ACCSCIENCE PUBLISHING

Artificial Intelligence in Health

ORIGINAL RESEARCH ARTICLE

Cooperative metaheuristic feature selection and out-of-fold weighted ensemble learning for chronic kidney disease prediction from a reduced attribute panel

Amaresh Ayyanahalli Math* and Ananda Murthy Meenakshi Sundaram*

School of Computer Science and Engineering, Faculty of Engineering and Technology, REVA University, Bangalore, Karnataka, India

Abstract

Chronic kidney disease is diagnosed from a broad laboratory panel whose size and composition are open to optimization. This study pairs a cooperative hybrid metaheuristic for attribute selection with a weighted voting ensemble. Genetic, swarm, and annealing sub-populations evolved in parallel over one shared fitness landscape, exchanged elites every five iterations, and were refined by annealing applied as a local search on the incumbent; the attributes retained across the elite band formed the subset, which fed four heterogeneous learners whose probabilities were fused using weights tuned from out-of-fold predictions only. Imputation, search, learners, and weights were confined to the training partition of a 10 times five-fold cross-validation, and every single-operator variant, four classical selectors, and the unreduced panel were evaluated under that identical protocol. The ensemble reached 99.00% accuracy, an area under the curve of 0.9994, and an F-measure of 99.19% from 11 of the 24 attributes, dropping 53% of the assays at a cost of two-thirds of a percentage point that is not statistically significant once the dependence between cross-validation folds is modeled. Accuracy held at 98.00% when a further fifth of all values was deleted, and an independent cohort of 200 records recorded as discretized intervals returned 98.00%. The pipeline is statistically equivalent to the strongest single learner and to the strongest classical selector while using fewer than half the attributes.

Keywords: Chronic kidney disease; Hybrid metaheuristic optimization; Feature selection; Ensemble classification; Nested validation; Reproducibility; External validation

***Corresponding authors:**
Amaresh Ayyanahalli Math (r21pcs28@reva.edu.in)
Ananda Murthy Meenakshi Sundaram (meenakshi.sa@reva.edu.in)

Citation: Math AA, Sundaram AMM. Cooperative metaheuristic feature selection and out-of-fold weighted ensemble learning for chronic kidney disease prediction from a reduced attribute panel. *Artif Intell Health*.
doi: 10.36922/AIH026330112

Received: August 11, 2026

Revised: September 1, 2026

Accepted: September 21, 2026

Published online: October 13, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Chronic kidney disease (CKD) is now among the fastest-growing contributors to global mortality, and the most recent burden of disease analyses place the number of people living with it in the hundreds of millions.[1,2] Diabetes and hypertension drive most of that growth. The clinical difficulty is that kidney function declines quietly: an estimated glomerular filtration rate can fall well below normal before a patient reports anything, and once the disease reaches the end stage, the only options are dialysis or transplantation. Detection therefore depends on laboratory screening, and screening is expensive in

exactly the settings where the disease is growing fastest.[3]

Two ideas from machine learning are relevant to that problem, and both are well established on their own. The first is feature selection. Clinical panels contain attributes that duplicate each other and that contribute nothing, so a search over subsets can reduce the number of assays a decision rule needs. The second is ensemble learning. A single classifier commits to one inductive bias, whereas a committee that pools several classifiers is usually more accurate and, more importantly for a clinical setting, less brittle. Neither idea is new. What is less settled is how they should be combined, how the combination should be validated, and how much either actually contributes once validation is done properly.

Metaheuristic algorithms are the standard tool for the search. Genetic algorithms, particle swarm optimization, and simulated annealing have all been applied to attribute selection.[4-6] Each has a characteristic failure. A genetic population loses diversity and converges early. A swarm follows its global best and can be trapped when that best sits in a local optimum. Annealing explores well but wanders, since it carries no population memory to exploit what it has found. These failures are complementary, which supports hybridization, and hybrid designs that pair one global search with annealing as a local refiner are reported to outperform either component.[7-9]

On the classification side, majority voting remains the reference rule for combining classifiers because it is simple and hard to break.[10,11] Its weakness is that it treats every member as equally competent. Weighted variants exist[12,13], but the weights are usually fixed in advance or fitted on the same data used to fit the members, reintroducing the optimism that the ensemble was supposed to remove.

1.1. Ensemble classification for chronic kidney disease

Ensembles dominate the CKD cohort: Chhabra et al.[14] found a stacking ensemble to be the strongest among 12 classifiers, with and without synthetic oversampling. Ensemble learning has been applied to renal decline in longitudinal cohorts[15], and hybrid deep frameworks combining biomarkers with ensemble feature engineering more recently.[16] Ensemble preprocessing pipelines for early detection[17] and ensemble deep learning with dimensionality reduction have also been described.[18] Comparative studies of CKD risk classifiers continue to emerge[19,20], and interpretable formulations are becoming more common.[21,22]

1.2. Voting and weighted fusion

Majority voting has been used for COVID-19 detection from chest radiographs[10] and for attack detection in connected devices.[11] Weighted variants have been reported for melanoma diagnosis[12] and skin lesion classification.[13] Analyses of metaheuristics for ensemble construction[23] and ensemble subset selection for high-dimensional, low-sample data[24] address the same combination problem from the optimization side.

1.3. Metaheuristic feature selection

Genetic algorithms remain widely used: penalty-embedded formulations for interpretable clinical selection[4], optimized variants for high-dimensional data[25], correlation-guided designs[26], applications to heterogeneous physiological signals[27], and guided population strategies with modified crossover have all been reported.[28] Swarm methods form the other main family, with multi-objective adaptive variants[5] and improved binary formulations.[29] Annealing is most often a refiner rather than a standalone search, combined with gray wolf optimization[6], a binary COOT algorithm on microarray data[7], a dwarf mongoose optimizer[8], Harris hawks optimization for deep feature screening[9], and a pigeon-inspired optimizer.[30] Improved annealing with a support vector machine has been reported for breast cancer[31], and filter-based rankers have been used to guide a grasshopper and annealing search.[32] Optimization-driven hybrids have also been built for tumor imaging.[33]

1.4. Wrapper design and data quality

Wrapper and hybrid strategies with metaheuristic engines have been compared systematically[34], and search-space division[35] and surrogate-assisted evaluation[36] both address the cost of wrapper evaluation; resampling and weighted losses can also influence the attributes selected by a wrapper.[37,38] Filter-based reduction has been applied to this cohort with a cost-sensitive boosting classifier[39], and hybrid signature-based selection with sequence models.[40] Three-stage designs that chain a filter to a wrapper[41] are structurally similar to the cascade used here.

This work addresses a gap in evaluation. Metaheuristic feature selection and optimization-driven weighted ensembles were both established, and recent work on this cohort already combines them, so the contribution claimed here is not the combination itself. Hybrid metaheuristics for CKD are normally assembled as a sequence, compared with their own components on a single split under an objective that differs from the baseline's, and paired with a voting rule whose weights are uniform or fitted in sample, with the reported subset the incumbent rather than a consensus; each of those choices favors the proposed method independently of its operators. This study holds all of them fixed, so every search variant, four classical selectors, and the unreduced panel share one objective, one

budget, one consensus rule, and one resampling protocol with the fusion weights fitted out of fold. What the pipeline delivers under those conditions is a panel of 11 attributes that performs on a par with the full 24 and with every alternative selector tried.

This study keeps the objective of the original design, namely metaheuristic feature engineering feeding a voting ensemble, rebuilds both halves, and evaluates them under a single matched protocol. The contributions are as follows.

- A cooperative co-evolutionary optimizer in which genetic, swarm, and annealing sub-populations evolve in parallel over one shared fitness landscape and exchange elites periodically, with annealing additionally applied as a memetic local search around the incumbent. The design is evaluated against its own components under an identical budget rather than against a differently configured baseline.
- A wrapper objective that couples the ranking quality of a candidate subset to its balanced accuracy, because accuracy alone saturates on this cohort and leaves the search without a gradient.
- A selection rule that reports the consensus of the elite band of visited solutions rather than the single best solution, which is an optimistically biased maximum of a noisy estimate.
- A fusion stage in which the combination weights are tuned on the simplex with a continuous particle swarm using out-of-fold probabilities only, so no member is credited for predictions it made on data it was fitted on. Majority voting, soft voting, and stacking are reported alongside it.
- A matched-protocol evaluation in which the cooperative hybrid, each single-operator variant, four classical selectors (L1-regularized logistic regression, random-forest recursive feature elimination, a mutual-information filter, and random-forest importance), and the unreduced panel receive the same nested resampling, the same base learners, and the same fusion rule, so the comparison isolates the selector.
- A statistical analysis that respects the dependence structure of a repeated cross-validation, reporting every comparison at the repetition level, where the 10 repetitions are completely disjoint partitions, and under the Nadeau–Bengio corrected resampled *t*-test, as well as fold-wise.
- An evaluation that measures ensemble diversity explicitly, traces performance as missing values and label noise are injected over independent corruption realizations rather than a single draw, reports the sensitivity of the external result to the interval-to-scalar harmonization, and validates on an independent cohort of 200 records.

The remainder of the paper is organized as follows. **Section 2** describes the cohorts and the method, **Section 3** reports the experiments, **Section 4** discusses them, and **Section 5** concludes.

2. Data and methods

2.1. Cohorts

The primary cohort is the CKD dataset from the University of California, Irvine machine learning repository: 400 records (250 CKD cases, 150 non-CKD cases) and 24 clinical attributes (11 numerical, 13 nominal); 778 of the 9600 attribute values are missing. Table 1 summarizes the panel.

External validation uses the risk-factor prediction of the CKD dataset from the same repository, identifier 857: 200 records, 128 of them are CKD. That cohort is distributed with 28 attributes, not 24. Twenty-four correspond one-to-one with the panel of the primary cohort and were the only ones used here; the remaining four (bp limit, grf, stage, and affected) were excluded. Comparability is only part of the reason. An estimated glomerular filtration rate (grf), a CKD stage (stage), and a disease indicator (affected) are derived from, or are restatements of, the outcome being predicted, so retaining them would leak the label rather than test transfer; bp limit is a coarsened duplicate of the diastolic pressure indicator already used. The two cohorts are separate collections deposited independently in the same repository, share no records, and cohort 857 was not consulted during development. They are not, however, samples from two different populations under two different acquisition protocols, and cohort 857 is a discretized re-encoding rather than a fresh set of raw measurements, so the external figures measure transfer to a differently encoded cohort.

Because cohort 857 records every continuous attribute as a discretized interval, values were mapped to representative scalars before scoring: a closed interval to its midpoint, an open interval to the boundary displaced by half the median width of that attribute's closed intervals, and the binary diastolic indicator to the representative normal and elevated values of the primary cohort, 70 and 90 mmHg; the categorical attributes are already indicators in both cohorts. Because any such mapping discards within-interval variation, **Section 3.12** re-scores the cohort under alternative mappings, with the subset, the models, and the weights held fixed. This cohort was scored once, after everything else was fixed.

