

# Exploring the Prediction of ICU Patient Readmission Using Machine Learning Techniques

Rishabh Sharad Pomaje

*Dept. of Electrical Engineering  
Stanford University  
rishabhp@stanford.edu*

Rutanshu Jhaveri

*ICME  
Stanford University  
rutanshu@stanford.edu*

Shruthi Shekar

*Dept. of Biomedical Data Science  
Stanford University  
scshekar@stanford.edu*

**Abstract**—ICU readmissions within 30 days pose significant clinical and operational burdens. Using data from MIMIC-III, we evaluate multiple predictive models and imputation techniques to predict all-cause 30-day readmission. Models trained on laboratory values, age, and ICU stay duration show that gradient-boosted trees achieve the strongest performance while capturing clinically meaningful patterns. Source code of the project is available on [GitHub](#).

## I. INTRODUCTION

Hospital ICU readmissions, especially those occurring shortly after ICU discharge, represent a major clinical and economic burden. In the United States, unplanned readmissions cost the Centers for Medicare and Medicaid Services (CMS) an estimated \$17-26 billion annually [1], simultaneously straining limited hospital capacity. Patients with conditions such as heart failure, frequent readmissions are associated with elevated mortality risk and increased psychological and social stress. Early identification of high-risk patients enables targeted interventions, discharge planning and timely follow-up.

We explore and evaluate different machine learning models to **predict the risk of all-cause 30-day readmission** using data from a patient's first ICU stay. We consider logistic regression, gradient-boosted decision trees, and Neural Network. The input to these models consists of numerous laboratory test measurements in addition to the age of the patient and the ICU length of stay (LoS). The models output the probability of readmission which is then used for classification.

## II. LITERATURE REVIEW

Increasing availability of Electronic Health Records (EHRs) has enabled a wide range of data-driven approaches in this field. Note that previous ML approaches to this problem rely on different datasets or adopt distinct cohort selection criteria, making direct comparisons

infeasible. As a result, we compare the qualitative merits and demerits of each approach. [2] derives a Readmission Risk Score (RRS) using logistic regression and features such as ALaRMS<sup>1</sup>, discharge history, and length of stay, but its linear structure limits its ability to capture nonlinear readmission patterns. [3] similarly uses regularized logistic regression on one year of temporal data only considering morbidly obese patients, restricting its generalizability.

[4] uses nationally representative Medicare data, focusing on data available at the first home healthcare visit, improving real-world clinical utility. However, despite employing gradient-boosted machines (GBMs), the model achieves only moderate discrimination (c-statistic = 0.66). [5] adopts an interpretable decision-tree approach for heart-failure patients, but its small sample size ( $n = 552$ ) limits the reliability and generalizability of its findings.

[6] uses time-series vital signs to train convolutional neural networks while accounting for asymmetric misclassification costs. Similarly, [7] employs longitudinal medical records to predict ICU readmission. Both DL approaches reduce the need for manual feature engineering but are computationally and data intensive.

Hospital readmission is an extensively studied subject. A comprehensive review of ML-approaches to the problem can be found in this survey paper [8]. To the best of our knowledge, very few papers consider data imputation techniques which we trial in this work.

## III. DATASET AND FEATURES

The MIMIC-III Clinical Database [9] contains de-identified EHR data recorded at Beth Israel Deaconess Medical Center in Boston, Massachusetts. It is a rich

<sup>1</sup>ALaRMS is an aggregated measure of clinical severity at admission

database that includes demographics, laboratory measurements, vital signs, diagnosis codes, clinical notes, additional patient information, etc. We focus on a set of laboratory test parameters (listed in [Table I](#)) and construct per-patient feature vectors using the mean, minimum, maximum, and standard deviation of each parameter. We also include patient age at admission and ICU LoS during the first visit. Lab features are typically available within 4-12 hours of admission, thereby supporting early readmission prediction. The target variable is defined as all-cause ICU readmission within 30 days.

