# Evolutionary Multi-Objective Feature Selection in High-Dimensional Biomedical Data

**Xiomara Clarisa Sarrea Vega**
Universidad Carlos III de Madrid

## Abstract

Feature selection in high-dimensional datasets is a combinatorial optimization challenge, particularly with small sample sizes. This study approaches it as a Multi-Objective Optimization Problem (MOOP), focusing on minimizing prediction error while simplifying the model. The NSGA-II algorithm was implemented to explore the huge search space, resulting in a robust Pareto Front and achieving a sparsity of 0.6% (4 features). The results indicate an enhanced efficacy over traditional statistical filtering techniques like Selectkbest, highlighting the benefits of synergistic feature combinations for improved outcomes.

## 1 INTRODUCTION

Optimization problems often involve more than one objective. In the context of biomarker discovery, we should prioritize the accuracy of the model. Selected features should clearly distinguish between patients who improve and those who don't. In addition, for simplicity and to make this reproducible, we want to find the smallest possible set of features. A simpler model with fewer variables is preferred as it reduces the risk of overfitting and enhances clinical interpretability.

Given a feature set of $D$ features, the search space consists of $2^D$ possible combinations, making methods such as brute force impossible to execute for a high number of features.

Traditional approaches like SelectKBest, use a univariate feature selection algorithm that ranks the best $k$ features using a statistical scoring system. Then it examines each one individually and selects the ones with the greatest scores. However, it misses the potential interactions between features, as it assumes feature independence (Guyon and Elisseeff, 2003). This weakness is the main motivation for employing a Multi-Objective Evolutionary Algorithm (MOEA). We used the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to search through all possible combinations (Deb et al., 2002). Through this search we aim to find the Pareto Optimal Front, which balances between high accuracy and simplicity.

## 2 PROBLEM FORMULATION

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be the dataset, where $\mathbf{x}_i \in \mathbf{R}^D$ is the feature vector ($D = 642$) and $y_i \in \{0, 1\}$ is the binary response label.

We define the decision vector as a binary mask $\mathbf{m} \in \{0, 1\}^D$, where $m_j = 1$ implies the selection of feature $j$. The goal is to find the optimal mask $\mathbf{m}^*$ that minimizes the vector objective function $\mathbf{F}(\mathbf{m})$:

$$\min_{\mathbf{m}} \mathbf{F}(\mathbf{m}) = \begin{bmatrix} f_1(\mathbf{m}) \\ f_2(\mathbf{m}) \end{bmatrix} \tag{1}$$

### 2.1 Objective 1: Predictive Error

As these kinds of model are not designed to maximize scores, our objective is translated to minimize the predictive error defined as $f_1$. It takes the features, uses the Hadamard product to mask out the ones we don't want, and estimates the generalization error via K-Fold Cross-Validation ($K = 3$) using a simple model (Logistic Regression). Then, it takes the average success score, and subtracts it from 1.

$$f_1(\mathbf{m}) = 1 - \frac{1}{K} \sum_{k=1}^{K} \text{AUC}_k(\mathcal{L}, \mathbf{X}_{test}^{(k)} \circ \mathbf{m}) \tag{2}$$

where $\circ$ denotes the Hadamard product (masking).

### 2.2 Objective 2: Complexity (Sparsity)

The second objective $f_2$ pulls the solution towards simplicity, to avoid a high number of features that would

only add more noise and increase the risk of overfitting. It implements a $L_0$ norm of the decision vector, which only counts the non-zero features. Since NSGA-II is a gradient-free method, the non-differentiability of the $L_0$ norm poses no optimization challenge.

$$f_2(\mathbf{m}) = \frac{||\mathbf{m}||_0}{D} = \frac{1}{D}\sum_{j=1}^{D} m_j \tag{3}$$

### 2.3 Constraints

The feasible region $\Omega$ is constrained to non-empty subsets, at least one feature must be used:

$$\text{s.t.} \sum_{j=1}^{D} m_j \geq 1 \tag{4}$$

## 3 METHODOLOGY

### 3.1 Dataset and Preprocessing

The study uses a high-dimensional EEG dataset consisting of $N = 24$ subjects (Respondents vs. Non-Respondents) and $D = 642$ spectral features derived from 32 electrodes. To ensure numerical stability for the linear estimators, all features were normalized using Min-Max Scaling ($x \in [0, 1]$). Missing values were imputed using the mean strategy prior to optimization.

