

Time-Series Forecasting in Healthcare: A Dual-Framework Approach with Latent Class Integration

1st Ruiying Chen
Maths Department
University College London
 London, United Kingdom
 zcahrc3@ucl.ac.uk

2nd Qingang Tang
Statistics Department
University of Oxford
 Oxford, United Kingdom
 corp3481@ox.ac.uk

Abstract—This study addresses the challenge of forecasting clinical time series data, leveraging the MIMIC-IV dataset to improve predictive accuracy through novel latent class integration techniques. We introduce two enhanced variational recurrent models, CVLSTM and LTVLSTM, that combine disease classification with time-series modeling to capture both local patterns and long-term temporal dependencies. CVLSTM incorporates latent class variables to classify diseases and improve prediction accuracy, while LTVLSTM extends this framework by learning local trends in subsequences and integrating them into the broader temporal structure of the data. Using Root Mean Square Error (RMSE) as the primary evaluation metric, our methods significantly outperform traditional ARIMA, LSTM, and VRNN models, with LTVLSTM achieving the highest accuracy on the MIMIC-IV dataset. These results demonstrate the potential of integrating disease-based priors and hierarchical variational inference for enhancing clinical time series forecasting.

Index Terms—Time series forecasting, MIMIC IV, VRNN, CVLSTM; LTVLSTM

I. INTRODUCTION

An estimated 20-40% of global health system spending is wasted due to inefficiencies [1]. Issues such as poor quality control, ineffective hospital management and unnecessary care place significant burdens on patients and the government [2]. Thus, there is substantial space to improve healthcare expenditure and quality of care [3], while the blackness of advanced technology have been considered as one of the fundamental issues.

By 2015, more than 80% of hospitals have implemented at least a basic Electronic Health Record (EHR) system [4]. The widespread adoption of EHR has led to significant improvements [5][6]. Most EHRs collect both continuous and categorical data from patients during their Intensive Care Unit (ICU) stay, enabling comprehensive analysis and monitoring. The multitude of data available in the EHRs make them suitable for high-dimensional analyses [7], including phenotype classification, risk of mortality prediction, length of stay prediction, etc.

Early time series forecasting primarily used methods such as autoregressive models (AR), moving average models (MA), and autoregressive integrated moving average

models (ARIMA) [8]. However, the model was assumed to have a linear relationship with observed values which led to unsatisfactory results for complex real world prediction. In recent years, research on clinical time series prediction has used deep neural networks to tackle these problems. Convolutional neural networks (CNN) [9] and recurrent neural networks (RNN) [10] were two types of DNN widely adopted for time series prediction. CNN could capture local patterns of short time series while RNN learns long term temporal dependencies of the entire time series. In fact, many research proposed using the combination of both. [11] stacks CNN with RNN for DNA sequence prediction that captured short and recurring sequence motifs and the spatial arrangement of these motifs at the same time. However, the problem with [11] was it could not capture temporal information with high accuracy since they were so sensitive to perturbations. This was extremely problematic especially for clinical data due to its high variance and missing data. While generative model became more popular, researchers started to use variational autoencoder (VAE) [12] in time series prediction. VAE encoded the data into latent space and captured the temporal information of time series more accurately [13]. [14] used this idea and introduced a RNN as the backbone to capture the long-term temporal dynamics of time series. To the best of our knowledge, existing models learnt either local trend patterns or temporal dynamics. This paper introduce two methods to overcome this issue.

In this paper, our contribution could be summarized as follows.

- 1) We first added a latent categorical variable on top of the VRNN model based on [14] that identifies the disease class. Having knowledge the diseases of patient allowed the model to recognize the seasonal trend of the time series object. We called this the Classifying Variational Long Short-term Memory (CVLSTM) Model.
- 2) Then we introduced another method which derived an objective following variational inference to integrate the learning of local patterns and temporal dynamics of time series.

The structure of the paper is as follows: we present our first method, CVLSTM, in Section II-A; then we describe how local information is extracted and combined with temporal information for clinical time series forecasting in Section II-B. The MIMIC-IV dataset and the data preprocessing steps are outlined in Sections III-A and III-B. Subsequently, we detail the implementation of our main method as well as the baseline methods in Section III-C. Finally, we present and analyze the experimental results in Section IV.

II. METHODS

This section first presents an approach called CVLSTM, which integrates a LSTM networks into both the encoder and decoder networks of a VAE model to enhance its sequential modeling capabilities [15][16][17]. This approach adds an additional continuous latent variable w , to the model to represent (classify) the probability of being in a class.

