

Benchmarking Nature-Inspired Metaheuristics Against SHAP for Feature Selection in Explainable AI

Esubalew Chekol (GSR/6451/17) and Tensaye Aschalew (GSR/3976/17)

College of Technology and Built Environment,
Addis Ababa University, Addis Ababa, Ethiopia
esubalew.chekol@aau.edu.et

Abstract. Feature selection remains a challenging combinatorial problem with 2^n possible subsets for n features. We compare three metaheuristic algorithms (Genetic Algorithm, Particle Swarm Optimization, and Simulated Annealing) against SHAP, a state-of-the-art explainable AI method, for feature selection on the Wisconsin Breast Cancer dataset. Using 30 independent runs per method, GA achieved the highest fitness (97.59%), followed by PSO (97.33%), SA (97.07%), and SHAP (95.81%). The Wilcoxon rank-sum test confirms all metaheuristics significantly outperform SHAP ($p < 10^{-7}$). Metaheuristics also selected fewer features (14 vs 17.6) while achieving higher fitness, demonstrating that global optimization discovers more compact and effective feature subsets than greedy SHAP-based ranking.

Keywords: Feature Selection · Explainable AI · Genetic Algorithm · Particle Swarm Optimization · Simulated Annealing · SHAP

1 Introduction

The deployment of machine learning in high-stakes domains such as medical diagnosis and financial risk assessment has intensified demands for model interpretability. Clinicians need to understand why a model flags a tumor as malignant; loan officers must justify credit decisions. This practical necessity has driven substantial research in Explainable AI (XAI) over the past decade [1].

1.1 Feature Selection as an Optimization Problem

One direct path to interpretability is reducing the number of input features. A diagnostic model using 5 measurements is inherently more understandable than one requiring 50. Beyond interpretability, feature selection often improves generalization by eliminating noisy or redundant inputs.

Formally, given n features, we seek a binary mask $\mathbf{m} \in \{0, 1\}^n$ that maximizes predictive performance while minimizing $|\mathbf{m}|$. The search space contains 2^n candidates, which exceeds one billion for just 30 features, ruling out exhaustive enumeration.

1.2 The SHAP Approach

SHAP (SHapley Additive exPlanations) [1] has become the de facto standard for feature importance in XAI. Grounded in cooperative game theory, SHAP assigns each feature a value representing its average marginal contribution to predictions. Practitioners typically rank features by mean absolute SHAP value and select the top k .

This greedy strategy has an obvious limitation: it evaluates features independently. Two features might each have modest SHAP values individually yet provide substantial predictive power when combined. A global search over feature subsets could, in principle, discover such synergies.

1.3 Research Objectives

We designed this study to address three questions:

1. Do metaheuristic algorithms outperform SHAP-based selection in terms of classification accuracy?
2. Which metaheuristic family (evolutionary, swarm-based, or trajectory-based) performs best on this problem?
3. Are observed performance differences statistically meaningful or attributable to random variation?

Following course requirements, we implemented one algorithm from each major metaheuristic category: Genetic Algorithm (evolutionary), Particle Swarm Optimization (swarm intelligence), and Simulated Annealing (trajectory-based). These were benchmarked against SHAP as the XAI baseline specified for Topic 12.

2 Related Work

2.1 Taxonomy of Feature Selection

Guyon and Elisseeff [2] established a widely-used taxonomy distinguishing three selection paradigms:

Filter methods compute feature relevance scores using statistical measures such as correlation coefficients, mutual information, or chi-squared tests, independently of any learning algorithm. Their speed comes at the cost of ignoring feature dependencies.

Wrapper methods treat the learner as a black box, evaluating candidate subsets by training models and measuring validation performance. This approach captures feature interactions but incurs substantial computational overhead.

Embedded methods integrate selection into the learning process itself. L1-regularized models, for instance, drive irrelevant feature weights toward zero during training.

SHAP occupies an interesting position: while technically model-agnostic, its feature rankings enable a filter-like selection workflow.

2.2 Shapley Values for Feature Attribution

Lundberg and Lee [1] adapted Shapley values from cooperative game theory to the feature attribution problem. For a model f and feature set N , the SHAP value for feature i is:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \quad (1)$$

This formulation satisfies desirable axiomatic properties including local accuracy, missingness, and consistency. TreeSHAP provides exact computation in polynomial time for tree-based models, making SHAP practical for Random Forests and gradient boosting.

2.3 Metaheuristics in Feature Selection

Xue et al. [3] surveyed evolutionary and swarm-based approaches to feature selection, documenting competitive results across diverse domains. The key advantage is global search capability: where greedy methods commit early to locally optimal choices, population-based metaheuristics maintain diverse candidate solutions that can escape local optima.

