Tatra Mt. Math. Publ. **93** (2026), 107–143
DOI: 10.2478/tmmp-2026-0010

TATRA MOUNTAINS
Mathematical Publications

PREDICTION OF DEPRESSION FOR INPATIENTS WITH CHRONIC HEART FAILURE

Igor Mračka — Tibor Žáčik

Mathematical Institute, Slovak Academy of Sciences, Bratislava, Slovakia

ABSTRACT. Depression is a common and prognostically important comorbidity of chronic heart failure (CHF). Because datasets from cohort studies in this research area are often small and strongly imbalanced, and a ready-to-apply training and evaluation procedure for binary classification in this setting is still lacking, we provide a rigorous methodological overview and elaborate it into a straightforward protocol: leave-one-out cross-validation (LOOCV) with fixed seeds and default library hyperparameters, and a full panel of confusion matrix metrics, including imbalance-adjusted summaries. As an illustration of the chosen methodology, we reformulated the 9-item Patient Health Questionnaire (PHQ-9) screening as a binary classification task and compared three deliberately chosen standard methods—ridge-regularized logistic regression, decision tree, and RUSBoost—on a public cohort published by Cheng and co-authors in 2023. These methods are evaluated at each learner's default operating point over the range of PHQ-9 binarization thresholds $\theta \in \{1, \dots, 15\}$.

RUSBoost obtained the best results at the standard diagnostic PHQ-9 cut-off $\theta = 10$—sensitivity 0.641, specificity 0.619, accuracy 0.621 and a sensitivity–specificity gap of 0.022. As the only learner with built-in imbalance handling, it also consistently outperforms the two observation-weighted baselines on these metrics across the entire range of thresholds. However, absolute sensitivity and specificity remain below clinically ambitious levels for all three models. We attribute part of this performance ceiling to selection bias introduced during cohort acquisition and eligibility filtering.

2020 Mathematics Subject Classification: Primary 62H30; Secondary 92C50, 68T05.
Keywords: depression prediction, chronic heart failure, small dataset, RUSBoost.
Funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project 09I05-03-V02-00084 - *Digital solutions in support of mental health in patients with Chronic Heart Failure* (DigiMent).

1. Introduction

Depressive symptoms affect a sizable fraction of chronic heart failure (CHF) patients. A widely cited meta-analysis estimates the point prevalence of clinically significant depression in CHF at roughly 21 percent [36] (classified as severe in half of those affected), which is several times the rate observed in the general population. Depression in CHF is not only common but also prognostically meaningful: it is associated with reduced treatment adherence, increased rehospitalization, and higher all-cause mortality [1]. Routine screening with instruments such as the Hamilton Depression Rating Scale (HAM-D) or the Hospital Anxiety and Depression Scale (HADS) is recommended in modern heart failure guidelines [29], but applying them to every admitted patient is not always feasible in busy inpatient settings. In these environments, deploying the short, 9-item Patient Health Questionnaire (PHQ-9), or an automated decision-support tool that prioritizes which patients should receive formal screening, could be highly beneficial.

A natural question is whether routinely collected demographic, clinical, and laboratory data can be used to flag patients at an elevated risk of depressive symptoms, thereby prioritizing them for clinical supervision. Some works, such as [10, 18, 19, 26], search for predictors of depression and show the potential to use them for screening purposes. In our work we formulate the screening task as a binary classification problem characterized by challenges typical of institutional clinical studies. The available cohort is usually small (n in the low hundreds). Once a higher PHQ-9 cut-off is applied, the prevalence of screen-positive cases drops significantly, creating a severe class imbalance; consequently, the absolute number of positive events n_+ can be small enough that multivariable models face a dangerously low events-per-variable ratio even when total number of screened patients n itself appears adequate. Furthermore, the misclassification costs of false positives and false negatives are highly asymmetric, and the data are frequently constrained by localized selection biases.

The CHF domain has seen growing interest in machine learning models for related endpoints such as worsening heart failure [30, 42]. The literature on depression-specific classifiers in CHF cohorts remains sparse [20, 39, 43].

This work is part of the DIGIMENT project (*Digital Solutions for Mental Health Support of Patients with Chronic Heart Failure*, April 2024–June 2026) [45, 46], which aims to develop digital tools for mental health support in CHF within a Central European setting. Until a localized cohort is established, DIGIMENT Work Package 2 (WP2) uses the public Cheng et al. data release [6, 7] as a methodological testbed. A binary classifier at a fixed cut-off omits within-class score order and is therefore, in principle, a simpler learning problem than a regressor that incorporates ordering across all cases. Therefore,

during the DIGIMENT research we preferred a methodology that uses a robust classifier at different cut-offs θ as a probe of the score range. Moreover, if such a test shows bad performance, the most robust alternative approach for modelling the ordered diagnostic score is to reformulate the problem as ordinal classification and use binary decomposition. Approaches based on binary decomposition in a similar context are presented, for example, in [25, 47]. During the DIGIMENT research we tried plenty of classifiers and regression approaches on the chosen testbed dataset, but almost all of that effort led only to poor performance. The very fact that this dataset posed a challenge also made it a prime candidate for developing a robust binary classification methodology.

Within the DIGIMENT project, we evaluated a broader suite of learners—including k-NN, kernel Naive Bayes, a quadratic-kernel SVM, and many others. However, rather than presenting a shallow comparison of all these models, we choose to report three representative methods in full: ridge-regularized Logistic Regression (LR), a Decision Tree (DT), and Random Under-Sampling Boosting (RUSBoost), all configured with hyperparameters fixed to library defaults without internal tuning. For each classifier we probe performance across a chosen range of the PHQ-9 total score; these threshold-specific learners could further serve as building blocks for Bayesian classifier fusion [21] (or related schemes such as Frank–Hall-like methods [13]).

We do not claim to present a deployable clinical decision-support tool here. The Cheng et al. cohort, which uses the PHQ-9, and the selected algorithms are chosen only as an example of applying the methodology to small, imbalanced datasets in the scope of predicting depression for CHF patients. The protocol adopted in this study is prepared for direct application to other similar training data and can potentially incorporate other psychometric tools (different from the PHQ-9). We discuss four critical issues in small-sample clinical machine learning: (i) validation protocol design when n and n_+ are small, (ii) evaluation metrics and learners under class imbalance, (iii) the selection of operating points on the ROC curve when misclassification costs are unknown, and (iv) model transferability from a single-region training sample to a different clinical context (in this case, Central Europe).

The rest of the paper is organized as follows. Section 2 introduces the PHQ-9 instrument and the cut-off definition used throughout. Section 3 develops binary classification metrics, class-imbalance considerations, and asymmetric error costs. Section 4 addresses small-sample modelling (events per variable, dataset merging, missing-data handling), leave-one-out cross-validation (LOOCV), leave-pair-out cross-validation (LPOCV), out-of-fold AUROC, and the evaluation protocol adopted in this study. Section 5 describes the Cheng et al. cohort and the PHQ-9 threshold sweep; Section 6 documents the experimental environment and the default LR, DT, and RUSBoost configurations. Section 7 reports the LOOCV

threshold sweeps, the $\theta = 10$ summary, and the LOOCV–LPOCV comparison for RUSBoost at high cut-offs. Section 8 discusses transferability to the Central European DIGIMENT context and Section 9 summarises performance limits and implications for follow-up work. For reproducibility, the MATLAB script `CLASSIFIERS.m` is provided in Appendix A.

2. The PHQ-9 instrument

Given that we illustrate the methodology on the Cheng et al. cohort, which uses the PHQ-9, we also adopt this instrument in the paper. The PHQ-9 is a nine-item questionnaire derived from the depression module of the Primary Care Evaluation of Mental Disorders [22]. In primary-care validation, it is usually self-administered on paper; however, in the Cheng et al. protocol, it was completed on the spot using a paper-and-pencil form during a face-to-face interview, with trained interviewers reading items verbatim for illiterate patients [7]. Each item is scored from 0 (not at all) to 3 (nearly every day); the total score therefore ranges from 0 to 27. The conventional categorization is: 0–4 none, 5–9 mild, 10–14 moderate, 15–19 moderately severe, and 20–27 severe depressive symptoms. The cut-off θ used in most screening studies is ≥ 10 for "probable major depression"; this threshold yielded approximately 88 % sensitivity and 88 % specificity against a structured clinical interview in the original primary-care validation [22]. We treat $\theta = 10$ as the default PHQ-9 cut-off and also analyse alternative cut-offs $\theta \in \{1, \ldots, 15\}$.

Definition 2.1. A patient is called *PHQ-9-positive at cut-off* θ if his/her total PHQ-9 score $S \in \{0, \ldots, 27\}$ satisfies $S \geq \theta$. The default value is $\theta = 10$.

The PHQ-9 is not the only validated screener among psychometric tools used in cardiac populations. Alternatives include the Beck Depression Inventory II (BDI-II, 21 items), the depression subscale of the Hospital Anxiety and Depression Scale (HADS-D, 7 items), the Center for Epidemiologic Studies Depression Scale (CES-D, 20 items), and, for older patients, the Geriatric Depression Scale (GDS-15 or GDS-30). In favor of the PHQ-9 are its brevity (typically completed in ≤ 4 minutes), free availability for clinical use, direct alignment with Diagnostic and Statistical Manual of Mental Disorders (DSM) diagnostic criteria, and good cross-cultural validity. Its main drawback in the cardiac setting is the somatic content of items 3 (sleep), 4 (fatigue), 5 (appetite), and 7 (concentration), all of which can be inflated by CHF itself; HADS-D used in [35] was designed precisely to suppress this somatic overlap [50] and is sometimes preferred in cardiology research.

More fundamentally, all these questionnaires are psychometric instruments based on self-reported depressive symptoms; they do not measure depression as such, but only a proxy for a patient's symptom burden. Therefore, a machine-learning decision-support tool trained on labels derived from such instruments can at best supplement that proxy, subject to their errors, and may additionally introduce new ones. Due to this, replacing any questionnaire with some machine-learning decision-support tool can be meaningful only if it legitimately decreases logistical burden—as in the Cheng et al. study [7]—or meaningfully increases population coverage.

3. Binary classification in medical diagnostics

A binary diagnostic classifier is a function $h : \mathcal{X} \to \{0, 1\}$ that maps a feature vector $x \in \mathcal{X} \subseteq \mathbb{R}^d$ to a predicted class label. In practice, this mapping is typically obtained by thresholding a continuous score function $s : \mathcal{X} \to \mathbb{R}$ at a specific value τ conventionally referred to as the operating point of the classifier:

$$h_\tau(x) = \mathbf{1}\left[s(x) \geq \tau\right]. \tag{1}$$

Here $\mathbf{1}$ is the indicator function, returning 1 for a positive diagnostic result when the condition in brackets is satisfied and 0 otherwise.

The score may be a probability estimate $P(Y = 1 \mid x)$, the output of a logistic regression, the vote share of an ensemble, or a distance-weighted neighborhood frequency (in k-NN). Once the score has been learned, the operating point τ in equation (1) can be tuned independently of training. This flexibility makes the Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves the natural summary objects in this setting. An ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) across operating points τ.

3.1. Class imbalance

A binary screening task is *imbalanced* when one class is markedly less frequent than the other, a pattern clearly illustrated by our cohort (see Section 5 and Table 1). Consequently, overall accuracy is a misleading metric under class imbalance. A naive majority-class baseline achieves an accuracy of $1 - p_+$ (where $p_+ = n_+/n$). When p_+ is small the accuracy can approach 1 without any actual learning. This demonstrates that not all performance metrics are suitable in this context.

For clarity and consistency, we present the complete suite of confusion-matrix-derived performance metrics here, some of which specifically target this imbalance issue. Let TP, TN, FP, and FN denote the true-positive, true-negative, false-positive, and false-negative counts, respectively.

Furthermore, let $P = n_+ = \mathrm{TP} + \mathrm{FN}$ and $N = n_- = \mathrm{TN} + \mathrm{FP}$ represent the actual class totals, where $n = P + N$ is the total cohort size. We use:

(a) True positive rate (sensitivity), also recall, $\mathrm{TPR} = \mathrm{Sn} = \mathrm{TP}/P$, and true negative rate, $\mathrm{TNR} = \mathrm{Sp} = \mathrm{TN}/N$ [16, 51];

(b) False positive rate, $\mathrm{FPR} = 1 - \mathrm{TNR} = \mathrm{FP}/N$, and false negative rate, $\mathrm{FNR} = 1 - \mathrm{TPR} = \mathrm{FN}/P$ [51];

(c) Positive predictive value (precision), $\mathrm{PPV} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$, and negative predictive value, $\mathrm{NPV} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FN})$, both of which depend on the model predictions and therefore change rapidly with the operating point [9, 38];

(d) Overall accuracy, $\mathrm{Acc} = (\mathrm{TP} + \mathrm{TN})/n$, and balanced accuracy, $\mathrm{BalAcc} = \frac{1}{2}(\mathrm{TPR} + \mathrm{TNR})$;

(e) Youden's index [49], $J = \mathrm{TPR} + \mathrm{TNR} - 1$, and the balance gap, $|\mathrm{Sn} - \mathrm{Sp}|$;

(f) F_1 score, $F_1 = 2\,\mathrm{TP}/(2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN})$ [9];

(g) Matthews correlation coefficient [28]:

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}};$$

(h) Cohen's kappa [8], $\kappa = (\mathrm{Acc} - p_e)/(1 - p_e)$, with the chance-agreement term $p_e = p_+\widehat{p}_+ + (1 - p_+)(1 - \widehat{p}_+)$, where $p_+ = P/n$ is the actual positive prevalence and $\widehat{p}_+ = (\mathrm{TP} + \mathrm{FP})/n$ is the predicted positive prevalence;

(i) Curve-based summaries integrated over classifier operating points: the area under the ROC curve (AUROC) and the area under the Precision-Recall curve (AUPRC) [16, 38, 51].

