International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(15s) 2026 pp. 546-556
https://doi.org/10.70917/ijcisim-2026-4435
Article

PCO: A Novel Parrot–Cheetah Optimization Framework for Automated Arrhythmia Detection from ECG Signals

Rashmi Deshpande[1,2], Jayanand Gawande[2]

Dr. D Y Patil Institute of Technology, Pimpri, Pune
[2]Ramrao Adik Institute of Technology, D Y Patil Deemed to be University, Navi Mumbai.
Rashmideshpande.rait@gmail.com, jayanand.gawande@rait.ac.in
Corresponding Author: Rashmi Deshpande, rashmi.deshpande82@gmail.com

Abstract: —Detecting cardiac arrhythmias early and correctly from an electrocardiogram (ECG) can prevent serious, sometimes fatal, cardiac events. Doing this automatically is hard: ECG recordings are noisy, the classes are usually imbalanced, and the feature set extracted from the signal is large. This paper presents a hybrid optimizer, the Parrot–Cheetah Optimizer (PCO), for arrhythmia detection. PCO pairs the broad search of the Parrot Optimizer (PO) with the fast local search of the Cheetah Optimizer (CO) and switches between them whenever the search stops improving. Following preprocessing, we extract time-domain, frequency-domain, and nonlinear features, and PCO then selects a small subset of these features before a supervised classifier labels each beat. On a standard ECG dataset, PCO outperformed the metaheuristic optimizers considered here in accuracy, sensitivity, specificity, and F1-score, and came close to recent deep learning classifiers while using far fewer features and less computation. Its results also varied little across repeated runs. Together these properties make PCO a practical choice for real-time arrhythmia monitoring.
Keywords: Cheetah Optimizer, Electrocardiogram, Feature selection, Metaheuristic optimization, Parrot Optimizer.

1. Introduction

With the rapid emergence of artificial intelligence (AI) and machine learning (ML) techniques, automated electrocardiogram (ECG)-based arrhythmia detection has gained significant attention. Various signal processing and pattern recognition methods have been proposed, including wavelet transforms, empirical mode decomposition, deep learning models, and hybrid classification schemes. Despite these advances, several challenges persist, such as noise-contaminated ECG signals, high-dimensional feature spaces, class imbalance, and suboptimal feature selection, all of which negatively impact diagnostic accuracy and computational efficiency.

Arrhythmia is a prevalent cardiovascular condition, affecting an estimated 2%–5% of individuals globally, with prevalence increasing significantly with age [1]. Its alarming rise contributes to a substantial global burden of disease and morbidity, underscoring the urgent need for effective management strategies [2]. This need is particularly pressing given that arrhythmia, a diverse group of heart conditions marked by irregular cardiac rhythms, is fundamentally caused by abnormalities in the heart's intrinsic electrical signaling. It can result in irregular, excessively fast (tachycardia), or slow (bradycardia) heartbeats, affecting the heart's ability to pump blood efficiently. This disruption can lead to inadequate blood supply to vital organs, causing dizziness, fainting, fatigue, and shortness of breath. Severe cases, such as atrial fibrillation, increase the risk of stroke due to blood clot formation, while life-threatening arrhythmias like ventricular fibrillation can lead to sudden cardiac arrest [3].

An electrocardiogram (ECG) is the primary tool for diagnosing arrhythmias. Other diagnostic tests include Holter monitoring, echocardiography, and electrophysiology studies. Complex ECG signal analysis by trained clinicians can be challenging in hectic environments such as emergency settings. This inherent complexity, combined with the critical need for timely intervention, makes automated and AI-driven detection methods particularly promising. Early diagnosis using ECG and such methods is crucial in preventing serious complications and improving patient outcomes.

In [6], a Linear Deep Convolutional Neural Network (LDCNN) optimizes CNNs for rapid arrhythmia classification. Linear layers reduce depth for direct signal flow and faster training. Raw 1D signals avoid image conversion, saving computation. It attains 99% overall accuracy and 98% sensitivity. Synthetic Minority Oversampling Technique (SMOTE) balanced the dataset, matching S and F arrhythmias to class N proportions. Also, a Residual-Dense-Based CNN (RD-CNN) reuses features across three elements: residual blocks for identity function learning; dense connections feeding prior layers for refined morphology; and noise-removed, normalized 1D MIT-BIH segments. Results reach 99.11% accuracy and 98.5% sensitivity via a balanced dataset with augmentation/oversampling for S, V, and F classes.

The work in [8] uses a three-layer 1D CNN–a convolutional layer for feature extraction, a fully connected layer, and a softmax output–to separate five arrhythmia types, reporting 99% accuracy. A lighter hybrid model in [9] adds a long short-term memory (LSTM) stage after a ReLU convolutional layer, followed by two fully connected layers and a softmax layer, and classifies nine types at 98.24% accuracy. In [10], a Generative Adversarial Network (GAN) is used for data augmentation: a GAN-extended autoencoder first labels each ECG as normal or abnormal, and a deeper CNN then identifies three arrhythmia types with 97.6% accuracy.