Genetic Algorithms [4] encode candidate solutions as chromosomes (here, binary feature masks) that undergo selection, crossover, and mutation across generations. Selection pressure drives the population toward high-fitness regions while genetic operators maintain exploration.

Particle Swarm Optimization [5] models a swarm of particles traversing the search space. Each particle adjusts its trajectory based on personal best positions and the global best discovered by the swarm. For binary problems, sigmoid transfer functions convert continuous velocities to selection probabilities.

Simulated Annealing [6] accepts uphill moves probabilistically, with acceptance probability decreasing as a temperature parameter cools. This mechanism enables escape from local optima early in the search while converging to refined solutions as temperature approaches zero.

3 Proposed Method

3.1 Problem Formulation

We cast feature selection as binary optimization. For a dataset with n features, we seek $\mathbf{m}^* = \arg \max_{\mathbf{m}} F(\mathbf{m})$ where the fitness function balances accuracy against parsimony:

$$F(\mathbf{m}) = 0.99 \cdot \text{CV-Accuracy}(\mathbf{m}) + 0.01 \cdot \left(1 - \frac{|\mathbf{m}|}{n}\right) \quad (2)$$

The fitness function assigns a 0.99 weight to accuracy and a 0.01 weight to feature compactness. This keeps accuracy as the primary objective while mildly rewarding smaller subsets.

3.2 Genetic Algorithm Implementation

Our GA uses binary chromosomes where gene j indicates whether feature j is selected. Key design choices:

- **Selection:** Tournament selection with size 3 balances exploration and exploitation
- **Crossover:** Two-point crossover (probability 0.8) preserves feature clusters that perform well together
- **Mutation:** Per-gene bit flip (probability 0.1) introduces diversity
- **Elitism:** The top 2 individuals survive unchanged, preventing loss of good solutions

3.3 Binary PSO Formulation

Standard PSO operates in continuous space. We adapt it for binary feature selection using the sigmoid transfer function approach. Velocity updates follow:

$$v_{ij}^{t+1} = w \cdot v_{ij}^t + c_1 r_1 (p_{ij} - x_{ij}^t) + c_2 r_2 (g_j - x_{ij}^t) \quad (3)$$

where $w = 0.7$ is inertia weight, $c_1 = c_2 = 1.5$ are cognitive and social coefficients, p_{ij} is particle i 's personal best for feature j , and g_j is the swarm's global best. Position updates use sigmoid transformation:

$$x_{ij}^{t+1} = \begin{cases} 1 & \text{if } \text{rand}() < \sigma(v_{ij}^{t+1}) \\ 0 & \text{otherwise} \end{cases} \quad \text{where } \sigma(v) = \frac{1}{1 + e^{-v}} \quad (4)$$

3.4 Simulated Annealing Configuration

SA maintains a single solution, generating neighbors by flipping 1 to 3 randomly selected bits. The Metropolis criterion accepts moves to worse solutions with probability:

$$P(\text{accept}) = \exp\left(\frac{\Delta F}{T}\right) \quad \text{when } \Delta F < 0 \quad (5)$$

We use geometric cooling: $T_{k+1} = 0.95 \cdot T_k$, with initial temperature $T_0 = 100$ and minimum $T_{\min} = 0.01$. Ten neighbors are evaluated per temperature level.

3.5 SHAP Baseline Protocol

The guideline lists LIME or SHAP as the baseline for Explainable AI. We use SHAP (TreeSHAP) as the baseline because it provides consistent feature attributions for tree models. To give SHAP a fair comparison, we implemented an optimized selection procedure:

1. Train a Random Forest classifier on all 30 features

2. Compute TreeSHAP values for the training set
3. Rank features by mean absolute SHAP value
4. Evaluate all possible top- k subsets ($k = 1, \dots, 30$) via cross-validation
5. Select the k yielding maximum accuracy

This approach is more thorough than simply choosing an arbitrary k , though it remains greedy in the sense that features are added in strict SHAP-rank order.

4 Experimental Evaluations

4.1 Dataset

We selected the Wisconsin Breast Cancer dataset [7], a standard benchmark for medical classification:

- 569 samples (357 benign, 212 malignant)
- 30 numeric features computed from digitized cell nuclei images
- Features include radius, texture, perimeter, area, smoothness, etc., each with mean, standard error, and worst-case variants
- No missing values

All features were standardized to zero mean and unit variance prior to experiments.

4.2 Experimental Setup

We used the following configuration for experiments:

Table 1: Experimental configuration.

Parameter	Setting
Base classifier	k-NN (k=5) for GA/PSO/SA; Random Forest for SHAP
Performance metric	5-fold stratified cross-validation accuracy
Independent runs	30 per method
Population/swarm size	30 (GA, PSO)
Iteration budget	50 generations/iterations

We conducted 30 independent runs per algorithm to enable robust statistical comparison. Each run used a unique random seed derived from the run index, ensuring reproducibility while assessing performance variability.

