Computer Methods and Programs in Biomedicine 246 (2024) 108060

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

A transformer-based diffusion probabilistic model for heart rate and blood pressure forecasting in Intensive Care Unit

Ping Chang [a], Huayu Li [a], Stuart F. Quan [b,c], Shuyang Lu [d,e], Shu-Fen Wung [f,g], Janet Roveda [a,f,h], Ao Li [a,f,*]

[a] Department of Electrical & Computer Engineering, The University of Arizona, Tucson, AZ, USA
[b] Division of Sleep and Circadian Disorders, Departments of Medicine and Neurology, Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA
[c] Asthma and Airway Disease Research Center, College of Medicine, The University of Arizona, Tucson, AZ, USA
[d] Department of Cardiovascular Surgery, Zhongshan Hospital, Fudan University, Shanghai, PR China
[e] The Shanghai Institute of Cardiovascular Diseases, Shanghai, PR China
[f] Bio5 Institute, The University of Arizona, Tucson, AZ, USA
[g] College of Nursing, The University of Arizona, Tucson, AZ, USA
[h] Department of Biomedical Engineering, The University of Arizona, Tucson, AZ, USA

ARTICLE INFO

Dataset link: https://physionet.org/content/mimiciii/1.4/

Keywords:
Deep learning
Time series forecasting
Sparse data
Vital signs
ICU

ABSTRACT

Background and Objective: Vital sign monitoring in the Intensive Care Unit (ICU) is crucial for enabling prompt interventions for patients. This underscores the need for an accurate predictive system. Therefore, this study proposes a novel deep learning approach for forecasting Heart Rate (HR), Systolic Blood Pressure (SBP), and Diastolic Blood Pressure (DBP) in the ICU.

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 Standardized Average Continuous Ranked Probability Score (SACRPS) 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.

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 [3,4]. 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 [5,6]. Vital sign prediction requires large amounts of data to develop accurate algorithms [7], but in many ICUs, the availability and quality of electronic health record data can be limited [8,9]. This makes it challenging to use classical statistical methods, which often

* Corresponding author at: Department of Electrical and Computer Engineering, The University of Arizona, 1230 E Speedway Blvd, Tucson, AZ, 85719, United States of America.
E-mail address: aoli1@arizona.edu (A. Li).

https://doi.org/10.1016/j.cmpb.2024.108060
Received 14 March 2023; Received in revised form 21 December 2023; Accepted 12 January 2024
Available online 8 February 2024

rely on manual feature engineering and cannot capture complex patterns [10,11]. 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 [12], the Long Short-Term Memory (LSTM) network is used to create a generative model, effectively reducing error propagation and improving HR prediction performance. In a comparison of Recurrent and Convolutional Neural Network (RNN and CNN) [13], using different horizon strategies on the MIMIC-II dataset, the bidirectional LSTM (Bi-LSTM) with the DIRMO strategy delivered the best predictions of HR and blood pressure. The TOP-Net model [14] 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 [15] predicts vital sign quantiles based on an attention mechanism, capturing anomaly temporal patterns and optimizing input time windows by calculating temporal importance.

Although these methods have shown promising results in vital sign forecasting, they still face several limitations when it comes to practical application in the ICU setting. First, these models require continuous monitoring of vital signs. However, given the complex and unpredictable conditions in the ICU, monitoring often becomes intermittent and sporadic. Second, these 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. Third, the datasets used to evaluate these 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 an effective and efficient deep-learning approach to forecast Heart Rate (HR), Systolic Blood Pressure (SBP), and Diastolic Blood Pressure (DBP) in the ICU setting. The use of diffusion probabilistic models (diffusion models for short) [16,17] in time series analysis has been gaining popularity due to their balance between flexibility and tractability [18]. 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 using a Transformer-based backbone leads to improved performance compared to baseline models. The ultimate goal is to promptly provide ICU caregivers with critical information 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.

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 [19] uses RNN 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 [20], hidden states are obtained through RNN, which are then input into linear layers with activation to generate the mean and variance of the assumed likelihood model that samples predictions. DeepFactor [21] assumes that time series are exchangeable and decomposes the joint distribution into global and local time series. RNN is subsequently employed to capture global non-linear patterns. Simultaneously, an assumed observation model is applied to discern local uncertainties conditioned on the global effects. The outputs of these two functions are used to generate the forecasting distribution. EnCQR [22] is a homogeneous ensemble approach. Member learners within this framework can be constructed utilizing various machine learning algorithms. EnCQR combines conformal prediction and quantile regression methodologies to construct prediction intervals devoid of reliance on specific distributional assumptions.

2.2. Diffusion model

The Diffusion model is a powerful generative model that learns the underlying distribution of data by transforming data samples into Gaussian noise, and has achieved state-of-the-art performance in various applications [23]. Initially, it attracted significant attention due to its superior performance in image synthesis compared to the Generative Adversarial Network (GAN) [24–26]. In recent years, its potential has expanded to domains such as protein sequence analysis [27], threat detection [28], audio synthesis [29], and probabilistic time series forecasting [16,17]. CSDI takes as input a matrix filled with both historical data and target and a mask matrix indicating missing values [17]. Its backbone is based on DiffWave [30], which enables correlation across all features and time points. The results from CSDI have shown that the diffusion model can be optimized by selecting the appropriate backbone for specific tasks.