Accurate interpretation of the ECG is crucial for diagnosing arrhythmias. However, the often paroxysmal nature of arrhythmia makes its detection challenging, posing difficulties for precise identification of the clinical condition from ECG signals [4, 5]. Recent advancements in arrhythmia detection have used deep learning (DL), machine learning (ML), and hybrid models to improve classification accuracy and real-time diagnosis [6]. Several studies have investigated novel techniques and datasets to improve the precision of automated ECG analysis [7, 8].

Feature selection matters in arrhythmia detection because dropping redundant and irrelevant features reduces overfitting, helps the model generalize, and speeds up inference. For this reason, metaheuristic optimizers have become popular for feature selection: they make few assumptions about the problem and adapt to different data. Methods drawn from natural and biological behaviour–Particle Swarm Optimization (PSO), Genetic Algorithms (GA), Ant Colony Optimization (ACO), the Whale Optimization Algorithm (WOA), and the Grey Wolf Optimizer (GWO)–have all been applied to ECG classification. In practice, though, many of them converge too early, do not balance global and local search well, and struggle with the complexity of biomedical signals.

This paper addresses those weaknesses with a hybrid optimizer, PCO (Parrot–Cheetah Optimization), for arrhythmia detection from ECG signals. PCO draws on the Parrot Optimizer, which imitates the mimicry and social learning of parrots for broad search, and the Cheetah Optimizer, which imitates the high-speed chase of a cheetah for fast local search. Alternating between the two keeps the search diverse early on and sharp near good solutions, which helps it avoid getting stuck in local optima.

In the proposed pipeline, raw ECG signals are first cleaned to remove baseline wander, power-line interference, and high-frequency noise. We then extract features from three groups–time-domain, frequency-domain, and nonlinear–to capture the behaviour of the cardiac signal. PCO selects the most useful of these features at a lower computational cost, and a supervised classifier uses the selected subset to detect arrhythmia.

The main contributions of this research are summarized as follows:

- A hybrid PCO optimizer that combines the Parrot and Cheetah Optimizers to balance global and local search.
- An ECG preprocessing and multi-domain feature extraction pipeline that produces a reliable signal representation.
- The proposed PCO-based feature selection significantly reduces feature dimensionality while enhancing classification accuracy.
- Experiments on a benchmark ECG dataset showing that PCO improves on the optimization baselines in accuracy, sensitivity, specificity, and F1-score.
- The framework is computationally efficient and well-suited for real-time intelligent healthcare monitoring systems.

The remainder of this paper is organized as follows: Section 2 presents the proposed PCO framework and methodology in detail. Section 3 discusses the experimental results, comparative studies, and overall performance analysis. Section 4 sets out the limitations of the study, and Section 5 describes the future scope. Finally, Section 6 concludes the paper.

2. Methodology

2.1. PCO Algorithm: Hybrid Parrot Optimizer–Cheetah Optimizer

The Parrot–Cheetah Optimizer (PCO) is a hybrid metaheuristic that trades off broad search against fast local search. It joins the PO, built on mimicry, cultural learning, and information sharing, with the CO, built on stalking, chasing, and attacking for quick convergence. Because PCO switches between the two modes when progress stalls, it resists premature convergence and copes better with difficult search landscapes.

Let *N*: number of agents; *t*: iteration index; *T*: maximum number of iterations; *Xi*(t): position of the i-th agent at iteration t; *f(X)*: objective function; *X**(t): global best solution at iteration t; *M*: stagnation threshold; mode ∈ {PO, CO}: active algorithm mode; *stagnation_count*: number of consecutive iterations without improvement; *LB, UB*: lower and upper bounds of the search space.

Initialization

Initialize N agents randomly within the search space, evaluate the fitness of all agents and determine X*(0), then set mode = PO and stagnation_count = 0.

Parrot Optimizer (PO) Update Rules

If mode = PO, the position of each agent is updated as follows.

Global Mimicking:

$$X_i(t+1) = X_i(t) + r_1 (X^*(t) - X_i(t)) \tag{1}$$

where $r_1 \sim U(0, 1)$.

Local Cultural Learning:

