Міністерство освіти і науки України

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра мультимедійних технологій

Кваліфікаційна робота

Освітній ступень – бакалавр

На тему**: «EARLY PREDICTION OF SEPSIS FROM TIME-SERIES CLINICAL DATA»**

Виконала: студентка 4-го року навчання,
Спеціальності F2 «Інженерія програмного забезпечення»

Речкалова Анна Владиславівна

Керівник Кузьменко Д.О., ст.викладач

Рецензент _____

Кваліфікаційна робота захищена з оцінкою _____

Секретар ЕК _____

«___» _____ 2026 р.

Київ-2026

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра мультимедійних технологій

Освітній ступень бакалавра

Спеціальність F2 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри інформатики

доцент, к.ф.-м.н

_____ Жежерун О.П.

(підпис)

«____» _____ 2025 р.

ЗАВДАННЯ

ДЛЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТЦІ

Речкаловій Анні Владиславівні

1. Тема роботи: «Early prediction of sepsis from time-series clinical data/Раннє прогнозування сепсису на основі часових рядів медичних даних»
   Керівник роботи: Кузьменко Дмитро Олександрович, старший викладач
2. Строк подання студентом роботи: «____» _____ 2026 р.
3. План роботи:
   1. Individual Assignment
   2. Abstract
   3. Introduction
   4. Literature review, data collection and synthesis, processing of materials and prior studies
   5. Research data and preprocessing

6. Experiments
7. Results
8. Software implementation of the system components
9. Discussion

«____» _____ 2025 р.

Керівник _____ (підпис)

Завдання отримав ____ (підпис)

ГРАФІК ПІДГОТОВКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

№ з/п	ПЕРЕЛІК РОБІТ	Термін виконання	Дата ознайомлення наукового керівника	Підпис наукового керівника	Примітка
1.	Вибір теми, затвердження її на засіданні кафедри та закріплення наукового керівника. Узгодження календарного графіка підготовки кваліфікаційної роботи. Ознайомлення студента з критеріями оцінювання кваліфікаційної роботи.	жовтень			
2.	Вивчення джерел літератури, збір та узагальнення фактів, даних, опрацювання матеріалів. Виконання аналізу попередніх досліджень дотичних до теми.	жовтень – листопад			
3.	Складання плану кваліфікаційної роботи та узгодження з науковим керівником.	до 14 грудня			
4.	Написання розділів та постановка експериментів.	грудень – березень			
	Розділ 1 Характеристика проблеми раннього виявлення критичних станів (сепсису). Огляд сучасних інформаційних технологій та методів Machine Learning у медицині. Аналіз існуючих аналогів та обґрунтування необхідності розробки власної системи. Постановка задачі дослідження.				
	Розділ 2				

	Попередній аналіз та підготовка вхідних даних. Обґрунтування вибору моделей та алгоритмів. Методи формування ознак для навчання моделей. Розробка методики проведення експерименту та критеріїв оцінки якості.				
	Розділ 3 Навчання та налаштування прогностичних моделей. Аналіз результатів обчислювальних експериментів. Опис програмної реалізації компонентів системи.				
8.	Проміжний звіт про виконання роботи.	до 31 січня			
9.	Чорнова версія кваліфікаційної роботи.	до 15 березня			
10.	Повне завершення написання кваліфікаційної роботи, оформлення її згідно вимогами й подання на відгук науковому керівнику.	квітень – початок травня			
11.	Попередній захист кваліфікаційної роботи.	з 30 березня по 1 квітня			
12.	Завантаження фінальної версії роботи.	до 15 травня			
13.	Захист кваліфікаційної роботи перед екзаменаційною комісією.	з 25 по 27 травня			

Студент: Речкалова А.В.

Керівник: Кузьменко Д.О.

“____” _____ 2025 р.

TABLE OF CONTENTS

ABSTRACT

This study addresses the problem of early sepsis prediction from clinical time series data collected in the intensive care unit. The PhysioNet/Computing in Cardiology Challenge 2019 dataset, comprising 40,336 ICU patients, is used as the experimental basis. Three model architectures were evaluated: LightGBM as a gradient boosting baseline, a Temporal Convolutional Network as a deep learning baseline and a hybrid TCN-LightGBM architecture. MNAR-aware feature engineering is applied. It treats missingness patterns as informative clinical signals. Feature selection is performed independently for each architecture: gain-based importance for LightGBM and permutation importance for TCN. Hyperparameter optimization is conducted via Bayesian search. Model-specific explainability is provided through SHAP attributions for LightGBM and WinTSR for TCN, demonstrating that both models capture clinically meaningful patterns through different representations. LightGBM achieves the highest performance, with a test utility score of 0.381 and AUPRC of 0.116. The hybrid architecture does not improve over the standalone models. A Streamlit demo is also provided for the LightGBM model.

Keywords: sepsis prediction, clinical time series, LightGBM, temporal convolutional network, SHAP, WinTSR, Missing Not At Random, explainability.

АНОТАЦІЯ

Дана робота розглядає задачу раннього прогнозування сепсису за клінічними часовими рядами, зібраними у відділенні інтенсивної терапії. Для проведення експериментів використовується набір даних PhysioNet/Computing in Cardiology Challenge 2019, що охоплює 40,336 пацієнтів ВІТ. У роботі оцінюються три архітектури моделей: LightGBM як базова модель на основі градієнтного бустингу, темпоральна згорткова мережа (TCN) як базова модель глибокого навчання, а також гібридна архітектура TCN-LightGBM. Застосовується MNAR-орієнтована інженерія ознак, що розглядає патерни пропущених значень як інформативні клінічні сигнали. Відбір ознак виконується окремо для кожної архітектури: на основі важливості за приростом для LightGBM та на основі пермутаційної важливості для TCN. Оптимізація гіперпараметрів проводиться за допомогою байєсівського пошуку. Інтерпретованість моделей забезпечується через SHAP-атрибуції для LightGBM та WinTSR для TCN. Обидва методи виявляють клінічно значущі патерни, хоча й через різні представлення. Найвищу якість прогнозування демонструє LightGBM: тестова utility score становить 0,381, AUPRC – 0,116. Гібридна архітектура не покращує результати порівняно з окремими моделями. Додатково розроблено демонстраційний застосунок на основі Streamlit для моделі LightGBM.

Ключові слова: sepsis prediction, clinical time series, LightGBM, temporal convolutional network, SHAP, WinTSR, Missing Not At Random, explainability.

LIST OF ABBREVIATIONS

- ICU – Intensive Care Unit
- MNAR – Missing Not at Random
- LightGBM – Light Gradient Boosting Machine
- TCN – Temporal Convolutional Network
- AUPRC – Area Under the Precision-Recall Curve
- SHAP – SHapley Additive exPlanations
- WinTSR – Windowed Temporal Saliency Rescaling

1. INTRODUCTION

Sepsis is a life-threatening organ dysfunction caused by a dysregulated host response to infection [1]. It caused about 166 million cases and 21.4 million deaths in recent estimates, representing 31.5% of total global deaths [2]. Each hour of delayed treatment increases the risk of death by approximately 7%, making early detection critical.

Current clinical tools such as SOFA and qSOFA provide manual assessments and are designed to diagnose existing sepsis rather than predict it in advance [3] . This limitation motivates the use of machine learning to predict sepsis earlier using continuously collected patient data from the intensive care unit.

The PhysioNet/Computing in Cardiology Challenge 2019 provides a benchmark dataset for this task [4]. It contains hourly ICU measurements from 40,336 patients across two hospital systems, with sepsis labels shifted six hours before clinical onset to enable predictive modeling. A key challenge in this dataset is the large proportion of missing values across clinical features. Missing values in ICU data are often not random: a measurement may be absent because a clinician decided it was unnecessary, which itself reflects the patient's condition. Singh et al. [5] demonstrated that modeling such absences as explicit binary features improves sepsis prediction. This study follows the same approach.

The research focus is clinical time-series data from intensive care unit patients. The subject of study is machine learning methods for early sepsis prediction and their interpretability.

The research scope of this study is to develop and compare machine learning approaches for early sepsis prediction from ICU time series data with a focus on model interpretability. To achieve this goal, the following tasks were completed:

1. Exploratory data analysis was conducted to identify class imbalance, missing value patterns and their clinical significance.

2. A preprocessing pipeline was developed, including forward-fill imputation with binary missingness indicators, sliding window aggregates and clinically motivated features.
3. Three model architectures were implemented and evaluated: LightGBM, Temporal Convolutional Network and hybrid TCN-LightGBM pipeline.
4. Feature selection and hyperparameter optimization were applied to each model.
5. Model performance was evaluated using the PhysioNet Challenge utility score and AUPRC.
6. Model-specific explainability methods were applied: SHAP for LightGBM and WinTSR for TCN to validate that the models capture clinically relevant patterns.

Three modeling paradigms are compared. LightGBM [6] is selected as a gradient boosting baseline due to its strong performance on tabular data. A Temporal Convolutional Network [7] is applied to raw clinical sequences as a deep learning approach. A hybrid pipeline is TCN-derived embeddings that serve as input to LightGBM. Model predictions are interpreted using SHAP [8] for LightGBM and WinTSR [9] for TCN.

This thesis' structure is described in the following sentences. Section 2 reviews related work on sepsis prediction. Section 3 describes the dataset, exploratory data analysis and preprocessing pipeline. Section 4 presents theoretical foundations with the three model architectures, feature selection and hyperparameter optimization procedures applied to each model. Section 5 reports experimental results and applies SHAP and WinTSR to analyze model behavior on individual patients. Section 6 compares metrics with official challenge results and present demo application. Section 7 presents discussion and outlines directions for future works.

2. RELATED WORK

Early sepsis prediction from clinical time series has been studied extensively using the PhysioNet/Computing in Cardiology Challenge 2019 benchmark [4]. The challenge defined a utility score that rewards early correct predictions and penalizes false alarms. Morrill et al. [10] achieved the highest score in the challenge using a gradient boosting model with features derived by signature-based method, alongside original and hand-crafted clinical features. Other high-ranking teams similarly relied on XGBoost [29, 30]. Deep learning approaches were also explored (LSTM, CNN, hybrid architectures) but generally did not surpass gradient boosting baselines on this benchmark [4, 29]. This showed that classical machine learning with careful feature engineering can outperform more complex approaches on structured clinical data.

Recent work has shifted to larger datasets. Johnson et al. [11] released MIMIC-IV. It is a critical care database covering a broader patient population. Models trained on larger datasets benefit from greater sample size and diversity which can improve generalization. However, clinical data varies significantly across hospitals due to differences in measurement protocols, patient demographics and clinical practices. This makes generalization a known challenge in sepsis prediction. This study uses the PhysioNet 2019 dataset to allow direct comparison with challenge results and to keep the evaluation setting consistent.

Missing values are a known problem in ICU data. It's especially applied to laboratory measurements that are ordered selectively rather than collected continuously. Two main directions have been explored. The first direction treats missingness as a problem to be solved through imputation or representation learning. For example, Du et al. [12] proposed SAITS. This model learns missing values from a weighted combination of two diagonally-masked self-attention blocks. This approach assumes missingness is random and aims to recover the true underlying value. Zhang et al. [13] introduced Raindrop, a graph-based network that learns representations directly from irregularly sampled data through dependency graphs without requiring explicit imputation. The second direction treats missingness as a signal rather than

noise. If a measurement is absent, it may mean the clinician did not see a reason to order it. Under this assumption, replacing missing values with estimated ones risks losing useful information. Singh et al. [5] showed that binary observation indicators improve sepsis prediction on the same benchmark, which this study interprets as evidence of Missing Not At Random [17] structure in ICU data. The same approach is adopted here.

