A Transformer-based Diffusion Probabilistic Model for Heart Rate and Blood Pressure Forecasting in Intensive Care Unit

Ping Chang[1], Huayu Li[1], Stuart F. Quan[2,3], Shuyang Lu [4,5], Shu-Fen Wung [6,7], Janet Roveda[1,6,8], Ao Li[1,6]

[1]Department of Electrical & Computer Engineering, The University of Arizona, Tucson, AZ, USA
[2]Division of Sleep and Circadian Disorders, Departments of Medicine and Neurology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA
[3]Asthma and Airway Disease Research Center, College of Medicine, The University of Arizona, Tucson, AZ, USA
[4]Department of Cardiovascular Surgery, Zhongshan Hospital, Fudan University, Shanghai, P.R. China
[5]The Shanghai Institute of Cardiovascular Diseases, Shanghai, P.R. China
[6]Bio5 Institute, The University of Arizona, Tucson, AZ, USA
[7]College of Nursing, The University of Arizona, Tucson, AZ, USA
[8]Department of Biomedical Engineering, The University of Arizona, Tucson, AZ, USA

Corresponding author:
Ao Li, Ph.D.
Department of Electrical and Computer Engineering
The University of Arizona
1230 E Speedway Blvd
Tucson, AZ, 85719
Email: aoli1@arizona.edu

Abstract

Background and objective: In the Intensive Care Unit (ICU), vital sign monitoring is critical, and an accurate predictive system is required. This study will create a novel model to forecast Heart Rate (HR), Systolic Blood Pressure (SBP), and Diastolic Blood Pressure (DBP) in ICU. These vital signs are crucial for prompt interventions for patients.
Methods: We extracted $24,886$ ICU stays from the MIMIC-III database which contains data from over 46 thousand patients, to train and test the model. The model proposed in this study, Transformer-based Diffusion Probabilistic Model for Sparse Time Series Forecasting (TDSTF), merges Transformer and diffusion models to forecast vital signs. The TDSTF model showed state-of-the-art performance in predicting vital signs in the ICU, outperforming other models' ability to predict distributions of vital signs and being more computationally efficient. The code is available at https://github.com/PingChang818/TDSTF.
Results: The results of the study showed that TDSTF achieved a Normalized Average Continuous Ranked Probability Score (NACRPS) of 0.4438 and a Mean Squared Error (MSE) of 0.4168, an improvement of 18.9% and 34.3% over the best baseline model, respectively. The inference speed of TDSTF is more than 17 times faster than the best baseline model.
Conclusion: TDSTF is an effective and efficient solution for forecasting vital signs in the ICU, and it shows a significant improvement compared to other models in the field.
Keywords: deep learning, time series forecasting, sparse data, vital signs, ICU

1 Introduction

Vital signs are crucial in monitoring patients' health and body functions in the ICU. Continuous monitoring systems alert caregivers of potential adverse events [1, 2]. A predictive warning system for vital signs can save valuable time by enabling prompt interventions. However, the implementation of such a system faces several challenges. ICUs are complex environments where patients have multiple underlying conditions, treatments, and interventions that can affect their vital signs, making it difficult to develop algorithms that accurately predict them. Vital sign prediction requires large amounts of data to develop accurate algorithms, but in many ICUs, the availability and quality of electronic health record data can be limited. This makes it challenging to use classical statistical methods, which often rely on manual feature engineering and cannot capture complex patterns. As a result, few attempts have been made to predict vital signs in the ICU using these methods.

With the development of deep learning, applying this new approach to predicting ICU vital signs is possible. As is well known, deep learning has revolutionized the field of time series forecasting in recent years, with its high representational power enabling successful predictions of vital signs in various studies. For instance, in Generative Boosting [3], the Long Short-Term Memory (LSTM) network is used to create a generative model, effectively reducing error propagation and improving HR prediction performance. The dataset was made more representative by utilizing mutual information-based clustering to train the model. In a comparison of Recurrent and Convolutional Neural Network (RNN and CNN) [4], using different horizon strategies on the MIMIC-II dataset, the bidirectional LSTM (Bi-LSTM) with the DIRMO strategy delivered the best predictions of blood pressure and HR. The TOP-Net model [5] predicts tachycardia onset using Bi-LSTM, with a prediction horizon of up to 6 hours. It was trained on the data of less than 6 thousand patients from the MIMIC-III database. The Temporal Fusion Transformer [6] predicts vital sign quantiles based on an attention mechanism, capturing anomaly temporal patterns and optimizing input time windows by calculating temporal importance.

Despite the promising results shown by these methods, deploying a vital sign forecasting system in the ICU remains challenging due to several limitations: (1) The models require continuous monitoring of vital signs, which increases the cost of deployment as the complex and unpredictable conditions in the ICU make intermittent monitoring ineffective and unsafe. (2) The models only consider vital signs, ignoring the interrelated events—interventions and conditions surrounding a patient in the ICU—that could improve forecasting accuracy. For example, the existing models should have addressed active interventions such as medications, potential medical procedures, and their impact on patient vital signs. (3) The datasets used to evaluate the models often have a limited number of subjects, leading to a lack of generalizability and potential bias, which is a major concern in critical ICU scenarios.

We aim to develop a vital sign forecasting model that can be easily deployed in the ICU setting without requiring any changes to the current data collection methods. The use of diffusion probabilistic models (diffusion models for short) [7, 8] in time series analysis has been gaining popularity due to their balance between flexibility and tractability [9]. These models have achieved state-of-the-art performance in time series forecasting. Our study aims to investigate the effectiveness of diffusion models in handling sparse time series data and making fast and accurate predictions of vital signs in the ICU setting. The triplet form of the diffusion model enhances its ability to process sparse data, and the use of a Transformer-based backbone leads to improved performance compared to baseline models. The ultimate goal is to provide ICU caregivers with critical information promptly by considering all recorded events, not just vital signs. The novel contributions of this paper include: (1) An Examination of the ability of the diffusion model to extract temporal dependencies from sparse time series data. (2) Use the triplet form to enhance the efficiency of the diffusion model when processing sparse data. (3) Fast and accurate forecasting of vital signs in the ICU setting. (4) Integration of all recorded events in the ICU setting for vital sign forecasting without being limited by the screened data. (5) Comparison of the proposed model with baseline models, showing improved performance using the Transformer as the backbone.

We look forward to this work being the starting point for tackling practical clinical ICU applications. It is essential to notice that subtle variations in vital indicators are warning signals of clinical deterioration that could eventually result in adverse outcomes. To further improve treatment outcomes, we may extend the proposed method to include electrocardiogram (ECG) signals to predict and optimize medication impact on patient conditions. We may evaluate how medical procedures improve a patient's breathing by including respiratory waveform. We may also expand the proposed model to identify false ICU alarms to reduce ICU caregivers' stress.

2 Related works

2.1 Probabilistic time series forecasting

In many real-world scenarios, the future is uncertain, and making a single best estimate of the outcome is impossible. To account for this uncertainty, probabilistic forecasting provides a range of possible outcomes and probabilities of each. This is particularly significant in the ICU setting, where caregivers must understand the risks associated with different decisions. Several probabilistic time series forecasting models have been proposed in recent years and achieved state-of-the-art performance. MQ-RNN [10] uses an LSTM network to generate hidden states of the input time series, which are then transformed into contextual information by a global Multilayer Perceptron Network (MLP). The local MLP uses contextual information and covariates to generate quantile predictions. In DeepAR [11], hidden states are obtained through LSTM, which are then input into linear layers with activation to generate the mean and variance of a Gaussian distribution that is used to sample predictions. DeepFactor [12] assumes that time series are exchangeable and decomposes the joint distribution into global and local time series. An LSTM is then used to learn a global deterministic function, while a Gaussian Process is used to learn local uncertainty. The outputs of these two functions are used to generate the forecasting distribution.