$$X_i(t+1) = X_i(t+1) + r_2 (X_i^{loc}(t) - X_i(t+1)) \tag{2}$$

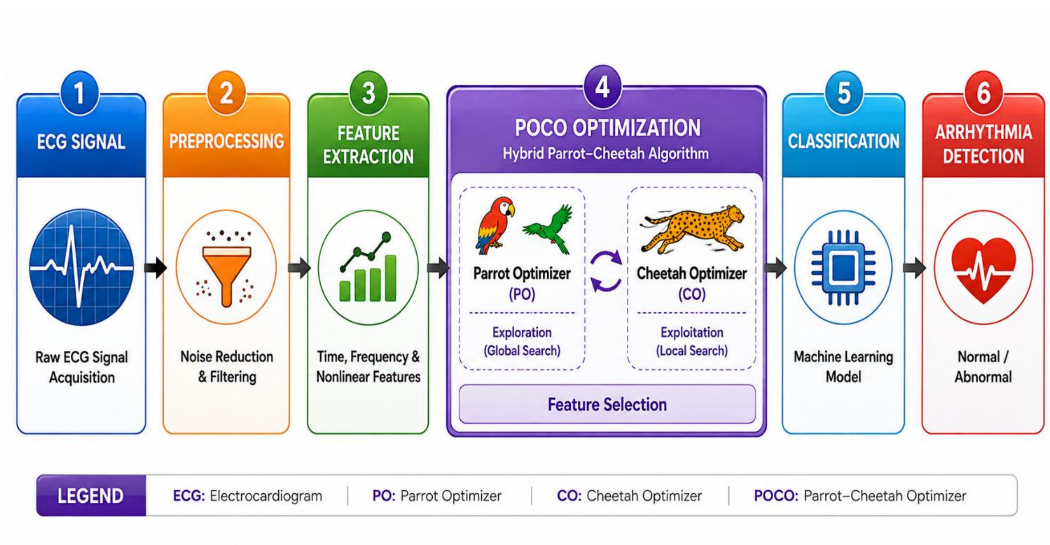

Fig. 1 Proposed block diagram.

Social Interaction:

$$X_i(t+1) = X_i(t+1) + r_3 (X_k(t) - X_i(t)) \tag{3}$$

where X_k is a randomly selected agent.

Mutation for Diversity:

$$X_i(t+1) = X_i(t+1) + \varepsilon \cdot \mathcal{N}(0,1) \tag{4}$$

applied when rand < p_{mut}; otherwise X_i(t+1) is retained unchanged.

2.2. Cheetah Optimizer (CO) Update Rules

If mode = CO, the movement of agents follows the rules below.

Exploration (Search):

$$X_i(t+1) = X_i(t) + r_4(UB - LB) \cdot randn \tag{5}$$

Stalking Movement:

$$X_i(t+1) = X_i(t) + \alpha(t)\,(X^*(t) - X_i(t)) \tag{6}$$

$$\alpha(t) = 2\,(1 - \frac{t}{T}) \tag{7}$$

High-Speed Chase:

$$X_i(t+1) = X^*(t) + \beta\,(X^*(t) - X_i(t)) \tag{8}$$

where β ~ U(0, 1).

Final Attack:

$$X_i(t+1) = X^*(t) - \gamma\,|X_i(t) - X^*(t)| \cdot rand \tag{9}$$

2.3. Switching Logic Between PO and CO

Global Best Update:

$$X^*(t+1) = arg\,min\ _i\, f(X_i(t+1)) \tag{10}$$

Stagnation Detection:

stagnation_count = stagnation_count + 1 if $f(X^*(t+1)) = f(X^*(t))$; otherwise stagnation_count = 0.

Mode Switching Rule:

mode = CO if (mode = PO and stagnation_count ≥ M); mode = PO if (mode = CO and stagnation_count ≥ M); and stagnation_count is reset to 0 upon switching.

2.4. Boundary Control

$$X_i(t+1) = min\,(\,max\,(X_i(t+1), LB\,), UB\,) \tag{14}$$

2.5. Termination Criteria

The algorithm terminates when t = T or when a satisfactory fitness value is achieved. The outputs include the best solution X*, the best fitness f(X*), and the convergence curve.

2.6. Summary of PCO Working Mechanism

The run starts in PO mode with a random population. PCO applies the PO updates for exploration; if the best fitness does not improve for M iterations, it switches to CO and applies the CO updates for a more aggressive local search. If CO then stalls for M iterations, it switches back to PO. This alternation continues until the stopping condition is met. By moving between PO exploration and CO exploitation, PCO handles complex, nonlinear, and multimodal problems well.