Metrics (a)–(h) strictly require, and are uniquely determined by, hard labels (explicit true/false) from the test cases; consequently, an explicit or implicit threshold τ must be established prior to evaluation. In numeric experiments, we report these metrics at the operating point given by each learner's default `predict` output on hard-label LOOCV predictions. Item (i) requires continuous out-of-fold testing classification scores and is discussed in Subsection 4.4.

The second consequence of class imbalance is that the training procedure must explicitly account for it. The standard mitigation techniques include:

- **Observation weighting:** multiplying the loss on positive instances by the inverse class frequency;
- **Cost-sensitive learning:** assigning explicit misclassification costs c_{FP} and c_{FN};
- **Operating-point tuning:** shifting the threshold τ away from its default value along the ROC curve;

- **Resampling:** ranging from simple random undersampling and oversampling to the synthetic minority oversampling technique (SMOTE) [5] and hybrid schemes like Tomek-link cleaning combined with random undersampling [11];
- **Algorithm-level methods:** ensembles that integrate imbalance handling directly into the training loop, such as RUSBoost [40], which interleaves random undersampling with boosting.

In this work, we employ RUSBoost as a representative imbalance-aware ensemble, which renders additional observation weighting unnecessary; inverse-class-frequency weights are retained for all other learners.

3.2. Asymmetric error costs

A missed case of depression (a false negative) and an unnecessary referral for clinical evaluation (a false positive) do not carry the same clinical weight. Performance on individual class labels, alongside the sensitivity–specificity trade-off, represent distinct aspects of classifier behavior. Under class imbalance, collapsing these independent dimensions into a single scalar metric forces strong implicit assumptions—such as treating error rates as equally critical (BalAcc or Youden's J), prioritizing the minority class (F_1, PPV), or applying a cost-sensitive anchoring to the operating point τ.

If the misclassification costs c_{FP} and c_{FN} are known (are set by a clinician or by insurance analysis), the Bayes-optimal operating threshold τ^* on a calibrated probability score is given by [51]:

$$\tau^* = \frac{c_{\mathrm{FP}}\, P(Y=0)}{c_{\mathrm{FP}}\, P(Y=0) + c_{\mathrm{FN}}\, P(Y=1)}. \tag{2}$$

The terms $P(Y=1)$ and $P(Y=0)$ represent the prior probabilities of a positive and a negative case in the population, respectively. Because (2) requires a calibrated score, which maps the raw classifier outputs onto a true probability space, the underlying score distribution must be known. To avoid optimistic bias and cherry-picking, it is essential to estimate this distribution strictly within the training data at each testing fold level rather than utilizing the resulting test partitions. On the other hand, if the operating point is established externally (independent of the empirical data), we can generate hard labels out-of-fold. Consequently, we can construct a pseudo-ROC curve from the out-of-fold testing scores and evaluate curve-based summaries at the test level.

The mechanism used to select this threshold determines how the metrics from Subsection 3.1 should be deployed for model comparison:

- **Fixed τ, or known c_{FP} and c_{FN}:** The threshold is known, or we have fixed misclassification costs where (2) can be applied at the fold level. Hard-label metrics (a)–(h) should be compared at this constant τ using

multi-dimensional combinations (such as reporting Sn and Sp together, or monitoring |Sn − Sp| and MCC), rather than relying on Acc alone.

- τ **set by convention:** When costs are unspecified, the threshold selection rule must be stated explicitly (e.g., using the equal-error point where FPR = FNR, or using each learner's default `predict` threshold) and applied consistently across all methods. This approach is standard when utilizing the MATLAB `predict` function and comparable machine learning toolboxes.
- **No fixed** τ**:** Classifiers should not be ranked using a single hard-label metric at an arbitrary threshold. Instead, their entire performance profiles should be evaluated via curve-based summaries (item (i)), such as AUROC or AUPRC. However, reporting the final deployed performance will still eventually require defining a specific τ.

Our clinical partners have not yet supplied explicit misclassification costs. Until they do, we treat false positives and false negatives as equally costly for *reporting*—standard in diagnostic screening—recently reviewed by Cullerne Bown [9], and use each learner's default `predict` rule rather than tuning τ^* on this cohort.

When c_{FP} and c_{FN} do become available, the optimal operating point can be computed via (2) at the fold level. By archiving the out-of-fold training score distributions for all cross-validation folds during the initial training phase, these thresholds can be established retrospectively.

4. Small datasets

Clinical machine-learning studies often rely on cohorts comprising only a few hundred patients. While small by the standards of contemporary machine learning, such sample sizes are typical of prospective, single-centre clinical studies. A small dataset offers clear practical advantages: it is cost-effective to assemble, straightforward to curate, transparent for clinical inspection, rapid to train on, and highly amenable to interpretable models. Furthermore, comprehensive data visualisation remains feasible, allowing preprocessing decisions to be audited individually.

The disadvantages are equally prominent: every performance estimator suffers from high variance, feature selection is inherently fragile, hyperparameter tuning is hazardous due to a noisy validation signal, and the coverage of the true patient population distribution is limited. In addition, small datasets are highly vulnerable to optimistic bias if the validation protocol exhibits even minor test result leakage [48]. When the target outcome is binarized at a stringent clinical cut-off, the positive class is frequently rare; consequently, the absolute count of

positive events (n_+) may be small even when the total cohort size (n) is moderate (see Subsection 4.1). These constraints motivate the concept of pooling across data sources as well as the design of out-of-sample performance estimation; we address pooling in Subsection 4.2 and validation in Subsection 4.4.

4.1. Events per variable

For multivariable binary prediction with many covariates, a tighter constraint than total cohort size n is often the number of *events* in the positive class, n_+ (the count that enters the denominator of sensitivity). A widely cited planning heuristic for logistic and related models recommends on the order of ten to twenty *events per variable* [33],

$$\mathrm{EPV} = \frac{n_+}{d},$$

where d is the number of predictors (or, conservatively, the number of fitted parameters); we reserve subscripted symbols such as p_+ and p_e for prevalences (Subsection 3.1). When EPV is low, estimates are unstable, differences between algorithms are difficult to separate from noise, and validation protocols with many tuned degrees of freedom are unreliable [48]. Strict cross-validation makes the constraint tighter still: each training split contains fewer than n_+ positives (e.g., at most $n_+ - 1$ under leave-one-out schemes), and encoding nominal covariates increases the effective input dimension beyond d, so the ratio relevant to a flexible learner can be lower than n_+/d suggests. Reporting EPV together with n and prevalence is therefore good practice in small clinical classification studies. We quantify n_+, d and EPV for the Cheng cohort in Section 5 and interpret the resulting bottleneck in Sections 6 and 7.

If a very small EPV occurs, it may motivate the application of a dimensionality reduction technique; however, whichever approach is chosen, it still adds at least one free parameter (often corresponding to the exact number of retained dimensions). While it is possible to bound the maximum number of parameters using the EPV heuristic at the order of ten to twenty, alternative regularisation techniques can seamlessly keep all available predictors in the game [17]. For logistic regression, it is standard to apply ridge regularisation, whereas a decision tree naturally regularises its structure by restricting the maximum tree depth.

4.2. Combining datasets from different sources

A frequent response in the discussed setting is to pool data across institutions in order to enlarge the training sample [34]. Pooling helps when the joint distribution $P(X, Y)$—where X are the clinical predictors and Y the target outcomes—is similar across sources, because it broadens the support of X and reduces variance. Conversely, pooling hurts when the sources are contradictory, i.e., when $P(Y \mid X)$ differs between them. When the sources are merely complementary,

i.e., when they cover non-overlapping regions of X, there is no fundamental barrier to integrating them into a more general model. However, doing so yields little to no performance benefit over simply using the two separate, smaller local models.

In any case, data integration requires caution because Simpson's paradox can occur, whereby a single global model performs worse than the individual local models [41]. To mitigate these risks, it is highly recommended to train separate models for each local dataset alongside a pooled global model, evaluating the performance of all models on both the individual and aggregated evaluation sets. Such strategies highlight the need for dedicated cross-model analysis tools, presenting a valuable avenue for future research.

Another issue is legislative. Patient-level data sharing across institutions, and especially across jurisdictions, is constrained by the General Data Protection Regulation in the European Union and by analogous regulations elsewhere.

These technical and legislative limitations naturally shift the focus away from raw data centralization toward alternative integration workflows. A pragmatic alternative is to share trained model parameters rather than raw data, which enables federated learning [34], distillation, and transfer-learning workflows while keeping the source data inside the institution of origin.

4.3. Handling missingness under sample constraints

On small clinical cohorts where sample preservation is paramount, discarding patient records due to isolated missing values is highly inappropriate. To prevent this loss of statistical power, missingness management strategies generally fall into five primary methodological approaches [12, 24]:

- **Constant Value or Indicator Imputation**: Replaces missing cells with a fixed constant (such as zero or an out-of-range value) and optionally creates a parallel binary flag column indicating that the data was missing. This treats missingness as an explicit feature, but it can introduce artificial step-functions into linear models or drastically inflate the feature count ($2d$) on small sample sizes.
- **Heuristic Summary Imputation (Mean/Median)**: Replaces missing entries with central tendency statistics. While mathematically straightforward, calculating these statistics globally across the entire cohort introduces severe risks of data leakage and optimistic performance overestimation.
- **Model-Based Imputation** (e.g., MICE): Uses iterative predictive models to estimate missing values based on the distribution of remaining features. Though highly flexible, these estimators can become statistically unstable when the underlying sample size is too small to reliably train the imputer itself.

- **Distance-Based Imputation** (e.g., k-NN Impute): Reconstructs missing entries by averaging values from the k most similar patient records in the feature space. While effective at preserving local feature relationships, distance metrics degrade rapidly in performance on small, high-dimensional datasets or when missingness occurs across variables used to calculate similarity.
- **Algorithm-Native Handling**: Some modern tree-based learners (such as XGBoost or LightGBM) do not require preprocessing at all. They use an approach called Sparsity-aware Split Finding, where missing values are automatically sent down whichever branch minimizes the training loss during a split. However, this is a property of the specific classifier and cannot be applied generally.

To enforce absolute isolation while maintaining maximum data retention, we focus on the per-fold algorithmic isolation strategy using median injection. There are several variants. A popular approach is *class-conditional per-fold median imputation*, which calculates separate medians for each target class using only that class's training data. While this actively preserves the distinct statistical profiles between positive and negative classes, it requires knowing the true label to impute a test case, thereby risking leakage of information of testing results. Moreover, if the positive minority class contains a very small number of cases, calculating a class-specific median introduces severe statistical instability.

Consequently, *unconditional per-fold median imputation* offers a more robust alternative. It calculates a single target-blind median using all available training observations in that fold. This approach avoids downstream mathematical bias and maximizes sample volume to provide a highly stable baseline. Although it shrinks variance toward the global training median and may dilute class-specific clinical differences—potentially degrading performance—it remains a conservative and safer choice, especially for small, imbalanced datasets where overfitting is the primary concern.

If the missingness of a specific feature (per class) exceeds a 25 % threshold, it is highly recommended to discard the entire column rather than attempting to reconstruct it from an unreliable remainder [15].

4.4. Validation and testing

Leave-one-out cross-validation. The dominant validation protocols in supervised machine learning: the train/validation/test split, K-fold cross-validation, and leave-one-out cross-validation (LOOCV), the latter being the $K = n$ limit of K-fold cross-validation [27].

When n is small, a hold-out test set is wasteful and its estimate is statistically noisy [17]; K-fold cross-validation with moderate K improves on that but still leaves out a non-trivial fraction of the data at each fit. LOOCV instead trains

on $n-1$ patients per fold, yields one out-of-sample prediction per patient, and is deterministic, which removes random partition noise.

These advantages come at a price: n model fits per learner, highly overlapping training folds, potentially high variance of the aggregate estimate, and no confidence intervals without an additional patient-level *bootstrap* on top of LOOCV [48], which we do not run here.

