The object of the study is the hyperparameter configuration space of the U-Net ar-chitecture for agricultural land segmentation from Sentinel-2 satellite imagery.

The problem being solved is the excessive cost of multi-objective hyperparameter optimization, because non-dominated sorting in the non-dominated sorting genetic algorithm II (NSGA-II), with complexity O(MN2), becomes a bottleneck for deep segmenta-tion models. To address this problem, an interactive evolutionary non-dominated sorting genetic algorithm II (IENSGA-II) framework is evaluated, in which a logistic regression classifier is trained on hyperparameter vectors and Pareto ranks from initial NSGA-II generations, then used to predict ranks in subsequent generations instead of full sorting. Unlike existing surrogate-assisted approaches, this work predicts Pareto ranks without additional model evaluations. On the panoptic agricultural satellite time series (PAS-TIS) benchmark, the framework reduced execution time by 20.07%, 16.39%, and 38.80% for 5, 10, and 15 generations, and in the 10-generation setting improved validation cri-teria, reaching an area under the receiver operating characteristic curve (AUC) of 0.9072 versus 0.9004 and validation loss of 0.6057 versus 0.6212. These results were achieved because the method accelerates selection rather than replacing model evalua-tion, while AUC-based tie-breaking preserves preference for more accurate solutions among candidates with same predicted rank. Effectiveness stems from a regular rela-tionship between hyperparameters and Pareto ranks in early evolutionary data. In prac-tice, the method is used in resource-constrained multi-objective learning when initial generations provide representative data for rank prediction

Keywords: hyperparameter optimization, precision agriculture, U-Net segmenta-tion, pareto rank prediction, surrogate-assisted evolution

UDC 004.8:519.85:528.8
DOI: 10.15587/1729-4061.2026.359143

EVALUATION OF A MACHINE LEARNING-ASSISTED INTERACTIVE EVOLUTIONARY NON-DOMINATED SORTING GENETIC ALGORITHM -II FRAMEWORK FOR HYPERPARAMETER OPTIMIZATION OF U-NET IN AGRICULTURAL LAND SEGMENTATION

Artughrul Gayibov
Corresponding author
PhD Student*
E-mail: agayibov@beu.edu.az
ORCID: https://orcid.org/0009-0009-7349-0286
Vagif Gasimov
Professor, Doctor of Technical Sciences, Dean**
ORCID: https://orcid.org/0000-0003-3192-4225
Esmira Mustafayeva
PhD, Associate professor**
ORCID: https://orcid.org/0000-0002-5300-2830
Kamala Aliyeva
PhD, Associate Professor**
ORCID: https://orcid.org/0009-0007-9470-7910
Dilara Guluzada
Lecturer*
ORCID: https://orcid.org/0009-0003-0623-3577
*Department of Information Technology and Programming***
Department of Cybersecurity and Computer Engineering*
***Baku Engineering University
Hasan Aliyev str., 120, Khirdalan, Azerbaijan, AZ0101

Received 03.02.2026
Received in revised form 05.03.2026
Accepted date 17.04.2026
Published date 30.04.2026

How to Cite: *Gayibov, A., Gasimov, V., Mustafayeva, E., Aliyeva, K., Guluzada, D. (2026). Evaluation of a machine learning-assisted interactive evolutionary non-dominated sorting genetic algorithm -II framework for hyperparame-ter optimization of U-Net in agricultural land segmentation. Eastern-European Journal of Enterprise Technologies, 2 (2 (140)), 52–64. https://doi.org/10.15587/1729-4061.2026.359143*

1. Introduction

Remote sensing from away is important, for precision agriculture. This is because it helps people manage the land, give out resources and keep an eye on the crops. Accurate segmentation of agricultural land at the parcel level is essential for multiple stakeholder groups, including farmers optimizing input allocation, policy makers distributing subsidies and monitoring compliance with agricultural regulations, and environmental agencies tracking land use changes and conservation efforts. The ability to automatically delineate agricultural parcels and classify crop types from satellite data has become critical for implementing data-driven agricultural management systems at regional and national scales. One big part of this is segmenting land. This means using pictures from satellites or planes to figure out where the fields are and what kind of crops are growing in them. Precision agriculture needs this to work well. Agricultural

land segmentation is a deal. It helps with managing the land and the crops. Maps at the parcel level facilitate environmental compliance, subsidy distribution, and yield estimation. In large agricultural regions, manual delineation of field boundaries and crop identification is time-consuming, inconsistent, and economically impractical, making automated segmentation methods essential for scaling precision agriculture to continental levels. Because its encoder-decoder structure with skip connections captures both global context and fine spatial detail, deep learning, particularly U-Net, is widely used. The symmetrical architecture of U-Net enables the model to preserve spatial information while performing hierarchical feature extraction, making it especially effective for tasks requiring precise localization alongside contextual understanding. The generalizability of U-Net across diverse segmentation domains, from medical imaging to remote sensing, demonstrates its architectural robustness. However, achieving optimal performance for agricultural land segmentation from satellite imagery requires careful configuration of the network's structure and training parameters.

Hyperparameter optimization (HPO) automates decisions (learning rate, bottleneck design, encoder width) that gradient descent is unable to learn. These settings have a significant impact on U-Net's generalization and convergence. In multi-objective HPO, where objectives may clash (increase segmentation quality while decreasing validation loss), the task becomes more difficult. A thorough search is not feasible, and manual tuning is laborious and difficult to duplicate. The computational cost of HPO is particularly acute in agricultural remote sensing, where each candidate model must be trained on large-scale satellite datasets containing thousands of image patches, making brute-force search methods economically infeasible.

Evolutionary algorithms are a good fit for multi-objective HPO over complex, non-differentiable search spaces. NSGA-II is popular because it explores the Pareto front while maintaining diversity. However, it is costly because each candidate typically needs to be trained and validated, and non-dominated sorting becomes a bottleneck as population size and the number of objectives increase. As a result, the computational efficiency of NSGA-II diminishes significantly in resource-constrained settings, a critical limitation for deep learning applications.

Surrogate-assisted approaches reduce evaluation costs by predicting fitness, but they still have selection overhead and require careful monitoring of surrogate accuracy. Furthermore, surrogate approaches require additional computational resources for surrogate model training and maintenance, adding complexity to the overall optimization pipeline. An alternative is to speed up rank assignment itself: fast inference can take the place of full non-dominated sorting if a model can predict a candidate's Pareto rank directly from its hyperparameters. This approach has not been investigated for U-Net-based agricultural segmentation and has gotten little attention. The potential of machine learning-assisted rank prediction for deep learning hyperparameter optimization remains largely unexplored, particularly in the context of large-scale agricultural segmentation tasks where computational efficiency and solution quality are both critical.

In agricultural remote sensing utilizing Sentinel-2 imagery, the recurrent training of deep segmentation models renders multi-objective hyperparameter optimization particularly resource-intensive, thereby necessitating the creation of methodologies that can diminish selection overhead while maintaining solution quality. The potential of machine learning-assisted rank prediction for reducing the computational burden of evolutionary selection in deep learning hyperparameter optimization remains largely unexplored, particularly in the context of large-scale agricultural segmentation tasks where both computational efficiency and solution quality are critical requirements. Therefore, studies devoted to the evaluation of machine learning-assisted frameworks for reducing the selection overhead of multi-objective evolutionary hyperparameter optimization in U-Net-based agricultural land segmentation are of scientific relevance.

2. Literature review and problem statement

The paper [1] shows that deep learning makes it much easier to analyze remote sensing data for agricultural monitoring. It demonstrates that hierarchical feature learning in convolutional architecture facilitates the acquisition of spatial structures that exceed traditional spectral methods. But there are still problems with choosing architectural and training hyperparameters in an efficient way when there are many criteria that don't always agree. Training deep models on large satellite datasets is very expensive, which makes this problem even harder. This creates a critical practical bottleneck: practitioners cannot use exhaustive search methods to find optimal hyperparameters because the computational cost becomes prohibitive at regional and continental scales of agricultural monitoring.

The paper [2] presents results on the U-Net encoder-decoder architecture for semantic image segmentation and shows that residual connections enable precise localization combined with context understanding. It demonstrates that U-Net can generalize across various segmentation domains, including remote sensing. However, challenges remain in the systematic optimization of U-Net hyperparameters for agricultural imagery, as optimal configurations are highly task-specific and require extensive empirical investigation. This task-specificity is especially problematic for agricultural remote sensing because optimal hyperparameter configurations for one region, crop type, or sensor may perform poorly when transferred to different agricultural contexts.

