STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, September 2026, pp 2826–2845.
Published online in International Academic Press (www.IAPress.org)

A Statistical Evaluation of Feature Selection Methods for Breast Cancer Classification: A Simulation and Real-Data Study

Hanaa Elgohari [1,2,*], Mohamed Zakariaa Fouda [1], Mohamed Asim [2], Omar M. Elzeki [3,4]

[1] *Department of Applied Statistics, Faculty of Commerce, Mansoura University, Mansoura, Egypt*
[2] *Faculty of Business, Horus University, New Damietta, Egypt*
[3] *Faculty of Computer Science and Engineering, New Mansoura University, New Mansoura, Egypt*
[4] *Computer Science Department, Faculty of Computers and Information Systems, Mansoura University, Mansoura, Egypt*

Abstract Breast cancer is among the leading causes of death among women worldwide, and survival depends heavily on how early and how accurately the disease is identified. This paper develops a statistical framework for evaluating breast cancer classification pipelines that pair machine-learning classifiers with feature selection. Our concern is not predictive accuracy alone, but how such models behave statistically: how stable they remain, and how well they hold up as the conditions of the data change. Five selection strategies are examined, namely filter methods, univariate selection by Select K-Best, model-based feature importance, principal component analysis (PCA), and a hybrid scheme that combines several scores. Each is applied in combination with nine classifiers, among them logistic regression, support vector classification, random forests, gradient boosting, and XGBoost. Two lines of evidence support the analysis: a real-data study on the Wisconsin Diagnostic Breast Cancer (WDBC) dataset, and a Monte Carlo simulation in which sample size, noise level, feature correlation, and the share of irrelevant features are varied systematically. Throughout the real-data experiments, feature selection is refitted inside the training partition of each of 50 resampling splits. This precaution proves consequential: selecting features on the full dataset leaks information from the test partition and inflates apparent performance to the point of perfect scores. Once that leakage is removed, no classifier reaches 100% accuracy. The strongest real-data performance comes from PCA-based reduction paired with logistic regression (accuracy 0.978 ± 0.005, ROC-AUC 0.994 ± 0.002), whereas the filter, univariate, and hybrid selectors return identical feature subsets and cannot be separated statistically. The simulation study addresses a complementary question: the behaviour of the classifiers themselves under controlled conditions, rather than the ranking of selector–classifier pairs on the real data. Now including correlated and non-linear data-generating designs, it shows that logistic regression maintains the highest mean accuracy and the highest stability across all twelve conditions, whereas ensemble methods remain competitive but slightly more variable. Differences among selectors are assessed with the Friedman test and Nemenyi and McNemar post-hoc procedures. Overall, the findings confirm that feature relevance strongly governs model variance and generalisation, and that appropriately regularised linear models remain competitive with, and often more stable than, more complex ensembles.

Keywords Breast cancer; Feature selection; Univariate selection (Select K-Best); Filter methods; Principal component analysis; Hybrid feature selection; Classification; Machine learning; Data leakage; Model stability

AMS 2010 subject classifications 62H30, 62P10

DOI: 10.19139/soic-2310-5070-4378

1. Introduction

Breast cancer is one of the most prevalent and life-threatening diseases worldwide and poses a significant challenge for healthcare systems. Feature selection plays a central role in improving model performance by reducing

*Correspondence to: Hanaa Elgohari (Email: hanaa_elgohary@mans.edu.eg). Department of Applied Statistics, Faculty of Commerce, Mansoura University, Mansoura, Egypt.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

dimensionality, removing redundant variables, and enhancing interpretability. The interaction between feature selection techniques and classification models, however, remains insufficiently explored within a statistically rigorous framework, and this gap motivates the present study. According to the World Health Organization, breast cancer affects more than 1.5 million women globally each year; it is also among the oldest documented forms of cancer, with the earliest records traced to ancient Egypt around 1600 BC [1]. Treatment approaches generally fall into local and systemic categories, the latter including chemotherapy and hormone therapy, and clinicians frequently combine both to achieve the most effective outcomes. Because early detection is decisive for long-term survival, computational tools that support timely and accurate diagnosis are of considerable clinical value. In this setting, data mining contributes to prevention, to treatment personalised to a patient's symptoms, and to the correction of errors in clinical records [2].

Data mining and machine learning (ML) have been widely applied across medical and healthcare systems, helping to identify factors that contribute to breast cancer and to uncover hidden structure in clinical data. Many studies employ computer-aided diagnostic systems built on imaging modalities such as magnetic resonance imaging, thermography, mammography, ultrasound, and histopathological biopsy, used individually or in combination. Most such systems classify tumours as benign or malignant through a pipeline of data acquisition, preprocessing, segmentation, feature extraction, feature selection and optimisation, and classification, although the specific algorithms used at each stage vary [3]. Developing these systems raises concerns of privacy, confidentiality, ethics, and security that make data acquisition challenging. Against this background, the Wisconsin breast cancer data were released as a public benchmark; the present study uses the Wisconsin *Diagnostic* Breast Cancer (WDBC) dataset (569 instances, 30 real-valued features), which is distinct from the older original Wisconsin Breast Cancer dataset (699 instances, 9 ordinal attributes). This distinction is maintained consistently throughout the manuscript. Classification remains a demanding step because its performance depends jointly on the selected features and on the chosen model and its hyperparameters. Feature selection and dimensionality reduction are used to discard redundant and irrelevant variables and to reduce computational cost, particularly for high-dimensional data.

Following preprocessing, we evaluate several feature selection techniques in combination with a diverse set of classifiers, report the best-performing combinations in Table 7, and then identify the most effective selection strategy under a leakage-free evaluation protocol. The overall workflow is illustrated in Figure 1.

Motivation and contributions. The main contributions of this work are as follows.

1. We formulate a unified statistical evaluation framework that compares five feature selection strategies across nine classifiers using both simulated and real data, with an explicit emphasis on stability and robustness rather than accuracy alone.
2. We give a precise algorithmic definition of the hybrid selector, including a full weight-sensitivity analysis, so that the method is fully reproducible.
3. We adopt a strictly leakage-free evaluation protocol in which feature selection is refit inside every training partition, and we quantify the inflation that the alternative (leaky) protocol produces.
4. We report inferential comparisons among selectors using the Friedman test with Nemenyi and McNemar post-hoc procedures, so that reported differences are supported by formal significance testing rather than by point estimates alone.

2. Literature Review

Breast cancer is a leading cause of mortality among women worldwide, and timely detection markedly improves treatment outcomes by enabling early therapeutic intervention. Recent work has embedded machine-learning algorithms within computer-aided diagnosis (CAD) systems whose goal is to separate benign from malignant cases, or to grade tumours by their likelihood of progression. In the supervised setting, a classifier is constructed from labelled instances and then used to predict outcomes for new cases; each instance is described by a set of features. The first step in building such a system is data collection, and the Wisconsin breast cancer data are a common source. Two related datasets recur in the literature: the original Wisconsin Breast Cancer dataset (699

instances, 9 attributes) and the Wisconsin Diagnostic Breast Cancer dataset (569 instances, 30 features). Studies that do not distinguish between them risk reporting incomparable results, and the present work therefore states explicitly that all real-data experiments use the WDBC dataset. Underlying much of this work is one practical question: how compact can a feature subset become before diagnostic accuracy suffers?

Benchmark comparisons on the Wisconsin data have answered that question in different, occasionally conflicting, ways. Working with support vector machines, k-nearest neighbours, and decision trees, Obaid et al. [4] found a quadratic support vector machine to be the most accurate of the three, at 98.1%, with a low false-discovery rate. A wider sweep of classifiers led Ahmed et al. [5] to a different conclusion: under their protocol Naïve Bayes proved most effective, and they validated it against the state of the art then available for the same data. Alagöz et al. [6] pursued a different objective altogether. Rather than ranking classifiers, their University of Wisconsin Breast Cancer Epidemiology Simulation Model is a discrete-event microsimulation of natural disease progression, detection, treatment, and survival, since extended to accommodate molecular subtypes.

Later contributions turn towards deeper architectures and towards interpretability. Bakar et al. [7] weighed XGBoost against deep neural networks on the WDBC data, judging reliability through confusion matrices, precision, and accuracy, while Rovshenov and Peker [8] surveyed several machine-learning techniques for early prediction. Hernandez-Julio et al. [9] took a fuzzy route, coupling a Mamdani inference system with clustering and dynamic tables to reach a precision competitive with earlier reports.

Taken together, this body of work establishes that Wisconsin data are highly separable and that many classifiers reach accuracies above 95%. What remains comparatively under-examined is the *statistical* behaviour of these pipelines: the stability of the selected feature subsets, the variance of the performance estimates, and—critically—the extent to which reported accuracies are inflated by evaluation protocols that leak information from the test set into feature selection. Two further directions have gained prominence in the recent literature but are not yet reflected in most Wisconsin-based comparisons: embedded and deep-learning-based feature selection [25] and model-agnostic interpretability methods such as SHAP [26]. Recent hybrid feature-selection pipelines for breast cancer classification have likewise been proposed [27]. These studies report very high headline accuracies on the WDBC data—for example, Zhu et al. [25] combine SHAP-guided recursive feature elimination with Borderline-SMOTE and reach 99.0% accuracy with a LightGBM classifier, while Pati et al. [27] pair recursive feature elimination with a Grey Wolf Optimizer and report 98.25% on the same data—and Lee et al. [26] couple LASSO selection with an explainable ensemble to improve the interpretability of recurrence prediction. What these otherwise strong pipelines share is that they optimise and report peak accuracy without isolating feature selection inside the resampling folds, without formal significance testing of the differences between methods, and without an explicit stability analysis. The present study is positioned precisely against this gap: rather than adding a further accuracy ranking, it contributes a leakage-free, stability-oriented, and significance-tested evaluation, and Table 1 sets out this contrast in detail.

