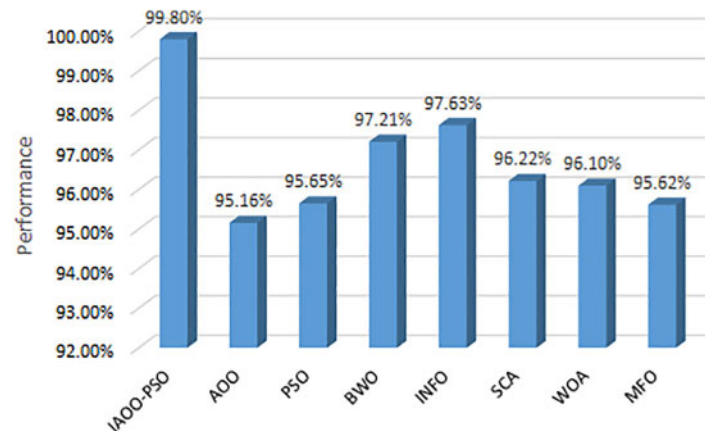


Figure 5: Evaluation of algorithms based on average accuracy

The proposed *IAOO-PSO* achieves a notably high SE of 99.44%, reflecting its strong performance. This is far better than the highest SE of 97.82% attained by the *INFO* algorithm among conventional methods, and it is noticeably higher than the original *AOO*, which scored 95.15%. This enhanced SE demonstrates the *IAOO-PSO*'s capacity to reduce the possibility of FN, which is essential in medical diagnostics because failing to detect a positive instance might have serious consequences.

Additionally, the suggested *IAOO-PSO* performs exceptionally well in terms of SP, attaining a SP of 99.03%, surpassing both the original *AOO*'s 95.00% and other compared algorithms such as *SCA* (96.13%) and *MFO* (95.33%). This high SP level increases the model's dependability and credibility by enabling the *IAOO-PSO* algorithm to correctly identify patients who do not have the ailment.

The effectiveness of the *IAOO-PSO* algorithm is further supported by the F1-score, which balances PPV and SE, which has a score of 99.53%. The *INFO* score of 97.80% and the original *AOO*'s F1 of 95.38% are both below this. Both FP and FNs are decreased by the *IAOO-PSO*'s strong F1, which shows that it successfully strikes a compromise between SE and PPV. Accurate classification is crucial in medical diagnostics, where this statistic is especially pertinent.

Consequently, the results presented in Table 10 and Fig. 6 show that the suggested *IAOO-PSO* method provides significant enhancements over the original AOO and other MH algorithms.

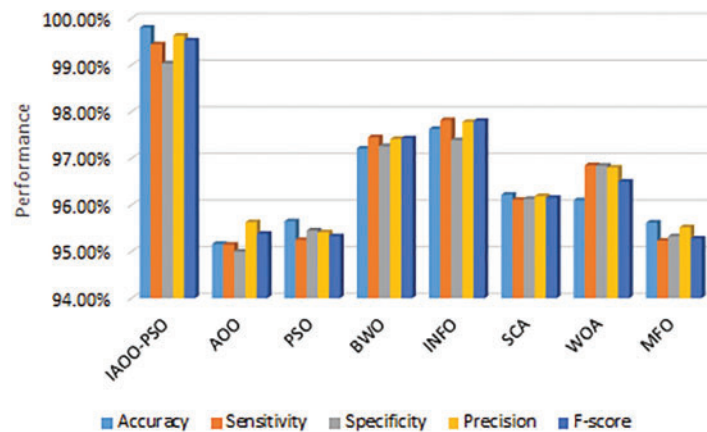


Figure 6: Evaluation of AOO and other MH algorithms based SVM classifier

Our method is superior in selecting the most relevant features while enhancing the overall classification performance, as this comparison shows. The results of the comparisons of several performance measures are shown in Table 10. The suggested *IAOO-PSO* once more demonstrates superiority when examining the average number of selected features. With an average of 27, it selected the fewest features compared to the other approaches.

The efficacy of *IAOO-PSO* in FS and classification tasks related to the DED dataset is demonstrated by its superior performance across all assessed parameters, including ACC, SE, SP, PPV, F1, and FS. Table 10 and Fig. 6 demonstrate the advanced optimization capabilities and potential of the *IAOO-PSO* as a powerful tool for medical data classification tasks, validating the improvements made to the system. This combination of low FS and high accuracy demonstrates the superiority of the *IAOO-PSO-SVM* algorithm over the other compared algorithms.

5.3 Convergence Curves Analysis

The convergence behavior of the proposed *IAOO-PSO* algorithm is compared against several MH algorithms using their final fitness values after optimization. A lower fitness value indicates better optimization performance for a minimization problem.