2.2. Overview

The pipeline has three phases (Figure 1). Data are

Table 1. Attributes of the CKD cohort (400 records: 250 CKD and 150 non-CKD)

Attribute	Code	Type	Observed range	Mean ± SD	Missing values (*n* [%])
Age (year)	age	Numerical	2 to 90	51.48 ± 17.17	9 (2.2%)
Blood pressure (mmHg)	bp	Numerical	50 to 180	76.47 ± 13.68	12 (3.0%)
Specific gravity	sg	Numerical	1.00 to 1.02	1.02 ± 0.01	47 (11.8%)
Albumin	al	Numerical	0 to 5	1.02 ± 1.35	46 (11.5%)
Sugar	su	Numerical	0 to 5	0.45 ± 1.1	49 (12.2%)
Red blood cells	rbc	Nominal	–	–	0 (0.0%)
Pus cell	pc	Nominal	–	–	0 (0.0%)
Pus cell clumps	pcc	Nominal	–	–	0 (0.0%)
Bacteria	ba	Nominal	–	–	0 (0.0%)
Blood glucose random	bgr	Numerical	22 to 490	148.04 ± 79.28	44 (11.0%)
Blood urea	bu	Numerical	1.5 to 391	57.43 ± 50.5	19 (4.8%)
Serum creatinine	sc	Numerical	0.4 to 76	3.07 ± 5.74	17 (4.2%)
Sodium	sod	Numerical	4.5 to 163	137.53 ± 10.41	87 (21.8%)
Potassium	pot	Numerical	2.5 to 47	4.63 ± 3.19	88 (22.0%)
Hemoglobin	hemo	Numerical	3.1 to 17.8	12.53 ± 2.91	52 (13.0%)
Packed cell volume	pcv	Numerical	9 to 54	38.88 ± 8.99	71 (17.8%)
White blood cell count	wbcc	Numerical	2,200 to 26,400	8,406.12 ± 2,944.47	106 (26.5%)
Red blood cell count	rbcc	Numerical	2.1 to 8.0	4.71 ± 1.03	131 (32.8%)
Hypertension	htn	Nominal	–	–	0 (0.0%)
Diabetes mellitus	dm	Nominal	–	–	0 (0.0%)
Coronary artery disease	cad	Nominal	–	–	0 (0.0%)
Appetite	appet	Nominal	–	–	0 (0.0%)
Pedal edema	pe	Nominal	–	–	0 (0.0%)
Anemia	ane	Nominal	–	–	0 (0.0%)

Abbreviation: CKD: Chronic kidney disease.

normalized and imputed. A cooperative hybrid metaheuristic then searches the space of attribute subsets. The selected subset feeds four heterogeneous base learners whose probabilistic outputs are fused via weighted voting, with majority voting, soft voting, and stacking reported for comparison.

2.3. Preprocessing

Attributes are scaled to the unit interval:

$$x_{normalized} = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

with the extrema taken over the training partition. Missing values are then filled using iterative tree-based imputation in the manner of missForest. Writing the data matrix as $X = (X_1, X_2, \ldots, X_p)$ of size $n \times p$, and splitting each attribute s into its observed part x^s_{obs} and its missing part x^s_{mis} indexed by $i^s_{mis} \subseteq \{1, \ldots, n\}$, the procedure initializes the gaps with attribute means, orders the attributes by their amount of missingness, and repeatedly regresses each incomplete attribute on the others until the imputed matrix stops changing. Algorithm 1 states the procedure.

Imputation is fitted on the training partition and applied to the test partition, never the reverse. This matters more than it might appear: an imputer fitted on the pooled cohort propagates information from the test records into training, and on a cohort where a third of one attribute is missing, that leak is not negligible.

2.4. Cooperative hybrid metaheuristic feature selection

A candidate subset is a binary vector $x \in \{0,1\}^p$ in which a one marks a retained attribute. Continuous coordinates are converted to bits using **Equation 2**:

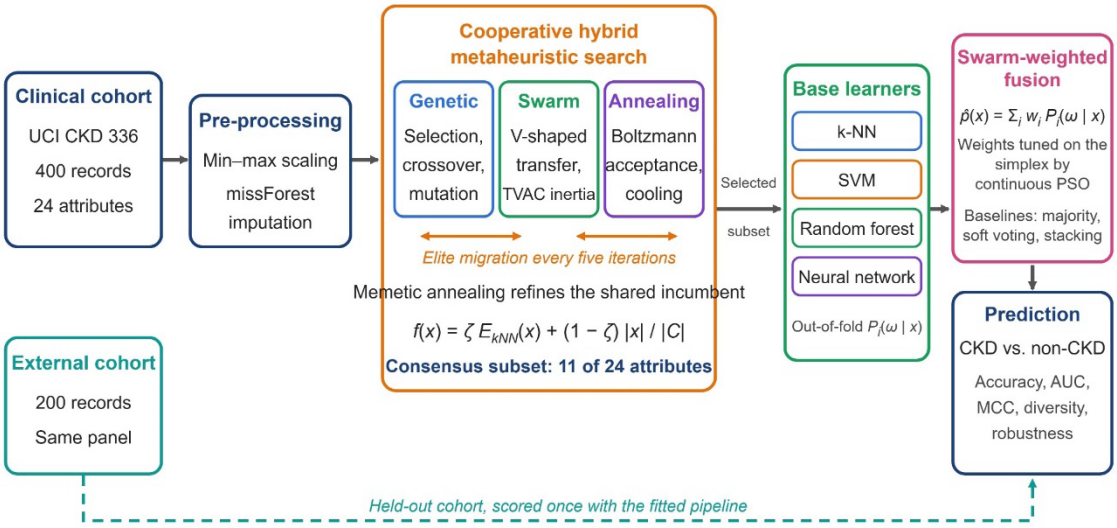


Figure 1. The proposed framework. Imputation, cooperative search, the base learners, and the fusion weights are all fitted within the training partition of each cross-validation fold.
Abbreviations: AUC: Area under the curve; CKD: Chronic kidney disease; k-NN: k-nearest neighbor; MCC: Matthews correlation coefficient; PSO: Particle swarm optimization; SVM: Support vector machine; TVAC: Time-varying acceleration coefficient; UCI: University of California, Irvine.

Algorithm 1. Iterative tree-based imputation

Step	Operation
Input	matrix X of size $n \times p$, stopping criterion γ
1	initialize the missing entries with the attribute means
2	$k \leftarrow$ indices of the columns sorted by amount of missingness
3	**while** γ is not satisfied **do**
4	$x_{old}^{imp} \leftarrow$ the previously imputed matrix
5	**for** s in k **do**
6	fit a tree-based regressor on $y_{obs}^{s} \sim x_{obs}^{s}$
7	predict y_{mis}^{s} from x_{mis}^{s}
8	$X_{new}^{imp} \leftarrow$ update the imputed matrix with the predictions
9	**end for**
10	update γ
11	**end while**
12	**return** the imputed matrix x^{imp}

$$x_i(t+1) = \begin{cases} 1 & \text{if} \frac{1}{1+e^{-x_i(t)}} > \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Three sub-populations evolve in parallel over one shared landscape, and each contributes a different operator:

(i) Genetic operators. A genetic algorithm represents candidates as bit strings and evolves them by selection, crossover, and mutation. Parents are chosen by binary tournament, recombined by two-point crossover with probability 0.8, and mutated bitwise with probability 0.05. The two fittest individuals of each generation survive unchanged.

(ii) Swarm operators. Particles carry a position and a velocity, $x_i = (x_{i1}, \ldots, x_{iD})$ and $v_i = (v_{i1}, \ldots, v_{iD})$, and are attracted to their own best position p_{best} and to the best position identified by the population g_{best}:

$$v_{id}^{t+1} = w^t \times v_{id}^t + c_1^t \times r_{1i} \times \left(p_{id} - x_{id}^t\right) + c_2^t \times r_{2i} \times \left(p_{gd} - x_{id}^t\right) \quad (3)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (4)$$

The inertia weight falls linearly from 0.9 to 0.4, while the acceleration coefficients vary with time, c_1 from 2.5 to 0.5 and c_2 from 0.5 to 2.5, so the population explores early and concentrates later. Bits are flipped through the V-shaped transfer function $\left|\tanh\left(v_{id}\right)\right|$, which responds to the magnitude of the velocity rather than to its sign.

(iii) Annealing operators. Simulated annealing perturbs a solution and accepts the result unconditionally when it improves and with the Boltzmann probability (p) of $e^{-\theta/T}$ when it does not, where θ is the loss in quality, and T is the temperature. The temperature starts at $T_0 = 2|N|$, twice the number of attributes, and cools geometrically as $T \leftarrow 0.93T$. Accepting a worse solution occasionally is what lets the chain leave a local optimum. Algorithm 2 states the annealing step.

2.4.1. Cooperation

The three sub-populations are not run in sequence. They advance together, and every five iterations, the best individual of each sub-population replaces the worst individual of its neighbor, so a discovery made by one operator becomes available to the others while the search runs. In addition, annealing is applied as a memetic local search on the incumbent solution at every iteration, which refines the shared best rather than restarting from a random point. Setting the active operator set to a single member recovers the standalone algorithms, and those variants form the ablation study in **Section 3.2**.

Algorithm 2 states the annealing step in its textbook form. As embedded in the cooperative search, it is not run to a temperature floor: each of the 30 outer iterations performs one Metropolis sweep over the annealing sub-population and then applies the geometric cooling once, so the chain is terminated by the shared iteration budget rather than by a temperature criterion. This keeps the evaluation budget identical across the variants of the

Algorithm 2. Annealing step within the cooperative search

Step	Operation
1	$T_0 = 2\|N\|$, where $\|N\|$ is the number of attributes
2	$BestSol \leftarrow S_i$ and evaluate $\delta(BestSol)$
3	**while** $T > T_{\min}$ **do**
4	generate a neighbor $TrialSol$ by flipping a random tenth of the bits
5	evaluate $\delta(TrialSol)$
6	**if** $\delta(TrialSol) < \delta(BestSol)$ **then** accept the trial and update $BestSol$
7	**else if** the two are equal **then** accept the trial when $\|TrialSol\| < \|BestSol\|$
8	**else** compute $\theta = \delta(TrialSol) - \delta(BestSol)$, draw $P \in [0,1]$ and accept when $P \le e^{-\theta/T}$
9	$T \leftarrow 0.93T$
10	**end while**
11	**return** $BestSol$

ablation, which would not hold if annealing ran to its own stopping condition. **Section 2.7** gives the complete settings.

2.4.2. Objective

The fitness of a subset follows the wrapper form of the original design:

$$f\left(x_i\left(t\right)\right)=\zeta E_{x_i(t)}+\left(1-\zeta\right)\left(\frac{\left|x_i\left(t\right)\right|}{\left|C\right|}\right) \quad (5)$$

where E is the classification error of a k-nearest neighbor wrapper, $\left|x\right|$ is the number of selected attributes, $\left|C\right|$ is the total number, and ζ balances the two terms and is fixed at 1 throughout this work, so that the cardinality weight in **Equation 5** is exactly zero and the objective reduces to the error term alone; **Section 3.13** reports the supporting experiment. The error term is estimated using repeated stratified cross-validation inside the training partition and is decomposed into a ranking component and a threshold component:

$$E_x=\omega\left(1-AUC\left(x\right)\right)+\left(1-\omega\right)\left(1-BA\left(x\right)\right) \quad (6)$$

with $\omega=0.7$. The decomposition is not cosmetic. Accuracy on this cohort saturates, so a large fraction of candidate subsets share the same inner accuracy, and an accuracy-driven objective is flat across much of the space; the area under the curve is continuous and keeps discriminating in that regime. The cardinality weight $1-\zeta$ is set to zero, because a penalty large enough to influence a saturated objective drives the search toward subsets that are small rather than good. Parsimony is obtained instead from the consensus rule in **Section 2.4.3**. Two points of notation should be explicit, because both have caused confusion. First, the trade-off parameter of **Equation 5** and the ranking weight of **Equation 6** are different parameters: the ranking weight is 0.7, and the trade-off parameter is 1. It is the latter that makes the cardinality term vanish, so the value 0.7 does not imply a cardinality weight of 0.3. Second, the ranking weight of **Equation 6** is unrelated to the class labels written with the same letter in **Section 2.5**. **Section 3.13** reports how the search behaves across a range of the cardinality weight.

