Gait Pattern Recognition through Force Sensor Platform based on XGBoost Model and Harris' Hawks Optimization

Lie Yu, Pengzhi Mei, and Lei Ding

***Abstract*—This study developed a gait pattern classification system based on ground contact forces measured by six force sensors embedded inside the shoe sole. The data transmission is facilitated via the Bluetooth module integrated into an STM32 microcontroller. The extreme gradient boosting (XGBoost) algorithm is used to identify the gait patterns, and the basic idea of XGBoost is to use second-order derivatives to make the loss function more precise, incorporate regularization to prevent tree overfitting, and enable block storage for parallel computation. By optimizing the XGBoost algorithm with four algorithms, the exploration capabilities of these algorithms are effectively incorporated into the fusion model. Experimental results indicate that the XGBoost algrithm optimized by Harris' hawks optimization (HHO) outperforms the other optimization algorithms. Specifically, the HHO-XGBoost achieved high values of 97.41%, 97.03%, and 97.22% severally in the metrics of precision, recall, and F1 score. This research illustrates the HHO-XGBoost method's superiority in gait phase recognition.**

***Index Terms*—Gait pattern classification, Ground contact force, XGBoost, Harris' Hawks Optimization.**

I. Introduction

Gait phase recognition is a widely adopted technique for assessing the movement state during human walking. It has been extensively utilized in diverse fields such as medicine [1-2], forensic detection [3-4], smart furniture control [5-6], and robot exoskeleton control [7]. Extracting the gait data would enable the researchers to conduct more comprehensive assessments of human movement and devise more effective response strategies. For individuals without physical impairments, during the same movement pattern, each leg exhibits similar motion, with only a phase lag between different limbs [8]. However, analyzing the gait data becomes crucial for controlling the exoskeleton for people who use the robotic exoskeletons or those with limb impairments. There are various methods for determining the movement state. For example, researchers can utilize computer vision to assess movement or employ a frame-level part feature extractor in combination with a micro-motion capture module to extract motion features [9-10]. A two-dimensional center of pressure can be used to extract the motion features and decompose the planar trajectories for accurate localization [11]. However, these approaches involve complex force processing or require handling intricate signals. Consequently, wearable sensors offer a cost-effective and easily installable solution for gait phase recognition [12-13]. This study introduces an intelligent shoe sole equipped with six force-sensitive resistors (FSRs) to collect data. By processing and labeling the data from the wearable sensors, the movement state was effectively determined.

Using insole pressure sensors to collect the ground reaction force (GRF) gait data features has become increasingly common. Before classifying data by machine learning algorithms, selecting appropriate features is essential. In gait data analysis, time domain features such as mean, variance, and kurtosis are often considered [14]. These are further transformed into frequency domain features through the Fast Fourier Transform [15]. Collected data undergoes weighting and normalization processes before classification by machine learning models.

With advances in artificial intelligence and machine learning, numerous classification algorithms have emerged [16-17]. As support vector machines (SVM) and gradient boosting are widely used for training and learning from labeled data [18-19].

In recent years, the XGBoost algorithm has been widely used as a robust classifier, and widely applied in data science competitions and industrial settings due to its exceptional performance [20-22]. Its optimization is largely depends on the hyperparameters. When these parameters are left at their default settings, optimal performance is often not achieved. Selecting a suitable optimization algorithm to identify the optimal hyperparameters is thus critical for enhancing the XGBoost's classification accuracy. When integrated with XGBoost, optimization techniques such as genetic algorithms (GA) [23], arithmetic optimization algorithm (AOA) [24], coati optimization algorithm (COA) [25], and Harris hawks optimization (HHO) [26-29] can effectively identify the optimal hyperparameters.

This study developed an intelligent insole system equipped with six wearable pressure sensors to collect the GRF data. The XGBoost model was optimized using the HHO to enhance the accuracy of motion state classification. By normalizing and weighting the pressure data, the movement process of each foot was categorized into four distinct states. The experimental results show that the HHO-XGBoost classifier achieved the best performance on this dataset, with an overall accuracy of 97.85%.

Manuscript received May 15, 2024; revised November 22, 2024.

This work was supported by the National Natural Science Foundation of China "Research on motion pattern recognition of exoskeleton robot based on curve similarity model" (NO.62106178).

Lie Yu is an Associate Professor at the School of Electronic and Electrical Engineering, Wuhan Textile University, Wuhan, China. (corresponding author to provide phone: +86 18607155647; e-mail: lyu@wtu.edu.cn)

Pengzhi Mei is a postgraduate student at the School of Electronic and Electrical Engineering, Wuhan Textile University, Wuhan, China. (e-mail: 1006242095@qq.com)

Lei Ding is an Associate Professor at the School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan, China. (e-mail: lding@wtu.edu.cn)

II. DATA ACQUISITION AND PROCESSING

A. *Instrumental setup*

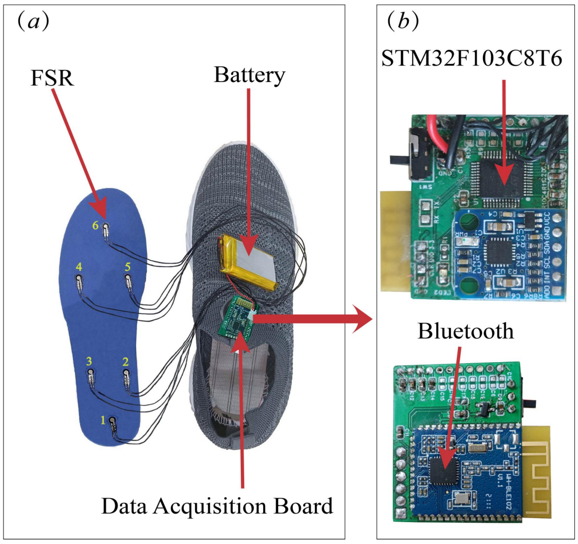

Fig. 1. (a) The intelligent shoe integrated with six FSRs. (b) The STM32 collects the force information and the Bluetooth module transmits the information into the host.