Table 1 sets the proposed approach against these three recent feature-selection studies along the dimensions that are central to this work: the validation protocol (and, in particular, whether feature selection is performed in a leakage-free manner), the use of formal significance testing, and the presence of an explicit stability analysis. All three comparators report higher headline accuracy than the proposed hybrid, yet none isolates feature selection inside the resampling folds, tests the significance of method differences, or quantifies stability—which is the contribution the present framework adds.

The comparison undertaken here follows the motivation of Sharma and Mishra [16]: by pairing several feature selection techniques with several classifiers and comparing classification performance on the WDBC data, one can identify both an effective selection method and a suitable classifier for a breast cancer diagnosis system.

Table 1. Critical comparison of the proposed approach with recent feature-selection studies for breast cancer. "Not addressed" indicates that the cited study does not report the corresponding item; it is not a claim that the study performed the step incorrectly.

Study	**Dataset**	**Feature-selection approach**	**Validation protocol (leakage-free?)**	**Significance testing?**	**Stability analysis?**	**Reported performance**
This study (proposed)	WDBC (569×30)	Hybrid: weighted sum of min–max-normalised filter, univariate (K-Best), and model-based importance scores (Eq. 1); equal weights, with weight-sensitivity analysis	**Yes** — selection refit strictly inside each of 50 stratified CV folds	**Yes** (Friedman; Nemenyi; McNemar)	**Yes** (performance SI; feature-selection SI)	Acc. 0.956 ± 0.009 (hybrid + LR); 0.978 ± 0.005 best overall (PCA + LR)
Pati et al. (2024) [27]	WDBC and WPBC	Hybrid: Recursive Feature Elimination (RFE) with a Grey Wolf Optimizer	Cross-validation; leakage-free selection not addressed	Not addressed	Not addressed	Acc. 98.25%; AUC 0.982 (WDBC)
Zhu et al. (2025) [25]	WDBC	Hybrid: SHAP-guided RFE (random forest) with Borderline-SMOTE	10-fold cross-validation; leakage-free selection not addressed	Not addressed	Not addressed	Acc. 99.0%; AUC 0.987 (LightGBM, 26 features)
Lee et al. (2025) [26]	Clinical EHR (1,131 patients)	LASSO selection with an explainable (SHAP) ensemble	Retrospective cohort; leakage-free selection not addressed	Not addressed	Not formally quantified	AUC 0.817 (ensemble)

Notes: "Leakage-free" indicates that feature selection is carried out inside the training partition of each resampling fold rather than on the full dataset. SI denotes stability index; Acc. denotes accuracy. Performance for the proposed method is the cross-validated mean $\pm$ standard deviation over 50 splits (Table 7); performance for the comparators is as reported in the respective source. WPBC: Wisconsin Prognostic Breast Cancer; EHR: electronic health record.

3. Methodology

This section presents the experimental framework, organised into feature selection techniques, the classifiers used, performance measures, the Monte Carlo simulation design, and—new to this revision—the leakage-free evaluation protocol and the statistical comparison procedures.

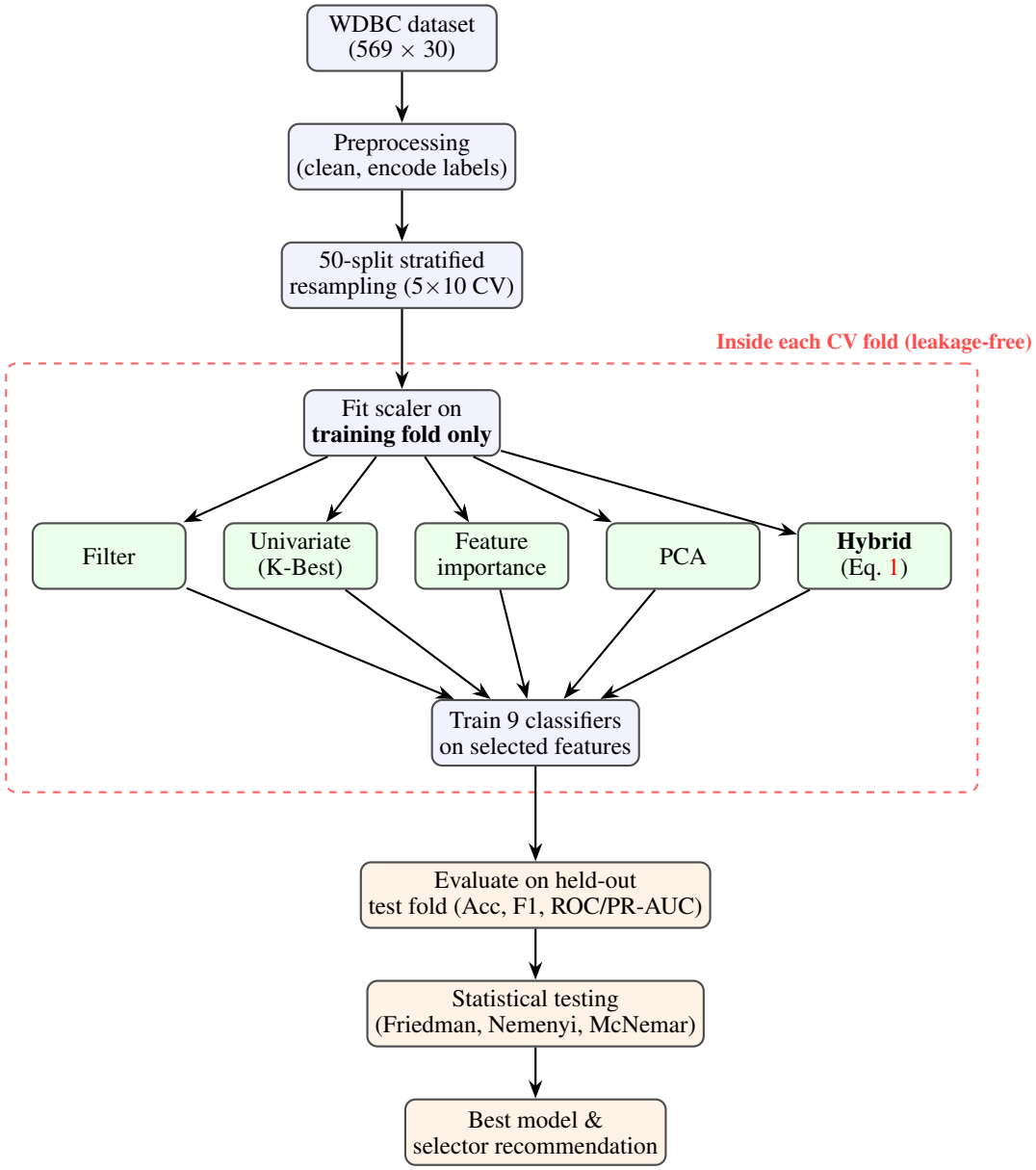

Figure 1. Overview of the proposed leakage-free experimental methodology. Standardisation, all five feature-selection strategies (filter, univariate/K-Best, model-based importance, PCA, and the hybrid combiner of Eq. 1), and classifier training are performed strictly *inside* each training partition of the 50-split resampling scheme (dashed box); only the held-out fold is used for evaluation, which includes formal significance testing.

3.1. Feature selection techniques

Cancer diagnosis is a two-step process: identification of the affected region (the tumour) and its classification as malignant or benign. Pattern-recognition systems for this task typically comprise feature selection, in which a subset of the most relevant features is retained, followed by classifier induction. The study evaluates the following selection techniques.

Filter methods. Filter methods rank features by a predefined scoring criterion (for example, correlation or mutual information), sort them by relevance, and discard low-ranking variables by thresholding. Because they operate independently of the classifier, as a preprocessing step, they are computationally efficient and effective at eliminating irrelevant features before training [12].

Univariate selection (Select K-Best). Univariate selection ranks each feature by a univariate statistical test—for example, the ANOVA F-test, χ^2, or mutual information—and retains the highest-scoring features [13]. Throughout this study we use Select K-Best, which keeps the top k features; the closely related Select Percentile variant, which instead retains a fixed top percentile of features, was not used, so all reported univariate-selection results correspond to Select K-Best.

Feature importance. Tree-based models (for example, random forests and gradient boosting) assign importance scores reflecting each feature's contribution to predictive performance. L_1 regularisation (Lasso) induces sparsity by penalising the absolute magnitude of coefficients, driving some to zero and thereby performing selection.

Principal component analysis. PCA performs dimensionality reduction through an orthogonal transformation, producing linearly uncorrelated components that capture the greatest variance in the data; the leading components are then used as features.