2.4.3. Selection rule

The search records every solution it evaluates. The final subset is not the incumbent but the consensus of the elite band:

$$s_j^*=\mathbf{1}\left[\frac{1}{\left|E\right|}\sum_{s\in E}s_j>0.5\right],\quad E=\left\{\text{the best 10\% of evaluated solutions}\right\} \quad (7)$$

The incumbent is the maximum of a noisy estimate taken over several hundred evaluations and is therefore optimistically biased. Averaging over the elite band damps that bias. The rule is applied identically to every search variant, so the ablation compares search mechanisms, not selection rules.

2.5. Ensemble classification

Four heterogeneous learners are fitted on the selected attributes:

(i) K-nearest neighbors classifies a query by the majority label among its k closest training records, with contributions weighted by inverse distance.
(ii) Support vector machine separates the classes by a maximum margin hyperplane in the feature space induced by a radial basis kernel, with class weights inversely proportional to class frequency.
(iii) Random forest grows an ensemble of decision trees on bootstrap samples with a random subset of attributes considered at each split, and predicts by aggregating the trees.
(iv) A neural network is a feedforward network with two hidden layers of 64 and 32 units trained using backpropagation.

The four are deliberately different in inductive bias. An ensemble gains nothing from members that make the same mistakes. **Section 3.5** measures how far apart these four actually are.

For fusion, let $C=\{C_1,\ldots,C_l\}$ be the classifiers, each mapping an object $x\in\mathbb{R}^n$ to a vector of class probabilities $\left[P_{C_i}(\omega_1|x),\ldots,P_{C_i}(\omega_c|x)\right]$. Majority voting takes the label supported by most members. Soft voting averages the probabilities:

$$m_j=\frac{1}{l}\sum_{i=1}^{l}P_{C_i}\left(\omega_j|x\right) \quad (8)$$

The proposed rule replaces the uniform average with a weighted one:

$$\hat{p}\left(\omega_j\mid x\right)=\sum_{i=1}^{l}w_i\,P_{C_i}\left(\omega_j\mid x\right),\quad w_i\geq 0,\quad \sum_{i=1}^{l}w_i=1 \quad (9)$$

and the weights are obtained using a continuous particle swarm that maximizes balanced accuracy on out-of-fold probabilities:

$$w^*=arg\max_{w\in\Delta^{l-1}}\left|BA\left(y_{tr},\left|P_{oof}\;w\right.\right)\right. \quad (10)$$

where P_{oof} is the matrix of probabilities each member produced for training records while those records were held

out from its own fitting. Using out-of-fold probabilities is the point. Weights fitted on in-sample probabilities reward members that overfit, which is precisely the failure a committee is meant to prevent. The swarm is seeded with the uniform weight vector, so soft voting is contained in its search space, and the optimizer can only improve on it in-sample.

2.6. Evaluation protocol and statistical analysis

Performance is estimated using stratified five-fold cross-validation repeated 10 times. Imputation, scaling, the metaheuristic search, the base learners, and the fusion weights are all fitted within the training partition of each fold. From the confusion matrix:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

$$TNR = \frac{TN}{TN + FP} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

$$FNR = \frac{FN}{FN + TP} \tag{16}$$

$$F = \frac{2PR}{P + R} \tag{17}$$

ensemble diversity is quantified pairwise. For two members with correctness vectors a and b, Yule's Q statistic is:

$$Q_{a,b} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{18}$$

where N^{11} counts records that both classify correctly, and so on. Values near one indicate members that fail together. The disagreement measure is the fraction of records on which two members output different labels, and the interrater agreement κ corrects that for chance. Fusion rules are compared using a Friedman omnibus test followed by the Nemenyi critical difference, and pairwise using Wilcoxon signed-rank tests with Holm correction over one family of all comparisons against the proposed ensemble. Confidence intervals come from 2,000 bootstrap resamples.

The 50 folds of a 10 times five-fold design are not 50 independent replicates: every observation is a test case 10 times, and any two training partitions overlap in most of their records, so treating them as independent understates the variance of a paired difference. Each comparison is therefore reported three ways: the fold-level Wilcoxon and Friedman tests, for continuity with the literature; a repetition-level analysis averaging the five folds of each repetition first, so the 10 repetition means are proper exchangeable blocks; and the Nadeau–Bengio corrected resampled t-test, which inflates the variance of the fold differences by $1/k + n_{test}/n_{train}$, equal to 0.45 here. Holm correction is applied within each family. **Section 3.11** reports all three side by side.

The same protocol is applied to four classical selectors, so the search is measured against something other than its own components: L1-regularized logistic regression with the penalty chosen using inner five-fold cross-validation and the non-zero coefficients retained; recursive feature elimination driven by a 200-tree random forest with the subset size chosen using inner five-fold cross-validated elimination; a mutual-information filter whose cut-off is chosen using inner five-fold cross-validation of the same nearest neighbor wrapper; and random-forest importance retaining attributes above mean importance. Each chooses its own subset size within the training partition and feeds the identical four base learners and the identical out-of-fold weighted fusion.

2.7. Implementation details and reproducibility

Every setting used in the study is listed here so that the experiments can be repeated. Table A1 presents the search and fusion parameters and Table A2 the imputer and the learners; nothing was tuned on any test partition, and the ablation variants differ only in which sub-populations are active. The optimizer runs a fixed 30 iterations with no early stopping, making the budget comparable across variants; fitness values are memoized on the attribute mask, so the search averaged 672 distinct subsets per fold for the cooperative hybrid and between 552 and 736 for the single-operator and two-operator variants. The wrapper fitness of a subset is the mean over a stratified five-fold cross-validation repeated twice inside the training partition; the out-of-fold probabilities the fusion weights are fitted on come from a separate stratified five-fold cross-validation with the same partition, and neither sees the outer test fold. The global seed is 42, and outer fold i uses seed 42 + i throughout, so a fold is reproducible in

isolation; the corruption study of **Section 3.7** uses seeds 42 + 1,000r + *i* for realization *r* and fold *i*. The environment is Python 3.14.6, scikit-learn 1.9.0, NumPy 2.5.2, SciPy 1.18.0, pandas 3.0.3, and joblib 1.5.3; per-fold runtimes are shown in **Section 3.9**.

3. Results

3.1. Search behavior

Figure 2 traces the best fitness against iteration for the cooperative hybrid and for its single-operator variants, averaged over the 50 folds. All variants minimized the same objective with the same evaluation budget, so the curves are directly comparable. The hybrid and the two population-based variants converged within roughly 10 iterations. Annealing alone was the slowest and settled at a clearly worse fitness (0.00233 vs. 0.00096 for the hybrid), as expected for a search with no population memory to exploit what it has found. The genetic sub-population reached the lowest fitness of all (0.00084), and the hybrid sat between it and the swarm.

That ordering did not carry through to downstream accuracy, and the reason is worth stating plainly. The fitness was measured with a nearest-neighbor wrapper on held-out inner folds, while the downstream number was produced using a four-member ensemble. Better wrapper fitness therefore does not guarantee a better ensemble, and on a cohort where the inner criterion is close to its ceiling, the residual differences between the searches are comparable to the noise in the estimate.

3.2. Effect of the search strategy

Table 2 and Figure 3 report the accuracy of the weighted ensemble under each search strategy. Each variant used the same objective, budget, and consensus selection rule, so the comparison isolates the search operators. The cooperative hybrid achieved 99.00% from 11.22 attributes, the annealing-only variant 99.48% from 13.58 attributes, and the genetic and swarm variants 99.17% and 99.12%. None of these differences survived Holm correction at the fold level, with adjusted *p*-values of 0.126, 0.601, and 0.968 against the annealing, genetic, and swarm variants, respectively, and all three rose to 1.000 under the dependence-aware analysis in **Section 3.11**. The five searches were therefore statistically equivalent on this cohort, suggesting that the choice of metaheuristic may have less influence than the literature on hybrid optimizers tends to suggest when all variants use the same objective, budget, and unbiased selection rule.

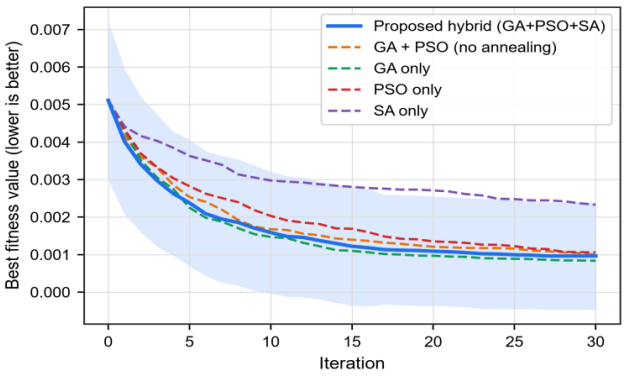


Figure 2. Best fitness against iteration for the cooperative hybrid and its single-operator variants, averaged over 50 folds. Shading is one standard deviation.
Abbreviations: GA: Genetic algorithm; PSO: Particle swarm optimization; SA: Simulated annealing.

Only the comparison with no attribute selection reached significance at the fold level. An ensemble with the full 24-attribute panel achieved 99.67% versus 99.00% for the selected subset, with a fold-level adjusted *p*-value of 0.004. **Section 3.11** shows that this figure depends on

Table 2. Effect of the search strategy on the accuracy of the weighted voting ensemble

Search strategy	Attributes	Accuracy (%)	Recall (%)	F1 (%)	MCC	AUC	*p* (Holm)
Cooperative hybrid, all three operators	11.2	99.00 ± 1.16	98.56	99.19	0.9793	0.9994	Reference
Genetic and swarm, no annealing	11.6	99.02 ± 1.27	98.60	99.21	0.9799	0.9995	1.000
Genetic algorithm only	10.9	99.17 ± 1.12	98.84	99.33	0.9829	0.9993	0.601
Particle swarm only	11.2	99.12 ± 1.14	98.76	99.29	0.9819	0.9994	0.968
Simulated annealing only	13.6	99.48 ± 0.88	99.32	99.57	0.9891	0.9998	0.126
No selection, all 24 attributes	24.0	99.67 ± 0.83	99.68	99.74	0.9933	1.0000	0.004

Notes: All variants minimize the same objective, use the same evaluation budget, and report the consensus of their own elite band, so the comparison isolates the search operators. The final column gives the Holm-adjusted *p*-value of a Wilcoxon signed-rank test against the cooperative hybrid, computed over 50 folds. Because those folds are not independent, Table 3 repeats every comparison at the repetition level and under the Nadeau–Bengio correction.