2.3. Transformer

The Transformer uses an encoder-decoder architecture [31], widely applied in Natural Language Processing (NLP) tasks. It is known for its 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 GPT-3 [32], 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, presenting data processing challenges. 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 [8,9].

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 [33]. Variational inference is a commonly

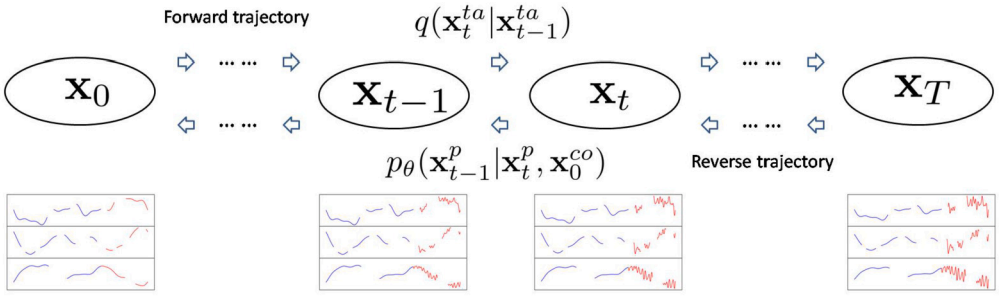


Fig. 1. Diagram of the diffusion processes in our forecasting model. The blue curves indicate the history events as the conditional data $\mathbf{x}_0^{co}$. The red curves symbolize the noisy target $\mathbf{x}_t^{ta}$ at time step t in the forward trajectory or the intermediate result $\mathbf{x}_t^p$ 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
$\mathbf{x}_0^{ta}$	Ground truth of target without noise
$\mathbf{x}_t^{ta}$	Noisy target during forward trajectory 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

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 [34]. 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.

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 Fig. 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 ground truth data $\mathbf{x}_0^{ta}$ at each step t to obtain the noisy target $\mathbf{x}_t^{ta}$. At the final step T, $\mathbf{x}_T^{ta} \sim N(\mathbf{0}, \mathbf{I})$. 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 scalar vector. 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, the process progressively denoises $\mathbf{x}_t^p$ until $\mathbf{x}_0^p$ is obtained. The final sampled $\mathbf{x}_0^p$ is drawn from a distribution designed to resemble the training data distribution. This process can also be represented as a Markov chain:

$$p_\theta(\mathbf{x}_{0:T}^p|\mathbf{x}_0^{co}) := p(\mathbf{x}_T^p)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}^p|\mathbf{x}_t^p, \mathbf{x}_0^{co})$$

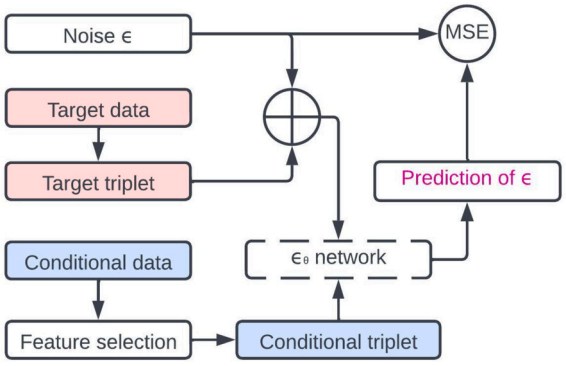


Fig. 2. The training procedure of our forecasting model minimizes the Mean Squared Error (MSE) between the noise prediction $\epsilon_\theta(x_t^{ta}, t|x_0^{co})$ and ϵ as the loss function. The implementation of ϵ_θ in the dashed box will be expanded and explained in detail later.

$$p_\theta(\mathbf{x}_{t-1}^p|\mathbf{x}_t^p, \mathbf{x}_0^{co}) := N(\mathbf{x}_{t-1}^p; \mu_\theta(\mathbf{x}_t^p, t|\mathbf{x}_0^{co}), \Sigma_\theta) \tag{2}$$

where μ_θ and Σ_θ are learnable functions that generate the mean and variance of the modeled distribution. Equation (1) implies that $\mathbf{x}_t^{ta} = \sqrt{\hat{\alpha}_t}\mathbf{x}_{t-1}^{ta} + \sqrt{1-\hat{\alpha}_t}\epsilon = \sqrt{\alpha_t}\mathbf{x}_0^{ta} + \sqrt{1-\alpha_t}\epsilon$, 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 $\mathbf{x}_0^{ta}$. As a result, it is advantageous to construct μ_θ and Σ_θ as follows:

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

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

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 [35].

3.2. Model structure

The objective of the model training is to maximize the Evidence Lower Bound of $p_\theta(\mathbf{x}_0^p|\mathbf{x}_0^{co})$ [24]. It can be expressed as the following equation:

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

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

Traditional methods, such as aggregation and imputation, are ineffective when dealing with extremely sparse data. Aggregation results in poor resolution and loss of temporal information, while imputa-