### 3.2 NSGA-II Algorithm

Unlike other genetic algorithms that force us to guess weights for different objectives, **NSGA-II** optimizes both objectives separately. It was implemented using the `pymoo` library (Blank and Deb, 2020), and it is mainly based on two fundamental mechanisms:

1. **Pareto Ranking:** Solutions are arranged into hierarchies known as fronts rather than a single score. If there is not other solution that is superior for both goals, it is ranked first (Front 1). By doing this, the best trade-offs are maintained instead of just the highest average.

2. **Crowding Distance:** It keeps the algorithm from focusing on a single kind of solution. Favouring the solution found in a less crowded area of the search space when two solutions are equally excellent. Forcing the search to go far and broad, generating a wide range of choices from complex but highly accurate to sparse but less accurate.

### 3.3 Genetic Operators

We customized the evolutionary operators to suit the structure of EEG data:

- **Encoding:** The solution is represented as a sequence of zeros and ones of length $D = 642$, where 1 means the feature is active and 0 means that is ignored.
- **Crossover:** We use Two-Point Crossover. As in our dataset features are organized by brain region, standard random shuffle would break these groups apart. Two-Point Crossover solves this by swapping whole blocks of neighbouring features at once. This keeps related signals from the same brain area together, ensuring the algorithm passes down complete regional patterns rather than mixed ones.
- **Mutation:** Bitflip Mutation was applied with a probability of $p_m = 0.05$. This operator introduces stochastic diversity by independently flipping each feature's state (from 0 to 1 or vice versa) with a 5% chance. This mechanism prevents the population from getting stuck in local optima, ensuring the algorithm explores diverse regions of the high-dimensional solution space.

### 3.4 Computational Complexity

The exhaustive search space for a binary vector of size $D = 642$ is $|\Omega| = 2^{642}$, rendering deterministic search ($O(2^D)$) intractable.

In contrast, the computational complexity of one generation of NSGA-II is dominated by the non-dominated sorting procedure:

$$C_{gen} \approx O(M \cdot P^2) \tag{5}$$

where $M = 2$ objectives and $P = 50$ is the population size we chose. This allows for convergence to a near-optimal Pareto front within feasible runtime, reducing the search magnitude to approximately $10^4$ evaluations.

### 3.5 Convergence Criteria

In real-world problems, the true global optimum is uncertain, therefore we need a reliable indicator to determine when to end the search. We employ the **Hypervolume (HV)** indicator, which calculates the total area covered by our Pareto Front. The larger, the closer that the model is to minimize both objectives.

We monitor variations in Hypervolume at each generation. The optimization is finished if the Hypervolume fails to increase by more than a threshold

($\Delta HV < 10^{-3}$) over a moving window of 30 consecutive generations (early stopping). Suggesting that the population has achieved the optimal trade-off curve and that further computations are unlikely to provide significant gains. In addition, the maximum number of generations was set to 1000.

# 4 RESULTS AND ANALYSIS

## 4.1 Convergence Analysis

The algorithm ran for 444 generations before satisfying the termination criteria. Figure 1 illustrates the Hypervolume trajectory. Showing that the population successfully converged to the Pareto Optimal Front, validating the efficiency of the search.

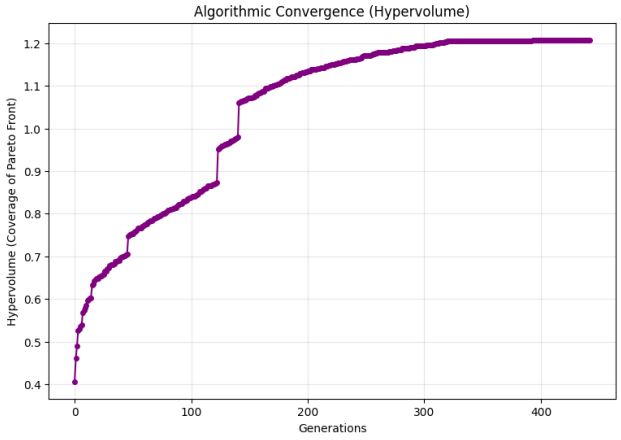


Figure 1: Algorithmic Convergence (Hypervolume metric). The plateau indicates mathematical convergence to the optimal front.

