CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (2) (2026), Pages 2852–2873
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Genetic Algorithm-Based Missing Value Imputation: Conditional Superiority and Deployment-Time Fragility Under Nested Cross-Validation

Moch. Faiz Febriawan, Kartika Chandra Dewi*, and Hasanuddin Al-Habib

Data Science Study Program, Faculty of Mathematics and Natural Sciences, Universitas Negeri Surabaya, Surabaya 60231, Indonesia

Abstract

Missing data impairs downstream supervised learning in tabular analytics. While genetic algorithm (GA) imputation shows promise, existing evaluations predominantly assume pristine holdout test data. This study presents a deployment-aware evaluation of evolutionary imputation paradigms (global GA-based and localized GA-KNN) using a 5-fold outer by 3-fold inner nested cross-validation protocol across six missingness tiers (15% to 90%), five seeds, and four downstream tuning regimes, totaling 4,200 factorial runs (8,400 evaluations) on the BankChurners benchmark. Our findings reveal three insights. First, under clean holdout validation, GA-KNN achieves conditional discriminative advantages (0.9565 ± 0.0335 ROC-AUC), significantly outperforming standard KNN ($p < 0.001$). Under 75% sparsity, GA-KNN attains a slight numerical gain over MissForest (+0.0058 ROC-AUC) that does not achieve significance under cross-validation dependence correction ($t_{\text{corr}} = 1.46, p_{\text{corr}} = 0.156$), while performing comparably to Mean ($p = 0.613$) and Median ($p = 0.707$) on aggregate averages. Second, under realistic deployment where inference queries contain missing attributes, GA-based methods suffer severe fragility, recording the steepest degradation (ΔROC-AUC $= -0.1200$) and dropping to 0.8365 ± 0.1122, significantly outperformed by simple Mean ($0.8644 \pm 0.0935, p < 0.001$) and Median ($0.8626 \pm 0.0943, p < 0.001$) across 60%–90% missingness due to inductive distance distortion. Third, imputer rankings reverse depending on holdout completeness. Consequently, evolutionary imputation provides targeted advantages for offline training, but requires architectural adaptation for streaming incomplete deployment.

Keywords: credit card churn; deployment robustness; genetic algorithms; missing value imputation; nested cross-validation

Copyright © 2026 by Authors, Published by CAUCHY Group. This is an open access article under the CC BY-SA License (https://creativecommons.org/licenses/by-sa/4.0)

1. Introduction

The presence of missing values in tabular datasets constitutes an enduring obstacle across computational intelligence, predictive analytics, and empirical data science [1, 2]. In high-stakes operational domains such as credit card churn prediction and financial risk modeling, incomplete records frequently arise from telemetry dropouts, disclosure reluctance, and transactional integration failures [3, 4]. Naive handling such as complete-case analysis or listwise deletion drastically

*Corresponding author. E-mail: kartikadewi@unesa.ac.id

Received: September 17, 2026 Revised: November 12, 2026 Accepted: November 18, 2026
DOI: https://doi.org/10.18860/cauchy.v11i2.46097

diminishes statistical power and injects selection bias into downstream estimators [5, 6]. Consequently, principled data imputation has emerged as an indispensable phase in machine learning pipelines. Established methodologies encompass classical univariate substitution (Mean, Median) [7, 8], multivariate distance-based modeling (k-NN, MICE) [9, 10], and recursive ensemble trees such as MissForest [11, 12].

Despite widespread empirical adoption, conventional imputation algorithms share a fundamental theoretical assumption: they are formulated as unsupervised reconstruction tasks minimizing statistical distance metrics (e.g., RMSE or MAE) between observed and imputed coordinates [13, 14]. However, minimizing feature reconstruction error does not necessarily maximize class-conditional separation along downstream classification boundaries [13, 15]. Imputations that appear statistically optimal under quadratic criteria may nonetheless fall on the incorrect side of a decision threshold, directly confounding supervised learners such as Random Forest. To overcome this disconnect, Genetic Algorithms (GAs) have been investigated as promising alternatives capable of treating imputation as an explicit search problem guided directly by downstream classification fitness [16–19]. Pioneering work by Shahzad et al. [20] demonstrated that optimizing chromosomes against classifier performance can yield superior accuracy. Recent advances have further introduced localized manifold constraints, such as the EvoImp framework by Junior et al. [21] and continuous GA variants by Chungnoy and Srivihok [22], to constrain evolutionary search within valid data neighborhoods.

Nevertheless, an essential limitation characterizes the existing literature on evolutionary imputation: previous empirical evaluations have almost exclusively assessed performance on holdout test partitions that are assumed to be complete and fully observed [20, 21]. In real-world production systems, incoming test instances at deployment time frequently arrive with missing entries that must be imputed inductively before inference can occur. The operational robustness of GA-imputed models under such incomplete-inference scenarios remains unexamined. Furthermore, prior studies have predominantly relied on unnested single train/test splits with small evaluation samples ($N \leq 5$), leaving open questions regarding methodological stability, risk of optimistic evaluation bias, and interactions with modern hyperparameter tuning regimes [23, 24].

To resolve these open questions, this study conducts an empirical investigation into genetic algorithm-based imputation with downstream fitness optimization for credit card churn prediction under a rigorous, deployment-aware nested cross-validation framework. We evaluate two evolutionary architectures: a global GA-based imputer with hybrid population seeding and a localized hybrid imputer (GA-KNN) incorporating neighborhood topological guidance [21]. The primary analytical contribution is the formulation and execution of a dual-scenario evaluation protocol (clean control validation versus inductive incomplete deployment) embedded within a 5-fold outer by 3-fold inner nested cross-validation design, encompassing 4,200 factorial outer evaluation runs across seven imputation techniques, six missing rates (15% to 90%), five random seeds, and four downstream tuning algorithms (Default RF, RandomizedSearchCV, HalvingSearchCV, and Optuna).

The empirical contributions of this study are threefold:

1. **Rigorous confirmation under zero-leakage evaluation**: We confirm with high statistical power ($N = 25$ fold-seed pairs, $df = 24$) that the performance advantage of GA-based and GA-KNN imputation on complete holdout data [17, 20, 21] is genuine and reproducible under strict nested cross-validation, with GA-KNN significantly outperforming distance-based KNN ($p < 0.001$) while exhibiting a modest numerical gain over MissForest at 75% sparsity that moderates to non-significance under cross-validation dependence correction ($p_{\text{corr}} = 0.156$).
2. **Empirical evidence of deployment-time fragility**: When evaluated under realistic deployment conditions where query instances also contain missing attributes, the evolutionary advantage reverses. GA-based models exhibit the steepest performance degradation

(ΔROC-AUC ≈ -0.120), substantially underperforming simple univariate baselines (Mean and Median) across 60%–90% missingness ($p_{\text{Holm}} < 0.00001$) due to inductive metric distortion in their neighborhood mapping mechanisms.

3. **Methodological insight on protocol sensitivity**: We demonstrate empirically that comparative imputer rankings can invert depending on whether holdout queries are complete or incomplete, providing actionable guidance regarding when evolutionary optimization is beneficial and when simple univariate strategies should be preferred.

2. Methods

This section details the mathematical formulation, dataset characteristics, baseline imputation strategies, evolutionary architectures, downstream classification models, and the nested cross-validation evaluation protocol. The overall research pipeline is illustrated in Fig. 1, with the data preprocessing and splitting workflow shown in Fig. 2.

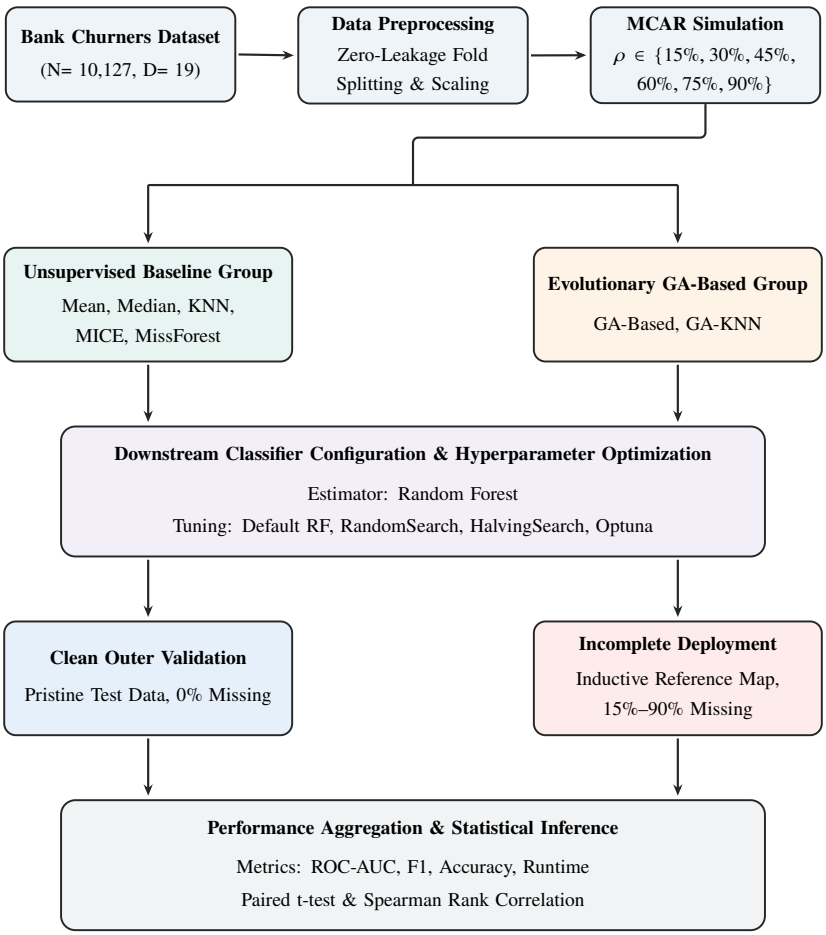

Fig. 1: Overall research framework and experimental pipeline of the comparative imputation study.

