Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 15 March 2026
Revised 20 April 2026
Accepted 14 May 2026

Natural Resources for Human Health 2026, Vol. 6 (1s): 312–332

KEYWORDS:
chronic kidney disease, feature selection, particle swarm optimisation, recursive feature elimination, support vector machine, external validation

An Intelligent Healthcare Framework for Early Chronic Kidney Disease Screening using Swarm-Optimized Feature Selection

A. M. Amaresh[1], A. Meenakshi Sundaram[2]

[1]Research Scholar, School of Computer Science and Engineering, REVA University, Bangalore, Karnataka, India
[2]Associate Professor, School of Computer Science and Engineering, REVA University, Bangalore, Karnataka, India
E-mail: amaresh.am03@gmail.com[1], meenakshi.sa@reva.edu.in[2]

ABSTRACT: Chronic kidney disease progresses silently, and the laboratory panel used to detect it is broad, so any screening model that reaches a diagnosis from fewer measurements has practical value. This work develops a five stage selection pipeline and pairs it with a supervised support vector machine. Mutual information first removes attributes that carry almost no information about the class. A hybrid weighted recursive elimination stage then ranks the survivors by fusing rank normalised importances taken from a linear support vector machine, a random forest and a gradient boosting model, each weighted by its own inner cross validation accuracy. Profiling the ranked prefixes supplies a starting point for a chaotic adaptive binary particle swarm that searches the attribute mask and the kernel parameters at the same time, and the attributes retained across the elite band of visited solutions form the final subset. Every stage is fitted inside the training partition of a 10 times 5 fold cross validation. The model reached 99.30 percent accuracy with an area under the curve of 0.9987 while using 14 of the 24 attributes, a result statistically indistinguishable from using the complete panel and ahead of recursive elimination, a genetic algorithm and a conventional binary swarm evaluated under the identical objective. On an independent cohort of 200 records the same subset returned 98.00 percent accuracy.

1. INTRODUCTION

Kidney failure has become a public health problem of a scale that few clinicians anticipated three decades ago. Chronic kidney disease (CKD) now affects roughly a tenth of the world population, and the most recent burden of disease estimates attribute more than a million deaths a year to it [1], [2]. The burden falls unevenly. Low and middle income countries carry a disproportionate share, and in those settings the diagnostic infrastructure that would allow early detection is often the resource in shortest supply [3]. Projections place CKD among the leading causes of death worldwide within the next fifteen years.

What makes the disease difficult is its silence. Kidney function can fall a long way before a patient notices anything, and by the time symptoms appear the damage is frequently irreversible. Diagnosis therefore rests on laboratory evidence: an estimated glomerular filtration rate below 60 mL/min/1.73 m^2, or markers of kidney damage such as albuminuria, that persist for three months or longer. Screening programmes built on that evidence work, but they require a broad panel of blood and urine assays. In a primary care clinic with a limited budget, the number of assays a screening rule needs is not a technical detail. It decides whether the rule is used at all.

Machine learning has been applied to this problem for more than a decade, and the reported accuracies on public CKD cohorts are high [4], [5], [6]. Two problems recur in that literature. The first concerns the protocol. Attribute selection is frequently carried out once on the pooled dataset and only afterwards is cross validation applied to the classifier, which lets

information from the test partitions influence which attributes are chosen. The optimism this introduces is not small, and we quantify it below. The second concerns what is actually being optimised. Wrapper searches on this cohort are usually driven by classification accuracy, and accuracy on CKD data saturates: a large fraction of candidate subsets achieve identical inner cross validation accuracy, so the search receives no usable gradient and the tie is broken by whatever secondary term the fitness function happens to contain, which is normally subset size. The result is a subset that is small rather than good.

This study keeps the objective of the original design, namely a swarm optimised attribute subset feeding a supervised classifier, and rebuilds the machinery around those two problems. The contributions are as follows.

- A filter, ranking and swarm cascade in which mutual information screening, a hybrid weighted recursive elimination stage and a chaotic adaptive binary particle swarm are chained so that each stage informs the next rather than operating independently.
- A wrapper fitness that couples the area under the curve with balanced accuracy. Because the ranking term is continuous it stays informative in the regime where accuracy has saturated, which is precisely the regime this cohort occupies.
- Simultaneous optimisation of the attribute mask and the radial basis kernel parameters, so that the kernel is retuned for every candidate subset instead of being held fixed while the subset changes underneath it.
- An elite archive consensus rule that replaces the single incumbent solution with the attributes retained across the best band of visited solutions, which raises both accuracy and the reproducibility of the selected subset.
- An evaluation protocol that keeps every fitted component inside the training partition, reports Wilcoxon and McNemar tests against nine competing configurations, measures selection stability, quantifies the optimism of the widely used leaky protocol, and validates the final subset on a second cohort of 200 records that took no part in training.

The rest of the article is organised as follows. Section 2 reviews recent work on attribute selection for clinical data and on metaheuristic search. Section 3 describes the cohorts and each stage of the method. Section 4 reports the experiments. Section 5 discusses what the results mean and where they stop, and Section 6 concludes.

2. RELATED WORK

Machine learning for chronic kidney disease. Dritsas and Trigka compared a wide set of classifiers for CKD risk prediction after class balancing and reported a rotation forest as the strongest model [4]. Debal and Sitote examined stage prediction with recursive elimination and univariate selection [5]. Explainable formulations have appeared more recently: Ahmed et al. coupled tree ensembles with attribution methods to expose which attributes drive a CKD decision [6], and comparable work in precision nephrology has combined SHAP and LIME with conventional classifiers [7]. Electronic health record cohorts have been used for incident CKD prediction at scale, with a systematic review by Haris et al. summarising model performance and, importantly, the variability of the validation protocols involved [8], [9], [10]. Analyses of cardiovascular risk within CKD populations have taken a similar route [11].

Attribute selection for clinical data. Reducing the measurement panel is a recurring theme. Alsekait et al. applied a snake optimisation algorithm to kidney and liver data and reported a large reduction in the attribute count [12]. A survey by Remeseiro and Bolon-Canedo covers filter, wrapper and embedded families as they apply to chronic disease diagnosis [13]. Filter based reduction combined with a cost sensitive boosting classifier has also been reported for this cohort [14], and hybrid signature based selection has been paired with deep sequence models [15]. Deep architectures applied to CKD, including autoencoder and support vector hybrids and imaging based pipelines, take the complementary route of learning representations rather than pruning inputs [16], [17]. Ensemble preprocessing pipelines have likewise been proposed for early detection [18].

Swarm and evolutionary search. Particle swarm optimisation remains the most widely used wrapper engine for attribute selection. Xue et al. developed multi objective swarm variants with adaptive strategies [19], and rank based particle distance formulations have been proposed to improve the exploitation of promising regions [20]. Improvements to the binary form of

the algorithm concentrate on initialisation and on the transfer function that maps velocity to a bit flip [21], [22], [23]. Adaptive binary variants with hybrid learning strategies [24], dual encoding schemes [25] and leader adaptive designs with explicit dimensionality reduction [26] all report gains over the classical formulation. Guided swarms have been applied to genomic data [27] and diversity driven multi strategy variants to general global optimisation [28]. Rough set theory has been combined with an improved swarm for the same purpose [29]. Tri stage designs that chain a filter to a wrapper are close in spirit to the cascade used here [30], as is wrapper based selection with a metaheuristic engine for early diabetes prediction [31] and metaheuristic search for cancer classification [32].

Recursive elimination and stability. Recursive elimination driven by a support vector machine remains a strong baseline and continues to be extended, for instance with Gaussian recursive variants for liver disease [33] and with applications in clinical risk modelling [34], [35]. A concern that receives less attention than accuracy is whether a selection algorithm returns the same attributes when the data are resampled. Nogueira et al. formalised ranking based stability measures [36], and later work has shown that unstable selection undermines the interpretation of a classifier even when its accuracy is unaffected [37], [38]. Hybrid metaheuristic designs combined with tree ensembles have been evaluated on other clinical prediction tasks [39].

Two gaps emerge. Most CKD studies report a single accuracy figure without stating whether selection was nested inside the resampling loop, and almost none report how reproducible the selected subset is. Both are addressed here.

3. MATERIALS AND METHODS

3.1 Cohorts

The primary cohort is the chronic kidney disease dataset of the University of California Irvine machine learning repository, collected over a two month period at a hospital in Tamil Nadu, India. It holds 400 records described by 24 clinical attributes together with a binary label, 250 of which are CKD and 150 non CKD. Eleven attributes are numerical and thirteen are nominal. Missing values are pervasive: 778 of the 9600 attribute values are absent, and the red blood cell count is missing in about a third of the records. Table 1 lists the attributes, their observed ranges and the amount of missing data.

An independent cohort is used for external validation. It contains 200 records described by the same 24 attributes, 128 of them CKD, and is distributed as the risk factor prediction of chronic kidney disease dataset of the same repository. Its values are recorded as discretised intervals rather than raw measurements, so each interval was mapped to a representative value before scoring: closed intervals to their midpoint, and open intervals to the boundary displaced by half of the median width of the closed intervals of that attribute. Diastolic pressure is recorded there as a binary indicator of an elevated reading and was mapped to the representative normal and elevated values of the primary cohort. This cohort takes no part in training, tuning or attribute selection. It is scored once, at the end.

