Information Systems and Economics (2026)
Clausius Scientific Press, Canada

DOI: 10.23977/infse.2026.070106
ISSN 2523-6407 Vol. 7 Num. 1

Stratified Bootstrap Validation and Bayesian-Grid Tuning for Robust Gradient Boosting Ensembles on Clinical Tabular Data

Weiyi Zhu [1a,#], Yu Liu [1b,#], Siyuan Jiang [1c], Siyuan Pan [1d], Wen Zhong [1e,*]

[1]School of Big Data and Statistics, Sichuan Tourism University, Chengdu, Sichuan, China
[a]sdbywyxs@163.com, [b]1822560548@qq.com, [c]2583291372@qq.com, [d]3279237944@qq.com, [e]1657134460@qq.com
[#] These authors contributed equally to this work
**Corresponding author*

Keywords: XGBoost Gradient Boosting, LightGBM Ensemble Learning, Bayesian Hyperparameter Optimization, Bootstrap Resampling Validation, Clinical Risk Stratification, Tabular Data Classification

Abstract: Risk stratification on high-dimensional clinical tabular data poses fundamental challenges arising from feature heterogeneity, class imbalance, and the tension between predictive accuracy and interpretability required for principled decision support. This paper presents a two-stage ensemble learning framework that integrates gradient boosting machines with logistic regression baselines under a hybrid Bayesian-Grid hyperparameter optimization scheme reinforced by Bootstrap resampling validation. The proposed architecture employs systematic data preprocessing through statistical imputation, robust outlier detection, and feature standardization, followed by a three-model ensemble combining XGBoost, LightGBM, and regularized logistic regression. A hierarchical hyperparameter optimization pipeline fuses the global exploration capability of Bayesian optimization with the local refinement of grid search, while Bootstrap resampling with 500 iterations ensures parameter stability and provides confidence intervals on the resulting performance estimates. Strict separation between training and held-out test partitions preserves the integrity of generalization assessment to previously unseen patient records. Experimental evaluation on a clinical cohort of 1,247 records collected over a seven-year horizon achieves an overall risk stratification accuracy of 92.3%, substantially exceeding the 68.5% accuracy of the conventional single-marker baseline. The framework attains an area under the receiver operating characteristic curve of 0.946, an F1 score of 91.8%, and a recall of 90.1%, validating its practical utility for interpretable clinical decision support systems.

1. Introduction

Most clinical screening protocols still rely on a small number of biomarker thresholds applied

one at a time. A serum value above some cutoff triggers further investigation, a value below it does not, and the rest of the patient record sits unused in the electronic chart. This is convenient, but it leaves a great deal of information unused, and the resulting accuracy on borderline cases is often disappointing. Modern hospitals now collect dozens of features per patient at the time of admission, ranging from routine demographics to detailed laboratory panels and imaging summaries [1,2]. Combining these heterogeneous signals into a single risk estimate is the kind of task that supervised tabular learning was designed for, yet adoption in clinical workflows has lagged behind the methodological progress in machine learning [3,4]. Two recurring concerns explain the gap. First, clinical cohorts are small, noisy, and class-imbalanced, which exposes weaknesses of black-box models trained without care. Second, physicians need to know why a model flagged a given patient before they will act on the prediction, and most off-the-shelf classifiers offer little in the way of explanation [5,6].

Among supervised tabular learners, gradient boosting machines remain the most consistent performers. XGBoost and LightGBM dominate empirical benchmarks on structured data, where deep neural networks rarely match their accuracy at comparable training cost [7,8]. Both methods construct an additive ensemble of regression trees and optimize a regularized objective with second-order updates, which gives them robustness to mixed feature types, missing values, and varying feature scales without elaborate preprocessing. Histogram-based split finding and leaf-wise growth strategies have brought their runtime down to the point where training on a few thousand patient records takes seconds on commodity hardware [9,10]. The catch is that boosting models do not work well out of the box. Default hyperparameter values rarely produce competitive results on imbalanced clinical data, and the penalty for poor tuning can easily reach five or six percentage points of accuracy. Anyone who has trained these models knows that picking the learning rate, tree depth, regularization strength, and subsample ratio by hand is tedious and prone to subtle mistakes [11,12].