Explainability of clinical prediction models has received growing attention. Clinicians need to understand which features drove a particular prediction before acting on it. This is especially important for sepsis where decisions must be made quickly and errors have serious consequences. Several methods have been proposed for explaining predictions from time series models. Tonekaboni et al. [15] proposed FIT. This method assigns importance to observations in a time series by measuring how much each feature explains the change in model predictions from one time step to the next. It relies on a recurrent generative model to estimate what the prediction would look like without a given feature. These conditions makes it well suited for RNN architectures but not directly applicable to TCN. Islam and Fox [9] and proposed WinTSR (built on the Temporal Saliency Rescaling method), which applies relevance scores at the window level rather than individual time steps. The method works in two stages: identifies which time windows contributed most to a prediction, then determines which features were important in these windows. This makes it well suited for capturing temporal dependencies that point-wise methods ignore. SHAP [8] assigns importance scores to individual features by measuring their contribution to a model prediction. It is used for tree-based models because it can compute exact Shapley values using the tree structure. WinTSR is used in this study for TCN explanations. SHAP is applied to LightGBM.

Unlike prior work that evaluates a single model class or applies model-agnostic attribution methods, this study directly compares how classical and deep learning models work with MNAR structure on a consistent benchmark, using architecture-appropriate explainability methods for each model.

3. METHODS

3.1. Dataset and Preprocessing

The data for this research are from the PhysioNet/Computing in Cardiology Challenge 2019 [4], which focuses on the early detection of sepsis. The dataset originates from patients in the intensive care unit (ICU) across three hospital systems. For this study, the public training set provided by the challenge was used, comprising data from two hospital systems (A and B) totaling 40,336 patients.

Each patient contains 40 time-dependent clinical variables and 1 variable, which is the Sepsis label. Clinical features can be categorized into three groups: Vital Signs (8 variables), Laboratory Values (26 variables), and Demographics (6 variables). All features are listed in Appendix A (Tables A.1-A.3).

Sepsis labels are defined according to the Sepsis-3 guidelines [1]. To enable prediction, the authors of the dataset shifted the labels by 6 hours. So, patients are marked to have sepsis 6 hours before it actually occurs. For sepsis patients, the sepsis label is 1, for non-septic patients, 0.

3.1.1. Exploratory Data Analysis

Exploratory Data Analysis [16] was performed to develop hypotheses, and take the steps for data preprocessing described in Section 3.1.2.

Class Imbalance

Of 40,336 patients: 2,932 have sepsis (7.3%), and 37,404 do not (92.7%). The dataset has a significant imbalance between groups, as shown in Figure 3.1.

Length of Stay in the ICU

The median ICU stay across all patients is 38 hours, with most stays between 24 and 50 hours. However, some patients stayed up to 336 hours, while the minimum is 8 hours (Figure 3.2).

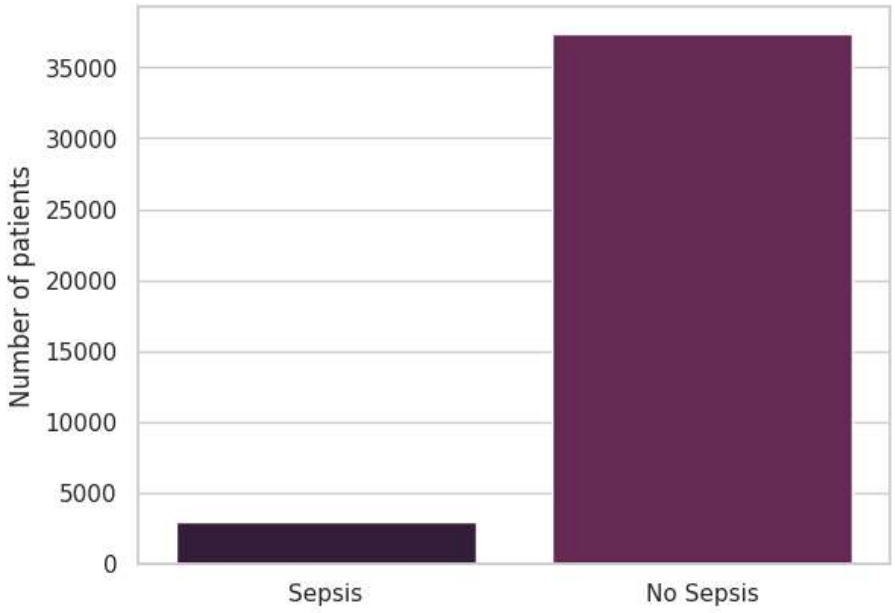

Figure 3.1: Number comparison of septic and non-septic patients

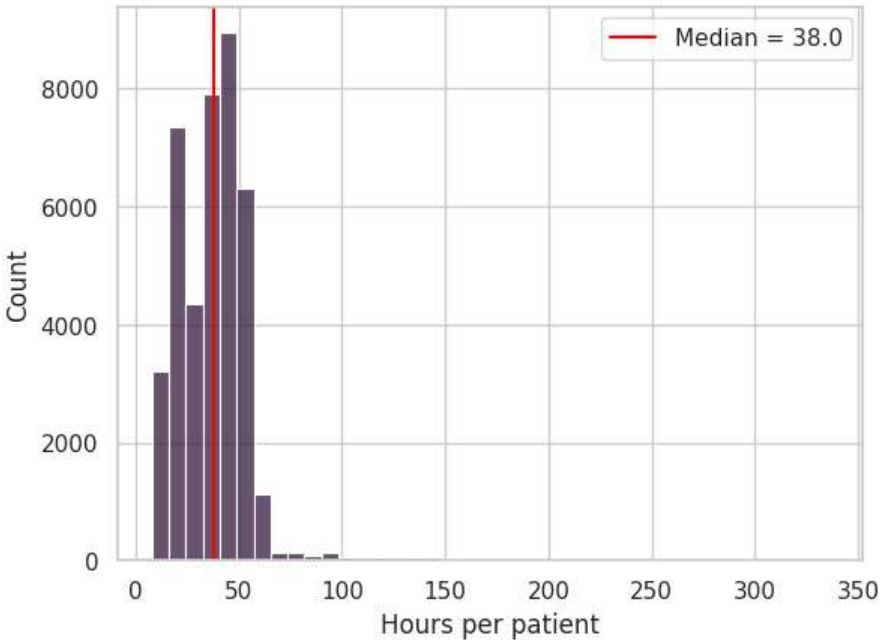

Figure 3.2: Distribution of recorded ICU hours per patient

Non-septic patients show a symmetric distribution, with both mean and median around 36-39 hours. In contrast, septic patients have a much higher mean (58.8 hours) despite a similar median (37.5 hours) (Table 4.1). This gap is caused by a few «long-stay» cases that raise the average upward. This distribution shows that models must handle time series of varying length.

Table 3.1: Mean and median length of ICU stay

Patient's health condition	**Mean (hours)**	**Median (hours)**
Septic	58.80	37.5
Non-septic	36.88	39.0

Time from ICU to Sepsis Diagnosis

The median between ICU admission and the first positive sepsis test is 29 hours, which means that for half of the patients, sepsis is detected within the first 29 hours of admission to the intensive care unit. However, for most patients, sepsis is detected in the first 10 hours (Figure 3.3).

Missing Values

The dataset contains a large number of missing values. Some laboratory parameters have almost complete data gaps, as shown in Table 3.2. For example, Bilirubin_direct (99.8%), Fibrinogen (99.3%), and TroponinI (99.0%).

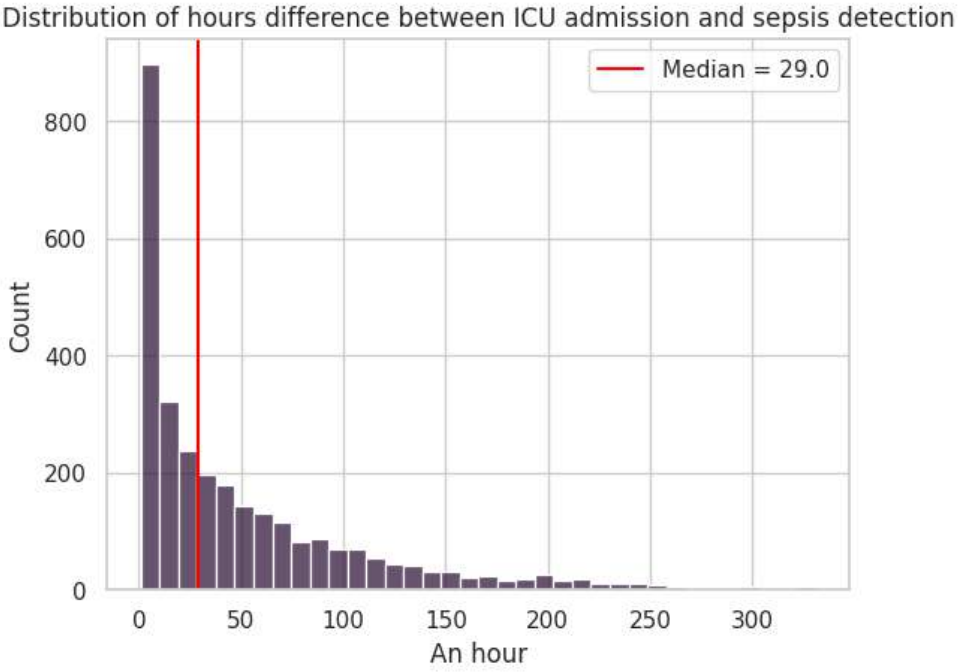

Figure 3.3: Distribution of hours difference between ICU admission and sepsis detection

Table 3.2: Top-10 features with the largest missing percentage

Variable	**Type**	**Missing (%)**
Bilirubin_direct	Laboratory	99.81%
Fibrinogen	Laboratory	99.34%
TroponinI	Laboratory	99.04%
Bilirubin_total	Laboratory	98.51%
Alkalinephos	Laboratory	98.39%
AST	Laboratory	98.38%
Lactate	Laboratory	97.33%
PTT	Laboratory	97.06%
SaO2	Laboratory	96.55%
EtCO2	Laboratory	96.29%

Vital signs are missing much less frequently: HR (9.9%), MAP (12.5%), and O2Sat (13.1%). The list of missing values percentage for all features is shown in Figure 3.4.

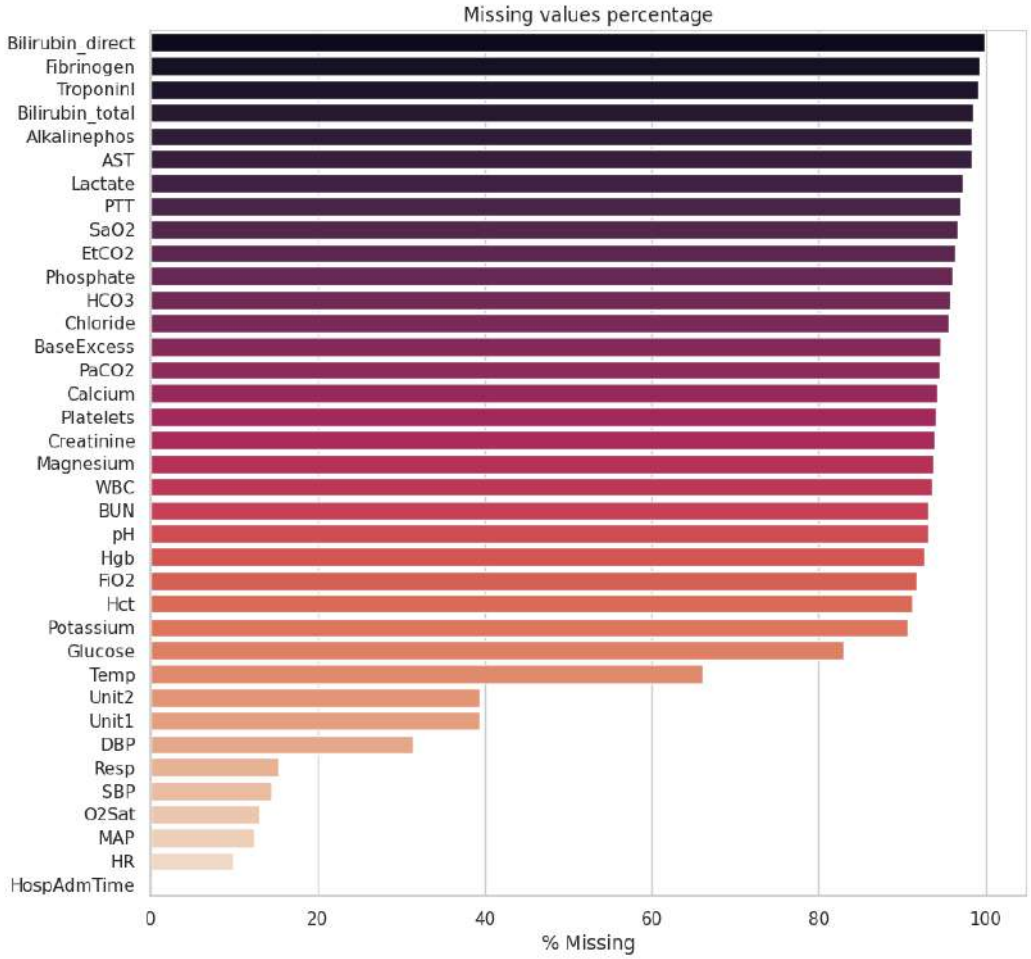

Figure 3.4: Missing values percentage for all features

This raises the question of whether it is appropriate to remove certain columns due to the large number of missing values. To test this hypothesis, an additional analysis was conducted.