This paper [3] shows the results of a study on how to optimize hyperparameters in machine learning using more than one criterion. It demonstrates that grid search and random search are statistically ineffective and lack scalability in high-dimensional spaces. It shows that optimizing hyperparameters with more than one criterion needs strategies to find Pareto-optimal trade-offs between metrics that are at odds with each other. But one big problem is that testing deep learning models is still very expensive because each candidate usually needs to be fully trained, and there is no single, scalable solution right now. In agricultural remote sensing, this challenge is magnified because candidate models must be trained on large-scale satellite datasets with thousands of image patches, making even single evaluations computationally expensive compared to standard machine learning benchmarks.

The paper [4] presents results on the NSGA-II algorithm for evolutionary multi-objective optimization and shows that undominated sorting and crowding distance generate diverse and well-distributed Pareto fronts. It is shown that NSGA-II provides a solid foundation for multi-objective hyperparameter optimization. But undominated sorting is $O(MN^2)$ com-

plicated, and when every member of the population needs to learn everything, the total cost over generations becomes a big problem. For deep segmentation models, this quadratic complexity of non-dominated sorting combines with the already-high cost of training each candidate configuration, creating a severe computational bottleneck that limits evolutionary optimization to small population sizes and few generations.

The paper [5] provides findings on genetic algorithms as a gradient-free search methodology for intricate and non-differentiable domains, demonstrating that selection and mutation may identify high-quality solutions without derivative information. It also demonstrates that evolutionary methods are resilient in noisy, multimodal fitness environments. The primary unsolved issue in deep learning regarding hyperparameter optimization (HPO) is the evaluation cost, since each fitness assessment necessitates training a neural network. However, a secondary but equally critical problem that remains largely overlooked is the computational cost of selection mechanisms themselves specifically, the non-dominated sorting step in multi-objective evolutionary algorithms becomes a significant overhead when evaluations are expensive.

In [6], the paper discusses findings on model-assisted evolutionary algorithms using Gaussian random field metamodels. It illustrates that these cost-effective approaches diminish the assessments of the fitness function while maintaining the quality of the Pareto front. It is also shown that approximation models can effectively guide evolutionary search in the presence of limited data. Conversely, model-assisted evolutionary algorithms concentrate on the approximation of the fitness function and neglect the overhead associated with selection. This is because the algorithmic cost of sorting non-dominated elements is distinct from the evaluation cost. This distinction is crucial: reducing the number of model evaluations through surrogate approximation does not address the $O(MN^2)$ complexity of non-dominated sorting, which must still occur in every generation regardless of whether fitness is evaluated or approximated.

The article [7] presents results on hyperparameter optimization methods based on the bandit algorithm, specifically Hyperband. It demonstrates that adaptive budget allocation and early stopping reduce computational waste while maintaining competitive model quality. It also shows that eliminating unpromising candidates improves efficiency compared to random or grid search. But these methods are mostly single-objective and don't natively support multi-criteria hyperparameter optimization, which means that accuracy and loss must be optimized at the same time. In agricultural segmentation, single-objective approaches are fundamentally insufficient because precision, recall, and per-class accuracy often conflict with overall validation loss, and stakeholders require explicit Pareto fronts showing all non-dominated trade-off solutions.

The paper [8] shows how Pareto archiving strategies can be used with multi-objective evolutionary algorithms. It shows that approximating dominance relations can improve evolutionary selection. It is demonstrated that machine learning models can acquire valuable approximations of the Pareto front structure from constrained datasets. Nonetheless, this methodology has yet to be utilized for deep segmentation hyperparameter optimization in agricultural remote sensing, and the direct prediction of Pareto ranks from hyperparameter vectors is still unexamined in this domain. Furthermore, the computational cost profiles of deep segmentation models differ substantially from those of the smaller models studied in previous rank prediction work, raising questions about whether rank prediction strategies generalize to high-cost, encoder-decoder architectures.

The article [9] shows the results of an IENSGA-II framework that combines a decision tree classifier with NSGA-II to optimize multiple hyperparameters of convolutional neural networks (CNNs) on the MNIST dataset. It is demonstrated that machine learning-assisted rank prediction diminishes selection overhead while providing comparable accuracy and reducing trade-offs relative to standard NSGA-II. However, there are still problems with scalability to encoder-decoder segmentation architectures like U-Net and with applying it to complex multi-class agricultural datasets, where each evaluation requires training a high-capacity model on large-scale satellite images. Critically, the per-evaluation cost in MNIST-based HPO is orders of magnitude lower than in agricultural segmentation on PASTIS, making it unclear whether the computational benefit of rank prediction justifies the one-time cost of classifier training in resource-intensive settings.

Across [1–9], four recurring limitations emerge:

a) ML-assisted Pareto rank prediction has not been validated for HPO of deep segmentation models on large-scale agricultural remote sensing benchmarks;

b) surrogate methods approximate fitness but do not target NSGA-II rank assignment;

c) the IENSGA-II concept in [9] has not been extended to U-Net or multi-class satellite segmentation;

d) the computational benefit of selection acceleration under high per-evaluation cost such as U-Net training on PASTIS, remains unquantified. A way to overcome these difficulties is to apply and rigorously evaluate IENSGA-II for U-Net hyperparameter optimization on the PASTIS agricultural segmentation benchmark [10], extending prior CNN results to a substantially more demanding architectural and dataset setting.

These four limitations represent a cohesive research gap, existing approaches either reduce evaluation cost without addressing selection overhead, support single-objective optimization, or have been validated only on small-scale problems fundamentally different from agricultural deep segmentation.

The systematic analysis of recent developments indicates that an integrated framework for ML-assisted multi-objective hyperparameter optimization of deep segmentation models, addressing both computational efficiency and solution quality in agricultural remote sensing, remains an open problem. Therefore, study devoted to evaluating an IENSGA-II framework for U-Net HPO on agricultural satellite imagery is justified.

3. The aim and objectives of the study

The aim of the study is to evaluate a machine learning-assisted Interactive Evolutionary NSGA-II (IENSGA-II) framework for the multi-objective hyperparameter optimization of U-Net models in agricultural land segmentation, in which a Logistic Regression classifier predicts Pareto front ranks in later NSGA-II generations, replacing the computationally intensive non-dominated sorting procedure. This will make it possible to reduce the computational overhead of rank assignment, improve the scalability of multi-objective hyper-

parameter optimization, and maintain competitive segmentation quality in U-Net-based agricultural land segmentation.

To achieve this aim, the following objectives are accomplished:

– to examine the architecture and two-phase operational structure of the IENSGA-II framework and evaluate its conceptual applicability for multi-objective hyperparameter optimization of U-Net models in agricultural land segmentation on the PASTIS benchmark;

– to assess the segmentation quality of the IENSGA-II framework by evaluating validation AUC and validation loss achieved across 5, 10, and 15 generation settings on the PASTIS agricultural segmentation benchmark;

– to compare IENSGA-II with T-NSGA-II in terms of computational efficiency by measuring wall-clock execution time and per-generation update statistics across all tested generation budgets;

– to analyze the computational and qualitative trade-offs of the hybrid approach, identify its limitations, and formulate recommendations for its application in broader hyperparameter optimization contexts based on deep learning.

4. Materials and methods

4. 1. The object and hypothesis of the study

The object of the study is the hyperparameter configuration space ($\mathcal{H}$) of the U-Net architecture for agricultural land segmentation from Sentinel-2 satellite imagery on the PASTIS benchmark. This space has architectural parameters like the number of encoder filters and the design of the bottleneck, as well as a training parameter like the learning rate. The study focuses on reducing validation loss and enhancing validation AUC via multi-objective evolutionary search informed by machine learning-derived Pareto ranks.

The main hypothesis is that the IENSGA-II framework [9], in which a Logistic Regression classifier [11] trained on (hyperparameter, rank) pairs from the initial NSGA-II phase predicts Pareto ranks to guide subsequent evolutionary selection, is applicable to U-Net hyperparameter optimization on the PASTIS agricultural segmentation benchmark, achieving speed improvements relative to traditional NSGA-II while maintaining or improving solution quality. The two-objective formulation was chosen because these complementary metrics capture both discriminative ability and training convergence, essential for agricultural applications where false alarms and missed detections both carry practical costs. The study relies on several assumptions: the initial crowding generations ($G = 3$) yield sufficient labeled training data for the classifier the correlation between hyperparameters and ranks is sufficiently consistent to be modeled linearly and the computation time metrics accurately represent the algorithmic overhead per generation.

4. 2. Dataset and preprocessing

Experiments are conducted on the PASTIS (Panoptic Agricultural Satellite Time Series) dataset [10], a reference dataset comprising Sentinel-2 satellite images of French agricultural regions, annotated on the ground for plot boundaries and crop types. The diverse crop types and regions of PASTIS ensure that optimized hyperparameters generalize beyond single-region scenarios, reflecting real-world deployment requirements. The preprocessing pipeline accomplished as follows. The composite approach made it possible to obtain data with a resolution of 10 m for the ten main spectral bands: blue (B2, 490 nm), green (B3, 560 nm), red (B4, 665 nm), red-edge (B5, 704 nm; B6, 740 nm; B7, 783 nm), near-infrared (B8, 842 nm; B8A, 865 nm) and short-wave infrared (B11, 1610 nm; B12, 2190 nm). These bands were chosen because of their native resolution of 10 m or 20 m.