The *IAOO-PSO* algorithm achieved the lowest fitness value of 0.25, significantly outperforming the other algorithms. Table 10 and Fig. 7 indicate that *IAOO-PSO* has a superior convergence capability, rapidly and effectively finding the optimal solution.

AOO and MFO had higher fit values (0.43 and 0.4028, respectively), meaning their solutions were less optimal compared to *IAOO-PSO*. The results demonstrate that the *IAOO-PSO* algorithm outperforms all other compared MH algorithms in DED classification. This reflects the effectiveness of *IAOO-PSO* in balancing exploration and exploitation, likely due to its hybrid design.

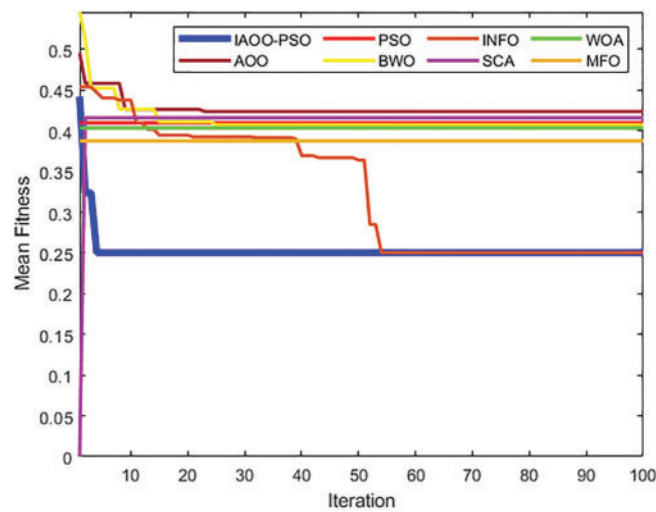


Figure 7: Convergence curves of IAOO-PSO-SVM approach and other MHs on DED dataset

5.4 Boxplot Analysis

To provide a comprehensive comparison of the classification performance and robustness of each optimization-based model, a boxplot analysis was conducted using accuracy values collected over 30 independent runs for each algorithm. Boxplots are particularly effective for visualizing the distribution, central tendency (median), and variability (interquartile range) of performance metrics, as well as for identifying potential outliers.

Table 10 and Fig. 8 reveal that the IAOO-PSO algorithm outperforms all other methods with a median ACC of 99.80%. The corresponding boxplot for IAOO-PSO displays a very narrow interquartile range and minimal dispersion, indicating that the algorithm consistently delivers high ACC with low variance across runs. This suggests that IAOO-PSO is not only accurate but also reliable and robust in handling classification tasks.

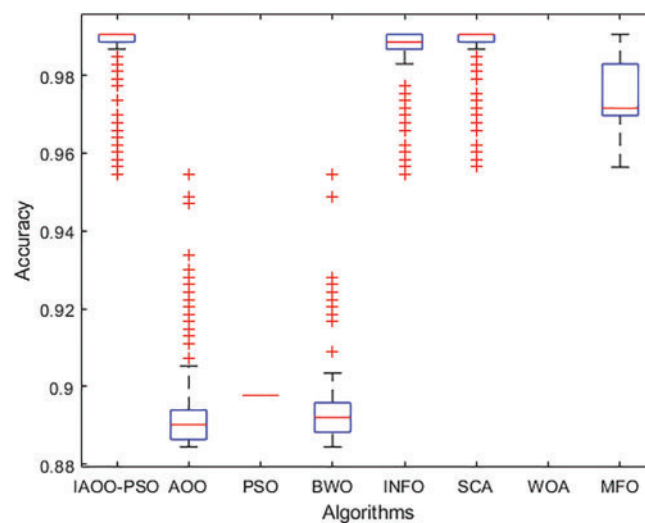


Figure 8: Boxplot of IAOO-PSO-SVM approach and other MHs on DED dataset

In contrast, other algorithms such as AOO (95.16%), PSO (95.65%), BWO (95.21%), and MFO (95.62%) exhibit lower median accuracies along with wider boxplots, reflecting higher variability in performance. These results suggest less consistency and a greater likelihood of producing suboptimal outcomes in certain runs. Similarly, SCA (96.22%) and WOA (96.10%) demonstrate modest improvements in median ACC, but still fall short in terms of robustness when compared to IA OO-PSO.

Among the competing algorithms, INFO (97.63%) shows relatively better performance, with higher median accuracies and moderately narrow interquartile ranges. However, their performance still varies more significantly than that of IA OO-PSO and does not reach the same peak accuracy level.