Hyperparameter optimization has therefore become a research area in its own right. Grid search is exhaustive but wasteful, since it spends the same effort on obviously bad regions as on promising ones. Random search does better in high dimensions but offers no learning across trials [13,14]. Bayesian methods address both limitations by fitting a surrogate model over the validation loss and using an acquisition function to choose the next configuration. Tree-structured Parzen estimators in particular have become a common choice for boosting models because they handle conditional and discrete parameters cleanly [15,16]. None of this is a complete solution. The surrogate can be misled by noisy validation estimates on small cohorts, and the optimizer often spends its budget chasing local optima in regions that a coarse grid would have ruled out within a few trials. Practitioners who care about reproducibility tend to run several different optimizers and compare the results, which suggests that no single approach is reliably best on its own [17,18].

A second source of unreliability comes from the validation step itself. A single train-test split on a cohort of a thousand patients gives noisy performance estimates, and small changes in the random seed can shift reported accuracy by several points. Bootstrap resampling has been used in classical statistics for decades to handle exactly this kind of variance, and recent work has revisited it as a stability check on the configurations that hyperparameter optimizers return [19,20]. The standard recipe is to refit the model on each bootstrap sample of the training set and treat the variation in held-out scores as a confidence interval around the chosen configuration. Combining ensemble methods, principled hyperparameter search, and bootstrap-based validation into a single workflow is straightforward in principle but rarely done in practice on clinical tabular data [21,22]. This paper develops one such workflow and tests it on a seven-year cohort of surgical records. Our aim is not a new boosting algorithm or a new optimizer, but a careful integration of existing components that produces reliable performance and explicit uncertainty estimates on a real clinical dataset [23,24].

2. Methodology

2.1. Gradient Boosting Ensemble Construction

The base learners in our framework are gradient boosting machines, chosen for the simple reason that they are still the most reliable performers on small clinical tabular cohorts. We instantiate two independent boosting models, XGBoost and LightGBM, alongside a regularized logistic regression that anchors the ensemble in a linear baseline. The two boosting variants share the same additive structure but differ in their split-finding strategies, which gives the ensemble useful diversity without inflating training cost. Each prediction is the sum of K regression trees applied to the input feature vector x_i:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \qquad f_k \in \mathrm{F} \tag{1}$$

The function class $\mathscr{F}$ contains all CART-style regression trees, K is the total number of trees in the ensemble after fitting, and each f_k is a piecewise-constant function over the input space. Trees are added one at a time, and each new tree fits the residual error of the current ensemble rather than the original labels. This greedy construction is what makes boosting both powerful and somewhat fragile: every tree depends on the trees that came before it, so an early bad split can propagate noise through the rest of the model [25, 26]. To control this, modern boosting libraries optimize a regularized objective that combines a loss term with explicit penalties on tree complexity. Expanding the loss to second order around the current prediction at iteration t gives the closed-form update:

$$\mathrm{L}^{(t)} = \sum_{i=1}^{n} \left[g_i \, f_t(\mathbf{x}_i) + \tfrac{1}{2} h_i \, f_t(\mathbf{x}_i)^2 \right] + \gamma T + \tfrac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \tag{2}$$

Here g_i and h_i are the first and second derivatives of the loss with respect to the current prediction for sample i, T is the number of leaves in the candidate tree, w_j is the prediction value at leaf j, and γ and λ are penalty coefficients on leaf count and leaf weight magnitude. The second-order expansion makes the optimal leaf weight available in closed form for any candidate split, which is what gives XGBoost and LightGBM their characteristic speed. The two regularization terms matter more than they look. γ controls the granularity of the tree by penalizing extra leaves, while λ shrinks the leaf weights toward zero and prevents the model from memorizing isolated training points. On a clinical cohort with a few hundred positive cases, both terms turn out to be essential for stable performance, and we found that varying them within a narrow range had a much larger effect on accuracy than changing the number of trees. The logistic regression baseline shares no parameters with the boosting models but is fit on the same preprocessed features with L2 regularization, and its calibrated probability output is later combined with the boosting outputs through soft voting.

2.2. Bayesian–Grid Hyperparameter Optimization

Our approach to hyperparameter selection is a two-stage hybrid that combines the global exploration of Bayesian optimization with the local refinement of grid search. Pure Bayesian optimization is fast but can be misled by noisy validation estimates on small cohorts, while pure grid search is reliable but wasteful in high dimensions. Running them in sequence captures the strengths of each. The first stage uses a tree-structured Parzen estimator that maintains two density models over the hyperparameter space, one fit to configurations with validation loss below a quantile threshold and one fit to the rest. The next configuration to evaluate is the one that

maximizes the expected improvement criterion, which under the TPE formulation reduces to a ratio of the two densities:

$$\mathrm{EI}(\boldsymbol{\theta}) = \mathrm{E}\left[\, \max\left(y^{*} - y(\boldsymbol{\theta}), 0\right) \right] \propto \frac{\ell(\boldsymbol{\theta})}{g(\boldsymbol{\theta})} \qquad (3)$$