2.2 Diffusion Model

The diffusion model is a type of generative model that learns the underlying distribution of data by transforming data samples into Gaussian noise. It has been proven to outperform Generative Adversarial Network (GAN) in image synthesis tasks [13,14]. In recent years, its potential has been explored in various fields such as protein sequence analysis [15], structured time series forecasting [16], threat detection [17], and audio synthesis [18]. The diffusion model has been increasingly used in probabilistic time series forecasting, as it has achieved state-of-the-art performance. This is demonstrated in studies such as TimeGrad [7] and CSDI [8]. TimeGrad conditions the diffusion process on hidden states extracted from historical data using an LSTM or GRU network, while CSDI takes as input a matrix filled with both historical data and target, and a mask matrix indicating missing values. Its backbone is based on DiffWave [19], which enables correlation across all features and time points. The results from TimeGrad and CSDI have shown that the performance of the diffusion model can be optimized by selecting the appropriate backbone for the task.

2.3 Transformer

The Transformer uses an encoder-decoder architecture [20], widely applied in Natural Language Processing (NLP) tasks. It is known for the ability to attend to specific parts of the input sequence, rather than considering the entire sequence equally. This is achieved through the use of an attention mechanism, which assigns a weight to each element of the input sequence, indicating the amount of attention the model should allocate to each element when making predictions. A notable extension of the Transformer is the Bert model [21], which has demonstrated strong semantic representation capabilities. ICU data share many characteristics with NLP data given the vast capacity of dictionaries. While multiple aspects of a patient's condition can be monitored in the ICU, only a limited number of them are recorded at any given time. Additionally, the recording intervals may be irregular, which present challenges in data processing. Furthermore, different patients may have different items recorded, and all possible items must be considered in the analysis. All of these factors contribute to the extreme sparsity of ICU data.

3 Methods

3.1 TDSTF

Generative models aim to learn the underlying distribution of an observation dataset, but the main challenge lies in marginalizing out the latent variables to calculate the normalizing constant for a valid distribution, which is intractable [22]. Variational inference is a commonly used solution, transforming the distribution calculation into an optimization problem. The diffusion model applies variational inference to approximate a data distribution. It is based on the idea that a continuous Gaussian diffusion process can be reversed with the same functional form as the forward process [23]. After learning the reverse process, the input pure noise will converge to data points sampled from the modeled distribution. This can be approximated with a discrete Gaussian diffusion process, given a large enough number of T diffusion steps. The Diffusion model is a powerful generative model that has achieved state-of-the-art performance in various applications.

We propose a Transformer-based Diffusion Probabilistic Model for Sparse Time Series Forecasting (TDSTF). The diffusion processes in terms of time series forecasting are manifested in Figure 1, divided into forward and reverse trajectories. Conditioned on the history observation $\mathbf{x}_0^{co}$, The purpose of our method is to learn $p_\theta(\mathbf{x}_0^{p}|\mathbf{x}_0^{co})$ parameterized by θ that approximates $q(\mathbf{x}_0^{ta}|\mathbf{x}_0^{co})$, so that $\mathbf{x}_0^{p}$ predicts the target $\mathbf{x}_0^{ta} \sim q(\mathbf{x}_0^{ta}|\mathbf{x}_0^{co})$. During forward trajectory, a small amount of noise is added to the training data $\mathbf{x}_0^{ta}$ at each step t, and finally the distribution is destroyed. The noise amount at each step is determined by a variance schedule $\beta_1, ..., \beta_T$. The forward process is expressed as a Markov chain:

$$\begin{aligned} q(\mathbf{x}_{1:T}^{ta}|\mathbf{x}_0^{ta}) &:= \prod_{t=1}^{T} q(\mathbf{x}_t^{ta}|\mathbf{x}_{t-1}^{ta}) \\ q(\mathbf{x}_t^{ta}|\mathbf{x}_{t-1}^{ta}) &:= N(\mathbf{x}_t^{ta}; \sqrt{1-\beta_t}\mathbf{x}_{t-1}^{ta}, \beta_t\mathbf{I}) \end{aligned} \tag{1}$$

where N stands for Gaussian distribution parameterized by $\sqrt{1-\beta_t}\mathbf{x}_{t-1}^{ta}$ as its mean and $\beta_t\mathbf{I}$ as its variance. Table 1 lists all notations used in our method for convenience. The bold font denotes a vector of variables. For instance, $\mathbf{x} = (x_1, x_2, ..., x_n)$ where n is the dimension of the vector.

The reverse process begins with a Gaussian noise sample $\mathbf{x}_T^{p} \sim N(\mathbf{0}, \mathbf{I})$. At each time step, $\mathbf{x}_t^{p}$ is denoised, until $\mathbf{x}_0^{p}$ is sampled from a distribution that is modeled to be close to the distribution

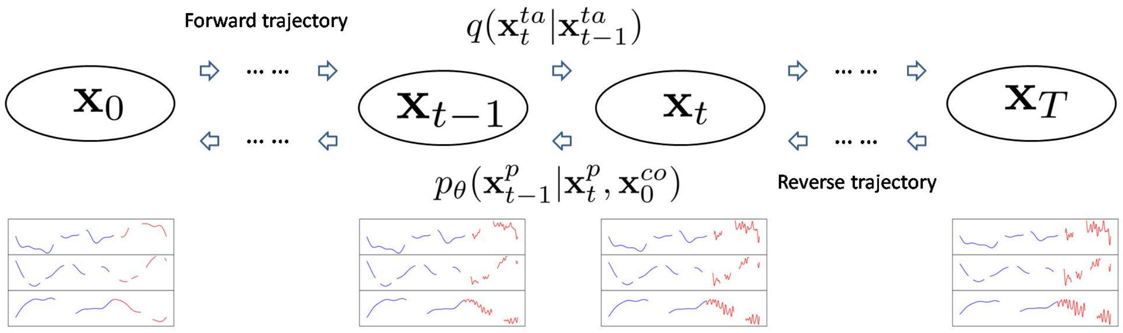

Forward trajectory
$q(\mathbf{x}_t^{ta}|\mathbf{x}_{t-1}^{ta})$
$\mathbf{x}_0$
$\mathbf{x}_{t-1}$
$\mathbf{x}_t$
$\mathbf{x}_T$
$p_\theta(\mathbf{x}_{t-1}^{p}|\mathbf{x}_t^{p}, \mathbf{x}_0^{co})$
Reverse trajectory

Figure 1: Diagram of the diffusion processes in our forecasting model. The blue curves indicate the history events the as the conditional data $\mathbf{x}_0^{co}$. The red curves symbolize the noisy target $\mathbf{x}_t^{ta}$ at time step t in the forward trajectory or the intermediate result $\mathbf{x}_t^{p}$ during prediction. The target is represented by $\mathbf{x}_0^{ta}$. The gaps among the curves symbolize the missing values in the sparse data. The forward trajectory q adds noise of increasing levels to $\mathbf{x}_0^{ta}$. The reverse trajectory p_θ then removes the noise from the pure noise $\mathbf{x}_T^{p}$ to generate samples.

Table 1: Notations used in the proposed model.

