Early Prediction of Sepsis Onset: Evaluating Supervised Machine Learning Techniques

Ava Berenji
Harvard-Westlake School, Los Angeles, CA

Abstract

Early detection of sepsis in intensive care units (ICUs) is critical for timely intervention and improved patient outcomes. This study evaluates the performance of supervised machine learning models in predicting sepsis onset using ICU electronic medical record (EMR) data. Four distinct feature engineering strategies were compared: Baseline Data (first recorded values), 24-Hour Baseline (first values within 24 hours of admission), 24-Hour Summary (baseline plus summary statistics), and 6-Hour Summary (summary statistics from the first 6 hours). This study provides insight into the impact of modeling choices on sepsis prediction performance and highlights practical considerations for the early detection of sepsis.

Keywords: sepsis prediction, machine learning, ICU, electronic health records (EHR), interpretable models, feature engineering, missing data imputation, comparative analysis

1 Introduction

1.1 Background

Sepsis is a life-threatening condition defined by the Sepsis-3 criteria as organ dysfunction resulting from a dysregulated host response to infection [1]. It remains a leading cause of global morbidity and mortality, particularly in intensive care units (ICUs). In 2017, sepsis affected an estimated 49 million people and contributed to 11 million deaths worldwide, accounting for nearly 20% of all global deaths [2]. Within the United States, it is implicated in one out of every three hospital deaths and imposes a substantial economic burden: U.S. hospital costs related to sepsis totaled $24 billion in 2013, the highest for any medical condition [3].

Timely recognition and treatment of sepsis are critical. Delays in therapy have been consistently associated with worsened outcomes; each hour of delay in antimicrobial administration increases mortality by approximately 4% [4]. Despite its urgency, sepsis diagnosis is challenging and lacks a definitive gold standard. In ICUs, diagnosis typically relies on clinical evaluation and the Sequential Organ Failure Assessment (SOFA) score, which assesses dysfunction in six organ systems. However, by not considering the complete feature set or temporal patterns embedded within EHR data, SOFA often misses crucial yet subtle signs of early physiological deterioration [1].

In recent years, machine learning (ML) models have shown promise in identifying sepsis risk by learning complex patterns from large electronic health record (EHR) datasets [5]. ML methods offer the potential to complement clinical judgment and outperform traditional scoring systems by integrating diverse physiologic variables to predict sepsis onset earlier and more accurately than traditional scoring systems, enabling earlier intervention. However, a key barrier to clinical implementation of ML models is the high variance between methodologies across different studies. Thus, this study aims to evaluate the potential and best practices for ML as a tool for early sepsis detection in the ICU.

1.2 Literature Review

Early detection of sepsis is a critical challenge in clinical care, as timely intervention significantly reduces mortality. Traditional rule-based scoring systems, such as SIRS [6], SOFA [7], and qSOFA [8], use fixed thresholds to flag at-risk patients. While clinically interpretable, these systems do not account for patient heterogeneity, neglect temporal trends, and fail to leverage the growing availability of electronic health record (EHR) data.

In recent years, machine learning (ML) has emerged as a promising alternative. ML models can learn complex, nonlinear relationships directly from EHR data, even in the presence of missing values and high dimensionality. Unlike static scores, ML models can be continuously retrained as new data become available, enabling adaptability and personalization. Numerous studies report that ML-based models outperform rule-based systems across key metrics such as sensitivity, specificity, and AUROC [9–11].

Many modern approaches focus on dynamic prediction, updating predictions hourly as new data arrive. This approach allows for the prediction several hours (often 4 to 8) prior to clinical diagnosis. A notable example is the 2019 PhysioNet/Computing in Cardiology Challenge, which asked participants to predict sepsis onset each hour using ICU time-series data [12].

Modeling techniques range from classical algorithms to deep learning. Ensemble tree models like Random Forest, XGBoost, and LightGBM consistently perform well, balancing predictive power and interpretability [13]. Gradient boosting, in particular, was central to effective models such as InSight [14], RoS [10], and all top five ranked PhysioNet challenge submissions [15–19]. To help ML models capture temporal dynamics, all these teams incorporated rolling window features, which summarize past data in a fixed-length window trailing the current hour, although window size varied among teams. Some studies combine multiple models to improve predictive performance. For instance, Mitra and Ashraf found that a simple ensemble, created from linear averaging of predictions from logistic regression, multilayer perceptron, XGBoost, and Random Forest, outperformed any individual model in AUROC [20]. Deep learning approaches, especially LSTMs, CNNs, and hybrid models, are increasingly common and excel at modeling raw time-series data, but often at the expense of interpretability [21].

Despite these advances, the field of sepsis prediction has considerable variability in methodology, which limits comparability across studies [22]. Differences exist in prediction forecasting (e.g. 4h, 6h, 12h, etc.), labeling criteria (e.g. Sepsis-2, Sepsis-3, SIRS, ICD-10 codes), dataset usage (e.g., MIMIC III, PhysioNet, private databases), and data preprocessing [13]. Missing values are handled through strategies ranging from exclusion or median/mean imputation to K-Nearest Neighbors (kNN), Multiple Imputation by Chained Equations (MICE), and forward/backward filling [23]. Tree-based models can natively handle missingness, and some studies, such as the fifth-place PhysioNet team, have found that imputation can degrade performance, opting instead for the native handling [19]. Others encode missingness directly as features, such as measurement frequency or binary missing flags [16–19].

Feature engineering strategies also vary widely. Common predictors include vital signs (e.g., respiratory rate, temperature, MAP, heart rate), labs (e.g., WBC, lactate, creatinine), and basic demographics (gender, age) [13]. Common hand-crafted features include summary statistics over sliding windows (e.g., mean, min, max, variance) [16–18], clinical ratios (e.g., shock index) [15], and first-order deltas [16, 18]. SHapley Additive exPlanations (SHAP) values are widely used for model interpretability and feature importance estimation [13].

In contrast to real-time prediction, some studies have explored static, early-window models that predict whether a patient will eventually develop sepsis. For instance, Faisal et al.'s CARS model used logistic regression on only first-recorded vitals and labs at Emergency Room (ER) admission, achieving AUROCs of 0.73–0.81 [24]. Similarly, Kuo et al. developed a shallow artificial neural network trained on one-hourly snapshots of EHR data from the ICU [PhysioNet 2019] to classify future sepsis onset up to 40 hours in advance, achieving AUROCs ranging from 0.76 to 0.82 despite intentionally preserving high missingness to simulate real-world ICU data [25]. While less complex and directly actionable, these studies demonstrate that meaningful early warning signals are present well in advance of sepsis diagnosis, highlighting the potential for training predictive models specifically on fixed, early-window data.

While ML models hold considerable promise for early sepsis prediction, there are many practical barriers to their clinical integration, including data availability, generalizability, overfitting, and notably, the high variance in methodologies across different studies [25, 26].

To address these gaps, the present study systematically benchmarks multiple interpretable ML models using static, early ICU data derived from the PhysioNet 2019 dataset. By comparing multiple preprocessing strategies and algorithm types, we seek to clarify the impact of different modeling decisions on model performance, thus informing the best practices for early sepsis prediction under a controlled environment.

2 Data Preprocessing

The data from this study were obtained from the PhysioNet/Computing in Cardiology Challenge 2019 [12], a publicly available collection of Electronic Medical Record (EMR) data from 40,336 ICU patients across two distinct hospitals: Beth Israel Deaconess Medical Center (Hospital A), and Emory University Hospital (Hospital B). These data, collected over the past decade with institutional review board approval, contain 40 clinical variables for each patient, binned into hourly values (see Table 11 for description of all variables, and Figures 4, 5, and 6 in the Appendix for histograms).