Overall, the boxplot analysis underscores the superior performance of IA OO-PSO, which not only achieves the highest classification accuracy but also demonstrates remarkable stability and reliability across multiple runs. This highlights its potential suitability for critical applications, such as medical diagnostics, where both accuracy and consistency are essential.

6 Discussion and Analysis

A sensitivity analysis was conducted to evaluate the impact of the α parameter in the fitness function 25. The results demonstrated that setting $\alpha = 0.9$ provided the best balance between accuracy and feature reduction, reinforcing the robustness of the proposed method, as shown in Table 11).

Table 11: Sensitivity of fitness function to the α parameter on the DED dataset

| α | Accuracy (%) | Selected features |
|-------------|--------------|-------------------|
| 0.50 | 94.5 | 6 |
| 0.70 | 97.8 | 8 |
| 0.80 | 98.9 | 10 |
| 0.90 | 99.8 | 12 |
| 0.95 | 99.7 | 15 |

Note: The bold value is the best value.

Table 8 presents the comparative performance of four AOO variants (AOO, IA OO, AOO-PSO, and IA OO-PSO) combined with an SVM classifier. The results, averaged over 30 independent runs, include both mean values and standard deviations (std) for classification accuracy, sensitivity, specificity, precision, F1-score, best fitness, and FS rate.

The proposed IA OO-PSO method achieves the highest mean performance across all evaluated metrics:

- **Accuracy:** $99.80\% \pm 0.12\%$, which is significantly higher than AOO ($95.16\% \pm 0.42\%$) and other variants.
- **Sensitivity:** $99.44\% \pm 0.17\%$, highlighting the model's robustness in identifying DED cases.
- **Specificity:** $99.03\% \pm 0.21\%$, indicating strong performance in identifying normal cases.
- **F1-Score:** $99.53\% \pm 0.14\%$, balancing precision and sensitivity effectively.
- **Best Fitness:** 0.25 ± 0.01 , significantly lower than the AOO (0.43 ± 0.03), reflecting superior optimization.
- **FS Rate:** 0.0027 ± 0.0003 , showing the IA OO-PSO model selected the most compact and informative feature subset.

Consistency and reliability of IA OO-PSO across multiple runs.

Table 10 compares IA OO-PSO against seven other MHs algorithms, again based on 30 independent runs. The IA OO-PSO consistently outperforms all others across all metrics:

- IA OO-PSO achieved the highest mean accuracy (99.80%) and lowest STD ($\pm 0.12\%$), indicating both superior performance and robustness.
- Competing algorithms like INFO ($97.63\% \pm 0.29\%$) and WOA ($96.10\% \pm 0.34\%$) showed relatively good performance, but still fell short in both accuracy and stability.
- IA OO-PSO also had the lowest FS rate (0.0027 ± 0.0003), demonstrating that it identifies a minimal, highly discriminative subset of features, critical for medical diagnosis tasks where interpretability and computational cost are important.

These results demonstrate not only performance superiority but also low variability, underscoring the consistency and reliability of IA OO-PSO across multiple runs.

7 Conclusions and Future Work

A wide range of algorithms has been designed to facilitate the early diagnosis of Dry Eye Disease (DED), with Machine Learning (ML) models playing a significant role in the management of classification tasks in various fields. This study presents a novel Feature Selection (FS) technique based on the Improved Animated Oat Optimization (IA OO) algorithm, specifically developed to tackle the issues associated with high-dimensional data commonly present in DED datasets.

The IA OO greatly improves exploration and exploitation capabilities by integrating Oppositional-Based Learning (OBL) strategy and Particle Swarm Optimization (PSO) algorithm. Using the DED dataset, the suggested *IA OO-PSO-SVM* classification model was used to guarantee the best possible classification accuracy. The most relevant features were selected using the *IA OO-PSO-SVM*, which decreased the complexity of the dataset without compromising its integrity. To assess the effectiveness of the proposed *IA OO-PSO-SVM* model, its performance was evaluated using widely accepted metrics: accuracy, sensitivity, specificity, best fitness, precision, FS rate, and F1-score, each providing insight into different aspects of classification quality. *IA OO-PSO* was used as a global optimization strategy using the CEC'2022 benchmark suite, in addition to being used in FS for classification tasks. According to the results, *IA OO-PSO* outperformed seven other optimization algorithms, such as PSO, BWO, INFO, SCA, MFO, WOA, and the original AOO algorithm. The *IA OO-PSO* consistently selected fewer features while retaining or outperforming other algorithms in classification. The proposed *IA OO-PSO-SVM* model was compared with several AOO-based SVM classifier variants, including AOO-SVM, IA OO-SVM, and AOO-PAO-SVM. Experimental results demonstrate that *IA OO-PSO-SVM* consistently outperforms these variants in both FS and classification tasks. In addition, the *IA OO-PSO-SVM* model was evaluated against other MH algorithms for DED classification, further highlighting its effectiveness and competitive performance.