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

Attribute	Code	Type	Observed range	Mean ± SD	Missing values
Age	age	numerical	2 to 90	51.48 ± 17.17	9 (2.2%)
Blood pressure	bp	numerical	50 to 180	76.47 ± 13.68	12 (3.0%)
Specific gravity	sg	numerical	1 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%)

Attribute	Code	Type	Observed range	Mean ± SD	Missing values
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%)
Haemoglobin	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	2200 to 26400	8406.12 ± 2944.47	106 (26.5%)
Red blood cell count	rbcc	numerical	2.1 to 8	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%)
Anaemia	ane	nominal	-	-	0 (0.0%)

3.2 Overview

Let $D = \{(x_i, y_i)\}_{i=1}^{n}$ be the cohort, with $x_i \in \mathbb{R}^p$, $p = 24$, and $y_i \in \{0,1\}$ where 1 denotes CKD. The task is to find a mask $s \in \{0,1\}^p$ and a classifier h such that h restricted to the attributes selected by s generalises well while $\|s\|_1 \ll p$. The pipeline realises this in five stages, shown in Figure 1: mutual information screening, hybrid weighted recursive elimination, prefix profiling, a chaotic adaptive binary particle swarm with joint kernel optimisation, and an elite archive consensus. A radial basis support vector machine is fitted on the resulting subset.

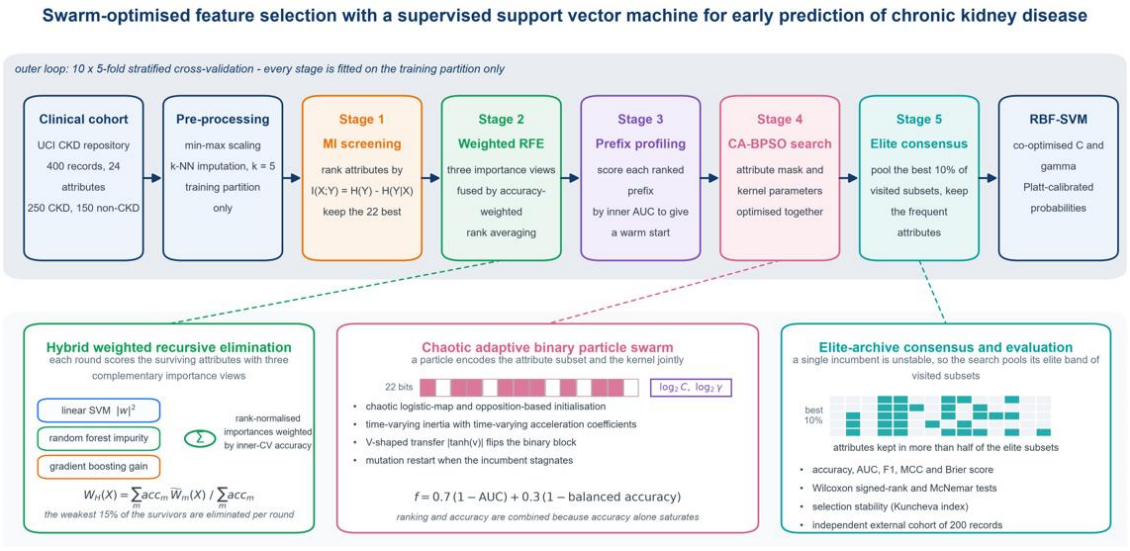

Fig. 1: The five stage pipeline. Screening, ranking, prefix profiling, the swarm search and the consensus rule are all fitted inside the training partition of every cross validation fold.

3.3 Pre-processing

Attributes are scaled to the unit interval by

$$x_j' = \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)} \tag{1}$$

with the minimum and maximum taken over the training partition only. Missing entries are then imputed by a distance weighted k nearest neighbour imputer with $k = 5$, again fitted on the training partition. Test partition values are transformed with the fitted objects and clipped to the unit interval. Scaling precedes imputation because the imputer relies on Euclidean distances, which would otherwise be dominated by attributes measured on large scales such as the white blood cell count.

3.4 Stage one: mutual information screening

Mutual information measures the dependence between an attribute and the label without assuming a functional form. For random variables X and Y,

$$I(X;Y) = H(Y) - H(Y|X) = \sum_{x \in X,\, y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{2}$$

where $H(\cdot)$ is entropy and $H(Y|X)$ is conditional entropy. The quantity is zero when the attribute and the label are independent and grows with dependence. It is also related to the entropies by

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{3}$$

Attributes are ranked by $I(X_j;Y)$ and the 22 highest are retained. The screen is deliberately mild. Its purpose is to remove attributes that carry essentially no class information before the expensive stages begin, not to make the selection decision.

3.5 Stage two: hybrid weighted recursive elimination

Recursive elimination repeatedly fits a model, discards the attributes it deems least important and refits. Its weakness is that the notion of importance belongs to one model. A linear support vector machine measures importance through the squared weight of a separating hyperplane, a random forest through impurity reduction, and gradient boosting through the gain of the splits it selects; the three disagree on this cohort. The elimination stage used here fuses them.

For the linear support vector machine, the decision function is $D(x) = \operatorname{sign}(x \cdot w)$ and the importance of attribute j is w_j^2. For the random forest, the importance of attribute X_j is the mean decrease in accuracy across the B trees of the forest,

$$W_R(X_j) = \frac{\sum_{t \in B} I^{(t)}(X_j)}{n_{tree}} \tag{4}$$

and for gradient boosting the importance follows the Gini criterion

$$W_G = 1 - \sum_{c=1}^{K} p_c^2 \tag{5}$$

accumulated over the splits that use the attribute. These three quantities live on incompatible scales, so combining them directly, as the weighted sum of the original design did, lets whichever model produces the largest raw numbers dominate the fusion. Each importance vector is therefore replaced by its rank within its own model,

$$\tilde{W}_m(X_j) = \frac{\operatorname{rank}_m(j) - 1}{|A| - 1} \tag{6}$$

where A is the set of surviving attributes, and the rank normalised vectors are averaged with weights given by the inner cross validation accuracy of the model that produced them,

$$W_H(X_j) = \frac{\sum_m acc_m \, \tilde{W}_m(X_j)}{\sum_m acc_m} \tag{7}$$

A model that classifies the training partition poorly therefore contributes little to the ranking. At each round the weakest 15 percent of the surviving attributes are eliminated, and the order of elimination, reversed, gives a complete ranking of the 22 screened attributes.

3.6 Stage three: prefix profiling

The ranking says which attributes matter but not how many to keep. Each ranked prefix of size k is therefore scored by repeated stratified inner cross validation, and the smallest prefix whose area under the curve lies within one standard error of the best is taken,

$$k^* = \min\left\{k: \mathrm{AUC}(k) \geq \max_{k'} \mathrm{AUC}(k') - \mathrm{SE}\right\} \tag{8}$$

The prefix of size k^* is not the final answer. It is one particle of the swarm described next, and its role is to place the search in a sensible region of a space with 2^{22} points.

3.7 Stage four: chaotic adaptive binary particle swarm with joint kernel optimisation

Particle swarm optimisation moves a population of candidate solutions through a search space under the influence of each particle's own best position and the best position found by the swarm. In the binary form used for attribute selection, each particle carries a bit per attribute. Four departures from the classical algorithm are made here.

Encoding. A particle is not a bit string alone. It carries 22 binary coordinates and two real coordinates holding $\log_2 C$ and $\log_2 \gamma$, the regularisation and width parameters of the radial basis kernel. Position and velocity are therefore

$$x_i^t = (b_{i1}^t, \ldots, b_{iD}^t,\ \log_2 C_i^t,\ \log_2 \gamma_i^t), \qquad v_i^t = \left(v_{i1}^t, \ldots, v_{i,D+2}^t\right) \tag{9}$$

Selecting attributes while holding the kernel fixed is inconsistent, because the optimal kernel width depends on the dimensionality of the space being searched. Optimising both together removes that inconsistency.

Initialisation. Half of the swarm is initialised from a logistic map, $z_{k+1} = 4z_k(1 - z_k)$, thresholded at 0.5, and the other half is the bitwise complement of the first. Chaotic sequences spread more evenly over the unit interval than pseudo random draws, and the opposition based complement guarantees that both dense and sparse subsets are represented from the first iteration. One particle is set to the prefix of Section 3.6 and one to the full screened pool.

Velocity update. The inertia weight decreases linearly and the acceleration coefficients vary with time, so that the swarm explores early and exploits late,

$$w^t = w_{max} - (w_{max} - w_{min})\frac{t}{T}, \qquad c_1^t = c_{1i} - \left(c_{1i} - c_{1f}\right)\frac{t}{T}, \qquad c_2^t = c_{2i} + \left(c_{2f} - c_{2i}\right)\frac{t}{T} \tag{10}$$

with w from 0.9 to 0.4, c_1 from 2.5 to 0.5 and c_2 from 0.5 to 2.5. The velocity is then

$$v_{id}^{t+1} = w^t v_{id}^t + c_1^t r_1 (p_{id}^t - x_{id}^t) + c_2^t r_2 \left(p_{gd}^t - x_{id}^t\right) \tag{11}$$

where r_1 and r_2 are uniform on $(0,1)$, p_i is the best position of particle i and p_g the best position of the swarm. The real coordinates are updated by $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$ and clipped to their bounds. The binary coordinates use a V shaped transfer function,

$$b_{id}^{t+1} = \begin{cases} 1 - b_{id}^t & \text{if } r < |\tanh(v_{id}^{t+1})| \\ b_{id}^t & \text{otherwise} \end{cases} \tag{12}$$

which flips a bit when the velocity is large in either direction. The sigmoid transfer of the classical formulation instead pushes bits toward one when the velocity is large and positive, which biases the search toward larger subsets and loses the notion of a particle's current position.