This study developed a smart intelligent shoe integrated with six FSRs to monitor the GRFs. As depicted in Figure 1, the FSRs with strong adhesion, bend resistance, and high sensitivity are placed at different points on the shoe sole. The force data from the FSRs are collected via an STM32 microcontroller, which is powered by a rechargeable 3.3 V battery. An amplifying circuit is used to increase the output voltage, which ranges from 0 to 3.3 V. Prior to the experiments, each FSR was calibrated by applying various standard weights to record the corresponding output voltages. The recorded output voltages are then converted into the corresponding force measurements.

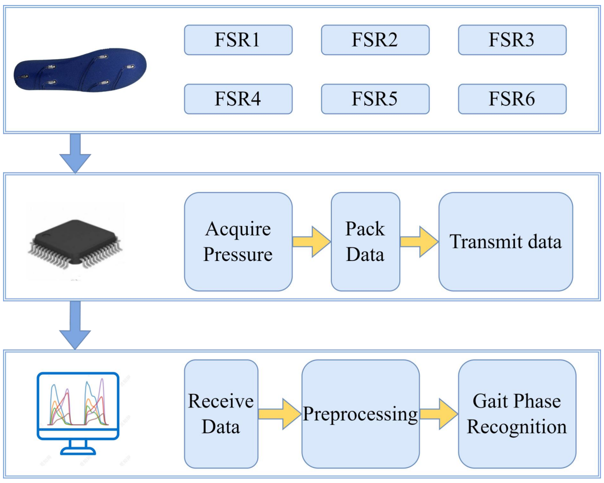

Fig. 2. The working principle and the experimental procedure

The primary operating principle of this system is illustrated in Figure 2. The 12-bit analog-to-digital converter (ADC) integrated into the STM32 is utilized to digitize the output voltage with a sampling frequency of 50 Hz. The acquired pressure signals are subsequently transmitted to a Bluetooth module via a serial port and then sent to the host computer. The computer collects data and stores it for further use. This series of operations is performed offline.

B. *Experimental protocol*

Ten participants (six males and four females, with a mean height of 160 ± 5.9 cm and a mean age of 22 ± 3.3 years) who were free from foot-related conditions volunteered for this study. The participants were instructed to don the smart shoe and walk continuously at a constant speed of 3 km/h on a treadmill for a duration of 2 minutes. Before data collection, participants were instructed to adjust the insole position. Ensuring correct placement helps maintain optimal sensor contact, thereby minimizing testing errors caused by insole movement.

C. *Data analysis*

During the experiment, several sets of plantar pressure data were collected. The collected voltage signals were digitized. Then the data were normalized and integrated to generate a schematic of plantar pressure during movement. Figure 3(a) displays the force values of the six FSRs, and Figure 3(b) shows the labeled diagram of the processed pressure values. It is evident that when the heel contacts the ground, the values of the FSRs in the rear foot (FSR1, FSR2, and FSR3) increase rapidly and quickly reach their peak. When the FSRs under the forefoot (FSR1, FSR2, and FSR3) are loaded, their corresponding force values increase steadily. Until the foot leaves the ground, all sensor forces drop to zero. This repetitive force pattern in the foot provides a reliable basis for state identification.

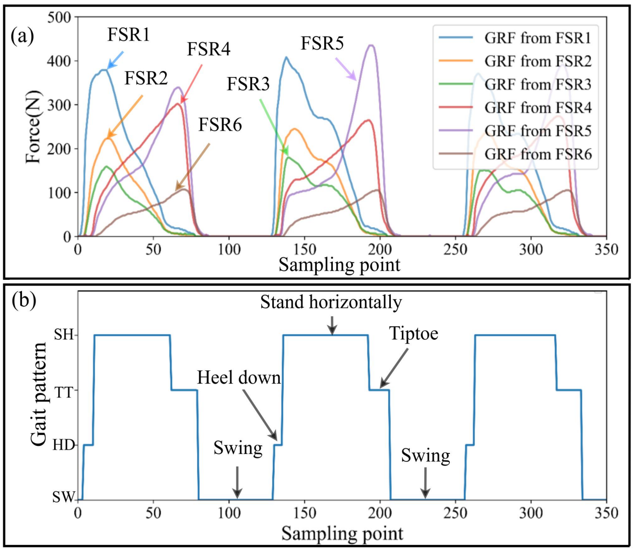

Fig. 3. Gait data are divided into specific gait patterns: (a) Data collected from six FSRs. (b) The labeling of gait patterns corresponds to the data.

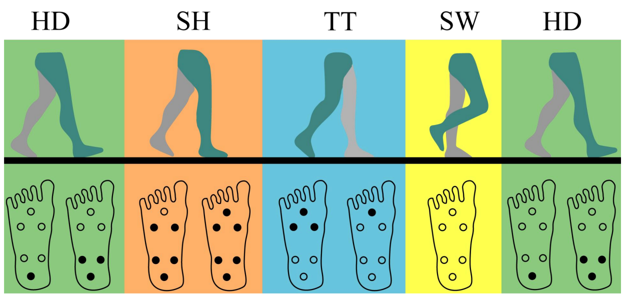

Fig. 4. Gait pattern division corresponding to different FSRs judged as on-ground or off-ground statuses.

D. Reference gait labels

In a typical walking cycle, foot movement is continuous and cyclic. It can be divided into four distinct phases: heel down (HD), standing horizontal (SH), toe tip (TT), and swing (SW). Figure 3 illustrates the labeling of gait patterns corresponding to pressure data during three cycles. Figure 4 shows the division of gait phases based on the data from various FSRs, which are categorized as either on-ground or off-ground statuses. In these figures, the white dot represents the off-ground status, while the black dot denotes the on-ground status. The differentiation between on-ground and off-ground statuses is determined by setting a threshold for the GRFs.