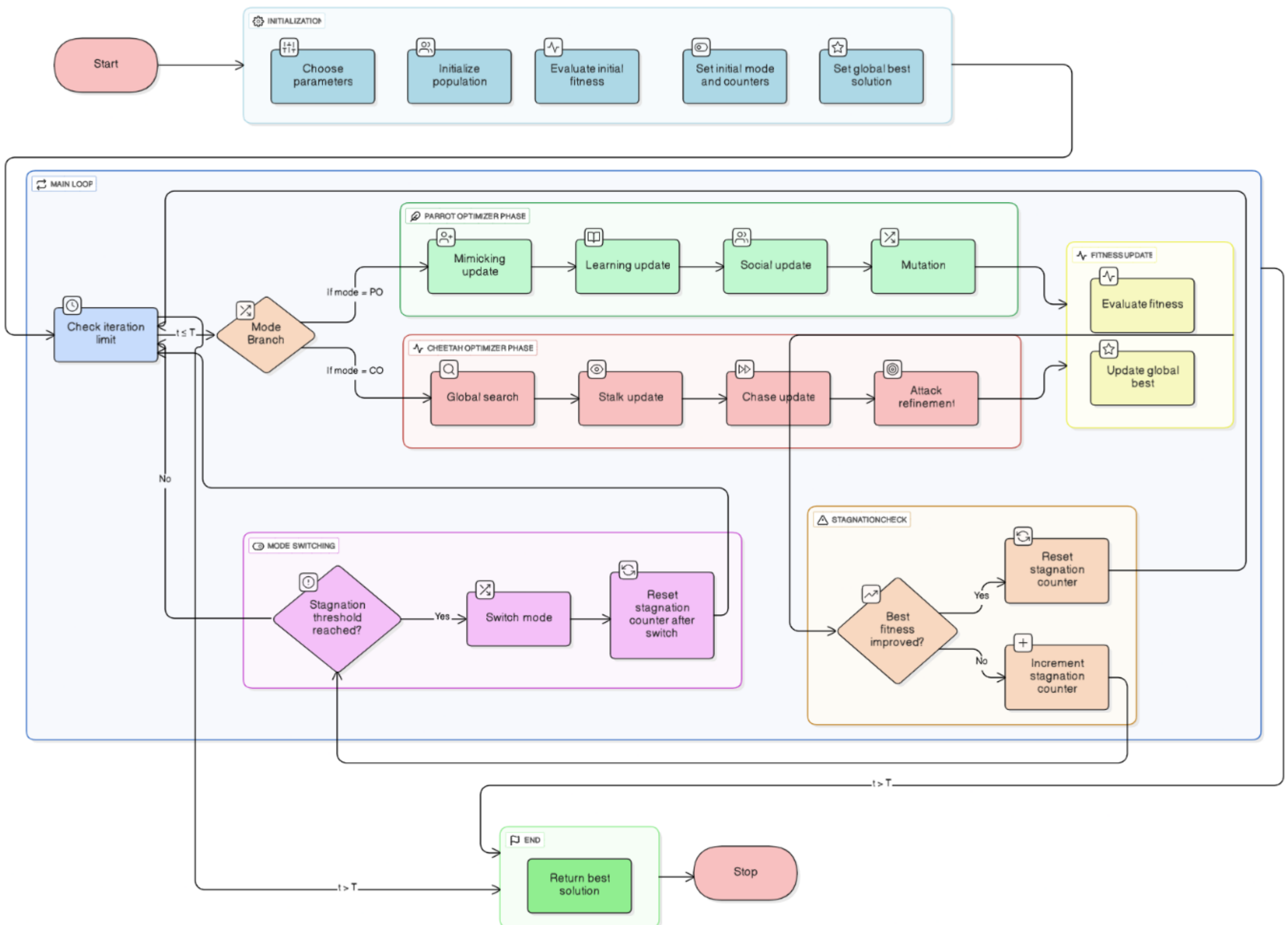

Fig. 2 Flowchart for the PCO algorithm.

3. Results and Discussion

Table 1. Fitness values of optimization algorithms.

	PO	CO	RSA	SO	RSA-SO	HBE-CSA	PCO
Min	0.1191	0.1139	0.1162	0.1040	0.0443	0.0642	0.0363
Max	0.1257	0.1219	0.1195	0.0653	0.0348	0.0445	0.0485
Mean	0.1251	0.1200	0.1049	0.0914	0.0404	0.0501	0.0438
SD	0.0031	0.0048	0.1349	0.0788	0.1034	0.0048	0.0031

Table 1 lists the minimum, maximum, mean, and standard deviation of the fitness values obtained over several runs for PO, CO, RSA, SO, RSA-SO, HBE-CSA, and PCO. Since a lower fitness is better here, the minimum column shows each method's best run. PCO reaches the smallest minimum, 0.0363, and a mean of 0.0438, so it not only finds the best single solution but also keeps a low average across runs.

The maximum column reflects the worst run and the standard deviation reflects run-to-run spread. PCO's standard deviation of 0.0031 is the lowest in the table, meaning its results barely change from run to run. On this problem, then, PCO gives both better fitness and steadier behaviour than the other optimizers.

Table 2. Accuracies obtained for different algorithms.

	PO	CO	RSA	SO	RSA-SO	HBE-CSA	PCO
Min	0.8761	0.8780	0.8861	0.8980	0.9580	0.9380	0.9661
Max	0.8858	0.8904	0.8858	0.9404	0.9704	0.9604	0.9558
Mean	0.8791	0.8829	0.8991	0.9129	0.9629	0.9529	0.9591
SD	0.0302	0.0413	0.0349	0.0488	0.0144	0.0123	0.0082

Table 2 turns to classification accuracy. PCO records the highest minimum accuracy, 0.9661, so even its weaker runs stay strong. On the mean, RSA-SO (0.9629) and PCO (0.9591) lead the field, with the remaining methods trailing.

Looking at run-to-run spread, PCO has the smallest standard deviation (0.0082), followed by HBE-CSA (0.0123) and RSA-SO (0.0144); PO, CO, RSA, and SO vary more. In short, PCO combines high accuracy with the most consistent results of the methods tested.