Frequently Measured Indicators in Septic Patients

It is important to determine which parameters are measured more frequently in patients with sepsis. Several features are measured more often in septic patients. The largest differences were observed for Lactate (+33.5%), FiO2 (+33.2%), PaCO2

(+31.8%), pH (+31.2%), BaseExcess (+22.1%), and Alkalinephos (+21.6%). Only features with a group difference higher than 10 percent were retained for further analysis (Table 3.3).

Table 3.3: Measurement frequency difference between sepsis and non-sepsis patients

Variable	**Type**	**Frequency difference**
Lactate	Laboratory	33.5%
FiO2	Laboratory	33.19%
PaCO2	Laboratory	31.84%
pH	Laboratory	31.21%
BaseExcess	Laboratory	22.13%
Alkalinephos	Laboratory	21.58%
AST	Laboratory	21.46%
Bilirubin_total	Laboratory	21.3%
SaO2	Laboratory	20.25%
PTT	Laboratory	17.17%
Chloride	Laboratory	12.49%
Phosphate	Laboratory	12.33%
EtCO2	Laboratory	10.79%

Timing of First Measurements Before Sepsis Onset

The next step was to investigate exactly when measurements of these indicators first appear relative to the time the sepsis label is appeared. For these features, the median time of the first measurement is around 25 hours before the sepsis onset label. For example, as shown in Figure 3.5, most patients have their first lactate measurement within the last 50 hours before onset, with a median of 23 hours before. It means they

appear before the diagnosis is made and can serve as early signals. The number of measurements grows as the patient gets closer to the critical moment.

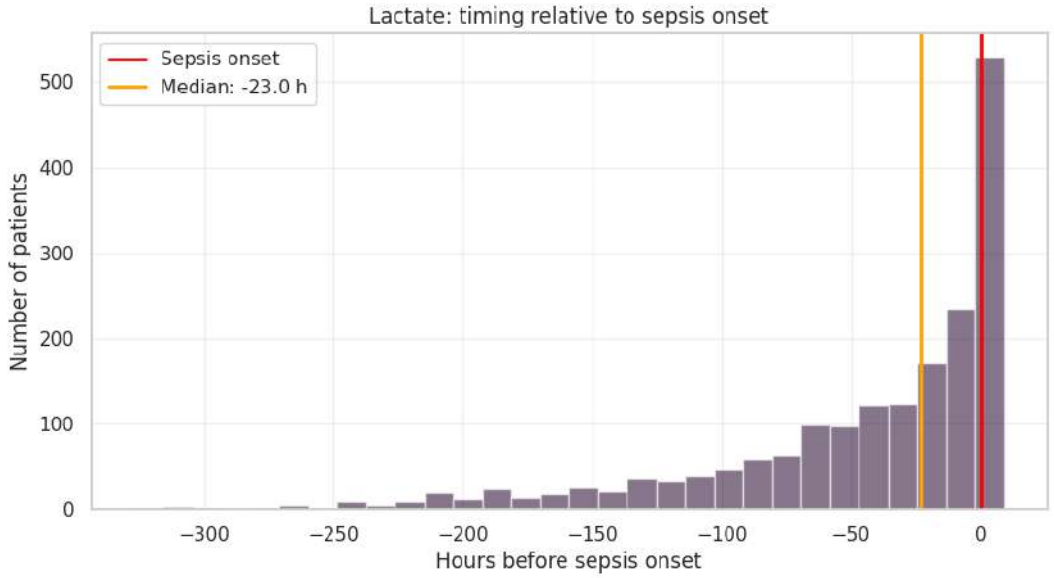

Figure 3.5: Median timing of first lactate measurement relative to sepsis onset

There is a significant amount of missing data, but despite this, it would be a mistake to discard these columns. The absence of certain data in the dataset indicates the patient's condition. For example, measured FiO2 and PaCO2 mean that a patient is connected to a ventilator and is in critical condition. So, missing values may contain additional information about the patient's condition, consistent with the Missing Not at Random (MNAR) [17] assumption.

3.1.2. Data processing

In total, 240 features were derived after the data processing: 40 original, 36 missing indicators, 2 manually created and 162 engineered.

Manually Added Features

Two derived metrics were also added to the dataset. The ShockIndex is defined as the ratio of heart rate to systolic blood pressure. The second is the bilirubin-to-creatinine ratio (BUN/CR ratio). Both features were proposed in Henry et al. work [18] as important features in sepsis detection.

Forward-fill and Missing Values Binary Mask

To handle a large number of missing values, the forward-fill method was used, meaning that missing values were imputed using the most recent known measurements. This approach is based on the assumption that vital signs do not change instantaneously.

Also, a binary missing indicator was added for each feature (except for demographic features). The value is 1 if the measurement was actually recorded at that hour and 0 if the value was filled in using forward-fill.

Feature engineering

One of the models used in this study is based on the gradient boosting trees algorithm. The main limitation of this approach is the lack of a memory mechanism. Therefore, the sliding windows method was applied to account this problem.

For vital signs, which are characterized by a relatively small number of missing values, statistical aggregates were calculated in 2-, 3-, 6-, and 12-hour windows: minimum, maximum, mean and standard deviation.

For parameters with a significant number of missing values but which are clinically important, the mean and measurement frequency were used in 6- and 12-hour windows. The choice of the mean is due to its ability to smooth out random shifts. Additionally, the mean reflects the patient's overall physiological state over a specific period. Also, for these features, measurement frequency was included as a separate feature. More frequent measurements of certain laboratory parameters typically

indicate increased clinical attention to the patient. This is consistent with the MNAR assumption.

3.2. Models

This study evaluates three model architectures representing different approaches to sepsis prediction from clinical time series: a classical machine learning model, a deep learning model, and a hybrid combination of both. The following subsections describe the motivation and architecture of each model: LightGBM as a gradient boosting baseline, a Temporal Convolutional Network as a deep learning approach, and a hybrid TCN-LightGBM architecture proposed in this work.

3.2.1. LightGBM

Analysis of top-performing submissions from the PhysioNet/Computing in Cardiology Challenge 2019 showed that gradient boosting methods, particularly LightGBM [8] and XGBoost [19], consistently achieved the highest scores.
LightGBM is a gradient boosting framework based on decision trees. Gradient boosting builds an ensemble of trees sequentially, where each tree fits the residual errors of the previous one. The final prediction is made by adding up the outputs from all the trees.

Traditional gradient boosted decision tree (GBDT) implementations are computationally expensive because they scan all data instances for every feature to estimate information gain. To address this, LightGBM introduces two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), as illustrated in Figure 3.6. GOSS focuses on instances with larger gradients while randomly downsampling the rest. EFB reduces the number of features by merging those that never have non-zero values at the same time into a single feature. Additionally, LightGBM uses a histogram-based algorithm to find the best split point. Instead of checking every possible value of a feature, it groups values into

bins and checks only those. This reduces the number of comparisons and speeds up training.

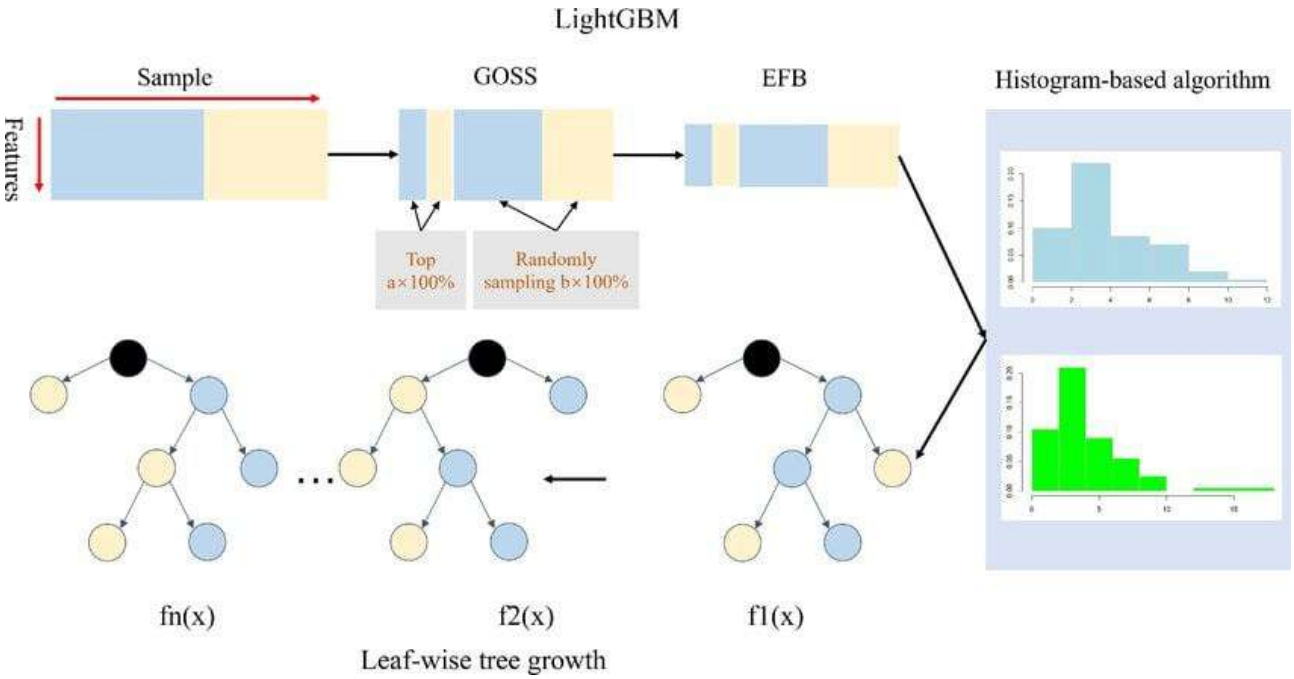

Figure 3.6: An overview of LightGBM [20]

Unlike XGBoost, LightGBM grows trees leaf-wise instead of level-wise. XGBoost expands all nodes at the same depth before moving deeper. LightGBM instead always splits the leaf with the highest loss reduction first. In conclusion, LightGBM is selected over XGBoost due to its significantly faster training speed.

3.2.2. Temporal Convolutional Networks

The deep learning models typically associated with time-series forecasting tasks are LSTM [21] and GRU [22]. However, a study by Bai et al. showed that convolutional networks can outperform RNNs [7]. They presented Temporal Convolutional Networks (TCN) that offer more stable gradients and better parallelism during training.

Causal Convolution

A key requirement for time series prediction is that the model cannot use future information to make predictions. TCN enforces this through causal convolutions, where the output at time depends only on inputs at time and earlier. This is achieved by applying left-side zero-padding of size (Figure 3.7), the formula of which is presented in the next section.