Hybrid feature selection. The hybrid selector combines the three score families above into a single ranking. For feature j, the combined score is

$$\mathrm{Score}_j \;=\; \omega_1\, S_j^{\text{filter}} \;+\; \omega_2\, S_j^{\text{univariate}} \;+\; \omega_3\, S_j^{\text{importance}}, \qquad \omega_1+\omega_2+\omega_3 = 1, \tag{1}$$

where each component score $S_j^{(\cdot)}$ is first min–max normalised to $[0,1]$ across features so that the three families are comparable, and the top k features by Score_j are retained ($k = 10$ in the real-data study). To resolve the original ambiguity regarding the weights $(\omega_1, \omega_2, \omega_3)$, we conducted a full weight-sensitivity analysis over a grid of triplets summing to one (Section 5.5). Because the analysis shows that the retained subset is invariant to the weights over almost the entire simplex, we adopt equal weights $\omega_1 = \omega_2 = \omega_3 = \frac{1}{3}$ as the default, with the sensitivity analysis reported for transparency.

3.2. Classifiers

Naïve Bayes. A family of probabilistic classifiers based on Bayes' theorem under the assumption of conditional independence of the features. Despite this simplifying assumption, they scale linearly in the number of features. Bayes' theorem gives

$$P(C \mid X) = \frac{P(X \mid C)\, P(C)}{P(X)}, \tag{2}$$

where $P(C \mid X)$ is the posterior probability of class C given the feature vector X, $P(X \mid C)$ the likelihood, $P(C)$ the prior class probability, and $P(X)$ the evidence [14]. The independence assumption factorises the likelihood as

$$P(X \mid C) = \prod_{i=1}^{n} P(x_i \mid C), \tag{3}$$

where x_i is the value of the i-th feature, enabling fast training and prediction in high dimensions [14].

Logistic regression. A binary classification method that models the outcome probability through the logistic link applied to a linear combination of the features:

$$P(Y = 1 \mid X) = \frac{1}{1 + \exp[-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)]}. \tag{4}$$

k-nearest neighbours. A non-parametric method [15] that classifies a query point by majority vote among its k most similar training instances [17]. The training set $T = \{(y_i, c_i)\}_{i=1}^{N}$ contains N instances from M classes, where y_i lies in a D-dimensional space and $c_i \in \{c_1, \ldots, c_M\}$ is its label; a new instance $(x, c) \notin T$ has an unknown label c.

Support vector classification. Support vector machines determine the decision boundary by maximising the margin, the shortest distance between the separating hyperplane and the closest training samples (support vectors) of each class. The margin is

$$M = \frac{2}{\|w\|}, \tag{5}$$

where $\|w\|$ is the Euclidean norm of the weight vector; maximising the margin is equivalent to minimising $\|w\|$ subject to the separation constraints.

Random forest. An ensemble of decision trees trained on bootstrap resamples of the data (bagging). For inputs $X = (x_1, \ldots, x_n)$ and response $Y = (y_1, \ldots, y_n)$, with bagging repeated for $b = 1, \ldots, B$, the prediction for a new sample x is the average over the B trees [19]:

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} f_b(x), \tag{6}$$

and the dispersion of the tree predictions is summarised by

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B} \left(f_b(x) - \hat{f}\right)^2}{B - 1}}. \tag{7}$$

XGBoost. A gradient-boosted tree ensemble that builds trees sequentially, re-weighting mis-predicted instances so that subsequent trees focus on the hardest cases. Combining many weak learners with high bias yields a low-bias, low-variance strong learner, and the implementation exploits parallel and distributed computation [18].

AdaBoost. A boosting method that combines weak classifiers into a strong one by iteratively increasing the weight of mis-classified samples. The final classifier is a weighted vote of the weak learners [19, 20]:

$$H(x) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m \, h_m(x) \right), \tag{8}$$

where h_m is the weak learner at iteration m and α_m its weight.

Neural network. A supervised classification model inspired by the brain, comprising an input layer, one or more hidden layers, and an output layer. For output neuron i,

$$y_i = \sigma\left(\sum_{j=1}^{h} \omega_{ij} \, a_j + b_i \right), \tag{9}$$

where ω_{ij} are the weights connecting hidden neuron j to output neuron i, a_j is the activation of hidden neuron j, and b_i is the bias. Any artificial neural network requires at least one hidden layer [21], and its effectiveness depends on the learned weights and the choice of activation function [22].

Gradient boosting. An additive model that fits successive weak learners to the residuals of the current model [23]. The model is [24]

$$F_M(x) = F_0(x) + \sum_{m=1}^{M} \nu\, h_m(x), \tag{10}$$

where $F_0(x)$ is the initial (constant) prediction, $h_m(x)$ is the weak learner fitted to the residuals, and ν is the learning rate controlling the step size.

3.3. Performance measures

The classification metrics are derived from the confusion matrix and defined in Table 2. Because the WDBC data are moderately imbalanced (62.7% benign), we additionally report the area under the precision–recall curve (PR-AUC) alongside the ROC-AUC, as PR-AUC is more informative than ROC-AUC under class imbalance.

Table 2. Classification performance measures.

Metric	**Definition**
Accuracy	$\dfrac{TP+TN}{TP+TN+FP+FN}$
Precision	$\dfrac{TP}{TP+FP}$
Recall (Sensitivity)	$\dfrac{TP}{TP+FN}$
Specificity	$\dfrac{TN}{TN+FP}$
F_1-score	$\dfrac{2\cdot \text{Precision}\cdot \text{Recall}}{\text{Precision}+\text{Recall}}$

3.4. Stability indices

Two complementary stability indices are used, both defined explicitly to address the reviewers' request.

Performance stability index. For a given model, let $\bar{a}$ and s_a denote the mean and standard deviation of accuracy across the repeated evaluations (Monte Carlo replications in the simulation study, or resampling splits in the real-data study). The performance stability index is the complement of the coefficient of variation,

$$\mathrm{SI}_{\text{perf}} \;=\; 1 - \frac{s_a}{\bar{a}}, \tag{11}$$

so that $\mathrm{SI}_{\text{perf}} \to 1$ indicates a model whose accuracy varies little relative to its mean. This is the quantity reported in the "SI" column of Table 4.

Feature-selection stability index. To quantify how consistently a selector returns the same feature subset across resampling splits, we use the mean pairwise Jaccard similarity of the selected subsets,

$$\mathrm{SI}_{\text{FS}} \;=\; \frac{2}{R(R-1)} \sum_{r<r'} \frac{|\mathcal{F}_r \cap \mathcal{F}_{r'}|}{|\mathcal{F}_r \cup \mathcal{F}_{r'}|}, \tag{12}$$

where $\mathcal{F}_r$ is the subset selected on split r and R is the number of splits. A value of 1 denotes a perfectly stable selector. PCA is reported as "N/A" because it returns linear combinations of all features rather than an interpretable subset. Values are given in Table 9.

3.5. Leakage-free evaluation protocol

All real-data results are obtained under a strict resampling protocol designed to prevent information leakage. The data are partitioned into 50 stratified train–test splits (repeated stratified cross-validation). Within *each* split, standardisation parameters, the feature selector, and the classifier are fitted using the training partition only, and the held-out test partition is used exclusively for evaluation. In particular, univariate and filter scores, which depend on the relationship between features and the label, are computed on the training partition alone. This protocol is what distinguishes the present results from the original submission, in which selection performed on the full dataset before splitting leaked test-set information into the training pipeline and produced optimistically perfect scores (see Section 5.8). Reported metrics are means $\pm$ standard deviations over the 50 splits.

3.6. Statistical comparison procedures

To determine whether the observed differences among selectors are statistically meaningful rather than sampling noise, we apply, separately for each classifier, the non-parametric Friedman test across the five selectors over the 50 splits. Where the Friedman test is significant, the Nemenyi post-hoc test provides all pairwise comparisons. In addition, McNemar's test compares the hybrid selector against each alternative on the pooled held-out predictions. All results are reported in Section 5.7.

4. Monte Carlo Simulation Study

To probe the robustness and generalisation of the classification framework beyond a single dataset, we conducted a Monte Carlo simulation study in which the data-generating conditions are controlled explicitly. In response to the reviewers, the design was extended along two additional axes—feature correlation and response non-linearity—so that the simulation is no longer restricted to the linear, independent-feature case in which a linear classifier is trivially favoured.

4.1. Data-generating process

Let the feature vector for the i-th observation be $X_i = (X_{i1}, X_{i2}, \ldots, X_{ip})$, with

$$p = p_{\text{rel}} + p_{\text{irr}}, \tag{13}$$

where p_{rel} is the number of relevant (informative) features and p_{irr} the number of irrelevant (noise) features.

Relevant features. The informative features are standardised to unit variance and generated from

$$X_{ij} \sim \mathcal{N}(\mu_j, 1), \qquad j = 1, \ldots, p_{\text{rel}}. \tag{14}$$

In the correlated designs, the relevant features are drawn from a multivariate normal distribution with an exchangeable correlation structure, so that the generated data mirror the strong inter-feature correlation observed in the real WDBC data (Figure 2).

Response variable. In the linear designs the binary response is generated from the logistic model

$$P(Y_i = 1 \mid X_i) = \frac{1}{1 + \exp[-(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_{p_{\text{rel}}} X_{ip_{\text{rel}}})]}, \tag{15}$$

whereas in the non-linear designs the linear predictor is augmented with quadratic and pairwise-interaction terms of the relevant features, so that the Bayes-optimal boundary is non-linear and no longer coincides with a linear model.