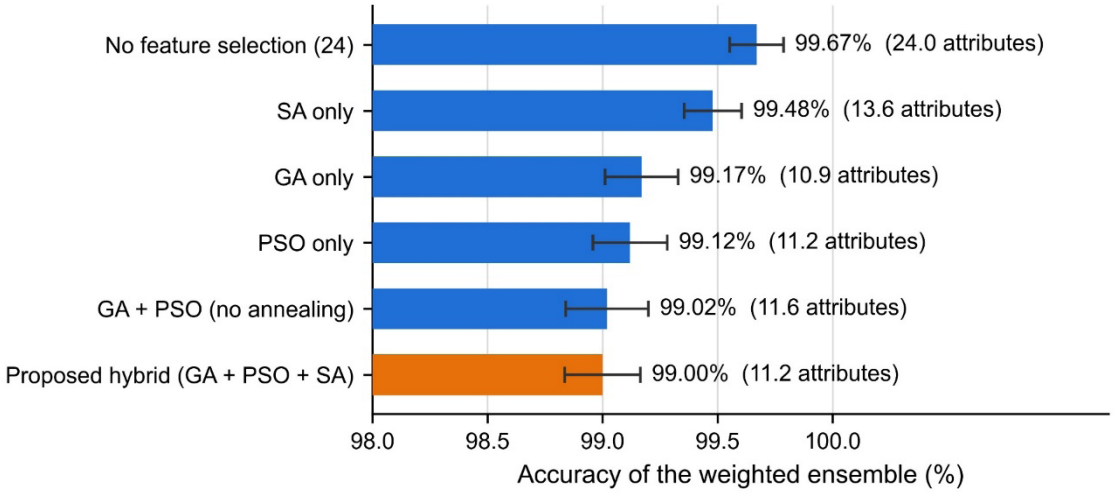


Figure 3. Accuracy of the weighted voting ensemble under each search strategy. Error bars are one standard error of the mean over 50 folds Abbreviations: GA: Genetic algorithm; PSO: Particle swarm optimization; SA: Simulated annealing.

Table 3. Every comparison against the proposed weighted ensemble under three analyses

Comparison against the proposed ensemble	Accuracy (%)	*p* (Holm, 50 folds)	*p* (Holm, 10 repetitions)	*p* (Nadeau–Bengio, unadjusted)
No selection, all 24 attributes	99.67	0.004	0.102	0.34
Simulated annealing only	99.47	0.126	1.000	0.62
Random forest alone	99.45	0.025	0.375	0.51
Genetic algorithm only	99.17	0.601	1.000	0.76
Particle swarm only	99.12	0.968	1.000	0.81
Genetic and swarm, no annealing	99.03	1.000	1.000	0.97
Stacking, logistic meta-learner	99.00	1.000	1.000	1.00
Soft voting	98.85	0.591	1.000	0.73
Majority voting	98.83	0.429	1.000	0.70
Support vector machine alone	98.77	0.591	1.000	0.73
Neural network alone	98.65	0.067	0.430	0.57
K-nearest neighbors alone	98.62	0.140	0.375	0.56

Notes: The fold-level column is the analysis reported in Tables 2 and 4 and treats the 50 folds as independent; the repetition-level column averages within each of the 10 repetitions first; the last column is the unadjusted Nadeau–Bengio corrected resampled *t*-test, every value of which is 1.000 after Holm correction.

treating the 50 folds as independent: at the repetition level, the adjusted *p*-value is 0.102 and the unadjusted Nadeau–Bengio *p*-value is 0.34, so the reduced panel is statistically equivalent to the full one. The magnitude of the trade is clear: selection costs about two-thirds of a percentage point while reducing the number of measurements by 53%. Whether that is worth making depends on the setting rather than on the statistics; this is discussed further in **Section 4**.

3.3. Selected attributes

Figure 4 shows how often each attribute was selected across the 50 folds. Eleven attributes cleared the consensus threshold: specific gravity, hemoglobin, hypertension, pedal edema, packed cell volume, diabetes mellitus, appetite, sodium, red blood cells, coronary artery disease, and sugar. The mean subset contained 11.22 attributes with a standard deviation of 2.00. Specific gravity and

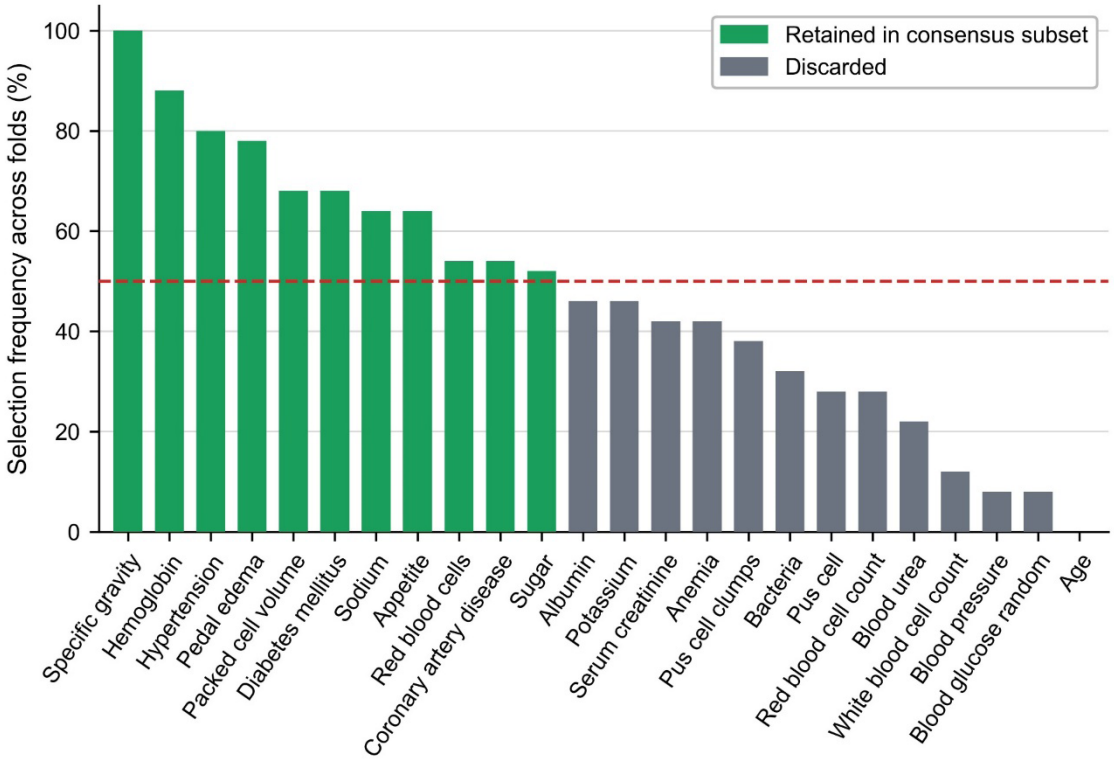


Figure 4. Attribute selection frequency across the 50 folds. Attributes above the dashed line form the consensus subset.

hemoglobin were chosen in nearly every fold, consistent with impaired urinary concentration and anemia resulting from reduced erythropoietin production, while the two major underlying conditions were represented directly by hypertension and diabetes.

3.4. Base learners and fusion rules

Table 4 reports the ranking quality, calibration, and statistical comparison; Table 5 reports the rates for the selected attributes. Among the individual learners, random forest was strongest at 99.45%, followed by support vector machine at 98.78%, neural network at 98.65%, and k-nearest neighbors at 98.63%. Weighted fusion reached 99.00%, ahead of majority voting at 98.83% and soft voting at 98.85%, and tied with stacking at 99.00%. Figure 5 shows the distribution of fold accuracy for the base learners and the fusion rules. The Friedman test over the 50 folds rejected equality ($p = 1.51 \times 10^{-16}$) and yielded a Nemenyi critical difference of 2.81 ranks; Figure 6 shows the resulting comparison.

Two comparisons in these tables deserve to be stated explicitly. First, the improvement of weighted fusion over majority voting (0.175 percentage points) is consistent in direction across folds but did not survive Holm correction (adjusted $p = 0.429$ at the fold level and 1.000 at the repetition level); it is a consistent improvement rather than a demonstrated one. Second, the random forest alone reached 99.45%, compared with 99.00% for the ensemble that contains it. That difference is significant fold-wise (adjusted p-value = 0.025) but not under the dependence-aware analysis in **Section 3.11**, where the repetition-level adjusted p-value is 0.375 and the unadjusted Nadeau–Bengio p-value is 0.510, so the committee and its strongest member are statistically equivalent. A committee gains over its members only when they fail on different records, and **Section 3.5** quantifies how little they do.

3.5. Diversity and weights

Table 6 reports the pairwise diversity of the ensemble members, Table 7 summarizes the combination weights assigned by the continuous swarm, and Figure 7 shows the learned weights and the kappa-error diagram. The members agree almost completely: the mean pairwise disagreement was 0.0078, meaning that two members gave different labels on fewer than one record in a hundred, and the mean interrater agreement was 0.984. The Q statistic averaged 0.389, lower than the disagreement figures alone

Table 4. Ranking quality, calibration, and statistical comparison of the fusion rules

Model	MCC	AUC	Brier score	Friedman rank	*p* (Holm)
Swarm-weighted voting	0.9793	0.9994	0.0073	7.73	Reference
Soft voting	0.9764	0.9995	0.0078	8.20	0.591
Majority voting	0.9758	0.9995	0.0078	8.34	0.429
Stacking, logistic meta-learner	0.9794	0.9994	0.0078	7.52	1.000
Neural network	0.9723	0.9991	0.0100	8.96	0.067
Support vector machine	0.9749	0.9987	0.0104	8.62	0.591
Random forest	0.9885	0.9983	0.0066	6.11	0.025
K-nearest neighbors	0.9718	0.9943	0.0113	8.98	0.140

Notes: The Friedman omnibus test over the 50 folds gives $p = 1.51 \times 10^{-16}$, with a Nemenyi critical difference of 2.81 ranks; over the 10 repetitions, the independent blocks of this design, it gives $p = 5.35 \times 10^{-7}$ with a critical difference of 6.27 ranks. Table 3 presents the pairwise values.
Abbreviations: AUC: Area under the curve; MCC: Matthews correlation coefficient.

Table 5. Base learners and fusion rules on the attributes chosen using the cooperative hybrid, averaged over 50 folds

Model	TPR (%)	FNR (%)	FPR (%)	TNR (%)	Accuracy (%)	Precision (%)	F-measure (%)
Swarm-weighted voting	98.56	1.44	0.27	99.73	99.00 ± 1.16	99.84	99.19
Soft voting	98.32	1.68	0.27	99.73	98.85 ± 1.36	99.84	99.06
Majority voting	98.28	1.72	0.27	99.73	98.83 ± 1.35	99.84	99.04
Stacking, logistic meta-learner	98.56	1.44	0.27	99.73	99.00 ± 1.21	99.84	99.19
Neural network	98.00	2.00	0.27	99.73	98.65 ± 1.36	99.84	98.90
Support vector machine	98.16	1.84	0.20	99.80	98.78 ± 1.39	99.88	99.00
Random forest	99.32	0.68	0.33	99.67	99.45 ± 0.88	99.80	99.56
K-nearest neighbors	97.96	2.04	0.27	99.73	98.62 ± 1.46	99.84	98.88

Abbreviations: FNR: False-negative rate; FPR: False-positive rate; TNR: True-negative rate; TPR: True-positive rate.