With imbalanced data there is a further cost, *stratification bias*: when the held-out patient is positive, the training fold contains one fewer positive than when the held-out patient is negative, so minority-class prevalence in training differs across folds; if $n^+ \approx 10$, that missing case is about 10 % of the minority class in that specific training fold [32]. Because models are not trained on the same class mix in every fold, LOOCV variability can be inflated when the positive class is rare.

Leave-pair-out cross-validation. Leave-pair-out cross-validation (LPOCV) is a combinatorial alternative that addresses this bias [31]: each fold withholds one positive patient i^+ and one negative patient i^-, trains on the remaining $n-2$ cases, and evaluates only that pair. Every training fold then has the same class counts $(n^+-1,\, n^--1)$, which eliminates the stratification bias that affects LOOCV when the positive class is rare. LPOCV is therefore a reasonable substitute for LOOCV on *small, strongly imbalanced* cohorts: the minority class is small enough that fold-to-fold training prevalence matters, yet n^+ is still small enough that the total number of fits, $n^+ \times n^-$, remains practically feasible. The scheme remains deterministic and uses almost all of the data; however, each minority patient appears in multiple test pairs, meaning that variability can remain high when n^+ is very small. Each fold compares one positive with one negative, which allows for an operating-point-free summary from pairwise rankings when a continuous score is available (equation (5) below).

Out-of-fold scores and AUROC. Confusion-matrix based metrics (a)–(h) in Subsection 3.1 depend only on the hard out-of-fold labels $\hat{y}_i \in \{0,1\}$ stored from each LOOCV or LPOCV fold.

If, in addition, a real-valued score $\hat{s}$ is stored for the test case in each LOOCV fold, we can also reconstruct a pseudo-ROC curve by integrating over operating points post hoc without refitting the models, enabling the theoretical evaluation of any curve-based metric. In a similar way, one could reconstruct the Precision-Recall Curve (PRC) and estimate the AUPRC from pooled out-of-fold scores. However, because precision depends directly on data prevalence, fluctuating minority-class proportions across training folds shift the baseline of the metric from fold to fold. Due to this, despite its popularity in imbalanced contexts, pooling out-of-fold scores to evaluate cross-validated AUPRC is highly unstable and strongly discouraged.

A more prominent way to calculate the AUROC than direct ROC reconstruction is via the Wilcoxon–Mann–Whitney statistic [16]. For LOOCV, this approach is mathematically equivalent to the result generated by the pseudo-ROC curve; in LPOCV, however, it differs fundamentally and provides a more rigorous formulation than relying on a pooled pseudo-ROC reconstruction. It is worth noting that unlike the AUROC, the AUPRC cannot be reduced to an unweighted, distribution-free pairwise rank statistic like the Wilcoxon–Mann–Whitney statistic, because precision dynamically penalizes ranking errors based on their specific location in the sorted list.

For LOOCV, patient i is scored exactly once, when i is the sole held-out case; denote that out-of-fold score by $\hat{s}_i$. Let $\mathcal{I}^+ = \{i : y_i = 1\}$ and $\mathcal{I}^- = \{j : y_j = 0\}$, with $n^+ = |\mathcal{I}^+|$ and $n^- = |\mathcal{I}^-|$. Define the pairwise win function

$$w(s^+, s^-) = \mathbf{1}[s^+ > s^-] + \tfrac{1}{2}\,\mathbf{1}[s^+ = s^-]. \tag{3}$$

Then

$$\widehat{\mathrm{AUROC}}_{\mathrm{LOOCV}} = \frac{1}{n^+ n^-} \sum_{i \in \mathcal{I}^+} \sum_{j \in \mathcal{I}^-} w\big(\hat{s}_i, \hat{s}_j\big). \tag{4}$$

The LOOCV estimation in equation (4) uses only the combination of single results derived for each out-of-fold test case.

For LPOCV, each fold withholds one positive $i^+ \in \mathcal{I}^+$ and one negative $i^- \in \mathcal{I}^-$, trains on the remaining $n - 2$ patients, and assigns fold-specific scores $\hat{s}_{i^+}^{(i^+,i^-)}$ and $\hat{s}_{i^-}^{(i^+,i^-)}$ to the two held-out cases (e.g., predicted probabilities for the positive class). The AUROC estimate is the mean pairwise win rate over all $n^+ n^-$ folds:

$$\widehat{\mathrm{AUROC}}_{\mathrm{LPOCV}} = \frac{1}{n^+ n^-} \sum_{i^+ \in \mathcal{I}^+} \sum_{i^- \in \mathcal{I}^-} w\Big(\hat{s}_{i^+}^{(i^+,i^-)}, \hat{s}_{i^-}^{(i^+,i^-)}\Big). \tag{5}$$

Crucially, the estimation for LPOCV de facto instantiates every necessary pairwise combination for the AUROC calculation as a separate, isolated model fold. Despite the fact that equations (4) and (5) use the same tie correction (3), they need not coincide numerically; this is because LOOCV assigns a single score per patient, whereas LPOCV scores each patient multiple times across different pairwise folds.

4.5. Protocol adopted in this study

On small clinical datasets, the risk of *overestimating* out-of-sample performance is often counterintuitively high [48]. This can happen through information leakage, repeated model comparison, or tuning on the same patients used for reporting; thus, obtaining credible results requires deliberately conservative choices rather than maximizing apparent performance. We adopt LOOCV for the results reported in Section 7, utilizing it strictly for evaluation rather than

as an inner validation loop for training. To avoid optimistic bias, we do not tune hyperparameters or optimize the number of predictors in an inner loop. Instead, we fix all hyperparameters to documented defaults (Subsection 6.2), retain all 29 non-redundant predictors on a common input space (Section 5), and fix all random seeds for replicability.

Furthermore, discarding entire patient records due to isolated missing values is highly inappropriate on small cohorts where sample preservation is paramount. Given the low prevalence of missingness in our data, we prevent sample attrition and preserve all available cases by reconstructing missing values via unconditional per-fold median imputation. As a consequence of prioritizing maximal data retention, we accept a very low EPV at stringent PHQ-9 cut-offs (Subsection 4.1) rather than reducing d via supervised feature selection on this cohort. To mitigate overfitting when retaining all predictors, we rely on structural algorithmic regularization embedded in the classifiers itself; specifically: ridge regularization for Logistic Regression, restrict splitting factor for Decision Trees and ensembling and boosting protection in RUSBoost.

The *primary* reported output is the hard out-of-fold label $\hat{y}_i \in \{0, 1\}$ from each learner's default `predict` on every LOOCV fold. To support post-training misclassification cost incorporation (Subsection 3.2) and post hoc AUROC calculation (Subsection 4.4), we archive both the hard test decisions at the default operating point and the continuous predicted scores across all training folds and test cases. Conversely, we deliberately avoid evaluating the AUPRC due to its cross-validated instability in imbalanced contexts (Subsection 4.4).

The protocol, as it is designed here, is further applied to the methodological testbed cohort (details in Section 5 and Section 6). The obtained results are briefly summarised in Section 7.

5. Data and target

The reference dataset is the cohort published by Cheng and co-authors in 2023 [7] and released as an open Figshare deposit [6]. It was assembled in a cross-sectional survey carried out at *three tertiary hospitals in Changsha, Hunan, China*, from November 2021 to February 2022, with 232, 200, and 148 patients enrolled at the three sites respectively, for a total of $n = 480$ CHF inpatients. Eligible patients were assessed 2–3 days after admission; written consent was obtained, and the PHQ-9 was administered through a face-to-face structured interview by trained interviewers, all master's-level nursing students.

The published inclusion and exclusion criteria define the support of the cohort and are relevant when discussing the transferability of any learned model (Section 8).

Inclusion criteria:

- confirmed chronic heart failure with New York Heart Association (NYHA) functional class II–IV;
- age ≥ 18 years;
- willingness to participate.

Exclusion criteria:

- a diagnosis of *acute* heart failure;
- a personal history of psychosis *or* current antidepressant therapy;
- active cancer or palliative-care stage;
- cognitive impairment severe enough to prevent the patient from completing the questionnaire independently or cooperatively.

The cohort therefore excludes both the mildest CHF cases (NYHA I) and patients in whom depression is already a clinically managed condition; it also excludes patients in whom CHF is dominated by end-stage cancer or by an acute decompensation. These boundaries are important: a screening model trained on this cohort is, strictly speaking, a model of *undiagnosed* depressive symptoms in NYHA II–IV chronic, non-acute CHF inpatients who can complete a structured questionnaire under interview conditions.

The PHQ-9 case mix used throughout this paper is recomputed directly from the total PHQ-9 score on the released file [6]. At the standard cut-off $\theta = 10$, we count $n_+ = 39$ positives (Table 1), and all numerical results in Section 7 are derived from this recomputation.

The cohort comprises 480 patients and 53 columns: 39 candidate feature columns recording demographic, clinical, and laboratory information, plus 14 PHQ-9 columns carrying the depression instrument itself (the nine PHQ-9 item scores, the total PHQ-9 score, and four derived PHQ-9 fields). The PHQ-9 columns are part of the target and are never used as classifier inputs.

From the 39 feature columns, we retain 29 primary sources and drop the remaining 10 as redundant (derived or ids). The exact list of the 29 retained predictors, together with their measurement scale, number of levels, and level dictionary, is given in Table 2.

By the standards of contemporary machine learning, this cohort is *small*: $n = 480$ patients across 29 classification predictors. The data are also strongly *imbalanced* once the PHQ-9 score is binarized at the clinically relevant cut-offs. Every patient has a non-missing total PHQ-9 score. On the integer scores actually observed in the cohort (0–23; none of the theoretical maximum 27), the mean is 3.029, the standard deviation is 3.975, and the median is 2. The distribution is strongly right-skewed: 163 patients (33.96 %) score zero, and counts decay rapidly toward higher scores. This concentration at low scores

explains why the positive prevalence at the conventional cut-off $\theta = 10$ is only $39/480 \approx 8.13$ % and under 3 % at $\theta = 15$ (Table 1).

TABLE 1. PHQ-9 case mix by cut-off θ on the full cohort ($n = 480$, no missing PHQ-9). For each $\theta \in \{1, \ldots, 15\}$, we list the number of positives $n_+ = |\{i : S_i \geq \theta\}|$, the number of negatives $n_- = |\{i : S_i < \theta\}|$, and the positive prevalence Prev. $= 100\, n_+/480$ (for S, see Definition 2.1). Counts are recomputed from the total PHQ-9 score on the released cohort.

θ	n_+ $(S \geq \theta)$	n_- $(S < \theta)$	Prev. (%)
1	317	163	66.042
2	255	225	53.125
3	188	292	39.167
4	145	335	30.208
5	109	371	22.708
6	89	391	18.542
7	69	411	14.375
8	59	421	12.292
9	49	431	10.208
10	39	441	8.125
11	29	451	6.042
12	25	455	5.208
13	19	461	3.958
14	15	465	3.125
15	14	466	2.917

The events-per-variable ratio (Subsection 4.1) is EPV $= n_+/d \approx 39/29 \approx 1.3$ at the default cut-off $\theta = 10$ and falls below 0.5 at $\theta = 15$. Thus, the practical bottleneck is the small number of screen-positive cases rather than the total sample size of $n = 480$. Small n, strong imbalance, and a low number of positives together fully justify deploying the protocol prepared in Subsection 4.5.

Among the 29 classification predictors, missing values occur in only three fields across the full cohort: NT-proBNP (column `BNP`; 19 missing, 3.96 %), left-ventricular ejection fraction (EF; 13 missing, 2.71 %), and family income bracket (1 missing, 0.21 %); all other retained predictors are complete. This situation is suitable for missing values reconstruction via median imputation (Subsection 4.3).

The typology of the 29 retained predictors is given in Table 2; their level dictionaries are taken from the demographic and medical-characteristics tables of Cheng et al. [7, Tab. 1, Tab. 2] where available, and cross-checked against the column metadata of the Figshare deposit [6]. Inside the classification pipeline,

continuous variables are z-score normalized on the training fold of each LOOCV iteration. Nominal variables are one-hot encoded into $L-1$ indicator columns by dropping one reference level. Ordinal variables are kept as integer indices on their natural order.

TABLE 2. Typology of the 29 retained predictors from the Cheng et al. [7] cohort, organized by measurement scale and listing, for each categorical predictor, the number of levels L and the level dictionary. The total PHQ-9 score is the target variable and is not listed. Continuous variables are z-score normalized inside each LOOCV fold; nominal predictor with L categories is represented by $L-1$ binary presence indicators; ordinal variables are kept on their integer index.