Variables include 8 vital signs, 26 laboratory values, and 6 demographic variables. In the full dataset, 2,932 patients (7.3%) developed sepsis during their ICU stay, while 37,404 (92.7%) did not.

The binary target variable, SepsisLabel, was defined based on the time of sepsis onset. For patients who developed sepsis during their ICU stay, the label was assigned a value of 1 for all times at and after the time six hours before the onset of sepsis. Otherwise, the label was 0. Without modifying the dataset, we corrected for this offset in code so that all sepsis onset times accurately reflected the true clinical timing.

The onset time (t_{sepsis}) followed Sepsis-3 clinical criteria, defined as the earlier of (i) clinical suspicion of infection and (ii) evidence of organ failure (a two-point increase in the SOFA score).

2.1 Cleaning and Feature Selection

Initial data cleaning involved the exclusion of 17 features based on redundancy, irrelevance, or excessive missingness.

Four features (EtCO2, Bilirubin_direct, Fibrinogen, TroponinI) were removed due to excessive missingness, having never been recorded in over 75% of patients. Hematocrit (Hct) was excluded due to redundancy with Hemoglobin (Hgb), which was retained as the more informative predictor.

An additional 11 features (Base Excess, Oxygen Saturation (SaO2), Aspartate Aminotransferase (AST), Alkaline Phosphatase, Calcium, Chloride, Magnesium, Phosphate, Potassium, Partial Thromboplastin Time (PTT), and Total Bilirubin (Bilirubin_total)) were also excluded based on clinical redundancy, irrelevance to early-onset sepsis prediction, or high missingness.

Furthermore, the binary indicators for Unit1 and Unit2 were combined into a single categorical variable, Unit. Patients in Unit 1 were labeled '1', while those in Unit 2 were labeled '2', and instances where both were missing were assigned '0' to denote an unknown unit.

Finally, three time-dependent identifiers (Hour, ICULOS, HospAdmTime) were removed. This yielded a baseline feature set of 21 predictors: HR, O2Sat, Temp, SBP, MAP, DBP, Resp, FiO2, Platelets, Hgb, WBC, Lactate, Creatinine, BUN, Glucose, pH, PaCO2, HCO3, Age, Gender, and Unit.

2.2 Feature Engineering

To evaluate the impact of feature engineering choices on model performance, we constructed four distinct feature sets, each characterized by different training/prediction windows and engineered features, as summarized in Table 1.

Table 1: Summary of Feature Engineering Strategies

Strategy	**Time Window**	**Features**	**Population / Prediction Horizon**
1. Baseline Data	First recorded value (restricted to pre-onset data)	21 clinical variables	All patients; predict sepsis during entire post-baseline ICU stay
2. 24-Hour Baseline	First 24 hours of ICU admission	21 clinical variables	All patients with $\geq$ 25h of data, septic patients must have $t_{\text{sepsis}} > 24$h; predict sepsis after 24h
3. 24-Hour Summary	First 24 hours of ICU admission	3 demographics; baseline, count, delta, range of 18 clinical variables	Same as Strategy 2.
4. 6-Hour Summary	First 6 hours of ICU admission	3 demographics; Baseline, count, delta, SD of 7 vitals; Baseline, count of 11 lab values	Septic patients with $7 \leq t_{\text{sepsis}} \leq 24$ and $\geq$ 7h of data, nonseptic patients with $\geq$ 24h of data; predict onset within the first 7 to 24h

Baseline Data. This strategy used only the first recorded value for each variable, restricted to pre-onset measurements to avoid temporal leakage. When trained using listwise deletion (removing any patient with a missing value), only 5,831 patients remained (4,996 non-septic (85.7%), 835 septic (14.3%)). The listwise deletion approach allows us to see how the models would perform on the subset of patients with complete data, though we acknowledge this is not a clinically realistic scenario.

24-Hour Baseline and Summary. These strategies incorporated all data from the first 24 hours of ICU admission. To ensure that septic patients had sufficient pre-onset data, only those with $t_{\text{sepsis}} > 24$ hours were included. After the inclusion criteria, 30,983 patients remained (29,247 nonseptic (94.4%), 1,736 septic (5.6%)). The 24-Hour Summary strategy added engineered features: baseline value, count (number of measurements), delta (last minus first value), and range, helping capture trends in patient physiology rather than a single snapshot in time.

6-Hour Summary. This strategy focused on very early prediction, aiming to predict sepsis within the first 7 to 24 hours after admission (a window excluded by the 24-hour models). Patients needed at least 7 hours of data available to be included, and nonseptic patients required at least 24 hours of ICU data to ensure definitive labeling. This strategy included 31,725 patients (30,572 nonseptic (96.4%), 1,153 septic (3.6%)).

Features were derived from the first 6 hours of ICU data. For vital signs, we computed baseline (first recorded value), count (number of measurements), delta (last minus first recorded value), and standard deviation. For laboratory variables, only baseline and count were used due to infrequent early draws. Standard deviation was chosen over range to better capture variability during the short window.

3 Methods

This study employed supervised machine learning to predict sepsis onset using ICU electronic medical record data. Four modeling strategies were evaluated: Baseline Data (first recorded values), 24-Hour Baseline (first recorded values $\leq$ 24h from admission), 24-Hour Summary (24-Hour Baseline with summary statistics added), and 6-Hour Summary (data from the first 6 hours, with summary statistics).

3.1 Model Types

We implemented seven classifiers: Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), k-Nearest Neighbors (kNN), Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR). Artificial Neural Networks (ANNs) were initially considered but excluded due to comparatively poor performance and high training time.

For datasets with native missingness (unimputed), only tree-based models (RF, XGBoost, LightGBM) were evaluated due to their ability to handle missing values without imputation.

3.2 Data Splitting and Validation

Each dataset was randomly split into 80% training and 20% testing sets. Model selection and hyperparameter tuning were performed via 5-fold cross-validation on the training set. Only SVM radial kernel results were reported, as that was consistently the best-performing kernel. Logistic regression used the logit link function. The value of k for the kNN model was optimized independently within each fold. Final performance metrics were reported on the held-out test set. No data from the test set was used during model training, feature selection, or imputation to prevent data leakage.

3.3 Missing Value Handling and Scaling

Because many models required complete data, we imputed missing values using the median of each feature, calculated from the training set only to avoid data leakage.

For models sensitive to feature scale (kNN, SVM, Logistic Regression), continuous predictors were standardized using Z-score normalization with the training set's mean and standard deviation to avoid data leakage.

We also evaluated tree-based models (Random Forest, XGBoost, LightGBM) with their native missingness handling to assess whether missingness patterns themselves were informative.

Baseline models were initially trained with listwise deletion to explore model performance on a fully observed reference dataset, though this is not clinically practical and ignores the fact that missingness can itself be predictive. This benchmark allows comparison with imputed and native-missingness approaches.

3.4 Handling Class Imbalance

The raw dataset was highly imbalanced, with nonseptic patients far outnumbering septic ones. Thus, the training set was balanced using random undersampling at a 1:1 septic-to-nonseptic ratio. The test set remained imbalanced to reflect clinical prevalence. Class weights were tested but not adopted, as they did not generalize well across models.

3.5 Evaluation Metrics

All models were evaluated using area under the receiver operating characteristic curve (AUROC), the area under the precision-recall curve (AUPRC), sensitivity, specificity, precision, F1-score, and accuracy. Confusion matrices were calculated at a threshold of 0.5.

3.6 Theoretical Framework

This study evaluated seven commonly used supervised learning algorithms: Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), k-Nearest Neighbors (kNN), Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR). A detailed overview of the theory underlying these classifiers, including their mathematical formulations, can be found in standard textbooks such as [27] and [28].