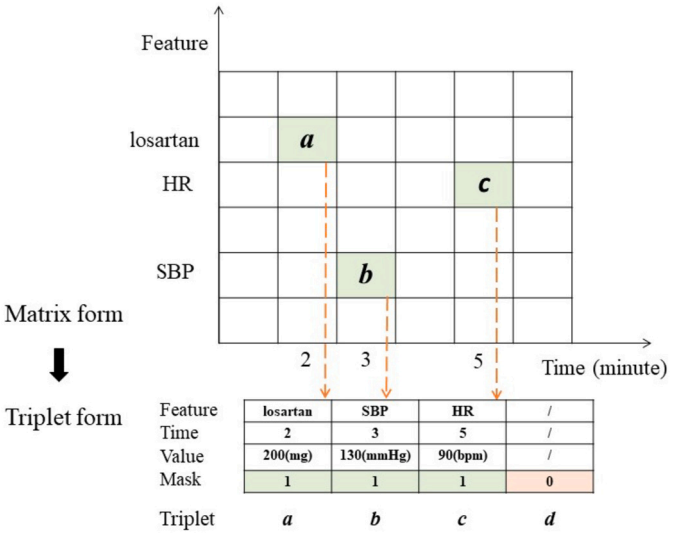


Fig. 3. An illustration of converting a sparse matrix to triplet representation. In this example, a patient received 200 milligram (mg) of losartan at the second minute. A Heart Rate (HR) of 90 beats per minute (bpm) and a Systolic Blood Pressure (SBP) of 130 millimeters of mercury (mmHg) are recorded in the third and fifth minute, respectively. Assuming all other data points in the matrix are missing, and the 3 valid event records ***a***, ***b***, and ***c*** line up in an array of triplets. The size of the input triplet arrays is preset, and it can be larger than the number of valid triplets. The mask value of 0 in ***d*** signifies the invalidity of this triplet, meaning the invalidity of its other 3 elements.

tion 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. Fig. 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 according to data preprocessing), the feature selection module prioritizes data points of the same features as the target, most correlated to the target data [14,15], 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 Fig. 4. The front stage maps the data into higher-dimensional spaces. A triplet's feature, value, and time components are 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 standardization). A random diffusion step is applied in each iteration to generate noisy target values. The diffusion step is represented as a vector obtained from a lookup table and projected through linear layers. The results of all these vectors are fed into the back stage.

Considering the trait similarity between the NLP data and the triplet data within our context, both originating from expansive and diverse spaces, we employ Transformer to build the back stage. This decision aligns with the prevailing discussions advocating for the attention mechanism to its efficacy in handling complex data representations [31,32,36]. The conditional triplet is transformed by the encoders 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 Q and K, and the scaling factor $\sqrt{d_k}$ enhances the model's robustness. K and V output from the top encoder layer are input into the decoder's encoder-decoder attention layers, together with the target Q, to obtain the cross attention. 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. The ϵ_θ architecture is characterized by its use of 3 Transformer blocks, each consisting of 2 stacked encoder-decoder Transformer that sequentially process the input data. This encoder–decoder pattern can refine the output of the first encoder-decoder Transformer, potentially correcting errors and adding complexity to the representation. The first block is connected through a skip connection that extends beyond the subsequent Transformer block, directly interfacing with the convolutional (Conv) decoder. The second block also utilizes a skip connection, contributing its processed features directly to the Conv decoder. The third block follows a direct sequential path, transmitting its output to the Conv decoder. The skip 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 [29,37]. The training process is outlined in Algorithm 1.

Algorithm 1 Training.

Input: $\mathbf{x}_0^{ta} \sim q(\mathbf{x}_0^{ta}|\mathbf{x}_0^{co})$, $\mathbf{x}_0^{co}$
Output: ϵ_θ 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

3.3. Model inference

The values of the predicted triplets in the model are initially Gaussian noise corresponding to the first diffusion step of the reverse trajectory (step T in Fig. 1). These noisy values are 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 triplets to predict provide important contextual information. The denoised values are then fed back into the model to generate a prediction of ϵ for the next diffusion 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 [38]. This dataset holds health information for over 46 thousand patients admitted to the Beth Israel Deaconess Medical Center (Boston,

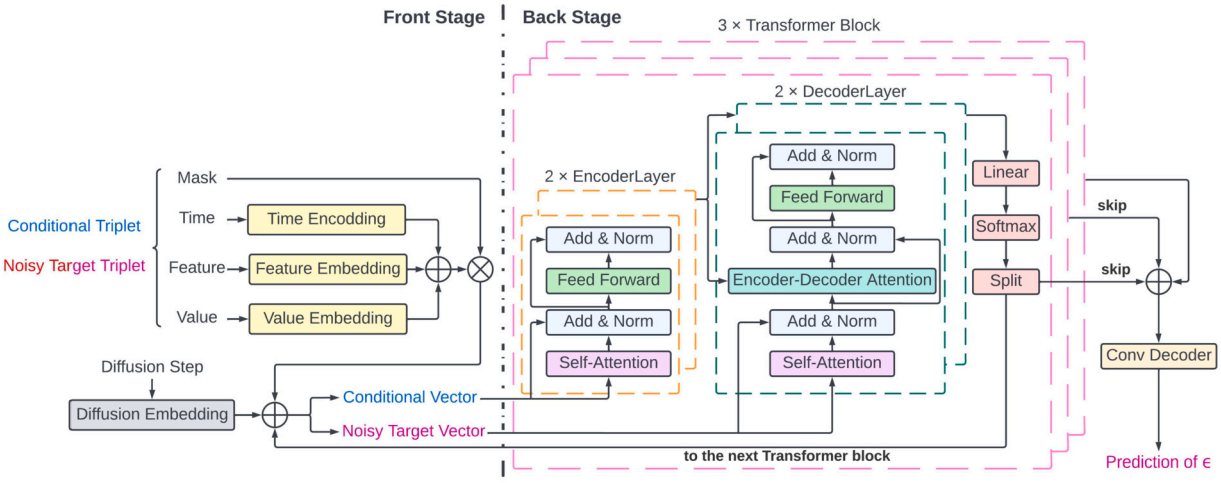


Fig. 4. The architecture of the ϵ_θ network, designed for predicting ϵ at the current step.