Predictor	L	Levels / units
Continuous (4)		
Age	—	years
Body mass index (BMI)	—	kg/m^2
NT-proBNP (column `BNP`, see Section 5)	—	pg/ml (same biomarker bands as in [7, Tab. 2])
Left-ventricular ejection fraction (EF)	—	percent
Ordinal (5)		
Education level	4	≤ 6 / 7–9 / 10–12 / >12 grades
Family income bracket	5	$\leq 10\,000$ / 10 000–30 000 / 30 000–50 000 / 50 000–100 000 / $>100\,000$ yuan per year
NYHA functional class	3	II / III / IV (class I excluded by the inclusion criteria)
Course-of-disease class (disease duration)	5	<1 / 1–3 / 3–5 / 5–10 / >10 years
Amount of medication	—	integer count: number of drugs the patient is currently taking
Nominal (20)		
Sex	2	female / male
Marital status	4	unmarried / married / divorced / widowed
Employment status	3	employed / retired / unemployed
Health insurance	2	no / yes
Place of residence	2	urban / rural
Living with family (cohabitant)	2	no / yes
Smoking, alcohol consumption, coronary heart disease, arrhythmia, hypertension, diabetes mellitus, hyperlipidemia, dilated cardiomyopathy, congenital heart disease, kidney disease, cerebrovascular disease, lung disease, liver disease	2	no / yes
Disease type (CHF aetiology)	≥ 2	categorical CHF aetiology code (level dictionary in the Figshare deposit [6])

6. Methods

6.1. Experimental environment

All experiments are carried out in MATLAB (R2023b)[1] with the *Statistics and Machine Learning Toolbox* and the *Parallel Computing Toolbox*. The Statistics and Machine Learning Toolbox provides a comprehensive set of standard supervised learners, exposes them through the `Classification*` class hierarchy and the interactive *Classification Learner* app, and supports observation weighting, prior specification, misclassification costs, and a number of resampling-based methods such as RUSBoost.[2]

LOOCV is by construction embarrassingly parallel: the n leave-one-out folds are independent and can be evaluated concurrently. We therefore wrap the outer LOOCV loop in a `parfor` block from the Parallel Computing Toolbox, with a parallel worker pool sized to the available logical CPU cores. The experiment used to obtain the numbers reported in Section 7 runs on a single workstation with an Intel Core i7 processor (6 physical cores, hyper-threading enabled, for 12 logical workers). All random number generators (RNGs) are initialized from a fixed seed at the start of each experiment so that the LOOCV scores and the reported metrics are bit-for-bit reproducible. Numerical values in all tables are quoted to three decimal places. An identical pipeline can be reproduced in Python using `scikit-learn` and `imbalanced-learn`, subject to minor cross-implementation differences.

6.2. Classifiers and default hyperparameters

For preprocessing, we use per-fold median imputation (Subsection 4.3) and the encoding rules summarized at the end of Section 5. The full methodology is described in detail in Subsection 4.5.

We restrict the main presentation to three representative methods: Logistic regression (LR), a Decision Tree (DT), and Random Under-Sampling Boosting (RUSBoost). To keep the protocol transparent and auditable, we use fixed library defaults and no inner-loop tuning. The random seed is fixed throughout.

The first of these (Logistic regression) is tied directly to the source publication. Cheng et al. model the association between depressive symptoms (PHQ-$9 \geq 10$) and the covariates using *binary logistic regression* with a backward likelihood-ratio elimination procedure on all variables that pass a liberal univariate screen [7]; we therefore include ridge-regularized logistic regression as a

[1]The MathWorks, Inc., Natick, MA, USA.

[2]Inside the app, one can import the data matrix and the target vector, choose a validation scheme (the app natively supports K-fold cross-validation; LOOCV is reached either by setting $K = n$ or by exporting the generated script and replacing the partition with `cvpartition(n,'Leaveout')`), and export the underlying script for further customization.

transparent machine-learning analogue of their inferential model, implemented via `fitclinear` rather than stepwise variable selection. Our LOOCV metrics in Section 7 should not be compared directly with the stepwise logistic regression tables in Cheng et al.: they use a reduced predictor set chosen on the full sample and report inferential association rather than out-of-sample sensitivity, specificity, and the other confusion-matrix measures defined in Subsection 3.1.

Decision trees can be interpreted as the base learner which, when ensembled via random undersampling [40], results in RUSBoost. We therefore include a *single* DT configured with the *same* weak-learner template (split budget, leaf size, and surrogate splits) as used in the RUSBoost ensemble. Any performance gap between the two models then isolates the specific effects of boosting and undersampling from any changes in underlying tree geometry.

RUSBoost, the third reported learner, was chosen because it is *designed for class imbalance*: each boosting round trains a weak tree on a random undersample of the majority class down to the minority count, and an AdaBoost-style reweighting then concentrates subsequent rounds on the hardest examples [40]. The Cheng et al. cohort is highly imbalanced at every clinically relevant cut-off $\theta \geq 10$ (Table 1), which is precisely the regime for which RUSBoost is intended. Because the internal undersampling step already rebalances the training subsample at every round, adding inverse-class-frequency observation weights on top of RUSBoost would correct the imbalance twice—once by resampling and once by reweighting—and overshoot toward the positive class. We deliberately omit observation weighting for RUSBoost while maintaining inverse-class-frequency weights for LR and DT, which do not handle imbalance internally; this is a strict methodological choice, not an oversight. The fold-seed schedule and the training-fold-only median imputation of missing values remain identical across all incorporated classifiers.

- **Logistic regression** (LR): A linear logistic model in the generalized linear model framework [44], fitted with MATLAB's `fitclinear` using the `Learner='logistic'` option and *ridge* (L_2) regularization with a penalty strength of $\lambda = 10^{-4}$. This regularization limits coefficient instability when the events-per-variable ratio is low (Subsection 4.1) while retaining all 29 predictors for direct comparability with DT and RUSBoost. Observation weights are set to inverse class frequency: $w_y = 1/P(Y = y)$.
- **Decision tree** (DT): Fitted via MATLAB's `fitctree` [44], which constructs a binary decision tree by greedy recursive partitioning under Gini impurity (`SplitCriterion='gdi'`). The hyperparameters are set to `MaxNumSplits` $= 10$, `MinLeafSize` $= 10$, and `Surrogate` $=$ `'off'`, with all predictors available at every split and no pruning override. Inverse-class-frequency observation weights are applied using the same rule as for LR.

These hyperparameters intentionally match the weak-learner template of the RUSBoost ensemble below.

- **RUSBoost** [40]: An ensemble of $T = 50$ weak learners fitted with MATLAB's `fitcensemble` [44] using the `Method='RUSBoost'` option. Each weak learner is a decision tree built with the exact hyperparameter template as the single DT above. The learning rate is fixed at 0.1, and the majority class is randomly undersampled to match the minority count at each boosting round. Because this internal undersampling step already rebalances the training subsample at every boosting round, additional inverse-class-frequency observation weighting is omitted to avoid the double-correction issue discussed in the previous section.

All three learners share a single companion script, `CLASSIFIERS.m` (Appendix A). This script executes the entire pipeline sequentially: the Cheng Figshare cohort [6] is loaded into the workspace matrix `DATA`; the 29 retained predictors and the total PHQ-9 score are selected as described in Section 5; missing values are handled according to Subsection 4.3; and a strict LOOCV routine is run over all cut-offs $\theta \in \{1, \ldots, 15\}$ with a fixed master seed of $\sigma_0 = 42$. The active model configuration is controlled via the `LEARNER` flag specified at the top of the file.

7. Results

All numbers in the main tables below come from LOOCV with a master random seed of $\sigma_0 = 42$ (Subsection 4.5): confusion-matrix metrics (a)–(h) from hard out-of-fold labels at the default operating point implied by `predict`, and $\widehat{\mathrm{AUROC}}_{\mathrm{LOOCV}}$ (equation (4)). We call an operating point *anti-predictive* when Youden's J is negative. Table 3 compares the three reported learners at the clinical cut-off $\theta = 10$; Tables 4, 5, and 6 list the full threshold sweep for LR, DT, and RUSBoost, respectively (columns defined in Table 4). Table 7 at the end of this section contrasts LOOCV with LPOCV for RUSBoost at high cut-offs.

At $\theta = 10$, ridge logistic regression with inverse-class-frequency weighting reaches Sn = 0.462, Sp = 0.683, Acc = 0.665, $J = 0.144$, and MCC = 0.084: the linear boundary under-predicts positives despite weighting. The weighted single decision tree (DT) occupies an intermediate position (Sn = 0.513, Sp = 0.580, Acc = 0.575, $|\mathrm{Sn} - \mathrm{Sp}| = 0.067$, $J = 0.093$, MCC = 0.052). RUSBoost improves sensitivity and specificity balance (Sn = 0.641, Sp = 0.619, $|\mathrm{Sn} - \mathrm{Sp}| = 0.022$) and reaches the largest values of Youden's J and MCC among the three ($J = 0.260$, MCC = 0.145); it is the only learner that handles class imbalance *algorithmically* via internal undersampling.

TABLE 3. LOOCV metrics at the standard PHQ-9 cut-off $\theta = 10$, fixed master seed $\sigma_0 = 42$. Column headings are abbreviated as in the threshold-sweep tables below (Tables 4, 5, and 6): Sn = TPR = TP/(TP + FN) (sensitivity, true positive rate), Sp = TNR = TN/(TN + FP) (specificity, true negative rate), Acc = (TP + TN)/n (overall accuracy), and |Sn − Sp| (balance gap).

Classifier	Sn	Sp	Acc	|Sn − Sp|
ridge Logistic Regression (weighted)	0.462	0.683	0.665	0.221
Decision Tree (weighted)	0.513	0.580	0.575	0.067
RUSBoost (unweighted)	0.641	0.619	0.621	0.022

TABLE 4. Threshold sweep for ridge-regularized logistic regression (LR, MATLAB `fitclinear` [44]) with inverse-class-frequency weighting under strict LOOCV with a fixed master seed $\sigma_0 = 42$: PHQ-9 cut-off θ; sensitivity Sn = TPR and specificity Sp = TNR; positive and negative predictive values PPV, NPV; accuracy Acc and balanced accuracy BalAcc = $\frac{1}{2}$(TPR + TNR); MCC; Cohen's kappa; F_1; Youden's J; the four confusion-matrix counts TP, TN, FP, FN; and $\widehat{\mathrm{AUROC}}_{\mathrm{LOOCV}}$ (item (i)), equation (4)) from archived out-of-fold positive-class scores.

θ	TPR	TNR	PPV	NPV	Acc	BalAcc	MCC	κ	F_1	J	TP	TN	FP	FN	AUROC
1	0.533	0.577	0.710	0.388	0.548	0.555	0.104	0.098	0.609	0.110	169	94	69	148	0.581
2	0.522	0.636	0.619	0.540	0.575	0.579	0.158	0.155	0.566	0.157	133	143	82	122	0.603
3	0.505	0.654	0.485	0.673	0.596	0.580	0.158	0.158	0.495	0.159	95	191	101	93	0.627
4	0.476	0.639	0.363	0.738	0.590	0.557	0.108	0.105	0.412	0.115	69	214	121	76	0.586
5	0.404	0.642	0.249	0.785	0.588	0.523	0.039	0.037	0.308	0.045	44	238	133	65	0.527
6	0.315	0.609	0.155	0.796	0.554	0.462	−0.062	−0.055	0.207	−0.077	28	238	153	61	0.497
7	0.449	0.623	0.167	0.871	0.598	0.536	0.052	0.042	0.243	0.072	31	256	155	38	0.552
8	0.525	0.639	0.169	0.906	0.625	0.582	0.111	0.086	0.256	0.164	31	269	152	28	0.569
9	0.429	0.654	0.124	0.910	0.631	0.541	0.052	0.040	0.192	0.083	21	282	149	28	0.568
10	0.462	0.683	0.114	0.935	0.665	0.572	0.084	0.060	0.183	0.144	18	301	140	21	0.603
11	0.414	0.696	0.081	0.949	0.679	0.555	0.057	0.037	0.135	0.110	12	314	137	17	0.616
12	0.400	0.690	0.066	0.954	0.675	0.545	0.043	0.027	0.114	0.090	10	314	141	15	0.569
13	0.263	0.727	0.038	0.960	0.708	0.495	−0.004	−0.003	0.067	−0.010	5	335	126	14	0.552
14	0.267	0.733	0.031	0.969	0.719	0.500	0.000	0.000	0.056	0.000	4	341	124	11	0.578
15	0.357	0.723	0.037	0.974	0.713	0.540	0.030	0.016	0.068	0.080	5	337	129	9	0.436

For LR (Table 4) J and MCC stay mildly positive at $\theta \in \{2, 3, 4, 8\}$ but are negative or near zero at several high cut-offs (e.g., $\theta = 6$ and $\theta = 13$), with mixed values elsewhere; this pattern is partly explained by increasing EPV as θ decreases. At $\theta = 14$, the classifier is entirely prevalence-driven ($J = 0$). With only 39 positives at $\theta = 10$ and 29 predictors, a ridge boundary cannot

stabilize at the highest cut-offs even with class weighting. Other tested off-the-shelf learners (k-NN, kernel Naive Bayes, a quadratic-kernel SVM, etc.) showed similarly limited out-of-fold behavior on this cohort and are not tabulated here.

