APSIPA Transactions on Signal and Information Processing, 2025, 14, e15

Original Paper

Robust ICU Mortality Prediction with Multi-Task Diffusion and Contrastive Learning Frameworks

Namtip Buranaburustam[1], Wuttipong Kumwilaisak[1], Chatchawarn Hansakunbuntheung[2], Nattanun Thatphithakkul[2*] and Kanya Kumwilaisak[3]

[1] *Electronics and Telecommunication Department, King Mongkut's University of Technology Thonburi, Bangkok, Thailand*
[2] *National Science and Technology Development Agency, Pathum Thani, Thailand*
[3] *Faculty of Medicine, Chulalongkorn University, Bangkok, Thailand*

ABSTRACT

Predicting death in the intensive care unit (ICU) plays an important role in clinical decision-making and patient care to increase hospital performance and help to communicate with patients and families about treatment decisions on time. Machine learning and deep learning have been used. Widely used in ICU patient data to predict mortality. The data are usually time series data, which have common data problems such as missing values and imbalance of classification. This paper presents a Multi-Task Diffusion Model (MTDM) designed to address the dual challenges of missing data and mortality prediction in ICU settings. The Multi-Task Diffusion Model (MTDM) introduces an innovative approach by integrating diffusion models for high-fidelity imputation of incomplete clinical time-series data and an LSTM network for mortality prediction, capturing temporal dependencies. By unifying imputation

*Corresponding author: nattanun.tha@nstda.or.th

Received 19 November 2024; revised 25 April 2025; accepted 01 May 2025
ISSN 2048-7703; DOI 10.1561/116.20240085

and prediction tasks, the MTDM ensures seamless optimization, addressing challenges such as noisy and missing data. Furthermore, the Siamese network with contrastive loss enhances feature representation by distinguishing between patient profiles with similar and dissimilar outcomes, enabling nuanced clinical insights. A feedback mechanism between the imputation and prediction models ensures joint optimization, improving overall performance even in the presence of noisy or incomplete data. The proposed Multi-Task Diffusion Model (MTDM) demonstrated superior imputation accuracy across varying missing data rates and achieved state-of-the-art performance in mortality prediction when evaluated on the Medical Information Mart for Intensive Care III (MIMIC-III) dataset, Medical Information Mart for Intensive Care IV (MIMIC-IV), and eICU Collaborative Research Database, underlining its robustness and efficacy for critical care applications. The experimental results confirm that integrating diffusion-based imputation with predictive modeling enhances the robustness and reliability of outcomes. The MTDM framework offers a comprehensive solution for ICU mortality prediction, addressing both data quality issues and predictive accuracy to support critical care decision-making.

Keywords: Missing data imputation, mortality prediction, diffusion model, multi-task learning, contrastive learning

1 Introduction

The Intensive Care Unit (ICU) is a vital hospital department that provides specialized care for patients with critical and life-threatening conditions. Accurate mortality prediction for ICU patients plays a crucial role in guiding clinical decisions, optimizing resource allocation, and improving patient outcomes. Reliable predictions help healthcare professionals tailor treatment strategies, allocate resources such as staff and medical equipment, and ensure ICU beds are reserved for high-risk patients. Additionally, these predictions facilitate transparent communication with patients and their families, helping them make informed decisions about care and manage expectations regarding potential outcomes.

Over the years, several severity scoring systems have been developed to assess ICU performance and estimate patient mortality risk. Prominent examples include the Acute Physiology and Chronic Health Evaluation (APACHE) [14], the Simplified Acute Physiology Score (SAPS) [16], and the Sepsis-related

Organ Failure Assessment (SOFA) [26]. These scoring systems rely on predefined clinical variables to assess patient prognosis. However, their static nature limits their adaptability to the complexities of real-world clinical data, which often contain irregularities and missing values.

To overcome these limitations, machine learning (ML) techniques such as eXtreme Gradient Boosting (XGB), K-Nearest Neighbor (KNN), and Random Forest (RF) have been explored for predicting mortality more accurately [1, 6, 22, 24]. These models leverage large datasets to learn complex patterns beyond the scope of traditional scoring systems. Comparative studies show that ML models outperform traditional scoring systems in predictive accuracy [12, 15], demonstrating their potential to enhance clinical decision-making and ICU management.

Deep learning (DL) techniques have further advanced the field of predictive healthcare by delivering state-of-the-art results for classification and prediction tasks. Models such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and other architectures have shown great promise in handling time-series data, capturing temporal dependencies, and improving prediction performance. Integrating DL into healthcare systems has enabled clinicians to analyze vast amounts of clinical data, leading to more personalized treatment plans, improved patient outcomes, and adaptive decision-support systems.

A major challenge in predictive modeling is the issue of missing data, which is common in clinical datasets and can degrade model performance. Traditional imputation techniques, such as mean imputation, k-nearest neighbors (KNN) [20], and matrix factorization, have been widely used to address this issue. While these methods provide simple and computationally efficient solutions, they often fail to capture the underlying relationships in the data, particularly for time-series datasets. Recent advancements in imputation methods leverage machine learning and deep learning models to better reconstruct missing data [4, 18, 28]. Recently, Diffusion Models have emerged as powerful tools for high-fidelity data imputation by iteratively refining noisy inputs [23, 25, 29]. These models capture intricate dependencies within the data by conditioning on observed values, making them particularly effective in healthcare applications where missing data is prevalent. Incorporating diffusion models into predictive frameworks enhances the robustness and reliability of predictions by reducing information loss and minimizing bias.

In this paper, we propose a Multi-Task Diffusion Model (MTDM), an end-to-end framework designed to overcome these limitations. By combining diffusion models for high-fidelity imputation with LSTM networks for temporal prediction, the MTDM ensures seamless integration between these processes. This approach uniquely addresses the dual challenges of incomplete data and mortality prediction, leveraging iterative imputation to enhance data quality while preserving temporal dependencies critical for accurate predictions. As

shown in Figure 1, the workflow begins with raw clinical time-series data being fed into the diffusion model, x_1 and x_2 are the random shuffling data pairs from preprocessed data, which imputes missing values through iterative noise prediction. The imputed data is then passed to the mortality prediction model, where an LSTM network captures temporal dependencies and predicts patient outcomes (mortality or survival). Our approach leverages diffusion models to ensure accurate data imputation. To further enhance feature representation, the MTDM employs a Siamese network with contrastive loss, which distinguishes between similar and dissimilar patient outcomes. Unlike conventional models that rely on less dynamic feature extraction methods, this architecture captures subtle yet clinically significant differences between patient profiles. For instance, patients with similar symptoms but differing mortality outcomes can be effectively distinguished, leading to more personalized and accurate predictions. This end-to-end design ensures robust performance, even when faced with incomplete or noisy data, by continuously refining both imputation and prediction processes.

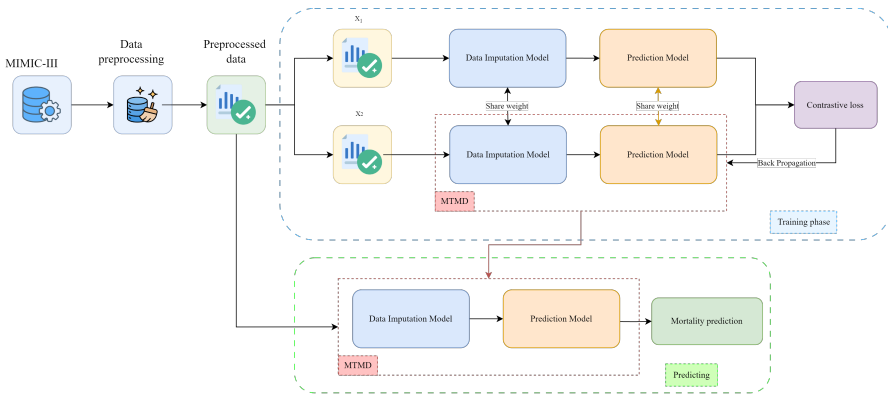


Figure 1: Overall diagram of our proposed architecture.

The key contributions of this paper are:

1. Unified Multi-Task Diffusion Framework: We propose an end-to-end MTDM that combines data imputation and mortality prediction, enabling seamless integration of both tasks.

2. Siamese Network with Contrastive Learning: We enhance feature extraction and representation learning by employing a Siamese network architecture with contrastive loss to distinguish between similar and dissimilar patient profiles.

3. Robust Handling of Missing Data: Our model utilizes diffusion-based imputation, ensuring reliable data reconstruction even with high missing data rates, which enhances the accuracy and reliability of predictions.

The remainder of this paper is organized as follows: Section 2 reviews related work on ICU mortality prediction, machine learning, deep learning, and diffusion models. Section 3 discusses the dataset and data preprocessing techniques, including handling missing data and class imbalance. Section 4 details the imputation of missing data in the ICU Dataset using Diffusion model. Section 5 describes the proposed mortality rate prediction with LSTM . Section 6 presents the contrastive learning framework with multi-task cost functions to optimize imputation and mortality rate prediction simultaneously. Section 7 presents the experimental setup, evaluation metrics, and performance comparisons with state-of-the-art models. Finally, Section 8 concludes the paper.