Notation	Description
$\mathbf{x}_t^{ta}$	Noisy target at step t
$\mathbf{x}_t^{p}$	Intermediate result during inference at step t
$\mathbf{x}_0^{co}$	Conditional data
ϵ_θ	Backbone network of the diffusion model parameterized by θ
q	Distribution of noisy target
p_θ	Distribution of output from the diffusion model
β_t	Diffusion schedule at step t
T	Number of diffusion steps

of the training data. This process can also be represented as a Markov chain:

$$\begin{aligned} p_\theta(\mathbf{x}^p_{0:T}|\mathbf{x}^{co}_0) &:= p(\mathbf{x}^p_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}^p_{t-1}|\mathbf{x}^p_t, \mathbf{x}^{co}_0) \\ p_\theta(\mathbf{x}^p_{t-1}|\mathbf{x}^p_t, \mathbf{x}^{co}_0) &:= N(\mathbf{x}^p_{t-1}; \mu_\theta(\mathbf{x}^p_t, t|\mathbf{x}^{co}_0), \Sigma_\theta) \end{aligned} \tag{2}$$

where μ_θ and Σ_θ are learnable functions that generate the mean and variance of the modeled distribution. Equation 1 implies that $\mathbf{x}^{ta}_t = \sqrt{\hat{\alpha}_t}\mathbf{x}^{ta}_{t-1} + \sqrt{1-\hat{\alpha}_t}\epsilon = \sqrt{\alpha_t}\mathbf{x}^{ta}_0 + \sqrt{1-\alpha_t}\epsilon$, where $\epsilon \sim N(\mathbf{0}, \mathbf{I})$. This means that we can sample at any time step t during the forward process, based only on $\mathbf{x}^{ta}_0$. As a result, it is advantageous to construct μ_θ and Σ_θ as follows:

$$\mu_\theta(\mathbf{x}^p_t, t|\mathbf{x}^{co}_0) = \frac{1}{\sqrt{\hat{\alpha}_t}}(\mathbf{x}^p_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(\mathbf{x}^p_t, t|\mathbf{x}^{co}_0)) \tag{3}$$

$$\Sigma_\theta(t) = \sigma^2_t = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t \ \ (t > 1) \tag{4}$$

where $\hat{\alpha}_t = 1 - \beta_t$ and $\alpha_t = \prod_{i=1}^{t}\hat{\alpha}_i$, in order for $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to be close to $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ as stated in [24].

3.2 Training procedure

The objective of the model training is to maximize the Evidence Lower Bound of $p_\theta(\mathbf{x}^p_0|\mathbf{x}^{co}_0)$ [13]. It can be expressed as the following equation:

$$\min_\theta E_t\|\epsilon - \epsilon_\theta(\mathbf{x}^{ta}_t, t|\mathbf{x}^{co}_0)\|^2_2 \tag{5}$$

where $\epsilon_\theta(\mathbf{x}^{ta}_t, t|\mathbf{x}^{co}_0)$ is a function that can be learned to predict ϵ for each step of the denoising process. The entire training procedure is illustrated in Figure 2. Before being input into ϵ_θ, $\mathbf{x}^{ta}_0$ and $\mathbf{x}^{co}_0$ are transformed into triplets.

When dealing with extremely sparse data, traditional methods such as aggregation and imputation are not effective. Aggregation results in poor resolution and loss of temporal information, while imputation introduces excessive noise. To overcome these issues, the triplet form compactly stores sparse data. Each triplet contains a feature, time, value, and a mask bit indicating the presence or absence of data, represented by 1 or 0, respectively. The absolute time of the valid data points from the raw data is transformed into a relative time range of 40 minutes. Figure 3 gives an illustration of converting a sparse matrix to a triplet representation. If the number of conditional triplets exceeds the input size (preset as a hyperparameter), the feature selection module prioritizes data points of the same features as the target, most correlated to the target data (as reported in [5,6]), and then fills in the remaining input triplets randomly. The Mean Squared Error (MSE) between the predicted noise and the Gaussian noise ϵ is used as the loss function.

We construct ϵ_θ using a deep neural network that is divided into two stages: the front stage and the back stage. This architecture is depicted in Figure 4. The front stage maps the data into higher-dimensional spaces. The feature, value, and time components of the triplet are each transformed into vectors. The embedding module maps the triplet features into vectors, the linear layer projects the triplet values into vectors, and a group of sinusoidal functions transforms the triplet times into vectors. To avoid disturbing the missingness representations, the representations of the feature and time also incorporate information about the missingness, and the values of triplets with masks of 0 are set to 0 (the mean value for all features after normalization). A random diffusion step is applied in each iteration to generate noisy target values. The diffusion step is represented as a

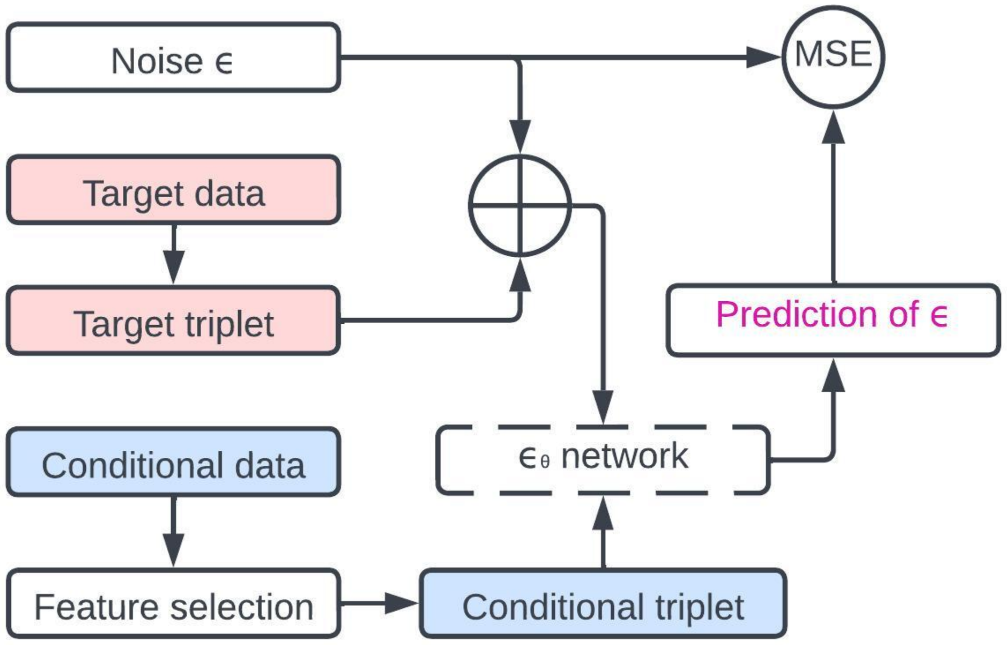
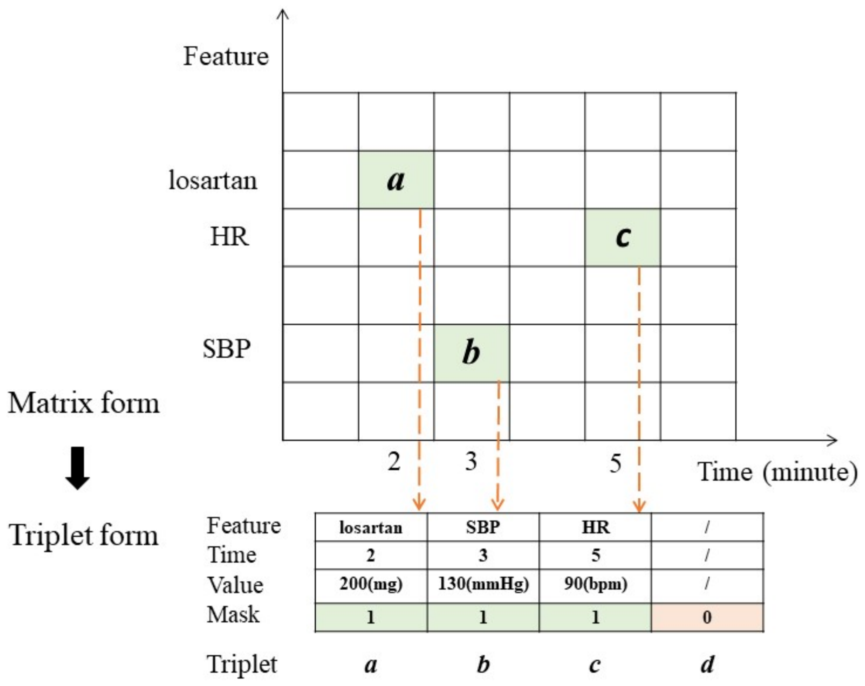


Figure 2: The training procedure of our forecasting model involves optimizing the Mean Squared Error (MSE) between the noise prediction $\epsilon_\theta(x_t^{ta}, t|x_0^{co})$ and ϵ that determines the noise amount added to the target. This MSE calculation serves as the loss function, which the model aims to minimize during training in order to improve its performance in predicting sparse time series data. The implementation of ϵ_θ in the dashed box will be expanded and explained in detail later.