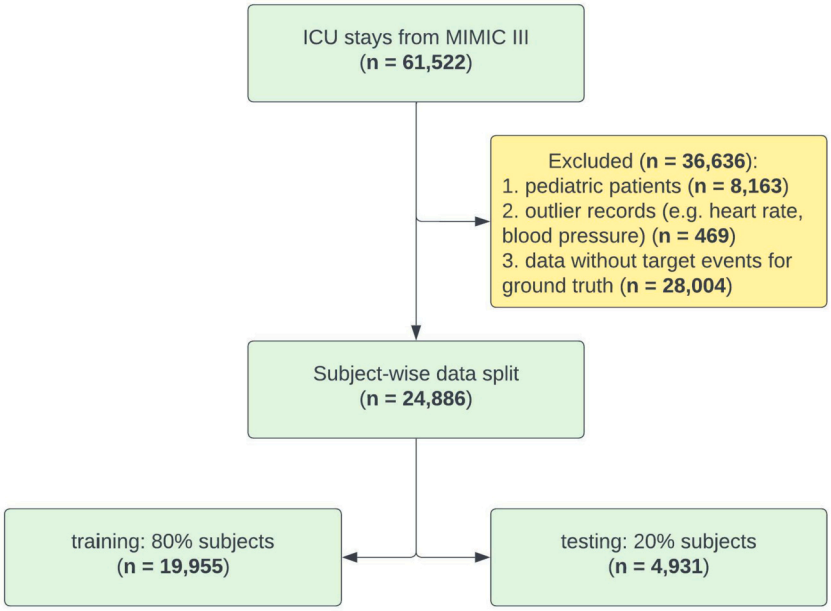


Fig. 5. Illustration of the data preprocessing steps involved in this study. n denotes the number of ICU stays.

MA) between 2001 and 2012. The data preprocessing approach is depicted in Fig. 5. The 3 steps in the yellow box are detailed as follows. First, we exclude records related to pediatric patients by filtering out individuals whose ages are greater than or equal to 18 years old. Second, we remove records with abnormal feature values. The features represent the medical events that happen to the patients, such as vital signs, medication usage, and biochemical test results. We eliminate outliers by retaining values within reasonable ranges for 117 numerical features, while non-null values are retained for other features categorized as yes-or-no items. Ultimately, we solely preserve the initial 40 consecutive minutes from each ICU stay, denoting this piece of data as an individual ICU sample. 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. We exclude ICU samples where the latter 10-minute segment lacks any target data. We standardize the feature values to ensure that it is suitable for input into the TDSTF model. We partially adopt the excluding method from [9] to steps 1 and 2. To prevent subject repetition, we divide the dataset into training and testing sets, randomly selecting 80% and 20% of the data by subject.

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.

After implementing the 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 standardization. Subjects in the

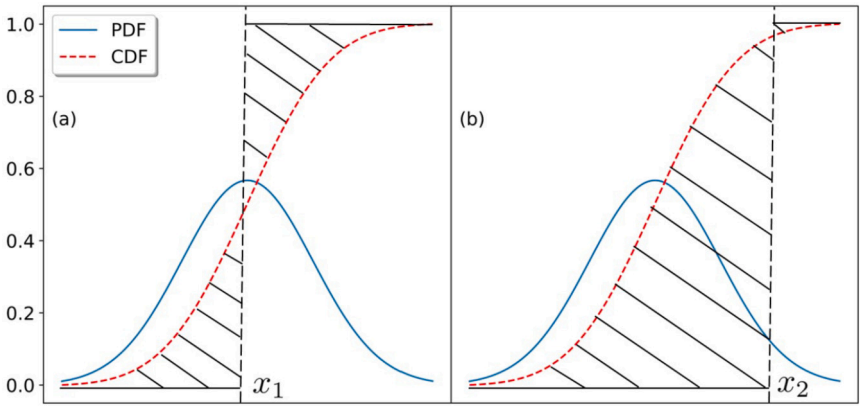


Fig. 6. A schematic illustration of the Continuous Ranked Probability Score (CRPS). The blue curves represent the modeled probability distribution functions, and the red dashed curves represent their cumulative distribution functions. The subplots (a) and (b) depict the same distribution with different 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.

training and testing datasets combined were generally older (age of 65.42 ± 16.27) and male (57.2%). The training data is further split into an $80-20\%$ ratio, with the latter portion used for validation. Note that one subject corresponds to one or more ICU stays. The goal of this study is to predict three target features: HR, SBP, and DBP [39].

4.2. Metrics

To thoroughly assess the performance of the proposed method, 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 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. A 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. Fig. 6 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 Standardized Average CRPS (SACRPS), when predicting a standardized target value x^{ta}, we adopt the same discrete quantile loss as in [17]:

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

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

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. SACRPS calculated as the CRPS divided by this scaling factor provides a more accurate evaluation of the model.