2 Related Work

Recent studies have explored both machine learning and deep learning approaches to predict mortality among ICU patients, utilizing various methodologies to improve clinical outcomes and decision-making processes.

2.1 Machine Learning for ICU Mortality Prediction

Machine learning models have been extensively applied to predict mortality in ICU patients [30, 2, 5, 17]. Techniques such as Support Vector Machines (SVM), Linear Discriminant Analysis (LDA), Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN), the Cox-Proportional Hazards (CPH) model, and the Fuzzy ARTMAP model have demonstrated effectiveness in leveraging clinical data for predictive purposes. These models aim to facilitate early mortality prediction, assist in optimizing treatment interventions, and enhance clinical resource management.

Performance evaluations using datasets such as the Medical Information Mart for Intensive Care (MIMIC) database indicate that these models can achieve high predictive accuracy, often comparable to traditional clinical scoring systems. By incorporating diverse patient attributes and time-series data, these models offer valuable insights that aid clinicians in making informed decisions, thereby improving patient outcomes and better allocating ICU resources.

2.2 Deep Learning Models for ICU Mortality Prediction

In recent years, deep learning approaches have gained attraction in the ICU setting, particularly for mortality prediction. Wang and Bi [27] introduced

multiple deep learning models, including an RNN-LSTM-based architecture using features similar to those in the Simplified Acute Physiology Score (SAPS II). Their experiments demonstrated strong predictive performance across key metrics, including precision, recall, F1-score, and area under the receiver operating characteristic curve (ROC-AUC).

Khaneja *et al.* [13] applied a discriminative neural network to assess the risk of coronary heart disease (CHD), leveraging deep learning techniques to identify complex patterns in patient data. Their work demonstrated improved prediction accuracy by uncovering latent relationships between clinical variables, illustrating the potential of deep learning in cardiovascular disease prediction and risk assessment.

2.3 Data Imputation

Effective data imputation is critical in handling missingness within ICU datasets, as incomplete data can significantly degrade model performance. Existing literature provides valuable approaches to addressing this challenge:

Lipton *et al.* [18]: This work introduced LSTM-based models capable of learning sequential dependencies in ICU data. Although effective in diagnosis and classification tasks, the model treated missing values as noise, limiting its capability to leverage the informativeness of missingness for imputation.

Che *et al.* [4]: This study proposed RNNs specifically designed for multivariate time series with missing values. The approach introduced masking and imputation gating mechanisms to dynamically incorporate the significance of missing data. By treating missingness as an informative signal, this method demonstrated substantial improvements in both imputation and downstream predictive tasks.

Younis *et al.* [28]: Focusing on interpretability, this research utilized a CNN-based framework for multivariate time-series analysis. Although not explicitly targeting imputation, the robust convolutional structure allowed implicit handling of missing data, emphasizing feature importance in classification tasks.

2.4 Addressing Missing Data with Diffusion Models

A persistent challenge in clinical data is the presence of missing values, which can severely impact the performance of predictive models. Diffusion models have recently emerged as a promising solution for high-fidelity data generation and imputation.

Tashiro *et al.* [25] introduced Conditional Score-based Diffusion Models for Imputation (CSDI), which represent a significant advancement in time-series imputation. Their approach leverages score-based diffusion models con-

ditioned on observed data to exploit correlations effectively, making it particularly useful for applications in healthcare and finance.

Building on this, Seki *et al.* [23] developed a diffusion model specifically designed for imputing missing values in time-series microbiome datasets. Their method demonstrated improved predictive accuracy, particularly in microbiome datasets such as 16S rRNA, highlighting the versatility of diffusion models in biological data analysis.

Zhao *et al.* [29] proposed an end-to-end mortality prediction framework for shock patients using a multi-task oriented diffusion model (MODM). This framework integrates data imputation with mortality prediction, addressing the limitations of traditional two-stage approaches. By unifying imputation and prediction within a single model, MODM offers a more robust solution for scenarios involving incomplete data, significantly enhancing predictive performance in critical care settings.

Overall, the advancements in both machine learning and deep learning, along with the integration of diffusion models for imputation, mark significant progress in ICU mortality prediction. These approaches not only offer improved predictive accuracy but also address challenges related to missing data and irregular sampling. As the field continues to evolve, integrating these technologies will be crucial in developing more reliable and actionable models for clinical decision-making and patient care.

3 Dataset and Preprocessing Methodology

In this study, we utilized three widely recognized datasets: the Medical Information Mart for Intensive Care III (MIMIC-III), the Medical Information Mart for Intensive Care IV (MIMIC-IV), and the eICU Collaborative Research Database, Figure 2 provides a comprehensive overview of the data extraction and preprocessing workflow employed in this study. The Medical Information Mart for Intensive Care III (MIMIC-III) dataset [11], a publicly available and extensively used clinical dataset, contains detailed information on more than 51,000 ICU stays from over 42,000 unique patients admitted to critical care units between 2001 and 2012. It provides a wealth of data, including demographics, vital signs, laboratory results, medications, and outcomes, making it a cornerstone for research in healthcare informatics, predictive modeling, and machine learning applications. This dataset is particularly valuable for its inclusion of time-series data and comprehensive documentation, enabling in-depth analysis of mortality prediction, length of stay, and other critical care-related phenomena.

The Medical Information Mart for Intensive Care IV (MIMIC-IV) dataset [10], offering updated clinical data from ICU stays between 2008 and 2019. It includes over 70,000 ICU admissions from the Beth Israel Deaconess Medical Center in Boston, Massachusetts. MIMIC-IV introduces improved data stan-

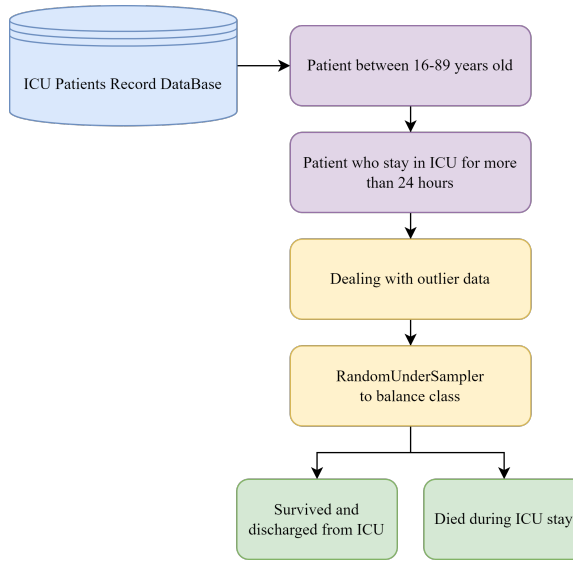


Figure 2: Schematic representation of the data extraction and preprocessing pipeline.

dardization, additional features, and updated patient care protocols, reflecting modern clinical practices. This dataset is particularly well-suited for examining the evolution of ICU care over time, enabling researchers to analyze trends and evaluate the performance of predictive models in contemporary healthcare settings. By integrating structured clinical data with unstructured notes, MIMIC-IV further supports advanced studies, such as natural language processing for clinical decision-making.

The eICU Collaborative Research Database [21], in contrast, aggregates data from over 200,000 ICU admissions across more than 200 hospitals in the United States. The eICU database captures a broader and more diverse patient population, providing a multi-center perspective on critical care. This dataset includes detailed information on diagnoses, interventions, vital signs, and outcomes, allowing for robust external validation of models developed on MIMIC datasets. Its multi-center nature makes it invaluable for studying variations in care practices and outcomes across institutions, which is essential for developing generalizable predictive models. Together, these datasets form a comprehensive foundation for investigating and improving critical care decision-making.

3.1 Data Extraction and Patient Cohort Selection

The raw consists of a range of different ages, medical conditions, and clinical protocols. To ensure relevance and consistency in the cohort for analysis, we applied the following rigorous cohort selection process.

3.1.1 Age Restriction

We define a set $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$, where each element p_i corresponds to a patient with an associated ICU stay. The first filtering criterion involved selecting patients based on their age. Let $A(p_i)$ denote the age of patient p_i. We applied the following age range restriction:

$$16 \leq A(p_i) \leq 89, \quad \forall p_i \in \mathcal{P}. \tag{1}$$

Patients below 16 years of age were excluded because they belong to a pediatric population, which follows different clinical protocols. Patients older than 89 years were also excluded due to potential data reliability issues, as older patients often have incomplete or less consistent medical records.