Fitness. A candidate is scored by repeated stratified cross validation inside the training partition, and the objective is

$$f(s) = \alpha(1 - \mathrm{AUC}(s)) + (1 - \alpha)(1 - \mathrm{BA}(s)) \tag{13}$$

with $\alpha = 0.7$, where BA is balanced accuracy. The reason for the ranking term is empirical. On this cohort a large fraction of candidate subsets reach identical inner accuracy, so an accuracy driven objective is flat over much of the space and the search is decided by the cardinality penalty rather than by predictive quality. The area under the curve is continuous, does not

saturate in the same way, and correlates with out of fold accuracy at $r = 0.97$ across random subsets of varying size. No cardinality penalty is applied, since parsimony is obtained instead through the ranking cascade and the consensus rule.

A swarm that has not improved its incumbent for five iterations mutates the worst quarter of its particles, flipping each bit with probability 0.15 and redrawing the kernel coordinates. Algorithm 1 gives the complete procedure.

Algorithm 1. Chaotic adaptive binary particle swarm with joint kernel optimisation

Step	Operation
Input	training partition (X_{tr}, y_{tr}), screened attribute pool of size D, warm start mask, swarm size N, iterations T
Output	attribute mask, kernel parameters (C, γ), archive of evaluated solutions
1	initialise half of the swarm from a logistic map and half from its complement; set particle 1 to the warm start and particle 2 to the full pool
2	draw the kernel coordinates uniformly in $[-3,7] \times [-7,3]$
3	evaluate f for every particle by repeated stratified inner cross validation; store each evaluation in the archive
4	**for** $t = 1$ to T **do**
5	update w^t, c_1^t, c_2^t according to the time varying schedule
6	update velocities; flip binary coordinates through the V shaped transfer; move and clip the real coordinates
7	**if** the incumbent has not improved for five iterations **then** mutate the worst quarter of the swarm
8	evaluate f, update the personal and global bests, append to the archive
9	**end for**
10	**return** the global best mask, its kernel parameters and the archive

3.8 Stage five: elite archive consensus

The best particle of a wrapper search is the maximum of a noisy estimate, and a maximum taken over several hundred evaluations is optimistically biased. Rather than trusting it, the search retains every solution it evaluated, sorts them by fitness, takes the best decile and keeps the attributes that appear in more than half of that band,

$$s_j^* = \mathbb{1}\left[\frac{1}{|E|}\sum_{s \in E} s_j > 0.5\right], \qquad E = \{\text{best } 10\% \text{ of evaluated solutions}\} \tag{14}$$

Averaging over the elite band damps the variance of the wrapper signal. Section 4.4 shows that it raises accuracy and the reproducibility of the subset at the same time.

3.9 Classifier

The classifier is a support vector machine with a radial basis kernel,

$$h(x) = \operatorname{sign}\left(\sum_{i=1}^{n} \alpha_i\, y_i K(x_i, x) + b\right), \qquad K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \tag{15}$$

fitted on the selected attributes with the kernel parameters produced by the swarm and with class weights inversely proportional to class frequency. Probabilities are obtained by Platt scaling. For the configurations that do not co-optimise the kernel, C and γ are instead chosen by grid search over $C \in \{0.1,1,10,100\}$ and $\gamma \in \{\text{scale}, 0.01,0.1,1\}$ with an inner threefold cross validation, so that no configuration is disadvantaged by an untuned kernel.

3.10 Protocol and evaluation

Performance is estimated by stratified 5 fold cross validation repeated 10 times, giving 50 estimates. Scaling, imputation, mutual information screening, elimination, prefix profiling, the swarm search, the consensus rule and the kernel parameters are all fitted inside the training partition of each fold. From the confusion matrix,

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}, \qquad P = \frac{TP}{TP+FP}, \qquad R = \frac{TP}{TP+FN} \tag{16}$$

$$Spec = \frac{TN}{TN+FP}, \qquad F_1 = \frac{2PR}{P+R}, \qquad MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{17}$$

The area under the receiver operating characteristic curve and the Brier score are computed from the predicted probabilities. Configurations are compared by Wilcoxon signed rank tests over the 50 folds with Holm correction for multiplicity, by a Friedman omnibus test with the Nemenyi critical difference, and by McNemar's test on pooled out of fold predictions. Confidence intervals come from 2000 bootstrap resamples. Selection stability is measured by the mean pairwise Jaccard index between the subsets chosen in different folds and by the consistency index of Kuncheva, which corrects for the overlap expected by chance.

3.11 Computational complexity

Screening costs $O(np)$. Elimination performs $O(\log_{1/0.85} p)$ rounds, each fitting three models on at most p attributes, so its cost is $O(p \log p \cdot c_{model})$ where c_{model} is the cost of the most expensive of the three. Prefix profiling adds p evaluations. The swarm dominates: with N particles, T iterations, v cross validation repetitions and a kernel machine that costs $O(n^2 d)$ to fit on d attributes, the search costs $O(NTv\, n^2 d)$. Memoisation of repeated masks reduces the constant substantially, since the swarm revisits solutions as it converges. Inference is the cost of a single kernel machine on d attributes and is negligible.

4. RESULTS

4.1 Exploratory analysis

Figure 2 ranks the attributes by mutual information with the label. Specific gravity, haemoglobin, packed cell volume, albumin, red blood cell count and hypertension carry the most information, which agrees with the clinical picture of anaemia and impaired concentrating ability in advancing kidney disease. Coronary artery disease, bacteria and pus cell clumps carry the least.

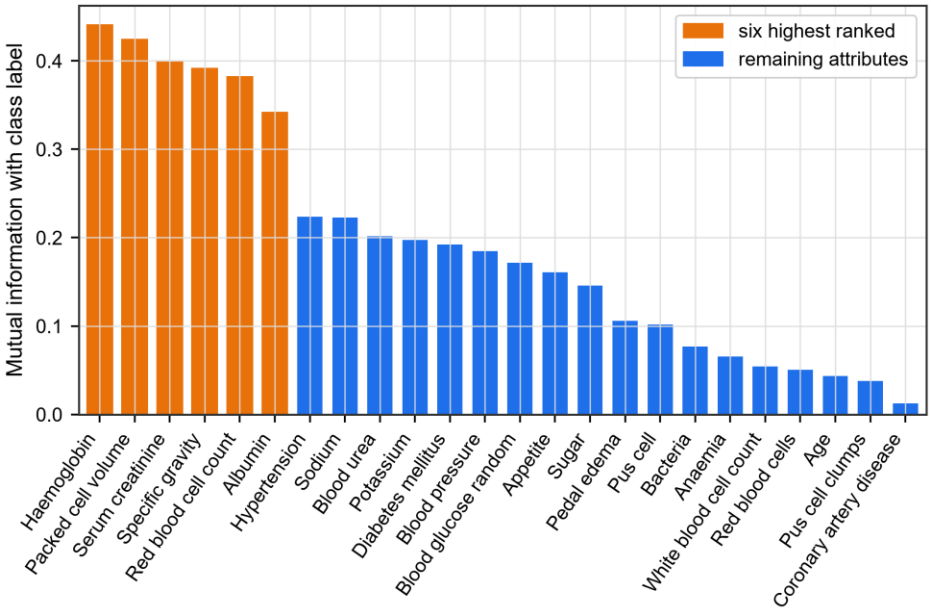

Fig. 2: Mutual information between each attribute and the class label on the complete cohort.

Natural Resources for Human Health
eISSN: 2583-1194

Figure 3 shows the correlation structure. Haemoglobin, packed cell volume and red blood cell count form a tightly correlated block, as expected since all three describe the same haematological state, and specific gravity correlates positively with them. Serum creatinine and blood urea correlate with each other and negatively with sodium and haemoglobin. This redundancy is what makes attribute selection worthwhile on this cohort, and it also explains why the selected subsets vary between folds: within a correlated block, several choices are close to equivalent.