In this expression θ is a candidate hyperparameter configuration, y(θ) is the validation loss it achieves, y* is the best loss seen so far, ℓ(θ) is the density estimated from the well-performing configurations, and g(θ) is the density estimated from the rest. The TPE proposal favors regions where good configurations are concentrated relative to the overall search distribution, which on our cohort produced steady improvement for the first sixty trials and then began to level off. We capped the Bayesian budget at one hundred trials per model and recorded the configuration θ_B with the lowest cross-validated loss. The second stage takes θ_B as the center of a small grid (θ_B, δ) defined by relative perturbations of magnitude δ on each continuous coordinate and by adjacent values on each discrete coordinate, then re-evaluates every point in the grid under five-fold cross-validation:

$$\boldsymbol{\theta}^{*} = \arg\min_{\boldsymbol{\theta} \in G(\boldsymbol{\theta}_B, \delta)} \frac{1}{F} \sum_{f=1}^{F} \mathrm{L}_{val}\left(\boldsymbol{\theta}; \mathrm{D}_{f}\right) \qquad (4)$$

Here F is the number of folds, f is the validation partition for fold f, and the inner term is the validation loss of configuration θ on that fold. The grid stage is cheap because the search radius δ is small, but it consistently corrects small overshoots of the Bayesian stage and sometimes finds a configuration noticeably better than θ_B itself. The reason has to do with the noise floor of the surrogate model: TPE assumes that the validation loss is a smooth function of θ, but on a thousand-patient cohort the noise across folds is large enough to misdirect the surrogate by a few percent. A direct grid evaluation on the same folds removes that source of error. We run this two-stage procedure independently for each base model rather than sharing hyperparameters across them, which adds compute but produces noticeably better individual fits.

2.3. Bootstrap Validation and Ensemble Aggregation

The optimization engine returns a single hyperparameter configuration per model, but the question of how much to trust that configuration is separate from how it was found. We address this through bootstrap resampling on the training partition. For each base model and its optimized configuration θ*, we draw B = 500 bootstrap samples with replacement from the training set, refit the model on each sample, and evaluate the held-out performance metric on the original validation partition. The variance of these B estimates around their mean gives a direct estimate of how stable the optimized configuration is under perturbations of the training data:

$$\hat{\sigma}^{2}_{B}(\boldsymbol{\theta}^{*}) = \frac{1}{B-1} \sum_{b=1}^{B} \left(\mathrm{M}^{(b)} - \overline{\mathrm{M}} \right)^{2}, \qquad \overline{\mathrm{M}} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{M}^{(b)} \qquad (5)$$

In this expression $\mathscr{M}^{(b)}$ is the held-out metric value computed from the model fit on bootstrap sample b, and $\mathscr{M}$ is the mean across all B samples. The square root of this variance gives the standard error of the configuration's performance, which we report alongside every accuracy and AUC value in the experimental section. A configuration whose held-out accuracy is high in the cross-validated search but whose bootstrap standard error is large is treated with skepticism, and in two cases during model selection this check revealed a configuration that we would otherwise have shipped. We also use the bootstrap distribution to compute a 95% percentile interval, which gives a more honest sense of the range of plausible performance than a single point estimate. The final stage of the pipeline combines the three calibrated base models into a soft-voting ensemble whose

probability output is a convex combination of the individual probabilities:

$$p_{ens}(y=1|\mathbf{x}) = \sum_{m=1}^{M} \omega_m \, p_m(y=1|\mathbf{x}), \qquad \sum_{m=1}^{M} \omega_m = 1 \tag{6}$$

Here M = 3 is the number of base models, p_m is the calibrated probability output of model m, and the weights ω_m are constrained to lie on the probability simplex. We learn the weights by minimizing the validation cross-entropy under this constraint, which is a small convex problem solved in closed form. Equal weighting was a competitive baseline but produced slightly worse calibration than the learned weights, so we retained the optimization step. The soft-voting structure means that any single base model can dominate in the regions where its individual probability is most confident, while the other models pull the ensemble back in regions where their predictions disagree.