When the heel touches the ground, it is defined as the HD state, in which only FSR1, FSR2, and FSR3 register force. When all six FSRs detect force, the label value reaches its maximum. We define this as the SH state, which is also the longest-lasting phase in the movement cycle. Only the three forefoot FSRs (FSR4, FSR5, and FSR6) register force when standing on the toes. We define it as the Toe Touch (TT) state. When stepping forward with the foot in the air, all FSR values drop to zero. This state is defined as the SW.

E. Data preprocessing

Due to the significant variations in gait patterns corresponding to the GRFs collected by sensors, data preprocessing is typically required. Normalization is an essential step that helps to mitigate dimensional disparities between different gait patterns. It can accelerate model convergence and enhance both accuracy and stability. By standardizing the data, biases between gait patterns can be reduced. Standardizing the data can also make it more comparable and interpretable. The normalization equation is presented below.

$$x_{norm} = \frac{x-\mu}{\sigma} \tag{1}$$

where μ represents the average value, and σ represents the standard deviation. Data normalization and standardization are common preprocessing techniques that improve the performance and stability of machine learning models. These methods ensure consistency in data normalization and reduce the impact of feature variation. It also makes the model process data more efficiently and accurately.

III. METHOD

To improve the accuracy of gait phase classification, this study employed the XGBoost algorithm for gait pattern identification. The XGBoost algorithm is an optimized distributed gradient boosting library. It is highly efficient and flexible, and it has highly strong adaptability. It offers significant advantages in terms of both time and performance when efficiently processing large datasets.

The collected data were normalized and labeled according to the rules outlined in Section II. In this study, 70% of the data were allocated for training, while the remaining 30% were reserved for testing. The training data were used to train the XGBoost model, and the testing data were input into the trained model to predict the gait patterns. The accuracy of the model was determined by the predicted gait patterns and the actual labeled data. To further enhance accuracy, we use the HHO algorithm to optimize the model parameters. The complete process of using the HHO algorithm to optimize the XGBoost for gait pattern classification is depicted in Figure 5.

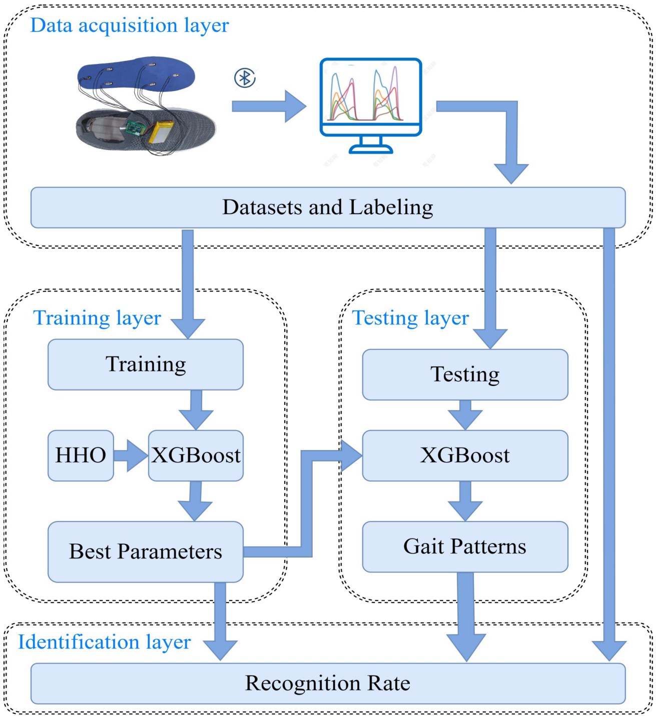

Fig. 5. Flow diagram of HHO optimizing XGBoost

A. Basic Principle of XGBoost

The XGBoost is an algorithm designed for practical implementation, which is based on the gradient-boosting decision tree framework. One of the main advantages of XGBoost is its support for linear classifiers. The XGBoost algorithm provides effective solutions by introducing different regularization techniques into the loss function. By the way, the prediction accuracy of the model is greatly improved. The predictive model of XGBoost can be formulated as follows:

$$\hat{y}_i^m = \sum_{k=1}^{m} f_k(x_i) \tag{2}$$

where m represents the number of decision trees, $\hat{y}$ is the prediction result, and f_k denotes the k-th decision tree. The objective function of XGBoost is divided into two parts, which are the loss function and regularization term. The objective function can be expressed as:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{m-1} + f_m(x_i)) + \Omega(f_m) + C \tag{3}$$

where l is the loss function, and Ω is the regularization term. The regularization term Ω can also be rewritten in the following.

$$\Omega(f_m) = \gamma J + \frac{1}{2}\lambda\sum_{J=1}^{J} w_j^2 \tag{4}$$

where J represents the number of leaf nodes, and ω is the scores for different leaf nodes. Moreover, both γ and λ are customization parameters. By combining Equations (3) and (4), the loss function can be expressed as:

$$Obj = \sum_{i=1}^{n}[l(y_i, \hat{y}_i^{m-1}) + g_i f_m(x_i) + \frac{1}{2}h_i f_m^2(x_i)] + \Omega(f_m) + C \tag{5}$$

where g_i represents the first derivative of the Taylor expansion, and h_i denotes the second derivative of the Taylor expansion. They are defined as follows:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{m-1})}{\partial \hat{y}_i^{m-1}} \tag{6}$$

$$h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{m-1})}{\partial (\hat{y}_i^{m-1})^2} \tag{7}$$

By substituting Equations (4), (6), and (7) into Equation (5), the solutions can be derived as follows:

$$w_j^* = -\frac{\sum g_i}{\sum h_i + \lambda} \tag{8}$$

$$Obj^* = -\frac{1}{2}\sum_{j=1}^{J}\frac{(\sum g_i)^2}{\sum h_i + \lambda} + \gamma J \tag{9}$$

where w^* represents the optimal solution for the weights, and Obj^* is the score of the loss function.