TABLE 5. Threshold sweep for weighted single DT (MaxNumSplits = 10, MinLeafSize = 10; $\sigma_0 = 42$). Columns are structured as in Table 4.

θ	TPR	TNR	PPV	NPV	Acc	BalAcc	MCC	κ	F_1	J	TP	TN	FP	FN	AUROC
1	0.394	0.632	0.676	0.349	0.475	0.513	0.026	0.022	0.498	0.026	125	103	60	192	0.541
2	0.510	0.502	0.537	0.475	0.506	0.506	0.012	0.012	0.523	0.012	130	113	112	125	0.434
3	0.660	0.449	0.435	0.672	0.531	0.554	0.108	0.099	0.524	0.108	124	131	161	64	0.537
4	0.317	0.549	0.234	0.650	0.479	0.433	−0.125	−0.121	0.269	−0.134	46	184	151	99	0.465
5	0.404	0.679	0.270	0.795	0.617	0.541	0.073	0.071	0.324	0.083	44	252	119	65	0.540
6	0.348	0.665	0.191	0.818	0.606	0.507	0.011	0.010	0.247	0.013	31	260	131	58	0.469
7	0.333	0.708	0.161	0.864	0.654	0.521	0.032	0.029	0.217	0.041	23	291	120	46	0.465
8	0.424	0.608	0.132	0.883	0.585	0.516	0.021	0.016	0.201	0.032	25	256	165	34	0.497
9	0.286	0.594	0.074	0.880	0.563	0.440	−0.075	−0.053	0.118	−0.120	14	256	175	35	0.452
10	0.513	0.580	0.098	0.931	0.575	0.547	0.052	0.032	0.164	0.093	20	256	185	19	0.399
11	0.276	0.721	0.060	0.939	0.694	0.498	−0.002	−0.001	0.098	−0.004	8	325	126	21	0.500
12	0.280	0.818	0.078	0.954	0.790	0.549	0.056	0.044	0.122	0.098	7	372	83	18	0.363
13	0.158	0.809	0.033	0.959	0.783	0.484	−0.016	−0.012	0.055	−0.033	3	373	88	16	0.513
14	0.267	0.830	0.048	0.972	0.813	0.548	0.045	0.030	0.082	0.097	4	386	79	11	0.540
15	0.286	0.824	0.047	0.975	0.808	0.555	0.048	0.031	0.080	0.110	4	384	82	10	0.543

TABLE 6. Threshold sweep for RUSBoost ($T = 50$ weak trees; $\sigma_0 = 42$). Columns are structured as in Table 4.

θ	TPR	TNR	PPV	NPV	Acc	BalAcc	MCC	κ	F_1	J	TP	TN	FP	FN	AUROC
1	0.571	0.577	0.724	0.409	0.573	0.574	0.140	0.134	0.638	0.148	181	94	69	136	0.604
2	0.549	0.502	0.556	0.496	0.527	0.526	0.051	0.051	0.552	0.051	140	113	112	115	0.545
3	0.606	0.572	0.477	0.693	0.585	0.589	0.174	0.170	0.534	0.178	114	167	125	74	0.617
4	0.552	0.561	0.352	0.743	0.558	0.556	0.104	0.097	0.430	0.113	80	188	147	65	0.576
5	0.550	0.585	0.280	0.816	0.577	0.568	0.114	0.101	0.372	0.135	60	217	154	49	0.589
6	0.517	0.604	0.229	0.846	0.588	0.560	0.095	0.081	0.317	0.120	46	236	155	43	0.606
7	0.551	0.620	0.196	0.892	0.610	0.586	0.122	0.098	0.289	0.171	38	255	156	31	0.554
8	0.593	0.561	0.159	0.908	0.565	0.577	0.101	0.071	0.251	0.154	35	236	185	24	0.561
9	0.592	0.596	0.143	0.928	0.596	0.594	0.115	0.079	0.230	0.188	29	257	174	20	0.534
10	0.641	0.619	0.130	0.951	0.621	0.630	0.145	0.093	0.216	0.260	25	273	168	14	0.585
11	0.655	0.678	0.116	0.968	0.677	0.667	0.168	0.105	0.197	0.334	19	306	145	10	0.646
12	0.560	0.701	0.093	0.967	0.694	0.631	0.125	0.078	0.160	0.261	14	319	136	11	0.623
13	0.632	0.712	0.083	0.979	0.708	0.672	0.146	0.082	0.146	0.343	12	328	133	7	0.681
14	0.600	0.735	0.068	0.983	0.731	0.668	0.131	0.070	0.122	0.335	9	342	123	6	0.704
15	0.643	0.685	0.058	0.985	0.683	0.664	0.118	0.055	0.106	0.327	9	319	147	5	0.679

RUSBoost (Table 6) maintains both Sn and Sp mostly between 0.50 and 0.74 across the range $\theta \in [3, 15]$, with Youden's J up to 0.343 at $\theta = 13$. The strongest

Sn–Sp balance at a clinically labelled cut-off is $\theta = 11$ (Sn $= 0.655$, Sp $= 0.678$, $J = 0.334$), with the standard cut-off $\theta = 10$ close behind ($J = 0.260$); peak MCC in the sweep is 0.174 at $\theta = 3$. Across all three baselines, RUSBoost leads on Sn and J for every cut-off $\theta \geq 5$ and on MCC at all $\theta \geq 5$ except $\theta = 8$ (where weighted LR is slightly higher), gaining roughly 0.10 to 0.20 in J over the single decision tree at clinically relevant cut-offs.

All result tables can be reproduced using Appendix A by using `LEARNER` flag toggled to `'lr'`, `'dt'` or `'rusboost'`.

Finally, for RUSBoost at the highest cut-offs $\theta \in \{13, 14, 15\}$, we contrast LOOCV with LPOCV (Subsection 4.4) and report hard-label metrics and ranking-based AUROC under both validation schemes (Table 7). LPOCV completely eliminates the fold-to-fold variation in training class prevalence that structurally affects LOOCV whenever the held-out sample patient happens to be positive. However, it does *not* eliminate the fundamental limitation imposed by a low n_+ count: the specific positive case under evaluation is never available to the training set under either strict cross-validation routine, meaning sensitivity estimates remain noisy when n_+ drops into the tens. In the present run, LOOCV and LPOCV yield identical Sn entries at $\theta \in \{13, 14, 15\}$ while Sp adjusts slightly, indicating that the baseline sensitivity–specificity balance gap is not mechanically solved by switching validation schemes. Instead, LPOCV primarily stabilizes the underlying ranking comparison between one positive and one negative sample per fold (as reflected by the higher AUROC values, which do not depend on selecting a single operating point on the ROC curve). Conversely, the hard-label metrics (a)–(h) remain heavily dominated by the scarcity of positives and the default configuration mapping of the `predict` function.

TABLE 7. Comparison of LOOCV and LPOCV for RUSBoost at high cut-offs $\theta \in \{13, 14, 15\}$ (master seed $\sigma_0 = 42$, three-decimal rounding). LOOCV rows are computed from one out-of-fold hard prediction and score per patient; LPOCV rows use pairwise folds (one positive and one negative held out) and report patient-level hard labels by majority vote over all test appearances, while AUROC is computed as the mean pairwise win-rate of the positive score over the negative score across all $n_+ n_-$ held-out pairs.

θ	Scheme	Sn (TPR)	Sp (TNR)	Acc	BalAcc	MCC	F_1	AUROC
13	LOOCV	0.632	0.712	0.708	0.672	0.146	0.146	0.681
	LPOCV	0.632	0.720	0.717	0.676	0.151	0.086	0.724
14	LOOCV	0.600	0.735	0.731	0.668	0.131	0.122	0.704
	LPOCV	0.600	0.720	0.717	0.660	0.123	0.064	0.744
15	LOOCV	0.643	0.685	0.683	0.664	0.118	0.106	0.679
	LPOCV	0.643	0.715	0.713	0.679	0.132	0.066	0.724

8. Transferability to the Central European context

The Cheng et al. cohort was recruited from three tertiary hospitals in Changsha, Hunan, China [7], representing a single urban region rather than a nationally representative sample. Three primary concerns limit the direct transfer of a model trained on these data to Central European clinical practice. First, the cultural framing of mental health differs, and the somatic expression of depression is reported to be more frequent in East Asian populations [37], which directly interacts with the somatic items of the PHQ-9 noted in Section 2. Second, the baseline prevalence of clinically significant depression in CHF and the underlying comorbidity mix (e.g., diabetes, ischemic vs. non-ischemic etiology, and age structure) differ; indeed, nursing-led studies in Slovak and Czech CHF populations suggest meaningfully different quality-of-life and symptom profiles [3]. Third, clinical guidelines and the resulting pharmacotherapy mix differ: Central European cardiology follows the European Society of Cardiology (ESC) heart-failure guidelines [29] and the ESC clinical consensus on worsening chronic heart failure [30], both of which have evolved through several recent updates. We therefore treat the Cheng et al. data primarily as a methodological testbed; actual clinical deployment in a Central European context would require local re-validation against a Central European cohort, such as one prepared for data collection under the DIGIMENT framework.

If a Slovak (Central European) CHF cohort is assembled under DIGIMENT, it would represent a distinct target domain. Simply pooling it row-wise with the Cheng cohort is not automatically beneficial (see also Subsection 4.2): it helps only if the conditional relationship $P(Y \mid X)$ is sufficiently similar across regions, and it can actively degrade performance under severe domain shift. A prudent workflow is therefore to train and evaluate (i) a Slovak-only model, (ii) a Cheng-trained model tested on Slovak data, and (iii) a pooled or adapted model (e.g., using domain recalibration), reporting predictive performance separately by data source.

Instrument choice for DIGIMENT is open and does not change the value of this paper's protocol. Three practical options are:

- **Keep PHQ-9** (and the same cut-offs θ) if the goal is direct comparison with the Cheng cohort;
- **Switch to HADS-D** (Section 2) if the goal is a cleaner depression signal with less overlap between depressive and CHF somatic symptoms;
- **Use a restricted PHQ-9 label** based on cognitive or less CHF-overlapping items (Remark 3) as an intermediate step, then reassess learners such as RUSBoost.

In each case the same LOOCV protocol, metrics, and imbalance-aware learners apply after the binary target is redefined. The contribution of this paper is that evaluation framework, not a fixed choice of questionnaire.

9. Conclusion

We have elaborated a methodology for binary classification on small, strongly imbalanced datasets typical of single-centre clinical cohort studies, and illustrated it with three standard classifiers—ridge-regularized logistic regression (LR), a decision tree (DT), and RUSBoost—on the Cheng et al. 2023 cohort [7] of 480 Chinese CHF inpatients. Evaluation relies on strict leave-one-out cross-validation and the fixed library-default hyperparameters listed in Subsection 6.2.

RUSBoost emerges as the strongest learner in Section 7: at $\theta = 10$ it reaches $\mathrm{Sn} = 0.641$, $\mathrm{Sp} = 0.619$, and $|\mathrm{Sn} - \mathrm{Sp}| = 0.022$, and at $\theta = 11$ its performance improves to $\mathrm{Sn} = 0.655$, $\mathrm{Sp} = 0.678$, and $J = 0.334$. The weighted baselines (LR, DT) trail on imbalance-aware metrics and behave anti-predictively at one or more cut-offs. Algorithmic undersampling inside RUSBoost therefore systematically outperforms observation weighting alone in the logistic and single-tree models tested here (Remark 5).

Even so, no learner exceeds 75 percent on both sensitivity (Sn) and specificity (Sp) anywhere in the threshold sweep. We record several contributing factors as separate remarks so that they can be cited individually in follow-up investigations.

Remark 1. The analyzed cohort is shaped by several acquisition filters that jointly remove or under-sample more severe depression cases (Section 5). *Eligibility exclusions* omit patients with a personal history of psychosis *or* current antidepressant therapy [7], removing many cases in which affective illness is already diagnosed or managed. This is precisely the group that often carries the strongest statistical signal for current symptoms. While that rule is appropriate for the source paper's clinical question because it limits PHQ-9 confounding, it attenuates what routine machine-learning predictors can learn. *Participation selection* operates similarly: enrollment required written consent and agreement to a research interview. The published file contains only the 480 patients who completed enrollment; how many eligible inpatients were approached or subsequently declined is not reported. Patients with more severe symptoms may be less willing or less able to participate, meaning the most severely affected cases may never appear in the data.

Remark 2. The absolute number of positive cases at the standard cut-off $\theta = 10$ is only 39 (Table 1). Even an ideal classifier would exhibit a confidence interval

on sensitivity of roughly ±15 percentage points at this sample size. With $d = 29$ retained predictors, the events-per-variable ratio EPV $= n_+/d$ is only ≈ 1.3 at $\theta = 10$ and falls below 0.5 at $\theta = 15$. The standard planning range of EPV $\in [10, 20]$ [33] would require $n_+ \in [290, 580]$, a threshold this cohort never approaches at clinical cut-offs $\theta \geq 10$ (it is met only under the liberal relabeling at $\theta = 1$, where $n_+ = 317$).