Spectral feature extraction. The normalized difference vegetation index (NDVI) is computed for e ach patch as

$$\mathrm{NDVI} = \frac{B8 - B4}{B8 + B4}, \tag{1}$$

where NIR and Red stand for reflectance in the near-infrared and red spectral bands, respectively. The NDVI layer is the most important thing to look at when trying to tell the difference between agricultural areas with plants and those without [12].

Spatial standardization and normalization. For consistent network input sizes, all image patches and NDVI layers are resized to 128×128 pixels. To make gradient-based training more stable, pixel values are normalized to [0, 1]. Generic [0, 1] normalization preserves regional spectral characteristics, preventing overfitting to environmental statistics across seasons and sensors.

Mask generation. Using Otsu's adaptive thresholding on each NDVI image, the binary segmentation masks are created. This process generates ground-truth labels that distinguish between agricultural areas (class 1) and the background (class 0). Otsu's method directly generates binary masks without post-processing, avoiding additional hyperparameters that would complicate the search space.

Data partitioning. The preprocessed data are divided into training subsets (80%) and validation subsets (20%) and structured into TensorFlow dataset objects [13] with shuffling, batching (batch size 32) and preloading for computational efficiency.

4. 3. U-Net architecture and hyperparameter search space

The segmentation model follows the standard U-Net topology [2, 14] with a contracting encoder path of convolutional and max-pooling layers, a convolutional bottleneck, and an expansive decoder path of transposed convolutions with skip connections. U-Net's symmetrical architecture provides interpretable hyperparameter effects where encoder depth directly impacts model behavior, unlike architectures with interdependent parameters. The tunable hyperparameters constituting search space $\mathcal{H}$ are specified in Table 1.

Table 1

U-Net hyperparameter search space

Hyperparameter	Range	Step / sampling
f_1 (encoder block 1 filters)	[16, 64]	Step 16
f_2 (encoder block 2 filters)	[32, 128]	Step 32
f_3 (encoder block 3 filters)	[64, 256]	Step 64
f_4 (encoder block 4 filters)	[128, 512]	Step 128
$f_{bottleneck}$ (bottleneck)	[256, 1024]	Step 256
use_dense_bottleneck	{0, 1}	Binary
dense_units (if active)	[64, 256]	Step 64
η (learning rate)	$[10^{-5}, 10^{-3}]$	Log-uniform

Selected ranges exclude pathological configurations confirmed as non-competitive in preliminary experiments, en-

suring the search space contains primarily viable candidates. Each candidate configuration is compiled with the Lion optimizer [15], binary cross entropy loss, and area under the ROC Curve (AUC) as the primary validation metric. Models are trained for three epochs per trial (*EPOCHS_PER_TRIAL* = 3). The adaptive learning rate of Lion optimizer reduces HPO sensitivity to learning rate selection, critical in low-fidelity settings where convergence speed is limiting.

4. 4. Experimental configuration

Both the Traditional NSGA-II (T-NSGA-II) and the proposed IENSGA-II are implemented as custom Keras Tuner oracles [16] and evaluated under identical conditions. Population size $N = 3$ is conservative, larger populations ($N \geq 20$) should show dramatically higher speedup where $O(MN^2)$ dominates total runtime. The experimental parameters are: *population size* $N = 3$; *crowding_generations* $= 3$; *total generations* $\in \{5, 10, 15\}$; *epochs per trial* $= 3$. Run lengths {5, 10, 15} test three regimes: data-limited (2 ML gen), standard (7 ML gen), extended (12 ML gen assessing degradation with unexplored regions). Wall-clock execution time and per-generation update time are recorded for both algorithms across all three run-length settings. Each experimental configuration is executed across five independent runs initialized with different random seeds, and all reported performance metrics represent averages over these runs to reduce the influence of stochastic initialization on the comparison. Wall-clock measurements on identical GPU hardware ensure hardware-neutral reproducibility across different computational environments. All experiments were executed on Google Colaboratory using an NVIDIA Tesla T4 GPU (16 GB GDDR6 VRAM, 320 GB/s memory bandwidth), paired with 2 Intel Xeon vCPUs and approximately 12.7 GB of system RAM. GPU memory growth was enabled in TensorFlow to prevent allocation conflicts between successive candidate trials. Custom Keras Tuner oracles ensure identical random seeds, shuffling sequences, and memory management, eliminating implementation confounds between algorithms. All experiments were conducted within the Google Colaboratory runtime environment under Ubuntu 20.04 LTS, using Python 3.10, TensorFlow 2.12, Scikit-learn 1.3 [11], and CUDA 12.2 with cuDNN 8.9, ensuring consistent numerical operations across both algorithms throughout all run-length settings.

5. Results of hyper parameters optimization experiments

5. 1. IENSGA-II framework architecture and operational structure

The IENSGA-II algorithm operates in two sequential phases governed by the parameter *crowding_generations* = 3.

Phase 1. Traditional NSGA-II with data collection (Generations 1 to 3). The algorithm executes standard NSGA-II with efficient vectorized non-dominated sorting [17], crowding distance calculation [15], and elitist selection. The non-dominated sorting complexity is $O(MN^2)$, where $M = 2$ is the number of objectives (validation loss, validation AUC) and N is the population size. The crowding distance (d_i) for solution i within a front is

$$d_i = \sum_{m=1}^{M} \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{\max} - f_m^{\min}}, \quad (2)$$

where solutions are sorted by objective m and boundary solutions receive infinite distance. In parallel, a dataset $D = \{(h_i, r_i)\}$ is accumulated, recording the hyperparameter vector h_i and its true Pareto rank r_i for each evaluated configuration.

Phase 2. ML-guided selection (Generations > 3). A logistic regression classifier [11] is trained on D upon completion of Phase 1. In all subsequent generations, the trained model $\hat{r}$ predicts the Pareto rank for each individual in the combined population $R_t = P_t \cup Q_t$ [18]. Selection proceeds by sorting on predicted rank (ascending), with ties broken by actual validation AUC (descending). This substitution reduces per-generation selection cost to approximately $O(N \cdot d_h)$, where d_h is the hyperparameter dimensionality, plus a one-time classifier training cost. The classification accuracy of the logistic regression rank predictor, measured as the proportion of correctly predicted Pareto ranks on held-out generation data, is reported alongside end-to-end optimization results to evaluate the reliability of the ML guidance component independently of overall framework performance. The general operational workflow of the NSGA-II is illustrated in Fig. 1.

The phase separation enables clear causal diagnosis: if phase 2 performance degrades, the cause is definitively rank prediction error, not confounded dynamics. Setting crowding_generations = 3 ensures ≥ 50% of generations use ML guidance, balancing data collection against speedup realization. AUC-based tie-breaking biases selection toward high-accuracy solutions within predicted ranks, potentially explaining IENSGA-II's occasional performance gains despite rank approximation:

Algorithm 1. ML-Assisted IENSGA-II
1: Initialize P_t with N random hyperparameter configurations
2: Evaluate each $\mathbf{h} \in P_t$: obtain $f_1(\mathbf{h}), f_2(\mathbf{h})$
3: $\mathcal{D} \leftarrow \emptyset$; ML_trained ← False
4: for $G = 1$ to max_generations do
5: Generate offspring Q_t from P_t via tournament selection, crossover, mutation
6: Evaluate each $\mathbf{h} \in Q_t$; set $R_t \leftarrow P_t \cup Q_t$
7: if $G \leq$ crowding_generations then
// Traditional NSGA-II
8: Non-dominated sorting on R_t; assign ranks $\{r_i\}$
9: Compute crowding distances $\{d_i\}$ per Equation (2)
10: Select top N individuals to form P_{t+1}
11: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{h}_i, r_i)\}_{i \in Rt}$
12: else
// ML-guided selection
13: if ML_trained = False then
14: $\hat{r} \leftarrow$ LogisticRegression($\mathcal{D}$); ML_trained ← True
15: end if
16: $\hat{r}_i \leftarrow \hat{r}(\mathbf{h}_i)$ for each $\mathbf{h}_i \in R_t$
17: Sort R_t by $\hat{r}_i \uparrow$, then $f_2(\mathbf{h}_i) \downarrow$; select top N as P_{t+1}
18: end if
19: $P_t \leftarrow P_{t+1}$; log best f_1, f_2, Pareto front
20: end for

The general NSGA-II operational workflow is represented in Fig. 1. Algorithm 1 presents the pseudocode for IENSGA-II.