B. Harris' hawks optimization

HHO is a heuristic method inspired by the cooperative hunting behavior of Harris hawks. It seeks to emulate the hawk's strategies in searching for globally optimal solutions. Figure 6 illustrates the flow of the HHO algorithm. The HHO algorithm divides the hunting process into two primary phases(exploration and exploitation), which imitate the natural hunting behaviors of Harris' hawks.

In the exploration phase, two strategies are employed for an equiprobable global search for prey. When P<0.5, each hawk moves based on the positions of other members and the prey. And while P>=0.5, the Harris hawks will randomly perch on a tree within the population range. The expression for this is as follows:

$$x_i(t+1) = \begin{cases} X_{rand}(t) - R_1\left|X_{rand} - 2R_2 x_i(t)\right|, q \geq 0.5 \\ X^*(t) - X_m(t) - R_3(L_b + R_4(U_b - L_b)), q < 0.5 \end{cases} \tag{10}$$

where x_i(t) denotes the current position vector of the hawk at iteration t, and x_i(t+1) represents the next position vector. X_{rand} is a randomly selected individual from the hawk population within the current generation. X_m signifies the average position of the current hawk population. Ub and Lb are the upper and lower bounds of the search range. R_1, R_2, R_3, and R_4 are random numbers within the interval (0, 1).

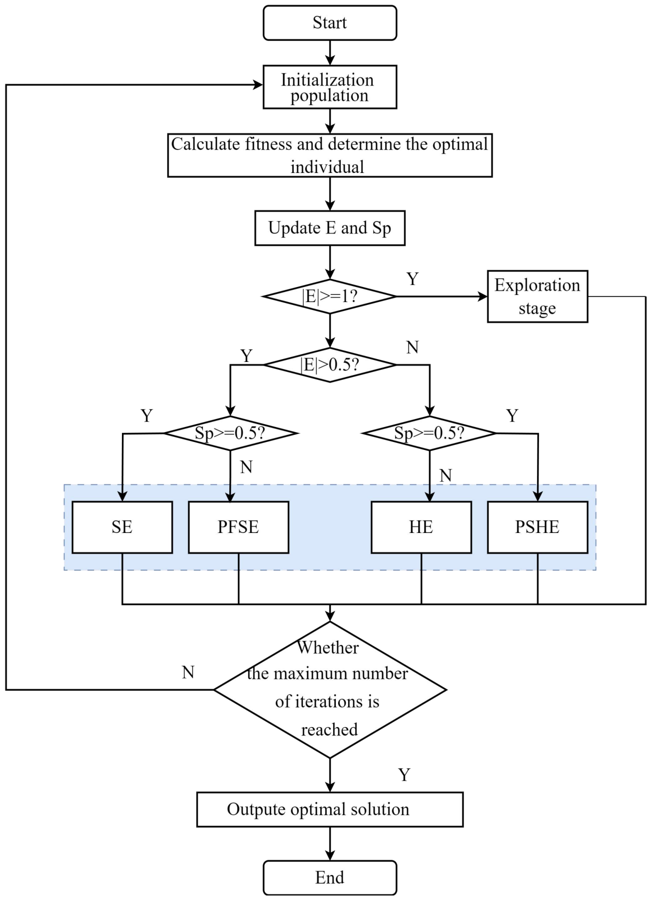

Fig. 6. Flowchart of the HHO Algorithm

As the prey attempts to escape, its energy gradually decreases. The energy E of the escaping prey is defined as follows:

$$E = 2E_0(1 - \frac{t}{M}) \tag{11}$$

where E_0 represents the initial escape energy of the prey, which is a random number between (-1, 1). M is the maximum evolutionary generation of the population, and t denotes the current evolutionary generation. If the absolute value of E is greater than 1, the exploration phase is engaged. Otherwise, the exploitation phase is initiated.

In the exploitation phase, the HHO algorithm employed four strategies: soft besiege, hard besiege, progressive rapid descent with soft besiege, and progressive rapid descent with hard besiege. We define S_p as the prey escape probability, which is a random number between 0 and 1. The S_p must be judged before they can be implemented. If S_p<0.5, there is a chance of escape.

Levy flight pattern was used during the exploration. It is a random walk pattern characterized by a series of short steps interspersed with occasional long jumps. This model is often used to describe the search behavior of animals or humans.

The Levy flight pattern balances local searching with broader exploration. It is more efficient than regular random walks in environments where the target's location is unknown. Its definition is as follows:

$$\begin{cases} Levy = 0.01\dfrac{\mu\sigma}{|v|^{\frac{1}{\beta}}} \\ \sigma = \left(\dfrac{\Gamma(1+\beta)\sin(\frac{\beta\pi}{2})}{\Gamma(1+\beta)\beta 2^{(\frac{\beta-1}{2})}} \right)^{\frac{1}{\beta}} \end{cases} \tag{12}$$

where u and v are random values ranging from 0 to 1, and β is set to 1.5.

The four strategies are specified as follows, with each one being determined by the specific values of two key parameters: E and S_p.

The first strategy is characterized by $0.5 \le |E| < 1$ and $S_p \ge 0.5$. In this scenario, the prey possesses sufficient energy and attempts to escape. It is encircled by the hawks to deplete its energy, followed by a surprise pounce. This behavior can be implemented through the following rules:

$$\begin{cases} x_i(t+1) = \Delta x_i(t) - E\left|JX^*(t) - x_i(t)\right| \\ \Delta x_i(t) = X^*(t) - x_i(t) \end{cases} \tag{13}$$

where J is the jump intensity of the prey, which possesses a random value in each iteration.