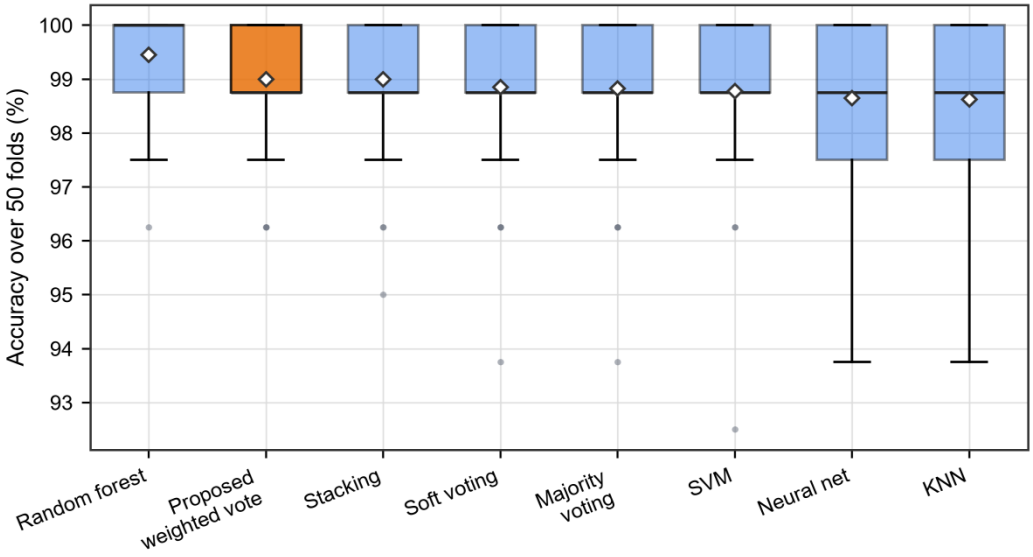


Figure 5. Distribution of fold accuracy for the base learners and the fusion rules. Diamonds mark the mean.
Abbreviations: KNN: K-nearest neighbor; SVM: Support vector machine.

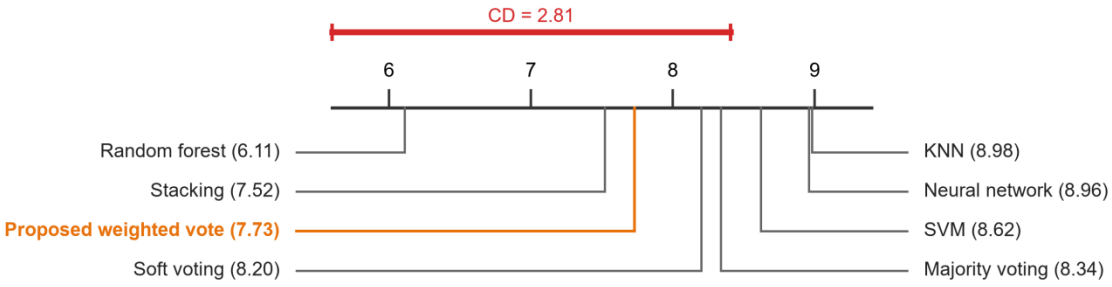


Figure 6. Nemenyi post-hoc comparison of the fusion rules and base learners over the 50 folds (Friedman $p = 1.51 \times 10^{-16}$). Methods further to the left rank better.
Abbreviations: CD: Critical difference; KNN: K-nearest neighbor; SVM: Support vector machine.

Table 6. Pairwise diversity of the ensemble members

Pair	*Q* statistic	Disagreement	κ
Support vector machine and k-nearest neighbors	0.5200	0.0050	0.9895
Random forest and k-nearest neighbors	0.2392	0.0112	0.9764
Neural network and k-nearest neighbors	0.5395	0.0053	0.9890
Support vector machine and random forest	0.2195	0.0102	0.9785
Support vector machine and neural network	0.5595	0.0038	0.9921
Random forest and neural network	0.2585	0.0115	0.9758

Notes: A *Q* statistic near 1 indicates members that fail on the same records; the disagreement measure and κ report how often two members differ and how far that exceeds chance.

Table 7. Combination weights assigned using the continuous swarm, summarized over the 50 folds

Member	Weight (mean ± SD)	Minimum	Maximum
K-nearest neighbors	0.2520 ± 0.1317	0.0037	0.6733
Support vector machine	0.2072 ± 0.1009	0.0000	0.5442
Random forest	0.3362 ± 0.1899	0.0112	0.8373
Neural network	0.2045 ± 0.1005	0.0000	0.4610

Note: A uniform ensemble would assign 0.25 to every member.

would suggest, indicating that the few errors the members made were only partly shared.

The swarm assigned 0.336 to random forest, 0.252 to k-nearest neighbors, 0.207 to support vector machine, and 0.205 to neural network, indicating that it identifies the strongest member and shifts weight toward it. It cannot shift far enough to recover the random forest on its own, because the weights are constrained to a simplex and the criterion it optimizes, out-of-fold balanced accuracy, is close to saturation. Replacing that criterion with a ranking-based one was tested and changed the weights by less than one part in a hundred, consistent with the diversity figures: when four learners disagree on fewer than 1% of records, no combination rule has much left to work with.

3.6. Discrimination, calibration, and errors

Figure 8 shows the receiver operating characteristic, the precision–recall curve, and the calibration of the pooled out-of-fold predictions of the first repetition. Figure 9 presents the confusion matrix. Averaged over the 50 folds, the area under the curve was 0.9994 with a 95% bootstrap interval of 0.9988 to 0.9998, and the accuracy interval was 98.67% to 99.30%. The Brier score was 0.0073, and the calibration curve remained close to the diagonal, indicating that the fused probability can be read as a risk estimate. In

the pooled matrix, there are three false negatives and no false positives out of 400 records. For a screening rule, that error direction was unfavorable, and the balanced class weighting used in the base learners kept it small.

3.7. Robustness

Table 7 and Figure 10 present what happens as the data degrade. At each level, the search, the base learners, and the fusion weights were refitted on the corrupted data, so these numbers describe the whole pipeline rather than a fixed model. Every level was repeated over five independent realizations, each with its own corruption mask and five-fold partition. Table 8 reports the mean over realizations with the standard deviation across them; that deviation ranged from 0.22% to 2.04%, the same order as the differences between adjacent corruption levels, indicating that individual non-monotonic steps carry no information. The trends survived averaging. Deleting an additional fifth of all attribute values left accuracy at 98.00%, and deleting two fifths left it at 97.00%, while corrupting a tenth of the training labels cost about 0.7 percentage points, and corrupting a fifth brought accuracy to 94.30%. The asymmetry is informative: missing values were handled by an imputer that could borrow from correlated attributes, whereas a corrupted label misdirected the wrapper, the base learners, and the fusion weights at once.

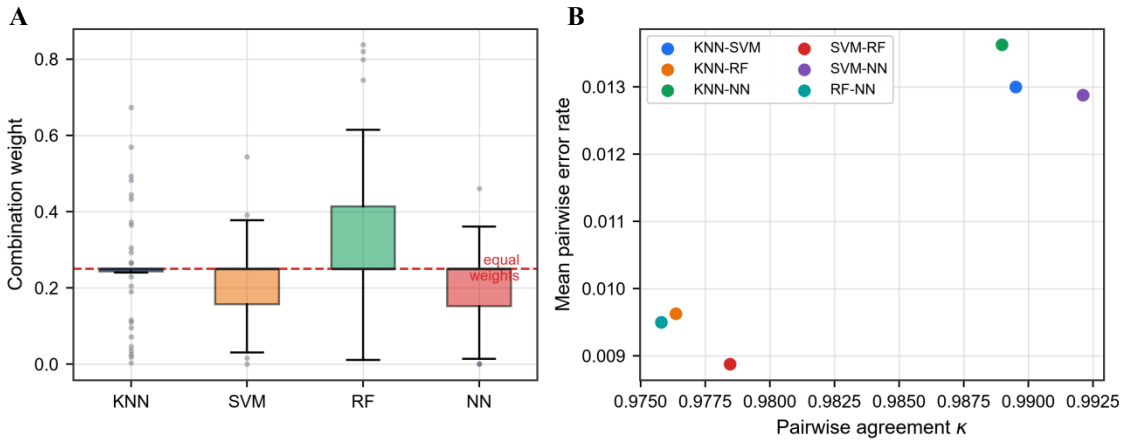


Figure 7. Ensemble combination weights and error–diversity analysis across the 50-fold evaluation. (A) Combination weights learned using the continuous swarm across the 50 folds. (B) Kappa-error diagram of the six member pairs; points toward the lower left combine low error with high diversity.
Abbreviations: KNN: K-nearest neighbor; NN: Neural network; RF: Random forest; SVM: Support vector machine.

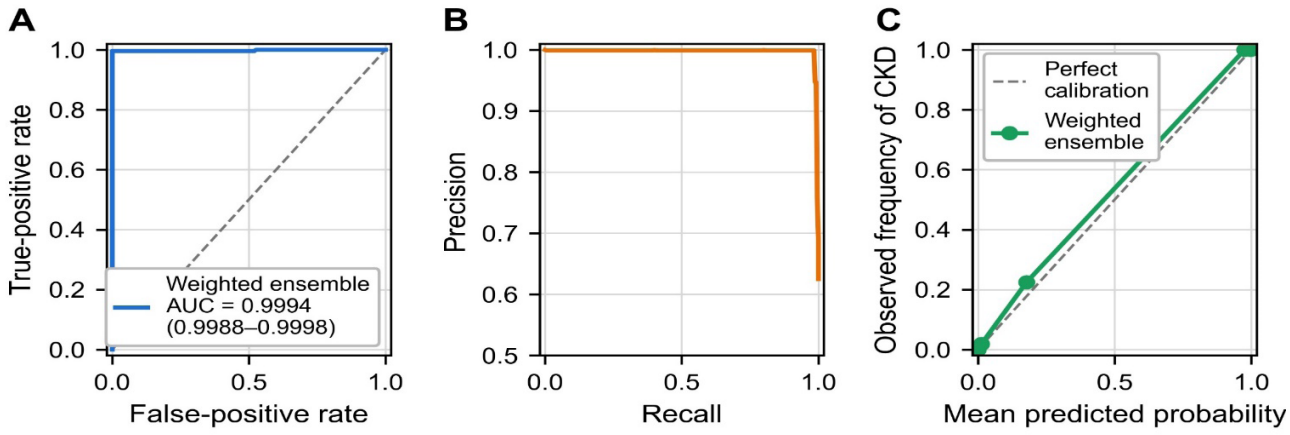


Figure 8. Discrimination and calibration of the pooled out-of-fold predictions of the first repetition. (A) Receiver operating characteristic. (B) Precision–recall curve. (C) Calibration of the pooled out-of-fold predictions of the first repetition.
Abbreviations: AUC: Area under the curve; CKD: Chronic kidney disease.