Remark 3. Items 3, 4, 5, and 7 of the PHQ-9 instrument are somatic in nature. In a CHF population, these items are frequently inflated by the underlying cardiovascular disease itself, irrespective of baseline mood state [2]. The target variable is therefore inherently noisy with respect to the latent "clinically meaningful depression" a physician would care about, forcing any classifier to implicitly model both the latent psychiatric state and this disease-induced score inflation. The Figshare deposit [6] additionally provides derived somatic and cognitive subscale scores; modeling these dimensions separately, rather than relying on the total score alone, represents a plausible follow-up pathway for DIGIMENT.

Remark 4. The 29 primary predictors are heavily dominated by cardiac, demographic, and biochemical variables. Psychosocial predictors that are clinically known to be highly informative (e.g., social support networks, prior depressive episodes, recent major life events, and perceived internal control) are entirely absent from the cohort. A meaningful improvement in out-of-fold performance is unlikely without expanding the feature space to include these dimensions.

Remark 5. A single shallow decision tree is a high-variance, piecewise-constant classifier: a single sub-optimal split near the root can dominate predictions across a small cohort. RUSBoost aggregates $T = 50$ such trees in an AdaBoost-style optimization loop with random undersampling of the majority class at each iteration [40], enabling subsequent trees to focus on cases that remain difficult for the accumulated ensemble while class balance is repeatedly refreshed. It is therefore expected *a priori* that a single DT is weaker than an ensemble built from identical weak-learner templates; Tables 5 and 6 quantify that specific gap on this dataset without confounding it with deeper trees or alternative split criteria.

Remarks 1–4 point to cohort acquisition constraints, the small absolute number of screen-positive cases, noisy PHQ-9 labeling in CHF, and limited predictors as the primary upper bounds on achievable Sn and Sp, rather than the specific choice among the three learners (Remark 5 concerns only the DT versus RUSBoost comparison). The methodological lessons for designing any follow-up cohort are therefore:

- **Include treated patients:** Admit patients on concurrent antidepressant therapy and use the medication itself as an explicit predictive feature; the

supervised problem then becomes modeling "current depressive burden" rather than "untreated depressive burden".

- **Reduce consent bias:** Embed screening routines directly into standard clinical admission processes so that severely affected patients are less likely to systematically self-select out.
- **Expand the feature space:** Incorporate key psychosocial predictors (social support, prior history, perceived control) and leverage the independent PHQ-9 somatic and cognitive subscales noted in Remark 3.
- **Enlarge and diversify the cohort:** Recruit patients across multiple clinical centers and geographic regions to increase both n and n_+, broaden the support of feature space X, and validate model transferability from the Chinese cohort to a Central European setting, as outlined in Section 8.

In summary, these cohort and labeling limitations likely cap achievable performance on these data at least as much as the choice of classifier family itself.

The evaluation framework developed in Sections 3 and 4 is largely cohort-agnostic and is directly applicable to other similar datasets from restricted clinical studies. For reproducibility, the peak RUSBoost results documented above are fully recoverable using the exact parameterization shipped in Appendix A (`CLASSIFIERS.m`).

Concrete next steps for this research line are:

- local validation on a Central European cohort that incorporates these revised inclusion criteria;
- elicitation of explicit false-positive and false-negative costs from clinical partners to adjust the operational decision threshold via equation (2);
- expansion of the feature space toward patient-reported outcomes, following recent patient-reported outcome modeling methodologies [42];
- given that the questionnaire score is ordinal and carries more information than a single binary label at the standard diagnostic threshold—higher granularity and the possibility of tracking score evolution over time—it is natural to explore how to retain that information while keeping the robust binary-classification methodology designed in this paper, for example via ordinal classification by binary decomposition [4, 13, 14, 21, 23];
- extending the training and evaluation methodology with a post-training analysis of salient features, to clarify what construct the questionnaire-derived label and the resulting predictor actually capture; and
- testing the transferability of learned models across jurisdictions *without pooling patient-level records*—for instance, training models locally and exchanging or adapting model parameters only, as permitted under the data governance constraints discussed in Subsection 4.2.

Conflict of Interest. The authors declare that they have no conflict of interest.

REFERENCES

[1] ABU SUMAQA, Y.—HAYAJNEH, F. A.: *Depression and anxiety in patients with heart failure: contributing factors, consequences and coping mechanisms: A review of the literature*, Working with Older People **26** (2022), 174–186. DOI: https://doi.org/10.1108/WWOP-12-2021-0061.

[2] BHATT, K. N.—KALOGEROPOULOS, A. P.—DUNBAR, S. B.—BUTLER, J.—GEORGIOPOULOU, V. V.: *Depression in heart failure: Can PHQ-9 help?*, Int. J. Cardiol. **221** (2016), 246–50. DOI: https://doi.org/10.1016/j.ijcard.2016.07.057.

[3] BOBČÍKOVÁ, K.—BUŽGOVÁ, R.: *Chronic heart failure, depression, and its links to selected aspects: a cross-sectional study*, Kontakt **24** (2022), 286–293. DOI: https://doi.org/10.32725/kont.2022.037.

[4] CARDOSO, J. S.—PINTO DA COSTA, J. F.: *Learning to classify ordinal data: The data replication method*, J. Mach. Learn. Res. **8** (2007), 1393–1429. https://jmlr.org/papers/volume8/cardoso07a/cardoso07a.pdf.

[5] CHAWLA, N. V.—BOWYER, K. W.—HALL, L. O.—KEGELMEYER, W. P.: *SMOTE: Synthetic minority over-sampling technique*, J. Artif. Intell. Res. **16** (2002), 321–357. DOI: https://doi.org/10.1613/jair.953.

[6] CHENG, L.: *Depressive symptoms of chronic heart failure*. Figshare dataset, November 2022. DOI: https://doi.org/10.6084/m9.figshare.21557889.v1.

[7] CHENG, L.—ZHONG, Z.—DING, S.—DUAN, Y.—SUN, N.—ZHENG, F.: *Low body mass index and disease duration as factors associated with depressive symptoms of Chinese inpatients with chronic heart failure*, J. Health Psychol. **28** (2023), 1227–1237. DOI: https://doi.org/10.1177/13591053231173583.

[8] COHEN, J.: *A coefficient of agreement for nominal scales*, Educ. Psychol. Meas. **20** (1960), 37–46. DOI: https://doi.org/10.1177/001316446002000104.

[9] CULLERNE BOWN, W.: *Sensitivity and specificity versus precision and recall, and related dilemmas*, J. Classif. **41** (2024), 402–426. DOI: https://doi.org/10.1007/s00357-024-09478-y.

[10] DUAN, J.—HUANG, K.—ZHANG, X.—WANG, R.—CHEN, Z.—WU, Z.—HUANG, C.—YANG, C.—YANG, L.: *Role of depressive symptoms in the prognosis of heart failure and its potential clinical predictors*, ESC Heart Failure **9** (2022), 2676–2685. DOI: https://doi.org/10.1002/ehf2.13993.

[11] ELHASSAN, A. T.—ALJOURF, M.—AL-MOHANNA, F.—SHOUKRI, M.: *Classification of imbalance data using Tomek Link (T-Link) combined with Random Undersampling (RUS) as a data reduction method*, Glob. J. Technol. Optim. **1** (2017), 1–11. DOI: https://doi.org/10.4172/2229-8711.S1:111.

[12] EMMANUEL, T.—MAUPONG, T.—MPOELENG, D.—SEMONG, T.—MPHAGO, B.—TABONA, O.: *A survey on missing data in machine learning*, J. Big Data **8** (2021), 140. DOI: https://doi.org/10.1186/s40537-021-00516-9.

[13] FRANK, E.—HALL, M.: *A Simple Approach to Ordinal Classification*. In: *Machine Learning: ECML 2001* (L. De Raedt, P. Flach, eds.), pp. 145–156, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. DOI: https://doi.org/10.1007/3-540-44795-4_13.

[14] FRANK, E.—KRAMER, S.: *Ensembles of nested dichotomies for multi-class problems*. In: *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, p. 39, New York, NY, USA, 2004. Association for Computing Machinery. DOI: https://doi.org/10.1145/1015330.1015363.

[15] HAIR, J. F., JR.—HAIR, J. F.: *Multivariate Data Analysis*. Always Learning. Prentice Hall, 7th edition, 2010. Digitized Aug 9, 2011, Original from Cornell University.

[16] HANLEY, J. A.—MCNEIL, B. J.: *The meaning and use of the area under a receiver operating characteristic (ROC) curve*, Radiology **143** (1982), 29–36. DOI: https://doi.org/10.1148/radiology.143.1.7063747.

[17] HARRELL, F. E., JR.: *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. Springer Series in Statistics. Springer Cham, 2 edition, 2015. DOI: https://doi.org/10.1007/978-3-319-19425-7.

[18] HAVRANEK, E. P.—SPERTUS, J. A.—MASOUDI, F. A.—JONES, P. G.—RUMSFELD, J. S.: *Predictors of the onset of depressive symptoms in patients with heart failure*, J. Am. Coll. Cardiol. **44** (2004), 2333–2338. DOI: https://doi.org/10.1016/j.jacc.2004.09.034.

[19] HAWORTH, J. E.—MONIZ-COOK, E.—CLARK, A. L.—WANG, M.—WADDINGTON, R.—CLELAND, J.: *Prevalence and predictors of anxiety and depression in a sample of chronic heart failure patients with left ventricular systolic dysfunction*, Eur. J. Heart Fail. **7** (2005), 803–808. DOI: https://doi.org/10.1016/j.ejheart.2005.03.001.

[20] JIANG, H.—HU, R.—WANG, Y.-J.—XIE, X.: *Predicting depression in patients with heart failure based on a stacking model.*, World J. Clin. Cases **12** (2024), 4661–4672. DOI: https://doi.org/10.12998/wjcc.v12.i21.4661.

[21] KIM, H.-C.—GHAHRAMANI, Z.: *Bayesian Classifier Combination*. In: *International Conference on Artificial Intelligence and Statistics*, 2012.

[22] KROENKE, K.—SPITZER, R. L.—WILLIAMS, J. B.: *The PHQ-9: Validity of a brief depression severity measure*, J. Gen. Intern. Med. **16** (2001), 606–613. DOI: https://doi.org/10.1046/j.1525-1497.2001.016009606.x.

[23] LI, L.—LIN, H.-T.: *Ordinal Regression by Extended Binary Classification*. In: *Advances in Neural Information Processing Systems* (B. Schölkopf, J. Platt, T. Hoffman, eds.), vol. 19. MIT Press, 2006. DOI: https://dl.acm.org/doi/10.5555/2976456.2976565.

[24] LITTLE, R. J. A.—RUBIN, D. B.: *Statistical Analysis with Missing Data*, vol. 337 of Wiley Series in Probability and Statistics. John Wiley & Sons, 3rd edition, 2019. DOI: https://doi.org/10.1002/9781119013563.

[25] LIU, D.—CHEN, Z.—MARRERO, W. J.—THESEN, T.: *Machine learning-based prediction of depression severity among medical students*. In: *medRxiv*, 2023. DOI: https://doi.org/10.1101/2023.12.14.23299975.

[26] LOSSNITZER, N.—HERZOG, W.—STÖRK, S.—WILD, B.—MÜLLER-TASCH, T.—LEHMKUHL, E.—ZUGCK, C.—REGITZ-ZAGROSEK, V.—PANKUWEIT, S.—MAISCH, B.—ERTL, G.—GELBRICH, G.—ANGERMANN, C. E.: *Incidence rates and predictors of major and minor depression in patients with heart failure*, Int. J. Cardiol. **167** (2013), 502–507. DOI: https://doi.org/10.1016/j.ijcard.2012.01.062.

[27] LUMUMBA, V. W.—KIPROTICH, D.—MPAINE, M. L.—MAKENA, N. G.—KAVITA, M. D.: *Comparative analysis of cross-validation techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in machine learning models*, Am. J. Theor. Appl. Stat. **13** (2024), 127–137. DOI: https://doi.org/10.11648/j.ajtas.20241305.13.

[28] MATTHEWS, B. W.: *Comparison of the predicted and observed secondary structure of T4 phage lysozyme*, Biochim. Biophys. Acta **405** (1975), 442–451. DOI: https://doi.org/10.1016/0005-2795(75)90109-9.