3.1.2 ICU Stay Duration

For each patient p_i, let $D(p_i)$ represent the duration of their ICU stay. We excluded patients with short ICU stays that lasted less than 24 hours, as these records often lack sufficient clinical data for meaningful analysis. The cohort is filtered according to the condition:

$$D(p_i) \geq 24 \text{ hours}, \quad \forall p_i \in \mathcal{P}. \tag{2}$$

3.1.3 Selected Variables

For each patient in the final cohort, we selected 16 relevant variables from the ICU dataset, consisting of both static demographic information and dynamic time-series measurements recorded during the first 24 hours of ICU admission. The variables are as follows:

- **Patient Information**: This includes ICU stay ID, age, gender, ethnicity, and a binary hospital mortality flag (indicating whether the patient survived or died during hospitalization).
- **Time-Series Variables**: The clinical time-series variables include heart rate (HR), temperature (T), systolic blood pressure (SBP), diastolic blood pressure (DBP), mean arterial pressure (MAP), respiratory rate (RR), oxygen saturation (SpO_2), glucose (G), b (Hb), potassium (K^+), and sodium (Na^+).

3.2 Data Preprocessing

Preprocessing is a critical phase in data analysis to ensure that the dataset is clean, free of erroneous values, and suitable for model training. The raw clinical data from ICU dataset contains several challenges, such as outliers, missing values, and class imbalance, each of which needs to be addressed systematically.

3.2.1 Outlier Detection and Correction

Outliers are data points that deviate significantly from the central distribution of the data. For time-series variables, such deviations could be caused by measurement errors or temporary clinical abnormalities. Outliers can distort the statistical properties of the data, and thus, their detection and treatment are necessary.

We employed the *interquartile range* (IQR) method for detecting outliers. For each variable x, the first quartile $Q_1(x)$ and the third quartile $Q_3(x)$ are computed, and the IQR is defined as:

$$\mathrm{IQR}(x) = Q_3(x) - Q_1(x). \tag{3}$$

Data points falling outside the range $[Q_1(x) - 1.5 \cdot \mathrm{IQR}(x), Q_3(x) + 1.5 \cdot \mathrm{IQR}(x)]$ are flagged as outliers:

$$\mathrm{Outliers}(x) = \{x_i \mid x_i < Q_1(x) - 1.5 \cdot \mathrm{IQR}(x) \text{ or } x_i > Q_3(x) + 1.5 \cdot \mathrm{IQR}(x)\} \tag{4}$$

To address outliers, we applied forward-backward imputation, wherein each outlier value is replaced with the nearest valid non-outlier value from adjacent time points. Let x_t represent the value at time step t, and let $x_{t'}$ be the nearest non-outlier value before or after time step t. The imputed value $\hat{x}_t$ is given by:

$$\hat{x}_t = \begin{cases} x_t & \text{if } x_t \text{ is not an outlier,} \\ x_{t'} & \text{if } x_t \text{ is an outlier.} \end{cases} \tag{5}$$

This ensures that the time-series data remains smooth and representative of the clinical trends, without being influenced by extreme or erroneous values.

3.2.2 Handling Class Imbalance in Mortality Prediction

The task of predicting patient mortality in the ICU is modeled as a binary classification problem, with label 0 representing patients who survived and label 1 representing patients who died. However, the dataset exhibits a significant class imbalance, with the majority of patients having survived.

Let N_0 represent the number of patients who survived and N_1 represent the number of patients who died. Example with the MIMIC-III dataset: Initially,

the dataset contains $N_0 = 42,692$ survivors and $N_1 = 5,141$ non-survivors, leading to an imbalanced ratio:

$$\text{Imbalance Ratio} = \frac{N_0}{N_1} \approx 8.3 : 1. \tag{6}$$

This imbalance can lead to biased model training, where the model becomes skewed towards predicting the majority class (survivors). To mitigate this, we apply *undersampling* to balance the dataset.

Let $\mathcal{D}_0$ and $\mathcal{D}_1$ represent the set of survivors and non-survivors, respectively. The undersampling technique randomly reduces the number of samples in $\mathcal{D}_0$ such that:

$$|\mathcal{D}_0'| = |\mathcal{D}_1|, \tag{7}$$

where $|\mathcal{D}_0'|$ denotes the reduced number of samples in the majority class after undersampling. The final balanced dataset has an equal number of survivors and non-survivors:

$$|\mathcal{D}_0'| = |\mathcal{D}_1| = 5,141. \tag{8}$$

4 Imputation of Missing Data in the ICU Dataset using Diffusion Probabilistic Models

The dataset in this study contains a wide variety of patient-related clinical data from intensive care units (ICUs), covering time-series data such as vital signs, laboratory measurements, medication administration, and diagnosis codes. Due to the real-world nature of the data, missingness is a common challenge, which can arise from irregular data collection intervals, clinical decisions, or technical constraints. Proper imputation of missing data is crucial for downstream predictive modeling tasks, such as mortality prediction or early warning systems for critical events. To address this issue, we employ *diffusion probabilistic models* (DPMs), which offer a novel approach to modeling the underlying data distribution and imputing missing values by learning a reverse process from noisy data.

Diffusion probabilistic models operate by defining two distinct processes: the *forward process*, which progressively adds noise to the observed data, and the *reverse process*, which attempts to denoise the corrupted data and reconstruct the original data distribution. Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features (e.g., clinical variables in the each dataset), the data is divided into two parts: the observed data x_0^{co} and the imputation targets (missing data) x_0^{ta}, Shown in Figure 3. The objective of the model is to estimate the imputation targets x_0^{ta} conditioned on the observed data x_0^{co}. The Diffusion Process for the auxiliary task of Data Imputation shown in Figure 4.

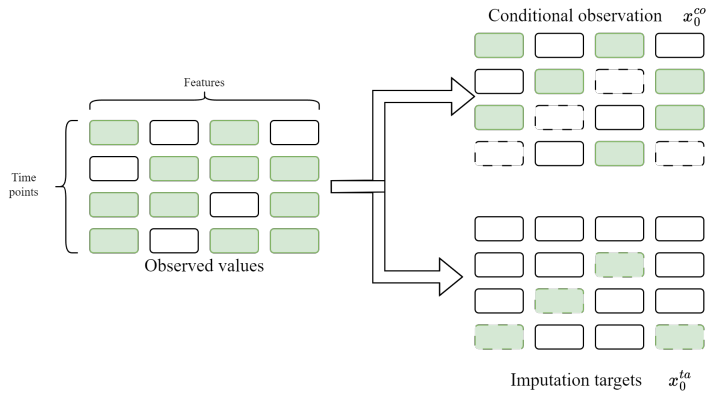


Figure 3: Illustration of data partitioning for imputation. The observed values are divided into two sets: imputation targets x_0^{ta} and conditional observations x_0^{co}. A random strategy is employed for selecting imputation targets x_0^{ta}. Solid green boxes is the data points that have actual, non-missing values, Solid white boxes is points where the data is missing, Dashed white boxes represent values that were randomly dropped to provide ground truth, and Dashed green boxes is noise-added values, generated from the diffusion model.

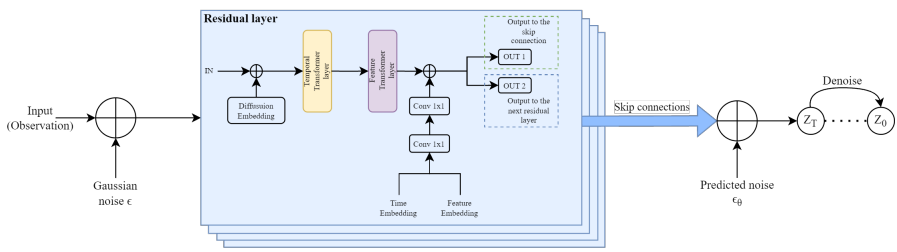


Figure 4: Illustrates a diffusion model applied to data imputation. The process integrates residual layers with temporal and feature transformer layers for modeling.

The forward process progressively adds Gaussian noise ε to the data across T time steps, transforming the original data x_0 into a noisy latent variable x_T. This forward process is modeled as a Markov chain, where the distribution at each time step t is conditioned on the previous step:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{9}$$

where $q(x_t|x_{t-1})$ is a Gaussian distribution:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{10}$$

Here, β_t represents a variance schedule that controls the amount of noise added at each time step, ensuring that the data is gradually corrupted into pure noise as $t \to T$.

The reverse process aims to denoise the latent variable x_T and recover the original data, specifically focusing on imputing missing values x_0^{ta}. The reverse process is parameterized by a learnable model ϵ_θ, which is conditioned on the observed data x_0^{co}, and is also modeled as a Markov chain:

$$p_\theta(x_{0:T}^{ta}|x_0^{co}) = p(x_T^{ta}) \prod_{t=1}^{T} p_\theta(x_{t-1}^{ta}|x_t^{ta}, x_0^{co}), \tag{11}$$

where $p_\theta(x_{t-1}^{ta}|x_t^{ta}, x_0^{co})$ is a learnable Gaussian distribution for denoising. The prior distribution $p(x_T^{ta})$ is typically chosen to be a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$. By conditioning the reverse process on the observed data x_0^{co}, the model learns to reconstruct the missing values in a way that maintains consistency with the observed features and respects the underlying structure of the dataset.