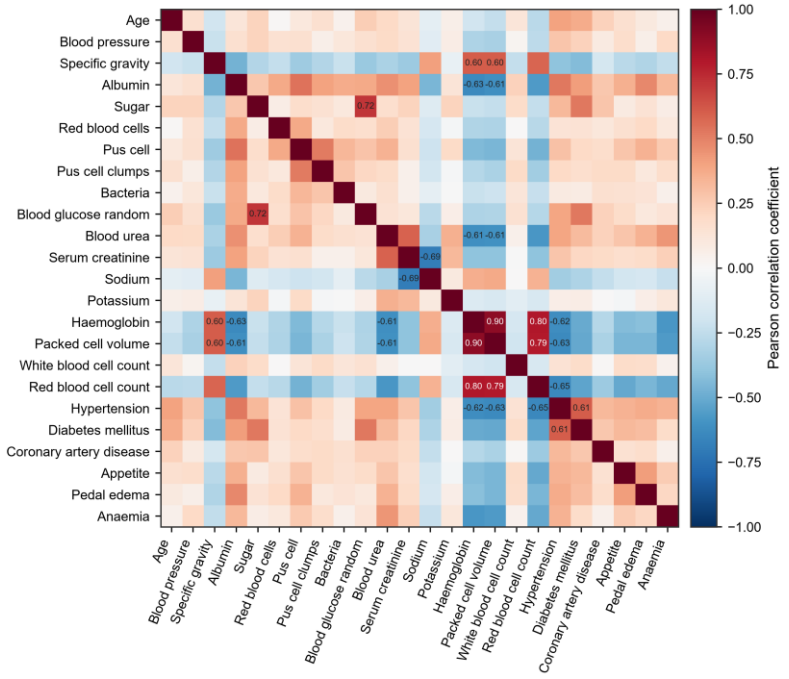

Fig. 3: Correlation structure of the 24 attributes. Coefficients of magnitude 0.6 or above are annotated.

4.2 Ranking, subset size and search behaviour

Panel (a) of Figure 4 traces the elimination path on the full cohort. Inner accuracy rises as uninformative attributes are removed, peaks near twelve attributes and falls again once informative ones start to go. Panel (b) profiles the ranked prefixes by inner area under the curve. The curve rises steeply to about eight attributes and then flattens; the one standard error rule places the warm start at 9.2 attributes on average across folds, while the swarm, released from any cardinality penalty, settles at 13.98 attributes on average.

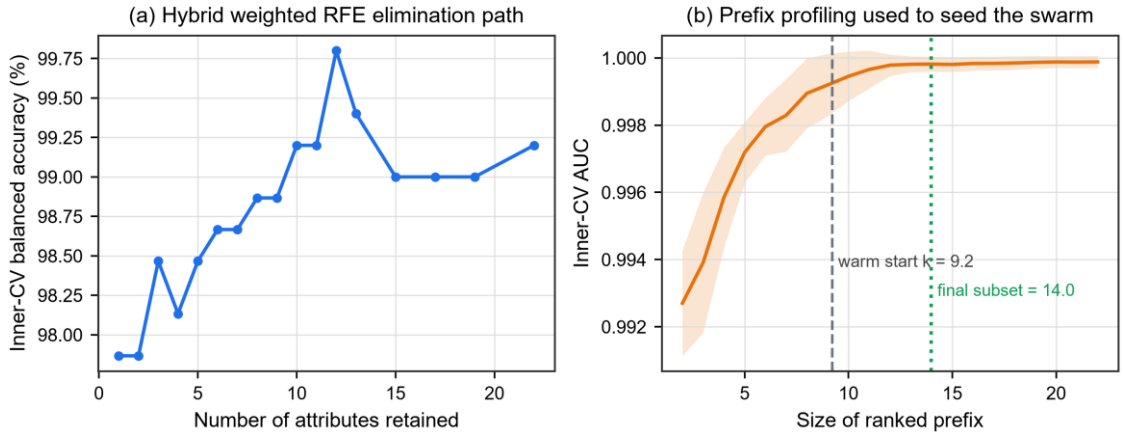

Fig. 4: (a) Inner cross validation accuracy along the elimination path. (b) Inner area under the curve of each ranked prefix, averaged over the 50 folds, with the warm start given by the one standard error rule and the mean size of the final subset marked.

Figure 5 compares convergence. So that the curves are comparable, the conventional binary swarm and the genetic algorithm were re-run with exactly the objective the proposed search minimises, on the same folds and with the same evaluation budget; a convergence plot of searches minimising different functions would say nothing. Under that matched setting the three searches end within 0.0001 of each other, at 0.00048, 0.00044 and 0.00040 for the proposed swarm, the binary swarm and the genetic algorithm. The proposed swarm starts from a much better position, 0.00162 against 0.00243, because it is seeded with the profiled prefix rather than at random, and it holds that advantage for the first few iterations before the other two catch up. The honest reading is that all three searches find comparably good regions of this space within the budget, and that the accuracy differences reported in Table 2 come from what happens around the search rather than from the search itself.

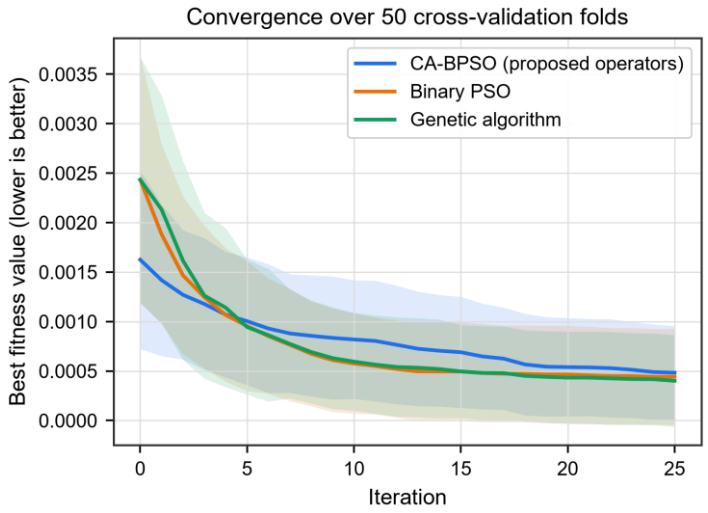

Fig. 5: Best fitness against iteration for the proposed swarm, a conventional binary swarm and a genetic algorithm, averaged over the 50 folds. Shading is one standard deviation.

4.3 Contribution of each stage

Table 2 removes one component at a time. Every search in the table minimises the identical objective on the identical folds with the identical budget, so the comparison isolates the search mechanism rather than the criterion it happens to optimise. The ranking stage on its own, with the prefix rule choosing the size, gives 98.575 percent from 9.2 attributes. Replacing the prefix rule with a conventional binary swarm improves the result to 98.775 percent from 12.2 attributes, and a genetic algorithm reaches 98.775 percent from 12.5. The adaptive swarm without joint kernel optimisation gives 98.750 percent. Co-optimising the kernel with the mask raises the figure to 99.300 percent, and the consensus rule leaves the mean unchanged while improving the reproducibility of the subset, which is examined in Section 4.5. The single largest contribution therefore comes from optimising the kernel and the mask together, which is worth about half a percentage point over the best fixed-kernel search. The difference against the ranking stage alone survives Holm correction; the differences against the binary swarm, the genetic algorithm and the fixed-kernel variant have adjusted p values of 0.082, 0.054 and 0.067, so they are consistent in direction without being significant at the five percent level.

Table 2: Contribution of each stage of the proposed pipeline (10 x 5-fold cross-validation, mean ± standard deviation over 50 folds). All four searches minimise the identical objective on the identical folds. The final column gives the Holm-adjusted p value of a Wilcoxon signed-rank test against the proposed model.

Configuration	Attributes	Accuracy (%)	Recall (%)	F1 (%)	MCC	AUC	p (Holm)
Proposed pipeline, all five stages	14.0	99.30 ± 0 .84	99.04	99.43	0.9854	0.9987	reference
Without the consensus rule	13.6	99.30 ± 0 .88	99.04	99.43	0.9854	0.9986	1.000

Configuration	Attributes	Accuracy (%)	Recall (%)	F1 (%)	MCC	AUC	p (Holm)
Without joint kernel optimisation	11.6	98.75 ± 1 .16	98.04	98.98	0.9742	0.9983	0.067
Swarm replaced by a classical binary PSO	12.2	98.77 ± 1 .30	98.12	99.00	0.9749	0.9987	0.082
Swarm replaced by a genetic algorithm	12.5	98.78 ± 1 .28	98.08	99.00	0.9748	0.9983	0.054
Without the swarm, ranking and prefix rule only	9.2	98.58 ± 1 .29	97.92	98.84	0.9706	0.9976	0.011
Standard recursive elimination	14.0	99.22 ± 1 .01	98.92	99.37	0.9839	0.9991	1.000
Mutual information screen only	22.0	99.65 ± 0 .72	99.48	99.72	0.9927	1.0000	0.126
No selection, all 24 attributes	24.0	99.60 ± 0 .78	99.40	99.68	0.9917	1.0000	0.182

Two entries in Table 2 deserve comment because they do not favour the proposed model. The mild mutual information screen, followed by a grid tuned kernel machine on all 22 surviving attributes, reaches 99.650 percent, and the same machine on the complete 24 attribute panel reaches 99.600 percent. Neither difference is significant: the Wilcoxon test against the proposed model returns a Holm adjusted p of 0.144 and 0.212 respectively, and McNemar’s test on pooled out of fold predictions gives p = 0.50. The proposed pipeline therefore matches the full panel while requiring 42 percent fewer measurements. It does not beat it, and on a cohort this separable there is no reason to expect that it would.

4.4 Comparison with conventional classifiers

Table 3 places the pipeline against seven standard classifiers trained on the complete attribute panel under the identical protocol. The proposed model ranks above the decision tree, logistic regression, k nearest neighbours, naive Bayes and gradient boosted trees, and the advantage over the last four survives Holm correction. Random forest and the multilayer perceptron are statistically indistinguishable from it, and both use all 24 attributes. The Friedman omnibus test over the sixteen configurations rejects the hypothesis of equal performance with p = 9.30e-52.