[29] MCDONAGH, T. A.—METRA, M.—ADAMO, M.—GARDNER, R. S.—BAUMBACH, A.—BÖHM, M.—BURRI, H.—BUTLER, J.—ČELUTKIENĖ, J.—CHIONCEL, O.—CLELAND, J. G. F.—COATS, A. J. S.—CRESPO-LEIRO, M. G.—FARMAKIS, D.—GILARD, M.—HEYMANS, S.—HOES, A. W.—JAARSMA, T.—JANKOWSKA, E. A.—LAINSCAK, M.—LAM, C. S. P.—LYON, A. R.—MCMURRAY, J. J. V.—MEBAZAA, A.—MINDHAM, R.—MUNERETTO, C.—FRANCESCO PIEPOLI, M.—PRICE, S.—ROSANO, G. M. C.—RUSCHITZKA, F.—KATHRINE SKIBELUND, A.: *2021 ESC Guidelines for the diagnosis and treatment of acute and chronic heart failure*, Eur. Heart J. **42** (2021), 3599–3726. DOI: https://doi.org/10.1093/eurheartj/ehab368.

[30] METRA, M.—TOMASONI, D.—ADAMO, M.—BAYES-GENIS, A.—FILIPPATOS, G.—ABDELHAMID, M.—ADAMOPOULOS, S.—ANKER, S. D.—ANTOHI, L.—BÖHM, M.—BRAUNSCHWEIG, F.—GAL, T. B.—BUTLER, J.—CLELAND, J. G. F.—COHEN-SOLAL, A.—DAMMAN, K.—GUSTAFSSON, F.—HILL, L.—JANKOWSKA, E. A.—LAINSCAK, M.—LUND, L. H.—MCDONAGH, T.—MEBAZAA, A.—MOURA, B.—MULLENS, W.—PIEPOLI, M.—PONIKOWSKI, P.—RAKISHEVA, A.—RISTIC, A.—SAVARESE, G.—SEFEROVIC, P.—SHARMA, R.—TOCCHETTI, C. G.—YILMAZ, M. B.—VITALE, C.—VOLTERRANI, M.—VON HAEHLING, S.—CHIONCEL, O.—COATS, A. J. S.—ROSANO, G.: *Worsening of chronic heart failure: definition, epidemiology, management and prevention. A clinical consensus statement by the Heart Failure Association of the European Society of Cardiology.*, Eur. J. Heart Fail. **25** (2023), 776–791. DOI: https://doi.org/10.1002/ejhf.2874.

[31] NUMMINEN, R.—MONTOYA PEREZ, I.—JAMBOR, I.—PAHIKKALA, T.—AIROLA, A.: *Quicksort leave-pair-out cross-validation for ROC curve analysis*, Comput. Stat. **38** (2022), 1579–1595. DOI: https://doi.org/10.1007/s00180-022-01288-3.

[32] PARKER, B. J.—GÜNTER, S.—BEDO, J.: *Stratification bias in low signal microarray studies*, BMC bioinformatics **8** (2007), 326. DOI: https://doi.org/10.1186/1471-2105-8-326.

[33] RILEY, R. D.—ENSOR, J.—SNELL, K. I. E.—HARRELL, F. E. J.—MARTIN, G. P.—REITSMA, J. B.—MOONS, K. G. M.—COLLINS, G.—VAN SMEDEN, M.: *Calculating the sample size required for developing a clinical prediction model*, BMJ (Clinical research ed.) **368** (2020), m441. DOI: https://doi.org/10.1136/bmj.m441.

[34] RODRIGUEZ, M. P.—NAFEA, M.: *Centralized and federated heart disease classification models using UCI dataset and their Shapley-value based interpretability*, 2024. DOI: https://doi.org/10.48550/arXiv.2408.06183.

[35] ROTVIG, C.—CHRISTENSEN, A. V.—JUEL, K.—SVENDSEN, J. H.—JØRGENSEN, M. B.—RASMUSSEN, T. B.—BORREGAARD, B.—THRYSOEE, L.—THORUP, C. B.—MOLS, R. E.—BERG, S. K.: *The association between cardiac drug therapy and anxiety among cardiac patients: results from the national DenHeart survey*, BMC Cardiovascular Disorders **22** (2022), 280. DOI: https://doi.org/10.1186/s12872-022-02724-4.

[36] RUTLEDGE, T.—REIS, V. A.—LINKE, S. E.—GREENBERG, B. H.—MILLS, P. J.: *Depression in heart failure a meta-analytic review of prevalence, intervention effects, and associations with clinical outcomes*, J. Am. Coll. Cardiol. **48** (2006), 1527–1537. DOI: https://doi.org/10.1016/j.jacc.2006.06.055.

[37] RYDER, A. G.—YANG, J.—ZHU, X.—YAO, S.—YI, J.—HEINE, S. J.—BAGBY, R. M.: *The cultural shaping of depression: somatic symptoms in China, psychological symptoms in North America?*, J. Abnorm. Psychol. **117** (2008), 300–313. DOI: https://doi.org/10.1037/0021-843X.117.2.300.

[38] SAITO, T.—REHMSMEIER, M.: *The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets.*, PloS one **10** (2015), e0118432. DOI: https://doi.org/10.1371/journal.pone.0118432.

[39] SALAHUDEEN, T.—MAALOUF, M.—ELFADEL, I. A. M.—JELINEK, H. F.: *Predicting depression severity using machine learning models: Insights from mitochondrial peptides and clinical factors.*, PloS one **20** (2025), e0320955. DOI: https://doi.org/10.1371/journal.pone.0320955.

[40] SEIFFERT, C.—KHOSHGOFTAAR, T. M.—VAN HULSE, J.—NAPOLITANO, A.: *RUSBoost: A hybrid approach to alleviating class imbalance*, IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans **40** (2010), 185–197. DOI: https://doi.org/10.1109/TSMCA.2009.2029559.

[41] SUBBASWAMY, A.—SARIA, S.: *From development to deployment: dataset shift, causality, and shift-stable models in health AI*, Biostatistics (Oxford, England) **21** (2020), 345–352. DOI: https://doi.org/10.1093/biostatistics/kxz041.

[42] SUN, Z.—WANG, Z.—YUN, Z.—SUN, X.—LIN, J.—ZHANG, X.—WANG, Q.—DUAN, J.—HUANG, L.—LI, L.—YAO, K.: *Machine learning-based model for worsening heart failure risk in Chinese chronic heart failure patients.*, ESC Heart Failure **12** (2025), 211–228. DOI: https://doi.org/10.1002/ehf2.15066.

[43] TAO, Y.—YAO, C.—LIU, R.—QIU, H.—LIU, X.—WANG, B.—LI, H.: *Biomarker-based depression risk prediction in chronic heart failure patients: an interpretable machine learning approach.*, Frontiers in endocrinology **16** (2025), 1737713. DOI: https://doi.org/10.3389/fendo.2025.1737713.

[44] THE MATHWORKS, INC.: *Statistics and Machine Learning Toolbox Documentation: `fitclinear`, `fitctree`, `fitcensemble`.* MathWorks, Natick, MA, 2024. URL: `https://www.mathworks.com/help/stats/index.html`.

[45] TRNAVA UNIVERSITY AND MATHEMATICAL INSTITUTE OF THE SLOVAK ACADEMY OF SCIENCES AND MOVING MEDICAL MEDIA S.R.O.: *DIGIMENT project, Work Package 1: Domain analyses (internal research report).* Technical report, Trnava University, Mathematical Institute of the Slovak Academy of Sciences, and Moving Medical Media s.r.o., Bratislava and Trnava, 2025.

[46] TRNAVA UNIVERSITY AND MATHEMATICAL INSTITUTE OF THE SLOVAK ACADEMY OF SCIENCES AND MOVING MEDICAL MEDIA S.R.O.: *DIGIMENT project, Work Package 2: Concept formulation and validation (internal research report).* Technical report, Trnava University, Mathematical Institute of the Slovak Academy of Sciences, and Moving Medical Media s.r.o., Bratislava and Trnava, 2025.

[47] VAN STEIJN, F.—SOGANCIOGLU, G.—KAYA, H.: *Text-based Interpretable Depression Severity Modeling via Symptom Predictions.* In: *Proceedings of the 2022 International Conference on Multimodal Interaction*, ICMI '22, p. 139–147, New York, NY, USA, 2022. Association for Computing Machinery. DOI: https://doi.org/10.1145/3536221.3556579.

[48] VAROQUAUX, G.: *Cross-validation failure: Small sample sizes lead to large error bars*, NeuroImage **180** (2018), 68–77. DOI: https://doi.org/10.1016/j.neuroimage.2017.06.061.

[49] YOUDEN, W. J.: *Index for rating diagnostic tests*, Cancer **3** (1950), 32–35. DOI: https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3.

[50] ZIGMOND, A. S.—SNAITH, R. P.: *The hospital anxiety and depression scale*, Acta Psychiatr. Scand. **67** (1983), 361–370. DOI: https://doi.org/10.1111/j.1600-0447.1983.tb09716.x.

[51] ZWEIG, M. H.—CAMPBELL, G.: *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine.*, Clinical Chemistry **39** (1993), 561–577. DOI: https://doi.org/10.1093/clinchem/39.4.561.

Appendix A. Companion MATLAB script `CLASSIFIERS.m`

This appendix lists the reference implementation `CLASSIFIERS.m` used for strict out-of-sample evaluation with leave-one-out cross-validation (LOOCV) and, optionally, leave-pair-out cross-validation (LPOCV) on the Cheng Figshare cohort [6] loaded into the workspace matrix `DATA`. The script selects the 29 retained predictors and total PHQ-9 score as in Section 5, applies training-fold-only median imputation as in Subsection 4.3, and sweeps PHQ-9 cut-offs $\theta \in \{1, \ldots, 15\}$ under the protocol of Subsection 4.5: each θ defines the binary label; one `templateTree` per θ for tree-based learners; fold-wise `rng`$(\sigma_0 + \theta + i)$ with fixed master seed $\sigma_0 = 42$ (and $\sigma_0 + \theta +$ foldId for LPOCV pairs); hard labels from default `predict`; confusion metrics (a)–(h) from out-of-fold hard predictions; AUROC from out-of-fold positive-class scores (LOOCV) or pairwise score comparisons (LPOCV). At the top of the file, set `LEARNER` to `'rusboost'` (default; reproduces Tables 6 and 3, third row), `'dt'` or `'lr'` for the other reported learners, and `CV_SCHEME` to `'loocv'` (default), `'lpocv'` or `'both'` (the latter for Table 7). Long source lines are kept on one row in the `.m` file; in the listing below, automatic line breaking applies at spaces, and a small hook arrow (↪) marks the continuation of the same logical line (the line number is not repeated).