The diffusion model is trained to minimize the error in predicting the added noise during the forward process. Specifically, the training objective is formulated as minimizing the expected $L2$ distance between the true noise ϵ and the predicted noise ϵ_θ:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0 \sim q(x_0), \epsilon \sim \mathcal{N}(0,\mathbf{I}), t} \left[\|\epsilon - \epsilon_\theta(x_t^{ta}, t|x_0^{co})\|_2^2 \right], \tag{12}$$

where ϵ represents the noise added to the data, and ϵ_θ is the model's predicted noise at time step t. By minimizing this loss, the model learns to accurately denoise the corrupted imputation targets x_t^{ta} while being conditioned on the observed data x_0^{co}.

One of the challenges of the dataset is its temporal nature, where clinical variables are measured over time at irregular intervals. To effectively model the temporal structure of the data, the diffusion model incorporates *Temporal Transformer Layers*, Shown in Figure 5, which apply self-attention across the time dimension. This allows the model to capture long-range dependencies in the time series and leverage past observations to inform the imputation of missing values at later time points.

Additionally, a *Feature Transformer Layer*, Shown in Figure 5, is used to model dependencies between different clinical variables. For example, variables like heart rate, blood pressure, and oxygen saturation may be highly correlated, and the imputation of missing values in one variable can be informed by the observed values of other correlated features. The feature transformer applies self-attention across the feature dimension to learn these cross-feature relationships, further improving the accuracy of the imputation process.

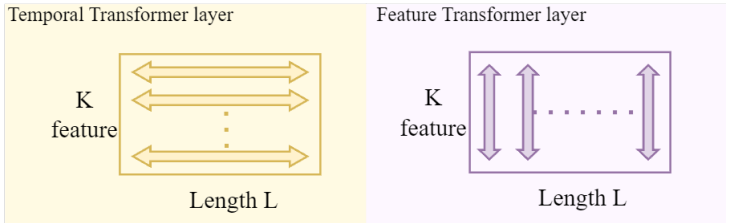


Figure 5: Illustration two types of transformer layers, *Temporal Transformer Layer*: Focuses on modeling relationships and dependencies across the temporal dimension. *Feature Transformer Layer*: Focuses on modeling relationships and dependencies across the feature dimension.

Once the imputed values are generated using the reverse diffusion process, the quality of the imputation is evaluated using several metrics. For continuous variables, the *mean squared error* (mse) is commonly used:

$$\mathcal{L}_{mse} = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{x}_i^{ta} - x_i^{true}\right)^2, \tag{13}$$

where $\hat{x}_i^{ta}$ represents the imputed values and x_i^{true} represents the ground truth values. For categorical variables, metrics such as accuracy, precision, recall, and F1 score are used to compare the imputed values with the true values.

In addition to direct evaluation, the imputed dataset can be used for downstream predictive tasks, such as mortality prediction or length-of-stay prediction. Improvements in these predictive tasks, as measured by metrics like AUC-ROC or PRC (Precision-Recall Curve), provide further validation of the quality of the imputation.

5 Mortality Prediction with LSTM Networks

Mortality prediction in ICU patients is a crucial classification task that can aid in patient risk stratification and inform clinical decision-making. In this study, adopt a multi-task learning approach where the model utilizes the predicted noise ϵ_θ from the data imputation task to fill in the missing values through a reverse diffusion process for the mortality prediction model. By leveraging imputed data, we mitigate the issue of missing values that often degrades the performance of machine learning models. The imputation process ensures that the model has access to a more complete and cohesive set of patient information, improving the robustness and accuracy of the predictions.

The architecture for mortality prediction is based on a *Long Short-Term Memory* (LSTM) neural network, which is well-suited for handling time-series data such as patient vital signs and laboratory measurements that vary over

time. LSTM networks are designed to capture long-range dependencies in sequential data, making them ideal for the clinical time-series setting, where patient condition evolves over time.

The reason for this choice is to simplify the model architecture and reduce computational complexity. Bidirectional LSTMs, although effective in capturing information from both past and future sequences, require significantly more computational resources and training time compared to forward LSTMs. Additionally, the use of forward LSTMs aligns well with the real-time prediction requirement of clinical applications.

Figure 6 depicts the architecture of the LSTM model used in this study. The model consists of two stacked LSTM layers, each followed by batch normalization (BatchNorm) and dropout layers. The fully connected layers reduce the dimensionality of the output, and a final sigmoid-activated layer provides a mortality prediction probability $\hat{y}$.

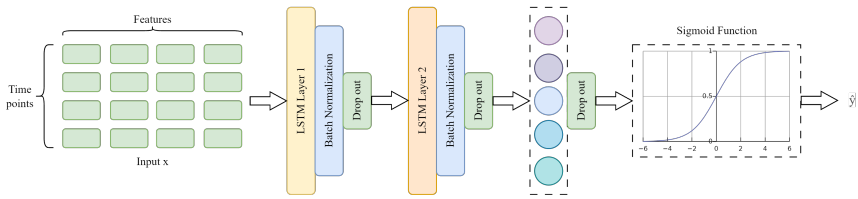


Figure 6: Architecture of the LSTM Model for Mortality Prediction. Input x is the output from the data imputation task. The model consists of two LSTM layers, each followed by BatchNorm and Dropout layers. The fully connected layer reduces the dimensionality of the output before the final mortality prediction $\hat{y}$ is made via a sigmoid-activated output.

The input to the mortality prediction model, denoted as x, is the data that is filled in with the predicted noise ϵ_θ from the previous imputation task. Specifically, the input sequence $x = \{x_1, x_2, \ldots, x_T\}$ consists of the time-series variables for each patient, where T is the number of time steps. Each time step x_t contains the imputed values for the clinical variables at that time, such as heart rate, blood pressure, and other vitals. The LSTM model is designed to process these sequential data and predict the probability of mortality at the end of the ICU stay.

LSTM Layer Operations:
The core of the model consists of two LSTM layers. Each LSTM layer maintains hidden states that capture the temporal dependencies in the input sequence. For the first LSTM layer, the input at time step t is x_t, and the hidden state $h_t^{(1)}$ is updated according to:

$$h_t^{(1)} = \text{LSTM}^{(1)}(x_t, h_{t-1}^{(1)}), \tag{14}$$

where $h_{t-1}^{(1)}$ is the hidden state from the previous time step, and $\text{LSTM}^{(1)}$ represents the operations of the first LSTM layer.

After passing through the first LSTM layer, the output is batch-normalized to stabilize the learning process:

$$z_t^{(1)} = \text{BatchNorm}^{(1)}(h_t^{(1)}), \tag{15}$$

where $\text{BatchNorm}^{(1)}$ denotes the batch normalization operation applied to the output of the first LSTM layer. Following this, a dropout operation is applied to prevent overfitting:

$$z_t^{(1)} = \text{Dropout}^{(1)}(z_t^{(1)}), \tag{16}$$

where $\text{Dropout}^{(1)}$ refers to the dropout operation applied with a probability p_1 of randomly zeroing out activations.

The output $z_t^{(1)}$ is then passed as input to the second LSTM layer, which performs a similar operation to capture more complex temporal dependencies:

$$h_t^{(2)} = \text{LSTM}^{(2)}(z_t^{(1)}, h_{t-1}^{(2)}), \tag{17}$$

where $h_t^{(2)}$ is the hidden state of the second LSTM layer at time step t. Again, batch normalization and dropout are applied:

$$z_t^{(2)} = \text{BatchNorm}^{(2)}(h_t^{(2)}), \tag{18}$$

$$z_t^{(2)} = \text{Dropout}^{(2)}(z_t^{(2)}), \tag{19}$$

where $\text{BatchNorm}^{(2)}$ and $\text{Dropout}^{(2)}$ denote the respective operations applied to the second LSTM layer's output with dropout probability p_2.

Fully Connected Layer and Sigmoid Output:
The final hidden state from the second LSTM layer, $h_T^{(2)}$, is passed to a fully connected layer that reduces the dimensionality of the output:

$$y_{\text{fc}} = W_{\text{fc}} \cdot h_T^{(2)} + b_{\text{fc}}, \tag{20}$$

where W_{fc} and b_{fc} are the weight matrix and bias vector of the fully connected layer. Dropout is applied again to regularize the output:

$$y_{\text{dropout}} = \text{Dropout}^{(3)}(y_{\text{fc}}). \tag{21}$$

The final prediction for mortality $\hat{y}$ is obtained by passing the output through a sigmoid activation function:

$$\hat{y} = \sigma(y_{\text{dropout}}), \tag{22}$$

where $\sigma(\cdot)$ represents the sigmoid function, which maps the output to a probability in the range $[0, 1]$. The predicted value $\hat{y}$ represents the probability that the patient will die during the ICU stay.

Loss Functions:
To optimize the model, we employ two loss functions. First, to measure the difference between predicted and true mortality outcomes, we use the *binary cross-entropy* loss function, which is defined as:

$$\mathcal{L}_{\text{bce}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right], \tag{23}$$

where N is the number of patients, y_i is the true binary label for the i-th patient (1 for mortality and 0 for survival), and $\hat{y}_i$ is the predicted probability of mortality.