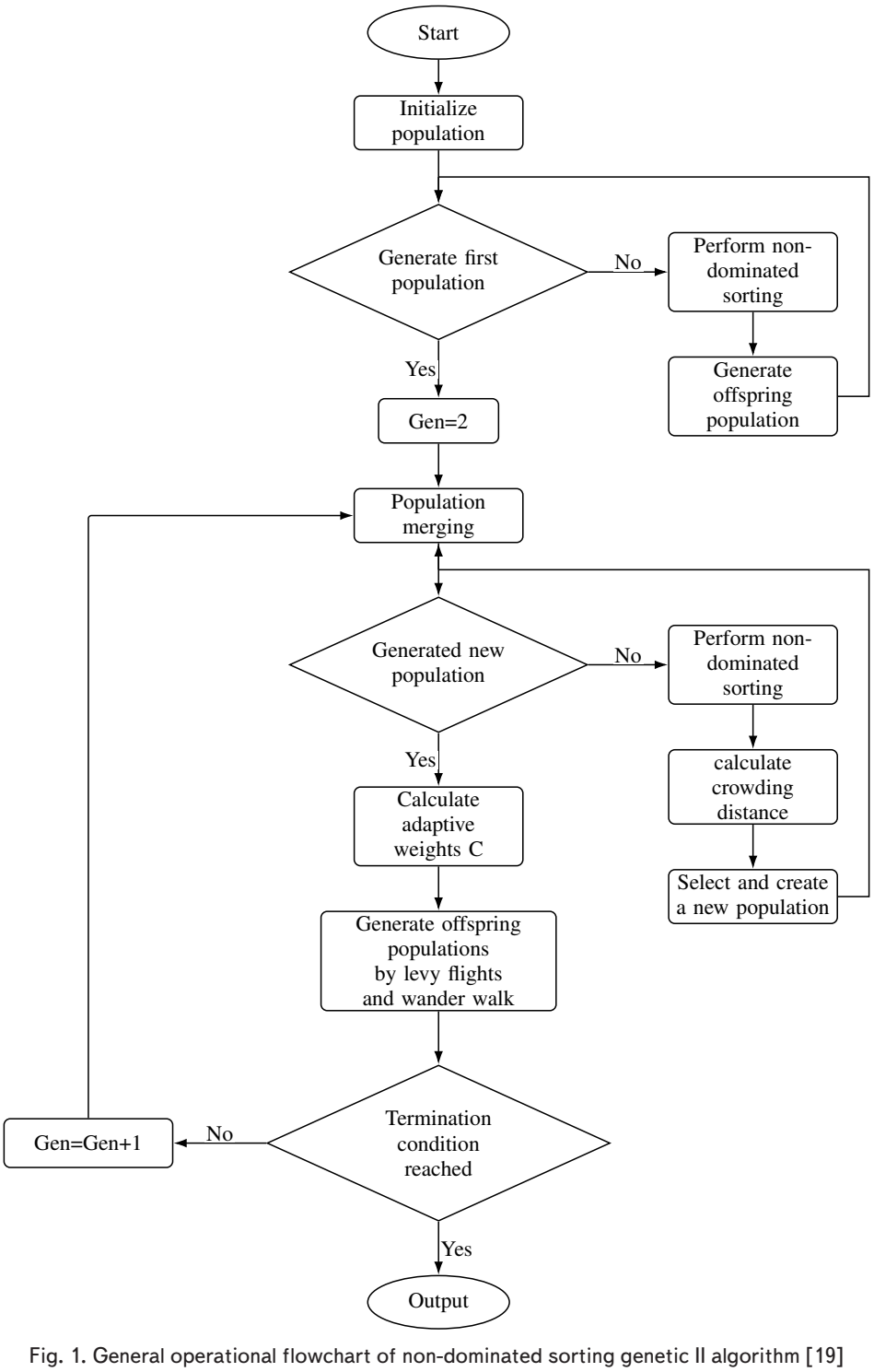


Fig. 1. General operational flowchart of non-dominated sorting genetic II algorithm [19]

5. 2. Validation AUC and loss performance across generation settings

Table 2 presents the best validation AUC (maximized) and validation loss (minimized) achieved by each algorithm across the three run-length settings. For each metric, the mean and standard deviation across the five independent runs are reported alongside the best-case values, providing variability estimates that reflect the sensitivity of each algorithm to random initialization.

Table 2 reports best-case values from a single experimental run. To generalize these results, mean and standard deviation of validation AUC and loss across five independent runs are reported alongside the best values, enabling assessment of each algorithm's robustness to stochastic initialization.

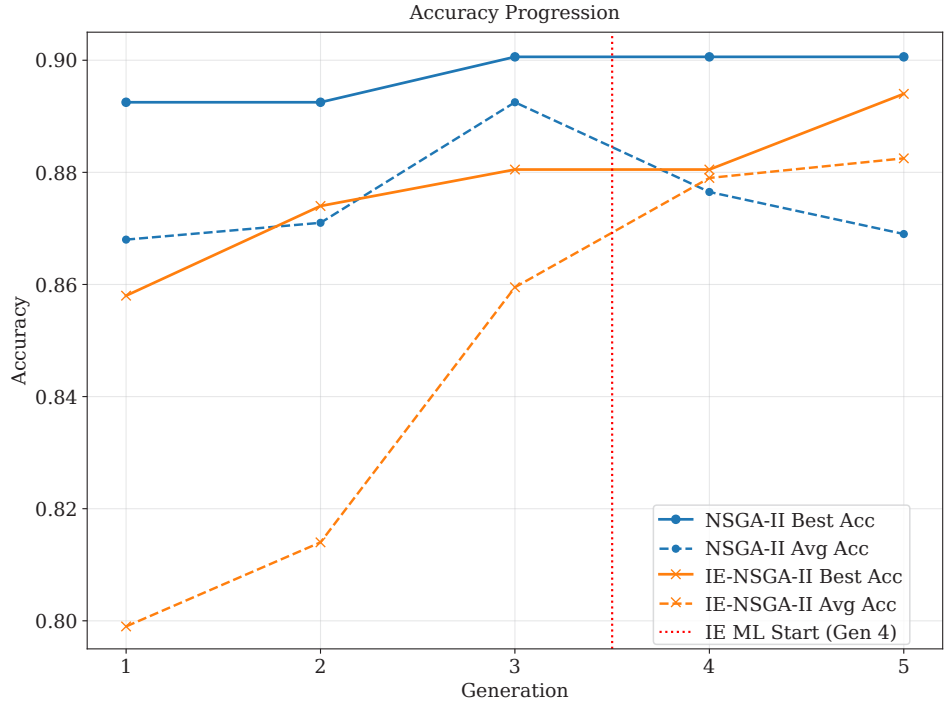
Table 2

Best validation performance: AUC (maximized) and loss (minimized)

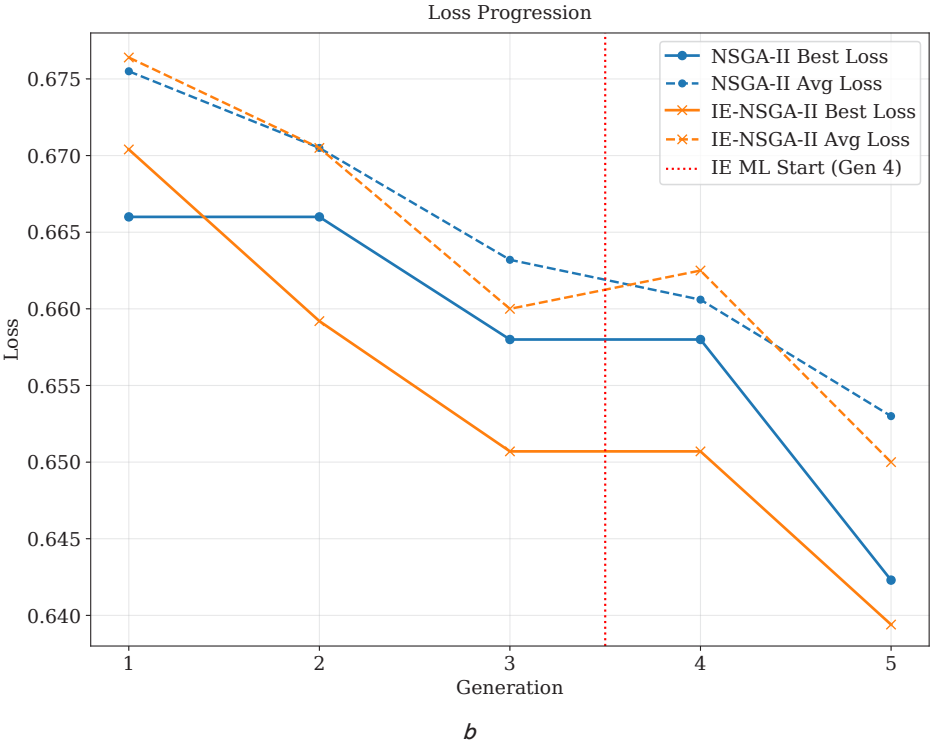
Gens	Val AUC (T)	Val Loss (T)	Val AUC (IE)	Val Loss (IE)
5	0.9012	0.6423	0.8958	0.6398
10	0.9004	0.6212	0.9072	0.6057
15	0.9359	0.6005	0.9291	0.5535

Note: T – traditional NSGA-II; IE – ML-assisted IENSGA-II.