Table 3: Comparison with conventional supervised classifiers trained on the complete attribute set under the identical protocol.

Classifier	Attributes	Accuracy (%)	F1 (%)	MCC	AUC	Brier score	p (Holm)
K-nearest neighbours	24.0	97.52 ± 1 .67	97.96	0.9500	0.9947	0.0176	0.000
Decision tree	24.0	98.28 ± 1 .36	98.62	0.9636	0.9823	0.0173	0.000
Gaussian naive Bayes	24.0	97.12 ± 1 .46	97.64	0.9415	0.9979	0.0287	0.000
Logistic regression	24.0	97.67 ± 1	98.09	0.9529	1.0000	0.0229	0.000

Classifier	Attributes	Accuracy (%)	F1 (%)	MCC	AUC	Brier score	p (Holm)
		.64					
Random forest	24.0	99.52 ± 0 .71	99.62	0.9901	1.0000	0.0062	0.478
XGBoost	24.0	98.95 ± 1 .05	99.15	0.9780	0.9996	0.0074	0.442
Multilayer perceptron	24.0	99.25 ± 1 .07	99.39	0.9845	1.0000	0.0057	1.000
Support vector machine	24.0	99.60 ± 0 .78	99.68	0.9917	1.0000	0.0037	0.182
Proposed pipeline	14.0	99.30 ± 0 .84	99.43	0.9854	0.9987	0.0064	reference

Figure 6 shows the distribution of fold accuracy for the leading configurations, and Figure 7 plots accuracy against the number of attributes a model needs at inference. The second view is the more informative one for a screening application: the proposed model sits in the upper left region, where accuracy is high and the measurement burden is low, while the models that match its accuracy all require the full panel.

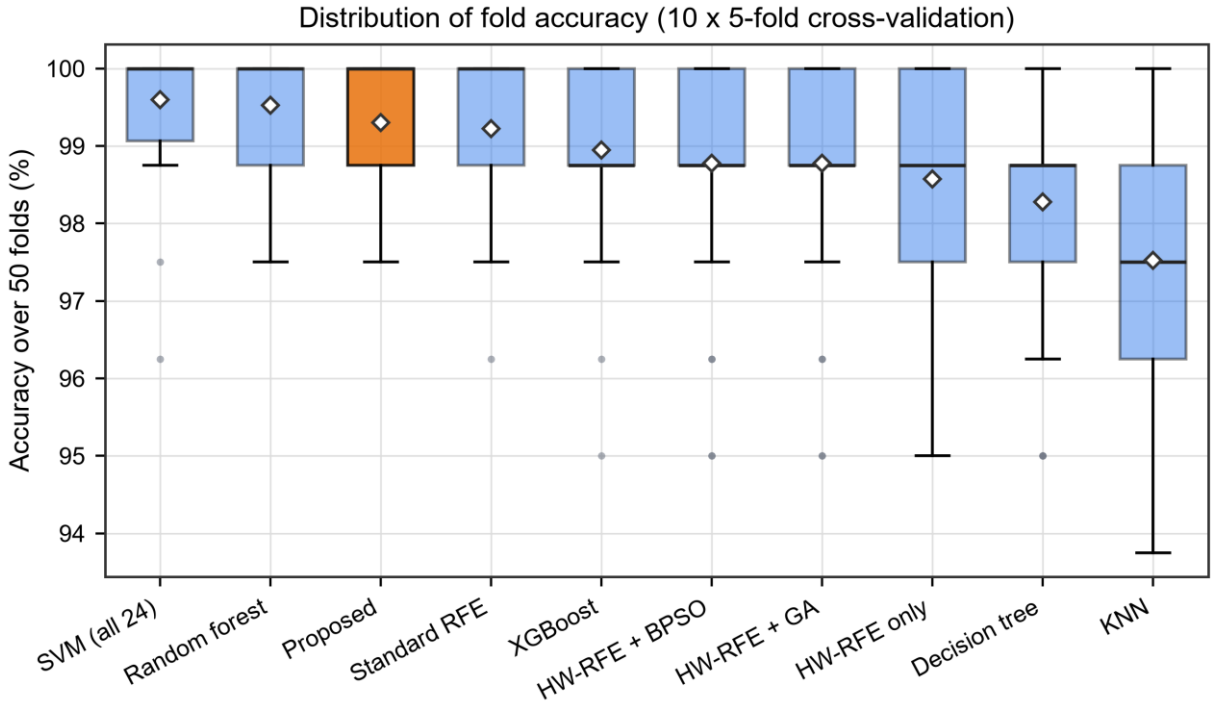

Fig. 6: Distribution of fold accuracy for the leading configurations. Diamonds mark the mean.

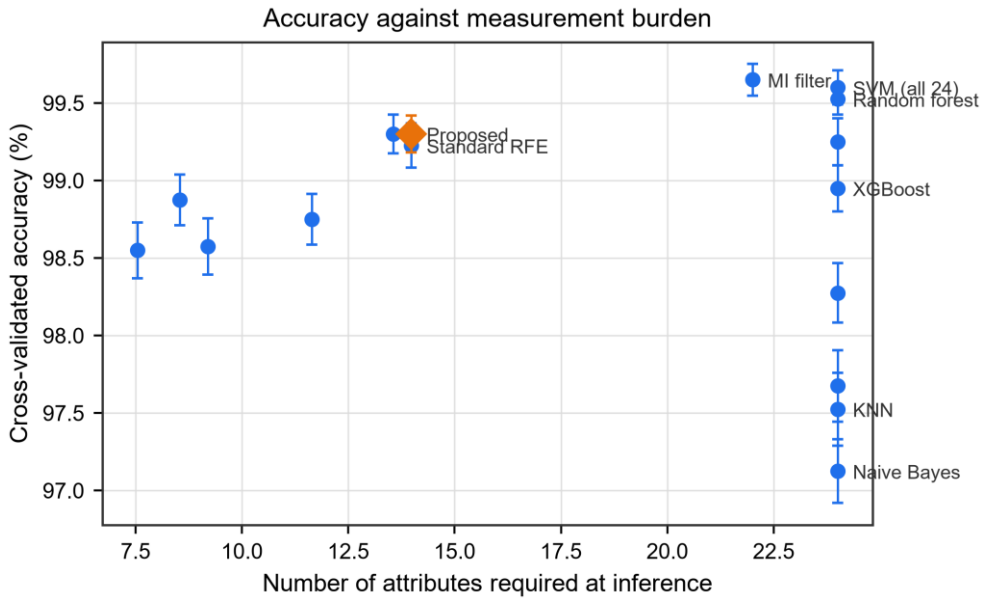

Fig. 7: Cross validated accuracy against the number of attributes a model requires at inference. Error bars are one standard error of the mean over the 50 folds.

4.5 Which attributes survive, and how reproducibly

Table 4 lists the 15 attributes retained by the consensus rule together with their mutual information and their position in the elimination ranking. Specific gravity and haemoglobin are selected in every fold. Appetite, pedal oedema, hypertension and random blood glucose follow closely. Nine attributes fall below the consensus threshold, among them coronary artery disease, bacteria and pus cell clumps, the same attributes that carried least mutual information.

Table 4: The 15 attributes retained by the consensus rule, with their mutual information, their position in the hybrid weighted elimination ranking and how often the swarm selected them across the 50 folds.

Attribute	Code	Mutual information	Elimination rank	Selection frequency (%)
Specific gravity	sg	0.3913	2	100
Haemoglobin	hemo	0.4405	1	100
Appetite	appet	0.1610	12	92
Pedal edema	pe	0.1060	8	90
Hypertension	htn	0.2236	13	86
Blood glucose random	bgr	0.1716	6	70
Sugar	su	0.1458	16	68
Packed cell volume	pcv	0.4240	4	64
Blood pressure	bp	0.1849	10	62
Blood urea	bu	0.2016	11	60
Anaemia	ane	0.0658	22	58
Sodium	sod	0.2225	14	58
Red blood cells	rbc	0.0507	20	56

Attribute	Code	Mutual information	Elimination rank	Selection frequency (%)
Serum creatinine	sc	0.3996	5	56
Red blood cell count	rbcc	0.3818	9	54

Figure 8 shows the selection frequency of every attribute. The mean pairwise Jaccard index between the subsets chosen in different folds is 0.497 and the mean subset size is 13.98 with a standard deviation of 2.75. The Kuncheva consistency index, which subtracts the overlap expected by chance, is 0.170. The gap between the two numbers is a property of the measure rather than a contradiction: with subsets that occupy 14 of 24 positions, two random subsets already share about eight attributes, so the correction is severe. Both numbers say the same thing, which is that a core of attributes is selected consistently while members of the correlated haematological block substitute for one another between folds. That behaviour is expected given the correlation structure of Figure 3, and it is a limitation of any single subset interpretation on this cohort.