2.1. Mathematical Formulation of Tabular Imputation

Let a tabular dataset be represented as a design matrix $\mathbf{X} \in \mathcal{X}^{N \times D}$, comprising N observation instances and D predictor features, partitioned into D_{num} continuous numerical attributes $\mathcal{X}_{num} \subseteq \mathbb{R}$ and D_{cat} discrete categorical attributes $\mathcal{X}_{cat}$. Each row vector $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ is associated with a binary ground-truth classification target $y_i \in \{0, 1\}$. Under the Missing Completely at Random (MCAR) mechanism, the probability of an entry being missing is independent of both observed coordinates $\mathbf{X}_{obs}$ and unobserved coordinates $\mathbf{X}_{mis}$ [7]. Formally,

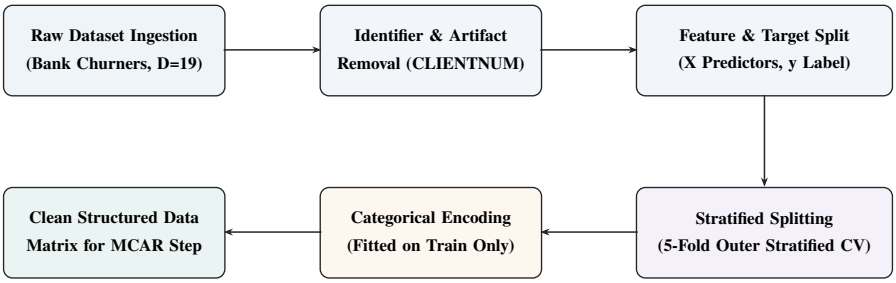

Fig. 2: Data preprocessing, stratified splitting, and controlled missingness injection pipeline.

the binary missingness indicator mask matrix $\mathbf{M} \in \{0,1\}^{N \times D}$ is defined in Eq. (1):

$$M_{i,j} = \begin{cases} 1, & \text{if } X_{i,j} \text{ is missing (unobserved)}, \\ 0, & \text{if } X_{i,j} \text{ is observed.} \end{cases} \tag{1}$$

The incomplete observed matrix is denoted by $\widetilde{\mathbf{X}} = \mathbf{X} \odot (\mathbf{1} - \mathbf{M}) + \text{NaN} \odot \mathbf{M}$, where $\odot$ denotes the Hadamard (element-wise) matrix product. As formalized in Eq. (2), the objective of an imputation function $f_{imp}\colon \widetilde{\mathbf{X}} \to \widehat{\mathbf{X}}$ is to generate a completed matrix $\widehat{\mathbf{X}} \in \mathcal{X}^{N \times D}$ such that:

$$\widehat{X}_{i,j} = \begin{cases} X_{i,j}, & \text{if } M_{i,j} = 0, \\ \hat{v}_{i,j}, & \text{if } M_{i,j} = 1, \end{cases} \tag{2}$$

where $\hat{v}_{i,j}$ represents the imputed replacement coordinate estimated by the algorithm.

2.2. Dataset Description and Preprocessing

The empirical evaluation is conducted on the benchmark BankChurners dataset [3], widely recognized in customer attrition analytics. The dataset contains $N = 10{,}127$ customer records described by $D = 19$ input attributes ($D_{num} = 13$ continuous attributes and $D_{cat} = 6$ categorical attributes). The target attribute is `Attrition_Flag`, exhibiting class imbalance (16.07% churners vs. 83.93% retained customers). Two extraneous Naive Bayes classifier artifact columns and the customer identifier (`CLIENTNUM`) were pruned.

To prevent information leakage, all preprocessing parameters were derived exclusively from the training partition of each cross-validation fold, as depicted in Fig. 2. Categorical features were encoded based solely on categories observed in the corresponding training partition. Continuous features were standardized using z-score normalization parameters derived exclusively from $\mathbf{X}_{train}$ according to Eq. (3):

$$z_{i,j} = \frac{X_{i,j} - \mu_j}{\sigma_j}, \tag{3}$$

where μ_j and σ_j denote the mean and standard deviation of feature j. Descriptive statistics of the continuous attributes are summarized in Table 1.

2.3. Controlled Missingness Simulation Protocol

Missing values were injected under a constrained MCAR protocol across six nominal rates: $\rho \in \{15\%, 30\%, 45\%, 60\%, 75\%, 90\%\}$. To ensure mathematical validity and avoid degenerate sample matrices, the stochastic masking was subjected to two feasibility constraints: (i) each row vector $\mathbf{x}_i$ retained at least one observed feature, preventing completely empty instances, and (ii) each feature column j retained at least one observed coordinate, preventing total feature collapse [25]. The simulated missingness strictly satisfied nominal targets for rates up to 75.0%, while the 90.0% nominal condition attained an actual missing rate of 89.3% due to these row/column feasibility constraints.

Table 1: Descriptive statistics of continuous predictor attributes in the BankChurners dataset

Feature Name	Mean	Std. Dev.	Min	Max
`Customer_Age`	46.33	8.01	26.00	73.00
`Dependent_count`	2.35	1.30	0.00	5.00
`Months_on_book`	35.93	7.99	13.00	56.00
`Total_Relationship_Count`	3.81	1.55	1.00	6.00
`Months_Inactive_12_mon`	2.34	1.01	0.00	6.00
`Contacts_Count_12_mon`	2.46	1.11	0.00	6.00
`Credit_Limit`	8631.95	9088.78	1438.30	34516.00
`Total_Revolving_Bal`	1162.81	814.99	0.00	2517.00
`Avg_Open_To_Buy`	7469.14	9090.69	3.00	34516.00
`Total_Amt_Chng_Q4_Q1`	0.76	0.22	0.00	3.397
`Total_Trans_Amt`	4404.09	3397.13	510.00	18484.00
`Total_Trans_Ct`	64.86	23.47	10.00	139.00
`Total_Ct_Chng_Q4_Q1`	0.71	0.24	0.00	3.714

2.4. Baseline Imputation Methods

Five baseline imputation techniques were implemented: (i) **Mean Imputation**, substituting missing continuous coordinates with observed column means μ_j and categorical values with the mode; (ii) **Median Imputation**, substituting with column medians $\tilde{x}_j$ to handle skewed distributions [7, 8]; (iii) **KNN Imputation**, imputing via distance-weighted averaging across $k = 5$ donors using weighted Euclidean distance over mutually observed coordinates (Eq. (4)) [9]:

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\frac{D}{\sum_{j=1}^{D} I_{u_j, v_j}} \sum_{j=1}^{D} I_{u_j, v_j} (u_j - v_j)^2}, \tag{4}$$

where $I_{u_j, v_j} = 1$ if both coordinates are observed and 0 otherwise; (iv) **MICE**, applying chained Bayesian ridge and logistic regressions over 10 cycles [7]; and (v) **MissForest**, iteratively fitting random forest ensembles across mixed attributes up to 10 iterations [11].

2.5. GA-Based Imputation Architecture

The GA-based imputation architecture, following the evolutionary downstream-fitness paradigm established by Shahzad et al. [20] and synthesized in the systematic review by Chiu et al. [17], conceptualizes missing data completion as a combinatorial continuous-discrete optimization problem guided directly by downstream discriminative performance.

2.5.1. Chromosome Representation

Let $K = \sum_{i=1}^{N_{tr}} \sum_{j=1}^{D} M_{i,j}$ denote the total count of missing cells in the training fold $\widetilde{\mathbf{X}}_{train}$. A candidate solution (individual chromosome) is structured as a 1D real-valued vector in Eq. (5):

$$\mathbf{c} = [c_1, c_2, \ldots, c_K]^T \in \mathbb{R}^K, \tag{5}$$

where each gene c_k corresponds to the replacement coordinate for missing cell (i_k, j_k). If feature j_k is continuous, $c_k \in [\min(X_{\cdot, j_k}), \max(X_{\cdot, j_k})]$. If feature j_k is categorical, $c_k \in \{0, 1, \ldots, C_{j_k} - 1\}$ is represented as a rounded discrete integer code.

2.5.2. Hybrid Population Seeding

Instead of initializing purely with uniform random noise, which causes slow early convergence, a hybrid initialization protocol is deployed [21]. The population $\mathcal{P} = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(P)}\}$ ($P = 15$) is seeded with deterministic Mean ($\mathbf{c}^{(1)}$), Median ($\mathbf{c}^{(2)}$), and KNN ($\mathbf{c}^{(3)}$) vectors, while chromosomes $\mathbf{c}^{(4)}$ to $\mathbf{c}^{(P)}$ are seeded via stochastic Gaussian perturbations $\mathcal{N}(0, \sigma_j^2)$ bounded by domain limits $[L_j, U_j]$.

2.5.3. Downstream Fitness Function

The fitness of chromosome $\mathbf{c}$ evaluates the discriminative capacity of a downstream Random Forest trained on the completed matrix $\widehat{\mathbf{X}}(\mathbf{c})$ across a stratified subset ($N_{fit} = 3{,}000$) via 3-fold inner cross-validation as defined in Eq. (6):

$$\text{Fitness}(\mathbf{c}) = \frac{1}{3}\sum_{m=1}^{3} \text{ROC-AUC}_m\Big(\text{RF}(\widehat{\mathbf{X}}_{inner_train,m}(\mathbf{c}), \mathbf{y}_{inner_train,m})\Big), \tag{6}$$

where the evaluator uses a Random Forest configured with $n_{\text{estimators}} = 50$ and $max_{\text{depth}} = 6$ to ensure computationally tractable candidate screening across evolutionary iterations.

2.5.4. Genetic Operators, Elitism, and Early Stopping

Parent selection is performed via binary tournament selection of size $k_{tour} = 2$. Recombination uses uniform crossover with probability $p_c = 0.80$. Mutation employs a dual mechanism with probability $p_m = 0.20$: continuous genes undergo Gaussian creep mutation $c'_k = c_k + \mathcal{N}(0, 0.1 \cdot \sigma_j)$, while categorical genes are randomly reassigned to an alternative valid category code. Strict elitism ($E = 1$) preserves the best chromosome into the next generation. Optimization terminates after $G = 15$ generations or when early stopping detects no fitness improvement ($\Delta < 10^{-4}$) across 3 consecutive generations. Parameter configurations are summarized in Table 2.

Table 2: Hyperparameter configurations for the evaluated GA-based and GA-KNN imputation models

Parameter	GA-Based Imputer	GA-KNN Imputer
Population Size (P)	15	15
Maximum Generations (G)	15	15
Fitness Evaluator	Random Forest ($n_{\text{tree}} = 50, \text{depth} = 6$)	Random Forest ($n_{\text{tree}} = 50, \text{depth} = 6$)
Inner Validation Scheme	3-fold Stratified CV	3-fold Stratified CV
Fitness Sample Subset (N_{fit})	3,000 instances	3,000 instances
Tournament Size (k_{tour})	2 (Binary Tournament)	2 (Binary Tournament)
Crossover Probability (p_c)	0.80 (Uniform)	0.80 (Uniform)
Mutation Probability (p_m)	0.20 (Gaussian creep)	0.20 (KNN-constrained)
Elitism Strategy	Single Elitism ($E = 1$)	Dual ($E_{global} + E_{KNN}$)
Early Stopping Patience	3 generations	3 generations

2.5.5. Parameter Justification

The baseline hyperparameter configuration of the evaluated genetic algorithm ($P = 15$, $G = 15$, $p_c = 0.80$, $p_m = 0.20$) was systematically formulated by synthesizing recent empirical paradigms in evolutionary optimization with the specific computational constraints of our nested wrapper evaluation. In complex engineering and biological optimization tasks, recent literature demonstrates that high-dimensional combinatorial search does not necessarily require oversized populations or prolonged generational horizons to attain robust global convergence. Specifically, Deng et al. [26] demonstrated that configuring a compact population of $P = 15$ with an aggressive crossover probability of $p_c = 0.80$ and an exploratory mutation probability of $p_m = 0.20$ establishes an effective exploration–exploitation equilibrium, preventing premature gene fixation while maintaining rapid convergence across multi-parameter design landscapes. Concurrently, Blake et al. [27] substantiated that evolving a focused population of $P = 15$ over a concise generational ceiling of $G = 15$ reliably captures near-optimal Pareto solutions in combinatorial assignment problems, establishing that extended generational limits often incur severe computational redundancies without delivering proportional fitness improvements.