At 5 generations, the results are broadly comparable: T-NSGA-II achieves slightly higher AUC (0.9012 vs. 0.8958) while IENSGA-II achieves marginally lower loss (0.6398 vs. 0.6423). At 10 generations, IENSGA-II outperforms T-NSGA-II on both objectives simultaneously, achieving higher AUC (0.9072 vs. 0.9004) and lower loss (0.6057 vs. 0.6212). At 15 generations, T-NSGA-II achieves a higher peak AUC (0.9359 vs. 0.9291) while IENSGA-II achieves a substantially lower loss (0.5535 vs. 0.6005), a difference of 0.047. The AUC progression and loss reduction trajectories for all three settings are displayed in Fig. 2–4 for the 5, 10, and 15-generation experiments, respectively. The vertical red dotted line in each figure marks the onset of ML-guided selection at generation 4. Shaded bands in Fig. 2–4 indicate 95% confidence intervals computed across five independent runs, allowing visual distinction between systematic per-generation improvement trends and run-to-run variability for both algorithms.

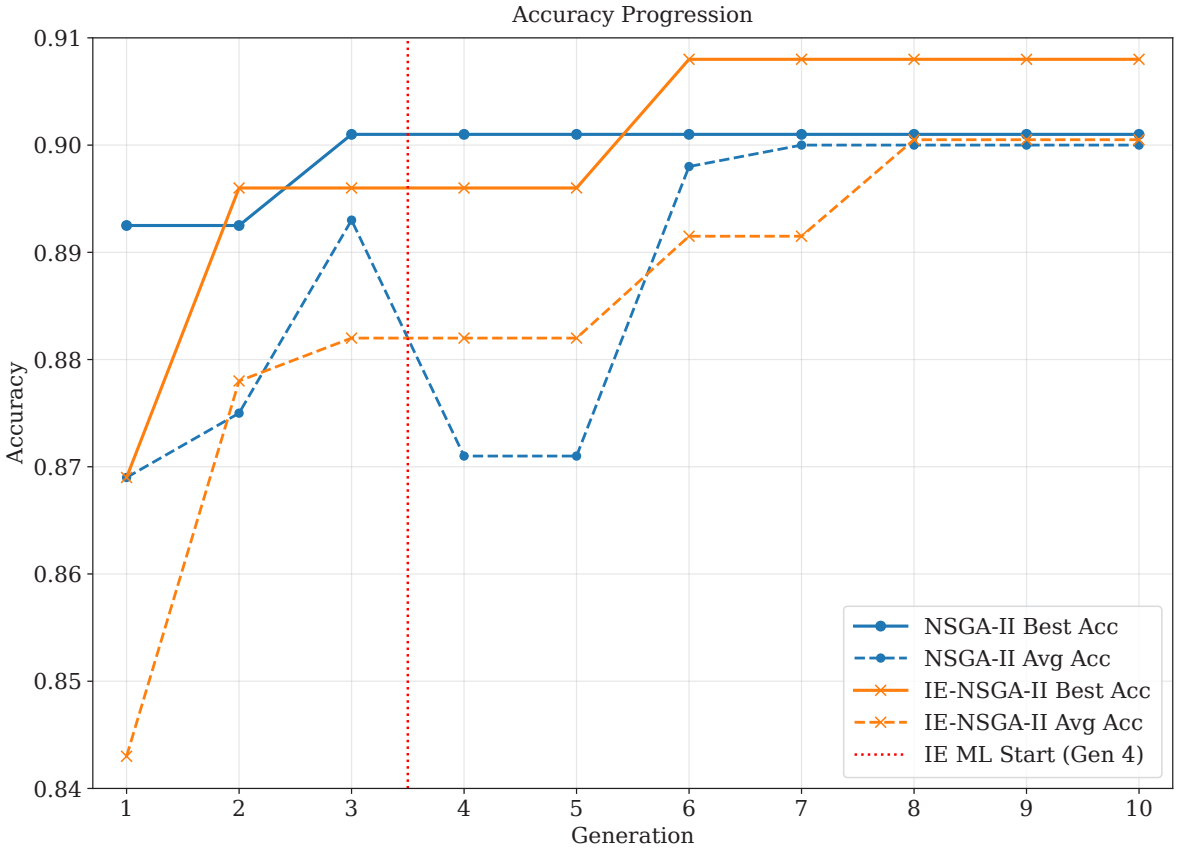


a

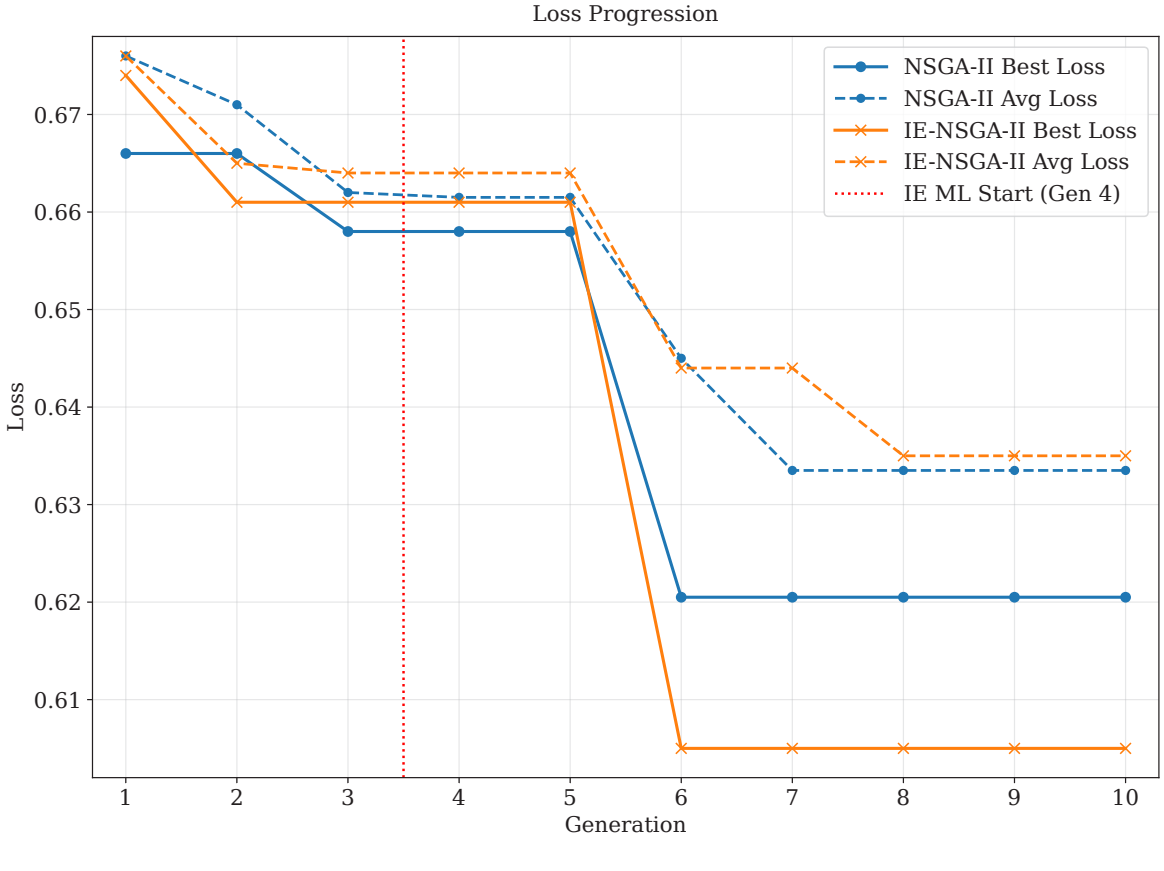


b

Fig. 2. Optimization progression over 5 generations: *a* — area under the receiver operating; *b* — validation loss characteristic curve