Feature	losartan	SBP	HR	/
Time	2	3	5	/
Value	200(mg)	130(mmHg)	90(bpm)	/
Mask	1	1	1	0
Triplet	*a*	*b*	*c*	*d*

Figure 3: An illustration of converting a sparse matrix to triplet representation. In this example, a patient received 50 milligram (mg) of losartan at the second minute. A Heart Rate (HR) of 90 beats per minute (bpm) and a Systolic Blood Pressure (SBP) of 130 millimeters of mercury (mmHg) are recorded at the third and fifth minute respectively. Assuming all other data points in the matrix are missing, and the 3 valid event records $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ line up in an array of triplets.

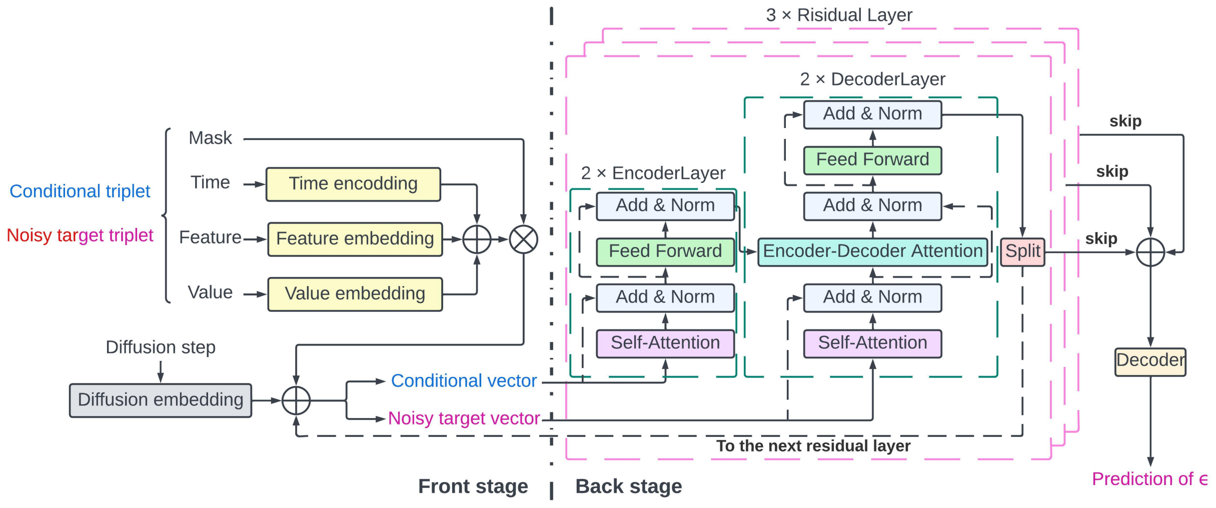


Figure 4: The architecture of the ϵ_θ network, designed for predicting ϵ at the current step. It uses residual connections, as indicated by the black dashed arrows. The residual connections allow information to be passed directly from one layer to another, allowing the network to effectively learn complex relationships in the data while mitigating the risk of vanishing gradients. By incorporating these connections, the network can better preserve important features in the data and make more accurate predictions.

vector, which is obtained from a lookup table and projected through linear layers. The result of all these vectors is fed into the back stage, which is a three-layer residual network that helps prevent the vanishing gradient problem [18, 25].

Each residual layer consists of two layers of Transformer encoder and two layers of Transformer decoder. The conditional and target vectors are input into the encoder and decoder, respectively. Within the encoder layers, each conditional triplet is transformed into Query (Q), Key (K), and Value (V) vectors. The self-attention layer calculates the correlation strength between one triplet and every other triplet through the following equation:

$$Attention = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

where d_k is the dimension of the Q and K vectors and the scaling factor $\sqrt{d_k}$ enhances the model's robustness. The target triplets then undergo self-attention calculation. The K and V vectors output from the top encoder layer are input into the decoder's encoder-decoder attention layers, where the target Q vectors extract the relevant information from the conditional sequence. The mask value of 0 for triplets acts similarly to the End Of Sentence (EOS) indicator in NLP tasks. The final prediction of ϵ is obtained by adding up the skip connections and transforming the result through another decoder based on a CNN. The multi-headed attention mechanism also enhances the model's representational power by expanding latent subspaces with multiple independent sets of Q, K, and V vectors. The training process is outlined in Algorithm 1.

3.3 Inference procedure

The values of the predicted triplets in the model are initially Gaussian noise and correspond to the first diffusion step of the reverse trajectory (step T in Figure 1). These noisy values are

Algorithm 1 Training

```
Input: $\mathbf{x}_0^{ta} \sim q(\mathbf{x}_0^{ta}|\mathbf{x}_0^{co})$, $\mathbf{x}_0^{co}$
Output: $\epsilon_\theta$ network trained
  repeat
    $t \sim Uniform(1, ..., T)$
    $\epsilon \sim N(\mathbf{0}, \mathbf{I})$
    Take gradient descent step on $\nabla_\theta \|\epsilon - \epsilon_\theta(\mathbf{x}_t^{ta}, t|\mathbf{x}_0^{co})\|_2^2$
  until Converged
```

then denoised using the output from ϵ_θ according to Equation 2, which is written as $\mathbf{x}_{t-1}^p = \frac{1}{\sqrt{\hat{\alpha}_t}}(x_t^p - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t^p, t|x_0^{co})) + \sigma_t \mathbf{z}$, where $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$. The features and times of the predicted triplets are input directly into the model, providing important contextual information. The denoised values are then fed back into the model to generate a prediction of ϵ for the next step and this process is repeated until the final step of the reverse trajectory to output $\mathbf{x}_0^p$. The detailed process for this inference is described in Algorithm 2.

Algorithm 2 Inference

```
Input: $\mathbf{x}_T^p \sim N(\mathbf{0}, \mathbf{I})$, $\mathbf{x}_0^{co}$
Output: $\mathbf{x}_0^p$ (prediction of $\mathbf{x}_0^{ta}$)
  for $t = T, ..., 2$ do
    $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$
    $\sigma_t^2 = \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$
    $\mathbf{x}_{t-1}^p = \frac{1}{\sqrt{\hat{\alpha}_t}}(x_t^p - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t^p, t|x_0^{co})) + \sigma_t \mathbf{z}$
  end for
  $\mathbf{x}_0^p = \frac{1}{\sqrt{\hat{\alpha}_1}}(x_1^p - \frac{\beta_1}{\sqrt{1-\alpha_1}}\epsilon_\theta(x_1^p, 1|x_0^{co}))$
```

4 Experiments

4.1 Data and preprocessing

In this study, we evaluate the model using the MIMIC-III dataset as outlined in [26]. This dataset holds health information for over 46 thousand patients admitted to the Beth Israel Deaconess Medical Center (Boston, MA) between 2001 and 2012. To prepare the data for analysis, we follow a preprocessing approach, depicted in Figure 5, which involves three steps. First, we exclude records related to pediatric patients. Second, we remove records with abnormal feature values. The features represent the events that happen to the patients, such as vital signs, medication usage, and biochemical test results. Finally, we retain only the first 40 consecutive minutes of each ICU stay. Of these 40 minutes, the former 30 minutes are used for the conditional data and the latter 10 minutes for the target data. Both the conditional and target data are screened to ensure non-emptiness. Part of our preprocessing approach has been borrowed from [27].