Data extraction from MIMIC-III was a challenging component of the project, as the database comprises multiple interlinked tables that must be queried<sup>2</sup> systematically. We begin by constructing a well-defined cohort for model development. Our patient selection criterion are: (i) patients aged 18 years or older<sup>3</sup>; (ii) exclusion of patients who died during their ICU stay; and (iii) exclusion of patients who were eventually readmitted to the ICU but only after a hospital discharge, as these cases do not constitute true readmissions.

These criterion facilitate the curation of a coarsely refined initial dataset from MIMIC database. However, depending on the model being used, additional preprocessing steps are necessary and/or beneficial. We drop non-informative identifiers, such as the ICU stay ID, to prevent introducing unintended bias into the models.

Clinical records are often plagued with missing data, as not all tests are conducted on each patient. Further, they are also imbalanced data where one outcome is more frequent than the other(s). To understand how such imperfections influence the model performance, we tried and tested the impact of the following data-transformation processes:

- 1) *Imputation*: This involves substituting missing data using certain strategies. We considered a simple median strategy as a starting point that substitutes a missing entry in a feature vector with the median value of that feature across the training set.
- 2) *Class Balance*: Given that the rate of readmission was  $\sim 10.74\%$ , we explored a balancing technique: majority-class downsampling and loss-reweighting. Such a method prevents a model from overfitting to the dominant class.

To prevent any data leakage, the dataset was split into training, (optionally validation), and test sets, prior to any

processing. For example, the in case of imputation, the median across only the training examples was used to impute all examples. This ensured that the model never saw the test set until final evaluation. In addition, to ensure that the distribution of classes is maintained across all the sets, a *stratified* split was performed. Summary statistics for the final cohort are provided in [Table II](#).

#### IV. METHODS

Suppose  $x \in \mathbb{R}^d$  denotes a patient's feature vector and  $y \in \{0, 1\}$  is used to denote the class (readmission (1) and non-readmission (0)) of that patient.

##### A. Logistic Regression (Baseline Model)

Logistic regression (LogReg or LR) is a standard linear model widely used in clinical risk prediction (e.g., [2] uses LogReg weights to define a *Readmission Risk Score*) and other classification tasks. LogReg models the probability of readmission as

$$P(y = 1 | x) = \sigma(w^\top \tilde{x}), \quad (1)$$

where  $\sigma(z) = \frac{1}{1+e^{-z}}$  is the sigmoid function,  $w \in \mathbb{R}^{d+1}$  is the weight vector, and  $\tilde{x} = (1, x) \in \mathbb{R}^{d+1}$ . The model parameter vector  $w$  is obtained by minimizing a loss function, typically negative log-likelihood:

$$L(\theta) = - \sum_{i=1}^n \ell(\theta) \quad (2)$$

where,  $\ell(\theta) = y^{(i)} \log \sigma(w^\top \tilde{x}^{(i)}) + (1 - y_i) \log(1 - \sigma(w^\top \tilde{x}^{(i)}))$ . Despite its simplicity, logistic regression provides a strong interpretability advantage and serves as a useful baseline for evaluating more expressive models.

##### B. Gradient Boosted Decision Trees

Gradient Boosted Decision Trees (GBDT) capture complex distributions and feature relations that linear models cannot represent. We used eXtreme Gradient Boosting (XGBoost) algorithm [12], an ensemble method based on gradient-boosted decision trees. XGBoost constructs a model that cumulatively uses the prediction of multiple decision trees. The output of an ensemble,  $\hat{y}$ , is given as,

$$\hat{y} = \sum_{k=1}^K f_k(\mathbf{x}), \quad f_k \in \mathcal{F}, \quad (3)$$

where each  $f_k$  is a regression tree and  $\mathcal{F}$  denotes the space of all possible trees. XGBoost minimizes the following regularized objective:

$$\mathcal{L} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \left( \gamma T_k + \frac{1}{2} \lambda \|\mathbf{w}_k\|^2 \right), \quad (4)$$

<sup>2</sup>The list of queries we used is documented [here](#).