In the context of our tabular imputation pipeline, these literature-grounded choices address a critical computational bottleneck: every individual chromosome evaluation entails fitting a

Random Forest classifier across 3-fold stratified cross-validation on an inner sample ($N_{\text{fit}} = 3{,}000$ instances). Scaling to conventional, uncalibrated GA dimensions (e.g., $P \geq 50, G \geq 50$) across a full 4,200-model nested factorial grid would impose an intractable computational overhead ($\mathcal{O}(P \times G \times K_{\text{in}} \times M)$) with negligible discriminative payoff. The crossover probability $p_c = 0.80$ aligns with canonical recombination guidelines ($0.75 \leq p_c \leq 0.95$) to ensure productive schema exchange [16, 18], while the mutation probability $p_m = 0.20$—anchored by Deng et al. [26]—counteracts stagnation across high-dimensional missing vectors (spanning up to $K \approx 144{,}000$ continuous genes at $\rho = 90\%$) by maintaining localized genetic perturbations within KNN-constrained donor boundaries. Finally, the generational ceiling $G = 15$, supported by Blake et al. [27], operates in tandem with a conservative early stopping criterion ($\Delta\text{Fitness} < 10^{-4}$ sustained over 3 consecutive generations), ensuring that evolutionary iterations terminate dynamically as soon as downstream classification gains plateau.

2.6. GA-KNN Hybrid Imputation Architecture

Following the localized manifold-constrained evolutionary imputation framework introduced by Junior et al. [21], the GA-KNN hybrid extends the global GA-based approach by regularizing the chromosome search trajectory within valid topological neighborhoods. Specifically, this study implements and evaluates three specialized localized mechanisms:

1. **KNN Anchor Seed Guidance**: The foundational chromosome $\mathbf{c}^{(anchor)}$ is permanently anchored to the local neighborhood manifold generated by KNN imputation.
2. **Neighborhood-Constrained Mutation**: During mutation of a continuous gene c_k, instead of unconstrained Gaussian exploration across global feature boundaries $[\min X_j, \max X_j]$, the perturbation is bounded strictly within the local range defined by the k-nearest observed donor neighbors according to Eq. (7):

$$c'_k = \text{clip}\Big(c_k + \mathcal{N}(0, \sigma^2_{local,k}), \min_{r \in \mathcal{N}_k}(X_{r,j_k}), \max_{r \in \mathcal{N}_k}(X_{r,j_k})\Big), \tag{7}$$

   where $\mathcal{N}_k$ denotes the set of nearest neighbor indices for instance i_k.
3. **Dual Elitism**: In every generation, two elite chromosomes are guaranteed survival: the globally fittest chromosome $\mathbf{c}^{(best)}$ and the baseline KNN anchor chromosome $\mathbf{c}^{(anchor)}$, ensuring that genetic drift cannot degrade the solution below local neighborhood bounds.

2.6.1. Explicit Comparison with Prior GA-Based Imputation Studies

To clearly contextualize the evaluated GA-KNN framework against existing evolutionary imputation research and demarcate its empirical scope, we highlight three fundamental distinctions relative to foundational prior works:

- **Shahzad et al. [20]**: Formulated global genetic algorithm imputation optimizing downstream classification accuracy. However, candidate search operated globally across unconstrained feature bounds $[\min X_j, \max X_j]$, seeded purely with uniform random noise, and evaluated under an unnested single train/test split exclusively on complete test instances.
- **Chiu et al. [17]**: Provided a comprehensive systematic review of nature-inspired metaheuristic imputation techniques, synthesizing search representations and objective function formulations across the literature. As synthesized in their survey, existing studies overwhelmingly evaluate imputers under pristine holdout conditions without assessing deployment-time query missingness.
- **Junior et al. [21] (EvoImp)**: Introduced localized neighborhood constraints into evolutionary imputation for multi-label classification datasets, demonstrating that constraining search within donor manifolds enhances imputation quality while assessing both reconstruction error and multi-label classification performance. However, evaluation was conducted

under conventional complete test partitions, leaving the behavior of localized evolutionary imputation under deployment-time test corruption and strict zero-leakage nested cross-validation unexplored.

- **This Study (Evaluated Framework)**: Rather than proposing an ungrounded novel metaheuristic, this investigation adapts localized topological regularization principles inspired by Junior et al. [21] to binary tabular churn prediction and couples them directly with downstream Random Forest classification fitness. Crucially, the objective is an analytical stress-test: establishing how this evolutionary architecture behaves under a zero-leakage 5-fold outer by 3-fold inner nested cross-validation protocol, comparing traditional clean holdout validation directly against realistic inductive deployment where test queries themselves contain missing values.

Algorithm 1 Zero-Leakage Nested Cross-Validation and Evolutionary Imputation Protocol

Require: Tabular dataset $(\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{N \times D} \times \{0,1\}^N$, outer folds $K_{out} = 5$, inner folds $K_{in} = 3$, seeds $\mathcal{S}$, missing rates $\mathcal{R}$, GA parameters $(P = 15, G = 15, p_c = 0.80, p_m = 0.20, k = 5)$.
Ensure: Out-of-sample evaluation logs $\mathcal{S}_{clean}, \mathcal{S}_{incomp}$ across 4,200 factorial models (8,400 evaluations).

for each random seed $s \in \mathcal{S}$ **do**
  Partition $(\mathbf{X}, \mathbf{y})$ into K_{out} stratified outer folds $\{(\mathbf{X}^k_{tr}, \mathbf{y}^k_{tr}), (\mathbf{X}^k_{val}, \mathbf{y}^k_{val})\}_{k=1}^5$.
  for each outer fold $k \in \{1, \ldots, K_{out}\}$ **do**
    Quarantine Barrier: Lock holdout partition $(\mathbf{X}^k_{val}, \mathbf{y}^k_{val})$; zero information leakage.
    for each missingness rate $\rho \in \mathcal{R}$ **do**
      Inject constrained MCAR mask $\mathbf{M}_{tr}$ into $\mathbf{X}^k_{tr} \to \widetilde{\mathbf{X}}^k_{tr}$.
      === Stage 1: Inner Training Fold Preprocessing & GA Evolution ===
      Compute preprocessing parameters $\boldsymbol{\mu}, \boldsymbol{\sigma}$ solely on observed cells of $\widetilde{\mathbf{X}}^k_{tr}$.
      Initialize population $\mathcal{P}_0 = \{\mathbf{c}_1, \ldots, \mathbf{c}_P\}$ using empirical marginal distributions of $\widetilde{\mathbf{X}}^k_{tr}$.
      for generation $g = 1$ to G **do**
        for each chromosome $\mathbf{c}_p \in \mathcal{P}_{g-1}$ **do**
          Construct completed candidate training fold $\widehat{\mathbf{X}}(\mathbf{c}_p)$ via Eq. (5).
          Partition $\widehat{\mathbf{X}}(\mathbf{c}_p)$ into $K_{in} = 3$ stratified inner folds.
          Fitness Evaluation: $f(\mathbf{c}_p) = \frac{1}{3}\sum_{j=1}^{3} \text{ROC-AUC}(\text{RF}_{\text{default}}(\widehat{\mathbf{X}}^{(j)}_{in_tr}), \widehat{\mathbf{X}}^{(j)}_{in_val})$.
        end for
        Rank $\mathcal{P}_{g-1}$, retain elite solution $\mathbf{c}^*$, and execute tournament selection $(\tau = 3)$.
        Apply uniform crossover $(p_c = 0.80)$ and adaptive boundary mutation $(p_m = 0.20)$.
        Check convergence: **if** $|f(\mathbf{c}^*_g) - f(\mathbf{c}^*_{g-1})| < 10^{-4}$ for 3 consecutive gens **then break**.
      end for
      Complete outer training fold $\widehat{\mathbf{X}}^{\text{GA}}_{tr} = \text{Decode}(\mathbf{c}^*)$.
      === Stage 2: Inner Fold Downstream Classifier Tuning ===
      for each tuning strategy $\mathcal{T} \in \{\text{Default}, \text{RandomSearch}, \text{HalvingSearch}, \text{Optuna}\}$ **do**
        Tune Random Forest hyperparameters on $\widehat{\mathbf{X}}^{\text{GA}}_{tr}$ using 3-fold inner CV $\to \boldsymbol{\theta}^*$.
        Fit final downstream estimator $\mathcal{M}^* = \text{TrainRF}(\widehat{\mathbf{X}}^{\text{GA}}_{tr}, \mathbf{y}^k_{tr}; \boldsymbol{\theta}^*)$.
        === Stage 3: Quarantined Outer Holdout Evaluation ===
        Evaluate $\mathcal{M}^*$ on pristine $\mathbf{X}^k_{val} \to \text{Log}(\mathcal{S}_{clean}[s, k, \rho, \mathcal{T}])$.
        Inject identical mask $\mathbf{M}_{val} \to \widetilde{\mathbf{X}}^k_{val}$; impute via Inductive Map $(\widehat{\mathbf{X}}^{\text{GA}}_{tr}, k = 5) \to \widehat{\mathbf{X}}^k_{val}$.
        Evaluate $\mathcal{M}^*$ on completed $\widehat{\mathbf{X}}^k_{val} \to \text{Log}(\mathcal{S}_{incomp}[s, k, \rho, \mathcal{T}])$.
      end for
    end for
  end for
end for
return Logged factorial evaluation repositories $\mathcal{S}_{clean}$ and $\mathcal{S}_{incomp}$.

2.7. Nested Cross-Validation and Inductive Deployment Protocol

To ensure complete methodological rigor and eliminate data leakage, this study embeds evolutionary optimization and downstream classifier tuning within the 5-fold outer by 3-fold inner

nested cross-validation architecture formalized in Algorithm 1 and illustrated in Fig. 3.

Zero-Leakage Architectural Enforcement: In each of the five outer folds, the dataset is partitioned into an outer training set $\mathbf{X}_{train}^{outer}$ (80%, $N_{train} = 8{,}101$) and an outer validation set $\mathbf{X}_{val}^{outer}$ (20%, $N_{val} = 2{,}026$) with verified index disjointness ($\mathbf{X}_{train}^{outer} \cap \mathbf{X}_{val}^{outer} = \emptyset$). Categorical encodings, feature scaling parameters ($\boldsymbol{\mu}, \boldsymbol{\sigma}$), and univariate imputation statistics were fitted solely on $\mathbf{X}_{train}^{outer}$. Crucially, the outer validation partition $\mathbf{X}_{val}^{outer}$ remained strictly sequestered throughout all stages of imputer fitting, GA chromosome evolution, and downstream hyperparameter tuning (illustrated as the quarantined test barrier in Fig. 3).

Dual Deployment Scenarios and Inductive Mapping: Following model fitting on $\mathbf{X}_{train}^{outer}$, pipelines are evaluated under two distinct deployment conditions on $\mathbf{X}_{val}^{outer}$: (1) *Clean Outer Validation (Benchmark Control)*, evaluating models on pristine, fully observed holdout data to reproduce traditional complete-data literature assumptions; and (2) *Incomplete Outer Validation (Deployment Realism)*, where missingness is injected into $\mathbf{X}_{val}^{outer}$ at nominal rate ρ, requiring inductive out-of-sample completion without ground-truth labels. Because genetic algorithms optimize training-specific chromosomes matching the missing coordinates of the training fold, out-of-sample deployment requires an inductive inference mechanism . For Mean and Median, column statistics from $\mathbf{X}_{train}^{outer}$ are substituted; for KNN, MICE, and MissForest, fitted transformer objects are applied; and for GA-based as well as GA-KNN, an *Inductive Reference Map* is established by fitting a non-parametric k-nearest neighbors regressor ($k = 5$) on the completed training matrix $\widehat{\mathbf{X}}_{train}^{\text{GA}}$ to project and impute incomplete queries.