The goal of this study is to predict three target features: heart rate, systolic blood pressure, and diastolic blood pressure. To prevent subject repetition, we divide the dataset into training and testing sets, randomly selecting 80% and 20% of the data by subject, respectively. We calculate the mean and standard deviation of each feature from the training dataset and use these values to normalize the training and testing datasets before conducting experiments. After implementing the

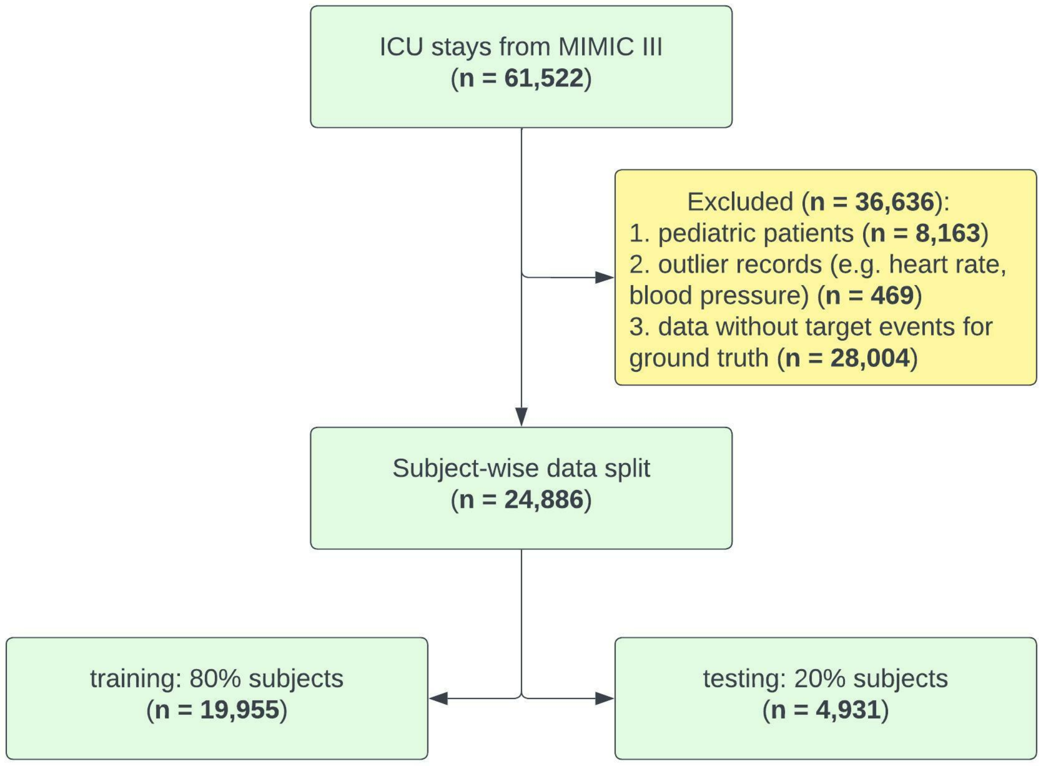

ICU stays from MIMIC III
(**n = 61,522**)
Excluded (**n = 36,636**):
1. pediatric patients (**n = 8,163**)
2. outlier records (e.g. heart rate, blood pressure) (**n = 469**)
3. data without target events for ground truth (**n = 28,004**)
Subject-wise data split
(**n = 24,886**)
training: 80% subjects
(**n = 19,955**)
testing: 20% subjects
(**n = 4,931**)

Figure 5: Illustration of the data preprocessing steps involved in this study. The raw data was obtained from the MIMIC-III dataset, which contains vital signs of patients in the ICU. We first extracted and organized the relevant data, which included the patient's vital signs and demographic information. The data was then preprocessed and normalized to ensure that it was suitable for input into the TDSTF model. The number of ICU stays, denoted by n, was also counted during this step. The preprocessed data was then used to train and evaluate the TDSTF model for its performance in predicting sparse time series data in the ICU setting.

Table 2: Subject number by sex; mean and standard deviation of subject age and target features.

	Training n=19,955	Testing n=4,931
M/F	5244/3919	1281/957
Age	65.43±16.35	65.36±15.94
HR	90.82±21.56	91.24±22.27
SBP	117.74±27.95	117.76±30.45
DBP	60.34±16.86	59.90±17.05

HR is heart rate in beats per minute (bpm); SBP is systolic blood pressure in millimeters of mercury (mmHg); DBP is diastolic blood pressure in mmHg; M is male; F is female; n is the number of ICU stays.

exclusion criteria in the preprocessing step, a total of $11,401$ subjects were included in the study. From these eligible subjects, we extracted $24,886$ ICU stays to test our method. Table 2 presents statistics for the eligible subjects and target features in both the training and testing datasets before normalization. Subjects in the training and testing datasets combined were generally older (age of 65.42 ± 16.27) and male (57.2%). Note that one subject corresponds to one or more ICU stays.

4.2 Model hyperparameters

In this study, we set the dimension of all vectors after embedding or encoding to 128 and the number of diffusion steps to 50. The noise-adding amount is set to $\beta_1 = 10^{-4}$ at the first diffusion step and $\beta_2 = 0.5$ at the last diffusion step, with a quadratic diffusion schedule previously described in [8]. The Transformer in ϵ_θ is equipped with 8 heads, and the mini-batch size is set to 32 samples. We use the ADAM optimizer [28] with a learning rate 10^{-3} which decays by a factor of 0.8 every 10 epochs. The training data is further split into an $80 - 20\%$ ratio, with the latter portion used for validation during early stopping to prevent overfitting. Starting from the $100th$ epoch, the model is tested with the validation dataset every 10 epochs. The training process will stop if the performance does not improve after 3 consecutive trials or after 200 total epochs. As is illustrated in Figure 6, the histogram shows the number of valid conditional data points in ICU samples from the training dataset, with a mean of 12.9. A total of 129 features are extracted from the raw data. So, each conditional input in matrix form has a size of 129 by 30 (minutes). This results in an average missingness of over 99.5% in the matrix form. More than 99.5% of the ICU samples from the training dataset have valid conditional data points that are less than or equal to 60, and we set the input size to 60.

4.3 Metrics

To thoroughly assess the performance of the model, multiple metrics are utilized to measure the differences between the forecast and the ground truth time series.

One of the simplest and most commonly used metrics for this purpose is the Mean Squared Error (MSE). It is defined as the squared $L2$ distance between each element in the input $\mathbf{x}$ and