4.3. Experiment setup

We use the ADAM optimizer [40] with a learning rate 10^{-3}. The mini-batch size is set to 32 samples. We implemented all the experiments using the Python programming language and the Pytorch framework [41]. 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. Comparison with baseline

Five models were employed as the baseline models for predicting vital signs with the MIMIC-III dataset. MQ-RNN encompasses a singular layer of bidirectional GRU, featuring a hidden size of 50. The encoder CNN employs kernel sizes of $[7, 3, 3]$ alongside channel counts of $[30, 30, 30]$. Each layer within the Forking MLP Decoder is dimensioned at 30. The lookback period is tailored to cover the entire input time length. Categorical feature embeddings assume a dimensionality of 50. DeepAR comprises 2 layers of LSTM, each with 40 cells. The Dimension of the embeddings for categorical features is set as 50. The context length is adjusted to encompass the prediction horizon. For evaluation and sampling predictions, the student-t distribution is employed. DeepFactor applies both global and local models instantiated as LSTM. The count of global factors stands at 10, with each global model housing a single hidden layer of 50 units. The local model comprises a hidden layer containing 5 units. The observation model is specified as student-t. EnCQR contains an ensemble of 5 member learners, each configured as a 5-layer LSTM with a hidden size of 128 and an $L2$ regularization factor of 10^{-4}. Conformalized prediction intervals are utilized as the output. CSDI involves 50 diffusion steps, with a noise schedule following the quadratic schedule from $\beta_1 = 10^{-4}$ to $\beta_{50} = 0.5$ with a quadratic diffusion schedule [17]. Feature, time, and diffusion embedding dimensions are established at 16, 128, and 128, respectively. To address the missing data, we adopted a mean imputation of 0 for MQ-RNN, DeepAR, DeepFactor, and EnCQR, whereas CSDI utilized a mask matrix to denote the positions of missing data. In our model, the input size of the conditional triplet array is set to 60 according to Fig. 7.

The experimental results are presented in Table 3, with 3 trials conducted. Both the SACRPS 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 SACRPS and MSE, respectively. Thus, TDSTF improved SACRPS and MSE by 18.9% and 34.3% over CSDI. Fig. 8 shows the violin histograms of SACRPS and MSE per ICU sample for

Table 3
SACRPS and MSE of the proposed model in comparison with the baselines. The best results are shown bold.

	MQ-RNN	DeepAR	DeepFactor	EnCQR	CSDI	TDSTF (Ours)
SACRPS	1.0276	0.9930	0.8221	0.8255	0.5515	**0.4379**
	1.0367	0.9615	0.8207	0.8256	0.5455	**0.4526**
	1.0385	0.9653	0.8238	0.8254	0.5439	**0.4408**
MSE	1.1014	1.0295	1.0325	1.2711	0.6430	**0.4061**
	1.1237	1.0304	1.0332	1.2710	0.6358	**0.4434**
	1.1186	1.0289	1.0324	1.2715	0.6236	**0.4008**

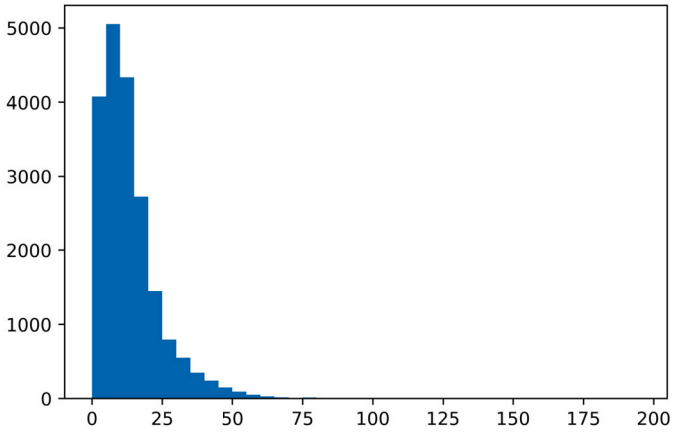


Fig. 7. Histogram showing the distribution of valid conditional data points derived from ICU samples within the training dataset, exhibiting a mean count of 12.9. On average, 99.7% of the conditional data within the matrix representation exhibit missing values. More than 99.5% of the ICU samples from the training dataset contain less than or equal to 60 valid conditional data points.

the first trial of TDSTF. The results concentrate on the median values 0.4651 and 0.1677 of SACRPS and MSE, which shows robustness of the model.

Fig. 9 shows some forecasting examples from the first trial of TDSTF. All the deterministic predictions 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 comparison 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 show 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 on the CSDI overhead due to its large amount of missingness input.

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

Table 4
SACRPS and MSE of the 3 ablation configurations, each with $\beta_1 = 10^{-4}$ and $\beta_T = 0.5$. The best results are shown in bold. TDSTF-c is the selected configuration.

Configuration	TDSTF-a	TDSTF-b	TDSTF-c
SACRPS	0.4843	**0.4328**	0.4379
	0.6072	0.4720	**0.4526**
	0.5162	0.4442	**0.4408**
MSE	0.7861	**0.3188**	0.4061
	1.2609	**0.4247**	0.4434
	0.6341	**0.3506**	0.4008

Table 5
SACRPS and MSE of TDSTF-c with three β_T values. The best results are shown in bold. $\beta_T = 0.5$ is the selected configuration.