3.6.1 Random Forest

The Random Forest (RF) algorithm is an ensemble method based on bagging (bootstrap aggregating), where multiple decision trees are trained on different bootstrap samples: random samples of the training data with replacement. At each node split, a random subset of features is considered to reduce the correlation between trees. For binary classification, the final prediction is made by majority vote across all trees, helping reduce model variance and overfitting.

Before training, three main hyperparameters must be set: minimum node size, number of trees, and number of variables sampled at each split.

Feature importance is commonly assessed using two metrics. Mean Decrease in Impurity (MDI) measures how much impurity decreases when the given feature is excluded. Mean Decrease in Accuracy (MDA) measures the drop in accuracy when values of a feature are randomly permuted.

3.6.2 XGBoost

Extreme Gradient Boosting (XGBoost) is an ensemble method that implements gradient boosting with performance optimizations. It builds an additive model by sequentially fitting decision trees, where each tree is trained to correct the residual errors of the current ensemble. Unlike traditional boosting algorithms that only use the first derivative (gradient) of a loss function, XGBoost also considers the second derivative (curvature). The prediction function is updated iteratively:

$$\hat{f}(x) = \hat{f}_0(x) + \sum_{b=1}^{B} \lambda \hat{f}_b(x)$$

where $\hat{f}_0(x)$ is the initial estimate (often set to a constant), $\hat{f}_b(x)$ is the fitted weak-learner at iteration b, and λ is the learning (shrinkage) rate that controls how much each new tree contributes to the final model. The final model $\hat{f}(x)$ is the sum of these weighted learners.

Key hyperparameters include: number of trees B, the number of splits k, and the learning rate λ.

3.6.3 LightGBM

Light Gradient Boosting Machine (LightGBM) is an ensemble method that also implements gradient boosting, designed for faster training and lower memory usage on large datasets. Like XGBoost, it builds an additive model by sequentially training decision trees to reduce residual errors. The prediction function is updated iteratively:

$$\hat{f}(x) = \hat{f}_0(x) + \sum_{b=1}^{B} \lambda \hat{f}_b(x)$$

where $\hat{f}_0(x)$ is the initial estimate, $\hat{f}_b(x)$ is the fitted weak learner at iteration b, and λ is the learning (shrinkage) rate.

LightGBM introduces two key innovations to improve speed and scalability: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS accelerates training by concentrating learning on the data points the model struggles with the most. EFB reduces dimensionality by bundling mutually exclusive features (usually binary), enabling efficient learning on high-dimensional datasets.

LightGBM grows trees leaf-wise rather than level-wise, selecting the leaf with the highest loss reduction at each step. This can lead to deeper, more complex trees that improve accuracy but may increase the risk of overfitting.

Key hyperparameters include: number of trees B, the number of splits k, and the learning rate λ.

3.6.4 Naive Bayes

The Naive Bayes (NB) algorithm is a probabilistic classifier based on Bayes' Theorem. For a response variable Y and predictors $X = (X_1, X_2, \cdots, X_k)$, Bayes' Theorem yields:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}.$$

The model "naively" assumes that all features are conditionally independent given the class label, allowing $P(X|Y)$ to factor as

$$\prod_{i=1}^{k} P(X_i|Y).$$

For each observation, the model computes the conditional probability of each class given the predictors and assigns the observation to the class with the highest probability.

The probability $P(Y = y)$ is estimated from the frequency of the classes in the training set. For categorical predictors, $P(X_i = x|Y = y)$ is estimated as empirical proportions, whereas for continuous predictors, the conditional distribution is assumed to be normal (Gaussian), with the mean and variance estimated from the data.

3.6.5 Support Vector Machine

The Support Vector Machine (SVM) algorithm is a margin-based classifier that aims to find the optimal hyperplane that separates data points of different classes. For binary classification, SVM seeks to maximize the margin: the distance between the closest data points (called *support vectors*) and the separating hyperplane. A larger margin generally improves generalizability to unseen data.

Mathematically, for input features $\mathbf{x}_i$ and labels $y_i \in \{-1, +1\}$, the linear SVM solves the primal formulation:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \quad \forall i,$$

where $\mathbf{w}$ is the normal vector to the hyperplane and b is the bias term.

To handle these constraints, SVM introduces Lagrange multipliers $\alpha_i \geq 0$, leading to the alternate dual formulation:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle,$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \quad \alpha_i \geq 0.$$

This formulation depends only on the inner products of the training points, making it straightforward to extend to nonlinear decision boundaries.

Nonlinear relationships between features and class labels can be captured using different kernels, which replace the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ in the dual formulation. In this study, the radial basis function (RBF) kernel was used:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

where γ controls the influence of each training example.

In practice, perfect separation is rare, so the soft-margin SVM introduces slack variables that allow points to violate the margin. A regularization parameter C controls how heavily such violations are penalized.

Key hyperparameters include the penalty parameter C and the kernel coefficient γ. These determine the trade-off between model complexity and classification error.

3.6.6 k-Nearest Neighbor

The K-Nearest Neighbors (kNN) algorithm is a simple, non-parametric method. For binary classification, it assigns a label to a new observation based on the majority class among its k nearest neighbors in the training set.

To classify a new observation, the algorithm computes the Euclidean distance between the observation and all training points. For the feature vectors $x = (x_1, x_2, \ldots, x_d)$ and $y = (y_1, y_2, \ldots, y_d)$, the Euclidean distance is defined as:

$$distance(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_d - y_d)^2}.$$

The k nearest neighbors (those points with the smallest distances) are identified, and the majority class among them is assigned to the new point. To avoid ties, k is typically chosen to be an odd number.

The key hyperparameter is the number of neighbors k.

3.6.7 Generalized Linear Models

The logistic model is a generalized linear model (GLM) commonly used for binary classification, where the response variable $y \in \{0, 1\}$. Each GLM models the probability $\pi = P(y = 1)$ as a function of predictor variables $x_1, x_2, \cdots, x_k$, using a different link function to relate the linear predictor $\beta_0 + \beta_1 + \cdots + \beta_k x_k$ to the probability π. The logistic model uses the logit link:

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k.$$

Here, $\pi/(1 - \pi)$ represents the odds of $y = 1$. Solving for π, we can write the fitted logistic regression model as:

$$\hat{\pi} = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_k x_k)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_k x_k)}.$$

4 Applications and Results

This section presents the performance metrics of various machine learning models across different data preprocessing and feature engineering strategies. The results are organized by data preparation approach: Baseline Data, 24-hour Baseline, 24-hour Summary, and 6-hour Summary. Performance is evaluated using Accuracy, Area Under the Receiver Operating Characteristic Curve (AUROC), Area Under the Precision-Recall Curve (AUPRC), Sensitivity, Specificity, Precision, and F1 Score.

4.1 Baseline Data

This strategy utilizes only the first available value for each feature (restricted to pre-onset data). The tables compare three different techniques for handling missing values: Listwise Deletion, Median Imputation, and Retained Missing Values (for tree-based models).

Table 2: Model Performance on Baseline Data Using Listwise Deletion

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.5666	**0.6864**	**0.2449**	**0.7437**	0.5370	**0.2117**	**0.3296**
XGBoost	0.5910	0.6553	0.2204	0.6264	0.5851	0.2014	0.3045
LightGBM	0.5947	0.6659	0.2191	0.6664	0.5836	0.2110	0.3195
kNN	0.6280	0.6532	0.2125	0.5832	0.6355	0.2109	0.3096
Naive Bayes	**0.6880**	0.6524	0.2067	0.4251	**0.7320**	0.2100	0.2809
SVM (radial)	0.5689	0.6761	0.2287	0.7090	0.5454	0.2069	0.3202
LR (logit)	0.5910	0.6553	0.2204	0.6264	0.5851	0.2014	0.3045

Table 2 presents the performance of models using listwise deletion. Random Forest demonstrated the highest AUROC (0.6864), AUPRC (0.2449), Sensitivity (0.7437), Precision (0.2117), and F1 Score (0.3296). Naive Bayes recorded the highest Accuracy (0.6880) and Specificity (0.7320), albeit with lower Sensitivity.