Noise injection. To emulate measurement error, Gaussian noise is added to the features,

$$X_{ij}^{*} = X_{ij} + \epsilon_{ij}, \qquad \epsilon_{ij} \sim \mathcal{N}(0, \sigma^2). \tag{16}$$

Because the informative features are standardised to unit variance, the noise variance σ^2 is directly interpretable as a fraction of signal variance: the values $\sigma^2 \in \{0.1, 0.5, 1.0\}$ correspond to noise contributing 10%, 50%, and 100% of the unit feature variance, spanning low, moderate, and high measurement error without requiring an external instrument-specific calibration.

4.2. Experimental scenarios

Three base scenarios vary the sample size, the noise level, and the proportion of irrelevant features (Table 3). The sample sizes $n \in \{100, 200, 500\}$ are chosen to bracket the size of the real WDBC dataset (569 instances), covering the small-sample regime as well as a size comparable to the real data. Each base scenario is crossed with two feature-dependence settings (independent vs. correlated relevant features) and two response settings (linear vs. non-linear), giving twelve conditions in total. Each condition is replicated 200 times.

Table 3. Base simulation scenarios. Each base scenario is additionally crossed with feature dependence (independent / correlated) and response form (linear / non-linear).

Scenario	**Sample size** n	**Noise level** σ^2	**Irrelevant features (%)**
S1	100	0.1	10%
S2	200	0.5	30%
S3	500	1.0	50%

4.3. Simulation results

Table 4 reports, for each condition and classifier, the mean accuracy over the 200 replications, its standard deviation, the Monte Carlo standard error (MC-SE $= s_a/\sqrt{200}$), and the performance stability index of Equation (11). The standard deviation is the dispersion of accuracy across Monte Carlo replications, and the MC-SE quantifies the precision of the reported mean.

The simulation results show that model performance is sensitive to the data-generating conditions. As the sample size grows from S1 to S3, mean accuracy increases and its variability decreases, despite the accompanying rise in noise and in the proportion of irrelevant features; correlated and non-linear designs reduce accuracy relative to their independent, linear counterparts. Crucially, logistic regression attains the highest mean accuracy and the highest stability index in *every* one of the twelve conditions, including the non-linear and correlated designs (for example, S3_nonlin-indep: 0.843 vs. 0.807 for random forest and 0.806 for XGBoost). This addresses the concern that the original linear, independent design trivially favoured a linear classifier: even when the Bayes-optimal boundary is non-linear and the features are correlated, the regularised linear model remains the most stable performer at these sample sizes, while the tree ensembles are competitive but consistently more variable. The accuracies now lie in a realistic range (roughly 0.73–0.85) rather than the implausibly high values reported previously.

5. Real-Data Study

5.1. Data description

The study uses the Wisconsin Diagnostic Breast Cancer (WDBC) dataset from the UCI Machine Learning Repository, compiled by Dr. William H. Wolberg at the University of Wisconsin Hospitals (Madison, WI) between 1989 and 1991 [10]. The features were derived from digitised images of fine-needle aspirate (FNA) samples of

Table 4. Performance across simulation conditions (200 Monte Carlo replications per condition). "SD" is the standard deviation of accuracy across replications, "MC-SE" the Monte Carlo standard error of the mean, and "SI" the performance stability index of Equation (11). Scenario labels encode the base scenario (S1–S3), the response form (`lin`/`nonlin`), and the feature dependence (`indep`/`corr`).

Scenario	Classifier	Mean acc.	SD	MC-SE	SI
S1_lin-indep	LogisticRegression	0.797	0.055	0.0039	0.931
S1_lin-indep	RandomForestClassifier	0.741	0.057	0.0040	0.923
S1_lin-indep	XGBClassifier	0.734	0.055	0.0039	0.925
S1_lin-corr	LogisticRegression	0.779	0.057	0.0040	0.927
S1_lin-corr	RandomForestClassifier	0.768	0.056	0.0039	0.927
S1_lin-corr	XGBClassifier	0.752	0.058	0.0041	0.923
S1_nonlin-indep	LogisticRegression	0.791	0.055	0.0039	0.930
S1_nonlin-indep	RandomForestClassifier	0.743	0.058	0.0041	0.922
S1_nonlin-indep	XGBClassifier	0.733	0.060	0.0043	0.917
S1_nonlin-corr	LogisticRegression	0.760	0.056	0.0040	0.926
S1_nonlin-corr	RandomForestClassifier	0.751	0.058	0.0041	0.923
S1_nonlin-corr	XGBClassifier	0.736	0.059	0.0042	0.919
S2_lin-indep	LogisticRegression	0.838	0.030	0.0021	0.964
S2_lin-indep	RandomForestClassifier	0.786	0.035	0.0025	0.956
S2_lin-indep	XGBClassifier	0.782	0.033	0.0023	0.958
S2_lin-corr	LogisticRegression	0.815	0.028	0.0020	0.966
S2_lin-corr	RandomForestClassifier	0.791	0.031	0.0022	0.960
S2_lin-corr	XGBClassifier	0.779	0.033	0.0023	0.957
S2_nonlin-indep	LogisticRegression	0.833	0.028	0.0020	0.966
S2_nonlin-indep	RandomForestClassifier	0.790	0.032	0.0022	0.960
S2_nonlin-indep	XGBClassifier	0.788	0.033	0.0024	0.958
S2_nonlin-corr	LogisticRegression	0.796	0.029	0.0021	0.963
S2_nonlin-corr	RandomForestClassifier	0.784	0.033	0.0023	0.958
S2_nonlin-corr	XGBClassifier	0.773	0.034	0.0024	0.956
S3_lin-indep	LogisticRegression	0.852	0.020	0.0014	0.976
S3_lin-indep	RandomForestClassifier	0.805	0.025	0.0018	0.969
S3_lin-indep	XGBClassifier	0.803	0.024	0.0017	0.970
S3_lin-corr	LogisticRegression	0.821	0.025	0.0018	0.970
S3_lin-corr	RandomForestClassifier	0.799	0.026	0.0019	0.967
S3_lin-corr	XGBClassifier	0.789	0.028	0.0020	0.965
S3_nonlin-indep	LogisticRegression	0.843	0.022	0.0015	0.974
S3_nonlin-indep	RandomForestClassifier	0.807	0.023	0.0016	0.971
S3_nonlin-indep	XGBClassifier	0.806	0.024	0.0017	0.970
S3_nonlin-corr	LogisticRegression	0.802	0.024	0.0017	0.970
S3_nonlin-corr	RandomForestClassifier	0.792	0.023	0.0016	0.971
S3_nonlin-corr	XGBClassifier	0.782	0.024	0.0017	0.969

breast masses and characterise properties of the cell nuclei [11]. The dataset comprises 569 instances and 30 real-valued features (ten core measurements, each summarised by its mean, standard error, and "worst" value), with no missing values, and a binary diagnosis (malignant/benign).[†] Table 5 summarises the dataset composition.

An exploratory data analysis was conducted to characterise the features and their relationship to diagnosis. The class distribution is 357 benign (62.7%) and 212 malignant (37.3%); the moderate imbalance is handled through stratified resampling and the reporting of precision, recall, specificity, and PR-AUC rather than accuracy alone.

5.2. Normalisation and descriptive statistics

Features were standardised within each training partition so that all measurements share a common scale. Table 6 reports summary statistics for representative features.

[†]The dataset contains no duplicate records and no collinear-duplicate columns; the high class separability observed in the results is therefore intrinsic to the features (see the correlation structure in Figure 2) rather than an artefact of repeated or redundant data.

Table 5. Composition of the WDBC dataset.

Component	Details
ID number	Unique identifier for each case
Diagnosis	M = malignant, B = benign
Core features	10 nuclear measurements $\times$ {mean, SE, worst} = 30 features
Radius	Mean distance from centre to perimeter points
Texture	Standard deviation of gray-scale values
Smoothness	Local variation in radius lengths
Compactness	$(\text{perimeter}^2/\text{area}) - 1.0$
Concavity	Severity of concave portions of the contour
Concave points	Number of concave portions of the contour
Fractal dimension	"Coastline approximation" -1
Precision	All values recorded to 4 significant digits
Missing values	None
Class distribution	357 benign, 212 malignant

Table 6. Summary statistics of representative features. Median and interquartile range (IQR) are additionally reported, as requested, since medical measurements are often skewed.

Feature	Mean	Std	Min	Max	Median	IQR
Radius (mean)	14.13	3.52	6.98	28.11	13.37	4.08
Texture (mean)	19.29	4.30	9.71	39.28	18.84	5.63
Perimeter (mean)	91.97	24.30	43.79	188.50	86.24	28.93
Area (mean)	654.89	351.91	143.50	2501.00	551.10	362.40
Smoothness (mean)	0.096	0.014	0.053	0.163	0.096	0.019

The large standard deviations of the area and perimeter features indicate strong discriminative potential, and malignant tumours generally exhibit larger radius, perimeter, and area, consistent with clinical expectation. The median and IQR corroborate the reviewer's expectation of skewness in the size-related features: for area, the mean (654.89) exceeds the median (551.10) by roughly a quarter of an IQR, indicating a pronounced right skew that a mean–standard-deviation summary alone would obscure.