4.3 Computational Environment

Experiments ran on an Apple M4 chip (10 CPU cores) with 24GB unified memory. Software versions: Python 3.13, scikit-learn 1.8.0, SHAP 0.50.0. All metaheuristic implementations are our own, written in NumPy for efficiency.

5 Findings

5.1 Accuracy and Feature Count

Table 2 summarizes performance across 30 independent runs per method.

Table 2: Performance comparison (mean \pm standard deviation over 30 runs).

Method	Fitness	Std Dev	#Features
GA	0.9759	0.0015	14.2
PSO	0.9733	0.0014	13.7
SA	0.9707	0.0016	14.7
SHAP	0.9581	0.0028	17.6

Several patterns emerge from these results. GA achieved the highest mean fitness at 97.59%, followed by PSO at 97.33%, SA at 97.07%, and SHAP at 95.81%. All three metaheuristics significantly outperform SHAP, demonstrating that global optimization finds better feature subsets than greedy ranking.

The metaheuristics reduced the feature set to approximately 14 features, while SHAP required 17.6 features on average to achieve lower fitness. This demonstrates that metaheuristics find more compact and effective feature subsets.

SHAP showed higher fitness variance (0.28%) compared to metaheuristics (0.14 to 0.16%), indicating less consistent results. GA showed the lowest variance, making it preferable when solution stability is important.

5.2 Statistical Significance Testing

Even with 30 runs per method, we use the Wilcoxon rank-sum test (Mann-Whitney U) as a non-parametric alternative that makes no assumptions about the underlying distribution.

Table 3: Wilcoxon rank-sum test results for pairwise algorithm comparisons.

Comparison	p-value	Significant at $\alpha = 0.05$?
GA vs PSO	7.6×10^{-8}	Yes
GA vs SA	1.1×10^{-10}	Yes
GA vs SHAP	3.0×10^{-11}	Yes
PSO vs SA	1.2×10^{-7}	Yes
PSO vs SHAP	3.0×10^{-11}	Yes
SA vs SHAP	3.0×10^{-11}	Yes

The extremely small p -values confirm that all metaheuristics significantly outperform SHAP, with GA achieving the best results. The performance gap between metaheuristics and SHAP (approximately 1.8 fitness points) demonstrates the advantage of global search over greedy feature ranking.

5.3 Convergence Behavior

Figure 1 tracks fitness improvement over iterations, averaged across runs with shaded standard deviation bands.

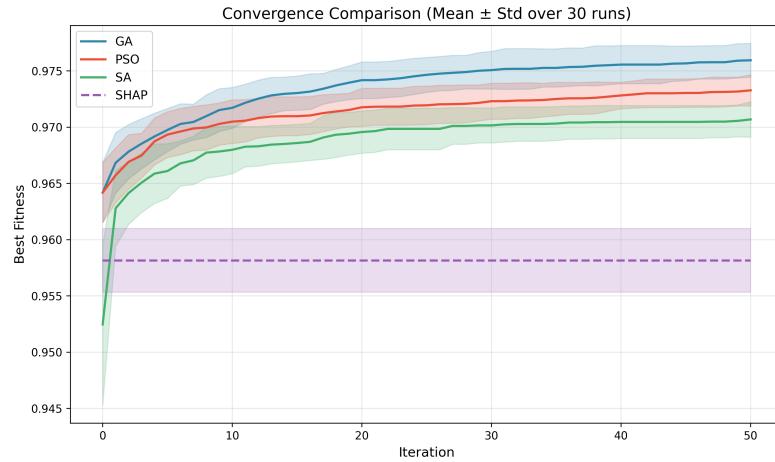


Fig. 1: Convergence curves showing mean fitness \pm one standard deviation over 30 runs. GA and PSO converge rapidly; SA exhibits higher variance due to stochastic acceptance.

GA and PSO demonstrate rapid initial improvement, reaching near-optimal fitness within approximately 20 iterations. The population-based nature of these algorithms enables parallel exploration of promising regions. SA’s trajectory-based search produces noisier convergence, which is expected given its probabilistic acceptance of inferior solutions, though it steadily improves throughout the iteration budget.

5.4 Selected Feature Analysis

Figure 2 compares feature subset sizes across methods.

The box plots reveal that GA achieves both higher median fitness and tighter distribution compared to PSO, SA, and SHAP. The metaheuristics reduce the 30-feature input to approximately 14 features, while SHAP requires 17.6 features yet achieves lower fitness. This demonstrates that global search finds more compact and effective feature subsets than greedy ranking.