<sup>3</sup>Considering adult ICU patients ensures a more homogenous population as there are more complications with respect to treatment protocols and other factors involved in pediatric patients.

Anion-Gap	Bicarbonate	Calcium
Chloride	Creatinine	Glucose
Hematocrit	Hemoglobin	MCHC
MCV (Mean Corpuscular Volume)	Magnesium	PT (Prothrombin Time)
Phosphate	Potassium	RDW (Red Cell Distribution Width)
RBC Count	Sodium	Urea-Nitrogen
WBC Count	Age (years)	ICU-time (hrs)

TABLE I  
STATIC VITAL PARAMETERS AND PATIENT ATTRIBUTES USED TO GENERATE THE FEATURE VECTOR FOR EACH ICU STAY.

	Train Set	Validation Set	Test Set	Total
No. of Entries	14939	6403	9147	30489
No. of Readmissions	1605	688	983	3276
(% Readmissions)	~10.74	~10.74	~10.74	~10.74

TABLE II  
STATISTICS OF OUR COHORT SET

where  $l(\cdot)$  is a differentiable loss function of convex nature,  $T_k$  is the number of leaves in the  $k$ -th tree,  $\mathbf{w}_k$  are the leaf weights, and  $\gamma, \lambda$  are regularization parameters that penalize model complexity. This model is particularly well suited for classification tasks on structured, tabular data, such as the clinical variables used in this project.

CatBoost [13] is another gradient-boosted decision tree algorithm that minimizes a regularized loss function designed to handle categorical features natively. It uses an ordered boosting scheme to reduce overfitting and employs symmetric trees for computational efficiency.

### C. Neural network

TabNet [14] is a deep learning architecture tailored for tabular data through the use of sequential attention that enables the model to dynamically select which features to consider at each decision step. As a DL model, it reduces dependence on extensive feature engineering, an important advantage given that feature selection can substantially affect the performance of non-DL models.

## V. EXPERIMENTS, RESULTS, AND DISCUSSION

In this section, we discuss and analyze the experiments conducted as a part of this project.

**Performance Metrics:** The dataset under consideration is imbalanced, with only about 10.74% of patients experiencing readmission making accuracy unsuitable performance metric. We instead prioritize Area under Receiver-Operating Curve (AUROC) for model evaluation. AUROC represents the probability that given a positive and a negative example, the prediction model will rank the positive higher than the negative. An ideal classifier will have an AUROC score of 1, while a non-ideal classifier a score of 0. Figure 1 illustrates the Receiver Operating Curves for the best model

observed from each family of models. In addition, we report weighted (class-frequency weighted) and macro (equal-weighted) precision and recall metrics. Intuitively, precision measures the fraction of correctly classified examples classwise<sup>4</sup>. Recall, is the proportion of a class being correctly classified<sup>5</sup>. Table III summarizes the performance of all evaluated models.

### A. Logistic Regression Experiments

The simplicity of LogReg allowed rapid experimentation with preprocessing techniques. All LogReg experiments used the scikit-learn Python library [15]. Since LogReg cannot handle missing values, stricter<sup>6</sup> preprocessing was required, resulting in 9,774 usable examples (~ 13% readmission rate). Despite this limitation, the model remained a useful baseline because the data distribution was nearly preserved.

Using default scikit-learn settings, the LogReg (Base) achieved an AUROC of 0.670. LogReg (Balanced) employed LogReg with minority oversampling and majority undersampling combined with loss upweighting, yielding an AUROC of 0.668. The downscaling factor  $\kappa = \frac{\text{num\_minority}}{\text{num\_majority}} = \frac{1}{4}$  was set empirically. The performance across  $\kappa \in \{3, 4, 5\}$  was consistent. LogReg (Imputation) used data transformed through median imputation to address missing data and demonstrated an AUROC score of 0.709.

**Discussion:** The purpose of the Base experiment was to provide a performance baseline on our data.

<sup>4</sup>In an imbalanced dataset where the number of actual positives is very low, precision is useful as a metric.