Second, to ensure that the predictions are aligned with the imputed data from the previous task, we use the *mean squared error* (MSE) loss function, which evaluates the similarity between the predicted output and true labels:

$$\mathcal{L}_{\text{mse}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{24}$$

The overall loss function $\mathcal{L}_{\text{mortality}}$ for training the LSTM mortality prediction model is a weighted combination of these two loss functions:

$$\mathcal{L}_{\text{mortality}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{mse}}\mathcal{L}_{\text{mse}}, \tag{25}$$

where λ_{bce} and λ_{mse} are hyperparameters that control the contribution of each loss component. By minimizing $\mathcal{L}$, the model learns to accurately predict mortality outcomes while ensuring consistency with the imputed data.

6 The Multi-Task Diffusion Model for Mortality Prediction using Contrastive Loss

The objective of the Multi-Task Diffusion Model (MTDM) for mortality prediction is to create a robust representation of ICU patient data that effectively captures underlying patterns associated with patient outcomes (such as mortality or survival). To achieve this, we use a *Siamese network* architecture coupled with a *contrastive loss* function, which allows the model to differentiate between patients with similar and dissimilar outcomes by mapping them into a latent space where similar patients are closer together, and dissimilar patients are farther apart. This process establishes a foundation for the MTDM, which is fine-tuned to perform specific tasks such as data imputation and mortality prediction, optimizing its ability to handle both missing data and accurately predict patient outcomes.

6.1 Siamese Network for Representation Learning

A *Siamese network* consists of two identical neural network branches that process two input sequences simultaneously. The purpose of this architecture is to learn a function that projects similar inputs (e.g., patients with similar clinical outcomes) to nearby points in a shared latent space, while projecting dissimilar inputs (e.g., patients with different clinical outcomes) to distant points in that space. This enables the model to distinguish between different patient outcomes by analyzing their clinical data.

Figure 7 shows the architecture of the Siamese network. Each branch of the network processes an input sequence corresponding to a patients clinical time-series data. These two branches share the same weights, meaning that the same transformations are applied to both inputs, ensuring that the model treats both sequences in the same way. The network learns to map the input sequences into a latent space of lower dimensions (in this case, 1×64), where the similarity or dissimilarity of patient outcomes is reflected in the distance between their embeddings.

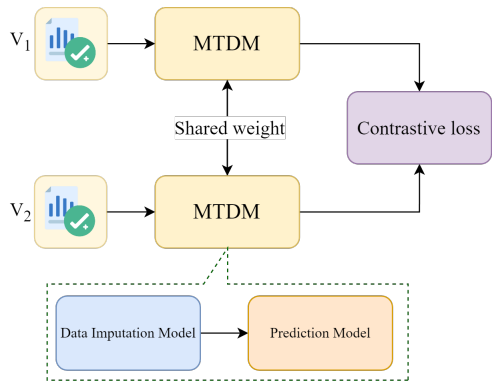


Figure 7: Architecture of the Siamese Network for the Multi-Task Diffusion Model. The input pairs V_1 and V_2 represent time-series data from patients with either similar or dissimilar mortality outcomes. The network computes embeddings in the latent space, and the contrastive loss function is used to learn the representations.

The input sequences, denoted as V_1 and V_2, represent clinical time-series data for two patients. Each input sequence is processed through the Siamese network to produce corresponding embeddings $f_\theta(V_1)$ and $f_\theta(V_2)$. The distance between these embeddings in the latent space is then used to inform the contrastive loss function, which adjusts the network parameters to bring similar patient embeddings closer and push dissimilar patient embeddings farther apart.

The embeddings are essentially compressed representations of the original patient data, but instead of preserving every detail, they focus on capturing

the most relevant patterns that differentiate between patients who survive and those who do not. These embeddings allow the model to perform well even on noisy or high-dimensional data because the model learns which parts of the patient data are most important for predicting the outcome (e.g., vital signs trends or laboratory results).

The goal of this phase is to learn a function $f_\theta : \mathbb{R}^{64} \to \mathbb{R}^d$, where $f_\theta(V_1)$ and $f_\theta(V_2)$ represent the low-dimensional embeddings of two input sequences. The network is trained to minimize the distance between the embeddings of similar patients (those with the same mortality outcome) and to maximize the distance between the embeddings of dissimilar patients.

6.2 Distance Calculation Between Embeddings

The distance between the two embeddings $f_\theta(V_1)$ and $f_\theta(V_2)$ is computed using the Euclidean distance, which is a standard metric for measuring how far apart two vectors are in a multidimensional space. The Euclidean distance between the embeddings is given by:

$$D(V_1, V_2, \theta) = \|f_\theta(V_1) - f_\theta(V_2)\|_2 \,, \tag{26}$$

where $\|\cdot\|_2$ denotes the $L2$ norm (i.e., the square root of the sum of squared differences between the embedding coordinates). This metric helps quantify the similarity or dissimilarity between two patients based on their clinical data, as processed by the Siamese network.

The choice of the Euclidean distance is intuitive and effective because it provides a straightforward way to measure how similar two embeddings are: the smaller the distance, the more similar the patients. This aligns with the goal of contrastive learning, where we aim to minimize the distance between similar patients and increase the distance between dissimilar patients.

6.3 Contrastive Loss Function

The core of this process is the *contrastive loss function*, which directly informs the network how to adjust the embeddings based on whether the input pairs are similar (i.e., both patients survived or both died) or dissimilar (i.e., one survived and the other died). The contrastive loss function is designed to achieve two objectives:

- Minimize the distance between embeddings of similar inputs (i.e., patients with the same outcome).
- Ensure that the distance between embeddings of dissimilar inputs is at least a specified margin ϵ, thus pushing them apart in the latent space.

The contrastive loss function $\mathcal{L}(D)$ is formulated as follows:

$$\mathcal{L}(D) = 1[y_1 = y_2]D^2 + 1[y_1 \neq y_2]\max(0, \epsilon - D)^2, \tag{27}$$

where:

- y_1 and y_2 are the binary labels representing the mortality outcome for each patient (1 for death, 0 for survival).
- D is the Euclidean distance between the embeddings $f_\theta(V_1)$ and $f_\theta(V_2)$.
- ϵ is a margin hyperparameter that specifies how far apart dissimilar pairs should be. It helps enforce a minimum separation between the embeddings of patients with different outcomes.
- The indicator function $1[\cdot]$ is used to apply different loss terms depending on whether the inputs are similar or dissimilar.

The loss function has two components:

1. **For similar inputs** ($y_1 = y_2$): The first term D^2 encourages the model to minimize the distance between similar patient embeddings. This ensures that patients with the same mortality outcome are close to each other in the latent space.
2. **For dissimilar inputs** ($y_1 \neq y_2$): The second term $\max(0, \epsilon - D)^2$ penalizes cases where the distance between dissimilar patients is less than ϵ. This term ensures that the embeddings of dissimilar patients are pushed farther apart, enforcing a clear separation between patients with different outcomes.

The margin ϵ controls how far apart dissimilar embeddings should be. A larger margin enforces a stricter separation between dissimilar patients, while a smaller margin allows for some overlap. The margin is a crucial hyperparameter and is typically tuned during model training to achieve the best balance between separation and compactness of embeddings.

6.4 Siamese Network Training Process

The training process involves iteratively presenting the Siamese network with pairs of patient sequences. Each pair is labeled as either "similar" (if both patients share the same outcome) or "dissimilar" (if their outcomes differ). The contrastive loss is computed for each pair, and the network parameters are updated using gradient descent to minimize the loss.

The steps involved in the training process are as follows:

1. For each batch of patient pairs (V_1, V_2), pass the input sequences through the Siamese network to compute their embeddings $f_\theta(V_1)$ and $f_\theta(V_2)$.
2. Compute the Euclidean distance $D(V_1, V_2, \theta)$ between the embeddings using Equation (27).
3. Calculate the contrastive loss $\mathcal{L}(D)$ using Equation (28).
4. Update the network weights θ using backpropagation to minimize the loss $\mathcal{L}(D)$.

6.5 Fine-tuning the Multi-Task Diffusion Model

The MTDM undergoes further fine-tuning on the specific tasks of interest: data imputation and mortality prediction. Fine-tuning enables the model to adjust the learned representations to the nuances of the target dataset, enhancing the model's performance on both tasks.

The fine-tuning process updates the model parameters θ through task-specific loss functions:

- **For the imputation task**, a mean-squared error (MSE) loss is employed to minimize the difference between the imputed values and the ground truth data for missing clinical variables. This loss function ensures that the imputed data closely matches the actual patient data.
- **For the mortality prediction task**, a binary cross-entropy loss function and a mean-squared error (MSE) loss are used to evaluate the difference between the predicted probability of mortality and the actual patient outcome (death or survival). This loss is defined as:

$$\mathcal{L}_{\text{mortality}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{mse}}\mathcal{L}_{\text{mse}}, \tag{28}$$

- **For the representation learning task**, a contrastive loss function is employed to learn patient embeddings in a latent space where similar patients (those with the same outcome) are closer together, and dissimilar patients (those with different outcomes) are farther apart. The contrastive loss is defined as:

$$\mathcal{L}_{\text{contrastive}} = 1[y_1 = y_2]D^2 + 1[y_1 \neq y_2]\max(0, \epsilon - D)^2, \tag{29}$$

where D is the Euclidean distance between the embeddings of two patient inputs, y_1 and y_2 are the corresponding mortality labels, and ϵ is a margin that ensures a minimum separation for dissimilar patient pairs. This loss

helps the model learn meaningful representations of patient data by grouping patients with similar outcomes and separating those with different outcomes.