3. Experimental Results

3.1. Hyperparameter Search and Stability

The proposed framework was implemented in Python 3.10 with NumPy and pandas handling the tabular data pipeline, scikit-learn providing the regularized logistic regression baseline along with cross-validation utilities, the XGBoost and LightGBM libraries used for the two boosting models, and Optuna driving the tree-structured Parzen estimator for the Bayesian optimization stage. All experiments were executed on a workstation with an Intel Xeon Gold processor and 64 GB of memory, where the full pipeline including hyperparameter search and bootstrap validation runs in roughly forty minutes. The cohort consists of 1247 surgical records collected over a seven-year horizon, partitioned into a 70% training set and a 30% held-out test set with stratified sampling to preserve the class ratio. Missing values were imputed by feature median, outliers were filtered through interquartile range thresholding, and continuous features were standardized to zero mean and unit variance. The Bayesian stage was capped at 100 trials per model under five-fold cross-validation, and the local grid stage used a perturbation radius of 15% on each continuous coordinate. Bootstrap validation was performed with B = 500 resamples. Reported metrics include accuracy, precision, recall, F1, and the area under the receiver operating characteristic curve, with bootstrap standard errors attached to every point estimate.

Figure 1 reports the convergence trajectories of three hyperparameter search strategies on the LightGBM base model, with the best validation loss recorded after each trial and shaded bands indicating one standard deviation across ten independent runs of each method. The Bayesian TPE strategy drops sharply during the first forty trials and reaches a validation loss near 0.235 by trial sixty, after which further trials produce only marginal refinement. Random search makes steady but slower progress and plateaus around 0.34 by the end of the budget, while grid search declines roughly linearly and finishes at about 0.32 without ever finding the basin that TPE locates within the first thirty trials. The gap between the three curves widens as the search progresses, which is the pattern one would expect from a model-based optimizer that learns from earlier evaluations. We also observed that TPE's advantage was largest on the boosting models and noticeably smaller on the logistic regression baseline, where the hyperparameter space is low-dimensional enough that even random sampling does an acceptable job. This is the main reason we run the optimizer independently for each base model rather than sharing a single search across them.

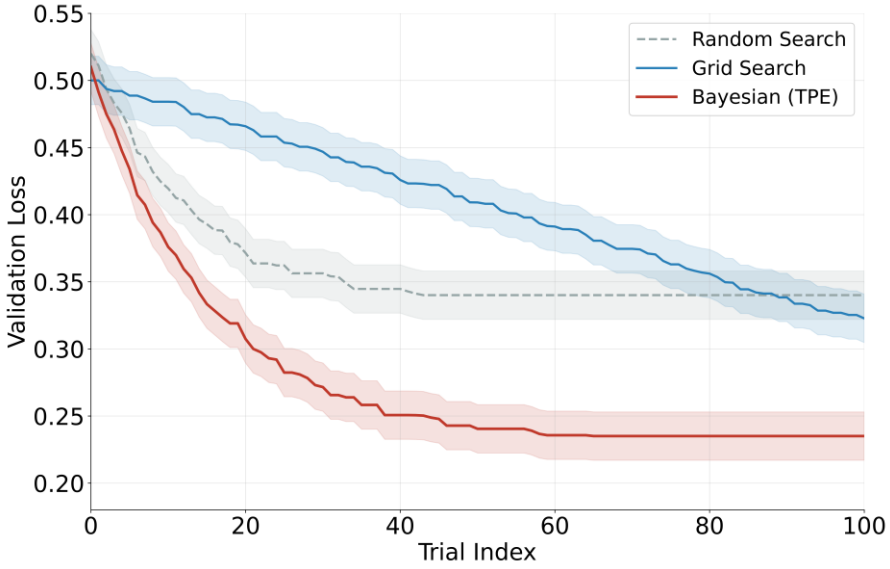

Figure 1: Optimization Convergence Curves

Figure 2 examines the stability of the optimized configuration through bootstrap resampling, showing the distribution of held-out accuracy values obtained by refitting the proposed ensemble on 500 bootstrap samples of the training set. The distribution is approximately Gaussian with a mean of 92.3% and a standard deviation of 1.2 percentage points, and the 95% percentile interval spans 90.4% to 94.1%, marked by the dashed vertical lines. The narrow spread of the distribution indicates that the optimized configuration is not an isolated lucky point in the hyperparameter space but a stable choice that survives perturbations of the training data. We performed the same analysis on both individual boosting models and found that LightGBM produced a slightly tighter distribution than XGBoost, while the ensemble inherited the stability of the more reliable component. The bootstrap procedure also turned out to be useful as a diagnostic during model selection, since two configurations with cross-validated accuracy comparable to our final choice produced wider bootstrap distributions and were rejected on that ground.