The second strategy applies when $|E| < 0.5$ and $S_p \ge 0.5$. In this case, the prey has low escape energy. The hawks are less likely to surround the prey for a surprise pounce. The solutions in this phase are defined as follows:

$$x_i(t+1) = X^*(t) - E\left|\Delta x_i(t)\right| \tag{14}$$

The third strategy is characterized by 0.5 ≤|E |< 1 and Sp < 0.5. In this scenario, the prey has sufficient energy to evade the hawks through rapid dives. Therefore, the position of the current solution can be updated as follows.

$$x_i(t+1) = \begin{cases} Y = X^*(t) - E\left|JX^*(t) - x_i(t)\right|, F(Y) < F(x_i(t)) \\ Z = Y + S \cdot Levy, F(Z) < F(x_i(t)) \end{cases} \tag{15}$$

where F is the fitness function, and S is a random vector ranging in (0, 1).

The fourth strategy can be described by the condition $|E| < 0.5$ and $S_p < 0.5$. In this phase, the prey has low energy, so the solutions are updated accordingly.

$$x_i(t+1) = \begin{cases} Y' = X^*(t) - E\left|JX^*(t) - x_m(t)\right|, F(Y') < F(x_i(t)) \\ Z' = Y' + S \cdot Levy, F(Z') < F(x_i(t)) \end{cases} \tag{16}$$

By evaluating the magnitude of different parameters, the strategy for HHO optimizing the XGBoost is determined.

C. HHO-XGBoost

This study utilizes the HHO algorithm to optimize the performance of XGBoost by tuning four key parameters: learning rate (LR), gamma (GM), subsample (SP), and maximum depth (MD). In the optimization process, these parameters are put into HHO for tuning. In each iteration, the tuned parameters are fed into XGBoost to compute the best fitness and determine the current optimal position. The data undergoes iterative processes for optimization and classification. The optimal fitness values and corresponding positions are recorded throughout the training phase.

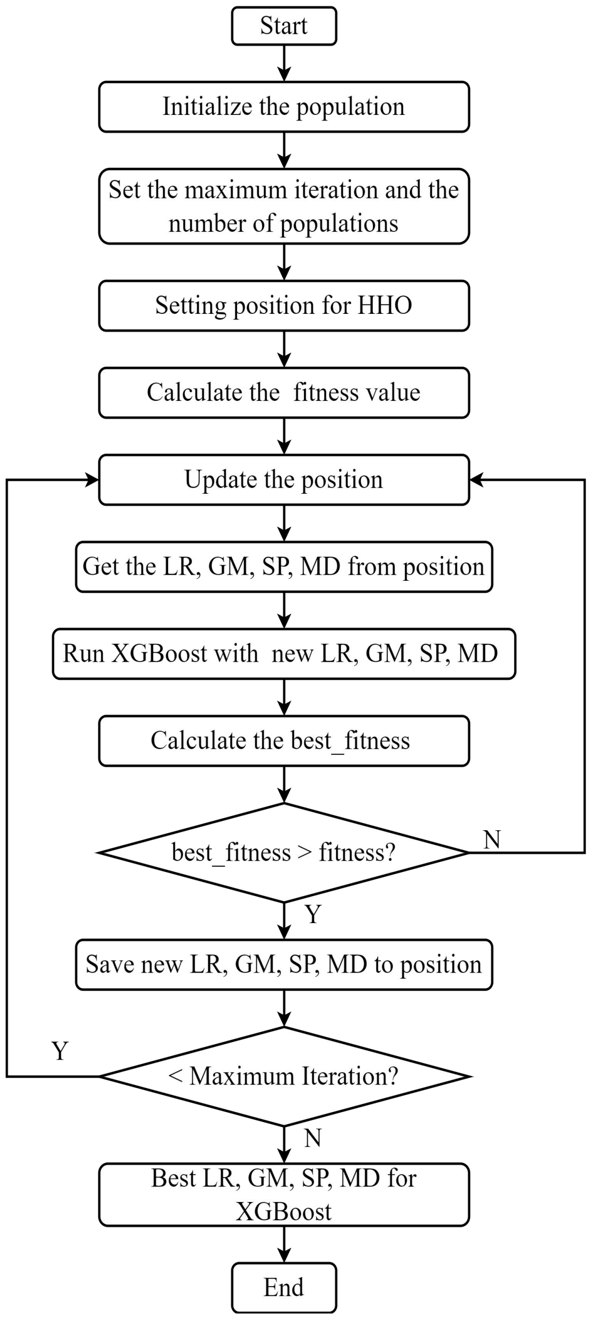

Fig. 7. Flowchart of the HHO-Optimized XGBoost Algorithm

Figure 7 depicts the procedure of the HHO optimizing XGBoost. During the optimization process, we need to initialize the population and set up the convergence curve. Find the optimal individual and ultimately determine the optimal parameters in the iterative exploration process. The main steps are as follows.

Step 1: Randomly initialize the population and determine the positions of the hawks and prey. The positions of the hawks contain the parameters(LR, GM, SP, and MD) in XGBoost.

Step 2: Calculate and update the positions of the hawks based on the prey's energy and escape probability.

Step 3: Use the updated positions' parameters as input to build the XGBoost model, then use the model to predict results. Repeat steps 2 and 3 until all positions are calculated.

Step 4: Search for the optimal global parameters among all positions.

Step 5: Before reaching the maximum iteration number, continue to execute steps 2, 3, and 4 until iterations are completed. Train the XGBoost model with these global parameters and select the best positions during the iteration process. Then record its parameters.

Step 6: Train the XGBoost model by the parameters, then use the model to evaluate the final classifier for identification and classification.

IV. RESULTS AND DISCUSSION

A. Experimental Setup and Parameter Configuration

All algorithms in this study were implemented in PyCharm using Python version 3.11.4. The experiments were conducted on a personal computer equipped with an AMD Ryzen 5 5600 6-core processor and 16.0 GB of RAM. The number of iterations for all algorithms was fixed at 150, with a population size of 30. The mutation probability of GA was set to 0.001. The gait data was divided into a 70% training set and a 30% testing set.

In the hyperparameter tuning of the optimized XGBoost algorithm, the LR was varied between 0.01 and 1, GM between 0 and 0.1, MD from 4 to 12, and SP was fixed at 0.1. Each algorithm was executed 10 times on the gait dataset to ensure the robustness and reliability of the results.