Table 3: Model Performance on Baseline Data Using Median Imputation

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6844	0.7743	0.2129	**0.7404**	0.6800	0.1536	0.2544
XGBoost	0.7106	0.7730	0.2140	0.6969	0.7116	**0.1593**	0.2593
LightGBM	0.7042	**0.7750**	**0.2144**	0.7147	0.7034	0.1589	**0.2599**
kNN	0.7445	0.6964	0.1530	0.5085	0.7630	0.1440	0.2244
Naive Bayes	**0.7814**	0.6916	0.1365	0.4294	**0.8090**	0.1499	0.2221
SVM (radial)	0.7020	0.7261	0.1667	0.6357	0.7072	0.1455	0.2368
LR (logit)	0.6563	0.6693	0.1400	0.5883	0.6616	0.1199	0.1992

Table 3 details the performance of models using median imputation. The decision tree models (Random Forest, XGBoost, and LightGBM) showed very similar performance in terms of AUROC (around 0.775) and AUPRC (around 0.21). Random Forest had the highest sensitivity (0.7404) by a large margin, though its specificity was lower than the gradient boosting models. XGBoost maintained the highest Precision (0.1593), and Naive Bayes maintained the highest Accuracy (0.7814) and Specificity (0.8090).

Table 4: Model Performance on Baseline Data Using Retained Missing Values

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6624	0.7104	0.1714	0.6463	0.6637	0.1310	0.2178
XGBoost	**0.7249**	**0.7992**	**0.2727**	**0.7323**	**0.7243**	**0.1725**	**0.2791**
LightGBM	0.7170	0.7986	**0.2727**	0.7320	0.7158	0.1680	0.2733

Table 4 highlights the performance of tree-based models that can natively handle missing values. XGBoost outperformed the other models across nearly all metrics, achieving the highest Accuracy (0.7249), AUROC (0.7992), AUPRC (0.2727), Sensitivity (0.7323), Specificity (0.7243), Precision (0.1725), and F1 Score (0.2791). LightGBM also performed very well, matching XGBoost's AUPRC, while Random Forest lagged behind both XGBoost and LightGBM on all metrics. This difference is expected, as Random Forest typically doesn't handle missing values as effectively as gradient-boosting methods like XGBoost and LightGBM without first undergoing explicit imputation.

4.2 24h Baseline

Table 5: Model Performance on 24-Hour Baseline Data Using Median Imputation

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6601	**0.7156**	**0.1320**	**0.6503**	0.6607	0.1021	0.1764
XGBoost	0.6725	0.7068	0.1316	0.6317	0.6750	**0.1063**	**0.1819**
LightGBM	0.6657	0.7080	0.1233	0.6469	0.6668	0.1033	0.1782
kNN	0.7480	0.6580	0.1048	0.4562	0.7653	0.1035	0.1688
Naive Bayes	**0.7772**	0.6606	0.0989	0.3917	**0.8000**	0.1043	0.1646
SVM (radial)	0.6873	0.6864	0.1154	0.5852	0.6934	0.1017	0.1733
LR (logit)	0.6449	0.6469	0.1086	0.5869	0.6486	0.0926	0.1599

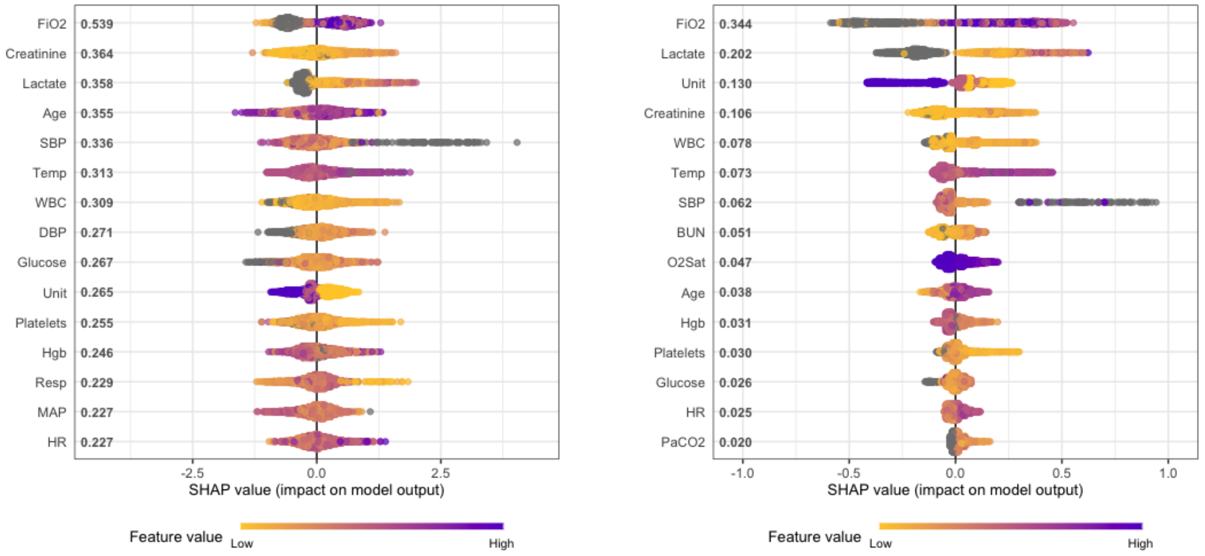
Table 5 summarizes the performance of models utilizing median imputation for the 24-hour baseline features. Random Forest achieved the highest AUROC (0.7156), AUPRC (0.1320), and Sensitivity (0.6503). XGBoost obtained the highest Precision (0.1063) and F1 Score (0.1819). Naive Bayes recorded the highest Accuracy (0.7772) and Specificity (0.8000), although its AUROC and AUPRC were much lower relative to the tree models.

Table 6: Model Performance on 24-Hour Baseline Data Using Retained Missing Values

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6411	0.6620	0.1008	0.5841	0.6444	0.0889	0.1542
XGBoost	**0.6801**	**0.7326**	**0.1407**	**0.6744**	**0.6804**	**0.1114**	**0.1911**
LightGBM	0.6764	0.7304	0.1376	0.6691	0.6768	0.1095	0.1881

When tree models retained missing values for the 24-hour baseline features (Table 6), both XGBoost and LightGBM demonstrated strong, similar performance. XGBoost achieved the highest Accuracy (0.6801), AUROC (0.7326), AUPRC (0.1407), Specificity (0.6804), Precision (0.1114), and F1 Score (0.1911). LightGBM was slightly behind across most metrics, whereas Random Forest consistently lagged behind both LightGBM and XGBoost.

4.2.1 Feature Importance Analysis



(A) XGBoost: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

(B) LightGBM: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

Figure 1: Global SHAP feature importance visualizations for (A) XGBoost and (B) LightGBM models trained on 24-hour baseline features. Higher absolute SHAP values indicate greater contribution to the model's predictions.

Global SHAP beeswarm plots (Figure 1A-B) show that baseline FiO2, Creatinine, Lactate, and Unit consistently ranked among the most influential predictors across both XGBoost and LightGBM models.

Predicted sepsis risk increased with higher FiO2, and, to a lesser extent, with higher creatinine and lactate levels. A high Unit value, which indicated admission to the Surgical ICU, strongly decreased predicted sepsis risk.