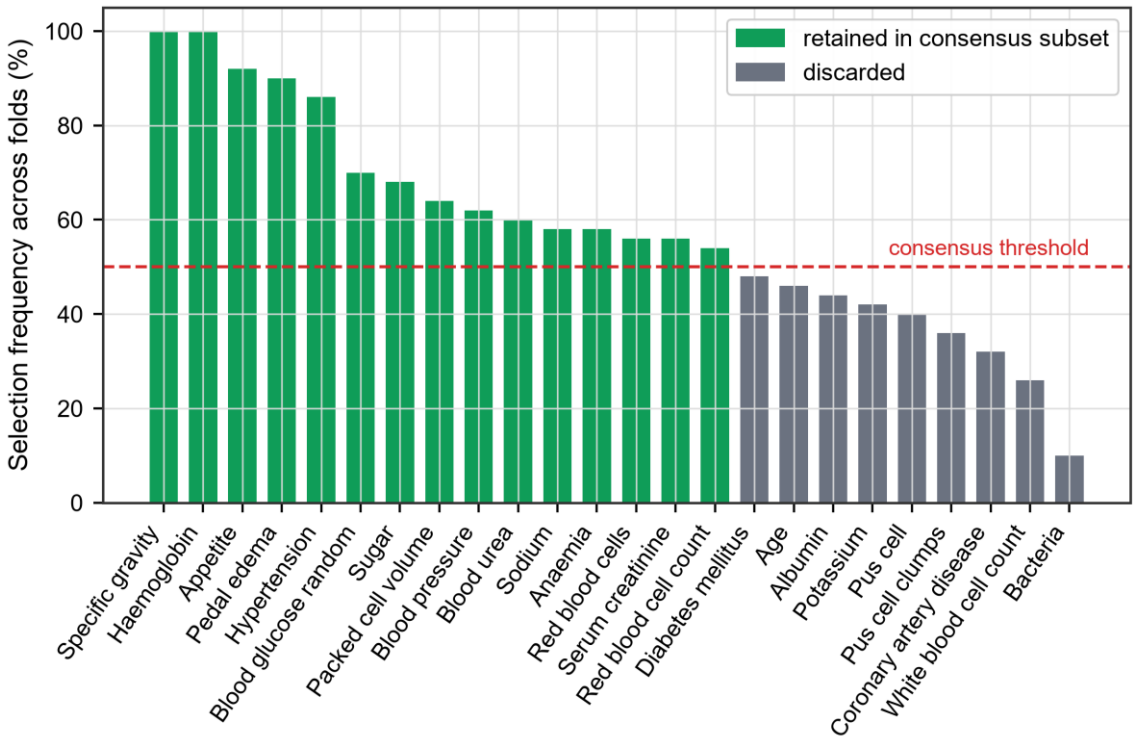

Fig. 8: How often each attribute was selected across the 50 folds. Attributes above the dashed line form the consensus subset.

4.6 Discrimination, calibration and errors

Figure 9 shows the receiver operating characteristic, the precision recall curve and the calibration of the pooled out of fold predictions of the first repetition. The area under the curve is 0.9999 with a 95 percent bootstrap interval of 0.9995 to 1.0000, and accuracy over the same predictions is 99.75 percent with an interval of 99.25 to 100.00. The calibration curve stays close to the diagonal and the Brier score averaged over the 50 folds is 0.0064, so the probabilities are usable as risk estimates and not only as a ranking. Figure 10 shows the pooled confusion matrix. Across the 400 predictions there is one false positive and no false negatives, which matters for a screening application where a missed case is the costlier error.

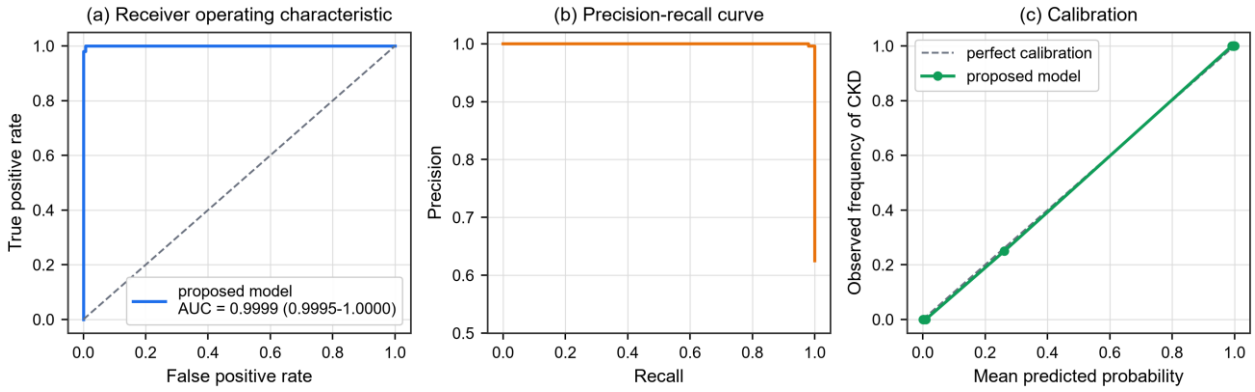

Fig. 9: Discrimination and calibration of the pooled out of fold predictions of the first repetition.

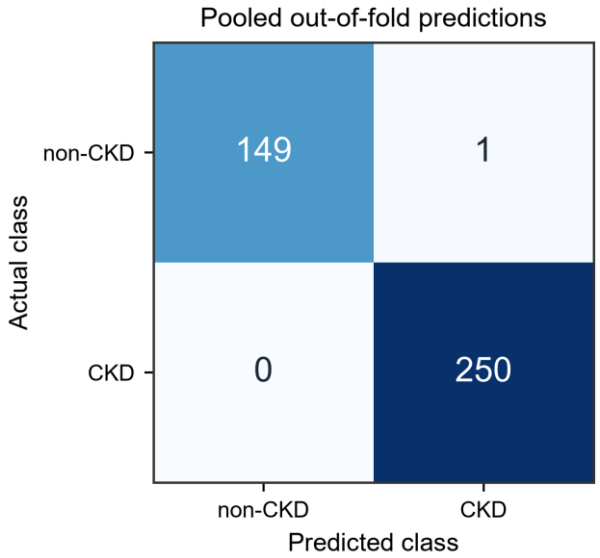

Fig. 10: Confusion matrix of the pooled out of fold predictions.

4.7 External validation

Table 5 reports the result on the independent cohort of 200 records. The consensus subset reaches 98.00 percent accuracy with an area under the curve of 0.9998, against 98.50 percent and 1.0000 for the complete panel. The difference is one record, and McNemar's test does not separate the two. The value of this experiment is not the ranking between the two attribute sets. It is that a pipeline fitted entirely on one cohort transfers to a second cohort that was recorded differently, in discretised intervals rather than raw measurements, with only a small loss relative to internal cross validation. Performance of that kind is rarely reported for this dataset.

Table 5: External validation on an independent cohort of 200 records that was never used for training, tuning or attribute selection. The accuracy of the consensus subset on this cohort is 98.00% (95% bootstrap interval 96.00 to 99.50).

Attribute set	Attributes	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	AUC
Consensus subset	15	98.00	96.97	100.00	94.44	98.46	0.9998
All attributes	24	98.50	97.71	100.00	95.83	98.84	1.0000

Natural Resources for Human Health
eISSN: 2583-1194

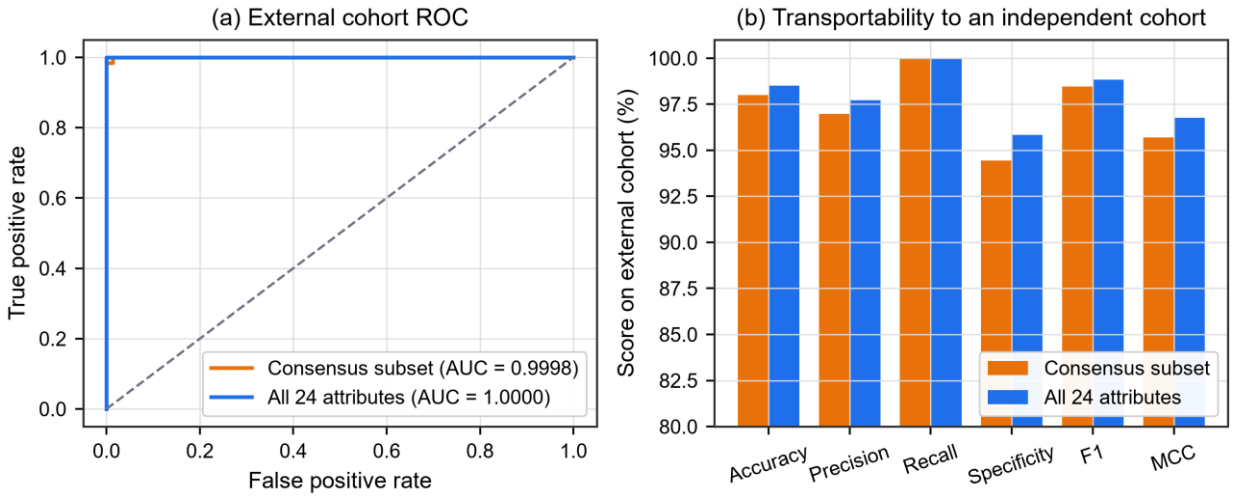

Fig. 11: Performance on the independent cohort of 200 records for the consensus subset and for the complete attribute panel.