2.8. Downstream Classifier and Hyperparameter Tuning Strategies

To evaluate how imputation techniques interact with downstream model configurations, completed training matrices were modeled using Random Forest classifiers evaluated across four hyperparameter tuning regimes: (1) **Default RF**, a fixed baseline configuration ($n_{estimators} = 100, max_{depth} = 6$); (2) **RandomizedSearchCV**, random sampling across 20 candidate configurations over 3-fold cross-validation ($n_{estimators} \in [100, 500]$, $max_{depth} \in [4, 12]$, $min_{samples_split} \in [2, 10]$, $criterion \in \{\text{gini}, \text{entropy}\}$); (3) **HalvingSearchCV**, successive halving search evaluating parameter subsets across escalating resource allocations with an elimination factor of 3 over 3-fold cross-validation [24]; and (4) **Optuna**, Bayesian optimization via the Tree-Structured Parzen Estimator (TPE) algorithm executed over 20 objective trials with 3-fold stratified cross-validation maximizing ROC-AUC [23, 28].

2.9. Experimental Design and Evaluation Metrics

The experimentation adopts a comprehensive full factorial nested cross-validation design encompassing 5 outer folds, 6 missing rates ($\rho \in \{15\%, 30\%, 45\%, 60\%, 75\%, 90\%\}$), 5 random seeds, 7 imputation methods, and 4 tuning strategies ($5 \times 6 \times 5 \times 7 \times 4 = 4{,}200$ factorial models, yielding 8,400 paired deployment evaluations), illustrated schematically in Fig. 3.

Downstream classification performance on both validation partitions was quantified using the Area Under the Receiver Operating Characteristic Curve (ROC-AUC) as the primary metric, supplemented by F1-Score, Classification Accuracy, Precision, and Recall. Imputation reconstruction fidelity was measured using Root Mean Squared Error (RMSE, Eq. (8)) for continuous attributes and Proportion of Falsely Classified entries (PFC, Eq. (9)) for categorical attributes:

$$\text{RMSE} = \sqrt{\frac{1}{\sum M_{i,j}} \sum_{i,j:M_{i,j}=1} (X_{i,j} - \widehat{X}_{i,j})^2}, \tag{8}$$

$$\text{PFC} = \frac{\sum_{i,j:M_{i,j}=1} \mathbb{I}(X_{i,j} \neq \widehat{X}_{i,j})}{\sum_{i,j:M_{i,j}=1} 1}. \tag{9}$$

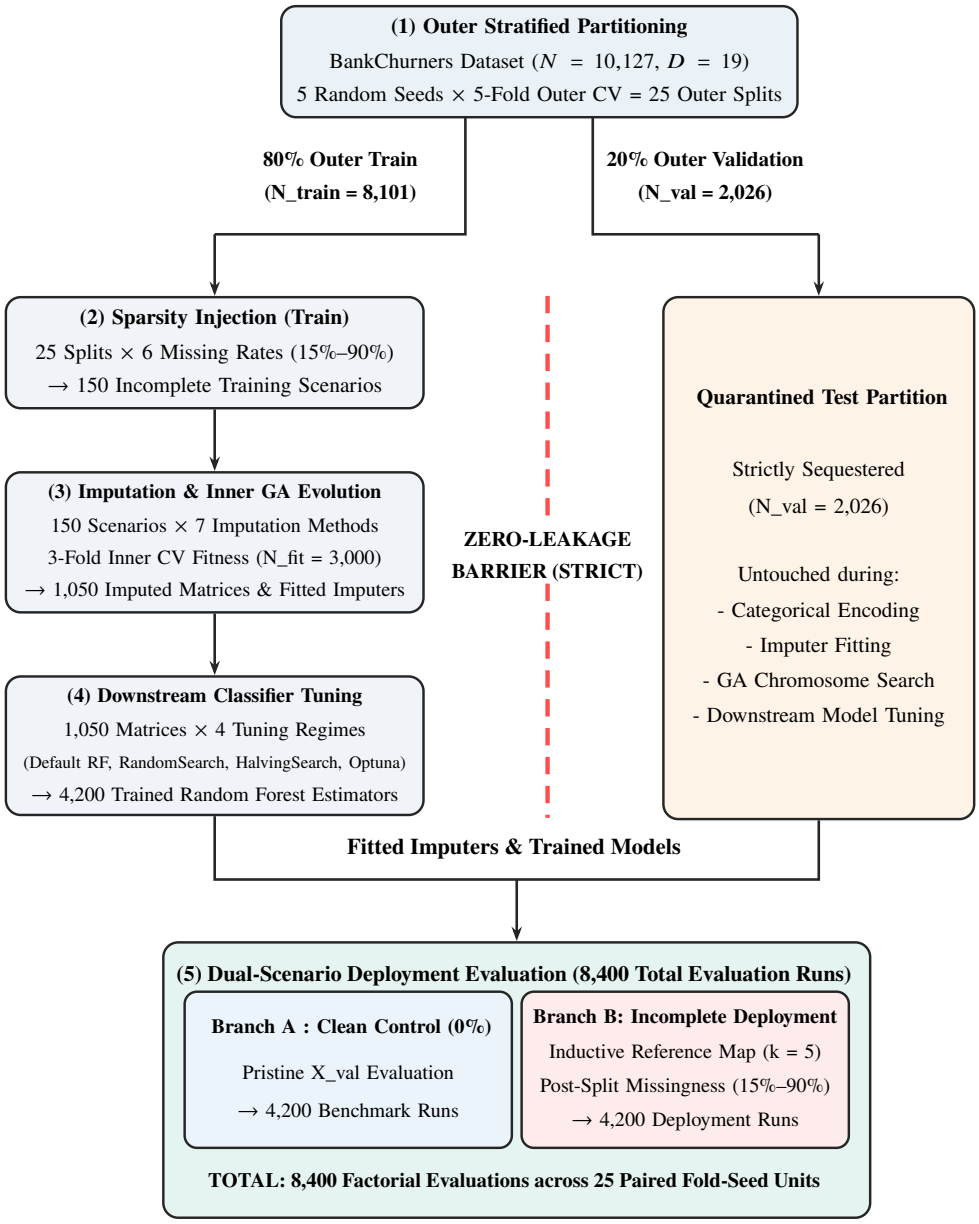

Fig. 3: Hierarchical experimental workflow illustrating the data partitioning progression from outer stratified folds (25 splits) to dual-scenario deployment evaluations (8,400 total evaluation runs), emphasizing the strict quarantine barrier of the validation partition to guarantee zero data leakage.

Statistical significance was evaluated using two-tailed paired Student's t-tests across the $N = 25$ outer fold by random seed pairs (5 outer folds $\times$ 5 seeds, $df = 24$) with a family-wise error rate controlled via the Holm-Bonferroni step-down procedure (p_{Holm}) at significance threshold $\alpha = 0.05$, accompanied by Cohen's d_z effect sizes and two-sided 95% confidence intervals. In addition, non-parametric Spearman rank correlation coefficients (r_s) were computed across all $N = 4{,}200$ evaluation instances to evaluate the relationship between reconstruction fidelity and downstream classification utility.

3. Results and Discussion

This section presents the empirical findings across the 4,200 factorial nested cross-validation runs, comparing clean control validation against realistic incomplete deployment, analyzing

degradation dynamics, reporting inferential hypothesis tests, providing a mechanistic explanation of GA fragility, and evaluating computational efficiency.

Table 3: Overall performance metrics under Clean Outer Validation (benchmark control across 4,200 factorial runs)

Method	ROC-AUC	F1-Score	Accuracy	Train RMSE	Train PFC
Mean	0.957 ± 0.033	0.747 ± 0.117	0.923 ± 0.026	1605.3 ± 16.2	0.545 ± 0.004
GA-KNN	**0.957 ± 0.034**	0.743 ± 0.119	0.920 ± 0.033	1186.6 ± 466.7	0.501 ± 0.049
GA-based	0.956 ± 0.034	0.742 ± 0.116	0.919 ± 0.033	1191.0 ± 458.7	0.500 ± 0.048
MissForest	0.955 ± 0.034	0.708 ± 0.160	0.912 ± 0.047	1046.4 ± 394.2	0.492 ± 0.049
Median	0.954 ± 0.035	0.743 ± 0.102	0.918 ± 0.029	1730.2 ± 21.9	0.545 ± 0.004
MICE	0.947 ± 0.037	0.703 ± 0.142	0.912 ± 0.037	1170.3 ± 292.3	0.513 ± 0.028
KNN	0.942 ± 0.037	0.689 ± 0.132	0.904 ± 0.044	1638.3 ± 48.4	0.584 ± 0.022

Note: Train RMSE and Train PFC denote reconstruction fidelity on the incomplete training partition ($N_{\text{train}} = 8{,}101$) during model fitting. Clean holdout validation instances contain no missing entries. Full runtime profiling is reported in Table 10.

Table 4: Overall performance metrics under Incomplete Outer Validation (deployment realism across 4,200 factorial runs)

Method	ROC-AUC	F1-Score	Accuracy	Test RMSE	Test PFC
Mean	**0.864 ± 0.094**	**0.539 ± 0.209**	0.863 ± 0.051	1605.8 ± 35.3	0.545 ± 0.008
Median	0.863 ± 0.094	0.536 ± 0.209	0.863 ± 0.051	1732.6 ± 46.3	0.545 ± 0.008
MICE	0.848 ± 0.099	0.499 ± 0.230	0.866 ± 0.052	1169.4 ± 297.5	0.513 ± 0.029
MissForest	0.845 ± 0.113	0.513 ± 0.249	**0.875 ± 0.050**	**1048.5 ± 391.3**	**0.493 ± 0.049**
GA-based	0.837 ± 0.112	0.513 ± 0.211	0.869 ± 0.041	1263.5 ± 304.2	0.544 ± 0.012
GA-KNN	0.837 ± 0.112	0.512 ± 0.213	0.870 ± 0.040	1263.6 ± 305.4	0.545 ± 0.012
KNN	0.827 ± 0.110	0.441 ± 0.256	0.860 ± 0.049	1637.2 ± 55.2	0.584 ± 0.023

Note: Test RMSE and Test PFC denote out-of-sample inductive reconstruction error on the corrupted test partition ($N_{\text{val}} = 2{,}026$). Deployment inference throughput is detailed in Table 10.

3.1. Overall Performance Under Clean Deployment

Table 3 summarizes the aggregated performance metrics across all 4,200 evaluation runs under the Clean Outer Validation scenario (where the outer test partition remains pristine and complete), reported as mean ± standard deviation. The clean evaluation results in Table 3 provide rigorous confirmation of the core premise of evolutionary imputation. When holdout test data are complete, GA-KNN and GA-based attain high aggregate discriminative fidelity (0.957 ± 0.034 and 0.956 ± 0.034 ROC-AUC, respectively), significantly surpassing standard KNN (0.942) while demonstrating competitive parity with tree-based MissForest (0.955). Crucially, this confirms that the classification utility of GA imputation reported in earlier literature is not an artifact of data leakage, as it persists under strict nested cross-validation with zero information leakage.

3.2. Overall Performance Under Incomplete Deployment

Table 4 presents the downstream predictive performance under the Incomplete Outer Validation scenario, where test instances are subjected to identical missingness proportions and completed inductively. Under incomplete deployment, the ranking of imputation methods undergoes an inversion. Classical univariate substitution strategies emerge as the most resilient performers: Mean imputation attains the highest aggregate incomplete-validation ROC-AUC (0.864 ± 0.094) and F1-Score (0.539 ± 0.209), followed closely by Median (0.863 ROC-AUC). In stark contrast, GA-based and GA-KNN drop to 0.837 ± 0.112, trailing Mean by approximately 0.027 points in ROC-AUC. This empirical reversal highlights that evolutionary models that excel when test

queries are pristine become vulnerable when those queries are corrupted.