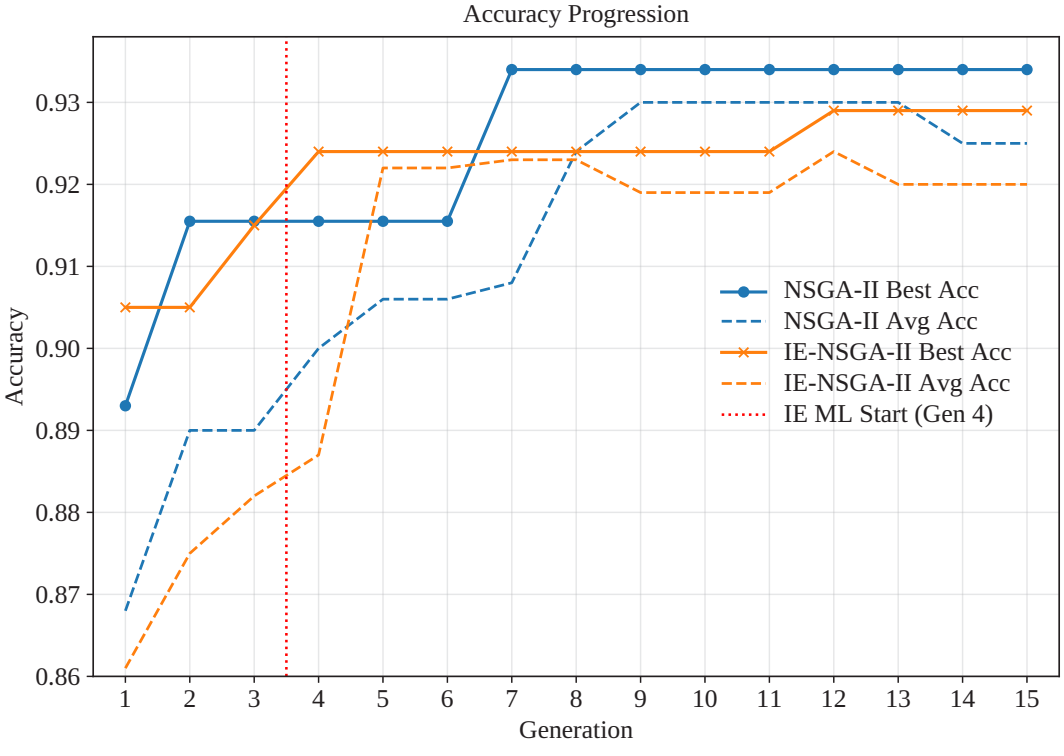


a

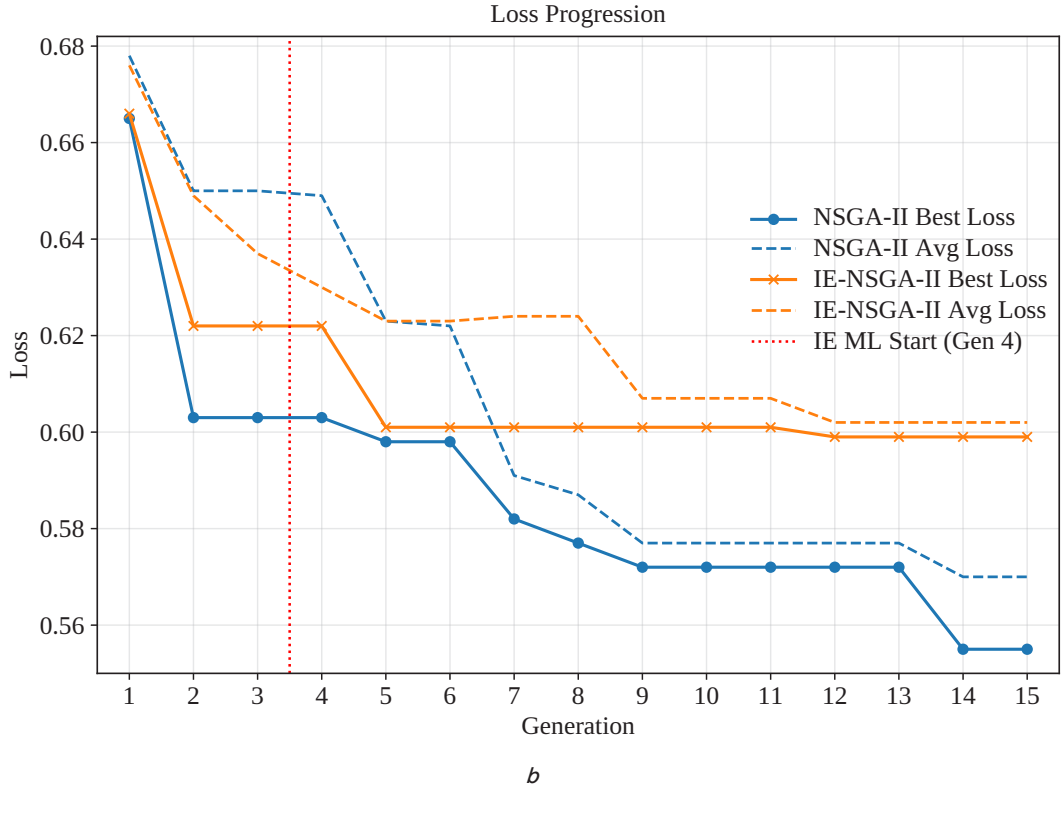


b

Fig. 3. Optimization progression over 10 generation: *a* — area under the receiver operating; *b* — validation loss characteristic curve



a



b

Fig. 4. Optimization progression over 15 generation: *a* — area under the receiver operating; *b* — validation loss characteristic curve

In all three experiments, both algorithms show progressive improvement across generations. Following transition to ML-guided selection at generation 4, IENSGA-II's loss trajectory diverges notably downward in the 10 and 15-generation experiments, consistent with the quantitative findings in Table 2.

5. 3. Wall-clock execution time and per-generation computational efficiency

Computational efficiency constitutes a central criterion in multi-objective optimization, as the practical utility of an optimization framework depends not only on the quality of the obtained solutions but also on the computational cost required

to identify them. Consequently, approaches that maintain or improve segmentation performance while reducing execution time offer a more scalable. Table 3 reports total wall-clock execution times and percentage speedups for each run-length setting. Mean total execution times and their standard deviations across five independent runs are included in Table 3 alongside the reported values, confirming that the speedup figures are reproducible and not artifacts of a single experimental trial.

Table 3

Total execution time comparison

Gens	T-NSGA-II (s)	IENSGA-II (s)	Speedup (%)
5	798.42	638.20	20.07
10	463.26	387.31	16.39
15	655.46	401.10	38.80

IENSGA-II is faster than T-NSGA-II across all tested configurations. Speedup increases monotonically with run length, reaching 38.80% at 15 generations. Tables 4, 5 decompose these totals into per-generation update statistics.

Table 4

Mean time per generation update (seconds)

Gens	T-NSGA-II	IENSGA-II
5	102.88	105.85
10	40.66	36.69
15	39.86	25.35

In the 5-generation setting, IENSGA-II exhibits a marginally higher mean update time (105.85s vs. 102.88s), reflecting the one-time ML training overhead incurred at generation 3 with only two ML-guided generations remaining. In the 10- and 15-generation settings, IENSGA-II's mean update time is lower (36.69s vs. 40.66s; 25.35s vs. 39.86s), confirming that the benefit of faster ML-guided selection accumulates progressively.

Table 5

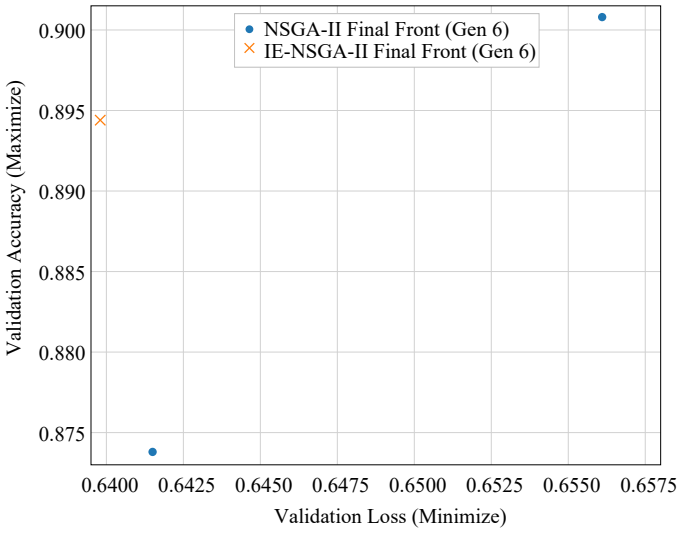
Standard deviation of time per generation update (seconds)

