Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 796–802, DOI 10.12694/scpe.v26i2.3902

APPLICATION RESEARCH OF IMPROVED PARTICLE SWARM COMPUTING INTELLIGENT ALGORITHM IN TRACK AND FIELD TRAINING TARGET OPTIMIZATION

CHAO WANG*

Abstract. This paper studies the influence of the athlete's state and wind direction on the Javelin flight trajectory. The process of plane throwing is analyzed theoretically. An accurate javelin flight mathematical model is established. Using the modified particle swarm optimization method, the paper finds a way to make the Javelin longer distance. This algorithm makes it portable and adaptive. Then, the entropy weight method is used to evaluate the weight of each factor. The experiment shows that this method can reflect the actual Javelin flying condition well, and its prediction effect is in accord with the demand of engineering applications.

Key words: Javelin; Particle swarm optimization; Flat throw motion; Entropy weight method

1. Introduction. Javelin throwing is an ancient sport. In ancient times, it was mainly used for hunting and fighting. Now it's an Olympic sport. In track and field competition, Javelin belongs to the relatively light equipment, but the technology is relatively complex. It can be divided into three stages: javelin holding, run-up and final throw. The most challenging part of this event is the final step after the jump. The quality of its throw will determine the trajectory and final landing of the Javelin. The thrower's motion in the air is determined by the motion form of the thrower and its technical parameters.

Domestic and foreign scholars have done a lot of research on object flight trajectory prediction. Literature [1] simulates surface targets by installing sensors on aircraft. However, this approach is too costly to be popularized. In literature [2], the Elman-NARX neural network is used for pilot track prediction based on Bayesian regularization. However, the application of this method is narrow and not easy to popularize. In reference [3], a method of track deviation compensation, the inverse functional of Lyapunov dynamics, is established. Four kinds of limited variable drag forces are used to simulate the actual situation, and the tracking error is small. The analysis method described in this paper is relatively simple in terms of the influence factors of each factor, so it is more applicable than the above methods.

2. Javelin model construction. The projectile presented in this paper is approximately a square in shape, and its cross-section is always constant. However, the shape of the national standard differs slightly from that described in the title. Firstly, the outline of the Javelin in the national standard is briefly explained (Figure 2.1).

Suppose D_6 is the center of gravity of the Javelin, $D_1 \sim D_9$ is the length of Part I, and the handle position $D_0 \sim D_1$ is Part II. Position $D_4 \sim D_0$ from the handle to the tip of the Javelin is the middle part, marked Part II.

It is necessary to determine the function relation of its external shape to obtain the projectile's area and the projectile center's position. The central axis corresponding to part I is in the range of $[0, 1200]$, while the Javelin gradually thickens as the axis length increases [4]. The long axis of the handle part II is between $[1200, 1800]$ and differs from the image prescribed by the state. The profile radius of this length is constant. In addition, this value includes the length of the long Javelin wrapped with the non-slip rope, which will be considered separately in later calculations. Particle III has a long axis at $[180, 2362)$. It becomes smaller as the radius of the section decreases, and the final tip N at $[2362, 2640]$ is approximately a convex cone. Because the

*Dalian University of Science and Technology, Dalian 116000, China (15040402587@163.com)

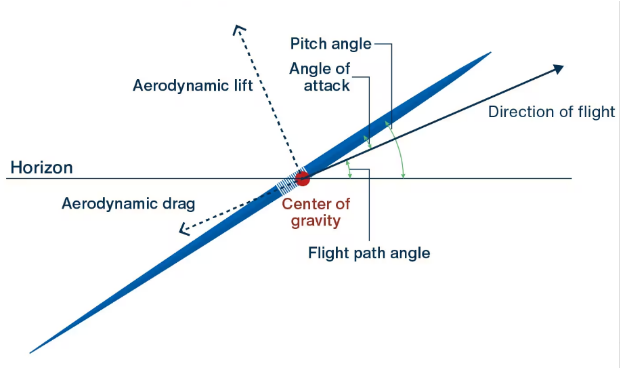

Fig. 2.1: Symbols of each part of the Javelin.