β_T	0.05	0.1	0.5
SACRPS	0.5898	0.4893	**0.4379**
	0.5884	0.4721	**0.4526**
	0.6002	0.5023	**0.4408**
MSE	0.5247	0.4613	**0.4061**
	0.5629	0.4443	**0.4434**
	0.5777	0.4847	**0.4008**

4.5. Ablation study

To assess the impact of Transformer block structures on the backbone's architecture and the diffusion model's performance, we conducted an ablation study. This study focused on 3 configurations, as delineated in Fig. 10. With a series of 3 trials, the outcomes are shown in Table 4. TDSTF-a and TDSTF-c require the same training duration of 0.5 hours. TDSTF-c demonstrates better performance and robustness compared to TDSTF-a. Although TDSTF-b achieves lower MSE than TDSTF-c, it shows higher SACRPS and demands a longer training time of 0.7 hours. However, TDSTF-c manages to maintain comparable overall performance with a shorter training duration. As a result, TDSTF-c achieves an optimal balance between performance and model efficiency.

Noise amount plays a critical role in diffusion models, thus prompting our investigation of the β schedule through additional 2 sets of experiments, each consisting of 3 trials, built upon TDSTF-c. In both scenarios, β_1 remains fixed at 10^{-4}, and T is still set to 50. β_T assumes values of 0.05 and 0.1 respectively. The results in Table 5 indicate that the schedule spanning from $\beta_1 = 10^{-4}$ to $\beta_T = 0.5$ yields the most optimal performance.

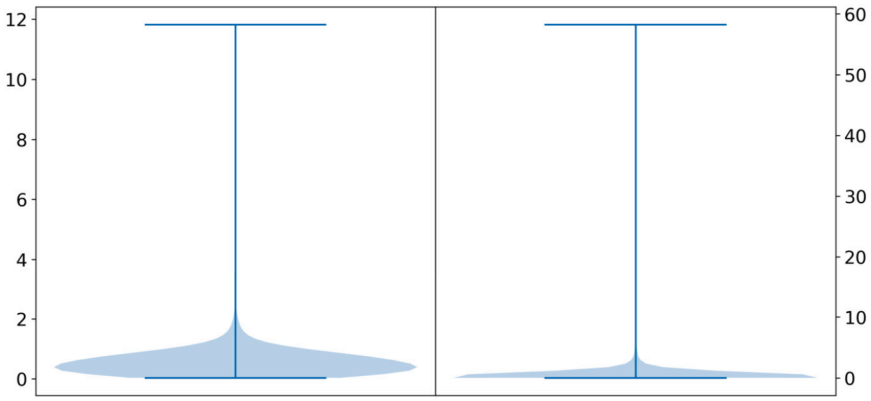
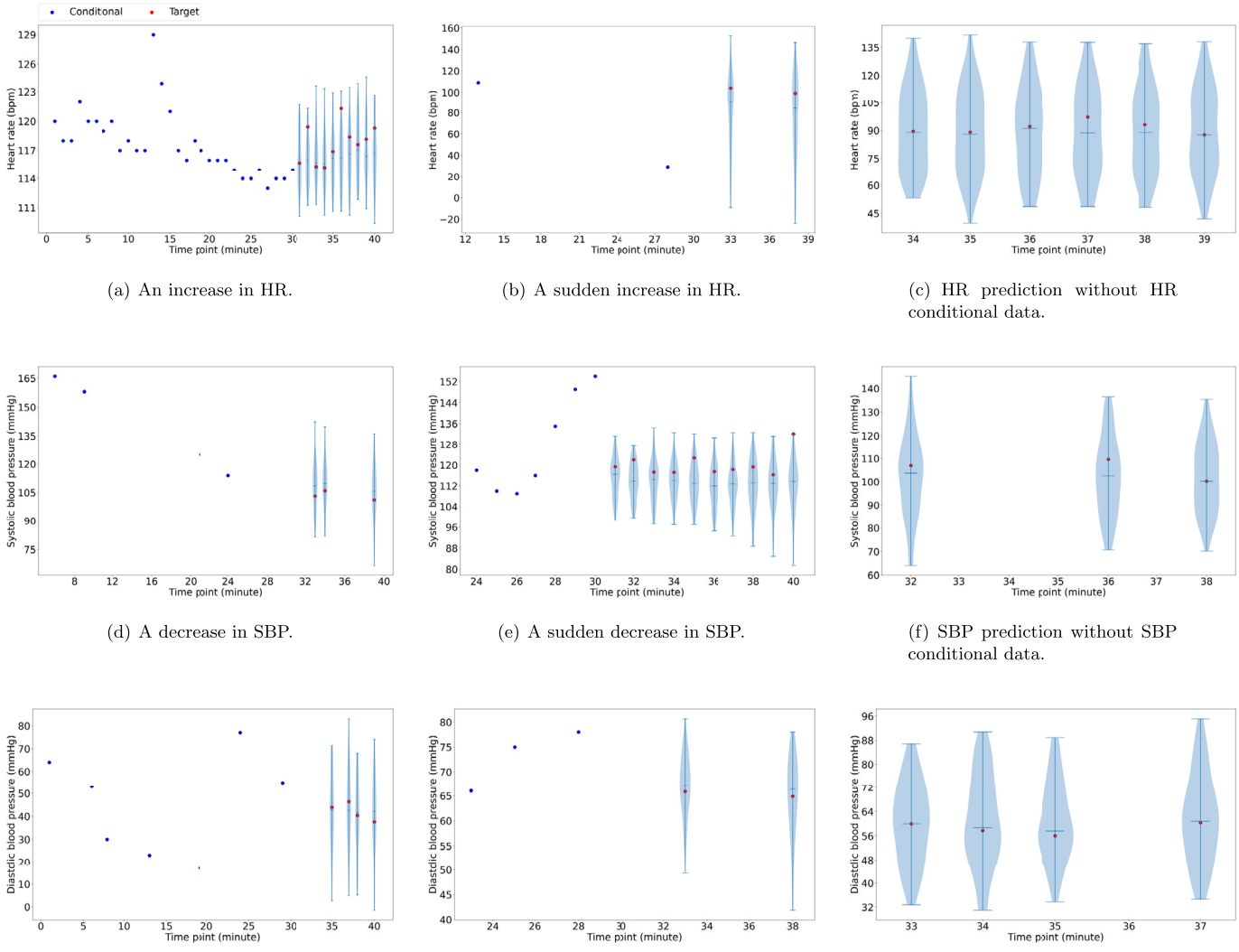