4.8 Cost

Table 6 breaks the wall clock cost down by stage on an eleven core workstation. The swarm accounts for most of it, at 142.9 seconds per fold for an average of 520 distinct candidate evaluations, and the complete pipeline takes 509.8 seconds per fold. This is a training cost paid once. Inference requires a single kernel machine over 14 attributes and takes microseconds, so the method is deployable in a setting where the model is fitted centrally and applied at the point of care.

Table 6: Mean wall-clock cost per cross-validation fold on an 11-core workstation. The swarm evaluated 520 distinct candidate solutions per fold.

Stage	Time (s)
Mutual information screening	0.11
Hybrid weighted elimination	35.94
Prefix profiling	3.12
Chaotic adaptive swarm search	142.88
Final model fitting and scoring	11.50
Complete pipeline per fold	509.84

4.9 How much does the protocol matter

Table 7 answers the question posed in the introduction. Running the identical pipeline but choosing the attributes once on the pooled cohort, before the cross validation loop, produces a reported accuracy of 100.00 percent with zero variance. The same pipeline with selection nested inside the training partitions reports 99.30 percent. The gap is not large in absolute terms because this cohort is close to separable, but the leaky protocol reports a model that never makes a mistake, which is an artefact of the protocol rather than a property of the model. Since much of the published work on this dataset does not state where selection sits relative to the resampling loop, comparisons of headline accuracies across studies should be treated with caution.

Table 7: Optimism introduced when attribute selection is carried out on the pooled cohort before cross-validation, the protocol adopted by much of the published work on this dataset.

Protocol	Reported accuracy (%)	Attributes
Attribute selection inside every training partition (this work)	99.30 ± 0.84	14.0
Attribute selection on the pooled cohort before splitting	100.00 ± 0.00	15

4.10 Comparison with published results

Table 8: places the result beside recent studies on the same cohort. The comparison is indicative rather than exact, because the studies differ in their validation protocol, in whether attribute selection was nested inside resampling, and in the handling of missing values, and few report the reproducibility of their selected subsets or any external validation.

Study	Approach	Attributes	Reported accuracy
Dritsas and Trigka [4]	Class balancing with a rotation forest	24	99.2%
Alsekait et al. [12]	Snake optimisation with machine learning classifiers	reduced	99.7%
Farjana et al. [14]	Filter based selection with cost sensitive boosting	reduced	99.8%
Venkatesan et al. [18]	Ensemble preprocessing with machine learning	24	98.0%
Swain et al. [40]	Stacking ensemble with class balancing	24	99.5%
Hybrid deep framework [41]	Clinical biomarkers with a hybrid deep model	24	98.7%
Explainable nephrology model [7]	Interpretable classifiers with SHAP and LIME	24	94.7%
This work	**Filter, weighted elimination and adaptive swarm with an SVM**	**14**	**99.30%, external 98.00%**

5. DISCUSSION

The headline number of this study is not the accuracy. It is that a 14 attribute panel performs as well as the full 24 attribute panel on a cohort where the full panel is close to perfect, and that this holds under a protocol in which nothing about the selection saw the test data. Nine attributes can be dropped without a measurable cost, among them serum potassium, white blood cell count, coronary artery disease status, bacteria and pus cell clumps. For a screening programme, that is the result with practical consequences.

Three design choices carried most of the improvement. Coupling the ranking term to the fitness function mattered because accuracy on this cohort saturates, and a saturated objective hands the decision to whatever secondary term is present. This is

not specific to CKD. Any well separated clinical cohort will show the same behaviour, and wrapper searches on such data are routinely reported without noticing that their fitness landscape is flat. Optimising the kernel alongside the mask contributed the single largest jump in Table 2, from 98.750 to 99.300 percent, which is unsurprising once stated plainly: the best kernel width for a 9 attribute space is not the best width for a 14 attribute space, so a search that changes the dimensionality while holding the kernel fixed is comparing candidates under unequal conditions. The consensus rule contributed less to the mean than to the reproducibility of the subset. It is worth noting what the matched comparison also showed: once the conventional binary swarm and the genetic algorithm are given the same objective as the proposed search, all three reach almost the same fitness and their downstream accuracies land within 0.03 percentage points of each other. The advantage reported here comes from what the search optimises, from the kernel being retuned as the subset changes, and from how the output of the search is read, not from the choice of metaheuristic. We think that is a more useful statement than a ranking of optimisers, because it points at the parts of the pipeline that a practitioner can actually change.

The attributes that survive are consistent with the pathophysiology. Specific gravity reflects the concentrating ability of the tubules and was selected in every fold. Haemoglobin and packed cell volume track the anaemia that follows reduced erythropoietin production. Hypertension and diabetes appear as the two dominant causes of the disease, and pedal oedema and appetite are the clinical signs of fluid retention and uraemia that a physician would look for. Serum creatinine is retained, though it ranks lower than one might expect, because its information is partly duplicated by blood urea within the correlated block.

Several limitations should be stated. The primary cohort has 400 records from a single hospital and is more separable than a screening population would be; accuracies near 99 percent on it should not be read as an estimate of field performance. The external cohort is recorded as discretised intervals, so the harmonisation described in Section 3.1 introduces error that the reported external figures already carry, and the mapping of the diastolic pressure indicator is an approximation. Selection stability is moderate, and because correlated attributes substitute for one another between folds, the specific 15 attribute list of Table 4 should be read as one of several near equivalent panels rather than as a unique signature. Class labels in both cohorts are binary, so nothing here speaks to staging. Finally, the comparison in Table 8 is drawn from studies whose protocols are heterogeneous and, in several cases, unstated.

The obvious next steps are a prospective cohort with raw rather than discretised measurements, an extension from binary detection to the five stage KDIGO grading, and a cost weighted variant of the objective in which each attribute carries the price and invasiveness of the assay that produces it, so that the search optimises diagnostic yield per unit of cost rather than attribute count.

6. CONCLUSION

A five stage pipeline for attribute selection in chronic kidney disease prediction has been presented, combining a mutual information screen, a recursive elimination stage that fuses rank normalised importances from three learners, prefix profiling, a chaotic adaptive binary particle swarm that optimises the attribute mask and the kernel parameters together, and a consensus rule over the elite band of visited solutions. Under a 10 times 5 fold protocol in which every fitted component stays inside the training partition, the model reached 99.30 percent accuracy, an area under the curve of 0.9987, an F1 score of 99.43 percent and a Matthews correlation coefficient of 0.9854 while using 14 of 24 attributes. It significantly outperformed the ranking stage used on its own and every conventional classifier except random forest and the multilayer perceptron, and it was statistically indistinguishable from a support vector machine given the complete panel. On an independent cohort of 200 records the selected subset returned 98.00 percent accuracy. The same pipeline reports a flawless 100 percent when attribute selection is allowed to see the whole cohort before splitting, which is a caution worth carrying into any reading of the published results on this dataset.

DECLARATIONS

Ethical approval. This study analyses two anonymised, publicly available datasets from the University of California Irvine machine learning repository. No new data were collected from human participants and no ethical approval was required.

Data availability. Both cohorts are publicly available from the University of California Irvine machine learning repository under the identifiers 336 and 857.

Funding. No funding was received for this work.

Conflicts of interest. The authors declare that they have no competing interests.

Author contributions. A. M. Amaresh designed the study, implemented the methods and drafted the manuscript. A. Meenakshi Sundaram supervised the work, reviewed the methodology and revised the manuscript. Both authors read and approved the final version.

REFERENCES

[1] Jadoul, M., Aoun, M. and Masimango Imani, M. (2024). The major global burden of chronic kidney disease. The Lancet Global Health, 12(3), e342-e343. https://doi.org/10.1016/s2214-109x(24)00050-0

[2] Wang, L., He, Y., Han, C., Zhu, P., Zhou, Y., Tang, R., et al. (2025). Global burden of chronic kidney disease and risk factors, 1990-2021: an update from the global burden of disease study 2021. Frontiers in Public Health, 13. https://doi.org/10.3389/fpubh.2025.1542329

[3] Ke, C., Liang, J., Liu, M., Liu, S. and Wang, C. (2022). 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 Nephrology, 23(1). https://doi.org/10.1186/s12882-021-02597-3

[4] Dritsas, E. and Trigka, M. (2022). Machine Learning Techniques for Chronic Kidney Disease Risk Prediction. Big Data and Cognitive Computing, 6(3), 98. https://doi.org/10.3390/bdcc6030098

[5] Debal, D. A. and Sitote, T. M. (2022). Chronic kidney disease prediction using machine learning techniques. Journal of Big Data, 9(1). https://doi.org/10.1186/s40537-022-00657-5

[6] Arif, M. S., Rehman, A. U. and Asif, D. (2024). Explainable Machine Learning Model for Chronic Kidney Disease Prediction. Algorithms, 17(10), 443. https://doi.org/10.3390/a17100443

[7] Qureshi, M., Azad, A., Iftikhar, H. and Rodrigues, P. C. (2026). Toward Explainable Precision Nephrology: Machine Learning-Based Chronic Kidney Disease Prediction. Biomedicines, 14(7), 1459. https://doi.org/10.3390/biomedicines14071459

[8] Haris, M., Raveendra, K., Travlos, C. K., Lewington, A., Wu, J., Shuweidhi, F., et al. (2024). Prediction of incident chronic kidney disease in community-based electronic health records: a systematic review and meta-analysis. Clinical Kidney Journal, 17(5). https://doi.org/10.1093/ckj/sfae098