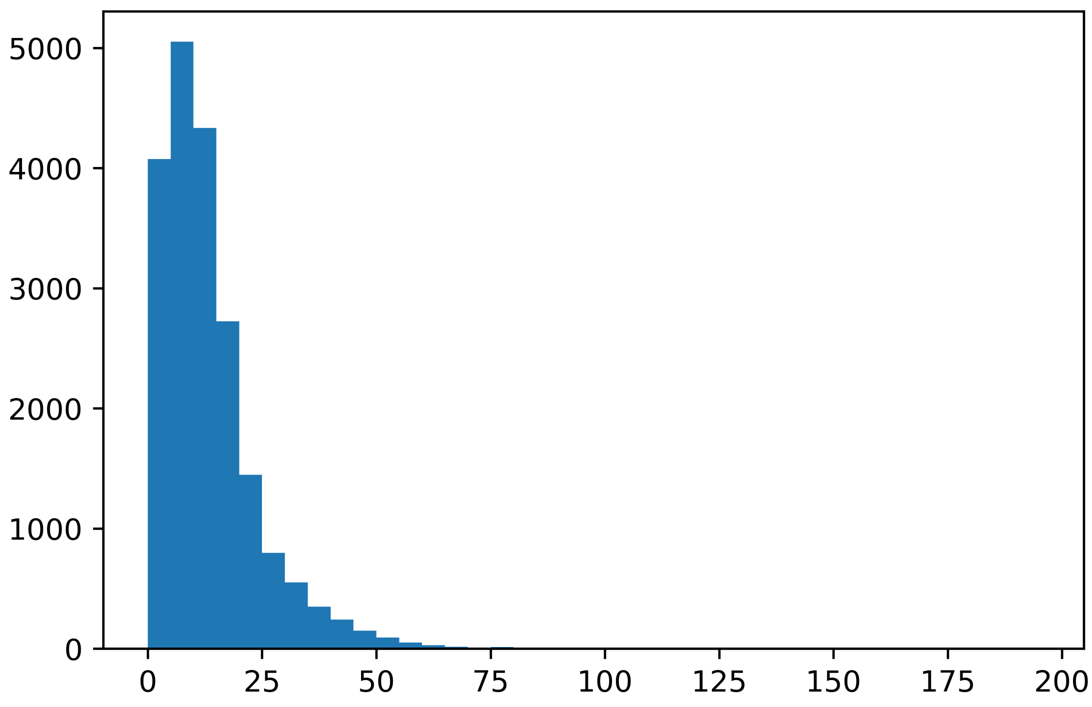


Figure 6: Histogram showing the distribution of the number of valid conditional data points of ICU stays from the training dataset. The mean number is 12.9.

target $\mathbf{y}$, calculated as

$$MSE(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2 \tag{7}$$

In our case, the median values of the generated samples act as the deterministic predictions for the calculation of MSE with the ground truth. Lower MSE value indicates better performance.

Another widely used metric for evaluating the accuracy of probabilistic forecasts is the Continuous Ranked Probability Score (CRPS). It is defined as:

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - I_{y \geq x})^2 dy \tag{8}$$

where F is the modeled Cumulative Distribution Function (CDF), x is the observation, and I is the Indicator function. Figure 7 illustrates how CRPS is calculated. It provides a comprehensive measure of forecast accuracy by evaluating the mean difference between the predicted CDF and the ground truth, taking into account both under-prediction and over-prediction. The smaller the CRPS value, the more the modeled distribution concentrates around the ground truth, indicating better performance.

To calculate the Normalized Average CRPS (NACRPS), when predicting a normalized target value x^{ta}, we adopt the same discrete quantile loss as in [8]:

$$NACRPS = \sum_{x^{ta}} \sum_{i=1}^{19} 2\Lambda_{0.05i}(x^p_{0.05i}, x^{ta})/19 \sum |x^{ta}| \tag{9}$$

$$\Lambda_\alpha(q, x) = (\alpha - I_{q \geq x})(x - q) \tag{10}$$

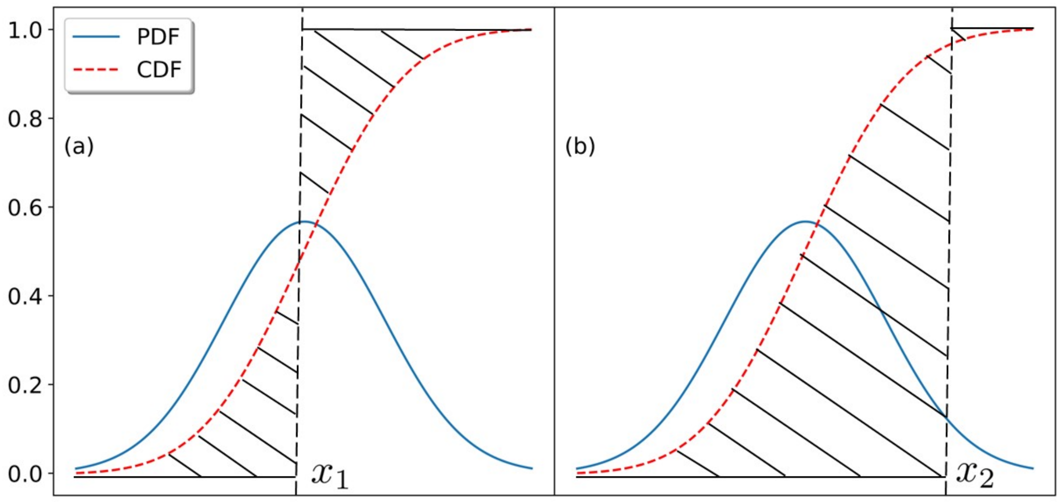


Figure 7: A schematic illustration of the Continuous Ranked Probability Score (CRPS). The blue curves represents the modeled probability distribution functions and the red dashed curves represent their cumulative distribution functions. The subplots (a) and (b) depict the same distribution with different ground truth x_1 and x_2. The black hatched area representing the integration as the CRPS value in (a) is smaller than that in (b), meaning the modeled distribution evaluates x_1 better.

where $x^p_{0.05i}$ is the value of $0.05i$ quantile level from the generated samples. The scaling factor $\sum |x^{ta}|$ represents the variance of the target distribution which becomes more difficult to predict as the variance increases. NACRPS calculated as the CRPS divided by this scaling factor provides a more accurate evaluation of the model.

We implemented the tests using the Python programming language and the Pytorch framework [29]. The experiments were conducted on a workstation that was equipped with an Intel i7-12700K processor and an Nvidia RTX 3090 graphics card.

4.4 Results

Four models were used as the baseline for predicting vital signs in the MIMIC-III dataset. To handle the missing data, the MQ-RNN, DeepAR, and DeepFactor models used a mean imputation of 0, while the CSDI model used a mask matrix to indicate missing data positions. The TimeGrad model was not included due to it being outperformed by the CSDI model and both models being diffusion models. Furthermore, TimeGrad does not have a mechanism for handling missing data, making it unsuitable for tasks with sparse time series.

The experimental results are presented in Table 3, with three trials conducted. Both the NACRPS and MSE results indicate that the CSDI and TDSTF models performed much better than the other models. On average, TDSTF obtained 0.4438 and 0.4168, while CSDI obtained 0.5439 and 0.6358 for NACRPS and MSE respectively. Thus, TDSTF improved NACRPS and MSE by 18.9% and 34.3% over CSDI. Figure 8 shows the violin histograms of NACRPS and MSE per ICU sample for the first trial of TDSTF. The results concentrate on the median values 0.4651 and 0.1677 of NACRPS and MSE, which shows the model's robustness.

Table 3: NACRPS and MSE of the proposed model in comparison with the baseline.

	MQ-RNN	DeepAR	DeepFactor	CSDI	TDSTF (Ours)
NACRPS	1.0276	0.9930	0.8221	0.5515	**0.4379**
	1.0367	0.9615	0.8207	0.5455	**0.4526**
	1.0385	0.9653	0.8238	0.5439	**0.4408**
MSE	1.1014	1.0295	1.0325	0.6430	**0.4061**
	1.1237	1.0304	1.0332	0.6358	**0.4434**
	1.1186	1.0289	1.0324	0.6236	**0.4008**

NACRPS is normalized average continuous ranked probability score; MSE is mean squared error. The best results are shown bold.