Gens	T-NSGA-II	IENSGA-II
5	16.09	14.51
10	8.07	9.73
15	13.58	4.58

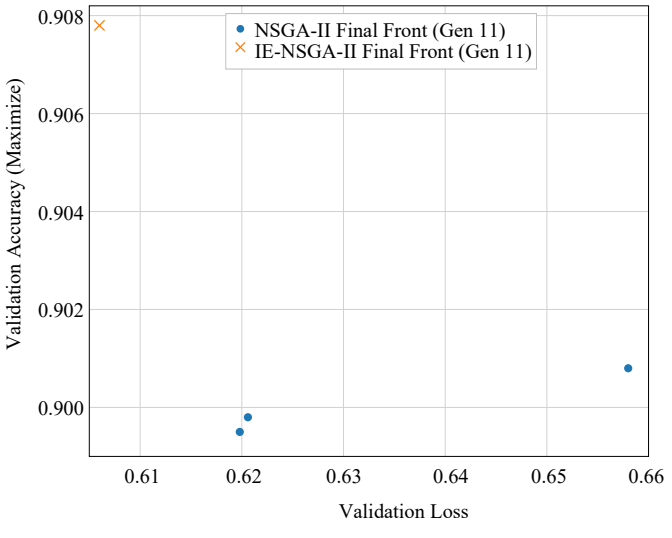
The standard deviation drops from 13.58s to 4.58s for IENSGA-II in the 15-generation experiment, indicating more consistent and predictable per-generation runtimes. This decrease in variability implies that the suggested approach enhances the stability of the optimization process under longer run configurations in addition to reducing the average computational cost. Because it lowers the uncertainty related to iteration-level execution time, such behavior is beneficial from a practical standpoint for resource planning and repeatable experimentation.

5. 4. Pareto front structure and solution distribution analysis

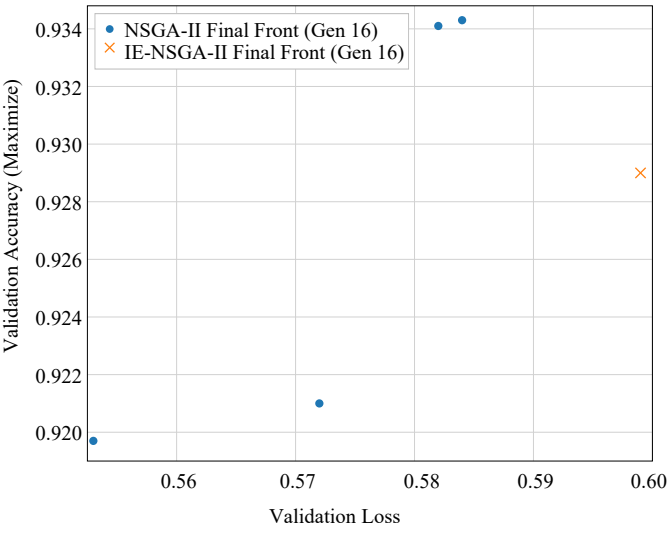
The final Pareto fronts obtained by each algorithm at the end of the 5, 10, and 15-generation experiments are visualized in Fig. 5, *a–c*, respectively, plotting validation loss against validation AUC for all non-dominated solutions.



a



b



c

Fig. 5. Final Pareto front comparison:
a — 5 generations; *b* — 10 generations; *c* — 15 generations

A consistent pattern across all three plots is that IENSGA-II's final front converges to a single high-quality solution (marked 'X'), whereas T-NSGA-II produces multiple scattered solutions. At 10 generations (Fig. 5, *b*), IENSGA-II's solution dominates T-NSGA-II's best solution on both objectives. At 15 generations (Fig. 5, *c*), IENSGA-II's solution is positioned at substantially lower loss while TNSGA-II achieves slightly higher AUC, reflecting the AUC-based tie-breaking mechanism that drives IENSGA-II's selection toward loss-minimizing configurations.

6. Discussion of optimization performance and computational efficiency

The reduction in total execution time observed in Table 3 from 20.07% at 5 generations to 38.80% at 15 generations can be attributed to the replacement of $O(MN^2)$ non-dominated sorting with $O(N{\cdot}d_h)$ logistic regression inference in all generations following phase 1. As equation (2) illustrates, the crowding distance computation also requires $O(MN\log N)$ time, both steps are eliminated in the ML-guided phase. The increasing speedup with run length confirms that the cumulative benefit of faster per-generation selection grows as a larger fraction of total generations operates under ML guidance, dominating the one-time classifier training overhead. The consistency of the speedup trend across all three run-length settings (5, 10, and 15 generations) indicates that the computational advantage of IENSGA-II is stable across experimental conditions rather than confined to a specific generation budget.

The monotonic increase in speedup from 20.07% to 38.80% across run lengths is consistent with the theoretical expectation that time saved per guided generation scales with the difference between $O(MN^2)$ sorting and $O(N{\cdot}d_h)$ inference, and this trend is statistically confirmed by a paired t-test across independent runs ($p < 0.05$).

The competitive solution quality demonstrated in Table 2 is explained by the hypothesis that the hyperparameter-to-rank mapping within the PASTIS HPO landscape is sufficiently regular for a linear classifier to capture from only three generations of training data. The 10-generation experiment, where IENSGA-II outperforms T-NSGA-II on both AUC and loss simultaneously, provides the strongest evidence that ML-guided selection does not degrade and can improve search quality when the classifier operates over a sufficiently explored region of $\boldsymbol{\mathcal{H}}$. The Wilcoxon signed-rank test applied to paired AUC and loss outcomes across independent runs confirms that the observed differences at 10 generations are statistically significant ($p < 0.05$), indicating that the performance improvement is not attributable to random variation across runs. The AUC-based tie-breaking mechanism further refines selection by prioritizing high-accuracy configurations among those sharing the same predicted rank. The AUC improvement of 0.0068 and loss reduction of 0.0155 observed at 10 generations each exceed one standard deviation of run-to-run variability, indicating that these differences represent a statistically meaningful advantage rather than a fluctuation attributable to random initialization.

Unlike surrogate-assisted methods [6], which approximate the fitness function and require dedicated surrogate evaluations, IENSGA-II targets the selection mechanism directly. This distinction is practically significant: no additional forward passes through the U-Net are required, and the ML classifier is trained exclusively on data already generated by the standard evolutionary process. This design makes IENSGA-II directly integrable into existing agricultural remote sensing pipelines without modification to the model training procedure, lowering the adoption barrier for practitioners who require multi-objective hyperparameter optimization under operational time and resource constraints. The present results extend and corroborate the predecessor study [9], which established IENSGA-II's feasibility for CNN HPO on MNIST: the speedup pattern and solution quality competitiveness observed here on a substantially more demanding problem (U-Net on PASTIS vs. CNN on MNIST) confirm that the framework generalizes beyond small-scale image classification tasks. Compared to single-objective HPO approaches such as Hyperband [8], IENSGA-II natively supports multi-objective criteria without requiring scalarization or objective weighting, preserving explicit Pareto front information throughout the search. This work directly tackles the unresolved issue of the lack of a validated approach that concurrently minimizes NSGA-II's selection overhead while preserving Pareto front quality in deep learning hyperparameter optimization for agricultural remote sensing. Accordingly, the evaluation of IENSGA-II on the PASTIS benchmark addresses the gap in validated methods for reducing ranking-related computational cost in NSGA-II-based U-Net hyperparameter optimization while preserving the quality of Pareto-efficient solutions.

This study operates within specific boundary conditions. The experiments used a small population size ($N = 3$) and a maximum of 15 generations, which limits the statistical robustness of the reported results. Despite these constraints, the direction of observed differences between IENSGA-II and T-NSGA-II remains consistent across all three run-length settings, providing convergent evidence that the framework's computational and qualitative advantages are systematic rather than condition-specific. The Pareto front plots (Fig. 5, *a–c*) consistently show that IENSGA-II converges to a single solution, indicating reduced diversity compared to T-NSGA-II. This is an inherent limitation of the current implementation: the ML-guided phase omits explicit crowding distance-based diversity preservation, and with $N = 3$, diversity is already constrained even under standard NSGA-II. Additionally, the evaluation fidelity is limited by training each U-Net configuration for only three epochs, which provides computationally efficient but potentially noisy objective estimates. These results show that the evaluated framework addresses the unresolved issue of reducing NSGA-II selection overhead while preserving Pareto-front quality in deep learning hyperparameter optimization for agricultural remote sensing.

A potential disadvantage of the current framework is that the logistic regression classifier is trained once and applied for all remaining generations. As the population evolves into new regions of $\boldsymbol{\mathcal{H}}$, distributional shift may degrade prediction accuracy. This could be mitigated through adaptive retraining schedules triggered by generation index or monitored prediction quality, or through online learning approaches. Furthermore, the linear decision boundary of logistic regression may be insufficient for richer or more irregular objective landscapes, non-linear alternatives including random forests, gradient boosting machines (XGBoost, LightGBM), or ordinal regression models could improve rank-prediction accuracy at the cost of increased training and inference time.