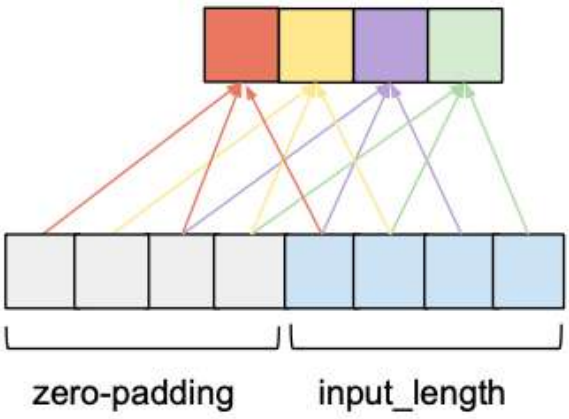

Figure 3.7: Causal convolution with left-side zero-padding [23]

The model must only rely on currently observed patient data $(x_0, ..., x_t)$ to generate a prediction (y_t), without access to future states.

Dilation

Dilation with a value of 1 (regular causal convolution) leads to a degradation problem because the network is only able to look back at a history with a size that is linear in the depth of the network. Dilated convolutions overcome this limitation by skipping inputs at a fixed interval. Also, the TCN achieves a large receptive field by increasing the dilation factor exponentially with the depth of the network (Figure 3.8). This allows the network to cover a longer history without adding more layers.

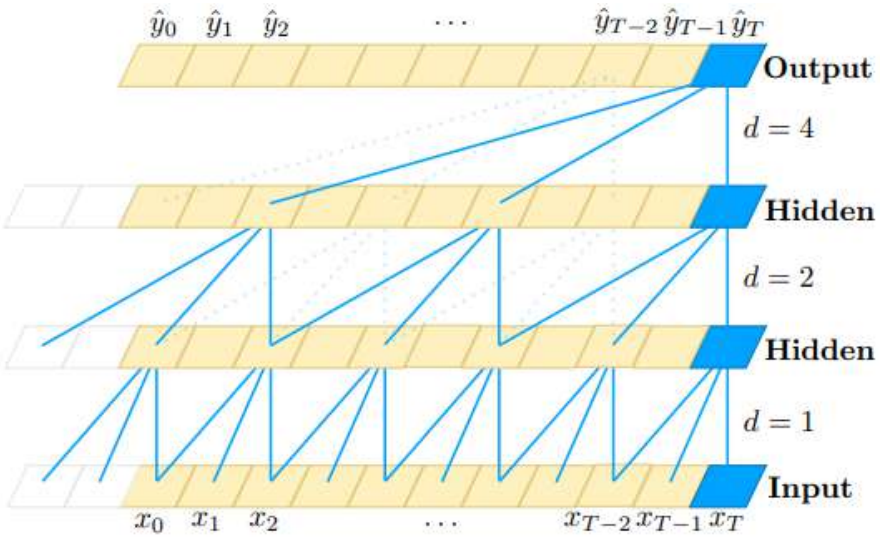

Figure 3.8: A dilated causal convolution with dilation factors d = 1, 2, 4 and filter size k = 3 [7]

The width of the receptive field of the dilated causal convolution is the following:

$$w = 1 + \sum_{i=0}^{n-1}(k-1) * b^i = 1 + (k-1) * \frac{b^n - 1}{b-1},$$

where *n* is a number of layers, *k* is the kernel size and *b* is the dilation value.

Residual blocks

Each layer of the TCN is organized into residual blocks (Figure 3.9). Each block contains two dilated causal convolutional layers with the same dilation factor, followed by ReLU activation, spatial dropout for regularization, and weight normalization. A residual connection adds the block input directly to its output, improving gradient flow during backpropagation. When the number of input and output channels differs, a 1×1 convolution is applied to match dimensions. So, the total width of the receptive field of a TCN:

$$w = 1 + \sum_{i=0}^{n-1} 2 * (k-1) * b^i = 1 + 2 * (k-1) * \frac{b^n - 1}{b-1},$$

where *n* is a number of layers, *k* is the kernel size and *b* is the dilation value.

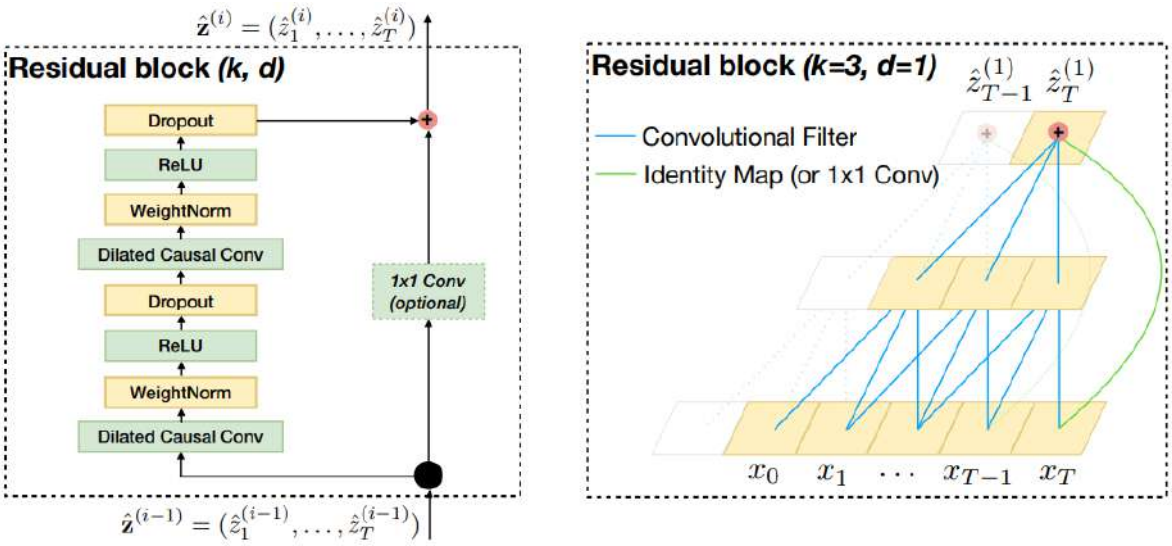

Figure 3.9: TCN residual block & An example of residual connection in a TCN [7]

Then, a minimum number of residual blocks for full history coverage:

$$n = \left\lceil \log_b \left(\frac{(l-1)(b-1)}{(k-1)*2} + 1 \right) \right\rceil,$$

where *l* is the length of the data series, *k* is the kernel size and *b* is the dilation value.

3.2.3. Hybrid TCN-LightGBM

The hybrid architecture combines the temporal representation capacity of TCN with the classification strength of LightGBM. The motivation is to investigate whether learned temporal embeddings can serve as more informative features than manually engineered statistical aggregates.

The pipeline consists of two stages. In the first stage, a pre-trained TCN is used as a feature extractor. The model is trained independently on the clinical time series and then frozen. For each patient, the TCN produces a sequence of embeddings. Each

timestep yields a fixed-size representation capturing temporal context from the patient's history up to that point. In the second stage, these per-timestep embeddings are passed directly to LightGBM as input features. This allows LightGBM to perform binary classification at each timestep using representations learned by the neural network rather than hand-crafted features.

3.3. Feature Selection

Feature selection was applied to reduce model complexity, remove redundant features, and improve training efficiency. Reducing the number of features also decreases the risk of overfitting and speeds up inference. Since the two architectures differ in how they process information, different selection methods were applied to each.

For LightGBM, gain-based importance was used. This is a built-in LightGBM metric that measures the total information gain contributed by each feature across all splits. Features were ranked by cumulative gain. Three subsets were then constructed based on thresholds of 99%, 95% and above-mean gain.

For the TCN, permutation importance was used [24]. This method evaluates each feature by randomly shuffling its values in the set and measuring the resulting change in model performance. Features with no positive contribution were removed.

3.4. Metrics

Two metrics were used to evaluate model performance: AUPRC and the utility score [4].

AUPRC (Area Under the Precision-Recall Curve) measures classification quality across all possible decision thresholds. It is suited for imbalanced datasets because it focuses on how well a model catches true positives without lots of false positives. A higher AUPRC indicates that the model achieves better precision at a given

level of recall. Unlike AUROC, AUPRC is not affected by the large number of true negatives, which makes it a more informative metric when the positive class is rare.

The utility score was designed specifically for this challenge and reflects clinical priorities. For sepsis patients, the score rewards predictions made between 12 hours before and 3 hours after the onset time. Predictions made more than 12 hours before onset receive a small penalty, reflecting that very early alarms carry limited clinical value. Missed detections receive a larger penalty. Figure 3.10 illustrates the utility functions for sepsis patients. For non-sepsis patients, any positive prediction is penalized as a false alarm.

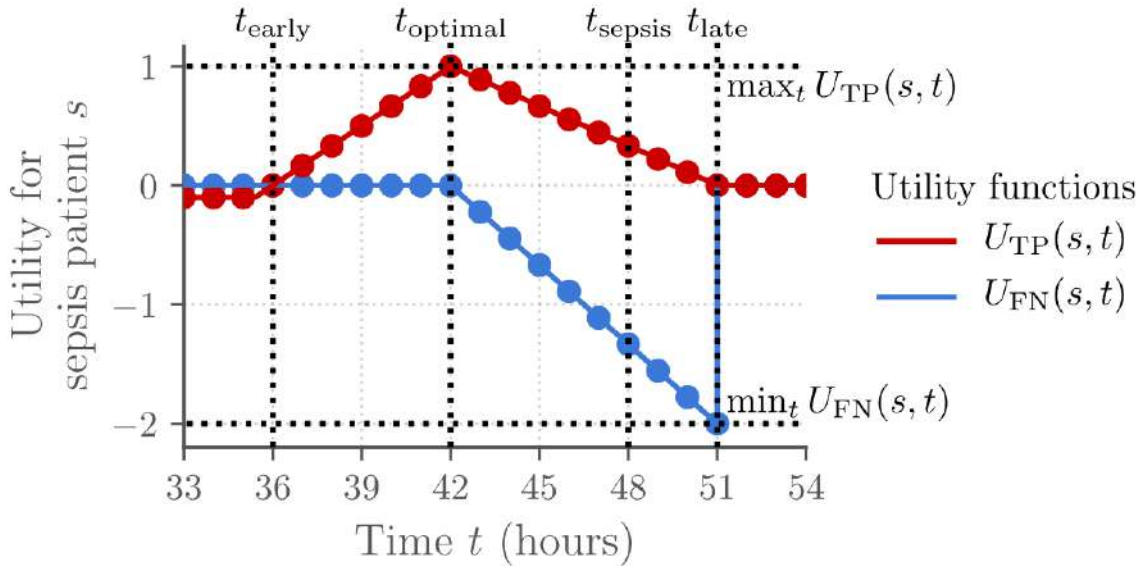

Figure 3.10: The utility function for a sepsis patient with sepsis time of 48 [4]

Hyperparameter optimization (described in the following section) was conducted using AUPRC as the objective, since it is threshold-independent and provides a stable signal during training. Final model evaluation and comparison were based primarily on the utility score.

3.5. Hyperparameter Search

Hyperparameter optimization was applied to improve predictive performance beyond the baseline configuration.

For LightGBM, two strategies were evaluated: random search [25] and Bayesian optimization. Random search samples hyperparameter combinations uniformly at random. Bayesian optimization was used via Optuna [26] which implements the Tree-structured Parzen Estimator (TPE) algorithm. Unlike random search, TPE uses results from previous trials to decide where to sample next, focusing on regions of the hyperparameter space that are more likely to improve performance.

For the TCN and the hybrid model, only Optuna was used. Running random search as a comparison baseline was not practical due to the higher computational cost of training neural networks. Early stopping was implemented manually and applied in the same way. For the TCN, optimization was conducted in two stages: the first stage searched over the full hyperparameter space including architecture parameters, and the second stage refined the continuous parameters over a narrowed range based on the best configuration found in the first stage.