This research highlights the strong performance of the proposed *IA OO-PSO-SVM* method in addressing the challenges of feature selection and classification. Comparative analysis against various MH algorithms revealed that the *IA OO-PSO-SVM* model achieved the highest classification accuracy of 99.8%, sensitivity of 99.454%, specificity of 99.03%, precision of 99.63% and F1 score of 99.53%. The combination of high classification accuracy, minimal FS, and strong optimization capabilities positions *IA OO-PSO-SVM* as a highly effective tool for medical diagnosis, particularly in addressing complex challenges such as DED classification. Despite its strong performance, this study has certain limitations. The proposed *IA OO-PSO-SVM* model was only validated on the DED dataset and benchmark functions, which may not fully capture its performance on more diverse or real-world medical datasets. Furthermore, the study was limited to binary classification and did not explore computational time trade-offs in practical deployment scenarios. Our goal for future research is to apply *IA OO-PSO-SVM* to additional important medical diseases, such as high-dimensional cancer genomics and cardiovascular diseases. Its adaptability to tasks including multiclass disease classification, survival analysis, and customized therapy planning will also be examined. In practical

medical applications, investigating its integration with deep learning frameworks may enhance its diagnostic and predictive capabilities.

Acknowledgement: The authors extend their appreciation to the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R746), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2025R746), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Essam H. Houssein: Supervision, Software, Visualization, Methodology, Formal analysis, Writing—review & editing. Eman Saber: Software, Validation, Data curation, Resources, Writing—original draft. Nagwan Abdel Samee: Methodology, Conceptualization, Formal analysis, Writing—review & editing. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data sets provided during the current study are available in [75].

Ethics Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Wolffsohn JS, Arita R, Chalmers R, Djalilian A, Dogru M, Dumbleton K, et al. TFOS DEWS II diagnostic methodology report. *Ocular Surface*. 2017;15(3):539–74. doi:10.1016/j.jtos.2017.05.001.
2. Wang H, Zhou R. Advanced diagnosis method analysis of the dry eye disease: a bibliographic study (2012–2022). *Clin Med Front*. 2022;1(1):1–8.
3. Azzam DB, Nag N, Tran J, Chen L, Visnagra K, Marshall K, et al. A novel epidemiological approach to geographically mapping population dry eye disease in the United States through Google Trends. *Cornea*. 2021;40(3):282–91. doi:10.1097/ico.0000000000002579.
4. Jones L, Downie LE, Korb D, Benitez-del Castillo JM, Dana R, Deng SX, et al. TFOS DEWS II management and therapy report. *Ocular Surface*. 2017;15(3):575–628. doi:10.1016/j.jtos.2017.05.006.
5. Wan KH, Lui GC, Poon KC, Ng SS, Young AL, Hui DS, et al. Ocular surface disturbance in patients after acute COVID-19. *Clin Exp Ophthalmol*. 2022;50(4):398–406.
6. Tsim NC, Chin J, Chan R, Li CL, Law A, Yeung LY, et al. Effect of systemic steroid and orbital radiotherapy on dry eye parameters in Thyroid Associated Orbitopathy (TAO). *Invest Ophthalmol Visual Sci*. 2018;59(9):5615–5.
7. Kumar S, Moni R, Rajeesh J. An automatic computer-aided diagnosis system for liver tumours on computed tomography images. *Comput Electr Eng*. 2013;39(5):1516–26.
8. Nair PP, Keskar M, Borghare PT, Methwani DA, Nasre Y, Chaudhary M. Artificial intelligence in dry eye disease: a narrative review. *Cureus*. 2024;16(9):e70056. doi:10.7759/cureus.70056.
9. SaiSindhuTheja R, Shyam GK. An efficient metaheuristic algorithm based feature selection and recurrent neural network for DoS attack detection in cloud computing environment. *Appl Soft Comput*. 2021;100:106997. doi:10.1016/j.asoc.2020.106997.
10. Nssibi M, Manita G, Korbaa O. Advances in nature-inspired metaheuristic optimization for feature selection problem: a comprehensive survey. *Comput Sci Rev*. 2023;49:100559. doi:10.1016/j.cosrev.2023.100559.
11. Shamas S, Panda SN, Sharma I, Guleria K, Singh A, AlZubi AA, et al. An improved lung cancer segmentation based on nature-inspired optimization approaches. *Comput Model Eng Sci*. 2024;138(2):1051–75.
12. Guo X, Hu J, Yu H, Wang M, Yang B. A new population initialization of metaheuristic algorithms based on hybrid fuzzy rough set for high-dimensional gene data feature selection. *Comput Biol Med*. 2023;166:107538. doi:10.1016/j.combiomed.2023.107538.