Fig. 8. Violin histograms of the SACRPS in the left and the MSE in the right per ICU sample from the first trial of TDSTF. The results for each metric concentrate around the median values of 0.4651 and 0.1677.



(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.

Fig. 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.

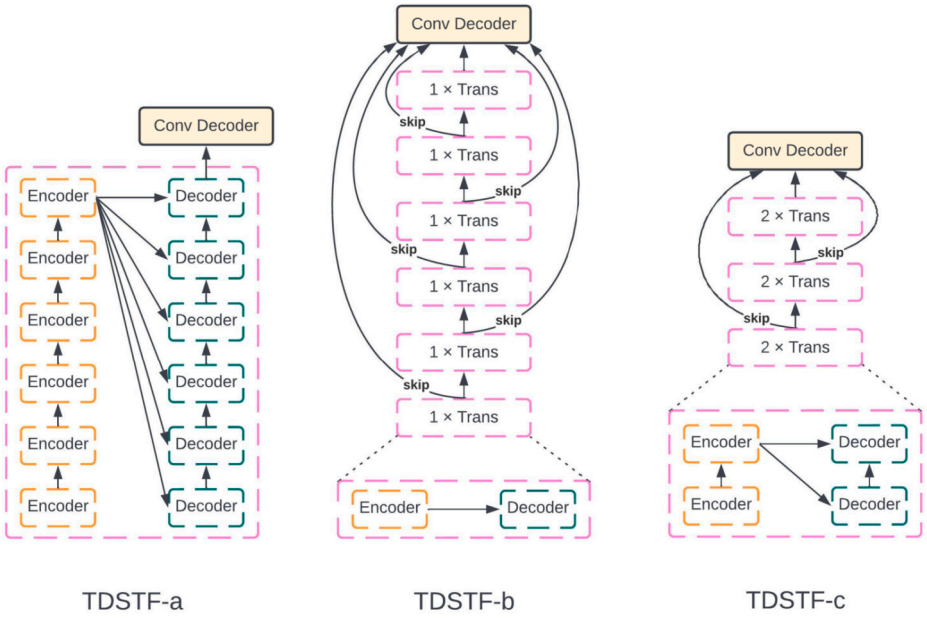


Fig. 10. The backstage structures for the ablation study.

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 show 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 Transformers, which allows for the ability to capture complex temporal dependencies and efficient computation and storage.

The TDSTF model is 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 SACRPS and MSE. It outperforms the baseline models, including MQ-RNN, DeepAR, DeepFactor, EnCQR, and CSDI. The mask used in CSDI reduces disturbance but cannot guarantee enough exclusion of invalid information from the sparse time series input. On average, 99.7% of conditional data points were missing, 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, DeepFactor, and EnCQR use RNN to extract hidden states and cannot fully utilize parallel computation, hindering model complexity [42]. The Transformer-based ϵ_θ network solves these issues. While DeepAR and DeepFactor assume likelihood that may be too simple to represent high-dimensional distributions. TDSTF, on the other hand, directly learns the underlying distributions.

TDSTF is effective because it accurately captures temporal patterns in sparse time series data, as shown in Fig. 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 increase quadratically with its size [43]. 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.

We look forward to this work serving as a foundation for addressing practical clinical applications in the ICU. 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 TDSTF to include core body temperature, breathing rate, and oxygen saturation as additional forecasting outputs. Additionally, we may explore the application of the TDSTF in reducing false alarms, with the aim of alleviating the stress experienced by ICU caregivers.

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 SACRPS 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.

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.

CRediT authorship contribution statement

Ping Chang: Conceptualization, Data curation, Formal analysis, Methodology, Visualization, Writing – original draft, Writing – review & editing, Investigation, Software, Validation. **Huayu Li:** Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Stuart F. Quan:** Funding acquisition, Writing – review & editing. **Shuyang Lu:** Writing – review & editing. **Shu-Fen Wung:** Writing – review & editing. **Janet Roveda:** Funding acquisition, Resources, Writing – review & editing. **Ao Li:** Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing 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

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.

References

[1] Tsuneaki Kenzaka, Masanobu Okayama, Shigehiro Kuroki, Miho Fukui, Shinsuke Yahata, Hiroki Hayashi, Akihito Kitao, Daisuke Sugiyama, Eiji Kajii, 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, Intern. Med. 51 (8) (2012) 871–876.

[2] Joo Heung Yoon, Lidan Mu, Lujie Chen, Artur Dubrawski, Marilyn Hravnak, Michael R. Pinsky, Gilles Clermont, Predicting tachycardia as a surrogate for instability in the intensive care unit, J. Clin. Monit. Comput. 33 (2019) 973–985.

[3] Christian P. Subbe, Michael Kruger, Peter Rutherford, L. Gemmel, Validation of a modified early warning score in medical admissions, Q. J. Med. 94 (10) (2001) 521–526.