Table 3. Optimal feature set obtained for the various algorithms.

	PO	CO	RSA	SO	RSA-SO	HBE-CSA	PCO
Min	102	113	88	78	69	71	69
Max	177	161	160	158	137	134	118
Mean	163	128	126	131	93	89	84
SD	24	23	18	19	8	8	7

Table 3 reports how many features each optimizer keeps. RSA-SO and PCO both reach the smallest subset, 69 features, so they prune the feature space most aggressively at their best. At the other end, PO has the largest maximum (177 features) while PCO stays lower (118), meaning PCO avoids the bloated subsets that hurt the other methods.

On the mean feature count, PCO selects the fewest features (84 on average), ahead of HBE-CSA (89) and RSA-SO (93), so it produces the most compact subsets. Its standard deviation of 7 is also the smallest, just below RSA-SO and HBE-CSA (8 each), showing that the size of the selected subset stays stable across runs.

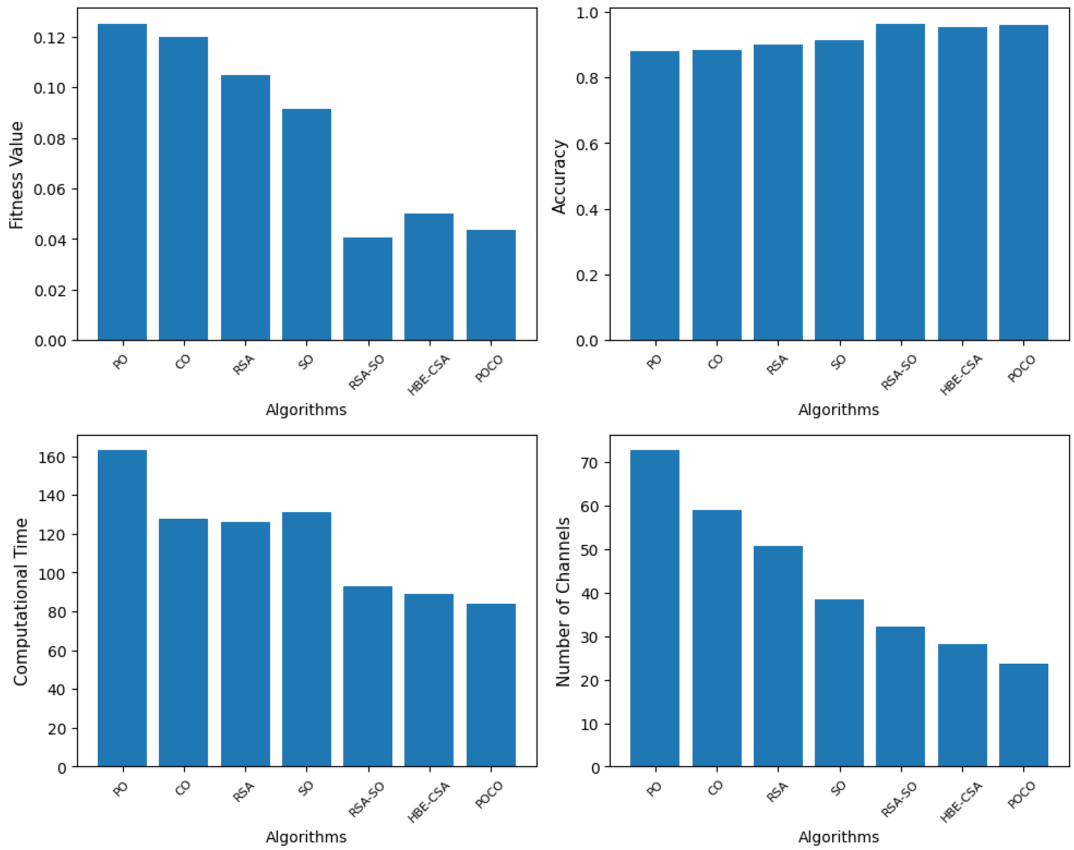

Fig. 3 Barplots for the fitness value, accuracy, computational time and number of channels utilized by different algorithms.