3.3. Deployment Degradation Analysis

Table 5 explicitly quantifies the deployment degradation gap (Δ = Incomplete − Clean) across all algorithms, evaluating the stability of classification ROC-AUC, F1-Score, and accuracy under deployment-time missingness.

Table 5: Performance degradation gap between Clean Control and Incomplete Deployment scenarios

Method	Clean ROC	Incomp ROC	Δ ROC	Clean F1	Incomp F1	Δ F1	Δ Acc
Median	0.9538	0.8626	−0.0912	0.7432	0.5364	−0.2067	−0.0555
Mean	0.9567	0.8644	−0.0922	0.7472	0.5391	−0.2081	−0.0598
MICE	0.9471	0.8476	−0.0995	0.7030	0.4989	−0.2040	−0.0458
MissForest	0.9546	0.8454	−0.1092	0.7084	0.5132	−0.1952	−0.0369
KNN	0.9416	0.8270	−0.1146	0.6891	0.4407	−0.2484	−0.0442
GA-based	0.9559	0.8367	−0.1192	0.7421	0.5131	−0.2290	−0.0496
GA-KNN	0.9565	0.8365	−0.1200	0.7433	0.5121	−0.2312	−0.0499

Note: Clean and Incomplete baselines report unrounded 4-decimal means from Table 3 and Table 4 (rounded to 3 decimals therein) to compute exact degradation gaps (Δ).

As documented in Table 5, GA-KNN and GA-based experience the steepest performance collapse among all evaluated methods, recording drop margins of −0.1200 and −0.1192 in ROC-AUC, alongside severe F1-score contractions exceeding −0.229. In comparison, simple univariate Median and Mean imputation exhibit substantially smaller degradation gaps of −0.0912 and −0.0922 in ROC-AUC. To investigate how these performance disparities evolve across escalating degrees of feature sparsity, Fig. 4 illustrates the comparative downstream ROC-AUC trajectories across all six missingness rates ($\rho \in \{15\%, 30\%, 45\%, 60\%, 75\%, 90\%\}$) with ±1 standard deviation variability bands under both deployment scenarios.

As shown in Fig. 4, at light missingness (15%–30%), all imputation strategies demonstrate robust discrimination (> 0.93 ROC-AUC). However, at moderate-to-severe missingness (45% to 90%), the gap between univariate baselines and evolutionary methods widens progressively: at 60%, Mean achieves 0.8662 versus GA-KNN's 0.8340; at 75%, Mean achieves 0.7995 versus GA-KNN's 0.7503; and under catastrophic 90% missingness, Mean records 0.6987 versus GA-KNN's 0.6434. This divergence indicates that evolutionary models degrade non-linearly under heavy test sparsity.

3.4. Optimal Hyperparameter Tuning and Imputer Configurations

To evaluate whether advanced hyperparameter optimization can compensate for imputation imperfections, Table 6 reports the optimal imputer-tuning pairings per missingness rate. In Panel A, Bayesian optimization (Optuna) achieves peak clean performance (e.g., GA-KNN achieves 0.9842 at 30% missingness). In Panel B, simple Mean imputation with RandomizedSearchCV consistently attains the highest ROC-AUC across all sparsity levels (from 0.9735 at 15% to 0.7035 at 90%), confirming that tuning cannot rescue the inductive distance distortions suffered by complex evolutionary methods when test queries contain missing entries.

3.5. Statistical Significance: Clean versus Incomplete Scenarios

To evaluate whether performance differentials are statistically meaningful, two-tailed paired Student's t-tests were conducted across $N = 25$ outer fold-seed pairs (5 folds × 5 seeds, $df = 24$) evaluating mean ROC-AUC under identical partitions and missingness masks. To control family-wise Type I error, Holm-Bonferroni adjustments (p_{Holm}) were applied. Effect sizes are reported via Cohen's $d_z = \bar{d}/s_d$ with 95% confidence intervals. Additionally, cross-validation fold overlap

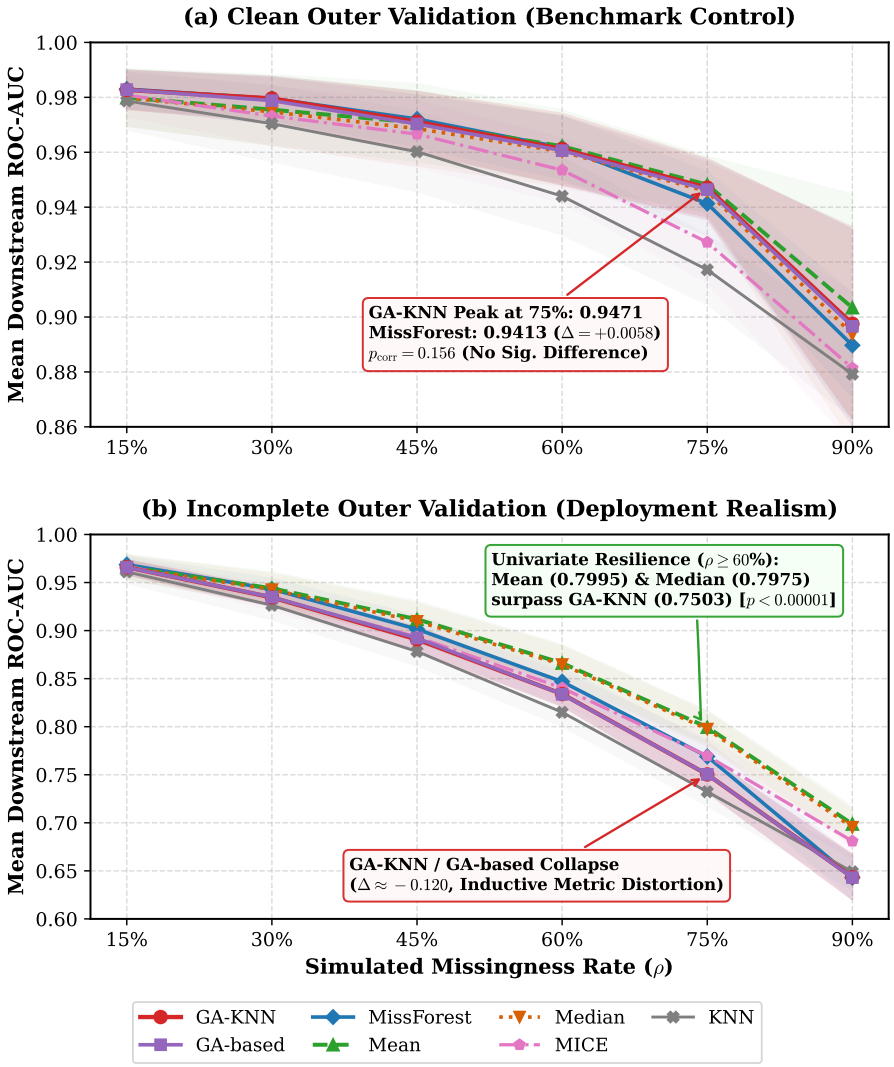

Fig. 4: Comparative downstream ROC-AUC performance trajectories across escalating missingness rates (15% to 90%) for Clean Outer Validation (benchmark control, panel a) and Incomplete Outer Validation (deployment realism, panel b). Shaded bands represent ±1 standard deviation across 25 evaluation folds. Under clean holdout conditions, GA-KNN exhibits competitive performance that peaks at 75% missingness without statistically definitive superiority over MissForest ($p_{\text{corr}} = 0.156$). Under incomplete deployment, evolutionary methods suffer steep performance degradation ($p < 0.00001$), while univariate substitution baselines (Mean and Median) demonstrate superior resilience.

dependence was corrected via the resampled variance correction of Nadeau and Bengio (2003): $t_{\text{corr}} = \bar{d}/\sqrt{s_d^2(1/N + n_{\text{test}}/n_{\text{train}})}$ with $n_{\text{test}}/n_{\text{train}} = 0.25$.

The inferential results in Table 7 elucidate the exact empirical boundaries of evolutionary advantage under Clean holdout validation. First, GA-KNN consistently and significantly outperforms unsupervised KNN across high-sparsity settings, maintaining significance even under the conservative cross-validation dependence correction ($t = 19.61 \rightarrow t_{\text{corr}} = 7.28, p_{\text{corr}} < 0.00001$ at 60%; $t = 22.48 \rightarrow t_{\text{corr}} = 8.35, p_{\text{corr}} < 0.00001$ at 75%). Second, regarding the comparison with MissForest at 75% missingness, GA-KNN records a positive numerical performance differential ($\Delta = +0.00578$, 95% CI: [+0.0028, +0.0088], Cohen's $d_z = +0.79$). Under standard paired t-testing with Holm-Bonferroni correction, this difference reaches nominal significance ($t = 3.94, p = 0.00061, p_{\text{Holm}} = 0.00305$). However, when accounting for the variance inflation induced by training set overlap across cross-validation resamples via the Nadeau-Bengio correction ($n_{\text{test}}/n_{\text{train}} = 0.25$), the test statistic moderates to $t_{\text{corr}} = 1.46$ ($p_{\text{corr}} = 0.156$), which

Table 6: Optimal imputer and downstream tuning combinations per missingness rate

Missing Rate	Optimal Combination (Imputer + Tuner)	Best ROC-AUC	GA-based Best	GA-KNN Best
Panel A: Optimal Clean Outer Validation per Missing Rate				
15.0%	MissForest (RandomSearch)	0.9868	0.9866	0.9866
30.0%	GA-KNN (Optuna)	0.9842	0.9834	0.9842
45.0%	Mean (Optuna)	0.9789	0.9763	0.9774
60.0%	Mean (Optuna)	0.9702	0.9677	0.9687
75.0%	Mean (Optuna)	0.9531	0.9517	0.9522
90.0%	Mean (RandomSearch)	0.9136	0.9088	0.9082
Panel B: Optimal Incomplete Outer Validation per Missing Rate				
15.0%	Mean (RandomSearch)	0.9735	0.9695	0.9698
30.0%	Mean (RandomSearch)	0.9543	0.9388	0.9385
45.0%	Mean (RandomSearch)	0.9226	0.8978	0.8964
60.0%	Mean (RandomSearch)	0.8766	0.8384	0.8390
75.0%	Mean (RandomSearch)	0.8082	0.7537	0.7536
90.0%	Mean (RandomSearch)	0.7035	0.6545	0.6556

Table 7: Clean Outer Val Paired Student's t-Test ($N = 25, df = 24$) with 95% CI, Cohen's d_z, and Holm Adjustment