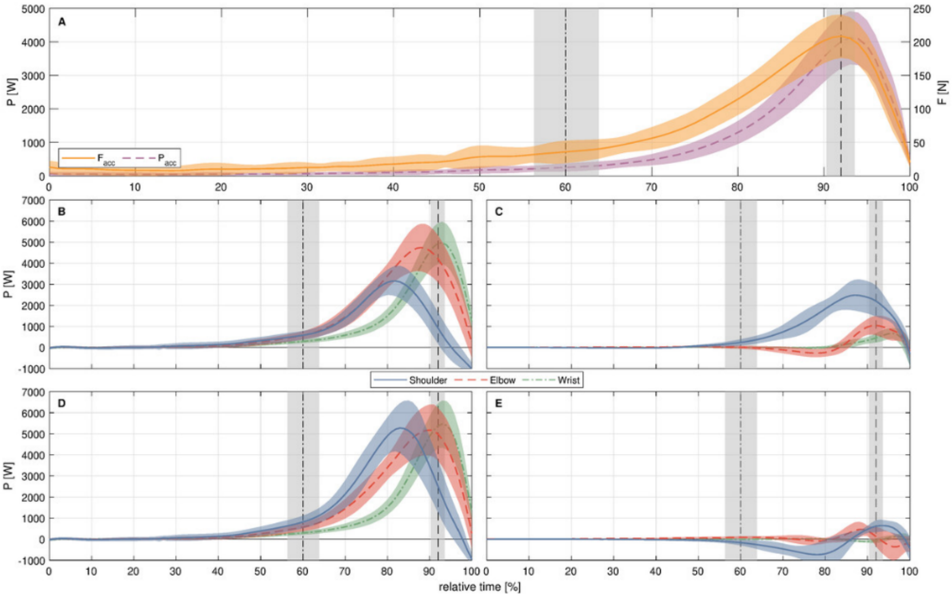

Fig. 2.2: Changes in relative errors of each part of the Javelin after fitting.

corresponding structure of the structure II part is cylindrical, it is not necessary to pay too much attention to it, and the rest of the part can be fitted with nonlinear equations.

Take half the length of each spindle. Use Google's deep network TensorFlow to fit radial parameters. The loss function is determined by the square of the difference between the measured and predicted values [5]. The algorithm uses the stochastic gradient method, sets the learning rate of the network to 0.01 , and conducts 1000 times of repeated training. The final error size obtained in the Tensor Board visual teaching software is shown in Figure 2.2.

When throwing, it is not possible to ensure that the muzzle velocity α of the Javelin is the same angle as the axis of the projectile [6]. If the initial angle of attack β is in the horizontal direction and the angle is more significant than the hand Angle α, the initial angle of attack β is greater than o. If the opposite, the initial angle of attack β is smaller than o. In the case of γ representing the gun Angle, each angle can meet the initial Angle of attack $\beta =$ gun Angle γ - shot angle α.

This paper assumes that the Javelin will not spin during flight. All the power is concentrated in the center

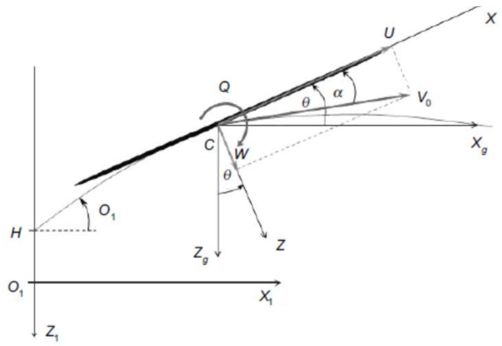

Fig. 2.3: Stress on Javelin during flight.

of the long Javelin. When the Javelin is thrown at the top, it is assumed to be in the horizontal position. At this time, when there is no wind, or the wind direction is horizontal, it is considered that it is not affected by wind resistance [7]. The trajectory of the Javelin has an Angle with the direction of the main shaft of the Javelin. Its dimensions are assumed to match the original conditions for easy calculation. An inverse relationship exists between wind speed and javelin speed under upwind conditions. This value corresponds directly to the squared value of the current rate [8]. In addition, it also relates to the wind coefficient c in the air, the air density p and the force area of the Javelin in the airflow. The forces on the throwing point are shown in Figure 2.3.