## 4.2 Selection of the optimal Pareto Solution

To select the optimal biomarker set, we analyzed the trade-off between training performance ($f_1$) and model complexity ($f_2$). Figure 2 visualizes the final non-dominated front found by NSGA-II.

The Knee Point (Red Dot) represents the solution where a marginal gain in accuracy would require a disproportionate increase in complexity. Table 1 details the performance of three solutions from this front.

Table 1: Performance of best Pareto Solutions.

| Sol | Features | $AUC_{train}$ | $AUC_{test}$ | $F1_{test}$ |
|---|---|---|---|---|
| 1 | 4 | 1.00 | 1.00 | 0.75 |
| 2 | 1 | 0.73 | 0.87 | 0.57 |
| 3 | 4 | 1.00 | 0.67 | 0.57 |

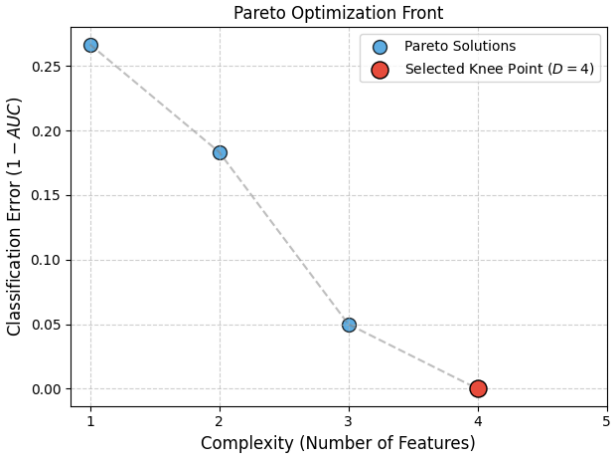


Figure 2: The Pareto Optimal Front. The "Knee Point" (Red) represents the optimal trade-off, achieving zero training error with minimal complexity ($D = 4$). Solutions to the left have higher error, while solutions to the right add complexity without performance gain.

Solution 1 ($D = 4$) achieves a perfect generalization. Solution 2 ($D = 1$) with only one feature achieves a good generalization, but a lower accuracy. Lastly, solution 3 ($D = 4$) probably reflects overfitting of the training set. For the next procedures, we will take solution 1 as the Pareto Optimal Front.

## 4.3 Benchmarking comparison

We compared the NSGA-II solutions against a univariate baseline (`SelectKBest` F-value) using the scikit-learn library (Pedregosa et al., 2011). As shown in Figure 3, only one Pareto Front solution is superior to the baseline curve (Blue Line). For a fixed complexity of 4 features, solution 1 of NSGA-II achieves a significantly higher AUC, demonstrating that combinatorial optimization captures synergistic information that univariate ranking misses, improving generalization. However, the other solutions are equal to or below the AUC values of the baseline.

## 4.4 Robustness of the Optimal Solution

To verify that solution 1 is biologically robust and not merely an artifact of the specific proxy used during optimization, we validated the selected feature vector $\mathbf{x}^*$ on three external black box classifiers that were not part of the optimization loop.

Results in Table 2 demonstrate a consistent performance. Despite being developed using a simple linear proxy, the features perform well on non-linear SVMs (Cortes and Vapnik, 1995) and on the sophis-

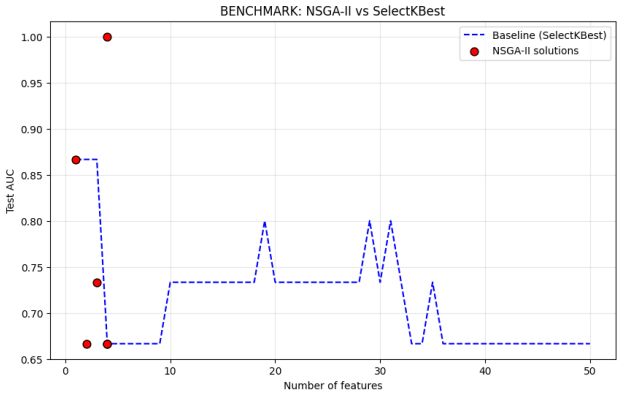


Figure 3: Performance Benchmark. NSGA-II solutions (Red) and univariate filter baseline (Blue).

ticated XGBoost (Chen and Guestrin, 2016) ensemble (AUC=0.97). This demonstrates that NSGA-II did not overfit to a particular method, but rather separated a linearly separable space that reflects a genuine underlying physiological process.