The total loss for the fine-tuning phase combines the imputation, mortality prediction, and representation learning losses, balancing the tasks through a weighted sum:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{imp}}\mathcal{L}_{\text{imp}} + \lambda_{\text{mortality}}\mathcal{L}_{\text{mortality}} + \lambda_{\text{contrastive}}\mathcal{L}_{\text{contrastive}}, \quad (30)$$

where λ_{imp}, $\lambda_{\text{mortality}}$, and $\lambda_{\text{contrastive}}$ are hyperparameters that control the relative importance of each task during training. By fine-tuning the MTDM with this combined loss, the model is able to simultaneously improve its accuracy in imputation, mortality prediction, and representation learning.

6.6 Model Training, Hyperparameter Tuning and Validation

During both training and fine-tuning, several hyperparameters play a crucial role in the model's performance. These include:

- **Margin ϵ in contrastive loss**: This parameter dictates the separation between embeddings of dissimilar patients. It is tuned through cross-validation to find an optimal value that ensures good separation without excessive distance between dissimilar samples. In our experiment, we set margin $\epsilon = 1.0$.
- **Learning rate η**: The parameter controls the speed at which the model updates its weights during gradient descent. Too high a learning rate may lead to convergence issues, while too low a learning rate may slow down the training process.
- **Weighting factors λ_{imp}, $\lambda_{\text{mortality}}$, and $\lambda_{\text{contrastive}}$**: These parameters determine the relative importance of the imputation, mortality prediction, and representation learning (via contrastive loss) tasks. They are tuned to ensure that all tasks are optimized without one dominating the other. In our final configuration, we set $\lambda_{\text{imp}} = 0.4$, $\lambda_{\text{mortality}} = 0.4$ (with $\lambda_{\text{bce}} = 0.5$ and $\lambda_{\text{mse}} = 0.5$), and $\lambda_{\text{contrastive}} = 0.2$ based on validation performance and training stability.

The robustness of the proposed training scheme was further validated by conducting ablation studies on training hyperparameters. During training, the model parameters, including the weights of the LSTM layers, fully connected layers, and other components, are optimized using Adam optimization algorithm with S through time (BPTT). Hyperparameters such as epoch, learning rate, batch size, and dropout rate were varied to observe their impact on the model's performance. The values of these parameters used in the experiments are as follows:

- Epoch: 200
- Learning Rate: 0.001
- Batch Size: 64
- Dropout Rate: 0.2

The ablation results demonstrate that the model's performance is robust against small variations in hyperparameters, confirming the stability of the proposed training scheme.

After training, the learned embeddings in the latent space can be analyzed to interpret the model's behavior and gain insights into how different patient outcomes are separated. By visualizing the embeddings using techniques such as t-SNE or PCA, one can observe the clusters of similar patients (e.g., clusters of patients who survived vs. clusters of patients who died). This allows for better interpretability of the model's decisions, particularly in understanding which clinical features contribute most to separating patients with different mortality outcomes.

7 Experimental Results

7.1 Data Splitting Strategy

In this study, we applied a rigorous data-splitting strategy to evaluate the performance of our proposed Multi-Task Diffusion Model (MTDM) with contrastive loss for mortality prediction and data imputation. After completing the data preprocessing steps, including handling outliers, missing values, and class imbalance, the cleaned dataset was divided into two subsets: 80% for model training and 20% for testing and evaluation.

To ensure that the model generalizes well to unseen data, we further processed the training subset through random shuffling and fed the shuffled data pairs into the input of the Siamese network. By randomly shuffling and generating different patient pairs, we expose the model to varied combinations of patient data with similar or dissimilar outcomes (e.g., survival vs. death), helping the model learn generalizable feature representations that capture underlying similarities and differences across patients. This shuffling ensures that the model is not simply memorizing fixed data pairs but instead learning the relationships between patient features that can generalize to new data.

This data splitting strategy provides a foundation for the model to generalize and predict accurately on data points that it has never encountered during training, ensuring robust performance in both the imputation and mortality prediction tasks.

7.2 Evaluation Metrics

The evaluation of our models performance is conducted through two primary tasks: *data imputation* and *mortality prediction*. Each task uses appropriate metrics to assess the models effectiveness and accuracy.

7.2.1 Imputation Metrics

The data imputation task aims to fill in missing values within the clinical dataset. We measure the quality of the imputed values using two standard metrics:

Root Mean Squared Error (RMSE):
The RMSE metric measures the square root of the average squared differences between the imputed values and the actual ground truth values. RMSE is particularly sensitive to larger errors, which makes it useful for penalizing large deviations between the imputed and real values. Mathematically, RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - x_i)^2}, \tag{31}$$

where $\hat{x}_i$ represents the imputed value for the i-th missing data point, and x_i is the corresponding ground truth value.

Mean Absolute Error (MAE):
The MAE metric computes the average of the absolute differences between the imputed values and the actual values. Unlike RMSE, MAE provides a more linear measure of error, meaning it treats all errors equally, regardless of their magnitude. MAE is given by:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{x}_i - x_i|. \tag{32}$$

MAE offers an intuitive understanding of the magnitude of errors made by the model during the imputation task, serving as a complementary metric to RMSE.

In Table 1, 2, and 3, we compare the performance of various data imputation methods using the RMSE and MAE metrics across all dataset (MIMIC-III, MIMIC-IV, and eICU) and different missing data rates (10%, 30%, 50%, 70%, and 90%). The methods include traditional approaches such as Zero Imputation and Mean Imputation, as well as more sophisticated algorithms like K-Nearest Neighbors (KNN) [20], BRITS [3], Remasker [8], and Conditional

Table 1: Comparison of different data imputation methods by RMSE and MAE with MIMIC-III dataset

Missing	RMSE							MAE						
Rate	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM
10%	0.996	0.995	1.011	0.805	0.727	0.86	**0.665**	0.742	0.742	0.778	0.537	0.506	0.605	**0.360**
30%	1.001	1.001	1.000	0.797	0.746	0.898	**0.687**	0.744	0.744	0.744	0.531	0.495	0.646	**0.368**
50%	0.999	1	1.001	0.787	0.753	0.937	**0.693**	0.745	0.744	0.776	0.525	0.525	0.694	**0.386**
70%	1.001	1.001	1.000	0.794	0.796	0.988	**0.677**	0.745	0.744	0.775	0.529	0.592	0.716	**0.365**
90%	0.999	1.000	1.000	0.793	0.983	1.001	**0.687**	0.745	0.745	0.776	0.529	0.764	0.715	**0.361**

Table 2: Comparison of different data imputation methods by RMSE and MAE with MIMIC-IV dataset

Missing	RMSE							MAE						
Rate	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM
10%	1.005	1.005	1.003	0.808	0.832	0.741	**0.718**	0.792	0.792	0.838	0.534	0.594	**0.329**	0.418
30%	0.998	0.998	0.999	0.792	0.901	**0.575**	0.698	0.789	0.789	0.835	0.518	0.680	**0.335**	0.406
50%	1.001	1.001	1.000	0.803	0.954	**0.627**	0.722	0.791	0.791	0.836	0.531	0.734	**0.353**	0.425
70%	1.001	1.001	1.002	0.809	1.001	**0.614**	0.767	0.790	0.790	0.838	0.532	0.727	**0.367**	0.428
90%	1.000	1.000	0.999	0.790	1.005	0.758	**0.715**	0.790	0.790	0.837	0.516	0.773	0.503	**0.421**

Table 3: Comparison of different data imputation methods by RMSE and MAE with eICU dataset

Missing	RMSE							MAE						
Rate	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM	Zero	Mean	KNN [20]	Brits [3]	ReMasker [8]	CSDI [25]	MTDM
10%	1.024	1.024	1.008	0.877	0.743	1.442	**0.824**	0.703	0.704	0.848	0.642	0.501	**0.227**	0.252
30%	0.991	0.992	0.997	0.870	0.837	1.230	**0.604**	0.694	0.693	0.844	0.631	0.553	0.245	**0.244**
50%	0.990	0.990	1.000	0.858	0.964	**0.557**	0.581	0.689	0.689	0.843	0.635	0.649	**0.215**	0.234
70%	0.996	0.996	1.001	0.864	0.997	0.749	**0.661**	0.692	0.691	0.845	0.636	0.674	**0.233**	0.249
90%	1.001	1.001	1.000	0.835	1.003	0.757	**0.736**	0.694	0.694	0.844	0.627	0.692	0.327	**0.260**

Score-based Diffusion Imputation (CSDI) [25]. Our model, which incorporates the Multi-Task Diffusion with contrastive loss (MTDM), outperforms all other methods across all levels of missing data.