3.6. Explainability

Explainability methods were applied to analyze individual model predictions and verify that the models focus on clinically relevant features. For each model, explanations are generated for a specific patient at a specific time. No explainability analysis was conducted for the hybrid model because of its two-stage architecture.

3.6.1. SHAP

For LightGBM, SHAP (SHapley Additive exPlanations) was used [8]. It is based on Shapley values from game theory. SHAP values satisfy three properties: local accuracy (the sum of attributions equals the model output), missingness (absent features have zero attribution) and consistency (if a feature's contribution increases in a modified model, its attribution does not decrease).

So, the Shapley value for the feature j is defined as:

$$\varphi_j(val) = \sum_{S \in \{1,\dots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} (val(S \cup \{j\}) - val(S)),$$

where S is a features' subset used in the model, p is the total number of features, $val(S)$ is the model prediction for the subset S.

A positive value φ indicates that the feature pushed the prediction toward sepsis, negative values indicate the opposite.

3.6.2. WinTSR

For the TCN, WinTSR (Windowed Temporal Saliency Rescaling) was used [9]. WinTSR is a model-agnostic method designed for time series data. The whole algorithm is described in Figure 3.11.

The method operates in two steps. In the first step, for each lookback window, all features are replaced with a baseline value and the change in the model output is measured.

baseline = $feature\_generator()$
for $l \leftarrow 0$ *to* L **do**
  # Mask all features at time l
  $X_t \setminus x_{:,l,t} = baseline_{:,l,t}$
  # Compute Time-Relevance Score
  $\Delta^{time}_{l,t} = |f(X_t) - f(X_t \setminus x_{:,l,t})|$
# to stabilize the temporal scaling normalize $\Delta^{time}_{:,t}$
for $l \leftarrow 0$ *to* L **do**
  for $j \leftarrow 0$ *to* J **do**
    # Mask feature j at lookback l
    $X_t \setminus x_{j,l,t} = baseline_{j,l,t}$
    # Compute Feature-Relevance
    $\Delta^{feature}_{j,l,t} = |f(X_t) - f(X_t \setminus x_{j,l,t})|$
    # Compute final importance
    $\phi_{j,l,t} = \Delta^{feature}_{j,l,t} \times \Delta^{time}_{l,t}$
return ϕ

Figure 3.11: Algoritm of WinTSR [9]

This gives a time-relevance score for each window, indicating how much the model output depends on observations at that time step. In the second step, for each window and each feature, the feature is masked individually and its relevance is computed. As the final step, we multiply time-relevance by feature relevance. For normalization, min-max scaling was used.

4. EXPERIMENTS

The dataset was split into the 75%, 15%, and 10% parts for training, validation, and testing, respectively. Records were grouped by IDs to ensure division by patient to prevent data leakage across splits. All training and evaluations were made using 2 GPU T4.

4.1. LightGBM

Model development proceeded through three stages: baseline evaluation, feature selection and hyperparameter optimization.

4.1.1. Baseline

A baseline model was trained with a minimal set of manually configured parameters. To address the severe class imbalance present in the dataset, *scale_pos_weight* was set to 12.75, computed as the ratio of negative to positive instances. All remaining LightGBM parameters were left at their library defaults. Training was 300 rounds with early stopping at 80 with the best utility 0.4309 and AUPRC 0.1168.

4.1.2. Feature Selection

Feature importance revealed that most features contributed no gain. The top 20 features by gain are visualized in Figure 4.1. ICULOS dominates by a large margin, followed by FiO2_Freq_6h, Unit1, Temp_Max_6h, Creatinine, HospAdmTime, and ShockIndex. It is a mix of raw clinical measurements, admission metadata and sliding-window aggregates.

Three feature subsets were then constructed based on cumulative gain thresholds and evaluated against the full feature set: features accounting for 99% of cumulative

gain (122 features), 95% (75 features), and features with individual gain above the mean (28 features).

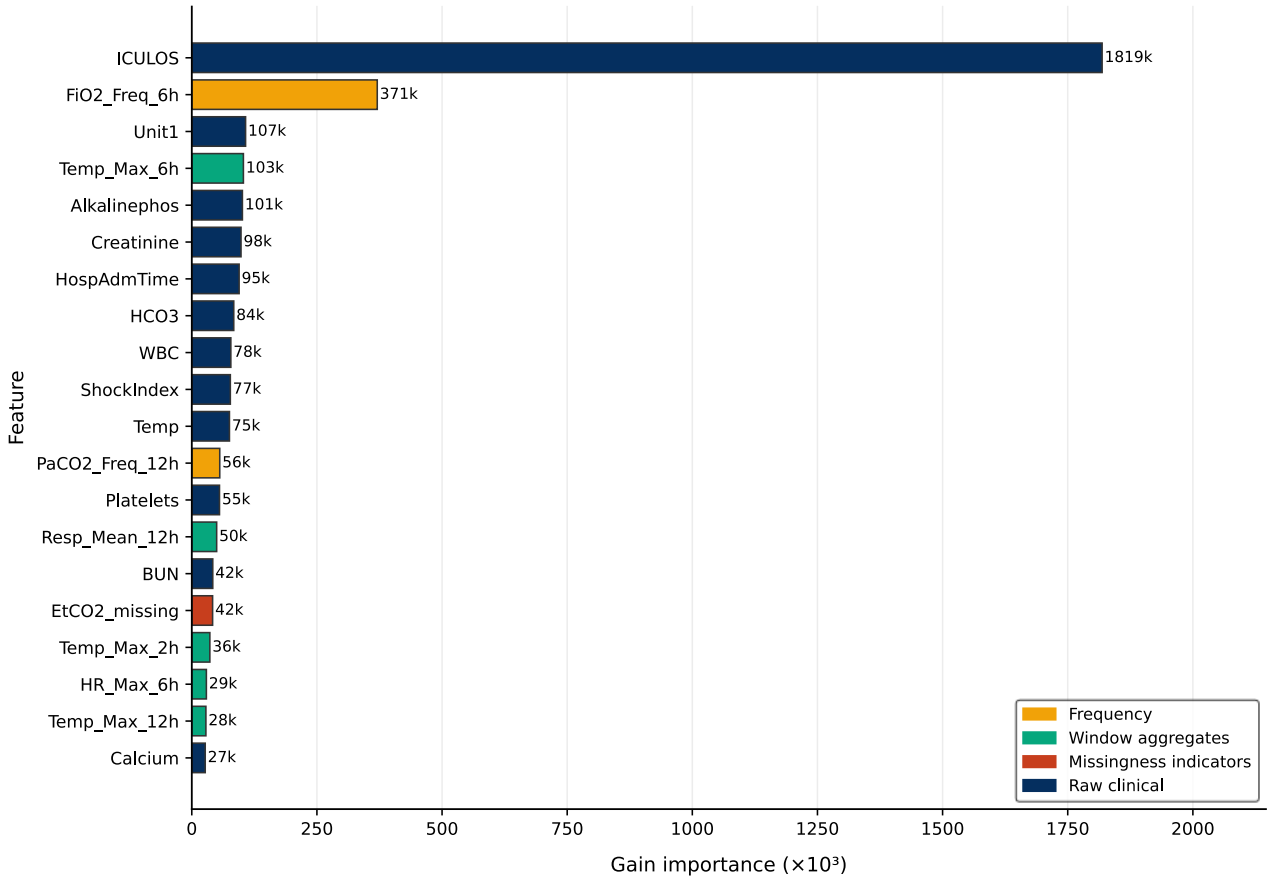

Figure 4.1: Top-20 features by gain importance in the LightGBM model

Each subset was evaluated using a decision threshold selected to maximize the utility score on the validation set. Results for all four configurations are presented in Table 4.1 (rows 1-4).

The 99% subset (122 features) achieved the highest validation AUPRC (0.1231) and utility score (0.4323). The more aggressive thresholds (95% and above-mean) both downed performance relative to the full feature set. The 122-feature subset was selected for hyperparameter optimization.

4.1.3. Hyperparameter Optimization

The number of boosting rounds was set to 1000 with early stopping applied after 30 rounds of no improvement on the validation AUPRC to prevent overfitting. Random search and Bayesian optimization were evaluated across 70 trials each, using the identical search space to allow a direct comparison of search strategies. The Optuna configuration achieved (AUPRC 0.1438), outperforming the random search configuration (AUPRC 0.1386). The full hyperparameter search space and best values are summarised in Appendix B (Table B.1). The Optuna configuration was selected as the final LightGBM configuration.

After final training on the parameters found by Optuna, the AUPRC is 0.1317, and the Utility score is 0.4614 were achieved. This is the best result on the validation set. All results of experiments are presented in Table 5.2.

Table 4.1: Experiments and results LightGBM

Feature Set	**Num of features**	**Thr**	**AUPRC**	**Utility**
All features (Baseline)	240	0.20	0.1168	0.4309
Cumulative sum 99%	122	0.20	0.1231	0.4323
Cumulative sum 95%	75	0.19	0.1183	0.4273
Gain above mean	28	0.22	0.1120	0.4230
Hyperparameter optimization	***122***	***0.21***	***0.1371***	***0.4614***

4.2. TCN

Model development followed the same three stages as LightGBM: baseline evaluation, feature selection and hyperparameter optimization.

4.2.1. Baseline

Two baseline configurations were evaluated prior to feature selection. The first used the complete 240-feature set. The second used only the 76 features comprising the original measurements and their binary missing indicators. Both models were trained with manually selected hyperparameters: four residual blocks with output channel configurations of [256, 256, 512, 512] and [128, 128, 256, 256] for the 240- and 76-feature configurations, respectively, a kernel size of 3, and a dropout rate of 0.1.

The 76-feature configuration achieved a higher validation utility score (0.4263 vs. 0.3806) and a higher validation AUPRC (0.1316 vs. 0.1185). The 76-feature set was selected as the basis for subsequent feature selection.

4.2.2. Feature Selection

Permutation importance was applied separately to both the 240- and 76-feature configurations. From the 240-feature set, 145 features with strictly positive importance were retained. From the 76-feature set, 54 features were retained. Evaluated on the validation set, the 54-feature subset achieved a utility score of 0.4369.
This result is higher than both the full 76-feature baseline (0.4263) and the 145-feature subset derived from the 240-feature configuration (0.3926). The 54-feature subset was selected for hyperparameter optimization. The top 20 features by contribution are visualized in Figure 4.2. ICULOS dominates (as for LightGBM).

4.2.3. Hyperparameter Optimization

Hyperparameter optimization was conducted over 40 trials. In the first stage, 20 trials were run over the full search space, including kernel size, channel strategy, dropout, learning rate and positive class weight. The best AUPRC score is 0.1352. The hyperparameter search for the first stage is summarized in Table B.2.