<sup>5</sup>Recall on the other hand, is less and less useful as the data becomes more imbalanced.

<sup>6</sup>Any patient record with missing feature value had to be discounted.

Family	Model	Precision		Recall		AUROC
		Macro	Weighted	Macro	Weighted	
Linear Classifier (Baseline)	LogReg (Base)	0.54	0.78	0.50	0.86	<b>0.670 ± 0.025</b>
	LogReg (Balanced)	0.56	0.82	0.62	0.66	<b>0.668 ± 0.025</b>
	LogReg (Imputation)	0.67	0.85	0.51	<b>0.89</b>	<b>0.709 ± 0.017</b>
GBDT	XGBoost (Base)	0.60	0.83	0.52	<b>0.89</b>	<b>0.687 ± 0.017</b>
	XGBoost (HyperOpt)	0.63	0.85	0.60	0.87	<b>0.745 ± 0.015</b>
	XGBoost (Imputed)	0.64	0.85	0.59	0.88	<b>0.746 ± 0.015</b>
	CatBoost (Base)	0.75	<b>0.86</b>	0.51	<b>0.89</b>	<b>0.726 ± 0.016</b>
	CatBoost (HyperOpt)	<b>0.76</b>	<b>0.86</b>	0.51	<b>0.89</b>	<b>0.737 ± 0.016</b>
Neural Networks	TabNet (Base)	0.55	0.83	0.51	<b>0.89</b>	<b>0.648 ± 0.018</b>
	TabNet (Pretrained)	0.56	0.85	<b>0.64</b>	0.67	<b>0.680 ± 0.018</b>

TABLE III

PERFORMANCE METRICS OF DIFFERENT MACHINE LEARNING MODELS FOR ICU READMISSION PREDICTION. ALL THE MODELS, WHENEVER POSSIBLE, WERE OPTIMIZED FOR MAXIMUM AUROC SCORE. THE BEST VALUES ARE HIGHLIGHTED IN BLUE.

Data balancing techniques, upsampling minority class examples or downsampling majority class examples from existing records, did not improve the performance. A possible cause for this could be complex data distribution. Fresh data that adds new information is required to capture the complex data distribution. Consequently, we discontinued balancing synthetic balancing techniques on other models. In contrast, imputing missing values improved the performance. This may be attributed to increased amount of data available since we no longer discard incomplete entries. This incorporates more information about the distributions during training, leading to improved performance and generalization.

### B. XGBoost Experiments

All XGBoost experiments were conducted using the `xgboost` Python library [16]. Using the default hyperparameters, **XGB (Base)**, achieved an AUROC of 0.687. **XGB (HyperOpt)** was a model tuned using Optuna library [18], selecting parameter ranges guided by prior work and employing the Tree-Structured Parzen Estimator (TPE) algorithm, which has been shown to efficiently explore hyperparameter spaces [17], an AUROC score of 0.745 was observed. The optimal configuration found on TPE search included a learning rate of 0.00105, a maximum tree depth of 17, and a minimum child weight of 15.<sup>7</sup>. Lastly, applying median imputation followed by hyperparameter tuning, **XGB (Imputed)**, produced performance nearly identical to the tuned XGBoost model with similar hyperparameters.

The **CatBoost (Base)** model, trained with default hyperparameters, achieved an AUROC of 0.726, while the

tuned **CatBoost (HyperOpt)** model, optimized using a TPE-based HyperOpt search, achieved a slightly improved AUROC of 0.737.

**Discussion:** The ability of XGB to internally handle missing values and find the optimal splits factoring the missing data significantly improved the performance as even incomplete records could be used. The results also indicate the importance of hyperparameter tuning and incorporating information of imbalanced data (which was done by setting `scale_pos_weights` parameter using class frequencies) can enhance predictive performance. By tuning key parameters like tree depth the model can better adapt to the structure of our clinical dataset. For instance, increasing tree depth allows XGBoost to consider more features and interaction terms at each split, enabling it to capture the nonlinear relationships present. These adjustments allow the model to extract more “signal” from noisy, incomplete data.