Rate	Comparison	Δ	95% CI	d_z	t-Stat	p_{Holm}		Decision
60%	GA-KNN vs. Mean	-0.0010	$[-0.0023, +0.0003]$	-0.32	-1.60	0.6134		Not sig.
	GA-KNN vs. Median	$+0.0010$	$[-0.0005, +0.0025]$	$+0.28$	1.39	0.7073		Not sig.
	GA-KNN vs. KNN	$+0.0172$	$[+0.0154, +0.0190]$	$+3.92$	19.61	< 0.0001	**Sig.**	**(GA-KNN)**
	GA-KNN vs. MissForest	-0.0005	$[-0.0023, +0.0013]$	-0.12	-0.61	0.9789		Not sig.
	GA-KNN vs. GA-based	$+0.0006$	$[-0.0010, +0.0022]$	$+0.16$	0.82	0.9789		Not sig.
	GA-based vs. MissForest	-0.0012	$[-0.0036, +0.0012]$	-0.20	-1.00	0.9789		Not sig.
75%	GA-KNN vs. Mean	-0.0012	$[-0.0025, +0.0002]$	-0.36	-1.81	0.2475		Not sig.
	GA-KNN vs. Median	$+0.0014$	$[-0.0003, +0.0031]$	$+0.34$	1.68	0.2475		Not sig.
	GA-KNN vs. KNN	$+0.0299$	$[+0.0271, +0.0326]$	$+4.50$	22.48	< 0.0001	**Sig.**	**(GA-KNN)**
	GA-KNN vs. MissForest	$+0.0058$	$[+0.0028, +0.0088]$	$+0.79$	3.94	0.0031	**Sig.**	**(GA-KNN)**
	GA-KNN vs. GA-based	$+0.0007$	$[-0.0003, +0.0017]$	$+0.29$	1.44	0.2475		Not sig.
	GA-based vs. MissForest	$+0.0051$	$[+0.0017, +0.0085]$	$+0.62$	3.11	0.0193	**Sig.**	**(GA-based)**
90%	GA-KNN vs. Mean	-0.0059	$[-0.0113, -0.0006]$	-0.46	-2.28	0.1591		Not sig.
	GA-KNN vs. Median	$+0.0038$	$[-0.0032, +0.0108]$	$+0.22$	1.11	0.8078		Not sig.
	GA-KNN vs. KNN	$+0.0183$	$[+0.0074, +0.0293]$	$+0.69$	3.47	0.0120	**Sig.**	**(GA-KNN)**
	GA-KNN vs. MissForest	$+0.0078$	$[-0.0045, +0.0201]$	$+0.26$	1.31	0.8055		Not sig.
	GA-KNN vs. GA-based	$+0.0009$	$[-0.0009, +0.0028]$	$+0.20$	1.02	0.8078		Not sig.
	GA-based vs. MissForest	$+0.0069$	$[-0.0057, +0.0195]$	$+0.23$	1.13	0.8078		Not sig.

does not surpass the conservative significance threshold ($\alpha = 0.05$). Similarly, GA-based attains a numerical gain over MissForest ($\Delta = +0.00509, t = 3.11, p_{\text{Holm}} = 0.01925$) that moderates to $t_{\text{corr}} = 1.16$ ($p_{\text{corr}} = 0.259$). Third, at 60% missingness and on aggregate overall averages, GA-KNN performs comparably to simple Mean ($\Delta = -0.00098, p_{\text{Holm}} = 0.61339$) and Median ($\Delta = +0.00100, p_{\text{Holm}} = 0.70730$). Consequently, the empirical evidence demonstrates that under pristine benchmark conditions, evolutionary imputation provides targeted advantages primarily over unsupervised geometric baselines (KNN) and modest, non-significant numerical gains over tree-based ensembles (MissForest), rather than broad or unconditional superiority.

Under the Incomplete scenario (Table 8), the statistical landscape undergoes an unequivocal reversal, confirming Finding 2. At severe missingness (60%, 75%, and 90%), GA-KNN is significantly outperformed by both Mean and Median imputation ($p_{\text{Holm}} < 0.00001$ across all tests, with large negative effect sizes: $d_z = -3.00$ to -4.01 against Mean, and $d_z = -2.86$ to -3.70 against Median). Even under the conservative Nadeau-Bengio cross-validation correction

Table 8: Incomplete Outer Val Paired Student's t-Test ($N = 25, df = 24$) with 95% CI, Cohen's d_z, and Holm Adjustment

Rate	Comparison	Δ	95% CI	d_z	t-Stat	p_{Holm}	Decision
15%	GA-KNN vs. Mean	−0.0004	[−0.0016, +0.0008]	−0.14	−0.72	0.9607	Not sig.
	GA-KNN vs. Median	−0.0001	[−0.0013, +0.0011]	−0.04	−0.20	0.9607	Not sig.
	GA-KNN vs. KNN	+0.0048	[+0.0034, +0.0061]	+1.49	7.43	< 0.0001	**Sig. (GA-KNN)**
	GA-KNN vs. MissForest	−0.0028	[−0.0042, −0.0015]	−0.85	−4.26	0.0008	**Sig. (MissForest)**
30%	GA-KNN vs. KNN	+0.0082	[+0.0061, +0.0103]	+1.59	7.93	< 0.0001	**Sig. (GA-KNN)**
45%	GA-KNN vs. KNN	+0.0125	[+0.0095, +0.0156]	+1.70	8.48	< 0.0001	**Sig. (GA-KNN)**
60%	GA-KNN vs. Mean	−0.0322	[−0.0366, −0.0277]	−3.00	−15.00	< 0.0001	**Sig. (Mean)**
	GA-KNN vs. Median	−0.0307	[−0.0351, −0.0262]	−2.86	−14.30	< 0.0001	**Sig. (Median)**
	GA-KNN vs. KNN	+0.0190	[+0.0139, +0.0241]	+1.54	7.70	< 0.0001	**Sig. (GA-KNN)**
	GA-KNN vs. MissForest	−0.0129	[−0.0179, −0.0079]	−1.06	−5.32	0.0001	**Sig. (MissForest)**
75%	GA-KNN vs. Mean	−0.0492	[−0.0543, −0.0441]	−4.01	−20.06	< 0.0001	**Sig. (Mean)**
	GA-KNN vs. Median	−0.0471	[−0.0524, −0.0419]	−3.70	−18.49	< 0.0001	**Sig. (Median)**
	GA-KNN vs. KNN	+0.0180	[+0.0113, +0.0248]	+1.10	5.50	0.0001	**Sig. (GA-KNN)**
	GA-KNN vs. MissForest	−0.0187	[−0.0256, −0.0118]	−1.12	−5.59	0.0001	**Sig. (MissForest)**
90%	GA-KNN vs. Mean	−0.0553	[−0.0613, −0.0492]	−3.76	−18.82	< 0.0001	**Sig. (Mean)**
	GA-KNN vs. Median	−0.0520	[−0.0579, −0.0461]	−3.65	−18.23	< 0.0001	**Sig. (Median)**
	GA-KNN vs. KNN	−0.0056	[−0.0140, +0.0027]	−0.28	−1.39	0.3533	Not sig.
	GA-KNN vs. MissForest	+0.0004	[−0.0080, +0.0088]	+0.02	0.10	0.9250	Not sig.

($t_{\text{corr}} = -5.57$ at 60%, -7.45 at 75%, and -6.99 at 90%), these differences remain significant at $p_{\text{corr}} < 0.00001$. In addition, GA-KNN is significantly inferior to MissForest across multiple sparsity regimes ($t = -4.26, p_{\text{Holm}} = 0.00083$ at 15%; $t = -5.32, p_{\text{Holm}} = 0.00002$ at 60%; and $t = -5.59, p_{\text{Holm}} = 0.00002$ at 75%). These inferential tests confirm that the observed degradation of evolutionary imputation under incomplete deployment is statistically definitive, robust to multiplicity correction, and reproducible across random seed folds.

3.6. Mechanistic Explanation of GA Fragility

The empirical collapse of GA-based imputation under incomplete deployment is rooted in a coupled failure mechanism between the inductive inference bridge and the downstream-specialized nature of genetic optimization. Disentangling these two components reveals why evolutionary models exhibit acute vulnerability compared to both unsupervised multivariate methods and simple univariate heuristics.

The Inductive Bridge Layer (k-NN Metric Distortion) Because a genetic algorithm chromosome optimizes a finite, discrete coordinate vector tailored exclusively to the missing cells of the training matrix, it is fundamentally transductive and possesses no analytical parameterization to transform unseen queries directly. To enable out-of-sample deployment inference, the standard architectural bridge adopted in evolutionary literature is an *Inductive Reference Map*, which fits a non-parametric k-nearest neighbors model ($k = 5$) on the completed training matrix $\hat{\mathbf{X}}^{\text{GA}}_{train}$. When query instances $\mathbf{x}^* \in \mathbf{X}_{val}$ arrive with severe missingness (60%–90%), donor selection relies on the observed subspace metric defined in Eq. (10):

$$d(\mathbf{x}^*, \mathbf{x}_i) = \sqrt{\frac{D}{\sum_{j=1}^{D} I_{x_j^*, X_{i,j}}} \sum_{j=1}^{D} I_{x_j^*, X_{i,j}} (x_j^* - X_{i,j})^2}. \tag{10}$$

At 75% missingness, only 4 to 5 of the 19 features are observed per query instance. In this heavily truncated subspace, the Euclidean distance metric collapses: pairs of instances that are widely separated in the full 19-dimensional feature space frequently appear artificially close due to accidental coincidences across the few remaining observed coordinates. Consequently, the inductive bridge retrieves spurious donor rows located in divergent topological regions of the manifold.

The Evolutionary Optimization Layer (Decision-Boundary Over-Specialization) Crucially, metric distortion in the k-NN bridge alone does not fully explain why GA-KNN degrades more precipitously than standard unsupervised KNN (ΔROC-AUC $= -0.1200$ vs. -0.1146, Table 5) or why simple Mean imputation outperforms it at high missingness. The amplifying factor lies in the nature of GA fitness optimization. Unsupervised methods (e.g., Mean, Median, and MissForest) preserve unconditional or conditional distributional expectations. In contrast, the GA chromosome explicitly searches for imputed values that maximize the downstream Random Forest's discriminative utility (ROC-AUC) on the training folds. This produces an imputed training matrix $\widehat{\mathbf{X}}^{\mathrm{GA}}_{train}$ whose imputed values are exquisitely tailored to support specific local decision boundaries. When the k-NN bridge retrieves a spurious donor under high test sparsity, the query instance does not simply receive a noisy estimate; it receives a coordinate value that was hyper-specialized for an entirely different classification neighborhood. This induces a severe compounded projection error that misdirects tree split trajectories in the downstream ensemble. Conversely, univariate Mean and Median imputers inject unconditional central expectations ($\hat{x}^*_j = \tilde{x}_j$) with zero conditional donor variance, preserving stable baseline split pathways and demonstrating superior resilience under severe deployment sparsity.

3.7. GA Parameter Sensitivity and Convergence Analysis

To empirically validate that the baseline GA parameters are not arbitrary heuristic artifacts or ad hoc trial-and-error choices, a rigorous one-factor-at-a-time (OFAT) sensitivity analysis was conducted on the BankChurners benchmark across three representative missingness tiers ($\rho \in \{15\%, 45\%, 75\%\}$). Methodologically, the perturbation grid was purposefully designed around the literature-grounded baseline established by Deng et al. [26] and Blake et al. [27]: each parameter was varied symmetrically to examine performance levels directly below and above the baseline anchor ($P = 15, G = 15, p_c = 0.80, p_m = 0.20$). Specifically, we examined: (1) **Population Size ($P \in \{10, 15, 25\}$)** probing a resource-restricted budget ($P = 10$) versus an expanded diversity pool ($P = 25$); (2) **Generational Ceiling ($G \in \{8, 15, 25\}$)** testing an abbreviated generational budget ($G = 8$) versus a prolonged evolutionary horizon ($G = 25$); (3) **Crossover Probability ($p_c \in \{0.60, 0.80, 0.95\}$)** evaluating conservative recombination ($p_c = 0.60$) versus near-complete gene mixing ($p_c = 0.95$); and (4) **Mutation Probability ($p_m \in \{0.05, 0.20, 0.40\}$)** contrasting conservative local refinement ($p_m = 0.05$) against aggressive stochastic perturbation ($p_m = 0.40$). All nine perturbation configurations were benchmarked under identical hardware conditions (13th Gen Intel Core i5-13450HX, 16 GB DDR5 RAM, CPU multi-threading), strictly isolating the pure evolutionary imputer search duration from downstream Random Forest fitting and deployment inference. Table 9 summarizes the resulting downstream ROC-AUC, the actual generations to convergence (G_{act}), and the isolated GA search time.