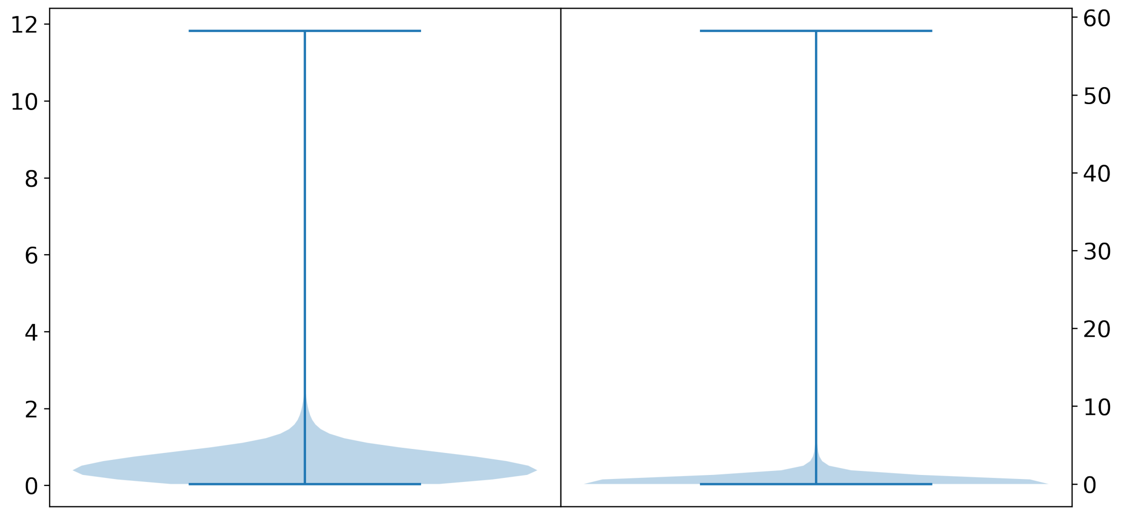
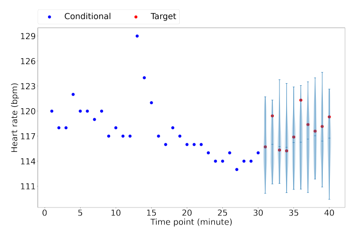
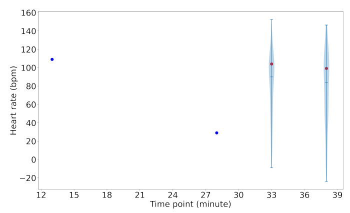


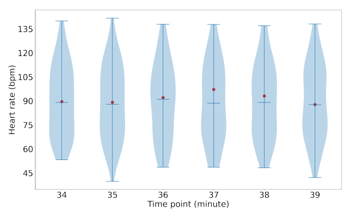
Figure 8: Violin histograms of the NACRPS in the left and the MSE in the right per ICU sample from the first trial of TDSTF. The results for each metric concentrate around the median values of 0.4651 and 0.1677 respectively.



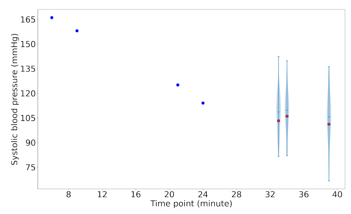
(a) An increase in HR.



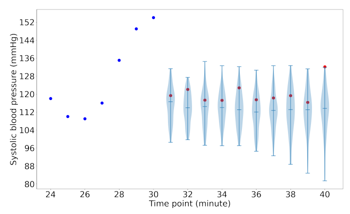
(b) A sudden increase in HR.



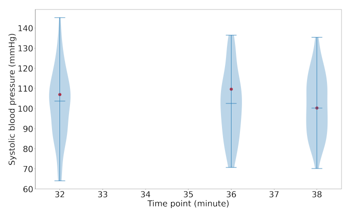
(c) HR prediction without HR conditional data.



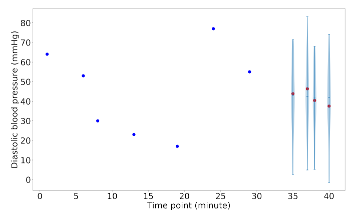
(d) A decrease in SBP.



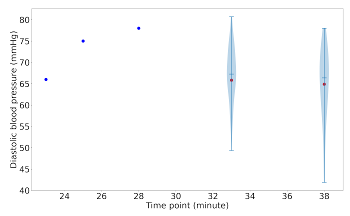
(e) A sudden decrease in SBP.



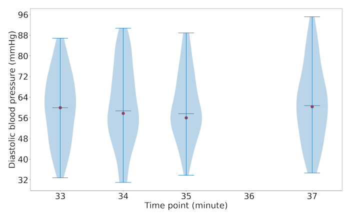
(f) SBP prediction without SBP conditional data.



(g) A decrease in DBP.



(h) A sudden decrease in DBP.



(i) DBP prediction without DBP conditional data.

Figure 9: Examples from the TDSTF test results from the first trial of TDSTF. HR predictions in subplots (a)-(c) are in beats per minute (bpm), while SBP and DBP predictions in subplots (d)-(i) are in millimeters of mercury (mmHg). The horizontal axis represents the relative time in minutes for each ICU stay. The red dots indicate target feature values, and the blue dots indicate conditional values of the same target feature. Forecasts based on all 129 features show high accuracy, even without the conditional data as is shown in (c), (f), and (i). The 95% confidence intervals are shown as the areas between top and bottom bars of the violin histograms, with wider areas corresponding to more confidence and the middle bars representing the median values of the generated data points.

Figure 9 shows some forecasting examples from the first trial of TDSTF. All predicted values fall into the 95% confidence interval. Subplot (a) predicts an increasing trend of HR over 100 beats per minute (bpm). A sudden increase in HR to around 100 bpm is captured in subplot (b). Both of them imply potential tachycardia. Subplot (d) shows a continuously decreasing SBP, and possible hypotension may be expected. Subplot (e) forecasts recovery from hypertension into a period of a normotensive SBP of around 120 millimeters of mercury (mmHg). A decreasing trend of DBP below 60 mmHg is seen in subplot (g), suggesting the patient is at risk for dangerously low blood pressure. Subplot (h) captures a sudden decrease in DBP, which should alert clinicians of the potential for further hypotension. The model also performed well on test cases without target conditional data, as is shown in subplots (c), (f), and (i).

The complexity of TDSTF was compared with CSDI using PyTorch-OpCounter [1]. The com-

[1]`https://github.com/Lyken17/pytorch-OpCounter`

parison considered the number of parameters and the Multiply-Accumulate (MAC) of the models. The number of parameters determines the model complexity, while the MAC number refers to the computational costs. The results showed that CSDI had 0.28 million parameters, while TDSTF had 0.56 million parameters. The training time for CSDI was 12 hours, while for TDSTF it was only 0.5 hours. The inference per sample consumed more than 55 billion MACs in CSDI and 0.79 million in TDSTF. Consequently, 14.913 and 0.865 seconds were required for inference per sample in CSDI and TDSTF respectively, and TDSTF was more than 17 times faster. The fact that TDSTF is more complex than CSDI, but consumes much less computation, verifies our assumption of the CSDI overhead due to its large amount of missingness input.

5 Discussion

This study presents a deep learning model, TDSTF, that can accurately predict sparse time series data in the ICU setting. The model was trained on MIMIC-III data and tested on several performance metrics. The results showed that TDSTF outperforms several baseline models and can detect important changes in the vital signs of ICU patients. The model consists of a residual network based on Transformer, which allows for the ability to capture complex temporal dependencies and efficient computation and storage.

The TDSTF model was developed to accurately forecast sparse time series data, without making assumptions about the underlying distribution. The model was trained and tested using the MIMIC-III ICU data, and evaluated using NACRPS and MSE. It performed better than the baseline models, including MQ-RNN, DeepAR, DeepFactor, and CSDI. The mask used in CSDI reduced disturbance but could not guarantee enough exclusion of invalid information from the sparse time series input. Over 99.5% of conditional data points were missing on average, which could negatively impact performance and waste computational resources. Setting the input size of conditional triplets to 60 for TDSTF improved the signal-to-noise ratio and outperformed CSDI because of its improved representational power. MQ-RNN, DeepAR, and DeepFactor used LSTM to extract hidden states, but could not capture temporal dependencies at arbitrary distances. Furthermore, they cannot fully utilize parallel computation, hindering model complexity [30]. The Transformer-based ϵ_θ network solved these issues. While DeepAR and DeepFactor's Gaussian process was too simple to represent high-dimensional distributions. TDSTF, on the other hand, directly learns the underlying distributions.