We found CatBoost competitive while offering significantly shorter training times, making it a strong choice for rapid prototyping and baseline comparisons. Its performance without tuning was comparable to the tuned model emphasizing the capability of CatBoost with default parameters.

### C. TabNet

We observed sub-par performance with TabNet on our dataset. Using unsupervised pre-training and minor hyperparameter tuning, we obtained AUROCs of 0.648 and 0.680, respectively. Training TabNet was highly time-intensive, which limited our exploration of the model’s architecture and optimize its parameters.

**Discussion:** To check for **overfitting**, we compared training and testing errors and found them to be closely

<sup>7</sup>Please refer to the code output in GitHub for the complete list of optimal hyperparameters.

aligned across all models, with test errors only marginally higher. Regularization mechanisms (e.g.,  $L_2$  penalties in XGB and LR) effectively mitigated overfitting. TabNet showed the largest difference (but still modest), suggesting that neural network may benefit from additional training data. Limited exploration with respect to parameter tuning could also be one of the reasons. We discuss more solutions in section VII.

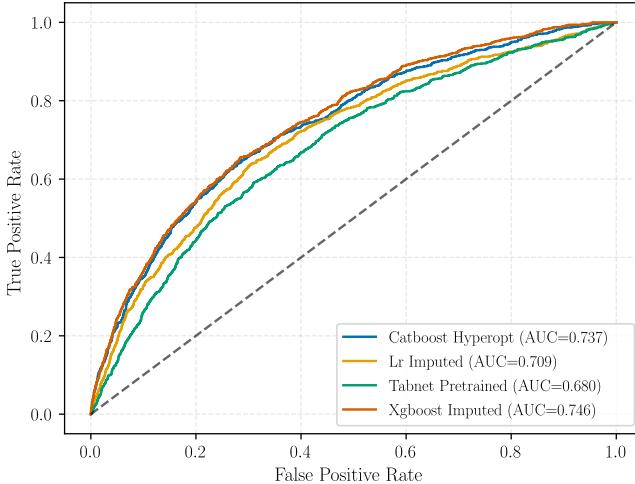


Fig. 1. A plot comparing the best receiver-operating curves for each model family.

## VI. INTERPRETATION OF BEST LEARNED MODEL

In this section, we analyze the top 20 feature importances from the best-performing model, XGB (Imputed) (Figure 2). The top 20 features correspond to inflammation, hematologic instability, renal stress, and acid-base or electrolyte imbalance-patterns. These factors are consistently associated with readmission.

RDW values and related hematologic indices rank highest. This is consistent with evidence linking RDW variation to systemic stress and adverse ICU outcomes [19]. Urea/BUN features reflect dehydration, catabolic stress, or impaired perfusion. These factors concur with predictors of early readmission [20]. Chloride and bicarbonate statistics highlight the importance of acid-base and fluid disturbances [21], [22] which correlate with mortality. Platelet count trends similarly signal ongoing inflammation/coagulopathy [23].

Overall, the feature profile aligns with medical literature. Importantly, the model prioritizes residual physiologic trends rather than isolated abnormal values, indicating that it detects sustained patterns of instability rather than noise. These trajectory-level signals reflect patterns clinicians often recognize informally but are not routinely quantified in current risk assessments.