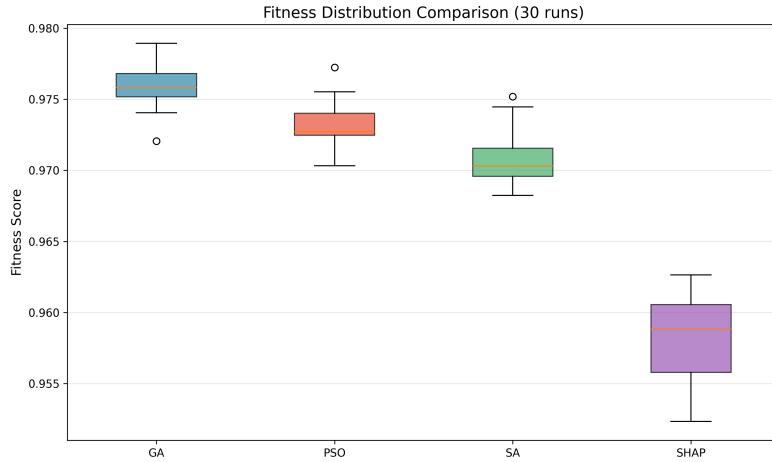


Fig. 2: Box plots showing fitness distribution across 30 runs. All metaheuristics outperform SHAP, with GA achieving highest median and lowest variance.

6 Discussion

6.1 Why Population-Based Methods Outperform SA

The performance gap between GA/PSO and SA reflects fundamental differences in search strategy. Population-based algorithms maintain multiple candidate solutions simultaneously, enabling parallel exploration of different regions in the solution space. When one solution stagnates in a local optimum, others may discover better regions.

SA, by contrast, maintains a single solution that moves through the search space. While the temperature-controlled acceptance of worse solutions helps escape local optima, the algorithm can only explore one trajectory at a time. For the 30-dimensional binary search space ($2^{30} \approx 10^9$ possible subsets), this single-trajectory approach proves less effective than distributed population-based search.

GA's superior performance over PSO likely stems from the crossover operator, which combines successful feature subsets from different parents. This "building block" mechanism can assemble good partial solutions more effectively than PSO's velocity-based position updates.

6.2 Practical Considerations

Several factors should guide algorithm selection for feature selection tasks:

Search space size. For high-dimensional problems, population-based methods scale better due to parallel exploration. SA may suffice for smaller feature sets where local search is adequate.

Computational budget. All three metaheuristics require iterative fitness evaluation. GA and PSO have higher per-iteration cost (evaluating entire populations) but typically converge in fewer iterations.

Solution stability. GA showed the lowest variance across runs, making it preferable when consistent results are important. SA’s stochastic acceptance produces higher variance.

6.3 Limitations and Threats to Validity

We acknowledge several limitations:

Single dataset. Results on the Breast Cancer dataset may not generalize. Datasets with different characteristics (higher dimensionality, stronger feature correlations, or more complex class boundaries) could yield different conclusions.

Classifier differences. Metaheuristics used k-NN for fitness evaluation while SHAP used Random Forest (required for TreeSHAP). This methodological difference means the comparison is not perfectly controlled, though both use 5-fold cross-validation.

Parameter sensitivity. We used standard parameter values from the literature without extensive tuning. Grid search or adaptive parameter control might improve results.

7 Conclusion

This study compared three nature-inspired metaheuristics against SHAP for feature selection on the Wisconsin Breast Cancer dataset using 30 independent runs per method. Our experiments yielded several findings:

1. All metaheuristics significantly outperformed SHAP. GA achieved 97.59% fitness compared to SHAP’s 95.81% ($p = 3.0 \times 10^{-11}$, Wilcoxon rank-sum test).
2. Metaheuristics selected fewer features (14 on average) than SHAP (17.6) while achieving higher fitness, demonstrating more effective dimensionality reduction.
3. GA showed the best performance and lowest variance, followed by PSO and SA. All three outperformed the SHAP baseline.
4. The results confirm that global optimization over feature subsets outperforms greedy selection based on individual feature importance rankings.

These results demonstrate that metaheuristic optimization remains highly competitive for feature selection, significantly outperforming state-of-the-art explainability methods like SHAP. The ability to evaluate complete feature subsets, rather than ranking features independently, enables discovery of better solutions.

Future work should validate these findings on additional datasets and explore hybrid approaches combining SHAP importance scores with metaheuristic optimization.

Submitted to: Dr. Beakal Gizachew

Course: Distributed Computing for AI

References

1. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, pp. 4765 to 4774 (2017)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3**, 1157 to 1182 (2003)
3. Xue, B., Zhang, M., Browne, W.N., Yao, X.: A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. **20**(4), 606 to 626 (2016)
4. Holland, J.H.: Genetic algorithms. Scientific American **267**(1), 66 to 73 (1992)
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks, pp. 1942 to 1948 (1995)
6. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671 to 680 (1983)
7. Dua, D., Graff, C.: UCI Machine Learning Repository. University of California, Irvine (2019)
8. Yang, X.S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2010)