In the initial problem, the effect of external wind speed is not considered, so it is only necessary to calculate the drag force caused by the Javelin under the action of the object's relative speed [9]. Its dimensions will also change with the angle β. The β value is the angle of the Javelin's spindle and the rate of fire. During the rising period, its angle continues to decline. The size of β during the decline period can be regarded as constant after it has increased to a certain point.

The basic principles of Newtonian mechanics and momentum are analyzed [9]. The formula to be achieved in the ascending phase is as follows:

$$\begin{cases} F_x = f_x \\ F_y = f_y + G \\ \frac{1}{2} \cdot m \cdot v_1^2 - \frac{1}{2} \cdot m \cdot v_0^2 = -\int_0^h F_x ds - \int_2^h F_y ds \\ f_x = f \cdot \cos\theta \\ f_y = f \cdot \sin\theta \\ f = \frac{1}{2} \cdot \varepsilon \cdot \rho \cdot L \cdot v^2 \\ L = L_1 \cdot \sin\theta \\ \int_2^h F_y ds = \frac{1}{2} \cdot m \cdot v_y^2 \\ v_y = v_0 \cdot \sin\alpha \\ G = mg \end{cases}$$

To go from the highest point to the lowest point during the throw, the formula needed is:

$$\begin{cases} F_x = f_x \\ F_y = G - f_y \\ \frac{1}{2} \cdot m \cdot v_2^2 - \frac{1}{2} \cdot m \cdot v_1^2 = -\int_0^h F_x ds - \int_0^h F_y ds \\ f_x = f \cdot \cos\alpha \\ f_y = f \cdot \sin\alpha \\ f = \frac{1}{2} \cdot \varepsilon \cdot \rho \cdot L \cdot v^2 \\ L = L_1 \cdot \sin\theta \\ \int_0^h F_y ds = \frac{1}{2} \cdot m \cdot v_y^2 \\ v_y = v_2 \cdot \sin\alpha \\ G = mg \end{cases}$$

The action function of the thrower to be constructed in this problem looks like this:

$$\begin{cases} \int_0^{l_1} \left(v_x^2+v_y^2\right) \cdot \sin 2\theta d v_x \cdot t=\frac{2 \cdot m \cdot v_0^2 \cdot \cos ^2 \alpha-2 \cdot m \cdot v_1^2}{\varepsilon \cdot \rho \cdot L_1} \\ \int_0^{l_2} \left(v_x^2+v_y^2\right) \cdot \sin \theta \cdot \cos \alpha d v_x \cdot t=\frac{m \cdot v_1^2-m \cdot v_2^2-m \cdot v_y^2}{\varepsilon \cdot \rho \cdot L_1} \\ l=l_1+l_2 \end{cases}$$

In this way, the mathematical models of the following several javelins can be expressed by the above formula.