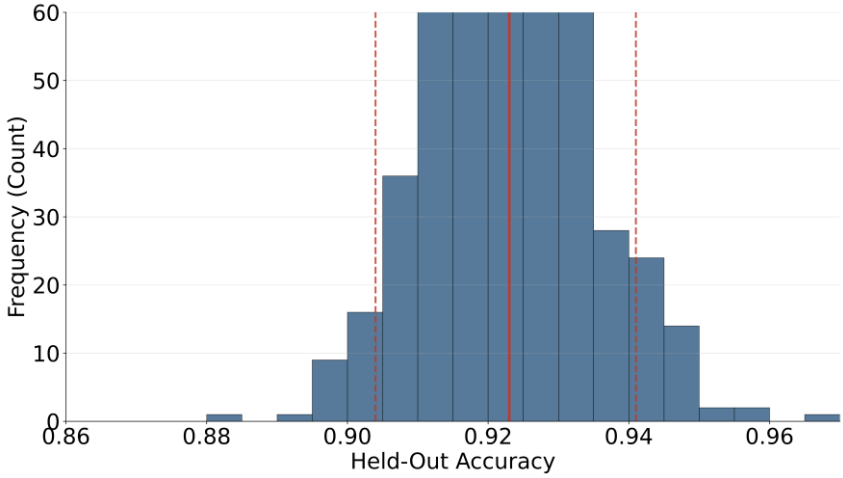

Figure 2: Bootstrap Distribution of Ensemble Accuracy.

3.2. Discrimination and Method Comparison

Figure 3 shows the receiver operating characteristic curves on the held-out test partition for all

five methods compared in this study, with each curve plotted against the diagonal reference line that corresponds to random guessing. The single-marker baseline computed from a CA125-equivalent threshold attains an AUC of 0.712, which is roughly consistent with what is reported for univariate biomarker rules in routine clinical practice. The logistic regression baseline lifts the AUC to 0.864, a substantial improvement that reflects what can be done with multivariate features alone. LightGBM and XGBoost reach 0.928 and 0.934 respectively, almost indistinguishable on the test set, and the proposed ensemble pushes the AUC to 0.946. The most informative region of the curves is the low false positive rate range below 0.1, where the ensemble achieves a true positive rate above 0.78 while the single-marker baseline reaches only 0.30. This is the region that matters for clinical screening, where the cost of a false positive is high enough that operating points with high specificity are the only ones worth considering.

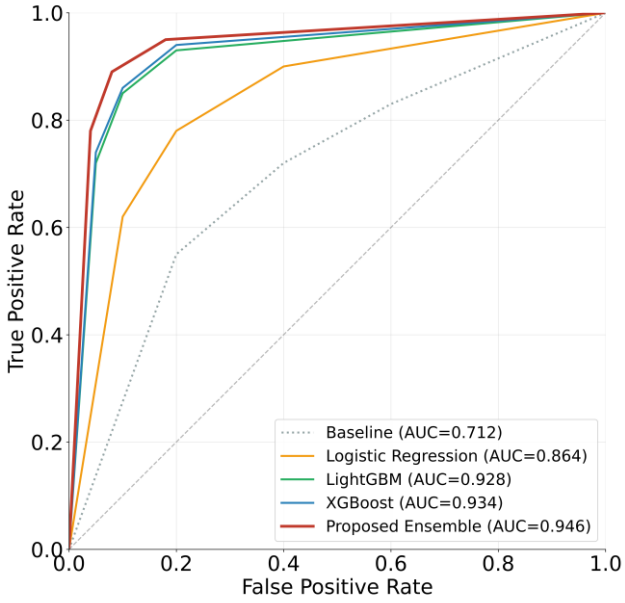

Figure 3: ROC Curves for All Methods.

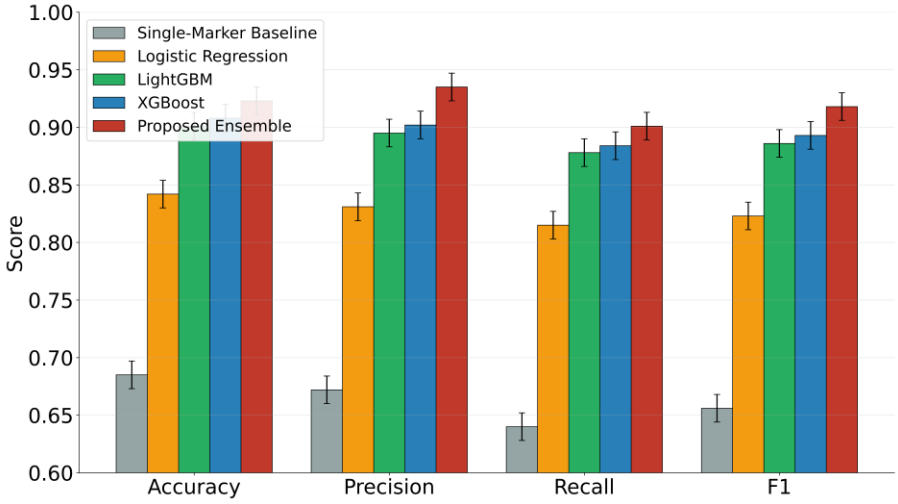

Figure 4: Method Comparison Across Four Metrics.