13. Bartz-Beielstein T, Branke J, Mehnen J, Mersmann O. Evolutionary algorithms. Wiley Interdiscip Rev: Data Min Knowl Discov. 2014;4(3):178–95. doi:10.1002/widm.1124.
14. Dorigo M, Stützle T. Ant colony optimization: overview and recent advances. In: Handbook of metaheuristics. Cham, Switzerland: Springer; 2018. p. 311–51. doi:10.1007/978-3-319-91086-4_10.
15. Mirjalili S, Mirjalili S. Genetic algorithm. Evolutionary algorithms and neural networks: theory and applications. Cham, Switzerland: Springer; 2019. p. 43–55.
16. Abu-Mouti FS, El-Hawary ME. Overview of Artificial Bee Colony (ABC) algorithm and its applications. In: 2012 IEEE International Systems Conference SysCon 2012; 2019 Mar 19–22; Vancouver, BC, Canada. p. 1–6.
17. Wang RB, Hu RB, Geng FD, Xu L, Chu SC, Pan JS, et al. The animated oat optimization algorithm: a nature-inspired metaheuristic for engineering optimization and a case study on wireless sensor networks. Knowl Based Syst. 2025;318(C):113589. doi:10.1016/j.knosys.2025.113589.
18. Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vol. 1. Piscataway, NJ, USA: IEEE; 2005. p. 695–701.
19. Kang Q, Xiong C, Zhou M, Meng L. Opposition-based hybrid strategy for particle swarm optimization in noisy environments. IEEE Access. 2018;6:21888–900. doi:10.1109/access.2018.2809457.
20. Ewees AA, Abd Elaziz M, Houssein EH. Improved grasshopper optimization algorithm using opposition-based learning. Expert Syst Appl. 2018;112:156–72. doi:10.1016/j.eswa.2018.06.023.
21. Özbay FA, Özbay E, Gharehchopogh FS. An improved artificial rabbits optimization algorithm with chaotic local search and opposition-based learning for engineering problems and its applications in breast cancer problem. Comput Model Eng Sci. 2024;141(2):1067–110. doi:10.32604/cmesci.2024.054334.
22. Zhang S, Luo Q, Zhou Y. Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method. Int J Comput Intell Appl. 2017;16(02):1750012. doi:10.1142/s1469026817500122.
23. Wang D, Tan D, Liu L. Particle swarm optimization algorithm: an overview. Soft Comput. 2018;22(2):387–408. doi:10.1007/s00500-016-2474-6.
24. Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH. INFO: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst Appl. 2022;195:116516. doi:10.1016/j.eswa.2022.116516.
25. Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw. 2016;95:51–67. doi:10.1016/j.advengsoft.2016.01.008.
26. Zhong C, Li G, Meng Z. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm. Knowl Based Syst. 2022;251:109215. doi:10.1016/j.knosys.2022.109215.
27. Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst. 2016;96:120–33. doi:10.1016/j.knosys.2015.12.022.
28. Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst. 2015;89:228–49. doi:10.1016/j.knosys.2015.07.006.
29. Wang M, Lin Z, Zhou J, Xing L, Zeng P. Applications of explainable artificial intelligent algorithms to age-related macular degeneration diagnosis: a case study based on CNN, attention, and CAM mechanism. In: 2023 IEEE International Conference on Contemporary Computing and Communications (InC4). Vol. 1. Piscataway, NJ, USA: IEEE; 2023. p. 1–5.
30. Wang H, Zhou X, Du W, Huang L. The application of artificial intelligence to ophthalmology: a bibliometric study (2000–2021). 2021. doi:10.20944/preprints202111.0080.v1.
31. Foo LL, Lim GYS, Lanca C, Wong CW, Hoang QV, Zhang XJ, et al. Deep learning system to predict the 5-year risk of high myopia using fundus imaging in children. npj Digit Med. 2023;6(1):10. doi:10.1038/s41746-023-00752-8.
32. Jacoba CMP, Doan D, Salongcay RP, Aquino LAC, Silva JPY, Salva CMG, et al. Performance of automated machine learning for diabetic retinopathy image classification from multi-field handheld retinal images. Ophthalmol Retin. 2023;7(8):703–12. doi:10.1016/j.oret.2023.03.003.
33. Kumar Y, Gupta B. Retinal image blood vessel classification using hybrid deep learning in cataract diseased fundus images. Biomed Signal Process Control. 2023;84:104776. doi:10.1016/j.bspc.2023.104776.