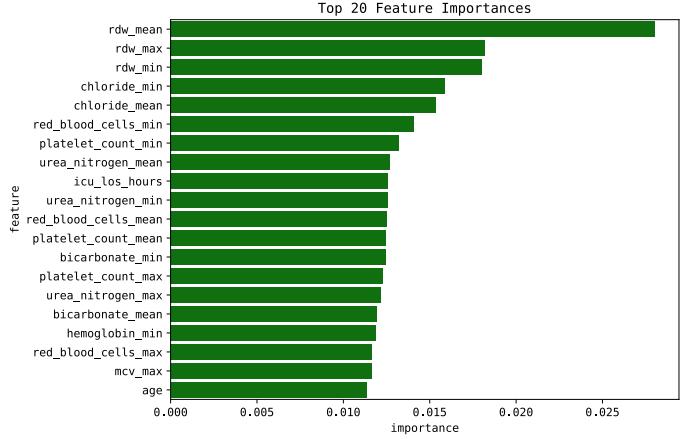


Fig. 2. Feature Importances observed in the best performing model - XGBoost with data imputation and hyperparameter tuning.

RDW, urea/BUN, platelet fluctuations, and electrolyte abnormalities consistently emerge as markers of short-term physiological instability. The model learned patients who appear clinically stable yet retain physiological signs of incomplete recovery. Thus the model captures meaningful biological patterns.

## VII. CONCLUSION AND FUTURE WORK

We evaluated different machine learning models and algorithms: Logistic Regression (Linear Classifiers), GBDT model (XGBoost and CatBoost), and a neural network (TabNet). The GBDT models demonstrated superior performance in comparison to others. Possible reasons for this include ability to handle data inconsistencies and tabular data being suited well to the algorithm. We also observed performance improvement with data imputation depending on the model.

Advanced imputation strategies, including KNN- or deep learning-based methods, should be explored in the future. Early feature-engineering experiments showed performance gains, suggesting value in augmenting the dataset with additional categorical variables (e.g., diagnostic or procedure codes) and information extracted from clinical notes for models such as FT-Transformer [24] and TabNet. Such attention-based deep learning architectures warrant further investigation to better leverage both structured and unstructured information.

## VIII. ACKNOWLEDGEMENT

We are thankful to our project mentor, Hercy Shen, for providing feedback and helpful suggestions. We also thank the entire CS229 faculty, TA and staff for a smooth conduction and support throughout the course.

## IX. CONTRIBUTIONS

- **Rishabh Sharad Pomaje** performed fine-grained data processing after database extraction. Explored, designed, implemented, and verified all Logistic Regression, XGBoost, CatBoost, and TabNet experiments. He analyzed and reasoned the performance of all models. He contributed to the writing, formatting, and presentation of all parts of the final report.
- **Rutanshu Jhaveri** did a detailed study of the database, and spent substantial efforts in designing and using Queries to get the initial coarse-data from MIMIC-III database. Notably, he also tried to get additional features such as disease codes and notes, but was constrained by computational limitations. He contributed to the initial implementation of XGBoost. He contributed to the literature review and the feature importance analysis section in the write-up.
- **Shruthi Shekar** introduced us to the problem. Helped in feature selection and qualitative (high-level) cohort data selection criterion, and qualitative field-specific analysis of the results and the corresponding section(s) in the write-up.

Everybody contributed to literature review.