4.3 24h Summary

Table 7: Model Performance on 24-Hour Summary Data Using Median Imputation

Model	**Accuracy**	**AUROC**	**AUPRC**	**Sensitivity**	**Specificity**	**Precision**	**F1 Score**
Random Forest	0.6927	0.7446	0.1644	**0.6749**	0.6938	0.1188	0.2020
XGBoost	0.7003	0.7457	0.1576	0.6674	0.7024	0.1206	0.2042
LightGBM	0.7039	**0.7473**	**0.1646**	0.6551	0.7069	0.1202	0.2031
kNN	**0.7753**	0.7009	0.1336	0.4888	0.7928	**0.1262**	0.2005
Naive Bayes	0.7711	0.6578	0.1020	0.3885	**0.7945**	0.1038	0.1637
SVM (radial)	0.7067	0.7438	0.1567	0.6576	0.7097	0.1217	**0.2054**
LR (logit)	0.7139	0.7372	0.1578	0.6311	0.7190	0.1207	0.2025

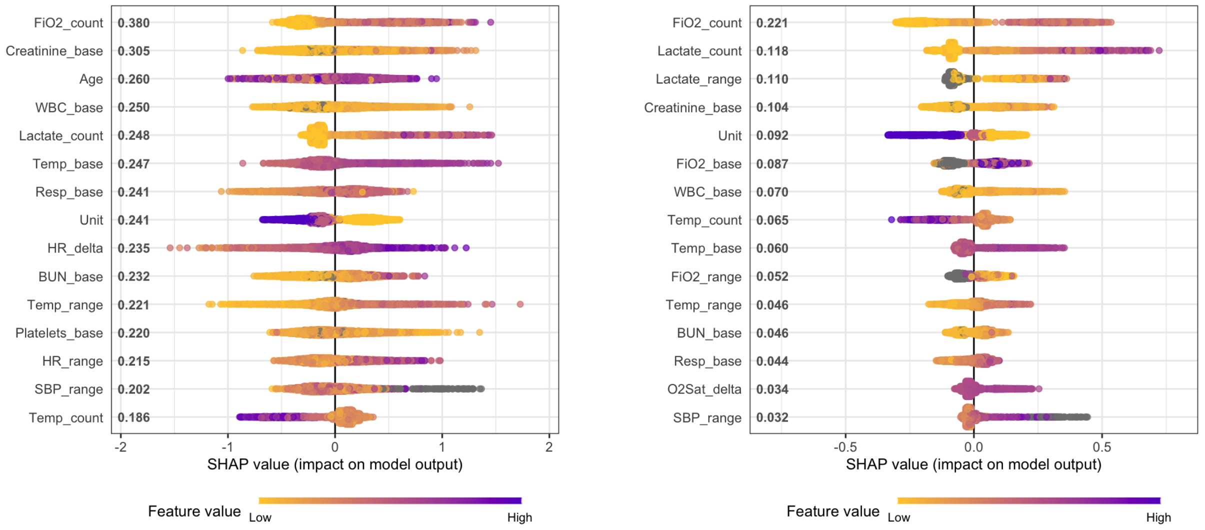
Table 7 presents the performance of models using median imputation. kNN achieved the highest Accuracy (0.7753) and Precision (0.1262), while Naive Bayes showed the highest Specificity (0.7945). Random Forest achieved the highest Sensitivity (0.6749), while LightGBM yielded the highest AUROC (0.7473) and AUPRC (0.1646). SVM achieved the highest F1 Score (0.2054) and overall performed similarly to many of the decision tree models. Logistic regression (LR) also performed competitively with an AUROC of 0.7372 and an F1 Score of 0.2025.

Table 8: Model Performance on 24-Hour Summary Data Using Retained Missing Values

Model	**Accuracy**	**AUROC**	**AUPRC**	**Sensitivity**	**Specificity**	**Precision**	**F1 Score**
Random Forest	0.7041	0.7445	0.1645	0.6594	0.7068	0.1210	0.2044
XGBoost	0.7011	0.7501	0.1634	**0.6735**	0.7029	**0.1217**	**0.2061**
LightGBM	**0.7051**	**0.7508**	**0.1658**	0.6613	**0.7078**	0.1215	0.2052

Table 8 presents the performance of tree-based models when natively handling missing values. Although XGBoost and LightGBM had very similar performance, LightGBM was the top-performing model in most metrics, achieving the highest Accuracy (0.7051), AUROC (0.7508), AUPRC (0.1658), and Specificity (0.7078). XGBoost achieved the best Sensitivity (0.6735), Precision (0.1217), and F1 Score (0.2061). Random Forest performed slightly lower than both LightGBM and XGBoost in this scenario.

4.3.1 Feature Importance Analysis



(A) XGBoost: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

(B) LightGBM: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

Figure 2: Global SHAP feature importance visualizations for (A) XGBoost and (B) LightGBM models trained on 24-hour summary features. Higher absolute SHAP values indicate greater contribution to the model's predictions.

Global SHAP beeswarm plots (Figure 2A-B) show that FiO2 count, baseline Creatinine, and Lactate count consistently ranked among the most influential predictors across both XGBoost and LightGBM models. Predicted sepsis risk increased with higher FiO2 and Lactate measurement counts, and, to a lesser extent, with higher creatinine levels.

4.4 6h Summary

Table 9: Model Performance on 6-Hour Summary Data Using Median Imputation

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6499	**0.7088**	0.0819	**0.6523**	0.6498	0.0676	0.1225
XGBoost	0.6571	0.7038	0.0831	0.6516	0.6573	**0.0689**	**0.1246**
LightGBM	0.6584	0.7001	**0.0841**	0.6474	0.6589	0.0687	0.1241
kNN	0.7423	0.6499	0.0631	0.4428	0.7540	0.0656	0.1143
Naive Bayes	**0.7564**	0.6601	0.0647	0.4259	**0.7692**	0.0670	0.1157
SVM (radial)	0.6522	0.7025	0.0839	0.6498	0.6523	0.0679	0.1229
LR (logit)	0.6599	0.6812	0.0780	0.6091	0.6618	0.0654	0.1181

Table 9 summarizes the performance of all models when median imputation was applied. Naive Bayes achieved the highest Accuracy (0.7564) and Specificity (0.7692). The decision tree models all showed similar performance. Random Forest recorded the best AUROC (0.7088) and Sensitivity (0.6523), while

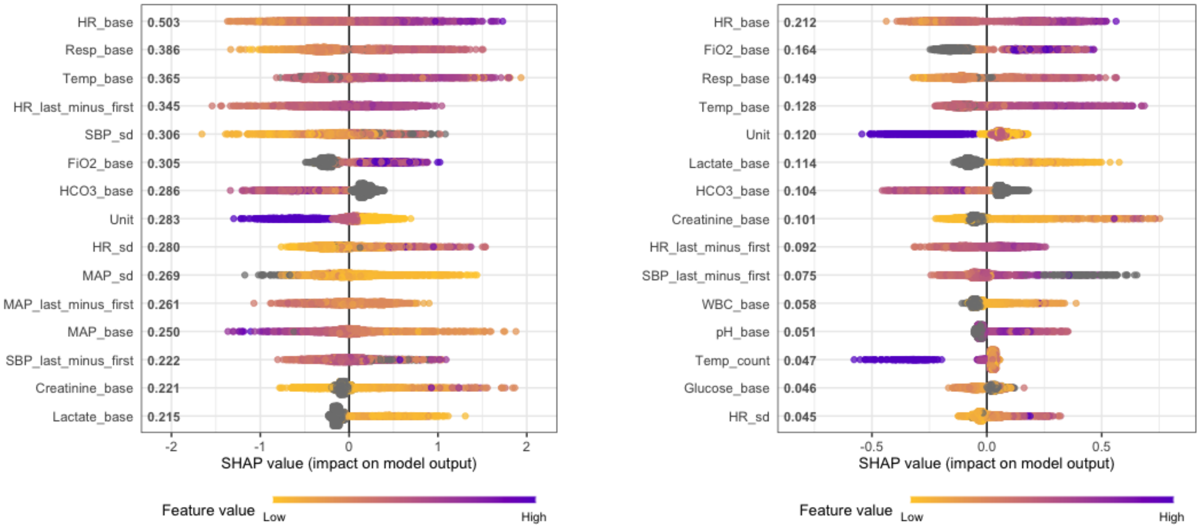
LightGBM achieved the top AUPRC (0.0841). XGBoost achieved the highest Precision (0.0689) and F1 Score (0.1246) by a small margin.