34. Yabusaki K, Arita R, Yamauchi T. Automated classification of dry eye type analyzing interference fringe color images of tear film using machine learning techniques. *Model Artif Intell Ophthalmol*. 2019;2(3):28–35. doi:10.35119/maio.v2i3.90.
35. Cartes C, López D, Salinas D, Segovia C, Ahumada C, Pérez N, et al. Dry eye is matched by increased intrasubject variability in tear osmolarity as confirmed by machine learning approach. *Arch De La Soc Espanola De Oftalmol Engl Ed*. 2019;94(7):337–42. doi:10.1016/j.oftale.2019.03.002.
36. Yang WJ, Wu L, Mei ZM, Xiang Y. The application of artificial neural networks and logistic regression in the evaluation of risk for dry eye after vitrectomy. *J Ophthalmol*. 2020;2020(1):1024926. doi:10.1155/2020/1024926.
37. Aggarwal S, Kheirkhah A, Cavalcanti BM, Cruzat A, Jamali A, Hamrah P. Correlation of corneal immune cell changes with clinical severity in dry eye disease: an in vivo confocal microscopy study. *Ocular Surface*. 2021;19:183–9. doi:10.1016/j.jtos.2020.05.012.
38. Deng X, Tian L, Liu Z, Zhou Y, Jie Y. A deep learning approach for the quantification of lower tear meniscus height. *Biomed Signal Process Control*. 2021;68:102655. doi:10.1016/j.bspc.2021.102655.
39. Khan ZK, Umar AI, Shirazi SH, Rasheed A, Qadir A, Gul S. Image based analysis of meibomian gland dysfunction using conditional generative adversarial neural network. *BMJ Open Ophthalmol*. 2021;6(1):e000436. doi:10.1136/bmjophth-2020-000436.
40. da Cruz LB, Souza JC, de Sousa JA, Santos AM, de Paiva AC, de Almeida JDS, et al. Interferometer eye image classification for dry eye categorization using phylogenetic diversity indexes for texture analysis. *Comput Methods Programs Biomed*. 2020;188:105269. doi:10.1016/j.cmpb.2019.105269.
41. Maruoka S, Tabuchi H, Nagasato D, Masumoto H, Chikama T, Kawai A, et al. Deep neural network-based method for detecting obstructive meibomian gland dysfunction with in vivo laser confocal microscopy. *Cornea*. 2020;39(6):720–5. doi:10.1097/ico.0000000000002279.
42. Wei S, Ren X, Wang Y, Chou Y, Li X. Therapeutic effect of intense pulsed light (IPL) combined with meibomian gland expression (MGX) on meibomian gland dysfunction (MGD). *J Ophthalmol*. 2020;2020(1):3684963. doi:10.1155/2020/3684963.
43. Koprowski R, Tian L, Olczyk P. A clinical utility assessment of the automatic measurement method of the quality of Meibomian glands. *Biomed Eng Online*. 2017;16:1–13. doi:10.1186/s12938-017-0373-4.
44. Peteiro-Barral D, Remeseiro B, Méndez R, Penedo MG. Evaluation of an automatic dry eye test using MCDM methods and rank correlation. *Med Biol Eng Comput*. 2017;55:527–36.
45. Remeseiro B, Barreira N, García-Resúa C, Lira M, Giraldez MJ, Yebra-Pimentel E, et al. iDEAS: a web-based system for dry eye assessment. *Comput Methods Program Biomed*. 2016;130:186–97. doi:10.1016/j.cmpb.2016.02.015.
46. Koh YW, Celik T, Lee HK, Petznick A, Tong L. Detection of meibomian glands and classification of meibography images. *J Biomed Optics*. 2012;17(8):086008. doi:10.1117/1.jbo.17.8.086008.
47. González N, Iloro I, Soria J, Duran JA, Santamaría A, Elortza F, et al. Human tear peptide/protein profiling study of ocular surface diseases by SPE-MALDI-TOF mass spectrometry analyses. *EuPA Open Proteom*. 2014;3:206–15. doi:10.1016/j.euprot.2014.02.016.
48. Grus FH, Augustin AJ. Analysis of tear protein patterns by a neural network as a diagnostical tool for the detection of dry eyes. *Electrophor Int J*. 1999;20(4–5):875–80.
49. Grus F, Augustin A, Evangelou N, Toth-Sagi K. Analysis of tear-protein patterns as a diagnostic tool for the detection of dry eyes. *Eur J Ophthalmol*. 1998;8(2):90–7. doi:10.1177/112067219800800207.
50. Pereira C, Gonçalves L, Ferreira M. Optic disc detection in color fundus images using ant colony optimization. *Med Biol Eng Comput*. 2013;51:295–303. doi:10.1007/s11517-012-0994-5.
51. Abdullah AS, Özok YE, Rahebi J. A novel method for retinal optic disc detection using bat meta-heuristic algorithm. *Med Biol Eng Comput*. 2018;56:2015–24.
52. Abed S, Al-Roomi SA, Al-Shayegi M. Effective optic disc detection method based on swarm intelligence techniques and novel pre-processing steps. *Appl Soft Comput*. 2016;49:146–63. doi:10.1016/j.asoc.2016.08.015.
53. Alshayegi M, Al-Roomi SA, Abed S. Optic disc detection in retinal fundus images using gravitational law-based edge detection. *Med Biol Eng Comput*. 2017;55:935–48. doi:10.1007/s11517-016-1563-0.