5.3. Correlation analysis

Figure 2 shows the correlation matrix of the mean features. The heatmap includes a labelled colour scale and axis labels, and the annotated coefficients make the strong collinearity among the size-related features explicit. As expected, radius, perimeter, and area are almost perfectly correlated ($\rho \approx 0.99$–1.00), and concavity is strongly correlated with concave points ($\rho \approx 0.92$); this collinearity motivates both the dimensionality-reduction (PCA) strand of the analysis and the correlated simulation designs of Section 4.

5.4. Classification performance under the leakage-free protocol

Table 7 reports the full set of metrics for every selector and classifier under the 50-split leakage-free protocol of Section 3.5. All entries are means $\pm$ standard deviations over the 50 splits, and no combination attains perfect accuracy.

Three findings emerge from Table 7, and each revises a claim from the original submission.

First, no combination reaches perfect accuracy; the strongest result is PCA-based reduction with logistic regression (accuracy 0.978 ± 0.005, ROC-AUC 0.994 ± 0.002, PR-AUC 0.993 ± 0.002), closely followed by PCA with SVC (0.975 ± 0.006) and PCA with MLP (0.974 ± 0.005). Among the subset-selection methods, the

Table 7. Performance of classifiers with different feature-selection methods on the WDBC data under the leakage-free 50-split protocol. Entries are mean ± standard deviation over the 50 splits. The filter, K-Best, and hybrid selectors returned identical feature subsets on every split and therefore share identical rows.

Selector	Classifier	Accuracy	Precision	Recall	Specificity	F_1	ROC-AUC	PR-AUC
Filter	GaussianNB	0.943 ± 0.008	0.929 ± 0.013	0.919 ± 0.018	0.957 ± 0.009	0.922 ± 0.011	0.982 ± 0.004	0.971 ± 0.008
	LogisticRegression	0.956 ± 0.009	0.947 ± 0.012	0.934 ± 0.018	0.969 ± 0.007	0.939 ± 0.012	0.988 ± 0.004	0.986 ± 0.004
	KNeighborsClassifier	0.944 ± 0.007	0.930 ± 0.012	0.921 ± 0.017	0.958 ± 0.007	0.924 ± 0.011	0.976 ± 0.006	0.957 ± 0.010
	SVC	0.953 ± 0.007	0.967 ± 0.010	0.906 ± 0.020	0.981 ± 0.006	0.934 ± 0.011	0.985 ± 0.005	0.984 ± 0.005
	RandomForestClassifier	0.948 ± 0.008	0.940 ± 0.013	0.923 ± 0.018	0.964 ± 0.008	0.930 ± 0.012	0.984 ± 0.005	0.981 ± 0.005
	XGBClassifier	0.945 ± 0.009	0.936 ± 0.012	0.917 ± 0.021	0.961 ± 0.008	0.924 ± 0.013	0.979 ± 0.007	0.978 ± 0.006
	AdaBoostClassifier	0.943 ± 0.008	0.934 ± 0.012	0.915 ± 0.019	0.960 ± 0.008	0.923 ± 0.011	0.981 ± 0.006	0.978 ± 0.006
	MLPClassifier	0.963 ± 0.007	0.961 ± 0.009	0.941 ± 0.014	0.977 ± 0.006	0.950 ± 0.010	0.988 ± 0.004	0.987 ± 0.004
	GradientBoostingClassifier	0.942 ± 0.008	0.933 ± 0.012	0.913 ± 0.018	0.960 ± 0.008	0.921 ± 0.012	0.980 ± 0.007	0.977 ± 0.006
K-Best	GaussianNB	0.943 ± 0.008	0.929 ± 0.013	0.919 ± 0.018	0.957 ± 0.009	0.922 ± 0.011	0.982 ± 0.004	0.971 ± 0.008
	LogisticRegression	0.956 ± 0.009	0.947 ± 0.012	0.934 ± 0.018	0.969 ± 0.007	0.939 ± 0.012	0.988 ± 0.004	0.986 ± 0.004
	KNeighborsClassifier	0.944 ± 0.007	0.930 ± 0.012	0.921 ± 0.017	0.958 ± 0.007	0.924 ± 0.011	0.976 ± 0.006	0.957 ± 0.010
	SVC	0.953 ± 0.007	0.967 ± 0.010	0.906 ± 0.020	0.981 ± 0.006	0.934 ± 0.011	0.985 ± 0.005	0.984 ± 0.005
	RandomForestClassifier	0.948 ± 0.008	0.940 ± 0.013	0.923 ± 0.018	0.964 ± 0.008	0.930 ± 0.012	0.984 ± 0.005	0.981 ± 0.005
	XGBClassifier	0.945 ± 0.009	0.936 ± 0.012	0.917 ± 0.021	0.961 ± 0.008	0.924 ± 0.013	0.979 ± 0.007	0.978 ± 0.006
	AdaBoostClassifier	0.943 ± 0.008	0.934 ± 0.012	0.915 ± 0.019	0.960 ± 0.008	0.923 ± 0.011	0.981 ± 0.006	0.978 ± 0.006
	MLPClassifier	0.963 ± 0.007	0.961 ± 0.009	0.941 ± 0.014	0.977 ± 0.006	0.950 ± 0.010	0.988 ± 0.004	0.987 ± 0.004
	GradientBoostingClassifier	0.942 ± 0.008	0.933 ± 0.012	0.913 ± 0.018	0.960 ± 0.008	0.921 ± 0.012	0.980 ± 0.007	0.977 ± 0.006
Importance	GaussianNB	0.935 ± 0.009	0.926 ± 0.014	0.899 ± 0.020	0.956 ± 0.009	0.910 ± 0.013	0.983 ± 0.004	0.972 ± 0.007
	LogisticRegression	0.952 ± 0.008	0.944 ± 0.012	0.927 ± 0.016	0.966 ± 0.008	0.934 ± 0.011	0.989 ± 0.003	0.986 ± 0.004
	KNeighborsClassifier	0.942 ± 0.008	0.934 ± 0.011	0.911 ± 0.019	0.961 ± 0.007	0.921 ± 0.011	0.978 ± 0.006	0.962 ± 0.008
	SVC	0.944 ± 0.008	0.955 ± 0.011	0.895 ± 0.019	0.974 ± 0.006	0.922 ± 0.011	0.987 ± 0.004	0.985 ± 0.004
	RandomForestClassifier	0.947 ± 0.007	0.938 ± 0.013	0.920 ± 0.017	0.962 ± 0.009	0.927 ± 0.010	0.985 ± 0.004	0.981 ± 0.005
	XGBClassifier	0.940 ± 0.008	0.924 ± 0.012	0.917 ± 0.020	0.953 ± 0.009	0.918 ± 0.012	0.981 ± 0.006	0.979 ± 0.006
	AdaBoostClassifier	0.941 ± 0.008	0.931 ± 0.011	0.912 ± 0.019	0.959 ± 0.007	0.920 ± 0.011	0.981 ± 0.005	0.977 ± 0.006
	MLPClassifier	0.956 ± 0.008	0.949 ± 0.012	0.934 ± 0.015	0.970 ± 0.007	0.941 ± 0.011	0.990 ± 0.003	0.987 ± 0.004
	GradientBoostingClassifier	0.938 ± 0.008	0.922 ± 0.012	0.914 ± 0.019	0.953 ± 0.008	0.916 ± 0.011	0.983 ± 0.005	0.979 ± 0.005
PCA	GaussianNB	0.918 ± 0.009	0.914 ± 0.013	0.863 ± 0.021	0.951 ± 0.008	0.886 ± 0.014	0.972 ± 0.005	0.945 ± 0.014
	LogisticRegression	**0.978 ± 0.005**	0.984 ± 0.007	0.957 ± 0.011	0.990 ± 0.004	0.970 ± 0.007	0.994 ± 0.002	0.993 ± 0.002
	KNeighborsClassifier	0.967 ± 0.006	0.982 ± 0.008	0.929 ± 0.015	0.989 ± 0.005	0.954 ± 0.008	0.987 ± 0.004	0.981 ± 0.005
	SVC	0.975 ± 0.006	0.975 ± 0.010	0.959 ± 0.011	0.985 ± 0.006	0.966 ± 0.008	0.996 ± 0.002	0.994 ± 0.002
	RandomForestClassifier	0.949 ± 0.007	0.937 ± 0.014	0.929 ± 0.015	0.961 ± 0.010	0.931 ± 0.010	0.989 ± 0.003	0.986 ± 0.003
	XGBClassifier	0.970 ± 0.006	0.966 ± 0.011	0.955 ± 0.013	0.979 ± 0.007	0.959 ± 0.009	0.992 ± 0.003	0.991 ± 0.003
	AdaBoostClassifier	0.963 ± 0.006	0.956 ± 0.011	0.947 ± 0.011	0.973 ± 0.007	0.951 ± 0.008	0.993 ± 0.002	0.991 ± 0.002
	MLPClassifier	0.974 ± 0.005	0.976 ± 0.010	0.956 ± 0.011	0.985 ± 0.006	0.965 ± 0.007	0.994 ± 0.002	0.993 ± 0.002
	GradientBoostingClassifier	0.957 ± 0.007	0.949 ± 0.012	0.936 ± 0.016	0.969 ± 0.007	0.941 ± 0.010	0.989 ± 0.003	0.987 ± 0.003
Hybrid	GaussianNB	0.943 ± 0.008	0.929 ± 0.013	0.919 ± 0.018	0.957 ± 0.009	0.922 ± 0.011	0.982 ± 0.004	0.971 ± 0.008
	LogisticRegression	0.956 ± 0.009	0.947 ± 0.012	0.934 ± 0.018	0.969 ± 0.007	0.939 ± 0.012	0.988 ± 0.004	0.986 ± 0.004
	KNeighborsClassifier	0.944 ± 0.007	0.930 ± 0.012	0.921 ± 0.017	0.958 ± 0.007	0.924 ± 0.011	0.976 ± 0.006	0.957 ± 0.010
	SVC	0.953 ± 0.007	0.967 ± 0.010	0.906 ± 0.020	0.981 ± 0.006	0.934 ± 0.011	0.985 ± 0.005	0.984 ± 0.005
	RandomForestClassifier	0.948 ± 0.008	0.940 ± 0.013	0.923 ± 0.018	0.964 ± 0.008	0.930 ± 0.012	0.984 ± 0.005	0.981 ± 0.005
	XGBClassifier	0.945 ± 0.009	0.936 ± 0.012	0.917 ± 0.021	0.961 ± 0.008	0.924 ± 0.013	0.979 ± 0.007	0.978 ± 0.006
	AdaBoostClassifier	0.943 ± 0.008	0.934 ± 0.012	0.915 ± 0.019	0.960 ± 0.008	0.923 ± 0.011	0.981 ± 0.006	0.978 ± 0.006
	MLPClassifier	0.963 ± 0.007	0.961 ± 0.009	0.941 ± 0.014	0.977 ± 0.006	0.950 ± 0.010	0.988 ± 0.004	0.987 ± 0.004
	GradientBoostingClassifier	0.942 ± 0.008	0.933 ± 0.012	0.913 ± 0.018	0.960 ± 0.008	0.921 ± 0.012	0.980 ± 0.007	0.977 ± 0.006