B. Performance Evaluation Metrics

Various performance evaluation metrics exist for assessing machine learning models. This study employs accuracy, precision, recall, and F1 score. Depending on the application, each of these metrics can provide unique insights into the model's performance. Each of these metrics is defined by specific mathematical formulas. By evaluating the model through them, the accuracy of the model can be more accurate and reliable.

C. Classifier Selection

During the initial stages of our research, we employed a range of algorithms for data classification, including natural gradient boosting (NGBoost), CatBoost, light gradient boosting machine (LightGBM), and XGBoost. The classification outcomes for each method are presented in Table I. The XGBoost significantly outperforms the other algorithms, achieving an accuracy of 97.22% and a notably short runtime of 0.3 seconds. This efficiency makes it highly practical for the classification of large datasets. Therefore, XGBoost has been chosen as the primary classifier for this study.

TABLE I
RECOGNITION RATES FOR FOUR ALGORITHMS

Algorithm	Precision	Recall	F1	Time
NGBoost	94.93%	90.76%	92.47%	40.9S
CatBoost	96.92%	96.44%	96.68%	6.6S
Lightgbm	97.06%	96.64%	96.85%	1.7S
XGBoost	97.22%	96.69%	96.95%	0.3S

D. Comparison of Optimization Algorithms

Throughout the study, we integrated four optimization algorithms (GA, AOA, COA, and HHO) with XGBoost. The hybrid algorithm resulting from the integration of GA is referred to as GA-XGBoost, with similar designations for the other algorithms, such as the AOA-XGBoost, the COA-XGBoost, and the HHO-XGBoost. Figure 8 presents the fitness curves of these algorithms through the iterative process. The HHO-XGBoost combination demonstrated superior fitness performance on the dataset used in this study, with the fitness value reaching 98.40% by the 60th iteration. Compared to other optimization algorithms, HHO-XGBoost exhibited faster convergence and greater stability, which is critical for real-time gait analysis applications. Table II presents the optimal hyperparameter values for each algorithm when their fitness is maximized.

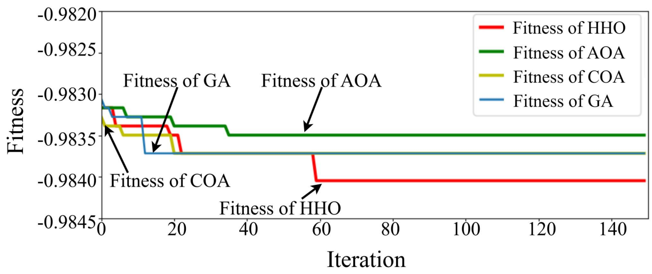

Fig.8. Fitness curves of different algorithms

TABLE II
COMPARISON OF HYPERPARAMETERS SELECTION

Algorithm	LR	GM	MD	SP
XGBoost	0.1	0	6	1
GA-XGBoost	0.1668	0	12	0.9551
AOA-XGBoost	0.2576	0.0050	11	1
COA-XGBoost	0.2580	0.0298	10	0.5055
HHO-XGBoost	0.5462	0.0692	12	1

E. Comparison of Algorithm Stability

To ensure the robustness of the algorithm and avoid the potential bias of evaluating a single dataset, we applied the HHO-XGBoost algorithm to the datasets of 10 different subjects. As shown in Table III, the test results indicate significant variations for different individuals. For example, the result from Male 5 achieved an accuracy of 99.11%, while the result from Male 6 obtained an accuracy of 95.24%. These discrepancies may stem from differences in foot force distribution due to varying exercise intensities among the subjects. Non-uniform force application can introduce biases during the data normalization process, which may lead to labeling errors. It is noteworthy that the 99.11% accuracy achieved by the result from Male 5 corresponds to data collected during smooth and consistent motion.

Despite varying effects across different participant datasets, the impact of the HHO-XGBoost algorithm on each individual's data is notably significant. Compared to other algorithms, this optimization technique achieves the most substantial improvement in accuracy. This finding further underscores the exceptional performance and stability of HHO-XGBoost in gait recognition tasks.

TABLE III
RECOGNITION RATES OF FIVE ALGORITHMS ON DATA FROM TEN TESTERS

Testers	XGBoost	GA-XGBoost	AOA-XGBoost	COA-XGBoost	HHO-XGBoost
Male 1	96.40%	96.51%	96.53%	96.49%	96.56%
Male 2	97.26%	97.38%	97.35%	97.39%	97.44%
Male 3	98.35%	98.43%	98.45%	98.47%	98.52%
Male 4	98.87%	98.97%	98.97%	99.00%	99.05%
Male 5	95.84%	95.92%	95.94%	95.97%	96.01%
Male 6	98.82%	98.90%	98.89%	98.94%	98.99%
Female 1	95.91%	96.02%	96.03%	96.05%	96.09%
Female 2	98.98%	99.09%	99.11%	99.10%	99.15%
Female 3	99.24%	99.32%	99.30%	99.36%	99.44%
Female 4	98.75%	98.87%	98.84%	98.89%	98.91%

F. *Results and discussion*

Table IV illustrates the results of various algorithms in classifying the gait patterns. Compared to other algorithms, the HHO-XGBoost algorithm indicate notable high results in four metrics. The HHO-XGBoost achieves higher in accuracy, precision, recall, and the F1 score by 0.35%, 0.36%, 0.47%, and 0.42% than the original XGBoost. The HHO showed better performance compared with the other optimization algorithms. The HHO-XGBoost algorithm realized the best performance in gait classification. The results confirm the feasibility and superiority of the HHO in optimizing the hyperparameters.