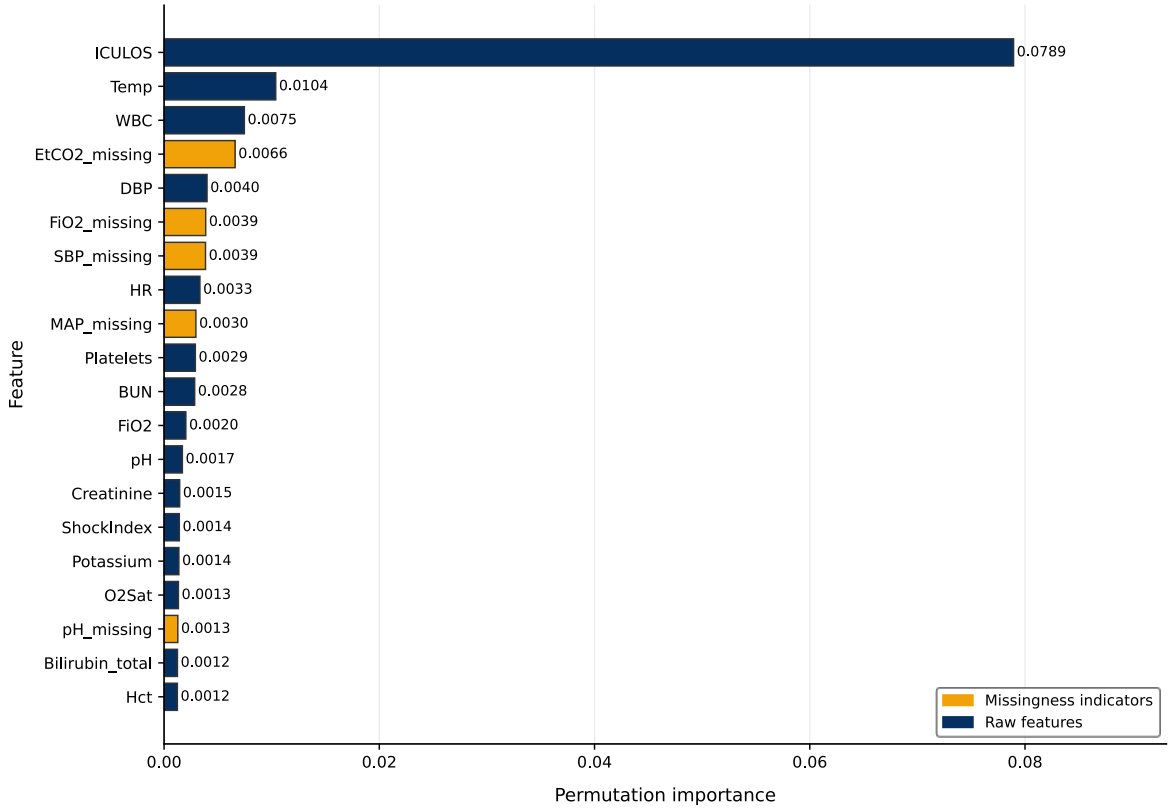

Figure 4.2: Top-20 features with positive contribution in the TCN model

In the second stage, architectural parameters (kernel size, channels) were fixed. Further 20 trials were run to refine the remaining parameters (dropout, learning rate, and positive class weight) over a narrowed search range, with median pruning applied to terminate unpromising trials early. The best AUPRC score is 0.1353. The full hyperparameter search space is summarized in Table B.3.

The final Optuna configuration achieved a validation AUPRC of 0.1316 and a utility score of 0.4266. The utility score is lower than the manually selected configuration (0.4369). The manually configured 54-feature model was retained as the final TCN configuration. All results of experiments are presented in Table 4.2.

Table 4.2: Experiments and results TCN

Feature Set	**Num of features**	**Thr**	**AUPRC**	**Utility**
All features (Baseline)	240	0.16	0.1185	0.3806
Original + Missing	76	0.18	0.1316	0.4263
Positive importance (from all features)	145	0.24	0.1294	0.3926
Positive importance (from originals + missings)	***54***	***0.22***	***0.1285***	***0.4369***
Hyperparameter optimization	54	0.24	0.1316	0.4266

4.3. Hybrid TCN-LightGBM

Three feature configurations were evaluated for the LightGBM classifier in the second stage: TCN embeddings only (128 features), embeddings concatenated with the 54 features used as TCN input (182 features total), and embeddings concatenated with the 40 raw clinical features from the original dataset (168 features total). Optuna optimization was applied to embeddings-with-raw-features configuration over 30 trials, as it achieved the highest validation AUPRC (0.1141). The search space was identical to that used for the standalone LightGBM optimisation. The best values identified are reported in Table B.4.

The results are summarized in Table 4.3**.** The optimised embeddings-with-raw-features configuration achieved the highest validation utility score among the hybrid variants (0.4473).

Table 4.3: Experiments and results LightGBM-TCN

Feature Set	**Num of features**	**Thr**	**AUPRC**	**Utility**
TCN embeddings only	128	0.27	0.1091	0.4213
Embeddings + features used for TCN	182	0.22	0.1073	0.4298
Embeddings + original features	168	0.20	0.1141	0.4252
Hyperparameter optimization	***168***	***0.23***	***0.1278***	***0.4473***

4.4. Explainability

Explainability analysis is demonstrated on patient *p000161*, a sepsis-positive case with a stay of 23 hours. The sepsis label is positive from hour 14. It corresponds to six hours before the clinical sepsis onset (hour 20) as defined by the challenge. Both models issue their first positive prediction at hour 6 (14 hours before the estimated clinical onset).

4.4.1. SHAP (LightGBM)

Figure 4.3 presents a SHAP heatmap (top-20 features) for patient p000161 across all 23 hours. *FiO2_Freq_6h* has a strong negative effect in the first 3 hours but from hour 4 it becomes strongly positive and stays the main influencing factor for the rest of the time. *Creatinine* and *Unit1* follow a similar pattern. Temperature (*Temp*) and its related measures (*Temp_Max_6h* and *Temp_Max_12h*) increase in importance after hour 4 but their effect is smaller compared to *FiO2_Freq_6h*. *ICULOS* contributes positively in early hours and turns strongly negative at hours 21-23. *Alkalinephos*, *Alkalinephos_Freq_6h* and *SBP_Std_6h* activate late with the strongest contributions at hours 20-21.

Figure 4.4 presents patient-level SHAP attributions for patient p000161 at hour 6. Among the top 15 features, 10 are raw clinical variables, 3 are window aggregates and 2 are frequency-based aggregates. *FiO2_Freq_6h* is the strongest positive contributor, followed by *Creatinine*, *Unit1* and *Temp*. Four features contribute negatively: *ShockIndex*, *Magnesium*, *HospAdmTime* and *Calcium*.

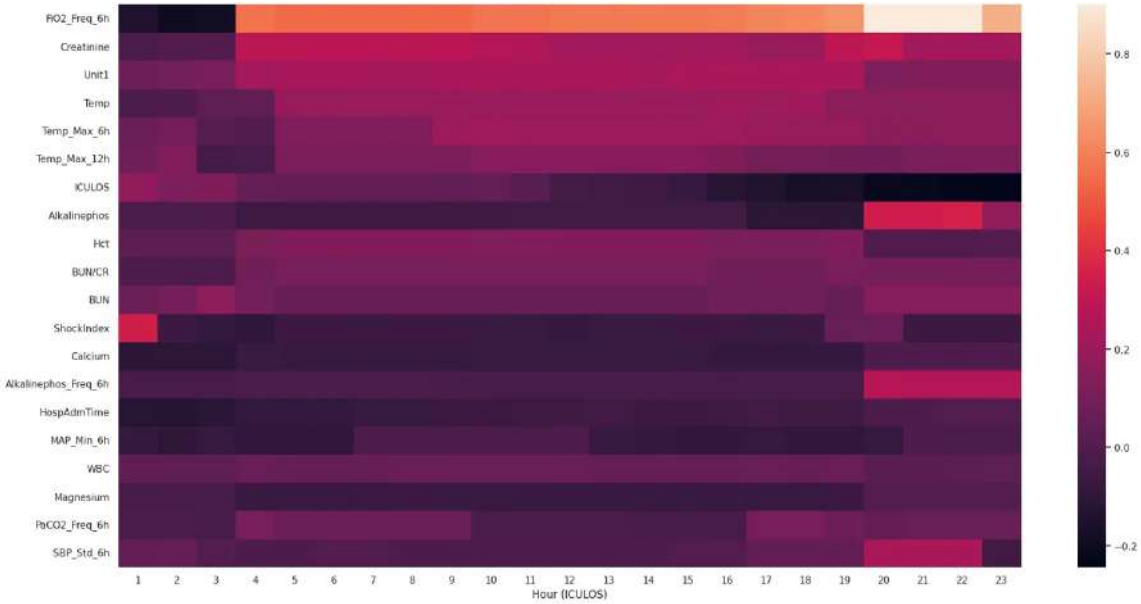

Figure 4.3: Top-20 features in SHAP heatmap for patient p000161 across all 23 hours

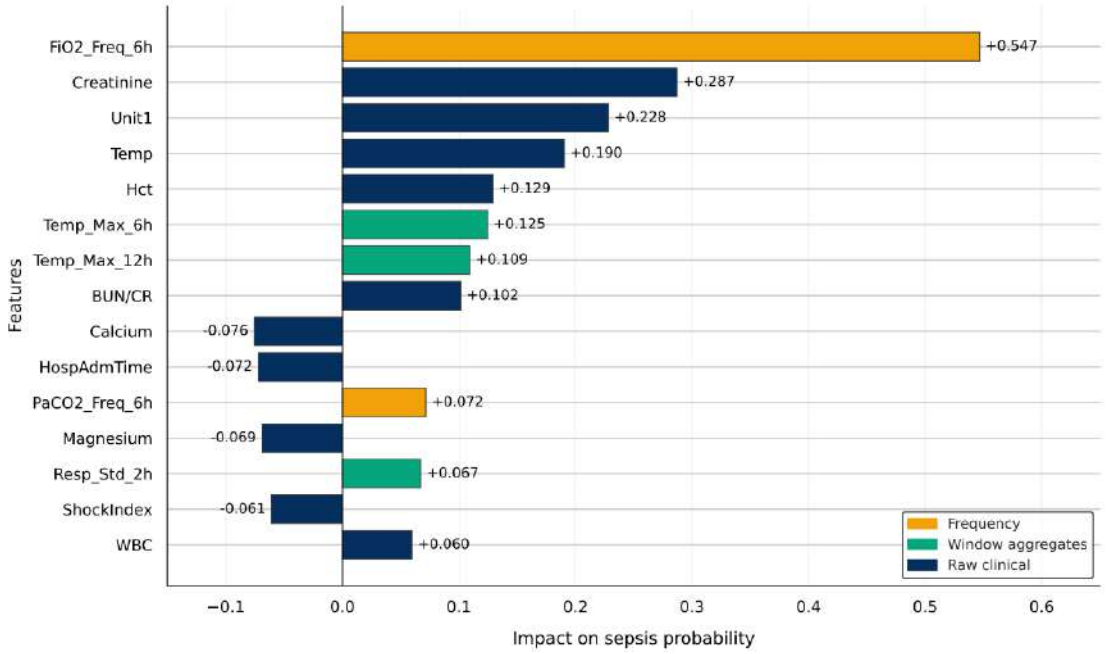

Figure 4.4: Top-15 SHAP feature importance for patient p000161 at hour 6

4.4.2. WinTSR (TCN)

Figure 4.5 presents a WinTSR heatmap (top-20 features) for patient *p000161* across all 23 hours. *EtCO2_missing* and *FiO2_missing* are the strongest features throughout the stay. Temp shows a moderate but consistent contribution. Other missingness indicators such as *Lactate_missing* and *AST_missing* are more active in early hours. After hour 19, attributions drop across all features.

Figure 4.6 presents patient-level WinTSR attributions for patient p000161 at hour 6. Among the top 15 features, 10 are missingness indicators and 5 are raw clinical variables. *EtCO2_missing* is the strongest contributor by a large margin, followed by Temp and Lactate_missing. The remaining features contribute at a similar and notably lower level.