54. Nam SM, Peterson TA, Butte AJ, Seo KY, Han HW. Explanatory model of dry eye disease using health and nutrition examinations: machine learning and network-based factor analysis from a national survey. *JMIR Med Inf.* 2020;8(2):e16153.
55. Yoo TK, Choi JY, Kim HK, Ryu IH, Kim JK. Adopting low-shot deep learning for the detection of conjunctival melanoma using ocular surface images. *Comput Methods Programs Biomed.* 2021;205:106086. doi:10.1016/j.cmpb.2021.106086.
56. Rajan S, Ponnan S. An efficient enhanced stacked auto encoder assisted optimized deep neural network for forecasting dry eye disease. *Sci Rep.* 2024;14(1):24945. doi:10.1038/s41598-024-75518-7.
57. Hsiao HK, Liu CC, Yu CY, Kuo SW, Yu SS. A novel optic disc detection scheme on retinal images. *Expert Systems Appl.* 2012;39(12):10600–6. doi:10.1016/j.eswa.2012.02.157.
58. Pruthi J, Arora S, Khanna K. Metaheuristic techniques for detection of optic disc in retinal fundus images. *3D Res.* 2018;9(4):47. doi:10.1007/s13319-018-0198-3.
59. Muthukannan P, Glaret subin P. Optimized convolution neural network based multiple eye disease detection. *Comput Biol Med.* 2022;146:105648. doi:10.1016/j.compbiomed.2022.105648.
60. Ran J, Niu K, He Z, Zhang H, Song H. Cataract detection and grading based on combination of deep convolutional neural network and random forests. In: 2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC). Piscataway, NJ, USA: IEEE; 2018. p. 155–9.
61. Prasad DK, Vibha L, Venugopal K. Machine learning based early detection of age-related macular degeneration: early warning system. In: 2018 Fourteenth International Conference on Information Processing (ICINPRO). Piscataway, NJ, USA: IEEE; 2018. p. 1–6.
62. Sisodia DS, Nair S, Khobragade P. Diabetic retinal fundus images: preprocessing and feature extraction for early detection of diabetic retinopathy. *Biomed Pharmacol J.* 2017;10(2):615–26.
63. Parashar D, Agrawal D. 2-D compact variational mode decomposition-based automatic classification of glaucoma stages from fundus images. *IEEE Trans Instr Meas.* 2021;70:1–10. doi:10.1109/tim.2021.3071223.
64. Qureshi I, Ma J, Abbas Q. Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning. *Multimed Tools Appl.* 2021;80(8):11691–721. doi:10.1007/s11042-020-10238-4.
65. Tayal A, Gupta J, Solanki A, Bisht K, Nayyar A, Masud M. DL-CNN-based approach with image processing techniques for diagnosis of retinal diseases. *Multimed Syst.* 2022;28(4):1417–38. doi:10.1007/s00530-021-00769-7.
66. Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput.* 1997;1(1):67–82. doi:10.1109/4235.585893.
67. Lindtner T, Uzan AY, Eder M, Bar-On B, Elbaum R. Repetitive hygroscopic snapping movements in awns of wild oats. *Acta Biomaterialia.* 2021;135:483–92. doi:10.1016/j.actbio.2021.08.048.
68. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'9—International Conference on Neural Networks*. Vol. 4. Piscataway, NJ, USA: IEEE; 1995. p. 1942–8.
69. Kennedy J. Swarm intelligence. In: *Handbook of nature-inspired and innovative computing: integrating classical models with emerging technologies*. Cham, Switzerland: Springer; 2006. p. 187–219.
70. Van den Bergh F. An analysis of particle swarm optimizers [Ph.D. thesis]. Pretoria, South Africa: University of Pretoria; 2007.
71. Mahdavi S, Rahnamayan S, Deb K. Opposition based learning: a literature review. *Swarm Evol Comput.* 2018;39:1–23. doi:10.1016/j.swevo.2017.09.010.
72. Drucker H, Wu D, Vapnik VN. Support vector machines for spam categorization. *IEEE Trans Neural Netw.* 1999;10(5):1048–54. doi:10.1109/72.788645.
73. Houssein EH, Emam MM, Ali AA, Suganthan PN. Deep and machine learning techniques for medical imaging-based breast cancer: a comprehensive review. *Expert Syst Appl.* 2021;167:114161. doi:10.1016/j.eswa.2020.114161.
74. Houssein EH, Oliva D, Celik E, Emam MM, Ghoniem RM. Boosted sooty tern optimization algorithm for global optimization and feature selection. *Expert Syst Appl.* 2023;213:119015. doi:10.1016/j.eswa.2022.119015.
75. Nagra D. Dry eye disease dataset. Kaggle; 2025 [Dataset]. [cited 2025 Aug 11]. Available from: https://www.kaggle.com/datasets/dakshnagra/dry-eye-disease?select=Dry_Eye_Dataset.csv.