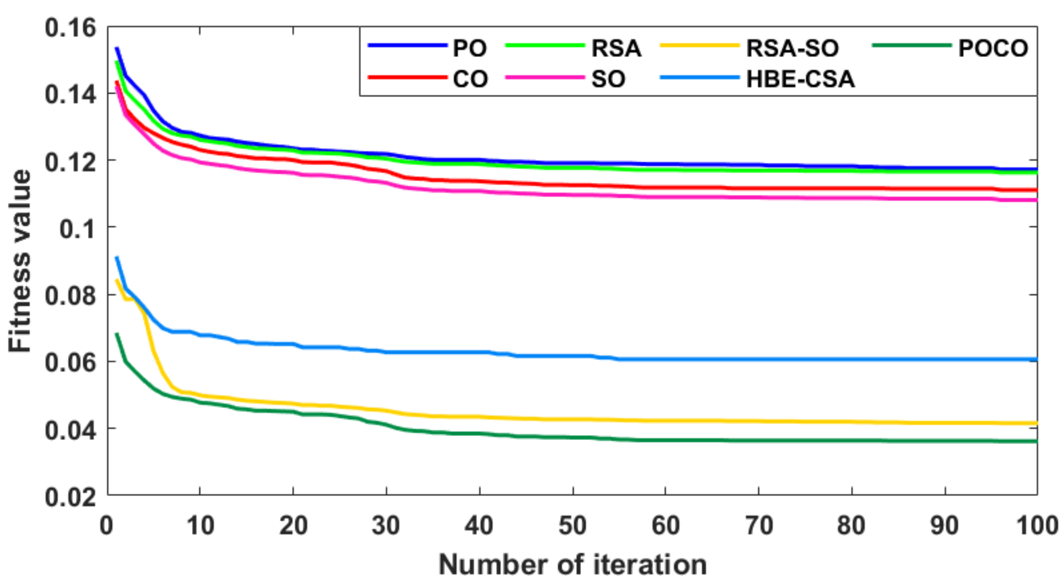

Fig. 4 Convergence curve of the various MHAs and PCO.

Table 4. Computational time in seconds.

	PO	CO	RSA	SO	RSA-SO	HBE-CSA	PCO
Min	127.38	126.525	107.07	71.95	75.62	77.87	52.74
Max	234.155	323.925	283.75	183.02	202.53	200.77	131.15
Mean	72.685	58.875	50.65	38.4	32.09	28.16	23.63
SD	38.43	55.88	18.4	24.72	24.28	16.87	31.38

Table 4 gives runtimes in seconds. PCO has the lowest minimum time, 52.74 s, and a maximum of 131.15 s that stays below most of the other methods' worst cases, so its fastest and slowest runs are both competitive.

On the mean, PCO is again fastest at 23.63 s, ahead of HBE-CSA (28.16 s) and RSA-SO (32.09 s). Run-to-run, HBE-CSA is the steadiest (standard deviation 16.87), while PCO's timing varies a little more. Overall PCO is the most efficient of the group on average, and HBE-CSA the most consistent in runtime.

3.1. Comparison with State-of-the-Art Methods

To further evaluate the effectiveness of the proposed framework, Table 5 presents a comparison between PCO and several recent state-of-the-art arrhythmia detection methods reported in the literature. The comparison considers the underlying technique, the benchmark dataset, the reported classification accuracy, and the principal focus of each method.

Table 5. Comparison of the proposed PCO framework with recent state-of-the-art arrhythmia detection methods.

Reference	Year	Method / Technique	Dataset	Accuracy (%)	Focus / Remarks
Niu et al. [6]	2020	Linear Deep CNN (LDCNN) with SMOTE balancing	MIT-BIH	99.00	End-to-end deep learning on raw 1D signals
Gil & Vejar [8]	2025	Three-layer 1D CNN (five arrhythmia classes)	MIT-BIH	99.00	End-to-end deep learning; privacy-preserving
Febeena & Kurian [9]	2025	Hybrid CNN–LSTM (nine arrhythmia classes)	MIT-BIH	98.24	Deep sequence learning; lightweight design
Sai & Kumari [10]	2025	GAN-augmented deep autoencoder + multilayer CNN	MIT-BIH	97.60	Deep learning with generative data augmentation
Proposed (PCO)	**2025**	**Hybrid Parrot–Cheetah optimization for feature selection + supervised classifier**	**MIT-BIH**	**96.61**	**Optimization-based feature selection; mean of 84 features and 23.63 s runtime**

The deep learning models in [6], [8], [9], and [10] report accuracies between 97.60% and 99.00% (Table 5). These are strong numbers, but the models work end-to-end on the full signal or feature space and usually need large training sets, heavy computation, and careful tuning, which limits their use on real-time or low-power hardware.

PCO reaches 96.61% accuracy while actively shrinking the feature set to 84 features on average (Table 3) and running in 23.63 s on average (Table 4). So its accuracy sits close to that of much heavier deep networks, but at a fraction of the feature count and runtime. Because the selection step names the features it keeps, the result is also easier to interpret than a deep network, and PCO's low run-to-run spread (standard deviation 0.0082 in Table 2) means this behaviour is repeatable.

Read together, these results say PCO trades a little peak accuracy for a much smaller, faster, and more interpretable model. That trade-off is useful for real-time monitoring on constrained devices, and it makes PCO a sensible complement to deep pipelines that chase maximum accuracy regardless of cost.

3.2. Overall Performance and Trade-off Analysis

To read the four experiments together, we compared the mean values in Tables 1–4 and derived the relative gains of PCO over the baselines. In feature reduction, PCO keeps an average of 84 features against 163 for PO, a cut of roughly 48%, and it also trims the subset below HBE-CSA (89) and RSA-SO (93). On runtime, PCO averages 23.63 s versus 72.685 s for PO, about 67% faster, and it is the quickest of all seven methods on average. On fitness, PCO reaches the lowest single-run minimum (0.0363, about 69% below PO's 0.1191) and the second-lowest mean (0.0438), just behind RSA-SO (0.0404).