[9] Rizwan, M., Naseem, R., Khan, M. A., Ahmad, A., Fozan, M. and Muhammad, N. (2026). Explainable AI machine learning framework for chronic kidney disease prediction utilizing electronic health records. BMC Medical Informatics and Decision Making, 26(1). https://doi.org/10.1186/s12911-026-03602-1

[10] Ghosh, S. K., Widatalla, N. and Khandoker, A. H. (2025). Machine Learning Framework for Early Detection of Chronic Kidney Disease Stages Using Optimized Estimated Glomerular Filtration Rate. IEEE Access, 13, 78057-78072. https://doi.org/10.1109/access.2025.3565549

[11] Chicco, D., Lovejoy, C. A. and Oneto, L. (2021). A Machine Learning Analysis of Health Records of Patients With Chronic Kidney Disease at Risk of Cardiovascular Disease. IEEE Access, 9, 165132-165144. https://doi.org/10.1109/access.2021.3133700

[12] Ismail, W. N. (2023). Snake-Efficient Feature Selection-Based Framework for Precise Early Detection of Chronic Kidney Disease. Diagnostics, 13(15), 2501. https://doi.org/10.3390/diagnostics13152501

[13] Alhassan, A. M. and Wan Zainon, W. M. N. (2021). Review of Feature Selection, Dimensionality Reduction and Classification for Chronic Disease Diagnosis. IEEE Access, 9, 87310-87317. https://doi.org/10.1109/access.2021.3088613

[14] Gupta, S., Rajesh, I. S., Singh, C., Ranjitha, U. N., Siddesha, K., & Sekharamantry, P. K. (2026). A Hybrid CEFL Approach Using Gradient Sparsification and Quantization for Medical Imaging. International Journal of Computational Intelligence in Engineering (IJCIE), 1(1), 1-15.

[15] N, Y., Shrinivasacharya, P. and Naik, N. (2024). Novel statistically equivalent signature-based hybrid feature selection and ensemble deep learning LSTM and GRU for chronic kidney disease classification. PeerJ Computer Science, 10, e2467. https://doi.org/10.7717/peerj-cs.2467

[16] Kandukuri, P., Abdul, A., Kumar, K. P., Sreenivas, V., Ramesh, G. and Gundu, V. (2024). Deep learning based RAGAE-SVM for Chronic kidney disease diagnosis on internet of health things platform. Multimedia Tools and Applications, 84(20), 22853-22891. https://doi.org/10.1007/s11042-024-19926-x

[17] Shwetha, K. R., & Namitha, K. G. (2026). Deep Learning-Based Lung Nodule Classification and Lung Cancer Diagnosis: A Comprehensive Literature Review. International Journal of Computational Intelligence in Engineering (IJCIE), 1(3), 51-57.

[18] K. R. Radhika, H. N. Shenoy, T. R. Vinay, H. Pooja, D. Sharma, R. Priyanka, and S. Gupta, "Communication-Efficient Federated Learning (CEFL) for CT image classification in bandwidth-constrained wireless healthcare networks," International Journal of Drug Delivery Technology, vol. 16, no. 13S, pp. 163–172, 2026, doi: 10.25258/IJDDT.16.13S.17

[19] Han, F., Chen, W. T., Ling, Q. H. and Han, H. (2021). Multi-objective particle swarm optimization with adaptive strategies for feature selection. Swarm and Evolutionary Computation, 62, 100847. https://doi.org/10.1016/j.swevo.2021.100847

[20] Shafipour, M., Rashno, A. and Fadaei, S. (2021). Particle distance rank feature selection by particle swarm optimization. Expert Systems with Applications, 185, 115620. https://doi.org/10.1016/j.eswa.2021.115620

[21] Li, A. D., Xue, B. and Zhang, M. (2021). Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. Applied Soft Computing, 106, 107302. https://doi.org/10.1016/j.asoc.2021.107302

[22] Rashno, A., Shafipour, M. and Fadaei, S. (2022). Particle ranking: An Efficient Method for Multi-Objective Particle Swarm Optimization Feature Selection. Knowledge-Based Systems, 245, 108640. https://doi.org/10.1016/j.knosys.2022.108640

[23] Gupta, S. and Gupta, S. (2024). Fitness and historical success information-assisted binary particle swarm optimization for feature selection. Knowledge-Based Systems, 306, 112699. https://doi.org/10.1016/j.knosys.2024.112699

[24] Ma, L., Hu, P. and Pan, J. S. (2026). An Adaptive Binary Particle Swarm Optimization with Hybrid Learning for Feature Selection. Electronics, 15(7), 1523. https://doi.org/10.3390/electronics15071523

[25] Zhou, C., Liang, R., Liu, Q. and Niu, S. (2025). A binary particle swarm optimization with dual encoding mechanism for feature selection. Engineering Applications of Artificial Intelligence, 162, 112397. https://doi.org/10.1016/j.engappai.2025.112397

[26] Yang, S., Wei, B., Deng, L., Jin, X., Jiang, M., Huang, Y., et al. (2024). A leader-adaptive particle swarm optimization with dimensionality reduction strategy for feature selection. Swarm and Evolutionary Computation, 91, 101743. https://doi.org/10.1016/j.swevo.2024.101743

[27] Ludwig, S. A. (2025). Guided Particle Swarm Optimization for Feature Selection: Application to Cancer Genome Data. Algorithms, 18(4), 220. https://doi.org/10.3390/a18040220

[28] Zhang, L. (2026). Adaptive diversity-driven multi-strategy particle swarm optimization for global optimization and feature selection. Applied Soft Computing, 201, 115525. https://doi.org/10.1016/j.asoc.2026.115525

[29] Feng, J. and Gong, Z. (2022). A Novel Feature Selection Method With Neighborhood Rough Set and Improved Particle Swarm Optimization. IEEE Access, 10, 33301-33312. https://doi.org/10.1109/access.2022.3162074

[30] Mandal, M., Singh, P. K., Ijaz, M. F., Shafi, J. and Sarkar, R. (2021). A Tri-Stage Wrapper-Filter Feature Selection Framework for Disease Classification. Sensors, 21(16), 5571. https://doi.org/10.3390/s21165571

[31] Le, T. M., Vo, T. M., Pham, T. N. and Dao, S. V. T. (2021). A Novel Wrapper-Based Feature Selection for Early Diabetes Prediction Enhanced With a Metaheuristic. IEEE Access, 9, 7869-7884. https://doi.org/10.1109/access.2020.3047942

[32] Meenachi, L. and Ramakrishnan, S. (2021). Metaheuristic Search Based Feature Selection Methods for Classification of Cancer. Pattern Recognition, 119, 108079. https://doi.org/10.1016/j.patcog.2021.108079

[33] Theerthagiri, P. and Devarayapattana Siddalingaiah, S. (2023). RG-SVM: Recursive gaussian support vector machine based feature selection algorithm for liver disease classification. Multimedia Tools and Applications, 83(20), 59021-59042. https://doi.org/10.1007/s11042-023-17825-1

[34] Yang, M., Wu, Y., Yang, X. B., Liu, T., Zhang, Y., Zhuo, Y., et al. (2023). Establishing a prediction model of severe acute mountain sickness using machine learning of support vector machine recursive feature elimination. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-31797-0

[35] Mohd Nafis, N. S. and Awang, S. (2021). An Enhanced Hybrid Feature Selection Technique Using Term Frequency-Inverse Document Frequency and Support Vector Machine-Recursive Feature Elimination for Sentiment Classification. IEEE Access, 9, 52177-52192. https://doi.org/10.1109/access.2021.3069001

[36] Rakesh, D. K., Anwit, R. and Jana, P. K. (2023). A new ranking-based stability measure for feature selection algorithms. Soft Computing, 27(9), 5377-5396. https://doi.org/10.1007/s00500-022-07767-5

[37] N. Ajay, H. S. Mohan, I. S. Rajesh, and S. Gupta, "A deep learning and blockchain-based security framework for cloud data protection," Journal of Discrete Mathematical Sciences and Cryptography, vol. 29, no. 6, pp. 2553–2587, 2026, doi: 10.47974/JDMSC-2823

[38] Łukaszuk, T., Krawczuk, J., Żyła, K. and Kęsik, J. (2024). Stability of Feature Selection in Multi-Omics Data Analysis. Applied Sciences, 14(23), 11103. https://doi.org/10.3390/app142311103

[39] Narasimhan, G. and Victor, A. (2025). A hybrid approach with metaheuristic optimization and random forest in improving heart disease prediction. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-024-73867-x

[40] Chhabra, D., Juneja, M. and Chutani, G. (2023). An Efficient Ensemble-based Machine Learning approach for Predicting Chronic Kidney Disease. Current Medical Imaging Reviews, 20. https://doi.org/10.2174/1573405620666230508104538

[41] Al Ghamdi, M. and Alyahyan, S. (2026). Enhanced Chronic Kidney Disease Detection: A Hybrid Deep Learning Framework Using Clinical Biomarkers and Ensemble Feature Engineering with DeepCKD-Net. Applied Sciences, 16(6), 3024. https://doi.org/10.3390/app16063024