Table 10: Model Performance on 6-Hour Summary Data Using Retained Missing Values

Model	Accuracy	AUROC	AUPRC	Sensitivity	Specificity	Precision	F1 Score
Random Forest	0.6400	0.6979	0.0794	**0.6472**	0.6397	0.0653	0.1186
XGBoost	**0.6598**	0.6965	**0.0830**	0.6263	**0.6610**	0.0670	0.1210
LightGBM	0.6571	**0.7000**	0.0827	0.6360	0.6579	**0.0674**	**0.1219**

Table 10 presents the performance of the tree-based models capable of natively handling missing values. XGBoost achieved the highest Accuracy (0.6598), AUPRC (0.0830), and Specificity (0.6610). LightGBM recorded the best AUROC (0.7000), Precision (0.0674), and F1 Score (0.1219). Random Forest attained the highest Sensitivity (0.6472).

4.4.1 Feature Importance Analysis



(A) XGBoost: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

(B) LightGBM: Global SHAP beeswarm plot showing the top 15 features ranked by mean SHAP value.

Figure 3: Global SHAP feature importance visualizations for (A) XGBoost and (B) LightGBM models trained on 6-hour summary features. Higher absolute SHAP values indicate greater contribution to the model's predictions.

Global SHAP beeswarm plots (Figure 3A-B) show that baseline HR, respiratory rate, FiO2, and temperature consistently ranked among the most influential predictors across both XGBoost and LightGBM models. Predicted sepsis risk increased with higher baseline HR, respiratory rate, FiO2, and temperature. Notably, patients for whom FiO2 was never recorded in the first six hours were less likely to be septic.

5 Discussion/Conclusion

5.1 Major Findings and Interpretation

This study systematically benchmarked seven machine learning models and four feature-engineering strategies for early sepsis prediction using ICU electronic medical record data. Gradient boosting models (XGBoost and LightGBM) consistently outperformed all other classifiers across most meaningful metrics, particularly when trained with 24-hour summary features. These models effectively captured complex non-linear relationships and leveraged informative missingness patterns, achieving AUPRC values two to four times higher than the baseline prevalence.

SVM (radial kernel) and logistic regression were occasionally competitive when summary features were included, performing comparably to tree-based models in AUROC and F1-score for certain datasets (e.g., 24-hour summary with median imputation; Table 7). This demonstrates that simpler models can perform reasonably well when provided with richer feature engineering.

Missing data handling emerged as a critical driver of performance. On baseline data, listwise deletion yielded stronger AUPRC, precision, and F1 score than median imputation, although this is likely due to the significant differences in class distribution. Median imputation improved AUROC, sensitivity, and specificity for most models. For tree-based models, retaining missing values natively was often far superior to median imputation, confirming that missingness patterns themselves can encode clinically relevant information.

Feature engineering choices also strongly influenced model performance. Incorporating summary statistics over the first 24 hours (counts, deltas, and ranges) substantially improved discriminative ability compared to single baseline measurements. These findings are consistent with previous PhysioNet Challenge results, where top teams similarly emphasized temporal features. Models using only six-hour data performed worse than those using 24-hour data, illustrating the trade-off between earlier predictions and predictive power.

The SHAP analysis provided additional interpretability, highlighting which features most influenced the models' predictions. For the 24-hour datasets, FiO2, lactate, and creatinine consistently had the largest impact, whereas in the 6-hour summary, HR, Resp, and FiO2 were most influential. Count features were more influential in the 24-hour window, likely due to their encoding of clinical concern. For each strategy, LightGBM and XGBoost consistently agreed on the most influential feature, although subsequent feature rankings varied between the two.

Overall, the consistent superiority of gradient boosting models with summary statistics suggests these approaches should be prioritized for early sepsis prediction. Allowing these models to natively handle missingness without imputation is critical. Furthermore, SHAP-based interpretability can help support clinical adoption, improving model transparency.

5.2 Limitations and Future Research

This study relied on a single dataset (PhysioNet 2019), which may not fully generalize to all ICU populations or hospital systems without external validation. Predictions were based on static early-window features and thus cannot provide precise onset timing. The use of median imputation, while effective, is a relatively simple approach; more sophisticated methods such as MICE or deep-learning-based imputation could yield further improvements. Similarly, undersampling addressed class imbalance but discards the majority-class information. Future work should evaluate alternative techniques such as oversampling or synthetic data generation (e.g., SMOTE) to preserve signal from the full dataset.

Despite these limitations, this controlled setting was valuable for systematically comparing model classes and evaluating the strength of early feature signals. Building on these insights, future work should focus on rolling, real-time prediction frameworks that continuously update risk scores and estimate the timing of sepsis onset. Incorporating richer temporal features (e.g., trends, rates of change) and survival analysis techniques could improve performance, as well as exploring deep learning architectures specifically designed for time-series data. Validating these models on diverse, multi-center datasets will help assess generalizability and potential for clinical deployment.

Supplemental Materials

The complete dataset and all Python code used in this study are available in the GitHub repository at https://github.com/ab2028/early-sepsis-pred. Readers are encouraged to access the repository to review and reproduce the analyses.

Acknowledgments

I would like to sincerely thank all my teachers who have played a role in shaping my academic journey. Your support, patience, and dedication have meant a great deal to me. A special thank you to Dr. Olga Korosteleva, Professor of Statistics at CSULB, for her exceptional mentorship and for always encouraging me to challenge myself and grow. I am also grateful to Dr. Oleg Gleizer at UCLA Math Circle for introducing me to this research opportunity. My gratitude also extends to Geoffrey Robertson, my Honors Biology teacher, and to my mathematics teachers, Katya Malikov and Junko Goldie Roberts, whose engaging instruction and dedication helped me build the essential foundations and skills for my research. I'm also grateful to Isha Agarwal, Jessika Baral, and Siya Goel for taking the time to discuss my ideas and provide feedback as I worked on this project. To my family – thank you for your constant love, encouragement, and belief in me. Your support has been invaluable.

References

[1] Singer, Mervyn et al. "The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". In: *JAMA* 315.8 (2016). DOI: 10.1001/jama.2016.0287.

[2] Rudd, Kristina E et al. "Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the Global Burden of Disease Study". In: *The Lancet* 395.10219 (2020). DOI: 10.1016/S0140-6736(19)32989-7.

[3] Agency for Healthcare Research and Quality. *Statistical Brief #204: National Inpatient Hospital Costs: The Most Expensive Conditions by Payer, 2013*. https://www.ncbi.nlm.nih.gov/books/NBK368492/. 2016.

[4] Kumar, Anand et al. "Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock". In: *Critical Care Medicine* 34.6 (2006). DOI: 10.1097/01.CCM.0000217961.75225.E9.

[5] Komorowski, Matthieu et al. "The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care". In: *Nature Medicine* 24.11 (2018). DOI: 10.1038/s41591-018-0213-5.

[6] Bone, R. C. et al. "Definitions for Sepsis and Organ Failure and Guidelines for the Use of Innovative Therapies in Sepsis". In: *Chest* 101.6 (1992), pp. 1644–1655. DOI: 10.1378/chest.101.6.1644.

[7] Vincent, J. L. et al. "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure". In: *Intensive Care Medicine* 22.7 (1996), pp. 707–710. DOI: 10.1007/BF01709751.