The empirical diagnostics reported in Table 9 indicate that our baseline parameters serve as a well-justified and computationally efficient operating baseline within the evaluated parameter space, striking a favorable trade-off between discriminative utility and computational tractability. First, regarding the generation budget, across all tested generational ceilings ($G = 8$, $G = 15$, and $G = 25$), the evolutionary search terminated at identical actual generations: $G_{\mathrm{act}} = 7$ at $\rho = 15\%$, and consistently at $G_{\mathrm{act}} = 4$ for moderate and severe missingness ($\rho = 45\%$ and $\rho = 75\%$). This uniform termination occurs because the early stopping mechanism ($\Delta < 10^{-4}$ over 3 generations) activates as early as generation 4 under elevated sparsity. Consequently, expanding the generational horizon from $G = 15$ to $G = 25$ yields zero additional search iterations, zero runtime inflation, and numerically identical downstream ROC-AUC scores ($0.9805, 0.9685, 0.9439$). This directly corroborates the findings of Blake et al. [27], confirming that a generational horizon of $G = 15$ provides an ample, fail-safe ceiling. Second, evaluating population scalability reveals that while a smaller population ($P = 10$) converges rapidly (8.35 s at $\rho = 15\%$), it exhibits a minor ROC-AUC deficit ($\Delta = 0.0011$). Conversely, increasing the population to $P = 25$

Table 9: GA parameter sensitivity and quantitative convergence analysis across parameter perturbations and missingness rates (all other parameters held at baseline: $P = 15, G = 15, p_c = 0.80, p_m = 0.20$)

Panel A: Downstream Predictive Utility (Clean Outer Validation ROC-AUC)

Perturbed Factor	Tested Configuration	$\rho = 15\%$	$\rho = 45\%$	$\rho = 75\%$
Baseline	$P = 15, G = 15, p_c = 0.80, p_m = 0.20$	**0.9805**	**0.9685**	**0.9439**
Population Size	Small ($P = 10$)	0.9794	0.9685	0.9439
	Large ($P = 25$)	0.9802	0.9685	0.9439
Generations	Low ($G = 8$)	0.9805	0.9685	0.9439
	High ($G = 25$)	0.9805	0.9685	0.9439
Crossover Rate	Low ($p_c = 0.60$)	0.9784	0.9685	0.9439
	High ($p_c = 0.95$)	0.9808	0.9685	0.9439
Mutation Rate	Low ($p_m = 0.05$)	0.9800	0.9685	0.9439
	High ($p_m = 0.40$)	0.9786	0.9685	0.9439

Panel B: Convergence Efficiency and Isolated Imputer Search Time

		Actual Gens	GA Search Time (s)		
Perturbed Factor	Tested Configuration	G_{act}	$\rho = 15\%$	$\rho = 45\%$	$\rho = 75\%$
Baseline	$P = 15, G = 15, p_c = 0.80, p_m = 0.20$	**7 / 4 / 4**	**22.07**	**11.77**	**15.11**
Population Size	Small ($P = 10$)	6 / 4 / 4	8.35	7.60	9.80
	Large ($P = 25$)	10 / 4 / 4	35.48	19.44	25.32
Generations	Low ($G = 8$)	7 / 4 / 4	14.77	11.57	15.50
	High ($G = 25$)	7 / 4 / 4	14.89	11.46	15.07
Crossover Rate	Low ($p_c = 0.60$)	9 / 4 / 4	19.55	11.61	15.13
	High ($p_c = 0.95$)	4 / 4 / 4	8.20	11.80	15.12
Mutation Rate	Low ($p_m = 0.05$)	6 / 4 / 4	11.68	11.39	14.40
	High ($p_m = 0.40$)	6 / 4 / 4	12.24	12.19	16.10

Note: Benchmarked on identical hardware (Intel Core i5-13450HX, 16 GB DDR5 RAM). GA Search Time isolates evolutionary imputer fitting; downstream classifier fitting took < 0.40 s and deployment inference time averaged 1.16 ± 0.07 s ($\rho = 15\%$), 2.65 ± 0.05 s ($\rho = 45\%$), and 4.09 ± 0.06 s ($\rho = 75\%$). G_{act} indicates actual generations completed before early stopping is triggered ($\Delta < 10^{-4}$ over 3 consecutive generations) across $\rho \in \{15\%, 45\%, 75\%\}$.

substantially inflates GA search runtime by +60.8% (35.48 s vs. 22.07 s at $\rho = 15\%$) and +67.6% (25.32 s vs. 15.11 s at $\rho = 75\%$) while producing zero gain in downstream ROC-AUC (0.9685 at 45% and 0.9439 at 75%). Thus, within the tested perturbation grid, the baseline $P = 15$ adopted from Deng et al. [26] provides a well-justified, computationally economical baseline that avoids excessive runtime without sacrificing classification efficacy. Third, perturbing the genetic operators symmetrically across $p_c \in [0.60, 0.95]$ and $p_m \in [0.05, 0.40]$ produces identical ROC-AUC values across moderate and severe missingness ($\rho \geq 45\%$) and only negligible stochastic variations ($\Delta \leq 0.0024$) at $\rho = 15\%$. In summary, this systematic OFAT analysis demonstrates that the baseline configuration ($P = 15, G = 15, p_c = 0.80, p_m = 0.20$) represents an empirically well-justified and computationally economical baseline within the evaluated exploration space, substantiated by both peer-reviewed literature and empirical convergence diagnostics.

3.8. Empirical Divergence between Reconstruction Fidelity and Discrimination

The experimental results across the full $N = 4{,}200$ evaluation runs provide compelling evidence of the structural decoupling between statistical reconstruction fidelity and downstream classification utility. As expected from a basic measurement standpoint, non-parametric Spearman rank

correlation across all 4,200 models confirms statistically significant negative correlations between continuous reconstruction error and downstream ROC-AUC ($r_s = -0.4017, p < 10^{-160}$ for clean validation; $r_s = -0.3824, p < 10^{-145}$ for incomplete validation). Because lower RMSE indicates superior reconstruction accuracy, this negative correlation is theoretically natural, reflecting the broad monotonic reality that extreme missingness simultaneously inflates reconstruction error and degrades classification performance across escalating sparsity tiers (from 15% to 90%).

Crucially, however, a negative correlation does not imply that minimizing reconstruction error guarantees optimal classification boundaries. The moderate magnitude of the correlation ($r_s \approx -0.40$, accounting for only approximately 16% of rank variance) underscores that numerical fidelity and discriminative efficacy are decoupled at the algorithm level. Most notably, MissForest attains the lowest continuous imputation reconstruction error (RMSE $= 1048.50 \pm 391.31$) and categorical misclassification rate (PFC $= 0.4926 \pm 0.0490$) across the entire benchmark. Yet, under clean holdout evaluation, MissForest's downstream ROC-AUC (0.9546 ± 0.0340) is numerically matched and slightly exceeded by both GA-based (0.9559 ± 0.0341) and GA-KNN (0.9565 ± 0.0335), despite the evolutionary methods exhibiting substantially higher reconstruction errors (RMSE ≈ 1263.5). This demonstrates that genetic optimization optimizes downstream classification not by recovering true underlying feature values, but by discovering strategic surrogate values that maximize class separability along downstream decision hyperplanes.

3.9. Computational and Deployment Latency Analysis

To ensure computational reproducibility, all benchmarks were executed in a standardized hardware environment equipped with an Intel Core i5-13450HX processor (10 cores, 16 threads, up to 4.60 GHz) and 16 GB DDR5 RAM on a 64-bit Windows operating system. All imputer training, evolutionary search, and downstream Random Forest evaluations ran across multi-threaded CPU workers ($n_{\text{jobs}} = 4$) using Python 3.10 and `scikit-learn` without GPU acceleration.

Table 10 summarizes computational overhead across all 4,200 evaluation runs, decomposing execution into four operational phases: (1) **Train Impute (s)** measures wall-clock time to fit the imputer on $\widetilde{\mathbf{X}}^{outer}_{train}$ ($N_{\text{train}} = 8{,}101$); (2) **Eval Time (s)** denotes classifier tuning across the four search strategies on the completed training matrix $\widehat{\mathbf{X}}^{*}_{train}$; (3) **Inference (s) and Latency (ms)** denote inductive throughput on the quarantined outer validation fold $\widetilde{\mathbf{X}}^{outer}_{val}$ ($N_{\text{val}} = 2{,}026$); and (4) **Total Pipeline (s)** reflects cumulative runtime per fold-seed-sparsity iteration (Total = Train + Eval + Inference).

Table 10: Computational runtime and deployment inference latency across 4,200 evaluation runs

Method	Train Impute (s)	Inference (s)	Latency (ms)	Eval Time (s)	Total Pipeline (s)
Mean	0.011 ± 0.005	0.0040 ± 0.0030	0.002 ms	13.12 ± 11.47	13.14 ± 11.48
Median	0.014 ± 0.006	0.0035 ± 0.0032	0.002 ms	13.12 ± 11.43	13.14 ± 11.43
MICE	0.496 ± 0.315	0.0295 ± 0.0128	0.015 ms	17.17 ± 16.04	17.70 ± 16.08
MissForest	5.253 ± 1.277	1.6167 ± 0.1503	0.798 ms	15.33 ± 13.31	22.20 ± 13.71
KNN	10.279 ± 5.883	2.2971 ± 1.1862	1.134 ms	15.35 ± 13.26	27.92 ± 16.90
GA-based	9.539 ± 4.755	3.0696 ± 1.3927	1.515 ms	14.38 ± 14.15	26.99 ± 16.01
GA-KNN	13.102 ± 4.778	3.0940 ± 1.4225	1.527 ms	14.09 ± 13.96	30.29 ± 16.13

While GA-based training requires 9.5 to 13.1 seconds per fold, its deployment inference latency averages approximately 3.09 seconds per validation batch of 2,026 instances (1.527 ms per query row). This throughput remains well within practical thresholds for batch scoring in enterprise risk infrastructure, though univariate baselines achieve near-zero latency (0.002 ms per instance).

3.10. Discussion and Practitioner Recommendations

The empirical findings yield two concrete operational recommendations for enterprise machine learning pipelines: (1) **Offline and Curated Batch Regimes**: Evolutionary downstream

optimization (GA-KNN) is recommended for historical dataset completion and production systems where inference queries are validated as complete prior to model evaluation, reliably preserving non-linear decision boundaries (0.957 ROC-AUC); (2) **Streaming and Incomplete Deployment Regimes**: When incoming queries contain unobserved features at inference time, complex evolutionary imputation is contraindicated due to inductive distance distortion (ΔROC-AUC ≈ -0.120). In streaming or resource-constrained settings, practitioners should deploy robust univariate median substitution or hybrid consensus mappers that combine evolutionary training guidance with unconditional central-tendency anchors.