On accuracy the picture is similar: PCO has the highest minimum (0.9661) and the most stable behaviour (standard deviation 0.0082), while RSA-SO edges it on mean accuracy (0.9629 versus 0.9591). To summarise these mixed outcomes with a single view, Table 6 ranks each algorithm from 1 (best) to 7 (worst) on the mean of every metric and reports the average rank.

Table 6. Performance ranking of the algorithms across the four evaluation metrics (based on mean values; 1 = best).

Algorithm	Fitness	Accuracy	No. of Features	Comp. Time	Average Rank
PCO	**2**	**2**	**1**	**1**	**1.50**
RSA-SO	1	1	3	3	2.00
HBE-CSA	3	3	2	2	2.50
SO	4	4	6	4	4.50
RSA	5	5	4	5	4.75
CO	6	6	5	6	5.75
PO	7	7	7	7	7.00

As Table 6 shows, PCO earns the best average rank (1.50): it is first on both the number of selected features and computational time, and second on fitness and accuracy behind RSA-SO. RSA-SO follows closely (2.00) and HBE-CSA is third (2.50), while PO, CO, RSA, and SO trail well behind. The takeaway is that PCO does not win every individual metric, but it is the most balanced method overall, pairing near-top accuracy with the smallest feature set and the shortest runtime — the combination that matters most for practical deployment.

3.3. Convergence Analysis

Figure 4 plots the best fitness against iteration for all methods. PCO drops quickly during the first 25–30 iterations and then settles at the lowest fitness in the plot, converging to a slightly better value than RSA-SO and clearly below PO, CO, RSA, SO, and HBE-CSA. The early steep descent reflects the Cheetah phase sharpening a good region once the Parrot phase has located it, while the smooth, flat tail indicates that the mode-switching rule keeps the search from drifting after convergence. This behaviour is consistent with the low run-to-run variance reported in Tables 1 and 2.

4. Limitations of the Study

Several limitations should be noted when interpreting these results. First, the evaluation uses a single benchmark ECG dataset, so the reported gains may not transfer unchanged to other populations, acquisition devices, or noise conditions; cross-dataset validation has not yet been carried out. Second, the present tables report accuracy at the aggregate level, whereas per-class sensitivity, specificity, and F1-score — which are important for imbalanced arrhythmia classes — are not yet tabulated, and a confusion matrix is not provided. Third, the improvements of PCO over the closest competitors (RSA-SO and HBE-CSA) are numerically small on fitness and accuracy, and no formal significance test (for example, a Wilcoxon signed-rank test) has been applied to confirm that these differences are statistically meaningful.

Fourth, although PCO has the lowest mean runtime, its runtime standard deviation (31.38 s in Table 4) is higher than that of HBE-CSA, indicating that the switching mechanism can make individual runs less predictable in time. Fifth, PCO introduces control parameters — the stagnation threshold M and the mutation probability — whose sensitivity has not been fully studied, and the choice of the downstream supervised classifier has not been varied.

Finally, the suitability for real-time and embedded use is argued from feature count and runtime rather than demonstrated on target hardware.

5. Future Scope

These limitations point to clear next steps. The framework should be validated on multiple and multi-lead datasets, such as PTB-XL and the INCART database, with explicit cross-dataset testing to establish generalization. Full per-class reporting — sensitivity, specificity, F1-score, and confusion matrices — together with statistical significance tests should accompany the accuracy figures so that the gains over RSA-SO and HBE-CSA can be assessed rigorously.

On the method side, PCO can be combined with deep feature extractors (for example, CNN or LSTM embeddings) so that learned and hand-crafted features are jointly selected, and it can be paired with different classifiers (SVM, random forest, or gradient boosting) to test robustness of the selected subset. Self-adaptive control of the stagnation threshold and mutation rate, and a multi-objective formulation that balances accuracy, feature count, and energy, are promising extensions. Finally, deploying the optimized model on edge or wearable hardware and measuring latency and power consumption would confirm its practicality for continuous, real-time arrhythmia monitoring.

6. Conclusion

This paper introduced PCO, a hybrid optimizer that pairs the Parrot Optimizer's broad search with the Cheetah Optimizer's fast local search for feature selection in ECG-based arrhythmia detection. After preprocessing the signals and extracting multi-domain features, PCO chose a compact subset for classification. Across the experiments it gave better fitness, more stable results, and stronger accuracy than the optimization baselines, and against recent deep learning methods it held competitive accuracy while using far fewer features and less time. In other words, the hybrid search is an effective and efficient way to select features for automated arrhythmia detection, which suits it to intelligent healthcare monitoring. Future work will test PCO on larger multi-lead datasets, pair it with deep feature extractors, and validate it on real-time embedded hardware.