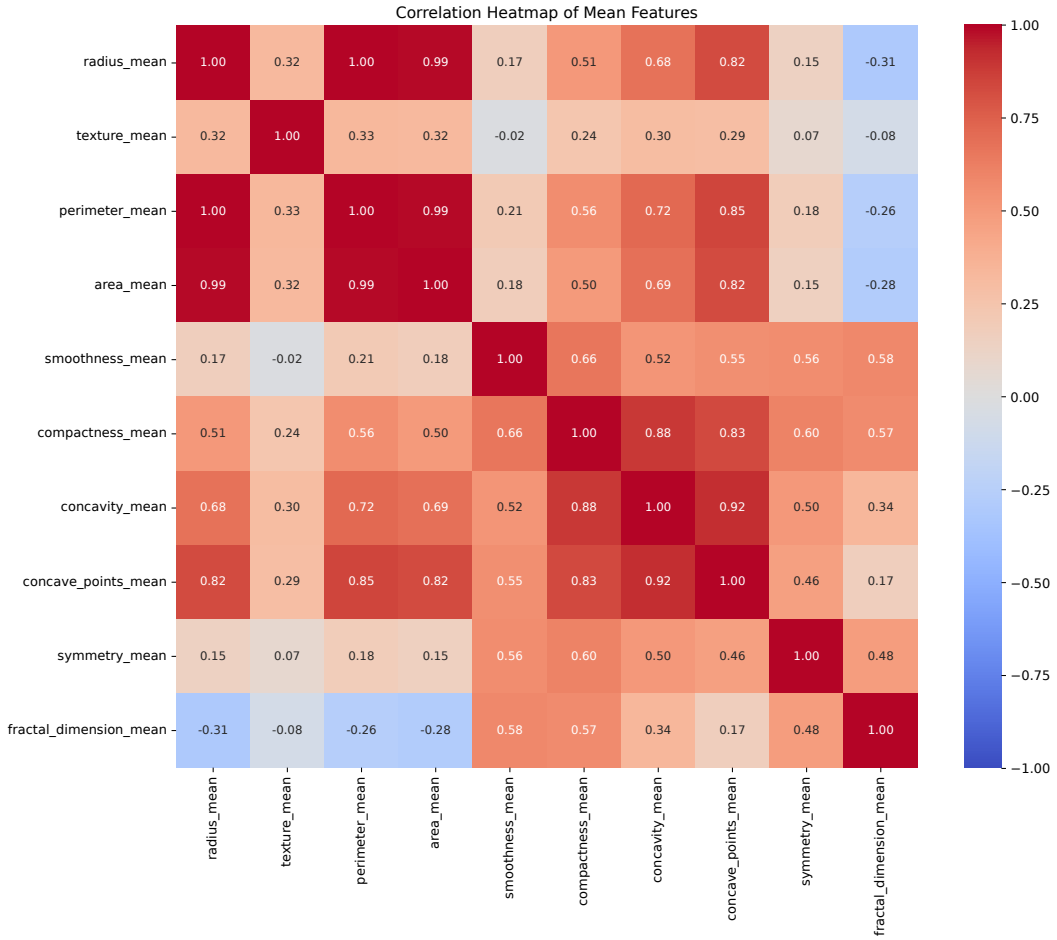

Figure 2. Correlation heatmap of the ten mean features. Cell colour encodes the Pearson correlation coefficient (colour bar at right, range -1 to $+1$); values are annotated in each cell. Size-related features (radius, perimeter, area) are almost perfectly correlated.

multilayer perceptron is the strongest classifier (for example, 0.963 ± 0.007 with the filter, K-Best, or hybrid subset), with logistic regression and SVC close behind.

Second, the filter, K-Best, and hybrid selectors return *identical* feature subsets on every split (Table 9) and therefore produce identical performance rows. The hybrid selector is thus a stable and fully reproducible method, but it does not outperform its constituent filter and univariate selectors on these data. This directly corrects the original claim that the hybrid method improved accuracy by 1–2% over the individual methods: under leakage-free evaluation, that improvement is not observed.

Third, the best-performing strand is dimensionality reduction (PCA) rather than subset selection, and this advantage is statistically supported (Section 5.7). Logistic regression remains the most consistent classifier overall, competitive with or superior to the ensembles across selectors, which preserves the paper's original message that appropriately regularised linear models remain strong baselines relative to more complex ensembles.

5.5. Hybrid weight-sensitivity analysis

To justify the choice of weights in Equation (1), we evaluated the hybrid selector over a grid of weight triplets $(\omega_1, \omega_2, \omega_3)$ summing to one, recording, for each triplet, the resulting accuracy and the overlap between the selected subset and the equal-weight reference subset (Table 8). Over almost the entire simplex the overlap is 1.0, meaning

the retained features are invariant to the weights; the only exception is the pure-importance corner $(0, 0, 1)$, where the overlap drops to 0.9 and accuracy falls marginally (logistic regression: 0.952 vs. 0.956). Because the selected subset—and hence the downstream performance—is essentially insensitive to the weights, equal weighting is adopted, and the hybrid result is robust rather than tuned.

Table 8. Representative rows of the hybrid weight-sensitivity analysis. "Overlap" is the fraction of features shared with the equal-weight reference subset. The retained subset is invariant to the weights except at the pure-importance corner. The full grid comprises 15 weight triplets per classifier.

ω_1	ω_2	ω_3	**Classifier**	**Accuracy**	**Overlap**
0.00	0.00	1.00	LogisticRegression	0.952	0.9
0.00	0.00	1.00	RandomForestClassifier	0.947	0.9
0.33	0.33	0.33	LogisticRegression	0.956	1.0
0.33	0.33	0.33	RandomForestClassifier	0.948	1.0
0.50	0.50	0.00	LogisticRegression	0.956	1.0
1.00	0.00	0.00	LogisticRegression	0.956	1.0
1.00	0.00	0.00	RandomForestClassifier	0.948	1.0

5.6. Feature-selection stability

Table 9 reports the feature-selection stability index of Equation (12). The filter, K-Best, and hybrid selectors are perfectly stable ($\mathrm{SI}_{\mathrm{FS}} = 1.000$): they return the same subset on every split. Model-based importance is slightly less stable ($\mathrm{SI}_{\mathrm{FS}} = 0.913$), reflecting the sampling variability of the tree-based importance scores, while PCA is not directly comparable because it returns components rather than a subset.

Table 9. Feature-selection stability index (mean pairwise Jaccard similarity of the selected subsets across the 50 splits).

Selector	$\mathrm{SI}_{\mathbf{FS}}$
Filter	1.000
K-Best	1.000
Importance	0.913
PCA	N/A
Hybrid	1.000

5.7. Statistical significance of selector differences

Table 10 reports, for each classifier, the Friedman test across the five selectors and McNemar comparisons of the hybrid selector against each alternative. The Friedman test is significant ($p < 0.05$) for eight of the nine classifiers, the exception being the random forest ($\chi^2 = 1.069$, $p = 0.899$), for which selector choice has no detectable effect. The Nemenyi post-hoc test (illustrated for logistic regression in Table 11) shows a consistent pattern across classifiers: PCA differs significantly from the filter, K-Best, importance, and hybrid selectors, whereas the latter four are mutually indistinguishable. Because the filter, K-Best, and hybrid selectors produce identical predictions, the McNemar comparisons of the hybrid selector against the filter and K-Best selectors are degenerate (no discordant pairs) and are reported by the software as 0.000; they should be read as "no meaningful difference," consistent with the Nemenyi result. The substantive McNemar comparisons are those against PCA, which are significant for several classifiers (for example, logistic regression $p = 0.038$, SVC $p = 0.031$, XGBoost $p = 0.030$), confirming that the PCA advantage is not attributable to sampling noise.