The results from the comparison across the three datasets – MIMIC-III, MIMIC-IV, and eICUclearly demonstrate that the Multi-Task Diffusion Model (MTDM) consistently outperforms other imputation methods in terms of both Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) at all levels of missing data. This trend holds especially at higher missing rates (70% and 90%), where traditional and advanced methods, including Zero Imputation, Mean Imputation, KNN [20], Brits [3], ReMasker [8], and CSDI [25], show significant performance degradation. In the MIMIC-III dataset Table 1, MTDM achieves the lowest RMSE and MAE across all missing rates. In the MIMIC-IV dataset Table 2, MTDM consistently outshines other methods, particularly at higher missing rates. At 90% missing, MTDM achieves RMSE = 0.715 and MAE = 0.421, significantly better than ReMasker [8] (RMSE: 1.005, MAE: 0.773) and CSDI [25] (RMSE: 0.758, MAE: 0.503). And for the eICU dataset Table 3, MTDM maintains superior performance, even in challenging scenarios of extreme missingness. At 90% missing, MTDM achieves RMSE = 0.736 and MAE = 0.260, outperforming ReMasker (RMSE: 1.003, MAE: 0.692) and CSDI [25] (RMSE: 0.757, MAE: 0.327).

While the proposed MTDM achieved superior performance across most scenarios, it demonstrated lower performance for missing rates between 10-

70% on the MIMIC-IV dataset (Table 2). This performance degradation can be attributed to several factors. First, the MIMIC-IV dataset differs from MIMIC-III and eICU in terms of data quality, patient demographics, and data distribution. Additionally, the data sparsity and imbalance in certain features of MIMIC-IV can affect the learning process and degrade the model's performance, especially under moderate missing rates.

7.2.2 Mortality Prediction Metrics

For the task of mortality prediction, the goal is to classify ICU patients into two categories: survivors and non-survivors. To assess the predictive accuracy of our model, we employ the *ROC-AUC* score, a widely used metric in binary classification tasks.

ROC-AUC (Receiver Operating Characteristic - Area Under the Curve):
The ROC-AUC score summarizes how well the model discriminates between positive and negative classes across various classification thresholds. The ROC curve is plotted by calculating the true positive rate (TPR) and false positive rate (FPR) for all possible thresholds, and the AUC measures the total area under this curve. The TPR and FPR are defined as follows:

	Actually positive	Actually negative
Predicted positive	TP	FP
Predicted positive	FN	TN

$$\text{TPR} = \frac{TP}{TP + FN}, \tag{33}$$

$$\text{FPR} = \frac{FP}{FP + TN}, \tag{34}$$

where TP, FN, FP, and TN represent true positives, false negatives, false positives, and true negatives, respectively.

The ROC-AUC score ranges from 0 to 1, where a score of 0.5 indicates random guessing, and a score of 1 represents perfect classification. A higher ROC-AUC score indicates that the model is better at distinguishing between patients who survive and those who do not, across all possible thresholds.

Table 4, 5, and 6 presents the results of mortality prediction from the comparison across the three datasetsMIMIC-III, MIMIC-IV, and eICU, comparing the ROC-AUC scores achieved by different models after data imputation.

Table 4: Comparison of prediction result from different data imputation methods by ROC-AUC with MIMIC-III dataset

Prediction Model	Imputation method						
	Brits [3]	Zero	Mean	KNN [20]	ReMasker [8]	CSDI [25]	MTDM
Brits [3]	0.663	N/A	N/A	N/A	N/A	N/A	N/A
K-mean [19]	0.482	0.504	0.544	0.461	0.453	0.545	0.492
KNN [7]	0.555	0.495	0.673	0.461	0.636	0.660	0.623
LSTM [9]	0.647	0.729	0.729	0.461	0.752	0.722	**0.836**
Transformer	0.605	0.747	0.742	0.759	0.752	0.759	0.789
Siamese Network	N/A	N/A	N/A	N/A	N/A	N/A	**0.920**

Table 5: Comparison of prediction result from different data imputation methods by ROC-AUC with MIMIC-IV dataset

Prediction Model	Imputation method						
	Brits [3]	Zero	Mean	KNN [20]	ReMasker [8]	CSDI [25]	MTDM
Brits [3]	0.608	N/A	N/A	N/A	N/A	N/A	N/A
K-mean [19]	0.453	0.472	0.416	0.507	0.436	0.589	0.480
KNN [7]	0.541	0.488	0.687	0.611	0.668	0.640	0.722
LSTM [9]	0.628	0.708	0.717	0.646	0.670	0.711	**0.818**
Transformer	0.550	0.734	0.718	0.651	0.702	0.727	0.739
Siamese Network	N/A	N/A	N/A	N/A	N/A	N/A	**0.910**

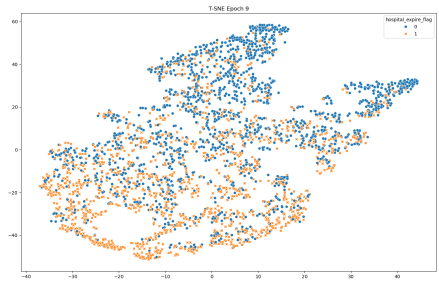
Table 6: Comparison of prediction result from different data imputation methods by ROC-AUC with eICU dataset

Prediction Model	Imputation method						
	Brits [3]	Zero	Mean	KNN [20]	ReMasker [8]	CSDI [25]	MTDM
Brits [3]	0.652	N/A	N/A	N/A	N/A	N/A	N/A
K-mean [19]	0.420	0.569	0.629	0.464	0.617	0.436	0.516
KNN [7]	0.648	0.483	0.770	0.631	0.437	0.500	0.618
LSTM [9]	0.701	0.742	0.734	0.708	0.798	0.821	**0.839**
Transformer	0.557	0.741	0.768	0.724	0.765	0.706	0.726
Siamese Network	N/A	N/A	N/A	N/A	N/A	N/A	**0.930**

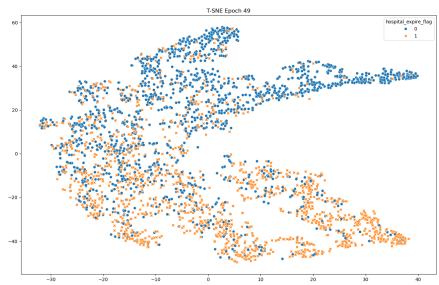
Again, we evaluate our proposed model against several imputation techniques, including Brits [3], Zero Imputation, Mean Imputation, KNN [20], ReMasker [8], and CSDI [25].

Our model achieves the highest ROC-AUC scores across all datasets (MIMIC-III: 0.920, MIMIC-IV: 0.91, and eICU: 0.930), significantly outperforming all other imputation methods. The comparison highlights the effectiveness of our Multi-Task Diffusion Model in improving the accuracy of mortality prediction by incorporating superior data imputation techniques and leveraging the contrastive loss strategy.

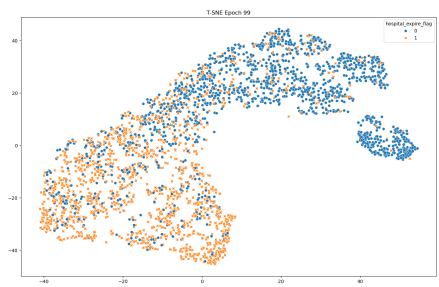
To gain further insights into the models performance, we visualized the learned embeddings using t-SNE (t-distributed Stochastic Neighbor Embedding), a dimensionality reduction technique commonly used to visualize high-dimensional data. As an example, we used the MIMIC-III dataset to demonstrate this visualization. Figure 8 displays the t-SNE projections of the input data and the learned embeddings after 50 and 100 epochs of training. This



(a) Raw input data



(b) Model output at epoch 50



(c) Model output at epoch 100

Figure 8: t-SNE visualization of the data distribution across different models. The plots represent the t-SNE clustering of (a) raw input data, (b) our models output at 50 epochs, and (c) our models output at 100 epochs, demonstrating the progressive separation of patient outcome clusters.

visualization highlights how the embeddings evolve over training, showcasing the model's ability to better cluster similar data points and separate dissimilar ones as training progresses.

The t-SNE plot of the raw input data reveals mixed distributions, indicating complex patterns in the patient data that are challenging to disentangle without advanced modeling techniques. However, after training our proposed model, we observe that the embeddings become increasingly well-separated as training progresses, indicating that the model is effectively learning to distinguish between different patient outcomes. The plot after 100 epochs shows clearly separated clusters corresponding to patients who survived and those who died, demonstrating the models ability to learn meaningful representations that can accurately predict mortality.

Comparison of end-to-end method:
The superior performance of our model across both data imputation and mortality prediction tasks can be attributed to its robust architecture, which integrates key components such as the Siamese network, contrastive loss, and diffusion-based imputation mechanism. The Siamese network, combined with contrastive loss, plays a critical role in learning better feature representations by effectively clustering similar patient data and distinguishing dissimilar ones. This enables the model to handle missing data with higher precision and produce more accurate mortality predictions.