TABLE IV
COMPARISON OF FIVE METHODS IN FOUR METRICS

Algorithm	Accuracy	Precision	Recall	F1 Score
XGBoost	97.50%	97.05%	96.56%	96.80%
GA-XGBoost	97.78%	97.35%	96.94%	97.15%
AOA-XGBoost	97.76%	97.38%	96.96%	97.17%
COA-XGBoost	97.79%	97.37%	96.98%	97.19%
HHO-XGBoost	97.85%	97.41%	97.03%	97.22%

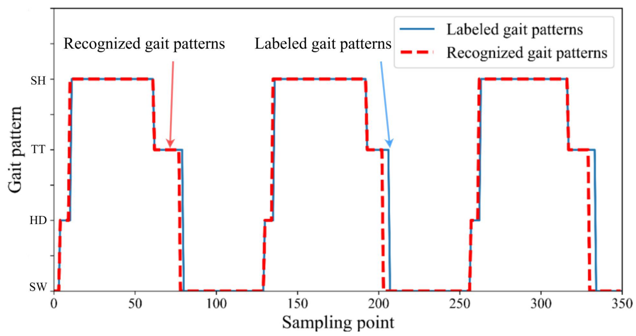

Fig.9. Comparison of labels and results

Figure 9 shows the recognition results compared with the labeled ones. The patterns identified by the HHO-XGBoost algorithm is extremely similar to the labeled ones, and the proposed method gained a high accuracy in gait division.

Figure 10 illustrates the recognition accuracy of the five different classifiers in categorizing four gait phases. The HHO-XGBoost algorithm achieved classification accuracy exceeding 95.31% across all actions. The SW and HD states even reached 98.74% and 98.57%. On the four states, the HHO-XGBoost algorithm showed the largest improvement in accuracy compared to the original XGBoost algorithm. The TT and SH states are susceptible to classification errors. The HHO-XGBoost algorithm improved the recognition rates of these two actions by 0.78% and 0.69%. Although the other three algorithms also demonstrated improvements in action recognition accuracy, their performance was inferior to that of the HHO-XGBoost. The confusion matrix is shown in Figure 11. Overall, the performance of the proposed method remains exceptionally strong.

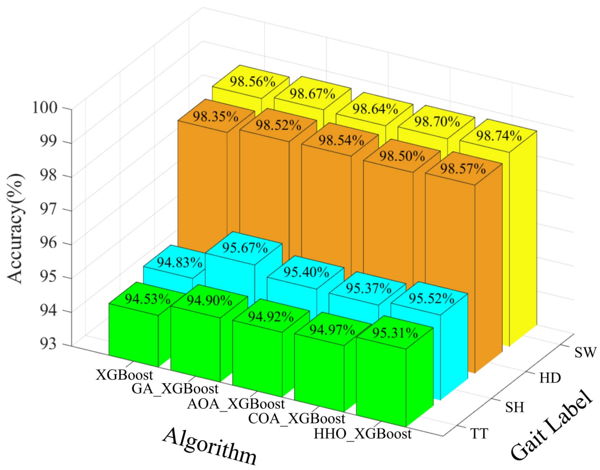

Fig.10. Accuracy of five classifiers for four phase classifications

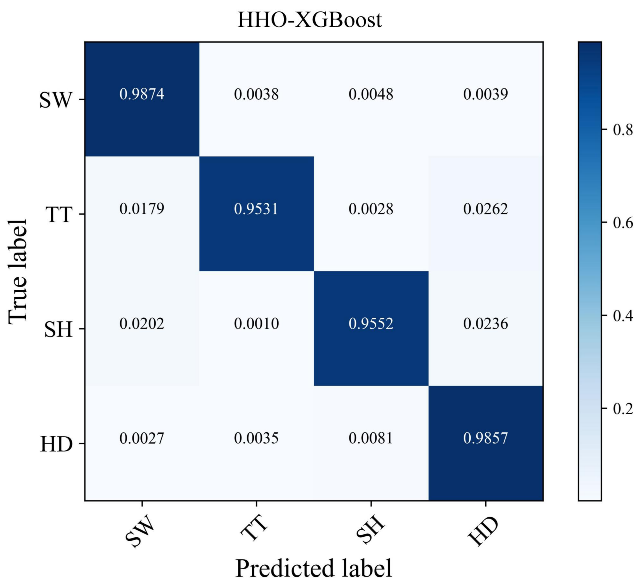

Fig.11. Confusion matrix for HHO-XGBoost algorithm

The results indicate that the multi-classification model HHO-XGBoost is capable of accurately identifying four distinct gait stages. The model demonstrates high recognition accuracy and practical applicability. This capability is essential for accurately assessing actions and facilitating timely adjustments in smart wearable devices.

V. CONCLUSION

In this study, an intelligent shoe is designed and integrated with six FSRs to collect the GRFs' information about human movement during walking. The XGBoost is used to classify the gait phase in the walking period. The

HHO algrithm was selected to optimize four hyperparameters of the XGBoost model. The experimental results demonstrate that the HHO-XGBoost algorithm achieved an accuracy of 97.85% for gait phase recognition tasks. The recognition rates for all four actions exceeded 95.31%, with the highest reaching 98.74%. Compared to other algorithms, the HHO-XGBoost algrithm exhibits superior performance and reliability. This study lays a solid foundation for future exploration in the field of gait recognition, particularly in scenarios where similar methods may be applied to various datasets. It also underscores the importance of continuous innovation in algorithmic approaches within the machine learning domain.

REFERENCES

[1] A. S. Moghaddam, and A. Etemad, "Deep Gait Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 264 - 284, 2023.

[2] J. C. Quillet, M. Siani-Rose, R. McKee, B. Goldstein, M. Taylor, and I. Kurek, "A machine learning approach for understanding the metabolomics response of children with autism spectrum disorder to medical cannabis treatment," *Scientific Reports*, pp. 13022, 2023.

[3] S. A. Raed, and Faqir, "Digital Criminal Investigations in the Era of Artificial Intelligence: A Comprehensive Overview," *International Journal of Cyber Criminology*, vol. 17, no. 2, pp. 77-94, 2023.

[4] M. Boukabous, and M. Azizi, "Image and video-based crime prediction using object detection and deep learning," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1630-1638, 2023.