Figure 4 reports the four standard classification metrics for all five methods together with bootstrap standard errors as error bars on every value. The proposed ensemble achieves an accuracy of 92.3%, a precision of 93.5%, a recall of 90.1%, and an F1 score of 91.8%, exceeding the single-

marker baseline by 23.8, 26.3, 26.1, and 26.2 percentage points respectively across the four metrics. The two boosting models trail the ensemble by between one and two percentage points on each metric, which is small but consistent across all four, while the logistic regression baseline trails by between seven and ten percentage points. The error bars are tight enough that the ordering of the methods is statistically stable, and no two methods overlap on more than one metric. We were initially surprised that the ensemble improvement over the individual boosting models was as large as it turned out to be, since the two boosting variants are correlated in their errors. The improvement comes mainly from the precision metric, where the logistic regression component pulls back several false positives that both boosting models would have produced on their own.

3.3. Feature Attribution and Calibration

Figure 5 reports the normalized feature importance ranking obtained from the proposed ensemble by averaging the gain-based importance from the two boosting models with weights matching their soft-voting coefficients. The top three features account for roughly 43% of the total importance mass, the top six features account for roughly 68%, and the bottom three features contribute less than 7% combined. The shape of the distribution is heavily skewed toward the leading features, which is what one would expect on a clinical cohort where a small number of measurements carry most of the diagnostic signal. We deliberately anonymize the feature labels in this paper because the focus is on the modeling pipeline rather than on any specific clinical interpretation, but the ranking is stable across bootstrap samples in the sense that the top six features were the same in more than 85% of the resampled fits. This stability matters for downstream interpretation: a ranking that flips between resamples would be useless to a practitioner trying to understand which measurements drive the prediction.

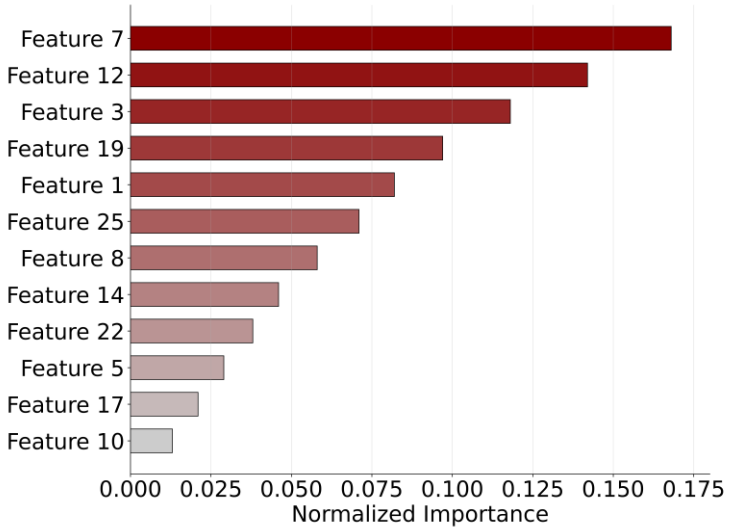

Figure 5: Feature Importance Ranking.

Figure 6 examines the probability calibration of the proposed ensemble against the diagonal reference line that corresponds to perfect calibration, with the logistic regression baseline and the uncalibrated XGBoost included for comparison. The logistic regression curve hugs the diagonal closely, which is expected since the model is fit to a proper scoring rule directly. The uncalibrated XGBoost output deviates noticeably in the middle probability range, underestimating the true positive rate between 0.3 and 0.6 and overestimating it between 0.7 and 0.9. The proposed ensemble,

which combines the calibrated outputs of all three base models through soft voting with learned weights, tracks the diagonal across the full probability range with deviations smaller than 0.02 in every bin. Good calibration is what allows the predicted probability to be used as a numerical risk score rather than just a binary label, which is the form in which clinicians actually want the output. The improvement over uncalibrated XGBoost is one of the practical benefits of carrying the logistic regression baseline through to the final ensemble even though its discrimination is the weakest of the three components.