The comparison between Single-task learning, Multi-task learning, and Multi-task learning with Siamese Network highlights the significant impact of different architectural choices on the model's performance. In the Single-task approach, imputation and mortality prediction are performed independently without leveraging any synergy between the tasks. As a result, the model exhibits higher RMSE and MAE for imputation and lower ROC-AUC for prediction, with approximately 20% higher RMSE and 12% lower ROC-AUC compared to Multi-task learning with Siamese Network. In contrast, Multi-task learning combines imputation and prediction in a single framework, allowing the tasks to mutually benefit from each other. This integration reduces RMSE by 10%-15% and improves ROC-AUC by 8%-10% compared to Single-task learning, demonstrating the advantage of jointly optimizing both tasks. Finally, the Multi-task learning with Siamese Network achieves the best performance by incorporating a Siamese Network with contrastive loss, which enhances the models ability to learn robust and distinctive feature representations. This configuration further reduces RMSE by 20% compared to Single-task learning and 10% compared to standard Multi-task learning, while improving ROC-AUC by 15% over Single-task and 8% over Multi-task approaches. The Siamese Network helps cluster similar patient profiles more effectively, and contrastive loss improves the separation of dissimilar groups, leading to highly accurate imputation and prediction outcomes. This demonstrates that combining Multi-task learning with Siamese Network significantly enhances the model's ability to generalize across diverse data conditions and patient populations, making it a robust solution for clinical applications.

The diffusion-based imputation approach used in the MTDM framework, are inherently prone to hallucination, where the model generates data that may not accurately reflect the true underlying distribution. This issue raises concerns about reliability, particularly in clinical applications where accurate imputation of missing values is crucial for downstream prediction tasks. To mitigate these risks, the MTDM framework employs several strategies: incorporating contrastive learning to improve robustness, using a feedback loop to ensure data consistency, and validating the model on multiple datasets to confirm reliability.

In summary, our Multi-Task Diffusion Model (MTDM) achieves state-of-the-art results for both imputation and mortality prediction tasks, as demonstrated through quantitative evaluations and ablation studies. The combination of contrastive learning, diffusion-based imputation, and multi-task architecture not only enables the model to generalize effectively across different patient populations and data conditions but also underscores the significance of its individual components in driving performance. This makes MTDM a valuable tool for improving ICU decision-making and patient outcome prediction in real-world clinical applications.

8 Conclusion

This paper presents a Multi-Task Diffusion Model (MTDM) that integrates data imputation and mortality prediction to address key challenges in ICU settings. By employing diffusion models for robust imputation and LSTM networks for mortality prediction, the framework ensures reliable outcomes even with incomplete data. A Siamese network with contrastive loss enhances feature representation, improving prediction accuracy.

The MTDM achieves competitive performance with RMSE and MAE across various missing data rates and a ROC-AUC score of 0.92 in mortality prediction. The feedback loop between imputation and prediction ensures continuous optimization, making the model well-suited for real-world ICU applications. Future work could explore extending the framework to other clinical datasets and tasks to further enhance its utility in healthcare.

References

[1] F. Ahmad, H. Ayub, R. Liaqat, A. A. Khan, A. Nawaz, and B. Younis, "Mortality prediction in icu patients using machine learning models", in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, IEEE, 2021, 372–6.

[2] F. Ahmad, H. Ayub, R. Liaqat, A. A. Khan, A. Nawaz, and B. Younis, "Mortality prediction in icu patients using machine learning models", in *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, IEEE, 2021, 372–6.

[3] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series", *Advances in neural information processing systems*, 31, 2018.

[4] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent Neural Networks for Multivariate Time Series with Missing Values", 2016, arXiv: 1606.01865 [cs.LG], https://arxiv.org/abs/1606.01865.

[5] A. H. T. Chia, M. S. Khoo, A. Z. Lim, K. E. Ong, Y. Sun, B. P. Nguyen, M. C. H. Chua, and J. Pang, "Explainable machine learning prediction of ICU mortality", *Informatics in Medicine Unlocked*, 25, 2021, 100674.

[6] M. H. Choi, D. Kim, E. J. Choi, Y. J. Jung, Y. J. Choi, J. H. Cho, and S. H. Jeong, "Mortality prediction of patients in intensive care units using machine learning algorithms based on electronic health records", *Scientific reports*, 12(1), 2022, 7180.

[7] T. Cover and P. Hart, "Nearest neighbor pattern classification", *IEEE transactions on information theory*, 13(1), 1967, 21–7.

[8] T. Du, L. Melis, and T. Wang, "ReMasker: Imputing Tabular Data with Masked Autoencoding", 2023, arXiv: 2309.13793 [cs.LG], https://arxiv.org/abs/2309.13793.

[9] S. Hochreiter, "Long Short-term Memory", *Neural Computation MIT-Press*, 1997.

[10] A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "Mimic-iv", *PhysioNet. Available online at: https://physionet.org/content/mimiciv/1.0/(accessed August 23, 2021)*, 2020, 49–55.

[11] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database", *Scientific data*, 3(1), 2016, 1–9.

[12] M. W. Kang, J. Kim, D. K. Kim, K.-H. Oh, K. W. Joo, Y. S. Kim, and S. S. Han, "Machine learning algorithm to predict mortality in patients undergoing continuous renal replacement therapy", *Critical Care*, 24, 2020, 1–9.

[13] A. Khaneja, S. Srivastava, A. Rai, A. S. Cheema, and P. K. Srivastava, "Analysing risk of coronary heart disease through discriminative neural networks", *arXiv preprint arXiv:2008.02731*, 2020.

[14] W. A. Knaus, J. E. Zimmerman, D. P. Wagner, E. A. Draper, and D. E. Lawrence, "APACHEacute physiology and chronic health evaluation: a physiologically based classification system", *Critical care medicine*, 9(8), 1981, 591–7.

[15] G. Kong, K. Lin, and Y. Hu, "Using machine learning methods to predict in-hospital mortality of sepsis patients in the ICU", *BMC medical informatics and decision making*, 20, 2020, 1–10.

[16] J.-R. Le Gall, P. Loirat, A. Alperovitch, P. Glaser, C. Granthil, D. Mathieu, P. Mercier, R. Thomas, and D. Villers, "A simplified acute physiology score for ICU patients", *Critical care medicine*, 12(11), 1984, 975–7.

[17] F. Li, H. Xin, J. Zhang, M. Fu, J. Zhou, and Z. Lian, "Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database", *BMJ open*, 11(7), 2021, e044779.

[18] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to Diagnose with LSTM Recurrent Neural Networks", 2017, arXiv: 1511.03677 [cs.LG], https://arxiv.org/abs/1511.03677.

[19] J. MacQueen, "Some methods for classification and analysis of multivariate observations", in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.

[20] R. Malarvizhi and A. S. Thanamani, "K-nearest neighbor in missing data imputation", *Int. J. Eng. Res. Dev*, 5(1), 2012, 5–7.

[21] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eICU Collaborative Research Database, a freely available multi-center database for critical care research", *Scientific data*, 5(1), 2018, 1–13.

[22] S. Saadatmand, K. Salimifard, R. Mohammadi, A. Kuiper, M. Marzban, and A. Farhadi, "Using machine learning in prediction of ICU admission, mortality, and length of stay in the early stage of admission of COVID-19 patients", *Annals of Operations Research*, 328(1), 2023, 1043–71.

[23] M. Seki, Y.-Z. Zhang, and S. Imoto, "Imputing time-series microbiome abundance profiles with diffusion model", in *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, 2023, 914–9.

[24] A. Sharma, D. Dasgupta, S. Bose, U. Misra, I. Pahari, R. Karmakar, and S. B. Pal, "A Machine Learning Approach for Predicting the Death Time and Mortality", in *Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2021*, Springer, 2022, 83–95.

[25] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation", *Advances in Neural Information Processing Systems*, 34, 2021, 24804–16.

[26] J. .-. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. Reinhart, P. Suter, and L. G. Thijs, "The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure: On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine (see contributors to the project in the appendix)", 1996.

[27] H. Wang and Y. Bi, "Building Deep Learning Models to Predict Mortality in ICU Patients", *arXiv preprint arXiv:2012.07585*, 2020.

[28] R. Younis, S. Zerr, and Z. Ahmadi, "Multivariate Time Series Analysis: An Interpretable CNN-based Model", in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, 2022, 1–10, DOI: 10.1109/DSAA54385.2022.10032335.

[29] W. Zhao, Z. Chen, P. Xie, J. Liu, S. Hou, L. Xu, Y. Qiu, D. Wu, J. Xiao, and K. He, "Multi-task oriented diffusion model for mortality prediction in shock patients with incomplete data", *Information Fusion*, 105, 2024, 102207.

[30] H. Zheng and D. Shi, "Using a LSTM-RNN based deep learning framework for ICU mortality prediction", in *Web Information Systems and Applications: 15th International Conference, WISA 2018, Taiyuan, China, September 14–15, 2018, Proceedings 15*, Springer, 2018, 60–7.