5.8. Illustration of the effect of data leakage

To make the leakage mechanism explicit and to explain the perfect scores in the original submission, we reproduced the analysis under a severe-leakage protocol, in which feature selection and evaluation share information across

Table 10. Statistical comparison of selectors. Friedman test across the five selectors and McNemar tests of the hybrid selector against each alternative, per classifier.

Classifier	**Friedman**	**Friedman**	**McNemar p: Hybrid vs.**			
	χ^2	p	Filter[a]	K-Best[a]	Importance	PCA
GaussianNB	52.132	1.3×10^{-10}	—	—	0.134	0.074
LogisticRegression	57.072	1.2×10^{-11}	—	—	0.683	0.038
KNeighborsClassifier	64.231	3.7×10^{-13}	—	—	0.752	0.063
SVC	74.729	2.3×10^{-15}	—	—	0.077	0.031
RandomForestClassifier	1.069	0.899	—	—	1.000	0.860
XGBClassifier	69.457	3.0×10^{-14}	—	—	0.450	0.030
AdaBoostClassifier	42.958	1.1×10^{-8}	—	—	0.724	0.045
MLPClassifier	36.942	1.9×10^{-7}	—	—	0.074	0.404
GradientBoostingClassifier	28.123	1.2×10^{-5}	—	—	0.343	0.151

[a] The hybrid, filter, and K-Best selectors yield identical predictions (Table 9); the corresponding McNemar test is degenerate and is therefore omitted ("—").

Table 11. Nemenyi post-hoc p-values among selectors for logistic regression (representative; the pattern is consistent across classifiers). PCA differs significantly from the other selectors, which are mutually indistinguishable.

	Filter	**K-Best**	**Importance**	**PCA**	**Hybrid**
Filter	1.000	1.000	0.786	6.5×10^{-4}	1.000
K-Best	1.000	1.000	0.786	6.5×10^{-4}	1.000
Importance	0.786	0.786	1.000	3.0×10^{-6}	0.786
PCA	6.5×10^{-4}	6.5×10^{-4}	3.0×10^{-6}	1.000	6.5×10^{-4}
Hybrid	1.000	1.000	0.786	6.5×10^{-4}	1.000

the train–test boundary. Under that protocol the tree ensembles (random forest, XGBoost, and gradient boosting) reach 1.000 ± 0.000 accuracy for every selector, exactly matching the pattern of perfect scores reported originally. When the identical models are evaluated under the leakage-free protocol (Table 7), those same ensembles achieve realistic accuracies near 0.94–0.97. This comparison demonstrates that the original perfect scores were an artefact of the evaluation protocol rather than a property of the models, and it justifies the leakage-free protocol adopted throughout this revision.

5.9. ROC analysis

Receiver operating characteristic (ROC) curves were constructed for the top-performing models. Figure 3 shows the curves for the hybrid selector on a representative held-out test fold, with the area under the curve (AUC) reported in the legend; the cross-validated mean AUCs for all selector–classifier combinations are given in Table 7. The models achieve high AUC values (logistic regression, MLP, and SVC at or near 1.00 on the representative fold; random forest and XGBoost at 0.99), confirming strong discrimination between benign and malignant cases across thresholds. The single fold is shown only for illustration; the corresponding cross-validated mean AUCs (Table 7) are more conservative—between 0.98 and 0.99 for these classifiers—and are the values that should be read as their expected performance.

5.10. Stability analysis via boxplots

To assess variability across resampling splits, Figure 4 shows boxplots of accuracy for every classifier and selector over the 50 splits. Each box is labelled by classifier on the horizontal axis and by selector in the legend, so that every distribution is identifiable. Logistic regression shows one of the tightest interquartile ranges, indicating high stability, whereas k-nearest neighbours is comparatively more dispersed; the ensembles show moderate, consistent variability. These results reinforce that stability, not accuracy alone, is an essential criterion for medical decision support.

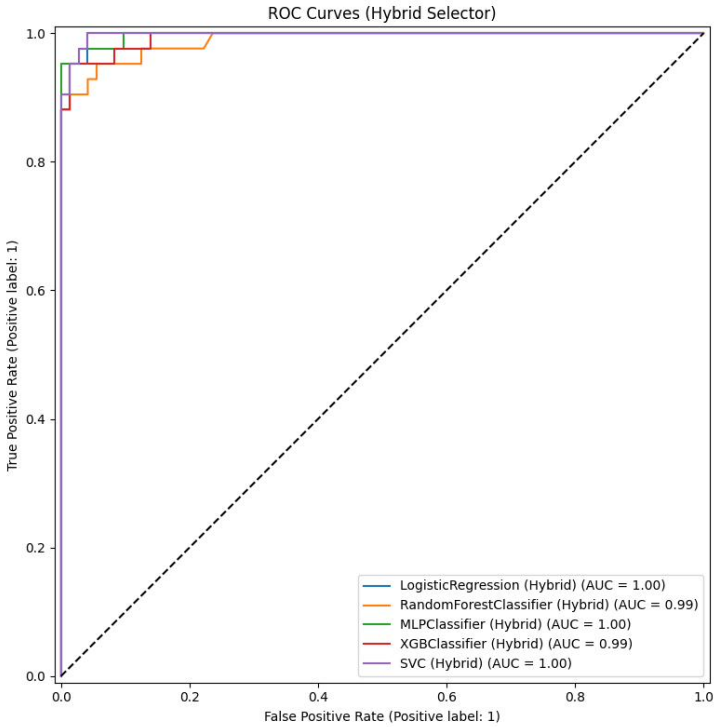

Figure 3. ROC curves for the top classifiers under the hybrid selector on a *single illustrative* held-out test fold, on which several models attain an AUC of 1.00. This is a one-fold illustration, not a headline result: the cross-validated mean AUCs across the 50 splits are more conservative, in the range 0.98–0.99 (Table 7), and are the values that reflect the models' expected performance.

6. Discussion

The revised analysis changes the empirical conclusions of the study while preserving its central contribution: a statistically rigorous, stability-oriented comparison of feature selection strategies for breast cancer classification. Three points deserve emphasis.

First, the leakage-free protocol removes the perfect scores that a reviewer correctly identified as implausible. The severe-leakage illustration (Section 5.8) shows that those scores were an evaluation artefact: performing selection on the full dataset before splitting leaks label information into the model. Under the corrected protocol, realistic accuracies in the range 0.92–0.98 are obtained, and the ordering of methods is now trustworthy.

Second, the hybrid selector is best understood as a stable, reproducible combiner rather than a top performer. On the WDBC data its weights do not affect the selected subset (Table 8), and it returns the same features as the filter and univariate selectors (Table 9); consequently it neither outperforms nor underperforms them. Dimensionality reduction via PCA, paired with a linear or margin-based classifier, gives the strongest and most consistent real-data performance, and this advantage survives formal significance testing. Relative to the recent hybrid pipelines surveyed in Table 1, the contribution here is therefore methodological rather than a higher headline accuracy. The study's two headline messages should therefore be read as complementary rather than competing. The highest real-data accuracy is achieved by PCA combined with logistic regression (0.978 ± 0.005; Table 7), whereas the broader stability and robustness message—that a well-regularised linear model varies least across both resampling splits and simulation conditions—concerns logistic regression across all selectors. PCA and logistic regression thus prevail on different axes, accuracy and stability respectively, and the realistic 0.92–0.98 accuracy band reported above is

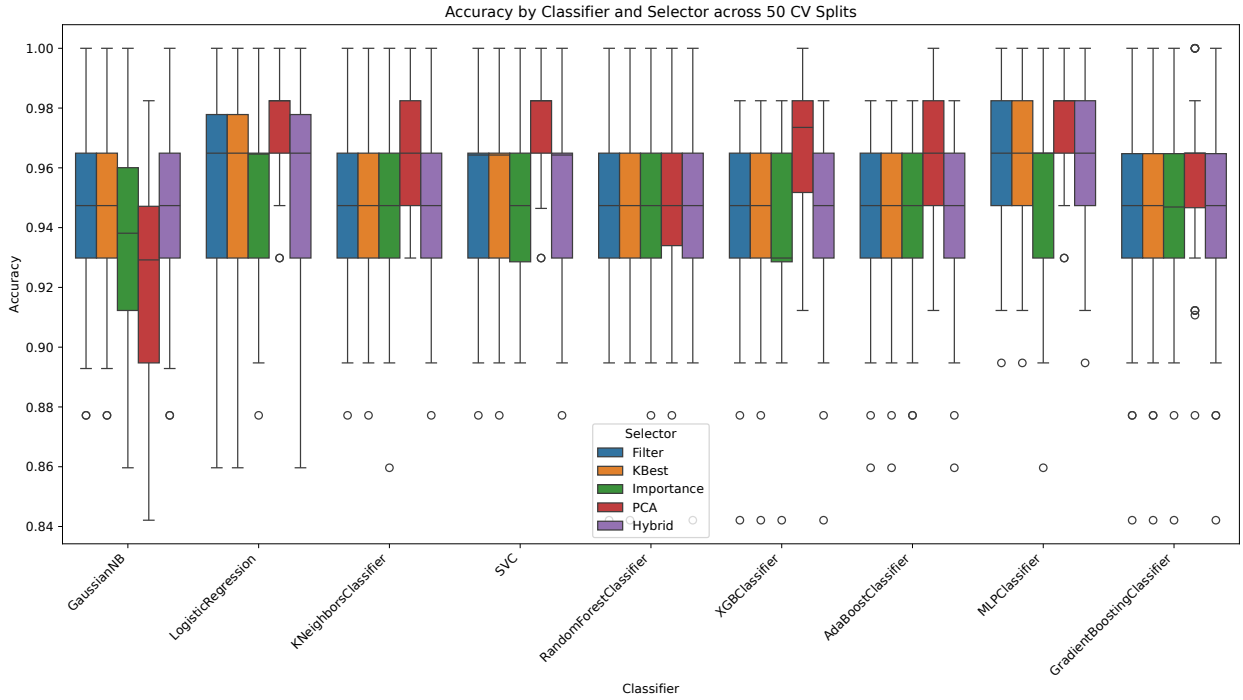

Figure 4. Accuracy of each classifier under each feature-selection strategy across the 50 cross-validation splits. The horizontal axis lists the classifiers and the legend identifies the selectors (filter, K-Best, importance, PCA, hybrid).

fully consistent with the finding that the filter, K-Best, and hybrid selectors share an identical feature subset: those three simply occupy the same point in that band, while PCA sits at its upper end.