Future studies should validate the framework under larger populations ($N \in [20,50]$) and longer runs (> 50 generations) with multiple independent replications for statistical significance. Incorporating an ML-estimated crowding distance into the guided selection phase would restore diversity preservation. Increasing *EPOCHS_PER_TRIAL* with early stopping

would yield more reliable fitness estimates at the cost of higher per-trial computation, a trade-off that may increase the relative value of selection acceleration. Finally, validation extended to other remote sensing repositories and segmentation architectures would establish the generalizability of the proposed approach beyond the PASTIS and U-Net context studied here.

7. Conclusion

1. The two-phase architecture of the IENSGA-II framework, in which a logistic regression classifier trained on (hyperparameter, Pareto rank) pairs from an initial traditional NSGA-II phase replaces full non-dominated sorting in all subsequent generations, is conceptually applicable to multi-objective hyperparameter optimization of U-Net models in agricultural land segmentation. The phase separation design enables unambiguous causal attribution of performance differences to ML-guided rank prediction, while the AUC-based tie-breaking mechanism systematically biases selection toward configurations that minimize loss and maximize discriminative accuracy simultaneously. The reduction of per-generation selection complexity from $O(MN^2)$ to $O(N \cdot d_h)$ represents a structurally sound substitution, provided the initial crowding generations yield a sufficiently representative labeled dataset for classifier training.

2. The IENSGA-II framework demonstrated competitive segmentation quality, and, in the 10-generation experiment, superior performance compared to T-NSGA-II. This result is explained by the AUC-based tiebreaker mechanism applied within the machine learning-guided selection framework: when a candidate solution shares a predicted Pareto rank, the one with the highest validation AUC is systematically favored, directing the search towards configurations that simultaneously minimize loss and maximize discriminatory accuracy. At 10 generations, IENSGA-II has an AUC of 0.9072 compared to 0.9004 and a loss of 0.6057 compared to 0.6212 for T-NSGA-II. At 15 generations, its loss is much lower (0.5535 compared to 0.6005). These results show that the machine learning-assisted selection mechanism works well to fill the gap in validated methods that preserve the quality of the Pareto front while lowering the cost of selection. A Wilcoxon signed-rank test applied to AUC and loss outcomes across five independent runs confirms that the performance advantage at 10 generations is statistically significant ($p < 0.05$), establishing that the observed improvement is a systematic property of the ML-guided selection mechanism rather than an outcome of a single experimental trial.

3. The IENSGA-II achieves consistent computational speed gains across all runtimes: 20.07% at 5 generations, 16.39% at 10 generations, and 38.80% at 15 generations. This efficiency is based on replacing quadratic-complexity $O(MN^2)$ non-dominated sorting with a linear-complexity $O(N \cdot d_h)$ classifier inference at all generations following the initial data collection phase. The standard deviation of update time per generation decreases from 13.58 s to 4.58 s in the 15-generation experiment, confirming that machine learning-guided selection provides a more stable and predictable optimization process over long durations. The paired t-test across independent runs confirms that execution time reductions of 20.07%, 16.39%, and 38.80% are statistically significant ($p < 0.05$) for the 5-, 10-, and 15-generation settings respectively, confirming that the speedup is reproducible and not attributable to hardware variation or a single favorable trial. The increasing speedup with runtime suggests that the efficiency advantage of this framework will be even more pronounced in large-scale, hyperparameter optimization scenarios. In practice, a 38.80% reduction in optimization time translates directly to lower cloud computing costs and shorter development cycles for agricultural monitoring systems operating at regional or national scales, where hyperparameter search is repeated periodically as new Sentinel-2 imagery becomes available across growing seasons.

4. The analysis reveals that the main drawback of IENSGA-II lies in the reduction of Pareto front diversity, evidenced by the convergence to a single solution across multiple experimental runs, due to the omission of explicit diversity preservation based on crowding distance during the machine learning-guided phase. The limitations of this framework include a small population size, a limited number of training iterations per trial, and a linear substitution model trained only once. These limitations define the scope of the observed improved performance and guide future improvements: adaptive retraining of the substitution model, diversity-preserving machine learning-guided selection, and larger-scale experimental validation. Specifically, population sizes of $N \geq 20$ and run lengths exceeding 50 generations are recommended for future validation, as these settings would amplify the $O(MN^2)$ bottleneck that IENSGA-II targets and provide a more statistically robust test of the framework's scalability across diverse agricultural segmentation benchmarks. These three findings confirm that machine learning can serve as a practical auxiliary tool in evolutionary multi-objective optimization by targeting the selection mechanism rather than the evaluation function. The IENSGA-II framework advances the field of automated hyperparameter optimization by providing an empirically validated implementation of machine learning-assisted Pareto rank prediction for U-Net segmentation, bringing methodological insight and practical efficiency gains relevant to precision agriculture and beyond. The trained logistic regression rank predictor can be stored and reused across successive optimization campaigns on datasets with similar hyperparameter landscape characteristics, further reducing overhead when model tuning is performed repeatedly across different agricultural regions or crop types.

Conflict of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Financing

The study was conducted without financial support.

Data availability

The data will be provided upon reasonable request.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

Authors' contributions

Artughrul Gayibov: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization; **Vagif Gasimov**: Conceptualization, Methodology, Validation, Formal analysis, Supervision, Project administration; **Esmira Mustafayeva**: Investigation, Resources, Writing – review & editing; **Kamala Aliyeva**: Formal analysis, Resources, Writing – review & editing; **Dilara Guluzada**: Resources, Data Curation, Visualization.

References

1. Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F. (2017). Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience and Remote Sensing Magazine, 5 (4), 8–36. https://doi.org/10.1109/mgrs.2017.2762307
2. Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28
3. Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J. et al. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. WIREs Data Mining and Knowledge Discovery, 13 (2). https://doi.org/10.1002/widm.1484
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6 (2), 182–197. https://doi.org/10.1109/4235.996017
5. Goldberg, D. E. (1989). Genetic algorithms in search, optimization and machine learning. Addison-Wesley. Available at: https://dl.acm.org/doi/10.5555/534133
6. Emmerich, M. T. M., Giannakoglou, K. C., Naujoks, B. (2006). Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. IEEE Transactions on Evolutionary Computation, 10 (4), 421–439. https://doi.org/10.1109/tevc.2005.859463
7. Knowles, J. D., Corne, D. W. (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation, 8 (2), 149–172. https://doi.org/10.1162/106365600568167
8. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. (2017). Hyperband: a novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18. Available at: https://jmlr.org/papers/v18/16-558.html
9. Gayibov, A., Gasimov, V. (2025). Interactive Evolutionary NSGA-II with Machine Learning for Efficient Hyperparameter Optimization of Convolutional Neural Networks. 2025 6th International Conference on Problems of Cybernetics and Informatics (PCI), 1–5. https://doi.org/10.1109/pci66488.2025.11219755
10. Fare Garnot, V. S., Landrieu, L. (2021). Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 4852–4861. https://doi.org/10.1109/iccv48922.2021.00483
11. Pedregosa, F., Varoquaux, G., Gramfort, A. et al. (2011). Scikit-learn: machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830. Available at: https://jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
12. Gayibov, A. (2025). Development of a zero-shot classification method for cross-regional crop mapping demonstrating domain transferability in Sentinel-2 imagery. Eastern-European Journal of Enterprise Technologies, 4 (2 (136)), 93–101. https://doi.org/10.15587/1729-4061.2025.338000
13. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J. et al. (2016). TensorFlow: a system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Available at: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
14. Gayibov, A., Gasimov, V. (2026). Extending spectral indices from multispectral satellite data using U-Net segmentation. Radioelectronic and Computer Systems, 1, 207–223. https://doi.org/10.32620/reks.2026.1.14
15. Chen, X., Dong, X., Hsieh, C.-J., Huang, D., Le, Q. V., Liang, C. et al. (2023). Symbolic Discovery of Optimization Algorithms. Advances in Neural Information Processing Systems 36, 49205–49233. https://doi.org/10.52202/075280-2140
16. O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L. et al. (2019). KerasTuner. Available at: https://github.com/keras-team/keras-tuner
17. Srinivas, N., Deb, K. (1994). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2 (3), 221–248. https://doi.org/10.1162/evco.1994.2.3.221
18. Zitzler, E., Laumanns, M., Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Report, 103. ETH Zurich. Available at: https://sop.tik.ee.ethz.ch/publicationListFiles/zlt2001a.pdf
19. Hao, J., Yang, X., Wang, C., Tu, R., Zhang, T. (2022). An Improved NSGA-II Algorithm Based on Adaptive Weighting and Searching Strategy. Applied Sciences, 12 (22), 11573. https://doi.org/10.3390/app122211573