[4] Daniel I. Sessler, Bernd Saugel, Beyond 'failure to rescue': the time has come for continuous ward monitoring, Br. J. Anaesth. 122 (3) (2019) 304–306.

[5] Alexa K. Doig, Frank A. Drews, Maureen R. Keefe, Informing the design of hemodynamic monitoring displays, CIN, Comput. Inform. Nurs. 29 (12) (2011) 706–713.

[6] Sarah A. Collins, Lena Mamykina, Desmond Jordan, Dan M. Stein, Alisabeth Shine, Paul Reyfman, David Kaufman, In search of common ground in handoff documentation in an intensive care unit, J. Biomed. Inform. 45 (2) (2012) 307–315.

[7] Ævar Örn Kristinsson, Ying Gu, Søren M. Rasmussen, Jesper Mølgaard, Camilla Haahr-Raunkjær, Christian S. Meyhoff, Eske K. Aasvang, Helge B.D. Sørensen, Prediction of serious outcomes based on continuous vital sign monitoring of high-risk patients, Comput. Biol. Med. 147 (2022) 105559.

[8] Marzyeh Ghassemi, Marco Pimentel, Tristan Naumann, Thomas Brennan, David Clifton, Peter Szolovits, Mengling Feng, A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, 2015.

[9] Sindhu Tipirneni, Chandan K. Reddy, Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series, ACM Trans. Knowl. Discov. Data (TKDD) 16 (6) (2022) 1–17.

[10] Inigo Jauregi Unanue, Ehsan Zare Borzeshi, Massimo Piccardi, Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition, J. Biomed. Inform. 76 (2017) 102–109.

[11] H. Ij, Statistics versus machine learning, Nat. Methods 15 (4) (2018) 233.

[12] Shiyu Liu, Jia Yao, Mehul Motani, Early prediction of vital signs using generative boosting via LSTM networks, in: 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2019, pp. 437–444.

[13] Shamsul Masum, John P. Chiverton, Ying Liu, 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, Proceedings, AI 2019, Cambridge, UK, December 17–19, 2019, vol. 39, Springer, 2019, pp. 269–282.

[14] 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 Med. Inform. 9 (4) (2021) e18803.

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

[16] Kashif Rasul, Calvin Seward, Ingmar Schuster, Roland Vollgraf, Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, in: International Conference on Machine Learning, PMLR, 2021, pp. 8857–8868.

[17] Yusuke Tashiro, Jiaming Song, Yang Song, Stefano Ermon, Csdi: conditional score-based diffusion models for probabilistic time series imputation, Adv. Neural Inf. Process. Syst. 34 (2021) 24804–24816.

[18] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, Surya Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.

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

[20] David Salinas, Valentin Flunkert, Jan Gasthaus, Tim Januschowski Deepar, Probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 36 (3) (2020) 1181–1191.

[21] Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, Tim Januschowski, Deep factors for forecasting, in: International Conference on Machine Learning, PMLR, 2019, pp. 6607–6617.

[22] Vilde Jensen, Filippo Maria Bianchi, Stian Normann Anfinsen, Ensemble conformalized quantile regression for probabilistic time series forecasting, IEEE Trans. Neural Netw. Learn. Syst. (2022).

[23] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, Ming-Hsuan Yang, Diffusion models: a comprehensive survey of methods and applications, ACM Comput. Surv. 56 (4) (2023) 1–39.

[24] Jonathan Ho, Ajay Jain, Pieter Abbeel, Denoising diffusion probabilistic models, Adv. Neural Inf. Process. Syst. 33 (2020) 6840–6851.

[25] Jiaming Song, Chenlin Meng, Stefano Ermon, Denoising diffusion implicit models, arXiv preprint, arXiv:2010.02502, 2020.

[26] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, Rianne van den Berg, Structured denoising diffusion models in discrete state-spaces, Adv. Neural Inf. Process. Syst. 34 (2021) 17981–17993.

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

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

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

[30] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, Bryan Catanzaro, Diffwave: a versatile diffusion model for audio synthesis, arXiv preprint, arXiv:2009.09761, 2020.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[32] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[33] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, Stephan Mandt, Advances in variational inference, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2018) 2008–2026.

[34] William Feller, On the theory of stochastic processes, with particular reference to applications, in: Selected Papers I, Springer, 2015, pp. 769–798.

[35] Calvin Luo, Understanding diffusion models: a unified perspective, arXiv preprint, arXiv:2208.11970, 2022.

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

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[38] Alistair Johnson, Tom Pollard, Roger Mark, MIMIC-III clinical database (version 1.4), PhysioNet 10 (C2XW26) (2016) 2.

[39] Craig Lockwood, Tiffany Conroy-Hiller, Tamara Page, Vital signs, JBI Evid. Synth. 2 (6) (2004) 1–38.

[40] Diederik P. Kingma, Jimmy Ba, Adam: a method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.

[41] 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, Adv. Neural Inf. Process. Syst. 32 (2019).

[42] Wim De Mulder, Steven Bethard, Marie-Francine Moens, A survey on the application of recurrent neural networks to statistical language modeling, Comput. Speech Lang. 30 (1) (2015) 61–98.

[43] Ming Ding, Chang Zhou, Hongxia Yang, Jie Tang Cogltx, Applying bert to long texts, Adv. Neural Inf. Process. Syst. 33 (2020) 12792–12804.