3. Particle swarm algorithm. Particle swarm is a population-based optimal search method in which each particle in the population represents a possible solution to the problem, and the fit value of the solution is associated with the objective function. During the search, the particle will adjust its speed according to its flight history and the optimal particle position [10]. They can do this by cooperating and competing to find the best particles. The modified equation of motion and positioning mode of the particle swarm optimization method is as follows:

$$v_{id}(t+1)=\lambda v_{id}(t)+\delta_1\left(\eta_{id}(t)-u_{id}(t)\right)+\delta_2\left(f_{id}(t)-u_{id}(t)\right)$$
$$u_{id}(t+1)=u_{id}(t)+v_{id}(t+1)$$

where $\delta_1=z_1$ Rand $1(), \delta_2=z_2$ Rand $2(\ ()$. The constants z_1 and z_2 represent the particle's effect on social and personal cognitive levels, and are generally expressed in the same numerical value. Rand1() And Rand 2() is any number in any range $(0,1)$. λ is the inertial weight. The concept of the grey correlation degree is used to study the correlation of each

$$U_0=\left(u_0(1), u_0(2), \ldots, u_0(n)\right),$$

$$U_1=\left(u_1(1), u_1(2), \ldots, u_1(n)\right), \ldots, .$$
$$U_m=\left(u_m(1), u_m(2), \ldots, u_m(n)\right)$$

The correlation factor $\zeta\left(u_0(r), u_i(r)\right)$ between U_0 and $U_1, U_2, \ldots, U_m$ is:

$$\zeta\left(u_0(r), u_i(r)\right)=\frac{\min_i \min_r\left|u_0(r)-u_i\left(r^r\right)\right|+\xi \max_i \max_r\left|u_0(r)-u_i\left(r^r\right)\right|}{\left|u_0(r)-u_i(r)\right|+\xi \max_i \max_r\left|u_0(r)-u_i(r)\right|}$$
$$r=1,2, \ldots, n$$

Here the degree of grey correlation between $\xi \in(0,1), U_i$ and U_0 is $\zeta\left(U_0, U_i\right)=\frac{1}{n} \sum_{r=1}^n \zeta\left(u_0(r), u_i(r)\right)$, where ξ is the differentiation coefficient. Regularization is usually used to improve the learning effect of neural networks.

Regularization is usually used to improve the learning effect of neural networks. The error function of the standard method is expressed by $W=\omega \cdot W_C+\varphi \cdot W_\Lambda$. $W_C=\frac{1}{2} \sum_{r=1}^K \sum_{j=1}^N\left(c_{jr}-y_{jr}\right)^2, N$ is the sample quantity. K is the number of output nodes of the neural network. c_{jr} is the quantity required. y_{jr} indicates the actual network output. $W_\Lambda=\frac{1}{2} \sum_{i=1}^\Lambda \lambda_i^2, W_\Lambda$ is for formal fines. Where λ_i is the weight and threshold for training the network of neurons. Where Λ is the number of connected weights and thresholds. φ, ω is the super parameter. The size of φ, ω directly affects the learning purpose of the network.

4. Model solving. The throw Angle is the angle of the throw point to the ground. The starting angle of attack is the angle between the motion trajectory of the center of the Javelin and the axis of the Javelin during the throwing process [11]. The Javelin has an Angle of attack of 0 during flight. The initial angle of attack is negative when above the vertical axis of the Javelin. The angle of attack is positive when the Javelin's trajectory is down. The same type of Javelin was tested in this study [12]. It is necessary to find the relationship between throwing distance and shooting Angle, initial attack Angle and shooting speed according to the existing experimental data, and then find the trajectory of the Javelin in the air.

First, three lines were constructed with the player's shot Angle, shot speed and initial attack Angle as three independent variables [13]. The internal variables of each line are 24 lines, which are the measured values of