Table 2: Validation of the Optimal Subset ($D = 4$) on External Classifiers.

| **Model Architecture** | $AUC_{test}$ | $F1_{test}$ |
|---|---|---|
| Proxy (LogReg) | 1.00 | 0.75 |
| Kernel (SVM RBF) | 0.87 | 0.67 |
| Bagging (Random Forest) | 0.87 | 0.75 |
| Boosting (XGBoost) | 0.97 | 0.75 |

### 4.5 Feature Interpretation

SHAP (Lundberg and Lee, 2017) analysis (Figure 4) allows to have a close look to the most influential features in the Logistic Regression model, separating subjects who responded to the tDCS treatment (positive SHAP values) from those who did not (negative SHAP values).

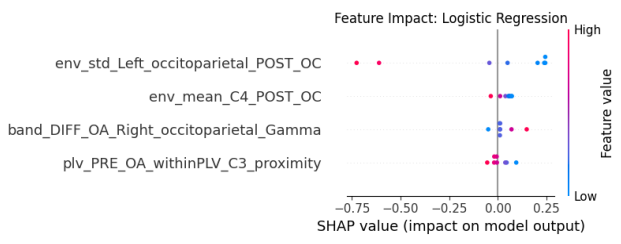


Figure 4: SHAP values for the selected biomarkers, confirming clinical interpretability.

The analyses of the POST-stimulation envelope in the occipito-parietal and sensorimotor areas indicate that a lower variability in the left occipito-parietal cluster and a lower amplitude at C4 correlate with a higher response probability to stimulation. This suggests a stabilization of oscillatory activity in responders, while increased values suggest hyper-variability associated with non-responders. Changes in spectral markers reveal that the reduction gamma power in the right occipito-parietal region is related to a responder classification. Additionally, higher baseline phase-locking value (PLV) in the C3 sensorimotor cluster improves responder predictions, emphasizing the role of network stability and baseline synchrony in tDCS response outcomes.

## 5 Limitations

While the evolutionary search successfully identified a robust feature subset, the study is limited by the small sample size ($N = 24$). In such HDLSS regimes, the risk of selection bias is inherent. Although we mitigated this using stratified cross-validation and external validation on black-box models, large-scale external replication would be required to confirm the clinical utility of the identified biomarkers.

## 6 CONCLUSION

This work investigates feature selection as a Multi-Objective Combinatorial Optimization challenge in High-Dimension, Small-Sample (HDLSS) conditions. It highlights the limits of common greedy algorithms like SelectKBest in capturing crucial feature interactions and the computational risk of brute-force approaches. The study successfully found a Pareto Optimal Front of trade-off solutions, resulted in a dimensionality reduction from 642 to 4 features, a 99.4% drop.

The results support the utility of a low-variance linear proxy (Logistic Regression), which identified a feature subset capable of generalizing to complex non-linear models like XGBoost (AUC 0.97). This suggests that the selected biomarkers may reflect genuine physiological patterns rather than simple linear artifacts. However, generalization in HDLSS regimes remains a fundamental challenge. Future research should prioritize validation on larger, new patient groups to ensure reliability, with the ultimate goal of applying these mathematical insights to improve patient care.

To facilitate reproducibility and further research, the complete implementation of the NSGA-II algorithm is hosted here on GitHub.

## References

J. Blank and K. Deb (2020). pymoo: Multi-objective optimization in Python. *IEEE Access* **8**:89435–89469.

T. Chen and C. Guestrin (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

C. Cortes and V. Vapnik (1995). Support-vector networks. *Machine Learning* **20**(3):273–297.

K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2):182–197.

I. Guyon and A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**(Mar):1157–1182.

S. M. Lundberg and S. I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, 4765–4774.

F. Pedregosa et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**:2825–2830.