The TDSTF is effective because it accurately captures temporal patterns in sparse time series data. As is shown in Figure 9. In addition to high accuracy in the slow change cases, the model successfully recognizes sudden changes, which are important indicators of changes in system states. The high accuracy in predicting vital signs without target conditional data shows that the model can extract interrelations among all features. The quick response of TDSTF is also crucial in the ICU setting. All these benefits help detect deteriorations in time, such as sudden tachycardia or developing hypotension and thus are important for timely interventions in the ICU setting.

However, there is still room for improving TDSTF. One issue is that the limitations in the size of the Transformer input can hinder its effectiveness, as the memory and time consumption increases quadratically with its size [31]. Moreover, the slow speed of the diffusion models can make fine-tuning their hyperparameters a time-consuming process. Additionally, the data used in the study may introduce bias into the model as the eligible subjects were generally older and male (as shown in Table 2), which could limit its generalizability to other populations. These issues highlight the importance of considering both the technical limitations and the demographic characteristics of the data when developing and evaluating deep learning models for forecasting sparse time series.

From a clinical application perspective, it is possible to include core body temperature, breathing rate, and oxygen saturation as additional forecasting outputs instead of this work's current three vital signs. It is essential to notice that while traditional ICUs use five vital signs, today, most ICUs also include assessments of pain, level of consciousness, and urine output as additional signals: the eight vital signs of patient monitoring. Here, assessments of pain and level of consciousness are more subjective than objective and thus require the proposed model to have the ability to provide reasonable predictions while accepting fuzzy inputs.

We also plan to apply the proposed TDSTF in Cardiovascular ICU (CVICU). As a special ICU unit, CVICU [32] monitors and treats people with heart surgery or other heart procedures. It also takes care of people before and after a heart transplant. Patients may stay in the CVICU for several days or weeks. CVICU involves 24/7 monitoring by cardiac intensivists, who are doctors/nurses with specialized training in providing intensive care personalized to the needs of heart and vascular patients. TDSTF may be extended to CVICU applications. Here, ECG signals will be incorporated into the inputs of the proposed model. Additional techniques need to be included in the current TDSTF model to identify acute myocardial infarction, ventricular fibrillation, arrhythmia, cardiac arrest, heart failure, shock, etc., together with active interventions such as percutaneous coronary intervention, defibrillation, respiratory management, and blood purification therapy.

An exciting application of the proposed TDSTF will be in false alarm reduction. Although most existing alarms in the ICU are false, this work may be extended to decrease the frequency of false alarms generated by bedside monitors in the ICU. For example, we may incorporate additional techniques in signal processing, feature extraction, and optimized machine learning. We ensure the heartbeats are properly tagged throughout the signal processing stage. We collect a set of signal quality indicators during feature extraction and characteristics pertinent to the arrhythmic alarms to differentiate between noise and normal physiological signals.

6 Conclusion

The TDSTF model offers a solution to the limitations of current state-of-the-art probabilistic forecasting models. By incorporating a Transformer-based backbone and using a triplet method to avoid input noise and increase speed, TDSTF was able to outperform the main baseline model, CSDI, on both NACRPS and MSE by an average of 18.9% and 34.3% respectively, and more than 17 times faster than CSDI in inference. The results of this study demonstrate the improved accuracy and efficiency of TDSTF for sparse time series forecasting, making it a more practical and valuable tool for forecasting vital signs in the ICU setting. Additionally, TDSTF's ability to capture temporal dependencies across all features further enhances its potential for real-world applications. Further studies of TDSTF's performance should be performed in datasets with other patient characteristics. If further validated, performance in real-time scenarios would be indicated.

Conflict of Interest
S.F.Q. is a consultant for Bryte Bed, Guidepoint Global, Cowen Services, Whispersom, DR Capital and Best Doctors and is a member of the Hypopnea Taskforce of the American Academy of Sleep Medicine. Other authors have nothing to disclose.

Data Availability Statement
The MIMIC-III dataset used in this study is available at https://physionet.org/content/mimiciii/1.4/. In order to access the data, researchers must complete the application form.

Funding Statement
This work was supported by grants from the National Heart, Lung, and Blood Institute (#R21HL159661) and the National Science Foundation (#2052528). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Tsuneaki Kenzaka, Masanobu Okayama, Shigehiro Kuroki, Miho Fukui, Shinsuke Yahata, Hiroki Hayashi, Akihito Kitao, Daisuke Sugiyama, Eiji Kajii, and Masayoshi Hashimoto. Importance of vital signs to the early diagnosis and severity of sepsis: association between vital signs and sequential organ failure assessment score in patients with sepsis. *Internal Medicine*, 51(8):871–876, 2012. 1

[2] Joo Heung Yoon, Lidan Mu, Lujie Chen, Artur Dubrawski, Marilyn Hravnak, Michael R Pinsky, and Gilles Clermont. Predicting tachycardia as a surrogate for instability in the intensive care unit. *Journal of Clinical Monitoring and Computing*, 33:973–985, 2019. 1

[3] Shiyu Liu, Jia Yao, and Mehul Motani. Early prediction of vital signs using generative boosting via lstm networks. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 437–444. IEEE, 2019. 1

[4] Shamsul Masum, John P Chiverton, Ying Liu, and Branislav Vuksanovic. Investigation of machine learning techniques in forecasting of blood pressure time series data. In *Artificial Intelligence XXXVI: 39th SGAI International Conference on Artificial Intelligence, AI 2019, Cambridge, UK, December 17–19, 2019, Proceedings 39*, pages 269–282. Springer, 2019. 1

[5] Xiaoli Liu, Tongbo Liu, Zhengbo Zhang, Po-Chih Kuo, Haoran Xu, Zhicheng Yang, Ke Lan, Peiyao Li, Zhenchao Ouyang, Yeuk Lam Ng, et al. Top-net prediction model using bidirectional long short-term memory and medical-grade wearable multisensor system for tachycardia onset: algorithm development study. *JMIR Medical Informatics*, 9(4):e18803, 2021. 1, 3.2

[6] Ratchakit Phetrittikun, Kerdkiat Suvirat, Thanakron Na Pattalung, Chanon Kongkamol, Thammasin Ingviya, and Sitthichok Chaichulee. Temporal fusion transformer for forecasting vital sign trajectories in intensive care patients. In *2021 13th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE, 2021. 1, 3.2

[7] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021. 1, 2.2

[8] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021. 1, 2.2, 4.2, 4.3

[9] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 1

[10] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017. 2.1

[11] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020. 2.1

[12] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In *International conference on machine learning*, pages 6607–6617. PMLR, 2019. 2.1

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2.2, 3.2

[14] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2.2

[15] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022. 2.2

[16] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 2.2

[17] Tsachi Blau, Roy Ganz, Bahjat Kawar, Alex Bronstein, and Michael Elad. Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint arXiv:2207.08089*, 2022. 2.2

[18] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 2.2, 3.2

[19] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 2.2

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2.3

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2.3

[22] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018. 3.1

[23] William Feller. On the theory of stochastic processes, with particular reference to applications. In *Selected Papers I*, pages 769–798. Springer, 2015. 3.1

[24] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 3.1

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3.2

[26] Alistair Johnson, Tom Pollard, and Roger Mark. Mimic-iii clinical database (version 1.4). *PhysioNet*, 10(C2XW26):2, 2016. 4.1

[27] Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022. 4.1

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4.2

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4.3

[30] Wim De Mulder, Steven Bethard, and Marie-Francine Moens. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, 30(1):61–98, 2015. 5

[31] Ming Ding, Chang Zhou, Hongxia Yang, and Jie Tang. Cogltx: Applying bert to long texts. *Advances in Neural Information Processing Systems*, 33:12792–12804, 2020. 5

[32] Shuji Kasaoka. Evolved role of the cardiovascular intensive care unit (cicu),doi: 10.1186/s40560-017-0271-7. pmid: 29299313; pmcid: Pmc5741934. *J Intensive Care*, pages 5 – 72, 2017 Dec 22. 5