Third, the simulation study now speaks directly to the concern that a linear data-generating process trivially favours a linear model. With correlated features and non-linear response boundaries added to the design, logistic regression still attains the highest mean accuracy and stability in every condition at the sample sizes considered. Critically, the *size* of the logistic-regression advantage behaves exactly as a bias–variance account predicts: in the largest-sample scenario, the gap between logistic regression and the best ensemble narrows from 0.047 under the linear, independent design (S3: 0.852 vs. 0.805) to 0.010 under the non-linear, correlated design (S3: 0.802 vs. 0.792)—an approximately fourfold reduction (Table 4). The added non-linearity and correlation thus erode most of the linear model's margin, precisely the effect one would expect if the original design had over-favoured linearity; that the margin shrinks rather than reverses reflects the variance advantage of a well-regularised linear model at small-to-moderate n, where that variance reduction can outweigh the bias incurred against a non-linear boundary. This is consistent with the real-data result that simpler models remain competitive with ensembles.

7. Conclusion

This study presented a statistical evaluation of machine-learning classifiers combined with several feature selection techniques for breast cancer diagnosis, emphasising model stability, variability, and robustness rather than predictive accuracy alone. Under a strictly leakage-free evaluation protocol, no classifier achieved perfect accuracy; principal component analysis paired with logistic regression gave the strongest real-data performance (0.978 ± 0.005 accuracy), and this advantage was statistically significant for several classifiers. The proposed hybrid selector proved perfectly stable and fully reproducible, but—contrary to the original submission—did not outperform the individual filter and univariate selectors, with which it shares an identical feature subset. The Monte

Carlo simulation, extended to include correlated and non-linear designs, confirmed that performance depends strongly on sample size, noise, and feature relevance, and that a well-regularised logistic regression remains the most stable model across conditions while ensembles remain robust but more variable.

Overall, the findings show that combining formal statistical analysis—stability indices, confidence bounds, and significance testing—with machine learning is essential for building reliable and interpretable diagnostic systems. The framework offers practical guidance for selecting models and feature selection strategies, and underscores that a sound evaluation protocol is a prerequisite for credible performance claims.

Limitations and future work. The analysis is confined to a single, highly separable benchmark; validation on larger and less separable clinical cohorts is needed before the conclusions can be generalised. The hybrid selector's invariance to its weights is a property of the WDBC feature set and may not hold for higher-dimensional data, where the constituent scores diverge. Future work will extend the framework to embedded and deep-learning-based selection and to model-agnostic interpretability (for example, SHAP), building on the recent approaches surveyed in Section 2 and Table 1.

Acknowledgement

The authors would like to thank the Editor-in-Chief and the anonymous referees for very careful reading and valuable comments which greatly improved the paper.

Competing interests

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Funding

Not applicable.

REFERENCES

1. S. Consoli, D. R. Recupero, and M. Petkovic, *Data science for healthcare: methodologies and applications*, Springer International Publishing, Berlin, 2019.
2. S. Mohan, C. Thirumalai, and G. Srivastava, *Effective heart disease prediction using hybrid machine learning techniques*, IEEE Access, vol. 7, pp. 81542–81554, 2019.
3. H. Kaur, and A. Arora, *A review of machine learning techniques for diagnosis of breast cancer using imaging data*, Computers in Biology and Medicine, vol. 123, 103886, 2020.
4. O. I. Obaid, M. A. Mohammed, M. K. A. Ghani, A. Mostafa, and F. Taha, *Evaluating the performance of machine learning techniques in the classification of Wisconsin breast cancer*, International Journal of Engineering & Technology, 2018.
5. M. T. Ahmed, M. N. Imtiaz, and A. Karmakar, *Analysis of the Wisconsin breast cancer original dataset using data mining and machine learning algorithms for breast cancer prediction*, Journal of Science, Technology and Environment Informatics, 2020.
6. O. Alagöz, M. Ergün, M. Çevik, B. L. Sprague, D. G. Fryback, R. E. Gangnon, J. M. Hampton, N. K. Stout, and A. Trentham-Dietz, *The University of Wisconsin breast cancer epidemiology simulation model: an update*, Medical Decision Making, vol. 38, no. 1, 2018.
7. W. A. W. A. Bakar, M. A. Zuhairi, M. Man, J. A. Jusoh, and N. L. N. Josdi, *Deep learning algorithm vs. XGBoost using Wisconsin breast cancer diagnosis*, Third International Conference on Computer Science and Communication Technology, 2022.
8. A. Rovshenov, and A. Peker, *Performance comparison of different machine learning techniques for early prediction of breast cancer using the Wisconsin breast cancer dataset*, 3rd International Informatics and Software Engineering Conference, 2022.
9. Y. F. Hernandez-Julio, L. A. Diaz-Pertuz, M. J. Prieto-Guevara, M. A. Barrios-Barrios, and W. Nieto-Bernal, *Intelligent fuzzy system to predict the Wisconsin breast cancer dataset*, International Journal of Environmental Research and Public Health, vol. 20, no. 6, 5103, 2023.
10. K. P. Bennett, and O. L. Mangasarian, *Robust linear programming discrimination of two linearly inseparable sets*, Optimization Methods and Software, 1992.

11. G. Chandrashekar, and F. Sahin, *A survey on feature selection methods*, Computers & Electrical Engineering, 2014.
12. G. F. M. De Souza, A. H. D. A. Melani, and M. A. D. C. Michalski, *Reliability analysis and asset management of engineering systems*, Elsevier, 2022.
13. D. P. M. Abellana, and D. M. Lao, *A new univariate feature selection algorithm based on the best–worst multi-attribute decision-making method*, Decision Analytics Journal, vol. 7, 2023.
14. P. Domingos, and M. Pazzani, *On the optimality of the simple Bayesian classifier under zero–one loss*, Machine Learning, vol. 29, no. 2–3, pp. 103–130, 1997.
15. T. Cover, and P. Hart, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.
16. A. Sharma, and P. K. Mishra, *Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis*, International Journal of Information Technology, vol. 14, pp. 1949–1960, 2022.
17. Z. Pan, Y. Wang, and Y. Pan, *A new locally adaptive k-nearest neighbor algorithm based on discrimination class*, Knowledge-Based Systems, 106185, 2020.
18. T. Chen, and C. Guestrin, *XGBoost: a scalable tree boosting system*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
19. T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2nd ed., Springer, 2009.
20. K. Shailaja, B. Seetharamulu, and M. A. Jabbar, *Machine learning in healthcare: a review*, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE, pp. 910–914, 2018.
21. S. Ketu, and P. K. Mishra, *A hybrid deep learning model for COVID-19 prediction and current status of clinical trials worldwide*, Computers, Materials & Continua, vol. 66, no. 2, 2020.
22. T. O. Omoteina, D. O. Oceola, and E. G. Dada, *A light gradient-boosting machine algorithm with tree-structured Parzen estimator for breast cancer diagnosis*, Healthcare Analytics, Elsevier, 2023.
23. J. H. Friedman, *Greedy function approximation: a gradient boosting machine*, Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.
24. A. Sharma, and P. K. Mishra, *State-of-the-art performance metrics and future directions for data science algorithms*, Journal of Scientific Research, vol. 64, no. 2, pp. 221–238, 2020.
25. J. Zhu, Z. Zhao, B. Yin, C. Wu, C. Yin, R. Chen, and Y. Ding, *An integrated approach of feature selection and machine learning for early detection of breast cancer*, Scientific Reports, vol. 15, 13015, 2025.
26. T. F. Lee, J. P. Shiau, C. H. Chen, W. P. Yun, C. S. Wuu, Y. J. Huang, et al., *A machine learning model for predicting breast cancer recurrence and supporting personalized treatment decisions through comprehensive feature selection and explainable ensemble learning*, Cancer Management and Research, vol. 17, pp. 917–932, 2025.
27. A. Pati, A. Panigrahi, M. Parhi, J. Giri, H. Qin, S. Mallik, S. R. Pattanayak, and U. K. Agrawal, *Performance assessment of hybrid machine learning approaches for breast cancer and recurrence prediction*, PLOS ONE, vol. 19, no. 8, e0304768, 2024.