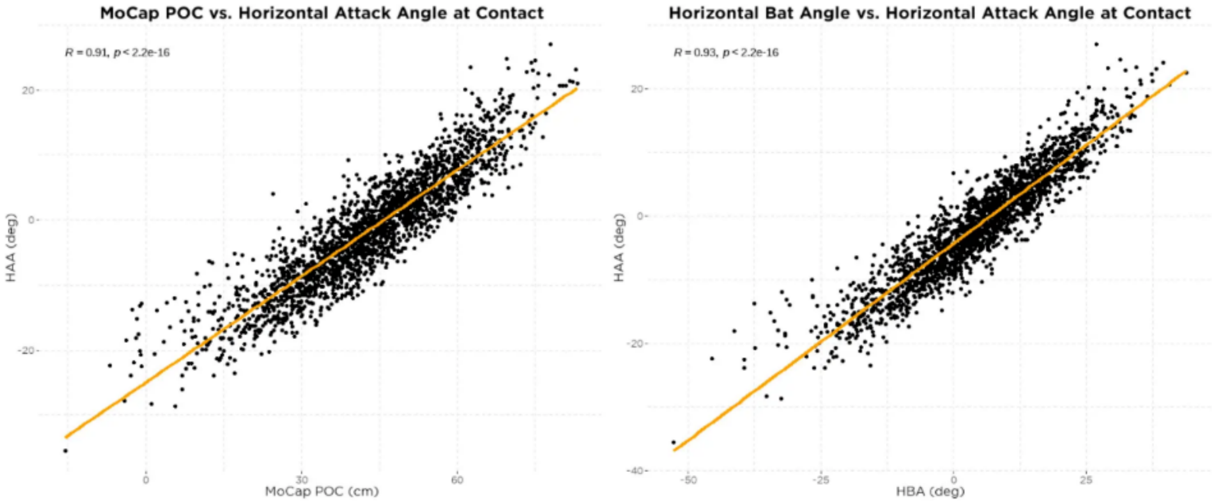

Fig. 4.1: Scatter diagram of the influence of shot Angle on throwing distance.

each independent variable. A line vector containing 24 measured values is created. SPSS statistical analysis tool was used to test the three independent vectors constructed using the throw's distance as the dependent variable. The three independent variables were analyzed statistically and tested by MATLAB. The R square values obtained by the two methods are compared with the regression coefficients obtained [14]. Through the statistics of the experimental results, it is found that factors such as the angle, the speed of the motion and the starting angle of the thrower are related to the motion of the Javelin.

From Figure 4.1, most of the points show a tendency to increase as the angle of the shot increases, but not as significantly as the speed of the shot increases. This is also consistent with the regression coefficient before the release speed is used as the independent variable, which is as high as 0.476. Although still a positive regression coefficient, its value is much smaller than the regression coefficient when the release speed is used as the independent variable. However, as people can see from the graph, there is also a positive correlation between the distance thrown and the angle of the shot [15].

MATLAB software is used to verify the multi-variable regression model. After some error points are eliminated, each data contains zero points, which verifies the accuracy and practicability of the method. When other initial angles of attack remain unchanged, the change of initial angle of attack does not change significantly [16]. The optimal time in the combination of various throwing angles and initial angles of attack is A0=42°, B0=-5°.

The particle swarm optimization method combines the individual optimal results with the population optimization results to solve the local optimization [17]. Positioning constraint [28,44], set the initial population N as 50, repeat 100 times, and randomly generate a 50x1 data matrix from 28-44. Compare the current fitness and the best solution set with the historical record, and update its optimal positioning and solution if it exceeds it.

It can be seen from Figure 4.2 that this method has successfully found the best scheme. The paper gets the best result: A0=42°, B0=-5°. The optimum initial attack Angle of the Javelin varies with the throwing Angle, so the optimum initial attack Angle is not a constant value. There is some correspondence between the two. When the projection Angle is large, the corresponding optimal initial impact Angle is smaller [18]. As the projection Angle decreases, the corresponding optimal initial attack Angle also increases, and in some cases, these two parameters are optimal.

The simulation results show that when other initial conditions are unchanged, whether downwind or upwind, the flight range of the Javelin will gradually become longer with the increase of the throwing Angle in the initial state. It shrinks again when the throwing Angle increases to 42°.