Listing 1. `CLASSIFIERS.m` (verbatim).

```
1  MASTER_SEED = 42;
2  LEARNER = 'lr';  % 'rusboost'|'dt'|'lr'
3  CV_SCHEME = 'loocv';   % 'loocv' | 'lpocv' | 'both'
4
5  hp = struct('MaxNumSplits', 10, 'MinLeafSize', 10, 'NumVariablesToSample', 'all',
     ↪ 'Surrogate', 'off', 'SplitCriterion', 'gdi', 'NumLearningCycles', 50,
     ↪ 'LearnRate', 0.1, 'Lambda', 1e-4, 'ClassNames', [0; 1]);
6
7  Sel_Pos = [1, 3, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,
     ↪ 29, 30, 31, 32, 33, 34, 36, 37, 38];
8  predictorNames = {'Age', 'Gender', 'BMI', 'BNP', 'EF', 'MaritalStatus', 'Education',
     ↪ 'EmploymentStatus', 'FamilyIncome', 'HealthInsurance', 'PlaceOfResidence',
     ↪ 'Cohabitant', 'Smoking', 'AlcoholConsumption', 'CoronaryHeartDisease',
     ↪ 'Arrhythmia', 'Hypertension', 'Diabetes', 'Hyperlipidemia',
     ↪ 'DilatedCardiomyopathy', 'CongenitalHeartDisease', 'KidneyDisease',
     ↪ 'CerebrovascularDisease', 'LungDisease', 'LiverDisease', 'DiseaseType',
     ↪ 'HeartFunctionClassification', 'CourseOfDisease', 'AmountOfMedication'};
9
```

```
PHQ9 = 49;
respName = 'TotalPHQScore';

DATA_SET1 = [DATA(:, Sel_Pos), DATA(:, PHQ9)];
th_vec = 1:15;
th_cnt = numel(th_vec);

runLoocv = any(strcmpi(CV_SCHEME, {'loocv', 'both'}));
runLpocv = any(strcmpi(CV_SCHEME, {'lpocv', 'both'}));

metricColNames = {'TPR', 'TNR', 'FPR', 'FNR', 'PPV', 'NPV', 'Accuracy',
    ↪ 'BalancedAccuracy', 'MCC', 'CohensKappa', 'F1', 'YoudenJ', 'TP', 'TN', 'FP',
    ↪ 'FN'};

if runLoocv
    LoocvMetricsMat = zeros(th_cnt, 16);
    LoocvAurocVec = nan(th_cnt, 1);
end
if runLpocv
    LpocvMetricsMat = zeros(th_cnt, 16);
    LpocvAurocVec = nan(th_cnt, 1);
    LpocvPairCountVec = zeros(th_cnt, 1);
end
OOF_LOOCV = cell(th_cnt, 1); OOF_LPOCV = cell(th_cnt, 1);

parfor j = 1:th_cnt
    th = th_vec(j);
    DATA_SET = DATA_SET1;
    DATA_SET(:, end) = DATA_SET(:, end) >= th;
    rngSeed = MASTER_SEED + th;

    if runLoocv
        outLoocv = evaluate_loocv(DATA_SET, rngSeed, LEARNER, hp, predictorNames,
    ↪ respName);
        LoocvMetricsMat(j, :) = outLoocv.metricsRow;
        LoocvAurocVec(j) = outLoocv.auroc;
        OOF_LOOCV{j} = outLoocv.oof;
    end
    if runLpocv
        outLpocv = evaluate_lpocv(DATA_SET, rngSeed, LEARNER, hp, predictorNames,
    ↪ respName);
        LpocvMetricsMat(j, :) = outLpocv.metricsRow;
        LpocvAurocVec(j) = outLpocv.auroc;
        LpocvPairCountVec(j) = outLpocv.nPairs;
        OOF_LPOCV{j} = outLpocv.oof;
    end
end

if runLoocv
    ResultTable_LOOCV = [table(th_vec(:), 'VariableNames', {'PHQ_threshold'}),
    ↪ array2table(LoocvMetricsMat, 'VariableNames', metricColNames),
    ↪ table(LoocvAurocVec, 'VariableNames', {'AUROC'})];
end

if runLpocv
```

```
59      ResultTable_LPOCV = [table(th_vec(:), 'VariableNames', {'PHQ_threshold'}),
    ↪ array2table(LpocvMetricsMat, 'VariableNames', metricColNames),
    ↪ table(LpocvAurocVec, LpocvPairCountVec, 'VariableNames', {'AUROC', 'N_pairs'})];
60  end
61
62  function out = evaluate_loocv(trainingData, rngSeed, learner, hp, predictorNames,
    ↪ respName)
63      n = size(trainingData, 1);
64      yTrue = logical(trainingData(:, end));
65      predHard = zeros(n, 1);
66      scorePos = nan(n, 1);
67      template = local_tree_template(hp);
68      rng(rngSeed, 'twister');
69      for ii = 1:n
70          rng(rngSeed + ii, 'twister');
71          [predHard(ii), scorePos(ii)] = classifier_fold_predict(trainingData, ii, [],
    ↪ learner, hp, predictorNames, respName, template);
72      end
73      [TP, TN, FP, FN] = local_counts(predHard, yTrue);
74      out.metricsRow = local_metrics_row(TP, TN, FP, FN);
75      out.auroc = local_auroc_scores(yTrue, scorePos);
76      out.oof = table((1:n)', yTrue, predHard, scorePos, 'VariableNames', {'Subject',
    ↪ 'Y_true', 'Y_pred_hard', 'Score_pos'});
77  end
78
79  function out = evaluate_lpocv(trainingData, rngSeed, learner, hp, predictorNames,
    ↪ respName)
80      n = size(trainingData, 1);
81      yTrue = logical(trainingData(:, end));
82      posIdx = find(yTrue); negIdx = find(~yTrue);
83      if isempty(posIdx) || isempty(negIdx), error('CLASSIFIERS:LpocvNeedTwoClasses',
    ↪ 'LPOCV requires at least one positive and one negative patient.'); end
84      nPairs = numel(posIdx) * numel(negIdx);
85      pairWins = 0;
86      foldId = 0;
87      votePos = zeros(n, 1); voteNeg = zeros(n, 1); scoreSum = zeros(n, 1); scoreCount =
    ↪ zeros(n, 1);
88      template = local_tree_template(hp);
89      rng(rngSeed, 'twister');
90      for ip = 1:numel(posIdx)
91          for jn = 1:numel(negIdx)
92              foldId = foldId + 1;
93              rng(rngSeed + foldId, 'twister');
94              ii = posIdx(ip); jj = negIdx(jn);
95              holdOut = [ii; jj];
96              [hP, sP] = classifier_fold_predict(trainingData, ii, holdOut, learner, hp,
    ↪ predictorNames, respName, template);
97              [hN, sN] = classifier_fold_predict(trainingData, jj, holdOut, learner, hp,
    ↪ predictorNames, respName, template);
98              pairWins = pairWins + local_pair_win(sP, sN);
99              votePos(ii) = votePos(ii) + hP; voteNeg(ii) = voteNeg(ii) + (1 - hP);
100             votePos(jj) = votePos(jj) + hN; voteNeg(jj) = voteNeg(jj) + (1 - hN);
101             scoreSum(ii) = scoreSum(ii) + sP; scoreCount(ii) = scoreCount(ii) + 1;
102             scoreSum(jj) = scoreSum(jj) + sN; scoreCount(jj) = scoreCount(jj) + 1;
103         end
```

```
    end
    hasVote = (votePos + voteNeg) > 0;
    predHard = zeros(n, 1);
    predHard(hasVote) = votePos(hasVote) > voteNeg(hasVote);
    [TP, TN, FP, FN] = local_counts(predHard, yTrue);
    out.metricsRow = local_metrics_row(TP, TN, FP, FN);
    out.auroc = pairWins / nPairs;
    out.nPairs = nPairs;
    out.oof = table((1:n)', yTrue, predHard, scoreSum ./ max(scoreCount, 1),
    ↪ scoreCount, 'VariableNames', {'Subject', 'Y_true', 'Y_pred_majority',
    ↪ 'Mean_score_pos', 'N_test_appearances'});
end

function [predHard, scorePos] = classifier_fold_predict(trainingData, testRow,
   ↪ extraHoldOut, learner, hp, predictorNames, respName, template)
    n = size(trainingData, 1);
    p = size(trainingData, 2) - 1;
    varNames = [predictorNames(:)', {respName}];
    tr = true(n, 1);
    tr(testRow) = false;
    if ~isempty(extraHoldOut), tr(extraHoldOut) = false; end
    trainM = trainingData(tr, :);
    testM = trainingData(testRow, :);
    [Xtr, Xte] = local_impute_median(trainM(:, 1:p), testM(:, 1:p));
    trainM(:, 1:p) = Xtr;
    testM(:, 1:p) = Xte;
    Ttr = array2table(trainM, 'VariableNames', varNames);
    Tte = array2table(testM, 'VariableNames', varNames);
    Xtr = Ttr(:, predictorNames);
    Xte = Tte(:, predictorNames);
    y_train = double(trainM(:, end));
    learner = lower(char(learner));
    if numel(unique(y_train)) < 2, predHard = double(y_train(1) == 1); scorePos =
    ↪ predHard; return; end
    if strcmp(learner, 'rusboost')
        Mdl = fitcensemble(Xtr, y_train, 'Method', 'RUSBoost', 'NumLearningCycles',
    ↪ hp.NumLearningCycles, 'Learners', template, 'LearnRate', hp.LearnRate,
    ↪ 'ClassNames', hp.ClassNames);
    elseif strcmp(learner, 'dt')
        Wt = local_invfreq_weights(y_train);
        Mdl = fitctree(Xtr, y_train, 'SplitCriterion', hp.SplitCriterion, 'Surrogate',
    ↪ hp.Surrogate, 'MaxNumSplits', hp.MaxNumSplits, 'MinLeafSize', hp.MinLeafSize,
    ↪ 'NumVariablesToSample', hp.NumVariablesToSample, 'ClassNames', hp.ClassNames,
    ↪ 'Weights', Wt);
    elseif strcmp(learner, 'lr')
        Wt = local_invfreq_weights(y_train);
        Mdl = fitclinear(Xtr, y_train, 'Learner', 'logistic', 'Regularization',
    ↪ 'ridge', 'Lambda', hp.Lambda, 'ClassNames', hp.ClassNames, 'Weights', Wt);
    else
        error('CLASSIFIERS:UnknownLearner', 'LEARNER must be rusboost, dt, or lr.');
    end
    lblTest = predict(Mdl, Xte);
    predHard = local_to_binary(lblTest);
    [~, scores] = predict(Mdl, Xte);
    scorePos = local_pos_score(Mdl, scores);
```

```
end

function Wt = local_invfreq_weights(y)
    y = logical(y(:));
    nt = numel(y);
    wPos = sum(y) / max(nt, eps);
    wNeg = sum(~y) / max(nt, eps);
    Wt = zeros(nt, 1);
    Wt(y) = 1 / max(wPos, eps);
    Wt(~y) = 1 / max(wNeg, eps);
end

function t = local_tree_template(hp)
    t = templateTree('MaxNumSplits', hp.MaxNumSplits, 'MinLeafSize', hp.MinLeafSize,
    ↪ 'NumVariablesToSample', hp.NumVariablesToSample, 'Surrogate', hp.Surrogate);
end

function w = local_pair_win(sP, sN)
    w = (sP > sN) + 0.5 * (sP == sN);
end

function a = local_auroc_scores(yTrue, scorePos)
    yTrue = logical(yTrue(:));
    scorePos = scorePos(:);
    ok = ~isnan(scorePos);
    yTrue = yTrue(ok);
    scorePos = scorePos(ok);
    posIdx = find(yTrue); negIdx = find(~yTrue);
    if isempty(posIdx) || isempty(negIdx), a = nan; return; end
    pairWins = 0;
    for ip = 1:numel(posIdx)
        sP = scorePos(posIdx(ip));
        for jn = 1:numel(negIdx)
            pairWins = pairWins + local_pair_win(sP, scorePos(negIdx(jn)));
        end
    end
    a = pairWins / (numel(posIdx) * numel(negIdx));
end

function s = local_pos_score(Mdl, scores)
    if size(scores, 2) == 1, s = scores(:, 1); return; end
    cn = Mdl.ClassNames;
    if isnumeric(cn), k = find(cn == 1, 1);
    elseif iscategorical(cn), k = find(cn == categorical(1), 1);
    else, k = 2; end
    if isempty(k), k = size(scores, 2); end
    s = scores(:, k);
end

function [TP, TN, FP, FN] = local_counts(pred, yTrue)
    pred = pred(:) ~= 0;
    yTrue = logical(yTrue(:));
    TP = sum(pred & yTrue); FN = sum(~pred & yTrue);
    FP = sum(pred & ~yTrue); TN = sum(~pred & ~yTrue);
end
```

```
203
204 function [Xtr, Xte] = local_impute_median(Xtr, Xte)
205     Xtr = double(Xtr);
206     Xte = double(Xte);
207     for c = 1:size(Xtr, 2)
208         col = Xtr(:, c);
209         good = col(~isnan(col));
210         if isempty(good), m = 0; else, m = median(good); end
211         col(isnan(col)) = m;
212         Xtr(:, c) = col;
213         te = Xte(:, c);
214         te(isnan(te)) = m;
215         Xte(:, c) = te;
216     end
217 end
218
219 function v = local_to_binary(pred)
220     p = pred(:);
221     if isnumeric(p), v = double(p == 1);
222     elseif islogical(p), v = double(p);
223     elseif iscategorical(p), v = double(string(p) == "1");
224     else, v = double(str2double(string(p)) == 1); end
225 end
226
227 function row = local_metrics_row(TP, TN, FP, FN)
228     TP = double(TP); TN = double(TN);
229     FP = double(FP); FN = double(FN);
230     n = TP + TN + FP + FN;
231     P = TP + FN;
232     Nneg = TN + FP;
233     TPR = TP / max(P, eps); TNR = TN / max(Nneg, eps);
234     FPR = FP / max(Nneg, eps); FNR = FN / max(P, eps);
235     PPV = TP / max(TP + FP, eps); NPV = TN / max(TN + FN, eps);
236     acc = (TP + TN) / max(n, eps);
237     balAcc = (TPR + TNR) / 2;
238     denomMcc = (TP + FP) * (TP + FN) * (TN + FP) * (TN + FN);
239     mcc = 0; if denomMcc > 0, mcc = (TP * TN - FP * FN) / sqrt(denomMcc); end
240     pPos = P / max(n, eps);
241     pPredPos = (TP + FP) / max(n, eps);
242     pe = pPos * pPredPos + (1 - pPos) * (1 - pPredPos);
243     kappa = 0; if abs(1 - pe) >= eps, kappa = (acc - pe) / (1 - pe); end
244     f1 = 0; if 2 * TP + FP + FN > 0, f1 = 2 * TP / (2 * TP + FP + FN); end
245     row = [TPR, TNR, FPR, FNR, PPV, NPV, acc, balAcc, mcc, kappa, f1, TPR + TNR - 1,
    ↪ TP, TN, FP, FN];
246 end
```

Received June 8, 2026
Revised July 27, 2026
Accepted July 28, 2026

Mathematical Institute
Slovak Academy of Sciences
Štefánikova 49
814 73 Bratislava
SLOVAKIA

E-mail: mracka@mat.savba.sk
zacik@mat.savba.sk