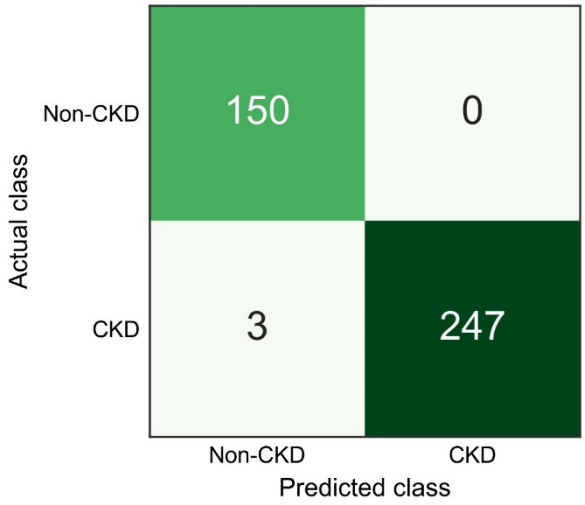


Figure 9. Confusion matrix of the pooled out-of-fold predictions
Abbreviation: CKD: Chronic kidney disease

Table 8. Degradation of the complete pipeline as missing values and label noise are injected

Condition	Accuracy (%)	AUC
0% additional missing values	98.90 ± 0.22	0.9977
5% additional missing values	98.90 ± 0.38	0.9989
10% additional missing values	99.00 ± 0.40	0.9992
20% additional missing values	98.00 ± 0.59	0.9975
30% additional missing values	98.05 ± 0.57	0.9952
40% additional missing values	97.00 ± 1.17	0.9900
5% of training labels corrupted	98.40 ± 0.82	0.9960
10% of training labels corrupted	98.25 ± 0.56	0.9957
15% of training labels corrupted	95.35 ± 1.32	0.9923
20% of training labels corrupted	94.30 ± 2.04	0.9883

Notes: Each cell is the mean over five independent corruption realizations, each a five-fold cross-validation in which the search, the base learners, and the fusion weights are refitted on the corrupted data; the deviation is across those five realizations, not across folds.
Abbreviations: AUC: Area under the curve.

3.8. External validation

Table 9 and Figure 11 report results on an independent cohort of 200 records. The weighted ensemble achieved 98.00% accuracy with an area under the curve of 0.9998, and majority voting achieved 99.00% on the same cohort. The gap is two records. More important than the ordering is that a pipeline fitted entirely on one cohort transfers to a second cohort recorded in a different format, with discretized intervals rather than raw measurements, with a performance decrease of approximately 1 percentage point relative to internal cross-validation. **Section 3.12** reports how much of the two-record gap between the two fusion rules can be attributed to the interval-to-scalar harmonization.

3.9. Computational cost and comparison with prior work

The complete pipeline required 401 seconds per fold on an 11-core workstation. The five searches accounted for 317 s, the base learners for 10.8 s, and the weight optimization for 1.7 s. Deploying the model requires only the latter two: fixing the attribute subset in advance required 12.5 seconds per fold, while inference using 11 attributes was immediate.

Table 10 presents the results alongside recent studies on the same cohort. The comparison is indicative rather than exact: studies differ in their validation protocol, whether attribute selection was nested inside the resampling loop, and how missing values were handled; few report an external cohort or the reproducibility of their selected subset.

Table 9 presents a context rather than a ranking. Optimization-driven selection and weighted or stacked ensembles are already present in the recent CKD literature, so assembling a metaheuristic, a wrapper objective, and a weighted vote is not the contribution claimed here; what the entries above have in common is that each reports one protocol in which the proposed method differs from its comparators in several respects at once, on a cohort where every reasonable method exceeds 98%. This study provides the counterfactual and an 11-attribute panel that maintains its accuracy on an independent cohort.

3.10. Comparison with classical feature selection

The ablation in **Section 3.2** compares the hybrid only with its own components. Table 11 adds four classical selectors under the identical protocol, each choosing its own subset size inside the training partition and feeding the same learners and fusion. All five landed within four-tenths of a percentage point of one another, and none of the differences survived correction at either level, so they are statistically equivalent; what separates them is cost and panel size. L1-regularized logistic regression achieved 99.35% from 11.20 attributes in 0.45 seconds per fold against 76.7% for the search; random-forest importance reached 99.00% from only 6.64 attributes, and recursive feature elimination is the exception on cost at 102 seconds. A study proposing a metaheuristic search on data of this kind should report this comparison.

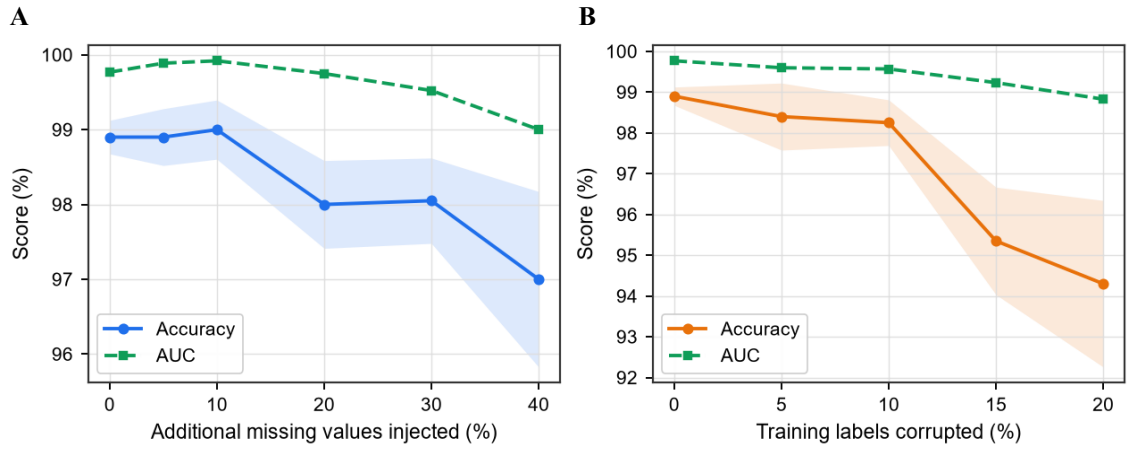


Figure 10. Behavior of the complete pipeline as (A) missing values and (B) label noise are injected. The search, the base learners, and the fusion weights are refitted at every level.
Abbreviations: AUC: Area under the curve.

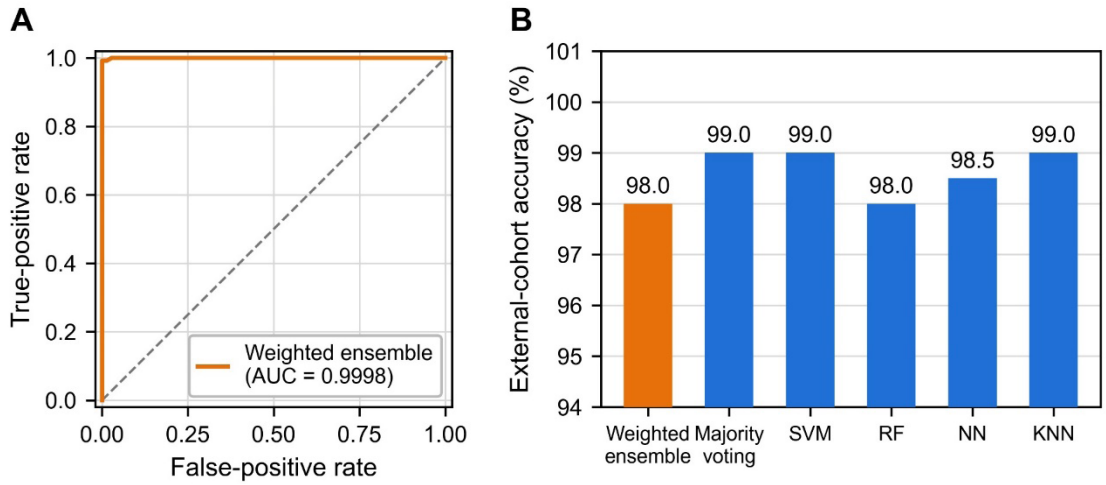


Figure 11. Performance on the independent cohort of 200 records. (A) External cohort ROC. (C) Transportability to an independent cohort.
Abbreviations: AUC: Area under the curve; KNN: K-nearest neighbor; NN: Neural network; RF: Random forest; ROC: Receiver operating characteristic; SVM: Support vector machine.

Table 9. External validation on an independent cohort of 200 records using the attributes and weights fitted on the primary cohort

Model	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	AUC
Swarm-weighted voting	98.00	96.97	100.00	94.44	98.46	0.9998
Majority voting	99.00	98.46	100.00	97.22	99.22	0.9998
Support vector machine	99.00	98.46	100.00	97.22	99.22	0.9998
Random forest	98.00	96.97	100.00	94.44	98.46	0.9998
Neural network	98.50	98.45	99.22	97.22	98.83	0.9997
K-nearest neighbors	99.00	98.46	100.00	97.22	99.22	0.9998

Note: The accuracy of the weighted ensemble on this cohort is 98.00% (95% bootstrap interval 96.00% to 99.50%).
Abbreviation: AUC: Area under the curve.

Table 10. Comparison with recent studies on the same chronic kidney disease cohort

Study	Approach	Attributes	Accuracy (%)
Dritsas and Trigka[19]	Class balancing with a rotation forest	24	99.2
Venkatesan *et al.*[17]	Ensemble preprocessing with machine learning	24	98.0
Chhabra *et al.*[14]	Stacking ensemble with synthetic oversampling	24	99.5
Al Ghamdi and Alyahyan[16]	Hybrid deep model on clinical biomarkers	24	98.7
Qureshi *et al.*[22]	Interpretable classifiers with SHAP and LIME	24	94.7
This work	Cooperative hybrid search with a swarm-weighted vote	11	99.00 98.00 (external)

Note: Validation protocols differ across studies, so the comparison is indicative rather than exact.
Abbreviations: LIME: Local Interpretable Model-agnostic Explanations; SHAP: SHapley Additive exPlanations.

Table 11. Classical feature selectors evaluated under the protocol used for the metaheuristic variants

Selector	Attributes	Selection time (s/fold)	Accuracy (%)	F1 (%)	AUC	p (Holm, 50 folds)	p (Holm, 10 repetitions)
Cooperative hybrid (proposed)	11.22	76.70	99.00 ± 1.16	99.19	0.9994	Reference	Reference
L1-regularized logistic regression	11.20	0.45	99.35 ± 0.92	99.47	0.9992	0.042	0.391
Recursive feature elimination, random forest	10.26	101.65	99.03 ± 1.24	99.21	0.9996	1.000	1.000
Mutual-information filter	9.92	1.22	98.98 ± 1.44	99.17	0.9993	1.000	1.000
Random-forest importance	6.64	1.07	99.00 ± 1.34	99.18	0.9997	1.000	1.000
No selection, all 24 attributes	24.00	0.00	99.67 ± 0.83	99.74	1.0000	0.001	0.039

Notes: Each selector chooses its own subset size using inner cross-validation within the training partition and feeds the identical four base learners and the identical out-of-fold weighted fusion. Selection time excludes the base learners and the fusion, which are common to every row. The last two columns give the Holm-adjusted Wilcoxon p-value against the cooperative hybrid over the 50 folds and over the 10 repetitions.
Abbreviation: AUC: Area under the curve.

3.11. Statistical analysis under the dependence of the folds

The 50-fold results of 10 repetitions of five-fold cross-validation are not 50 independent measurements. Table 3 repeats every comparison under two analyses that avoid that assumption: one averaged the five folds within each repetition, so the 10-repetition means are proper exchangeable blocks, and the other was the Nadeau–Bengio corrected resampled t-test. Under both, the methods compared here are statistically equivalent. The Friedman critical difference widened from 2.81 to 6.27 ranks and separated no pair, and the two comparisons significantly moved fold-wise to 0.102 for the unreduced panel and 0.375 for the random forest. Part of this is power, since a cohort on which every method exceeded 98.60% leaves five or six errors per fold to separate methods by; the grounds for preferring one method here are therefore cost and parsimony rather than discrimination.