[8] Seymour, Christopher W. et al. "Assessment of Clinical Criteria for Sepsis: For the Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3)". In: *JAMA* 315.8 (2016), pp. 762–774. DOI: 10.1001/jama.2016.0288.

[9] Rafiei, Alireza et al. "SSP: Early prediction of sepsis using fully connected LSTM-CNN model". In: *Computers in Biology and Medicine* 128 (2021), p. 104110. ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2020.104110.

[10] Delahanty, Ryan J. et al. "Development and Evaluation of a Machine Learning Model for the Early Identification of Patients at Risk for Sepsis". In: *Annals of Emergency Medicine* 73.4 (2019), pp. 334–344. DOI: 10.1016/j.annemergmed.2018.11.036.

[11] Deng, Hong-Fei et al. "Evaluating machine learning models for sepsis prediction: A systematic review of methodologies". In: *iScience* 25.1 (2022). Open Access, p. 103651. DOI: 10.1016/j.isci.2021.103651.

[12] Reyna, Matthew A. et al. "Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019". In: *Wolters Kluwer Health, Inc.* 48.2 (2020). Editor's Choice, pp. 210–217. DOI: 10.1097/CCM.0000000000004145.

[13] Bomrah, Sherali et al. "A scoping review of machine learning for sepsis prediction: feature engineering strategies and model performance–a step towards explainability". In: *Critical Care* 28.1 (2024), p. 148. DOI: 10.1186/s13054-024-04948-6.

[14] Mao, Qingqing et al. "Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and ICU". In: *BMJ Open* 8.1 (2018). ISSN: 2044-6055. DOI: 10.1136/bmjopen-2017-017833.

[15] Morrill, James H. et al. "Utilization of the Signature Method to Identify the Early Onset of Sepsis From Multivariate Physiological Time Series in Critical Care Monitoring". In: *Wolters Kluwer Health, Inc.* 48.10 (2020), e976–e981. DOI: 10.1097/CCM.0000000000004510.

[16] Du, John Anda, Sadr, Nadi, and Chazal, Philip de. "Automated Prediction of Sepsis Onset Using Gradient Boosted Decision Trees". In: *Computing in Cardiology*. Vol. 46. Team "Sepsyd", 2nd place in the PhysioNet/Computing in Cardiology Challenge 2019. IEEE. 2019, pp. 1–4. DOI: 10.22489/CinC.2019.423. URL: https://www.cinc.org/archives/2019/pdf/CinC2019-0423.pdf.

[17] Zabihi, Morteza, Kiranyaz, Serkan, and Gabbouj, Moncef. "Sepsis Prediction in Intensive Care Unit Using Ensemble of XGBoost Models". In: *Computing in Cardiology*. Vol. 46. Team "Separatrix", 3rd place in the PhysioNet/Computing in Cardiology Challenge 2019. IEEE. 2019, pp. 1–4. DOI: 10.22489/CinC.2019.238.

[18] Li, Xiang et al. "A Time-Phased Machine Learning Model for Real-Time Prediction of Sepsis in Critical Care". In: *Wolters Kluwer Health, Inc.* 48.10 (2020). PhysioNet/Computing in Cardiology Challenge 2019, e884–e888. DOI: 10.1097/CCM.0000000000004494.

[19] Singh, Janmajay et al. “Utilizing Informative Missingness for Early Prediction of Sepsis”. In: *Computing in Cardiology.* Vol. 46. Team “CTL”, 5th place in the PhysioNet/Computing in Cardiology Challenge 2019. IEEE. 2019, pp. 1–4. DOI: 10.22489/CinC.2019.280.

[20] Mitra, Avijit, and Ashraf, Khalid. “Sepsis Prediction and Vital Signs Ranking in Intensive Care Unit Patients”. In: *arXiv preprint arXiv:1812.06686* (2018). DOI: 10.48550/arXiv.1812.06686.

[21] Samek, Wojciech et al. “Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications”. In: *Proceedings of the IEEE* 109.3 (2021), pp. 247–278. DOI: 10.1109/JPROC.2021.3060483.

[22] Fleuren, Lucas M. et al. “Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy”. In: *Intensive Care Medicine* 46.3 (2020), pp. 383–400. DOI: 10.1007/s00134-019-05872-y.

[23] Moor, Michael et al. “Evaluating machine learning models for sepsis prediction: A systematic review of methodologies”. In: *iScience* 25.2 (2022), p. 103651. DOI: 10.3389/fmed.2021.607952.

[24] Faisal, Muhammad et al. “Development and External Validation of an Automated Computer-Aided Risk Score for Predicting Sepsis in Emergency Medical Admissions Using the Patient’s First Electronically Recorded Vital Signs and Blood Test Results”. In: *Wolters Kluwer Health, Inc.* 46.4 (2018), pp. 612–618. DOI: 10.1097/CCM.0000000000002967.

[25] Kuo, Yao-Yi, Huang, Shu-Tien, and Chiu, Hung-Wen. “Applying artificial neural network for early detection of sepsis with intentionally preserved highly missing real-world data for simulating clinical situation”. In: *BMC Medical Informatics and Decision Making* 21.1 (2021), p. 290. DOI: 10.1186/s12911-021-01651-z.

[26] Wang, Zichen et al. “A methodological systematic review of validation and performance of sepsis real-time prediction models”. In: *npj Digital Medicine* 8.1 (2025), p. 190. DOI: 10.1038/s41746-025-01587-1.

[27] James, Gareth et al. *An Introduction to Statistical Learning with Applications in R.* 2nd. New York, NY: Springer, 2021. ISBN: 978-1-0716-1417-4.

[28] Cerulli, Giovanni. *Fundamentals of Supervised Machine Learning With Applications in Python, R, and Stata.* Cham: Springer, 2023. DOI: 10.1007/978-3-031-23248-3.

Appendix. Variable Definitions and Distributions

Table 11 lists all variables and their definitions. Figures 4, 5, and 6 show the variable distributions.

Table 11: List of Variables Used in the Study

Category	**Variable (Description)**
Vitals	HR – Heart rate (beats/min)
	O2Sat – Oxygen saturation (%)
	Temp – Body temperature (°C)
	SBP – Systolic blood pressure (mmHg)
	MAP – Mean arterial pressure (mmHg)
	DBP – Diastolic blood pressure (mmHg)
	Resp – Respiratory rate (breaths/min)
Labs	FiO2 – Fraction of inspired oxygen (%)
	Platelets – Platelet count ($\times 10^9$/L)
	Hgb – Hemoglobin (g/dL)
	WBC – White blood cell count ($\times 10^9$/L)
	Lactate – Serum lactate (mmol/L)
	Creatinine – Serum creatinine (mg/dL)
	BUN – Blood urea nitrogen (mg/dL)
	Glucose – Blood glucose (mg/dL)
	pH – Arterial blood pH
	PaCO2 – Partial pressure of CO2 (mmHg)
	HCO3 – Bicarbonate (mmol/L)
Demographics	Age - Age in years (capped at 100 for those 90+)
	Gender - Female (0) or Male (1)
	Unit - Administrative identifier for ICU unit (0=Missing, 1=MICU, 2=SICU)