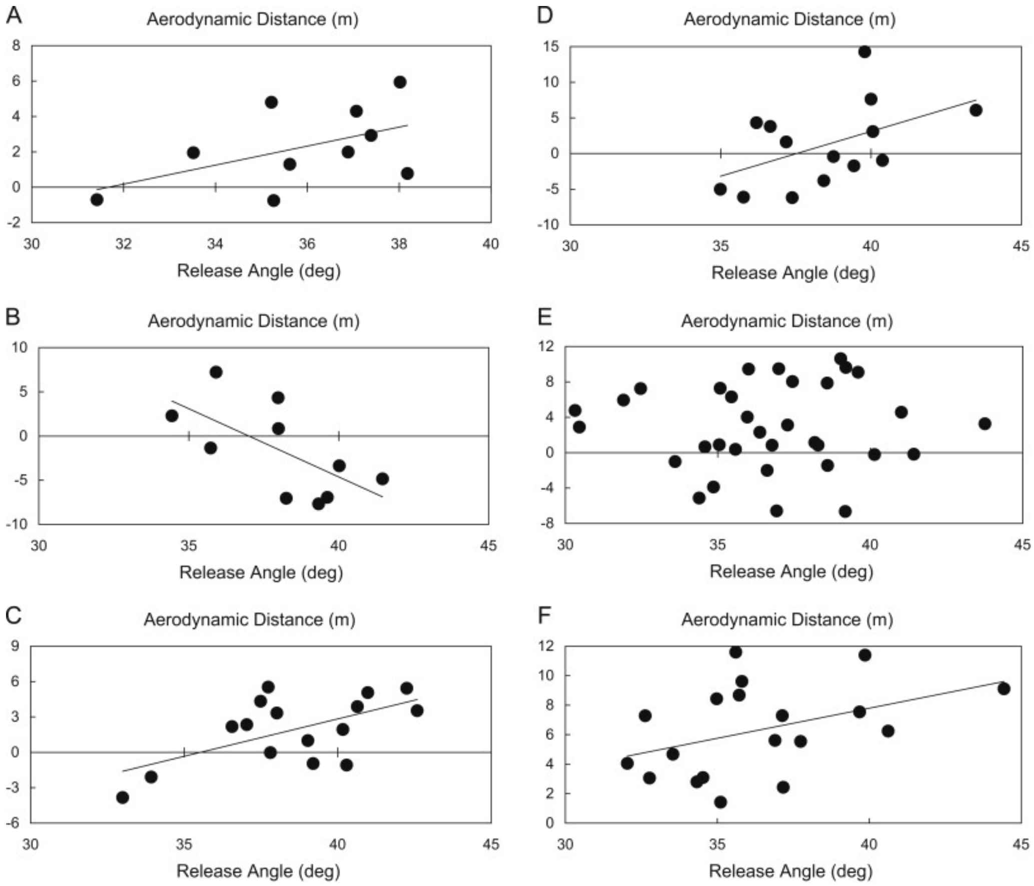

Fig. 4.2: Variation of throwing distance with shot Angle.

5. Conclusion.

1. When other initial values are the same, the particle swarm optimization method is used to conduct the research. The results show that when the throwing Angle is 0°-42°, the throwing amount increases with the increasing angle.
2. Under the premise that other initial conditions are the same and the throwing Angle is fixed, the joint action of the initial attack Angle and the initial inclination angle greatly influence the projectile's range. The optimal throwing quantity of this combination is A0=42°and B0=-5°.

REFERENCES

[1] Pawan, Y. N., Prakash, K. B., Chowdhury, S., & Hu, Y. C. (2022). Particle swarm optimization performance improvement using deep learning techniques. Multimedia Tools and Applications, 81(19), 27949-27968.

[2] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering, 29(5), 2531-2561.

[3] Cansu, T., Kolemen, E., Karahasan, Ö., Bas, E., & Egrioglu, E. (2023). A new training algorithm for long short-term memory artificial neural network based on particle swarm optimization. Granular Computing, 8(6), 1645-1658.