3.12. Sensitivity of the external result to the harmonization

The external cohort records every continuous attribute as an interval, so a scalar must be chosen before scoring. That choice was treated as an experiment: the subset, the fitted learners, and the fusion weights were held fixed, and only the mapping changed. Table 12 reports the midpoint rule of **Section 2.1** alongside the lower edge, the upper edge, 20 uniform draws within each interval, and four codings of the diastolic indicator. Accuracy of the weighted ensemble spanned 97.00% to 99.50% across the four rules, so the external figures of **Section 3.8** should be read with that spread attached; the ordering of the fusion rules was unchanged under every mapping, and the coding of the diastolic indicator left accuracy identical.

3.13. Sensitivity to the cardinality weight of the objective

In **Section 2.4**, the trade-off parameter in **Equation 5** was fixed at 1, setting the cardinality weight to zero and leaving

parsimony to the consensus rule. Table 13 reports what the search returned across a range of that parameter over 10 outer folds per setting. The usable range was narrow: moving the weight from 0 to 0.01 cut the panel from 11.90 attributes to 7.70 attributes, with accuracy increasing from 99.13% to 99.25%, well within fold-to-fold deviations; whereas at 0.05, the panel collapsed to 4.40 attributes, and at 0.20 to 2.40, accuracy reduced to 97.88%. This is because when many subsets share the same inner error, the cardinality term becomes the only discriminating part of the objective. The results here were produced at zero weight, the conservative end of that range.

4. Discussion

Four findings emerge from the experiments, each stated at a level of strength supported by the evidence.

First, the pipeline delivers a compact panel at no measurable cost in discrimination: 11 attributes reproduced 99.00% where all 24 yielded 99.67%, and once fold dependence was modeled, the two were statistically equivalent. Dropping 13 assays is a substantial saving where laboratory access is the binding constraint but provides no savings where the full panel is measured regardless; thus, the rationale for selection is operational rather than based on accuracy advantage. Second, a weighted vote whose weights are learned from out-of-fold probabilities consistently outperformed an unweighted vote, achieving 99.00% versus 98.83% and 98.85%, respectively. However, the 0.175-percentage-point difference did not remain statistically significant after correction, indicating a consistent but not statistically demonstrated improvement. Third, the pipeline showed greater robustness to missing data than to label noise, maintaining 98.00% accuracy when an additional fifth of all values was deleted but decreasing to 94.30% when one-fifth of the training labels were corrupted. This asymmetry was evaluated using five independent corruption realizations per level.

Fourth, and most consequential for how work of this kind should be reported, the choice of selector did not substantially differentiate the methods in this study. Given the same objective, budget, and selection rule,

Table 12. External cohort of 200 records scored under alternative harmonizations of its interval-coded attributes

Harmonization of the interval-coded attributes	Weighted ensemble accuracy (%)	Majority voting accuracy (%)	Weighted ensemble AUC
Interval midpoint (the rule used in **Section 2.1**)	98.00	99.00	0.9998
Lower edge of every interval	97.00	97.00	1.0000
Upper edge of every interval	99.50	99.50	0.9999
Uniform draw inside every interval, 20 draws	98.40 ± 0.45	98.90 ± 0.21	0.9997
Diastolic indicator coded as 70 and 90 mmHg	98.00	99.00	0.9998
Diastolic indicator coded as 75 and 95 mmHg	98.00	99.00	0.9998
Diastolic indicator coded as 80 and 100 mmHg	98.00	99.00	0.9998
Diastolic indicator coded as 76.5 and 76.5 mmHg	98.00	99.00	0.9998

Notes: The selected subset, the four base learners, and the fusion weights are those fitted on the primary cohort and are held fixed; only the mapping changes. The uniform row is the mean and standard deviation over independent draws.
Abbreviation: AUC: Area under the curve.

Table 13. Effect of the cardinality weight in Equation 5 on the subset the cooperative search returns and on the accuracy of the resulting ensemble, over 10 outer folds per setting, with every other setting held fixed

Trade-off (Equation 5)	Cardinality weight	Attributes retained	Accuracy (%)	AUC
1.00	0.00	11.90 ± 1.85	99.13 ± 1.03	0.9982
0.99	0.01	7.70 ± 2.26	99.25 ± 0.87	0.9997
0.95	0.05	4.40 ± 0.84	98.87 ± 1.24	0.9977
0.90	0.10	3.70 ± 0.82	98.62 ± 1.09	0.9956
0.80	0.20	2.40 ± 0.52	97.88 ± 1.96	0.9893

Note: The configuration used throughout the paper is the first row; deviations are across folds.
Abbreviation: AUC: Area under the curve.

the genetic, swarm, and annealing variants finished within a fraction of a percentage point of each other and of the cooperative hybrid. **Section 3.10** extends this finding beyond the metaheuristic family: L1-regularized logistic regression selected a subset of the same size and achieved performance within 0.4 percentage points of the hybrid while requiring less than half a second per fold, compared with 76.7 seconds for the search, while random-forest importance matched the hybrid using one-third fewer attributes. Published comparisons on this dataset usually differ from their baselines in several respects simultaneously; when these differences are controlled, the observed performance ordering disappears. On cohorts of this size and separability, these results indicate that a regularized linear selector can provide similar predictive performance at substantially lower computational cost.

The ensemble showed a similar pattern. A random forest trained on the selected attributes achieved 99.45% accuracy, compared with 99.00% for the committee containing it; this difference was statistically significant in fold-wise testing but not in the dependence-aware analyses. The diversity measurements provide context for this result: the four members disagreed on 0.78% of records, with a mean interrater agreement of 0.984, indicating substantial agreement among the ensemble members. Thus, the limited diversity may constrain the potential benefit of combining their predictions. This pattern may reflect characteristics of the cohort as well as the ensemble design, given the high separability of the classes and the similar decision boundaries produced by the evaluated learners. Future studies could examine whether greater predictive diversity can be achieved by training members on different attribute subsets or resampled datasets.

The attributes selected using the search are clinically coherent. Specific gravity and hemoglobin were selected in almost every fold, reflecting impaired urinary concentration and anemia associated with reduced erythropoietin production. Hypertension and diabetes mellitus were selected directly as the two dominant causes of the disease. Packed cell volume provides information related to the hemoglobin signal, while appetite and pedal edema are clinical manifestations associated with uremia and fluid retention.

Several limitations bound these conclusions. The primary cohort contains 400 records from a single hospital and is more separable than a screening population: every method exceeds 98.60% accuracy, leaving five or six errors per fold with which to distinguish them. Thus, the equivalences reported here describe the resolution of this cohort rather than showing that no differences would exist in a larger cohort, and accuracies near 99% should not be read as field performance. The external cohort is a discretized re-encoding deposited in the same repository rather than a sample from a different population; therefore, **Section 3.8** measures transfer across an encoding rather than across settings, while **Section 3.12** shows accuracy varying by roughly 2.5 percentage points across defensible interval mappings. Because the labels are binary, these analyses do not address disease staging. The wrapper uses a k-nearest neighbor classifier, whereas the target model is a four-member ensemble; aligning the two would require evaluating the ensemble within the search, which was not feasible at this budget. The fusion weights are constrained to a simplex over four fixed members, the search budget was fixed in advance for comparability, and **Section 3.13** shows that zero cardinality weight represents the conservative end of a narrow usable range.

5. Conclusion

A cooperative hybrid metaheuristic for attribute selection was combined with a weighted voting ensemble for CKD prediction and evaluated against its component methods, four classical selectors, and the unreduced panel under an identical protocol. Genetic, swarm, and annealing subpopulations evolved together under a shared objective, exchanged elites during the run, and were refined by annealing as a memetic local search. The reported subset was the consensus of the elite band rather than a single incumbent, and four heterogeneous learners were then fused using weights tuned on out-of-fold probabilities. Under a 10-times, five-fold protocol confining imputation, search, learners, and weight optimization to the training partition, the ensemble achieved 99.00% accuracy, an area under the curve of 0.9994, an F-measure of 99.19%, and a Matthews correlation coefficient of 0.9793 using 11 of the 24 attributes. It maintained 98.00% accuracy when an additional fifth of all values was deleted and achieved 98.00% accuracy on an independent cohort of 200 records recorded in a different format.

Three further findings qualify these results. Given the same objective, budget, and selection rule, the five metaheuristics showed no statistically significant differences in performance, as did the four classical selectors, three of which required less than one-fiftieth as much. Attribute selection incurred a small and non-significant reduction in accuracy relative to the full panel, making the reduction in measurement burden its primary practical benefit. The four ensemble members also disagreed on fewer than 1% of records, indicating limited predictive diversity and, consequently, limited potential benefit from fusion. Together, these findings support a compact, reproducible approach for this cohort, consisting of an 11-attribute panel evaluated using a four-member

ensemble whose components showed similar predictive performance.

Acknowledgments

None.

Funding

None.

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: Amaresh Ayyanahalli Math, AnandaMurthy Meenakshi Sundaram
Formal analysis: Amaresh Ayyanahalli Math
Investigation: Amaresh Ayyanahalli Math, AnandaMurthy Meenakshi Sundaram
Methodology: Amaresh Ayyanahalli Math
Writing–original draft: Amaresh Ayyanahalli Math
Writing–review & editing: Amaresh Ayyanahalli Math, AnandaMurthy Meenakshi Sundaram

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

Both cohorts are publicly available from the University of California Irvine Machine Learning Repository under the identifiers 336 and 857. The analysis code and the random seeds needed to reproduce every number reported here are available from the corresponding author on request and are deposited in a public repository (URL).

References

1. Jadoul M, Aoun M, Masimango Imani M. The major global burden of chronic kidney disease. *Lancet Glob Health.* 2024;12(3):e342-e343.

   doi: 10.1016/S2214-109X(24)00050-0

2. Wang L, He Y, Han C, et al. Global burden of chronic kidney disease and risk factors, 1990-2021: an update from the Global Burden of Disease Study 2021. *Front Public Health.* 2025;13:1542329.

   doi: 10.3389/fpubh.2025.1542329

3. Ke C, Liang J, Liu M, Liu S, Wang C. Burden of chronic kidney disease and its risk-attributable burden in 137 low- and middle-income countries, 1990-2019: results from the Global Burden of Disease Study 2019. *BMC Nephrol.* 2022;23(1):17.

   doi: 10.1186/s12882-021-02597-3

4. Stephan T, Mishra V. A penalty-embedded genetic algorithm for interpretable clinical feature selection in heart disease prediction. *Healthc Anal.* 2026;9:100458.

   doi: 10.1016/j.health.2026.100458

5. Han F, Chen WT, Ling QH, Han H. Multi-objective particle swarm optimization with adaptive strategies for feature selection. *Swarm Evol Comput.* 2021;62:100847.

   doi: 10.1016/j.swevo.2021.100847

6. Abdel-Basset M, Sallam KM, Mohamed R, Elgendi I, Munasinghe K, Elkomy OM. An improved binary grey-wolf optimizer with simulated annealing for feature selection. *IEEE Access.* 2021;9:139792-139822.

   doi: 10.1109/ACCESS.2021.3117853

7. Pashaei E, Pashaei E. Hybrid binary COOT algorithm with simulated annealing for feature selection in high-dimensional microarray data. *Neural Comput Appl.* 2023;35(1):353-374.

   doi: 10.1007/s00521-022-07780-7

8. Akinola OA, Ezugwu AE, Oyelade ON, Agushaka JO. A hybrid binary dwarf mongoose optimization algorithm with simulated annealing for feature selection on high dimensional multi-class datasets. *Sci Rep.* 2022;12(1):14945.