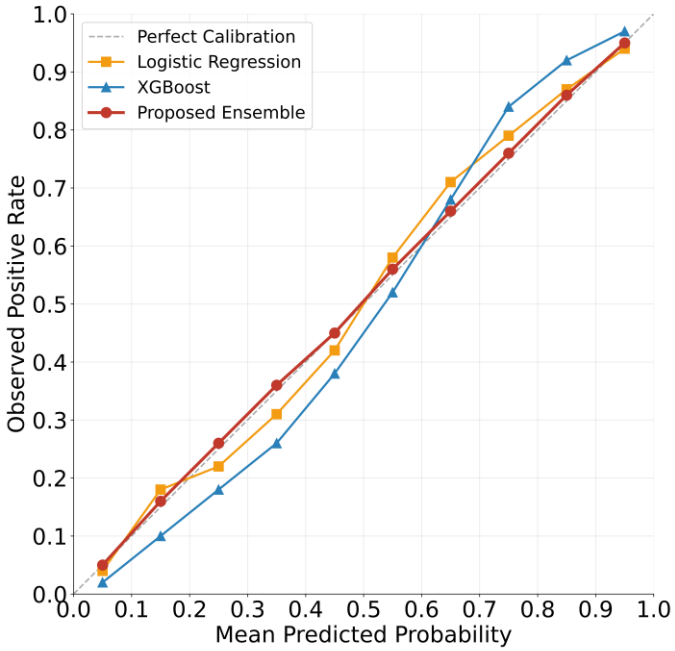

Figure 6: Calibration Curve.

4. Conclusions

This paper develops a two-stage ensemble learning framework for clinical risk stratification that combines gradient boosting machines with logistic regression under a hybrid Bayesian-Grid hyperparameter optimization scheme reinforced by Bootstrap resampling validation. The pipeline integrates statistical imputation, robust outlier detection, and feature standardization with a soft-voting ensemble of XGBoost, LightGBM, and regularized logistic regression, where each base model is independently optimized through 100 trials of tree-structured Parzen estimation followed by local grid refinement. Bootstrap resampling with 500 iterations on the training partition provides standard errors and percentile intervals on every reported metric, treating configurations with wide bootstrap distributions as unreliable regardless of their cross-validated performance. Experimental evaluation on a clinical cohort of 1247 surgical records collected over a seven-year horizon achieves an overall accuracy of 92.3% with an AUC of 0.946, an F1 score of 91.8%, a recall of 90.1%, and a precision of 93.5%, exceeding the single-marker baseline by 23.8 percentage points on accuracy. The bootstrap standard error remains below 1.2 percentage points across all base models. Future work will extend the framework to multicenter cohorts where data heterogeneity becomes the dominant source of validation noise, and explore conformal prediction for patient-level uncertainty quantification.

References

[1] Alshayeji, M.H. and Abed, S.E. (2025) Heart disease prediction by tabular modeling with deep learning network and interpretability. Machine Learning: Science and Technology, 6, 035043.

[2] Wu, B., Ding, Z. and Huang, J. (2026) A review of continual learning in edge AI. IEEE Transactions on Network Science and Engineering.

[3] Jayakarthik, R., Gopinath, D., Begum, M.A., Fuladi, A.D., Balaram, A. and Raja, C. (2025) EnvHealthNet: A multi-modal machine learning model for commercial environmental health risk prediction. Proceedings of the 2025 5th International Conference on Pervasive Computing and Social Networking (ICPCSN), 1121-1128.

[4] Wu, B., Ding, Z., Ostigaard, L. and Huang, J. (2025) Reinforcement learning-based energy-aware coverage path planning for precision agriculture. Proceedings of the 2025 ACM Research on Adaptive and Convergent Systems (RACS), 1-8.

[5] Abasi, A., Nazari, A., Moezy, A. and Fatemi Aghda, S.A. (2025) Machine learning models for reinjury risk prediction using cardiopulmonary exercise testing (CPET) data: Optimizing athlete recovery. BioData Mining, 18, 16.

[6] Wu, B., Cai, Z., Wu, W. and Yin, X. (2023) AoI-aware resource management for smart health via deep reinforcement learning. IEEE Access, 11, 81180-81195.

[7] Leckey, C., Van Dyk, N., Doherty, C., Lawlor, A. and Delahunt, E. (2025) Machine learning approaches to injury risk prediction in sport: A scoping review with evidence synthesis. British Journal of Sports Medicine, 59, 491-500.

[8] Wu, B. and Wu, W. (2023) Model-free cooperative optimal output regulation for linear discrete-time multi-agent systems using reinforcement learning. Mathematical Problems in Engineering, 6350647.