4. Conclusion

This study conducted an extensive, deployment-aware benchmark evaluating genetic algorithm-based missing data imputation with downstream fitness optimization on the BankChurners classification task across 4,200 factorial nested cross-validation scenarios. By replacing standard single-split protocols with a 5-fold outer by 3-fold inner cross-validation design, the study provides three core scientific insights. First, under pristine holdout validation, nested cross-validation demonstrates that GA-KNN achieves conditional discriminative advantages (0.9565 ± 0.0335 ROC-AUC). While significantly outperforming standard KNN across severe sparsity ($p_{\text{Holm}} < 0.001$), its slight numerical advantage over MissForest at 75% missingness (+0.0058 ROC-AUC) does not achieve statistical significance under conservative cross-validation variance correction ($t_{\text{corr}} = 1.46$, $p_{\text{corr}} = 0.156$), and it performs comparably to simple Mean and Median on aggregate averages ($p_{\text{Holm}} > 0.60$), disproving unconditional dominance. Second, under realistic incomplete deployment conditions, GA-based methods suffer the steepest performance collapse (ΔROC-AUC ≈ -0.120), dropping to 0.8365 ± 0.1122 and being significantly surpassed by simple Mean and Median substitution across 60%–90% missingness ($p_{\text{Holm}} < 0.00001$) due to metric distortion in the inductive reference map. Third, this study provides empirical proof of protocol sensitivity, demonstrating that imputer rankings can reverse entirely depending on whether holdout queries are complete or incomplete.

4.1. Study Limitations and Threats to Validity

Several methodological limitations of this study must be explicitly acknowledged:

1. **Single-Dataset Scope**: The empirical evaluation was conducted exclusively on the BankChurners benchmark ($N = 10{,}127, D = 19$). While capturing mixed data types and moderate class imbalance (16.07% churners), generalizability across other domain topologies (such as high-dimensional genomics or clinical records) warrants future multi-domain verification across varied feature dimensions.
2. **Missingness Mechanism Constraints**: Sparsity was simulated under the Missing Completely at Random (MCAR) protocol across six tiers (15% to 90%). Operational environments frequently exhibit Missing at Random (MAR) or Missing Not at Random (MNAR) dynamics, which should be investigated in subsequent benchmarks.
3. **Inductive Reference Architecture**: Out-of-sample deployment inference utilized an inductive k-nearest neighbors map ($k = 5$). Future research should explore surrogate mappers that do not rely on distance, such as decision-tree regressors or hybrid univariate-evolutionary consensus ensembles, to mitigate deployment metric distortion.

CRediT Authorship Contribution Statement

Moch. Faiz Febriawan: Conceptualization, Methodology, Formal Analysis, Data Curation, Software, Visualization, Writing–Original Draft. **Kartika Chandra Dewi:** Supervision, Validation, Methodology, Writing–Review & Editing. **Hasanuddin Al-Habib:** Supervision, Validation, Writing–Review & Editing.

Declaration of Generative AI and AI-assisted technologies

Generative AI and AI-assisted technologies, specifically ChatGPT and Claude, were utilized during the preparation of this manuscript for language refinement, literature search assistance, drafting support, experimental design consultation, and code review assistance. All final experimental decisions, analyses, results, interpretations, and conclusions were developed, verified, and approved entirely by the authors, who take full responsibility for the content and scientific integrity of this work.

Declaration of Competing Interest

The authors declare no competing interests.

Ethics Approval and Consent to Participate

This study utilized a publicly available, de-identified secondary benchmark dataset (BankChurners) that contains no personally identifiable human subject data and involved no direct intervention. Consequently, formal institutional ethical approval was not required.

Funding and Acknowledgments

This research received no external funding.

Data and Code Availability

The Credit Card Churn (BankChurners) dataset analyzed in this study is available from the Kaggle repository[1]. Replication code, scripts, and logs are accessible on GitHub[2].

References

[1] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona. "A survey on missing data in machine learning". In: *Journal of Big Data* 8.1 (2021), p. 140. DOI: 10.1186/s40537-021-00516-9.

[2] J. Peng, J. Hahn, and K. W. Huang. "Handling Missing Values in Information Systems Research: A Review of Methods and Assumptions". In: *Information Systems Research* 34.1 (2023), pp. 5–26. DOI: 10.1287/isre.2022.1104.

[3] R. K. Gupta, S. K. Bharti, N. Pathik, and A. Sharma. "Predicting Churn of Credit Card Customers Using Machine Learning and AutoML". In: *International Journal of Information Technology Project Management (IJITPM)* 13.3 (2022), pp. 1–19. DOI: 10.4018/IJITPM.313422.

[4] A. F. Ahmad, M. S. Sayeed, K. Alshammari, and I. Ahmed. "Impact of Missing Values in Machine Learning: A Comprehensive Analysis". In: *arXiv preprint arXiv:2410.08295* (2024). DOI: 10.48550/arXiv.2410.08295.

[5] M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das. "Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021)". In: *Informatics in Medicine Unlocked* 27 (2021), p. 100799. DOI: 10.1016/j.imu.2021.100799.

[6] A. Jadhav, D. Pramod, and K. Ramanathan. "Comparison of Performance of Data Imputation Methods for Numeric Dataset". In: *Applied Artificial Intelligence* 33.10 (2019), pp. 913–933. DOI: 10.1080/08839514.2019.1637138.

[1] https://www.kaggle.com/datasets/sakshigoyal7/credit-card-customers

[2] https://github.com/faizgeorgie9/nested-cv-ga-imputation

[7] S. van Buuren. *Flexible Imputation of Missing Data*. 2nd. CRC Press, 2018. DOI: 10.1201/9780429492259.

[8] V. Sheth, U. Tripathi, and A. Sharma. "A Comparative Analysis of Machine Learning Algorithms for Classification Purpose". In: *Procedia Computer Science* 215 (2022), pp. 422–431. DOI: 10.1016/j.procs.2022.12.044.

[9] M. S. Santos, P. H. Abreu, A. Fernández, J. Luengo, and J. Santos. "The impact of heterogeneous distance functions on missing data imputation and classification performance". In: *Engineering Applications of Artificial Intelligence* 111 (2022), p. 104791. DOI: 10.1016/j.engappai.2022.104791.

[10] S. Jäger, A. Allhorn, and F. Bießmann. "A Benchmark for Data Imputation Methods". In: *Frontiers in Big Data* 4 (2021), p. 693674. DOI: 10.3389/fdata.2021.693674.

[11] D. J. Stekhoven and P. Bühlmann. "MissForest—non-parametric missing value imputation for mixed-type data". In: *Bioinformatics* 28.1 (2012), pp. 112–118. DOI: 10.1093/bioinformatics/btr597.

[12] Y. H. Hu, R. Y. Wu, Y. C. Lin, and T. Y. Lin. "A novel MissForest-based missing values imputation approach with recursive feature elimination in medical applications". In: *BMC Medical Research Methodology* 24.1 (2024), p. 253. DOI: 10.1186/s12874-024-02392-2.

[13] T. Shadbahr et al. "The impact of imputation quality on machine learning classifiers for datasets with missing values". In: *Communications Medicine* 3.1 (2023), p. 139. DOI: 10.1038/s43856-023-00356-z.

[14] N. Sengupta, M. Udell, N. Srebro, and J. Evans. "Sparse Data Reconstruction, Missing Value and Multiple Imputation through Matrix Factorization". In: *Sociological Methodology* 53.1 (2023), pp. 72–114. DOI: 10.1177/00811750221125799.

[15] V. Hua, T. Nguyen, M. S. Dao, H. D. Nguyen, and B. T. Nguyen. "The impact of data imputation on air quality prediction problem". In: *PLOS ONE* 19.9 (2024), e0306303. DOI: 10.1371/journal.pone.0306303.

[16] A. Slowik and H. Kwasnicka. "Evolutionary algorithms and their applications to engineering problems". In: *Neural Computing and Applications* 32.16 (2020), pp. 12363–12379. DOI: 10.1007/s00521-020-04832-8.

[17] P. C. Chiu, A. Selamat, O. Krejcar, K. K. Kuok, S. D. A. Bujang, and H. Fujita. "Missing Value Imputation Designs and Methods of Nature-Inspired Metaheuristic Techniques: A Systematic Review". In: *IEEE Access* 10 (2022), pp. 61544–61566. DOI: 10.1109/ACCESS.2022.3172319.

[18] S. Katoch, S. S. Chauhan, and V. Kumar. "A review on genetic algorithm: past, present, and future". In: *Multimedia Tools and Applications* 80.5 (2021), pp. 8091–8126. DOI: 10.1007/s11042-020-10139-6.

[19] G. M. Almeida, C. Camilo-Junior, S. Correa, and K. Cardoso. "A Genetic Algorithm for Efficiently Solving the Virtualized Radio Access Network Placement Problem". In: *ICC 2023 - IEEE International Conference on Communications*. 2023, pp. 1874–1879. DOI: 10.1109/ICC45041.2023.10279334.

[20] W. Shahzad, Q. Rehman, and E. Ahmed. "Missing Data Imputation using Genetic Algorithm for Supervised Learning". In: *International Journal of Advanced Computer Science and Applications* 8.3 (2017), pp. 423–431. DOI: 10.14569/ijacsa.2017.080360.

[21] A. F. L. Jacob Junior, F. A. do Carmo, A. L. de Santana, E. E. Carvalho Santana, and F. M. Franca Lobato. "EvoImp: Multiple Imputation of Multi-label Classification data with a genetic algorithm". In: *PLOS ONE* 19.1 (2024), e0297147. DOI: 10.1371/journal.pone.0297147.

[22] K. Chungnoy, T. Tanantong, and P. Songmuang. "Missing value imputation on gene expression data using bee-based algorithm to improve classification performance". In: *PLOS ONE* 19.8 (2024), e0305492. DOI: 10.1371/journal.pone.0305492.

[23] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[24] D. S. Soper. "Hyperparameter Optimization Using Successive Halving with Greedy Cross Validation". In: *Algorithms* 16.1 (2023), p. 17. DOI: 10.3390/a16010017.

[25] C. Y. Guo, Y. C. Yang, and Y. H. Chen. "The Optimal Machine Learning-Based Missing Data Imputation for the Cox Proportional Hazard Model". In: *Frontiers in Public Health* 9 (2021), p. 680054. DOI: 10.3389/fpubh.2021.680054.

[26] Jianjiao Deng, Yunuo Qin, Xi Chen, Yanyong He, Yu Song, Xinpeng Zhang, Wenting Ma, Shoukui Li, and Yudong Wu. "Multi-Objective Optimization Design of Low-Frequency Band Gap for Local Resonance Acoustic Metamaterials Based on Genetic Algorithm". In: *Machines* 13.7 (2025), p. 610. DOI: 10.3390/machines13070610.

[27] John T. Blake, Nicole Rickards, Jonathan Mack, Matthew Seftel, and Angela Gupta. "Genetics by genetic algorithm: Defining an ideal platelet donor population to support patients with HLA-mediated alloimmune refractoriness". In: *Transfusion* 65.12 (2025), pp. 2398–2405. DOI: 10.1111/trf.18446.

[28] S. Watanabe. "Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance". In: *arXiv:2304.11127* (2023). DOI: 10.48550/arXiv.2304.11127.