76. Chu X, Ilyas IF, Krishnan S, Wang J. Data cleaning: overview and emerging challenges. In: Proceedings of the 2016 International Conference on Management of Data; 2016 Jun 26–Jul 1; San Francisco, CA, USA. p. 2201–6.
77. Kotsiantis SB, Kanellopoulos D, Pintelas PE. Data preprocessing for supervised learning. *Int J Comput Sci.* 2006;1(2):111–7.
78. Arcuri A, Fraser G. Parameter tuning or default values? An empirical investigation in search-based software engineering. *Empir Softw Eng.* 2013;18(3):594–623. doi:10.1007/s10664-013-9249-9.
79. Mack GA, Skillings JH. A Friedman-type rank test for main effects in a two-factor ANOVA. *J Am Stat Assoc.* 1980;75(372):947–51. doi:10.1080/01621459.1980.10477577.
80. Abd Elaziz ME, Ewees AA, Oliva D, Duan P, Xiong S. A hybrid method of sine cosine algorithm and differential evolution for feature selection. In: International Conference on Neural Information Processing. Cham, Switzerland: Springer; 2017. p. 145–55.
81. Sun Y, Huang Q, Liu T, Cheng Y, Li Y. Multi-strategy enhanced Harris Hawks optimization for global optimization and deep learning-based channel estimation problems. *Mathematics.* 2023;11(2):390. doi:10.3390/math11020390.
82. Rao RV, Patel V. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica.* 2013;20(3):710–20. doi:10.1016/j.scient.2012.12.005.
83. Yu X, Wu L, Xu C, Hu Y, Ma C. A novel neural network for solving nonsmooth nonconvex optimization problems. *IEEE Trans Neural Netw Learn Syst.* 2019;31(5):1475–88. doi:10.1109/tnnls.2019.2920408.
84. Tiwari A, Bansal JC, Sharma H. EEFO: enhanced exploitation firefly optimizer for global numerical problems. *Expert Syst Appl.* 2023;216:120161.
85. Zhang H, Wang Z, Liu Y. Manta ray foraging optimization algorithm: an effective bio-inspired optimizer for global optimization. *Appl Soft Comput.* 2022;114:108190.
86. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S. Equilibrium optimizer: a novel optimization algorithm. *Knowl Based Syst.* 2020;191:105190. doi:10.1016/j.knosys.2019.105190.
87. Yıldız BS, Pholdee N, Panagant N, Bureerat S, Yıldız AR, Sait SM. A novel chaotic Henry gas solubility optimization algorithm for solving real-world engineering problems. *Eng Comput.* 2022;38(Suppl 2):871–83. doi:10.1007/s00366-020-01268-5.
88. Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Adv Eng Softw.* 2014;69:46–61. doi:10.1016/j.advengsoft.2013.12.007.
89. Gharehchopogh FS, Ucan A, Ibrikci T, Arasteh B, Isik G. Slime mould algorithm: a comprehensive survey of its variants and applications. *Arch Comput Methods Eng.* 2023;30(4):2683–723. doi:10.1007/s11831-023-09883-3.
90. Chitra B, Kumar S. An optimized deep learning model using mutation-based atom search optimization algorithm for cervical cancer detection. *Soft Comput.* 2021;25(24):15363–76. doi:10.1007/s00500-021-06138-w.
91. Eid HF. Binary whale optimisation: an effective swarm algorithm for feature selection. *Int J Metaheuristics.* 2018;7(1):67–79. doi:10.1504/ijmheur.2018.10012912.