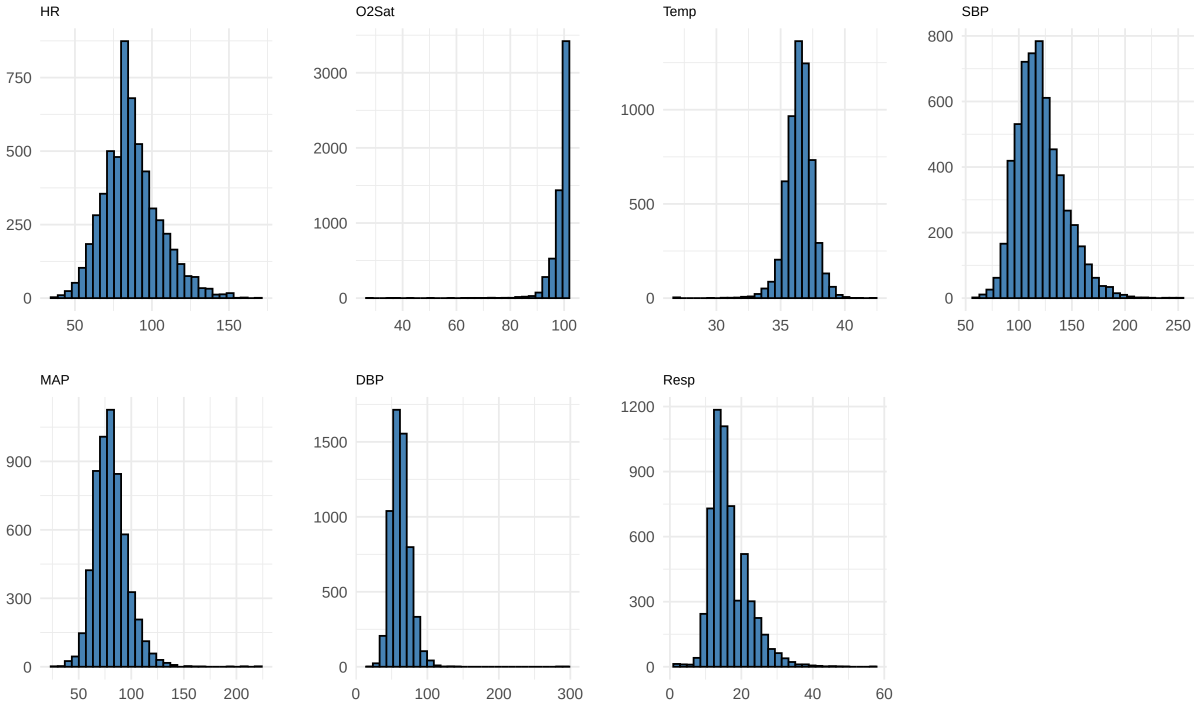


Figure 4: Histograms of vital sign variables used in the study.

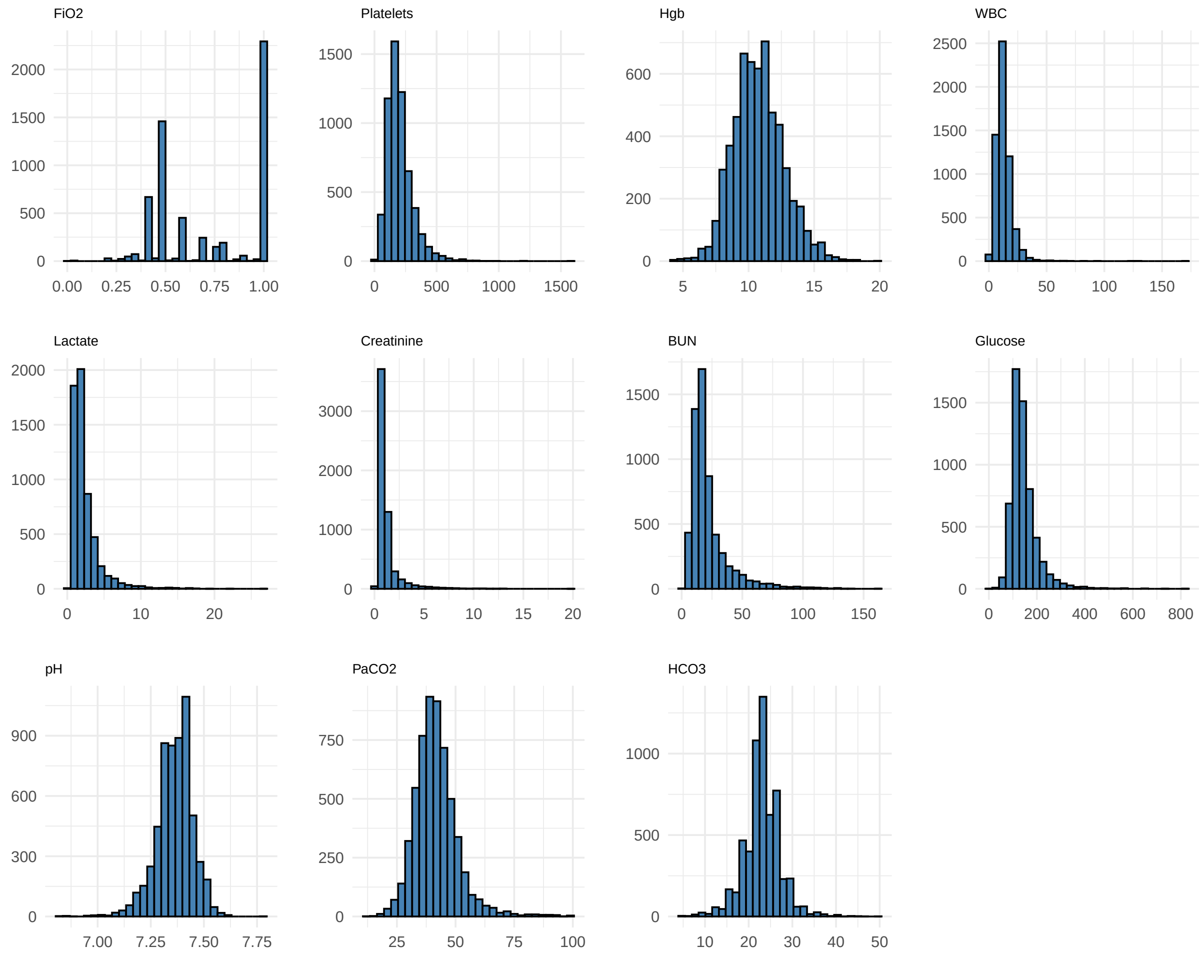


Figure 5: Histograms of lab variables used in the study.

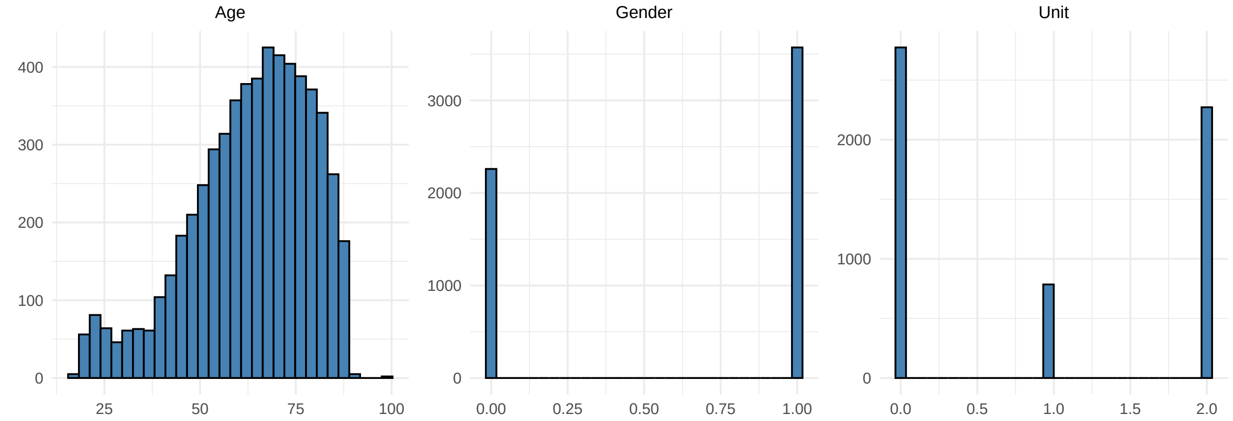


Figure 6: Histograms of demographic variables used in the study.