[5] S. B. Slama, and M. Mahmoud, "A deep learning model for intelligent home energy management system using renewable energy," *Engineering Applications of Artificial Intelligence*, vol. 123, pp. 106388, 2023.

[6] Y. Q. Yu, Z. L. Hao, G. J. Li, Y. Q. Liu, R. Yang, and H. H. Liu, "Optimal search mapping among sensors in heterogeneous smart homes," *International Journal of Cyber Criminology*, vol. 20, no. 2, pp. 1960-1980, 2022.

[7] S. Qiu, Z. C. Pei, C. Wang, and Z. Y. Tang, "Systematic Review on Wearable Lower Extremity Robotic Exoskeletons for Assisted Locomotion," *Journal of Bionic Engineering*, vol. 20, no. 2, pp. 436-469, 2023.

[8] X. Ma, X. X. Zhang, and J. Xu, "Robotic Leg Prosthesis: A Survey From Dynamic Model to Adaptive Control for Gait Coordination," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 32, pp. 607-624, 2024.

[9] B. Chen, C. Y. Chen, et al., "Computer Vision and Machine Learning-Based Gait Pattern Recognition for Flat Fall Prediction," *Sensors*, vol. 22, no. 20, 2022.

[10] C. Fan, Y. J. Peng, et al., "GaitPart: Temporal Part-based Model for Gait Recognition," *IEEE conference on Computer Vision and Pattern Recognition*, pp. 14225-14233, 2020.

[11] M. Mangalam, D. G. Kelty- Stephen, I. Seleznov, A. Popov, A. D. Likens, K. Kiyono, and N. Stergiou, "Older adults and individuals with Parkinson’s disease control posture along suborthogonal directions that deviate from the traditional anteroposterior and mediolateral directions," *Scientific Reports*, vol. 14, no. 1, 2024.

[12] S. Lokavee, J. Pengjiam, V. Tantrakul, and T. Kerdcharoen, "Innovative Sleep Monitoring: A Non-Invasive Approach Using Force-Sensing Resistors for Analyzing Sleep Quality and Detecting Sleep-Related Breathing Disorders," *International Journal of Geoinformatics*, vol. 20, no. 3, pp. 17-27, 2024.

[13] S. K. Manna, M. A. H. B. Azhar, and A. Greace, "Optimal locations and computational frameworks of FSR and IMU sensors for measuring gait abnormalities," *Heliyon*, vol. 9, no. 4, 2024.

[14] F. Gao, T. Tian, T. Yao, and Q. Zhang, "Human gait recognition based on multiple feature combination and parameter optimization algorithms," Computational Intelligence and Neuroscience, 2021.

[15] V. Monchiet, and G. Bonnet, "FFT based iterative schemes for composite conductors with uniform boundary conditions," European Journal of Mechanics-A/Solids, vol. 103, 2024.

[16] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," Cognitive Robotics, vol. 3, pp. 54-70, 2023.

[17] Z. P. Yao, Y. W. Lum, et al., "Machine learning for a sustainable energy future," Nature Reviews Materials, vol. 8, no. 3, pp. 202-215, 2023.

[18] L. Yu, G. T. Hu, L. Ding, N. Luo, and Y. Zhang, "Gait Pattern Recognition based on Multi-sensors Information Fusion through PSO-SVM Model," Engineering Letters, vol. 32, no. 5, pp. 974-980, 2024.

[19] X. L. Sun, and J. Q. Fu, "Many-objective optimization of BEV design parameters based on gradient boosting decision tree models and the NSGA-III algorithm considering the ambient temperature," Energy, vol. 228, 2024.

[20] G. Abdurrahman, and M. Sintawati, "Implementation of XGBoost for classification of parkinson's disease," Journal of Physics: Conference Series, vol. 1538, no. 1, pp. 436-469, 2020.

[21] C. Wang, C. Y. Deng, and S. Z. Wang, "Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost," Pattern Recognition Letters, vol. 136, pp. 190-197, 2020.

[22] H. Jiang, Z. He, G. Ye, and H. Y. Zhang, "Network Intrusion Detection Based on PSO-XGBoost Model," IEEE Access, vol. 8, pp. 58392-58401, 2020.

[23] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, "Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection," Sensors, vol. 24, no. 4, 2024.

[24] Aslan, Serpil, S. Kızıloluk, and E. Sert. "TSA-CNN-AOA: Twitter sentiment analysis using CNN optimized via arithmetic optimization algorithm," Neural Computing and Applications, vol. 35, no. 14, pp. 10311–10328, 2023.

[25] R. Somula, Y. Cho, and B. K. Mohanta. "EACH-COA: an energy-aware cluster head selection for the internet of things using the coati optimization algorithm," Information, vol. 14, no. 11, pp. 601, 2023.

[26] W. Y. Guo, P. Xu, F. Dai, F. Q. Zhao, and M. F. Wu, "Improved Harris hawks optimization algorithm based on random unscented sigma point mutation strategy," Applied Soft Computing, vol. 113, 2021.

[27] A. A. Dehkordi, A. S. Sadiq, S. Mirjalili, and K. Z. Ghafoor, "Nonlinear-based Chaotic Harris Hawks Optimizer: Algorithm and Internet of Vehicles application," Applied Soft Computing, vol. 109, 2021.

[28] J. Jafari-Asl, M. E. A. B. Seghier, S. Ohadi, J. Correia, and J. Barroso, "Reliability analysis based improved directional simulation using Harris Hawks optimization algorithm for engineering systems," Engineering Failure Analysis, vol. 135, 2022.

[29] X. Y. Wang, Y. Yao, J. L. Lin, D. Chen, Y. Cai, S. Q. Li, Y. L. Shang, and F. Z. Gao, "Indoor Localization Method Enhanced by an Improved Harris's Hawk Optimization-Based Particle Filter," IAENG International Journal of Computer Science, vol. 51, no. 3, pp. 267-275, 2024.