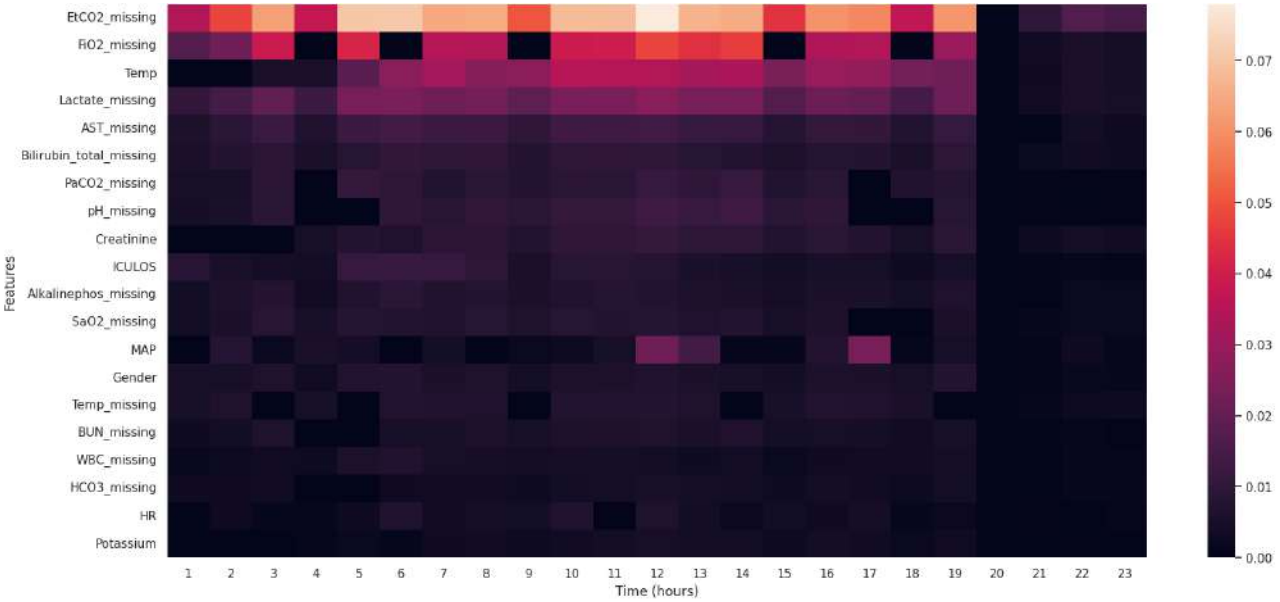

Figure 4.5: Top-20 features in WinTSR heatmap for patient p000161 across all 23 hours

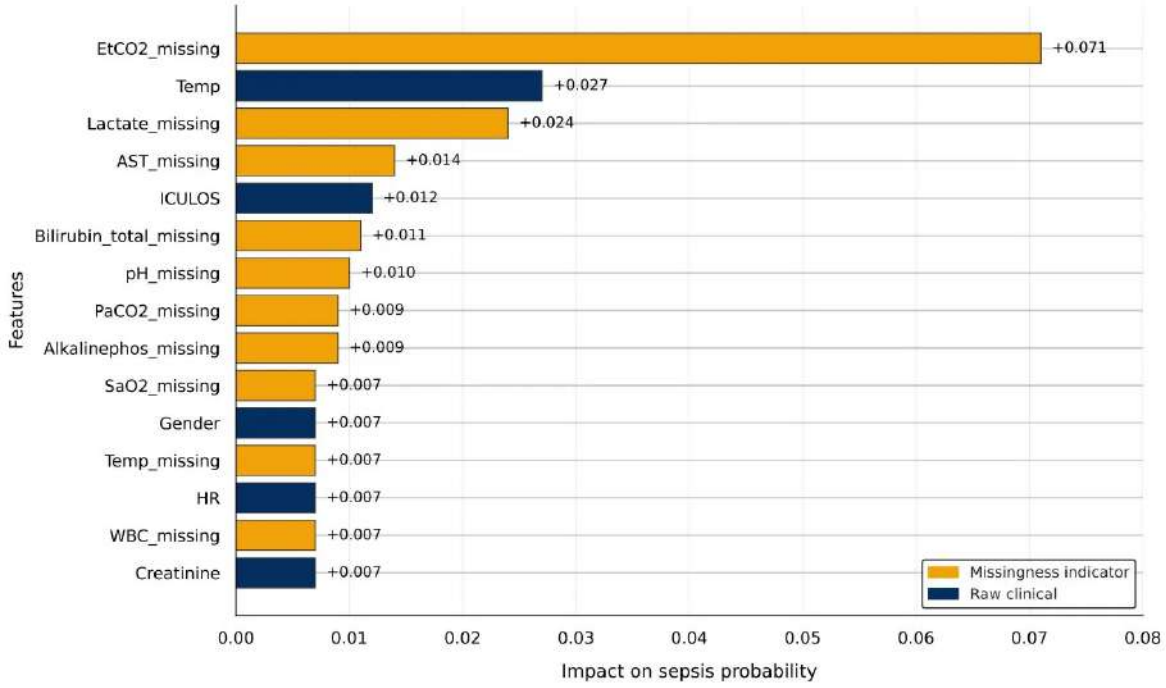

Figure 4.6: Top-15 WinTSR feature importance for patient p000161 at hour 6

5. RESULTS

5.1. Model Comparison

Table 5.1 summarizes model performance on validation and test sets. LightGBM achieves the highest utility score on both sets (0.461 validation, 0.382 test), outperforming the TCN (0.437, 0.374) and the hybrid TCN-LightGBM (0.447, 0.372). The hybrid model does not consistently improve over standalone approaches.

Table 5.1: Model performance comparison on validation and test sets

Model	**Val AUPRC**	**Val Utility**	**Test AUPRC**	**Test Utility**
LightGBM	***0.137***	***0.461***	***0.116***	***0.381***
TCN	0.129	0.437	0.105	0.374
Hybrid	0.1278	0.4473	0.1036	0.372

A performance drop is observed across all models between validation and test sets. This gap is most pronounced for LightGBM (AUPRC 0.137 vs. 0.116, utility 0.461 vs. 0.382). The drop in utility score is partly attributable to threshold sensitivity: the decision threshold was optimized on the validation set and may not transfer perfectly. The drop in AUPRC likely reflects the smaller size of the test set (10% of the data).

Table 5.2 presents the official challenge leaderboard results for the top five teams. The winning team achieved a final utility score of 0.360 across all three hospital systems (A, B, and C). Direct comparison with these results is not possible for two reasons. First, the official final score includes System C, which consistently produces negative utility scores across all teams (e.g., -0.123 for rank 1) and pulls the final score down. Second, this study uses only systems A and B. On systems A and B separately, the top teams achieved scores in the range of 0.401–0.434, which is comparable to the LightGBM result of 0.382 obtained in this study on the combined A+B test set.

Table 5.2: Clinical utility scores for the teams with the five highest scores on the full test set according to the PhysioNet/Computing in Cardiology Challenge 2019 [27]

Rank	Team	Final Score	Score A	Score B	Score C
1	James Morrill, Andrey Kormilitzin, Alejo Nevado-Holgado, Sumanth Swaminathan, Sam Howison, Terry Lyons	0.360	0.433	0.434	-0.123
2	John Anda Du, Nadi Sadr, Philip de Chazal	0.345	0.409	0.396	-0.042
3	Morteza Zabihi, Serkan Kiranyaz, Moncef Gabbouj	0.339	0.422	0.395	-0.146
4	Xiang Li, Yanni Kang, Xiaoyu Jia, Junmei Wang, Guotong Xie	0.337	0.420	0.401	-0.156
5	Janmajay Singh, Kentaro Oshiro, Raghava Krishnan, Masahiro Sato, Tomoko Ohkuma, Noriji Kato	0.337	0.401	0.407	-0.094

5.2. Demo application

A demonstration application was developed as a practical artifact of this study. Its purpose is to illustrate how early sepsis prediction with model explainability can be presented to an end user. The application was built using Streamlit [28] and deployed at https://early-sepsis-prediction-lgbm.streamlit.app/.

The LightGBM model was selected for the demo as the best-performing model in this study and the most lightweight (342 KB). The application accepts patient data in PSV format. Three demo from the dataset are provided for exploration.

Figure 5.1 shows the prediction chart for a demo patient. Sepsis risk is displayed over the duration of the ICU stay with a dashed alert threshold (0.21 for this model) line. Points are colored by predicted risk level.

Figure 5.1: The prediction chart showing sepsis risk over time (deployed demo application)

Clicking on a point opens a SHAP feature attribution chart for that hour (Figure 5.2). The chart shows which features contributed most to the risk score at that moment, with bars colored by feature type.

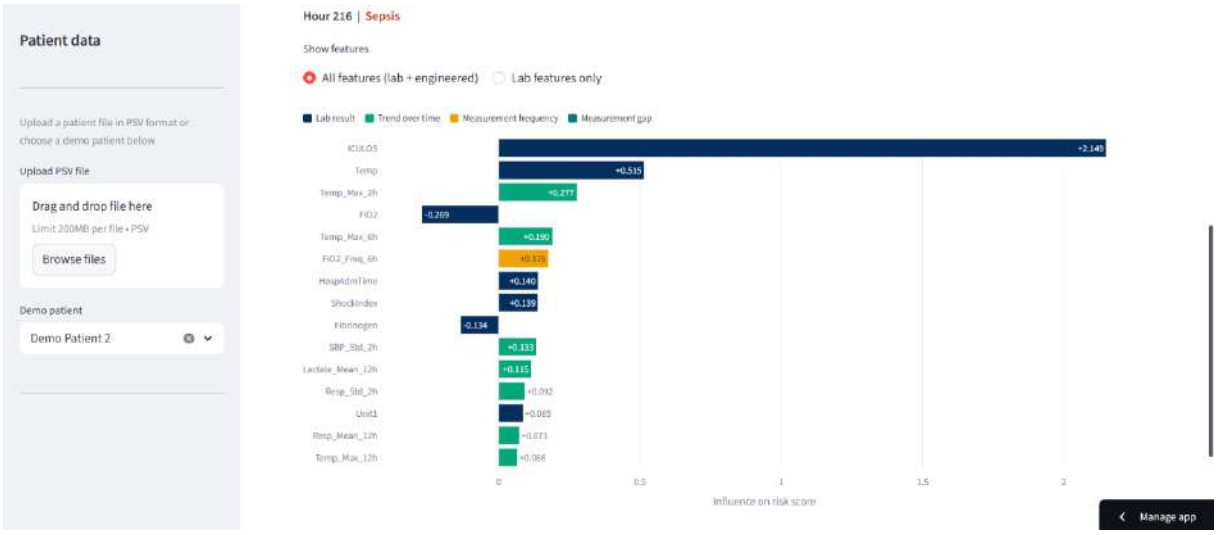

Figure 5.2: The SHAP feature attribution chart for chosen hour (deployed demo application)

Two display modes are available: all engineered features (including sliding window aggregates, measurement frequency and missingness indicators) or raw laboratory measurements only.

The application is intended as a research prototype demonstrating the output of the study rather than a clinical decision support tool.

6. DISCUSSION

This study compared three modeling approaches for early sepsis prediction from ICU clinical time series: LightGBM, TCN and hybrid TCN-LightGBM.

LightGBM outperforms both TCN and the hybrid model on all reported metrics. LightGBM used 122 features, including sliding window aggregates that encode temporal patterns explicitly. TCN achieved a competitive result using only 54 features (fewer than one third of the 122 features used by LightGBM) which suggests that deep learning approaches can be effective even with minimal feature engineering. The hybrid model did not improve over standalone LightGBM. The TCN was trained independently, so its embeddings are optimized for sequence modeling rather than tree-based classification. This is interpreted as a negative result: learned temporal embeddings did not provide additional signal beyond manually engineered features.

We find that both models rely heavily on measurement-related features. LightGBM relies on a mix of raw clinical variables, time-related features and measurement frequency aggregates. 8 of the top 20 features by gain importance value are frequency-based (one of them is missing indicator) or temporal aggregates. For TCN, the reduced 76-feature set consisting of original measurements and binary missingness indicators outperformed the full feature set, confirming that missingness patterns carry predictive signal.

Patient-level attribution analysis confirms this finding. For LightGBM, the six-hour FiO2 measurement frequency dominates. For TCN, the EtCO2 missingness indicator leads. Both variables are ordered selectively based on patient condition: EtCO2 is collected only in mechanically ventilated patients, FiO2 is collected more frequently when respiratory support is needed. For clinicians, the presence or absence of these measurements reflects decisions about care intensity, which the models learned to use as a predictive signal.

Our work has certain limitations. Results are based on a single dataset and two hospital systems, which may limit generalizability. Future work may explore joint

training strategies for hybrid architectures and multimodal extensions combining clinical time series with imaging or clinical notes.