   doi: 10.1038/s41598-022-18993-0

9. Bandyopadhyay R, Basu A, Cuevas E, Sarkar R. Harris Hawks optimisation with simulated annealing as a deep feature selection method for screening of COVID-19 CT-scans. *Appl Soft Comput.* 2021;111:107698.

   doi: 10.1016/j.asoc.2021.107698

10. Chandra TB, Verma K, Singh BK, Jain D, Netam SS. Coronavirus disease (COVID-19) detection in chest X-ray images using majority voting based classifier ensemble. *Expert Syst Appl.* 2021;165:113909.

    doi: 10.1016/j.eswa.2020.113909

11. Maz YA, Anbar M, Manickam S, Rihan SDA, Alabsi BA, Dorgham OM. Majority voting ensemble classifier for detecting keylogging attack on Internet of Things. *IEEE Access.* 2024;12:19860-19871.

    doi: 10.1109/ACCESS.2024.3362232

12. Ennaji A, Sabri MA, Aarab A. Ensemble learning with weighted voting classifier for melanoma diagnosis. *Multimed Tools Appl.* 2025;84(9):6073-6089.

    doi: 10.1007/s11042-024-19143-6

13. Okuboyejo DA, Olugbara OO. Classification of skin lesions using weighted majority voting ensemble deep learning. *Algorithms.* 2022;15(12):443.

doi: 10.3390/a15120443

14. Chhabra D, Juneja M, Chutani G. An efficient ensemble-based machine learning approach for predicting chronic kidney disease. *Curr Med Imaging.* 2024;20:e080523216634.

doi: 10.2174/1573405620666230508104538

15. Chen H, Huang Y, Chen L. Ensemble machine learning for predicting renal function decline in chronic kidney disease: development and external validation. *Front Med.* 2025;12:1598065.

doi: 10.3389/fmed.2025.1598065

16. Al Ghamdi M, Alyahyan S. Enhanced chronic kidney disease detection: a hybrid deep learning framework using clinical biomarkers and ensemble feature engineering with DeepCKD-Net. *Appl Sci.* 2026;16(6):3024.

doi: 10.3390/app16063024

17. Venkatesan VK, Ramakrishna MT, Izonin I, Tkachenko R, Havryliuk M. Efficient data preprocessing with ensemble machine learning technique for the early detection of chronic kidney disease. *Appl Sci.* 2023;13(5):2885.

doi: 10.3390/app13052885

18. Sengothai R, Sivaraman R. Early prediction of chronic kidney disease via ensemble-deep-learning approach and improved dimensionality reduction approach. *Int J Math Oper Res.* 2024;28(4):431-454.

doi: 10.1504/ijmor.2024.140348

19. Dritsas E, Trigka M. Machine learning techniques for chronic kidney disease risk prediction. *Big Data Cogn Comput.* 2022;6(3):98.

doi: 10.3390/bdcc6030098

20. Debal DA, Sitote TM. Chronic kidney disease prediction using machine learning techniques. *J Big Data.* 2022;9(1):109.

doi: 10.1186/s40537-022-00657-5

21. Arif MS, Rehman AU, Asif D. Explainable machine learning model for chronic kidney disease prediction. *Algorithms.* 2024;17(10):443.

doi: 10.3390/a17100443

22. Qureshi M, Azad A, Iftikhar H, Rodrigues PC. Toward explainable precision nephrology: machine learning-based chronic kidney disease prediction. *Biomedicines.* 2026;14(7):1459.

doi: 10.3390/biomedicines14071459

23. Taji K, Sohail A, Shahzad T, Shoaib Khan B, Adnan Khan M, Ouahada K. An ensemble hybrid framework: a comparative analysis of metaheuristic algorithms for ensemble hybrid CNN features for plants disease classification. *IEEE Access.* 2024;12:61886-61906.

doi: 10.1109/ACCESS.2024.3389648

24. Mandal AK, Nadim M, Saha H, Sultana T, Hossain MD, Huh EN. Feature subset selection for high-dimensional, low sampling size data classification using ensemble feature selection with a wrapper-based search. *IEEE Access.* 2024;12:62341-62357.

doi: 10.1109/ACCESS.2024.3390684

25. Feng G. Feature selection algorithm based on optimized genetic algorithm and the application in high-dimensional data processing. *PLoS One.* 2024;19(5):e0303088.

doi: 10.1371/journal.pone.0303088

26. Zhou J, Hua Z. A correlation guided genetic algorithm and its application to feature selection. *Appl Soft Comput.* 2022;123:108964.

doi: 10.1016/j.asoc.2022.108964

27. Saibene A, Gasparini F. Genetic algorithm for feature selection of EEG heterogeneous data. *Expert Syst Appl.* 2023;217:119488.

doi: 10.1016/j.eswa.2022.119488

28. Naskar A, Ghosh S, Kundu M, Sarkar R. Feature selection using guided population based genetic algorithm with modified crossover and parent selection. *Appl Soft Comput.* 2025;172:112872.

doi: 10.1016/j.asoc.2025.112872

29. Li AD, Xue B, Zhang M. Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. *Appl Soft Comput.* 2021;106:107302.

doi: 10.1016/j.asoc.2021.107302

30. Huang W, Tian H, Wang S, Zhang C, Zhang X. Integration of simulated annealing into pigeon inspired optimizer algorithm for feature selection in network intrusion detection systems. *PeerJ Comput Sci.* 2024;10:e2176.

doi: 10.7717/peerj-cs.2176

31. Kiani Sarkaleh M, Azgomi H, Kiani-Sarkaleh A. Breast cancer classification using feature selection via improved simulated annealing and SVM classifier. *Diagnostics.* 2026;16(4):637.

doi: 10.3390/diagnostics16040637

32. Sharifai AG, Zainol ZB. Multiple filter-based rankers to guide hybrid grasshopper optimization algorithm and simulated annealing for feature selection with high dimensional multi-class imbalanced datasets. *IEEE Access.* 2021;9:74127-74142.

doi: 10.1109/ACCESS.2021.3081366

33. Özkan Y, Özçelik YB, Altan A. Optimization-driven hybrid machine learning framework for brain tumor classification in MRI with metaheuristic feature selection. *Diagnostics.* 2026;16(5):819.

doi: 10.3390/diagnostics16050819

34. Alyasiri OM, Cheah YN, Abasi AK, Al-Janabi OM. Wrapper

and hybrid feature selection methods using metaheuristic algorithms for English text classification: a systematic review. *IEEE Access.* 2022;10:39833-39852.

doi: 10.1109/ACCESS.2022.3165814

35. Chaudhuri A. Search space division method for wrapper feature selection on high-dimensional data classification. *Knowl Based Syst.* 2024;291:111578.

doi: 10.1016/j.knosys.2024.111578

36. Guo H, Ma J, Wang R, Zhou Y. Feature library-assisted surrogate model for evolutionary wrapper-based feature selection and classification. *Appl Soft Comput.* 2023;139:110241.

doi: 10.1016/j.asoc.2023.110241

37. Chamseddine E, Mansouri N, Soui M, Abed M. Handling class imbalance in COVID-19 chest X-ray images classification: using SMOTE and weighted loss. *Appl Soft Comput.* 2022;129:109588.

doi: 10.1016/j.asoc.2022.109588

38. Yang Y, Mirzaei G. Performance analysis of data resampling on class imbalance and classification techniques on multi-omics data for cancer classification. *PLoS One.* 2024;19(2):e0293607.

doi: 10.1371/journal.pone.0293607

39. Ebiaredoh-Mienye SA, Swart TG, Esenogho E, Mienye ID. A machine learning method with filter-based feature selection for improved prediction of chronic kidney disease. *Bioengineering.* 2022;9(8):350.

doi: 10.3390/bioengineering9080350

40. N Y, Shrinivasacharya P, Naik N. Novel statistically equivalent signature-based hybrid feature selection and ensemble deep learning LSTM and GRU for chronic kidney disease classification. *PeerJ Comput Sci.* 2024;10:e2467.

doi: 10.7717/peerj-cs.2467

41. Mandal M, Singh PK, Ijaz MF, Shafi J, Sarkar R. A tri-stage wrapper-filter feature selection framework for disease classification. *Sensors.* 2021;21(16):5571.

doi: 10.3390/s21165571

Appendix

Tables A1 and A2 present the complete settings used for every experiment, so that the results can be reproduced exactly. They are referenced in **Section 2.7**. Identical settings were used for every variant of the ablation, which differed only in which sub-populations were active, and nothing was tuned on the test partition of any fold. The search averaged 672 distinct subsets per fold for the cooperative hybrid and between 552 and 736 for the single-operator and two-operator variants.

Table A1. Complete settings of the cooperative search and of the combination-weight optimizer

Component	Settings
Cooperative search	Population 24 as three sub-populations of 8 (one of 24 for a single operator); 30 iterations, fixed, no early stopping; elite migration every 5 iterations, best of each replacing the worst of its neighbor; Bernoulli (0.5) initialization, all-zero repaired
Genetic operators	Binary tournament; two-point crossover $p = 0.8$; bitwise mutation $p = 0.05$; two fittest carried forward
Swarm operators	Inertia 0.9 to 0.4; cognitive 2.5 to 0.5; social 0.5 to 2.5, all linear; velocity clamp −6 to 6; V-shaped transfer |tanh(v)|
Annealing operators	T0 = 2|N| = 48; cooling T < −0.93 T once per iteration; neighbor flips max(1, 0.1|N|) = 2 bits; Metropolis acceptance exp(-theta/T); stopped by the shared 30-iteration budget, not a temperature floor; memetic local search of three single-bit greedy trials on the incumbent per iteration
Objective and selection	zeta = 1.0, so cardinality weight 1 − zeta = 0; ranking weight 0.7 in **Equation 6**; inner resampling stratified 5-fold × 2 repeats; wrapper k-NN, k = 7, distance weights; elite band the best 10% of evaluated subsets, attribute retained above 50% of the band
Weight optimizer	20 particles, 40 iterations; uniform vector seeded as particle 1; inertia 0.9 to 0.4; acceleration 1.5 and 1.5; velocity clamp −0.3 to 0.3; criterion balanced accuracy on out-of-fold probabilities

Table A2. Complete settings of the imputer and the four base learners

Component	Setting
Scaling and imputation	Min–max to the unit interval, then iterative (missForest-style) imputation with an extra-tree regressor, eight rounds, mean initialization; both fitted on the training partition
K-nearest neighbors	k = 5, distance weighting, Euclidean
Support vector machine	RBF kernel, C = 10, gamma = scale, balanced class weights, probability estimates enabled
Random forest	300 trees, Gini, unlimited depth, sqrt(p) attributes per split
Neural network	Hidden layers of 64 and 32 units, ReLU, Adam, learning rate 0.001, up to 1,200 epochs
Stacking comparator	The same four members, logistic regression meta-learner, inner five-fold
Tuning procedure	None of the four was tuned per fold; the settings above were fixed once, before the evaluation, from ranges reported for this cohort in prior work[14,17,19,20]

Notes: Values not listed are the scikit-learn 1.9.0 defaults. Randomness. The global seed is 42, and outer fold i uses seed 42 + i for the imputer, the search, every base learner, the inner splits, and the weight optimizer. Hence, a fold is reproducible in isolation. The corruption study in **Section 3.7** uses seeds 42 + 1,000r + i for realization r and fold i. The environment is Python 3.14.6, scikit-learn 1.9.0, NumPy 2.5.2, SciPy 1.18.0, pandas 3.0.3, and joblib 1.5.3.