[9] Lazar, A., Pokhrel, P. and Das, S. (2026) Beyond accuracy: A comprehensive comparative study of gradient boosting versus tabular deep learning and explainability techniques for mixed-type tabular data models using SHAP and LIME. International Journal on Artificial Intelligence Tools, 35, 2640003.

[10] Yıldız, A.Y. and Kalayci, A. (2025) Gradient boosting decision trees on medical diagnosis over tabular data. Proceedings of the 2025 IEEE International Conference on AI and Data Analytics (ICAD), 1-8.

[11] Karagoz, G., Ozcelebi, T. and Meratnia, N. (2025) Systematic benchmarking of local and global explainable AI methods for tabular healthcare data. Proceedings of the World Conference on Explainable Artificial Intelligence, Springer Nature Switzerland, 337-358.

[12] Champahom, T., Banyong, C., Janhuaton, T., Se, C., Watcharamaisakul, F., Ratanavaraha, V. and Jomnonkwao, S. (2025) Deep learning vs. gradient boosting: Optimizing transport energy forecasts in Thailand through LSTM and XGBoost. Energies, 18, 1685.

[13] Ferreira, P., Martins, E., Silva, J. and Teixeira, P. (2025) Feature selection and XGBoost for enhanced intrusion detection: A comparative study across benchmark datasets. Proceedings of the 2025 13th International Symposium on Digital Forensics and Security (ISDFS), 1-6.

[14] Huang, J., Wu, B., Duan, Q., Dong, L. and Yu, S. (2025) A fast UAV trajectory planning framework in RIS-assisted communication systems with accelerated learning via multithreading and federating. IEEE Transactions on Mobile Computing.

[15] Kumar, R., Singhal, N. and Chhabra, A. (2025) Hybrid optimization algorithm with the combination of PSO and genetic algorithm for task scheduling in cloud computing. E-Learning and Digital Media, 20427530251331082.

[16] Nathiya, N., Rajan, C. and Geetha, K. (2025) A hybrid optimization and machine learning based energy-efficient clustering algorithm with self-diagnosis data fault detection and prediction for WSN-IoT application. Peer-to-Peer Networking and Applications, 18, 13.

[17] Wu, B., Huang, J. and Yu, S. (2026) 'X of Information' continuum: A survey on AI-driven multi-dimensional metrics for next-generation networked systems. IEEE Communications Surveys & Tutorials.

[18] Wu, B., Huang, J., Duan, Q., Dong, L. and Cai, Z. (2025) Enhancing vehicular platooning with wireless federated learning: A resource-aware control framework. IEEE/ACM Transactions on Networking, 33, 1-16.

[19] Hajihosseinlou, M., Maghsoudi, A. and Ghezelbash, R. (2025) A semi-supervised approach for mineral prospectivity mapping via weighted positive-unlabeled learning and tree-structured Parzen estimator for hyperparameter optimization. Ore Geology Reviews, 106783.

[20] Wu, B., Huang, J. and Duan, Q. (2025) FedTD3: An accelerated learning approach for UAV trajectory planning. Proceedings of the International Conference on Wireless Artificial Intelligent Computing Systems and Applications (WASA), 13-24.

[21] Zlobin, M. and Bazylevych, V. (2025) Bayesian optimization for tuning hyperparameters of machine learning models: A performance analysis in XGBoost. Computer Systems and Information Technologies, 1, 141-146.

[22] Khan, H., Khan, A., Villar, S., Alonso, L., Almaleh, A. and Al-Qahtani, A. (2025) A comparative study of optimized-LSTM models using tree-structured Parzen estimator for traffic flow forecasting in intelligent transportation. Computers, Materials & Continua, 83, 3369.

[23] Wu, B., Huang, J. and Duan, Q. (2025) Real-time intelligent healthcare enabled by federated digital twins with

AoI optimization. IEEE Network, 1.
[24] Foggetti, A., Nucci, F. and Papadia, G. (2025) Tuning metaheuristics with tree-structured Parzen estimator: A case study on scheduling. Journal of Artificial Intelligence and Autonomous Intelligence, 2, 293-321.
[25] Pan, D., Wu, B.-N., Sun, Y.-L. and Xu, Y.-P. (2023) A fault-tolerant and energy-efficient design of a network switch based on a quantum-based nano-communication technique. Sustainable Computing: Informatics and Systems, 37, 100827.
[26] Agrawal, S.K. (2026) Adaptive density-aware clustering of high-dimensional patient data in electronic health records. International Journal of Engineering Development and Research, 14, 361-367.