References

1. S. Khurshid, S. H. Choi, L.-C. Weng, E. Y. Wang, L. Trinquart, E. J. Benjamin, P. T. Ellinor, and S. A. Lubitz, "Frequency of cardiac rhythm abnormalities in a half million adults," Circulation: Arrhythmia and Electrophysiology, vol. 11, no. 7, p. e006273, 2018.
2. Y.-Y. Jo, J.-M. Kwon, K.-H. Jeon, Y.-H. Cho, J.-H. Shin, Y.-J. Lee, M.-S. Jung, J.-H. Ban, K.-H. Kim, S. Y. Lee, et al., "Detection and classification of arrhythmia using an explainable deep learning model," Journal of Electrocardiology, vol. 67, pp. 124–132, 2021.
3. B. Breve, L. Caruccio, S. Cirillo, V. Deufemia, and G. Polese, "Visual ECG Analysis in Real-world Scenarios," DMSVIVA, pp. 46–54, 2021.
4. A. Mustaqeem, S. M. Anwar, A. R. Khan, and M. Majid, "A statistical analysis based recommender model for heart disease patients," International Journal of Medical Informatics, vol. 108, pp. 134–145, 2017.
5. J. Gao, H. Zhang, P. Lu, and Z. Wang, "An effective LSTM recurrent network to detect arrhythmia on imbalanced ECG dataset," Journal of Healthcare Engineering, vol. 2019, no. 1, p. 6320651, 2019.
6. L. Niu, C. Chen, H. Liu, S. Zhou, and M. Shu, "A deep-learning approach to ECG classification based on adversarial domain adaptation," in Healthcare, vol. 8, p. 437, MDPI, 2020.
7. N. Guynn, K. Regis, A. Eckert, M. D'Souza, S. Turbow, S. Westerman, L. Sperling, and M. Lloyd, "Sustained Monomorphic Ventricular Tachycardia Accurately Detected by Wearable Technology," Case Reports, vol. 30, no. 9, p. 103273, 2025.
8. K. Gil and A. Vejar, "Privacy-preserving Framework for Automated Detection of Arrhythmia in ECG Data," Journal of Telecommunications and Information Technology, Special Issue, pp. 25–30, 2025.
9. K. Febeena and C. Kurian, "MAB-Net: an enhanced MobileNet with attention and BiLSTM model for arrhythmia detection using electrocardiogram signals," Iran Journal of Computer Science, pp. 1–16, 2025.
10. Y. P. Sai and L. Kumari, "A novel inference system for detecting cardiac arrhythmia using deep learning framework," Neural Computing and Applications, pp. 1–17, 2025.
11. D. Kim, K. R. Lee, D. S. Lim, K. H. Lee, J. S. Lee, D.-Y. Kim, and C.-B. Sohn, "A novel hybrid CNN-transformer model for arrhythmia detection without R-peak identification using stockwell transform," Scientific Reports, vol. 15, no. 1, p. 7817, 2025.
12. D. Kwon, H. Kang, D. Lee, and Y.-C. Kim, "Deep learning-based prediction of atrial fibrillation from polar transformed time-frequency electrocardiogram," PLOS One, vol. 20, no. 3, p. e0317630, 2025.
13. C.-L. Liu, B. Xiao, and C.-H. Hsieh, "Multimodal fusion of spatial-temporal and frequency representations for enhanced ECG classification," Information Fusion, p. 102999, 2025.

14. H. Wang, Y. Zhou, X. Niu, D. Liu, L. Li, Y. Duan, and Z. Wang, "A Low-Cost and stable DAL arrhythmia detection algorithm," Expert Systems with Applications, vol. 261, p. 125482, 2025.
15. R. d. T. Guerra, C. K. Yamaguchi, S. F. Stefenon, L. d. S. Coelho, and V. C. Mariani, "Deep Learning Approach for Automatic Heartbeat Classification," Sensors, vol. 25, no. 5, p. 1400, 2025.
16. X. Wang, Z. He, Z. Lin, Y. Han, W. Su, B. Tan, and S. Xie, "PRA-Net: Polymorphic Representation Aware Network for Accurate Cardiac Arrhythmia Detection," IEEE Transactions on Instrumentation and Measurement, 2025.
17. A. Apostol and M. Nutu, "Arrhythmia Classification from 12-Lead ECG Signals Using Convolutional and Transformer-Based Deep Learning Models," arXiv preprint arXiv:2502.17887, 2025.
18. H. Ren et al., "Heterogeneous feature fusion based machine learning strategy for ECG diagnosis," Expert Systems with Applications, 2025.
19. H. Welle, C. Nagel, A. Loewe, R. Mikut, and O. Dössel, "Classification of Bundle Branch Blocks with QRS Templates Extracted from 12-lead ECGs," Current Directions in Biomedical Engineering, vol. 7, no. 2, pp. 582–585, 2021.
20. H. Yang, Y. Zhao, and X. Yang, "PearsonUnet: ECG Classification Model Integrating Pearson Related Beats," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1–11, 2025.