## REFERENCES

- [1] M. Alvarado, B. Lahijanian, Y. Zhang, and M. Lawley, “Penalty and incentive modeling for hospital readmission reduction,” *Operations Research for Health Care*, vol. 36, p. 100376, 2023.
- [2] Y. P. Tabak, X. Sun, C. M. Nunez, V. Gupta, and R. S. Johannes, “Predicting readmission at early hospitalization using electronic clinical data: An early readmission risk score,” *Med. Care*, vol. 55, no. 3, pp. 267–275, mar 2017.
- [3] P. Brzan, Z. Obradovic, and G. Stiglic, “Contribution of temporal data to predictive performance in 30-day readmission of morbidly obese patients. *peerj*. 2017; 5: e3230.”
- [4] C. D. Jones, J. Falvey, E. Hess, C. R. Levy, E. Nuccio, A. E. Barón, F. A. Masoudi, and J. Stevens-Lapsley, “Predicting hospital readmissions from home healthcare in medicare beneficiaries,” *Journal of the American Geriatrics Society*, vol. 67, no. 12, pp. 2505–2510, 2019.
- [5] Y. Kang, M. D. McHugh, J. Chittams, and K. H. Bowles, “Utilizing home healthcare electronic health records for tele-homecare patients with heart failure: a decision tree approach to detect associations with rehospitalizations,” *CIN: Computers, Informatics, Nursing*, vol. 34, no. 4, pp. 175–182, 2016.
- [6] H. Wang, Z. Cui, Y. Chen, M. Avidan, A. B. Abdallah, and A. Kronzer, “Predicting hospital readmission via cost-sensitive deep learning,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1968–1978, 2018.
- [7] S. Barbieri, J. Kemp, O. Perez-Concha, S. Kotwal, M. Gallagher, A. Ritchie, and L. Jorm, “Benchmarking deep learning architectures for predicting readmission to the icu and describing patients-at-risk,” *Scientific reports*, vol. 10, no. 1, p. 1111, 2020.
- [8] Y. Huang, A. Talwar, S. Chatterjee, and R. R. Aparasu, “Application of machine learning in predicting hospital readmissions: a scoping review of the literature,” *BMC medical research methodology*, vol. 21, no. 1, p. 96, 2021.
- [9] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.
- [10] S. Davis, J. Zhang, I. Lee, M. Rezaei, R. Greiner, F. A. McAlister, and R. Padwal, “Effective hospital readmission prediction models using machine-learned features,” *BMC Health Services Research*, vol. 22, no. 1, p. 1415, Nov 2022. [Online]. Available: <https://doi.org/10.1186/s12913-022-08748-y>
- [11] A. Moerschbacher and Z. He, “Building prediction models for 30-day readmissions among icu patients using both structured and unstructured data in electronic health records,” in *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2023, pp. 4368–4373.
- [12] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. ACM, Aug. 2016, p. 785–794. [Online]. Available: <http://dx.doi.org/10.1145/2939672.2939785>
- [13] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” 2019. [Online]. Available: <https://arxiv.org/abs/1706.09516>
- [14] S. O. Arik and T. Pfister, “Tabnet: Attentive interpretable tabular learning,” 2020. [Online]. Available: <https://arxiv.org/abs/1908.07442>
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [16] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, Y. Li, J. Yuan, and D. Cortes, *xgboost: Extreme Gradient Boosting*, 2025, r package version 3.2.0.0. [Online]. Available: <https://github.com/dmlc/xgboost>
- [17] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” *Advances in neural information processing systems*, vol. 24, 2011.
- [18] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2623–2631.
- [19] G. Ephrem, “Red blood cell distribution width is a predictor of readmission in cardiac patients,” *Clinical Cardiology*, vol. 36, no. 5, pp. 293–299, 2013. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/clc.22116>
- [20] M. Pishgar, J. Theis, M. Del Rios, A. Ardati, H. Anahideh, and H. Darabi, “Prediction of unplanned 30-day readmission for icu patients with heart failure,” *BMC medical informatics and decision making*, vol. 22, no. 1, p. 117, 2022.
- [21] N. Sagar, S. Lohiya, and A. N. Sagar, “A comprehensive review of chloride management in critically ill patients,” *Cureus*, vol. 16, no. 3, 2024.
- [22] P. Yeh, Y. Pan, L. N. Sanchez-Pinto, and Y. Luo, “Hyperchloremia in critically ill patients: association with outcomes and prediction using electronic health record data,” *BMC Medical Informatics and Decision Making*, vol. 20, no. Suppl 14, p. 302, 2020.

- [23] C. T. Anthon, F. Pène, A. Perner, E. Azoulay, K. Puxty, A. Van De Louw, A. Barratt-Due, S. Chawla, P. Castro, P. Póvoa *et al.*, “Thrombocytopenia and platelet transfusions in icu patients: an international inception cohort study (plot-icu),” *Intensive care medicine*, vol. 49, no. 11, pp. 1327–1338, 2023.
- [24] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, “Revisiting deep learning models for tabular data,” 2023. [Online]. Available: <https://arxiv.org/abs/2106.11959>