REFERENCES

1. Singer, M., Deutschman, C. S., Seymour, C. W., et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). JAMA, 315(8):801–810, 2016.
2. Gray, A., Chung, E., Hsu, R., et al. Global, regional, and national sepsis incidence and mortality, 1990–2021: a systematic analysis. The Lancet Global Health, 13:e2013– e2026, 2025.
3. Kilinc Toker, A., Kose, S., & Turken, M. (2021). Comparison of SOFA Score, SIRS, qSOFA, and qSOFA + L Criteria in the Diagnosis and Prognosis of Sepsis. The Eurasian journal of medicine, 53(1), 40–47. https://doi.org/10.5152/eurasianjmed.2021.20081
4. Reyna, M., Josef, C., Jeter, R., Shashikumar, S., Moody, B., Westover, M. B., Sharma, A., Nemati, S., & Clifford, G. D. (2019). Early Prediction of Sepsis from Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019 (version 1.0.0). PhysioNet. RRID:SCR_007345. https://doi.org/10.13026/v64v-d857
5. Janmajay Singh, Kentaro Oshiro, Raghava Krishnan, Masahiro Sato, Tomoko Ohkuma, and Noriji Kato. Utilizing informative missingness for early prediction of sepsis. In 2019 Computing in Cardiology (CinC), pages 1–4. IEEE, 2019.
6. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.
7. Bai, S., Kolter, J. Z. & Koltun, V. (2018). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. CoRR, abs/1803.01271.
8. Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

9. Islam, Md Khairul, & Fox, Judy. A Windowed Temporal Saliency Rescaling Method for Interpreting Time Series Deep Learning Models. Retrieved from https://par.nsf.gov/biblio/10583568.
10. J. Morrill, A. Kormilitzin, A. Nevado-Holgado, S. Swaminathan, S. Howison and T. Lyons, "The Signature-Based Model for Early Detection of Sepsis From Electronic Health Records in the Intensive Care Unit," 2019 Computing in Cardiology (CinC), Singapore, 2019, pp. Page 1-Page 4, doi: 10.22489/CinC.2019.014.
11. Johnson, A., Bulgarelli, L., Pollard, T., Gow, B., Moody, B., Horng, S., Celi, L. A., and Mark, R. MIMIC-IV. PhysioNet, 2024. doi: 10.13026/kpb9-mt58. Version 3.1.
12. Du, W., Cote, D., and Liu, Y. SAITS: Self-Attention-based Imputation for Time Series. Expert Systems with Applications, 219:119619, 2023. doi: 10.1016/j.eswa.2023. 119619.
13. Zhang, X., Zeman, M., Tsiligkaridis, T., and Zitnik, M. Graph-guided network for irregularly sampled multivariate time series. In International Conference on Learning Representations, 2022.
14. Bento, J., Saleiro, P., Cruz, A. F., Figueiredo, M. A. T., and Bizarro, P. TimeSHAP: Explaining recurrent models through sequence perturbations. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 2565–2573, 2021. doi: 10.1145/ 3447548.3467166.
15. Tonekaboni, S., Joshi, S., Campbell, K., Duvenaud, D. K., and Goldenberg, A. What went wrong and when? Instance-wise feature importance for time-series blackbox models. In Advances in Neural Information Processing Systems, volume 33, pp. 799–809, 2020.
16. Cox, N. J., & Jones, K. (1981). Exploratory data analysis. Quantitative Geography, London: Routledge, 135-143.
17. Donald B. Rubin, Inference and missing data, Biometrika, Volume 63, Issue 3, December 1976, Pages 581–592, https://doi.org/10.1093/biomet/63.3.581

18. Henry KE, Hager DN, Pronovost PJ, Saria S. A targeted real-time early warning score (TREWScore) for septic shock. Science Translational Medicine August 2015; 7(299):299ra122–299ra122. ISSN 1946-6234, 1946-6242.
19. Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785
20. Progress and perspectives on genomic selection models for crop breeding - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/An-overview-of-LightGBM57-LightGBM-includes-the-GOSS-EFB-histogram-based-feature_fig3_390605031
21. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
22. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734. https://doi.org/10.3115/v1/D14-1179
23. Francesco Lässig. (2021, July 6). Temporal convolutional networks and forecasting. Unit8. Available from: https://unit8.com/resources/temporal-convolutional-networks-and-forecasting/
24. Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. 2019. “All Models Are Wrong, but Many Are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously.” Journal of Machine Learning Research: JMLR 20: 177. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8323609/.
25. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13, 281–305.

26. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2623–2631. https://doi.org/10.1145/3292500.3330701
27. Reyna, M. A., Josef, C. S., Jeter, R., Shashikumar, S. P., Westover, M. B., Nemati, S., Clifford, G. D., & Sharma, A. (2020). Early prediction of sepsis from clinical data: The PhysioNet/Computing in Cardiology Challenge 2019. Critical Care Medicine, 48(2), 210–217. https://doi.org/10.1097/CCM.0000000000004145
28. Streamlit. (2024). Streamlit: A faster way to build and share data apps (Version 1.55.0) [Software]. https://streamlit.io
29. Y. Chang et al., "A Multi-Task Imputation and Classification Neural Architecture for Early Prediction of Sepsis from Multivariate Clinical Time Series," 2019 Computing in Cardiology (CinC), Singapore, 2019, pp. Page 1-Page 4, doi: 10.22489/CinC.2019.110.
30. J. A. Du, N. Sadr and P. d. Chazal, "Automated Prediction of Sepsis Onset Using Gradient Boosted Decision Trees," 2019 Computing in Cardiology (CinC), Singapore, 2019, pp. Page 1-Page 4, doi: 10.22489/CinC.2019.423.

APPENDIX A

This appendix provides descriptions of all clinical variables included in the PhysioNet/Computing in Cardiology Challenge 2019 dataset. Tables A.1, A.2 and A.3 list vital signs, laboratory values and demographic variables respectively.

Table A.1: Vital signs [4]

Variable	**Description**
HR	Heart rate (beats per minute)
O2Sat	Pulse oximetry (%)
Temp	Temperature (Deg C)
SBP	Systolic BP (mm Hg)
MAP	Mean arterial pressure (mm Hg)
DBP	Diastolic BP (mm Hg)
Resp	Respiration rate (breaths per minute)
EtCO2	End tidal carbon dioxide (mm Hg)

Table A.2: Laboratory values [4]

Variable	**Description**
BaseExcess	Measure of excess bicarbonate (mmol/L)
HCO3	Bicarbonate (mmol/L)
FiO2	Fraction of inspired oxygen (%)
pH	N/A
PaCO2	Partial pressure of carbon dioxide from arterial blood (mm Hg)
SaO2	Oxygen saturation from arterial blood (%)
AST	Aspartate transaminase (IU/L)
BUN	Blood urea nitrogen (mg/dL)
Alkalinephos	Alkaline phosphatase (IU/L)

Calcium	(mg/dL)
Chloride	(mmol/L)
Creatinine	(mg/dL)
Bilirubin_direct	Bilirubin direct (mg/dL)
Glucose	Serum glucose (mg/dL)
Lactate	Lactic acid (mg/dL)
Magnesium	(mmol/dL)
Phosphate	(mg/dL)
Potassium	(mmol/L)
Bilirubin_total	Total bilirubin (mg/dL)
TroponinI	Troponin I (ng/mL)
Hct	Hematocrit (%)
Hgb	Hemoglobin (g/dL)
PTT	partial thromboplastin time (seconds)
WBC	Leukocyte count (count*10^3/µL)
Fibrinogen	(mg/dL)
Platelets	(count*10^3/µL)

Table A.3: Demographics [4]

Variable	**Description**
Age	Years (100 for patients 90 or above)
Gender	Female (0) or Male (1)
Unit1	Administrative identifier for ICU unit (MICU)
Unit2	Administrative identifier for ICU unit (SICU)
HospAdmTime	Hours between hospital admit and ICU admit
ICULOS	ICU length-of-stay (hours since ICU admit)

APPENDIX B

This appendix provides the full hyperparameter search spaces and the best values selected during hyperparameter optimization. Table B.1 covers LightGBM, B.2 and B.3 – TCN (Phase 1) and TCN (Phase 2) respectively. Table B.4 lists best values for the hybrid model (the hyperparameter search space is the same as for LightGBM).

Table B.1: The full hyperparameter search space and best values for LightGBM

Parameter	**Type**	**Search range**	**Best value**
max_bin	Integer	[64, 512]	512
lambda_l1	Float (log)	[1e-8, 10.0]	1.2958264081350343e-07
lambda_l2	Float (log)	[1e-8, 10.0]	6.151037127082509e-05
max_depth	Integer	[3, 15]	4
feature_fraction	Float	[0.4, 1.0]	0.9522861735245841
min_data_in_leaf	Integer	[20, 150]	94
learning_rate	Float (log)	[0.005, 0.1]	0.07227593260441952
scale_pos_weight	Float	[10, 15]	11.379394665628105
min_gain_to_split	Float	[0.0, 1.0]	0.35449824214080233
top_rate	Float	[0.1, 0.3]	0.204634181603536
other_rate	Float	[0.05, 0.2]	0.05349474656893903

Table B.2: The full hyperparameter search space and best values for TCN (Phase 1)

Parameter	**Type**	**Search range**	**Best value**
kernel_size	Integer	[3, 5, 7]	3
channels	Categorical	[64, 64, 128, 128], [64, 96, 128, 192], [64, 128, 256, 256]	[64, 64, 128, 128]

dropout	Float	[0.05, 0.3]	0.3
learning_rate	Float (log)	[1e-5, 1e-3]	1.787e-4
pos_weight	Float	[10.0, 15.0]	12.12

Table B.3: The full hyperparameter search space and best values for TCN (Phase 2)

Parameter	**Type**	**Search range**	**Best value**
dropout	Float	[0.22, 0.35]	0.32
learning_rate	Float (log)	[8e-5, 2.5e-4]	1.005e-4
pos_weight	Float	[11.5, 13.0]	11.7

Table B.4: The full hyperparameter best values for the hybrid model

Parameter	**Type**
max_bin	165
lambda_l1	7.814e-2
lambda_l2	2.709e-8
max_depth	4
feature_fraction	0.9704
min_data_in_leaf	106
learning_rate	0.0134
scale_pos_weight	11.98
min_gain_to_split	0.6434
top_rate	0.1869
other_rate	0.0994

APPENDIX C

This appendix provides hour-by-hour prediction tables for patient p000161. Tables C.1 and C.2 show the annotated sepsis label, predicted probability and binary prediction for LightGBM and TCN respectively.

Table C.1: LightGBM's predictions

Hour	Annotated onset	Probability	Prediction
1	0	0.0886	0
2	0	0.0531	0
3	0	0.0472	0
4	0	0.1527	0
5	0	0.2095	0
6	0	0.2104	1
7	0	0.2399	1
8	0	0.2350	1
9	0	0.2409	1
10	0	0.2462	1
11	0	0.2369	1
12	0	0.2231	1
13	0	0.2159	1
14	1	0.2163	1
15	1	0.1969	0
16	1	0.1970	0
17	1	0.2290	1
18	1	0.2187	1
19	1	0.2496	1
20	1	0.5999	1

21	1	0.5563	1
22	1	0.5805	1
23	1	0.4004	1

Table C.2: TCN's predictions

Hour	**Annotated onset**	**Probability**	**Prediction**
1	0	0.1206	0
2	0	01010	0
3	0	0.0937	0
4	0	0.1294	0
5	0	0.1471	0
6	0	0.2388	1
7	0	0.2467	1
8	0	0.2275	1
9	0	0.2517	1
10	0	0.2237	1
11	0	0.2261	1
12	0	0.1902	0
13	0	0.1992	0
14	1	0.1780	0
15	1	0.2308	1
16	1	0.1892	0
17	1	0.2218	1
18	1	0.2954	1
19	1	0.2244	1
20	1	0.5437	1

21	1	0.4455	1
22	1	0.3722	1
23	1	0.3181	1