[4] Liu, X. H., Zhang, D., Zhang, J., Zhang, T., & Zhu, H. (2021). A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. Cluster Computing, 24(3), 1901-1915.

[5] Berbek, M. I., & Oglah, A. A. (2022). Adaptive neuro-fuzzy controller trained by genetic-particle swarm for active queue management in internet congestion. Indonesian Journal of Electrical Engineering and Computer Science, 26(1), 229-242.

[6] Usmani, U. A., Watada, J., Jaafar, J., Aziz, I. A., & Roy, A. (2021). Particle swarm optimization with deep learning for human action recognition. Int. J. Innovative Comput. Inform. Control, 17(6), 1843-1870.

[7] Yilmaz, A., & Yolcu, U. (2022). Dendritic neuron model neural network trained by modified particle swarm optimization for time-series forecasting. Journal of Forecasting, 41(4), 793-809.
[8] Elsedimy, E. I., AboHashish, S. M., & Algarni, F. (2024). New cardiovascular disease prediction approach using support vector machine and quantum-behaved particle swarm optimization. Multimedia Tools and Applications, 83(8), 23901-23928.
[9] Yousef, B. A., Rezk, H., Abdelkareem, M. A., Olabi, A. G., & Nassef, A. M. (2020). Fuzzy modeling and particle swarm optimization for determining the optimal operating parameters to enhance the bio-methanol production from sugar cane bagasse. International Journal of Energy Research, 44(11), 8964-8973.
[10] Ben Seghier, M. E. A., Carvalho, H., Keshtegar, B., Correia, J. A., & Berto, F. (2020). Novel hybridized adaptive neuro-fuzzy inference system models based particle swarm optimization and genetic algorithms for accurate prediction of stress intensity factor. Fatigue & Fracture of Engineering Materials & Structures, 43(11), 2653-2667.
[11] Golafshani, E. M., Behnood, A., & Arashpour, M. (2023). Predicting the compressive strength of eco-friendly and normal concretes using hybridized fuzzy inference system and particle swarm optimization algorithm. Artificial Intelligence Review, 56(8), 7965-7984.
[12] Kumar, G., Singh, U. P., & Jain, S. (2022). An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting. Soft Computing, 26(22), 12115-12135.
[13] Diaz Martinez, D., Trujillo Codorniu, R., Giral, R., & Vazquez Seisdedos, L. (2021). Evaluation of particle swarm optimization techniques applied to maximum power point tracking in photovoltaic systems. International Journal of Circuit Theory and Applications, 49(7), 1849-1867.
[14] Berlin, S. J., & John, M. (2020). Particle swarm optimization with deep learning for human action recognition. Multimedia Tools and Applications, 79(25), 17349-17371.
[15] Sahour, S., Khanbeyki, M., Gholami, V., Sahour, H., Karimi, H., & Mohammadi, M. (2024). Particle swarm and grey wolf optimization: enhancing groundwater quality models through artificial neural networks. Stochastic Environmental Research and Risk Assessment, 38(3), 993-1007.
[16] Nayak, J., Swapnarekha, H., Naik, B., Dhiman, G., & Vimal, S. (2023). 25 years of particle swarm optimization: Flourishing voyage of two decades. Archives of Computational Methods in Engineering, 30(3), 1663-1725.
[17] Balasubramanian, K., Ananthamoorthy, N. P., & Ramya, K. (2022). An approach to classify white blood cells using convolutional neural network optimized by particle swarm optimization algorithm. Neural Computing and Applications, 34(18), 16089-16101.
[18] Saha, S., Saha, A., Roy, B., Sarkar, R., Bhardwaj, D., & Kundu, B. (2022). Integrating the Particle Swarm Optimization (PSO) with machine learning methods for improving the accuracy of the landslide susceptibility model. Earth Science Informatics, 15(4), 2637-2662.

Edited by: Hailong Li
Special issue on: Deep Learning in Healthcare
Received: May 11, 2024
Accepted: Jun 20, 2024