A. VRNN with latent class (CVLSTM)

1) *Introducing Classifying VAE*: Firstly, we describe how class variables were embedded into the VAE model. Patients with similar health statuses were grouped by summarizing disease diagnoses in the MIMIC-IV dataset into five general classes. This simplification was implemented to reduce computational complexity and improve model generalizability. These five classes were represented by an additional continuous latent variable, w , which follows a multinomial distribution parameterized by θ . The joint distribution of the observed data X and the latent variables z and w is expressed as:

$$p_\theta(X, z, w) = p_\theta(X|z, w)p_\theta(z)p_\theta(w) \quad (1)$$

where X is the observed data, z and w are latent variables. We also assumed that z and w are independent variables.

During training, we assumed both X and w were given; while during inference, only X was given. This was due to the VAE's limitation in inferring discrete latent variables. Instead, our approach modelled them as class probabilities rather than treating them as discrete categorical variables.

Then, we would like to learn the entire posterior of class probabilities: $p_\theta(w|X)$. Following the derivation from [18], which built on the modification of [12]: we constructed a variational lower-bound on the log of the marginal likelihood

$$p_\theta(X) = \int p_\theta(X|z, w)p_\theta(z)p_\theta(w) \partial w \partial z \quad (2)$$

with

$$\begin{aligned} & \log p_\theta(X) - \mathcal{D}[q_\phi(z, w | X) \parallel p_\theta(z, w | X)] \\ &= \mathbb{E}_{(z, w) \sim q_\phi(z, w | X)} [\log p_\theta(X | z, w)] \\ & - \mathcal{D}[q_\phi(z, w | X) \parallel p_\theta(z, w)] \end{aligned} \quad (3)$$

Then follow [19], let us rewrite the right-hand side as:

$$\begin{aligned} & \mathbb{E}_{(z, w) \sim q_\phi(z, w | X)} [\log p_\theta(X | z, w)] \\ & - \mathbb{E}_{w \sim q_\phi(w | X)} [\mathcal{D}[q_\phi(z | X, w) \parallel p_\theta(z)]] \\ & - \mathcal{D}[q_\phi(w | X) \parallel p_\theta(w)] = -\mathcal{L}_{VAE}(X) \end{aligned} \quad (4)$$

where $p_\theta(z)$ and $p_\theta(w)$ are priors, and $q_\phi(w|X)$ and $q_\phi(z|w, X)$ are variational approximations to the true posteriors. Same as [12], we aim to optimize the marginal likelihood, which can be achieved via maximizing (4) using stochastic gradient descent on θ and ϕ .

Furthermore, we need to ensure that $\mathcal{D}[q_\phi(z, w | X) \parallel p_\theta(z, w | X)]$ is small as possible. Even we don't have access to the true posterior $p_\theta(z|X, w)$, we assume that the true class is denoted as $\tilde{w}$. This setup allows us to minimize the categorical cross-entropy loss between $q_\phi(w|X)$ and $\tilde{w}$ [18], encouraging $q_\phi(w|X)$ to classify X . We end up with the final objective, denoted as $(\mathcal{L})$, is therefore as follows:

$$\mathcal{L}(X) = \mathcal{L}_{VAE}(X) + \alpha \mathbb{E}_{w \sim q_\phi(w|X)} [\mathcal{L}_c(w; \tilde{w})] \quad (5)$$

$\mathcal{L}_c(w; \tilde{w})$ represents the categorical cross-entropy loss between a sampled $w \sim q_\phi(w|X)$ and the true class $\tilde{w}$.

The core concept of our model is that samples from $q_\phi(w | X)$ are utilized in the recognition model for z , which also influence the classification loss. This approach aims to improve the classification of X , resulting in enhanced reconstruction of X and control over the class of generated samples. Similar to the standard VAE, we employ the Stochastic Gradient Variational Bayes (SGVB) estimator of the ELBO, where we draw samples $w \sim q_\phi(w | X)$ and $z \sim q_\phi(z | w, X)$ for each minibatch during training.

One limitation of the standard VAE is that the latent variables z and w cannot be discrete. To overcome this, we apply the "reparameterization trick" to generate z and w as differentiable, deterministic functions of X and auxiliary variables ϵ with independent marginals. Although $q_\phi(z | w, X)$ can be modeled as a standard multivariate Gaussian, $q_\phi(w | X)$ must represent the posterior probabilities that X belongs to each class. However, using the Dirichlet distribution for $q_\phi(w | X)$ is incompatible with the reparameterization trick.

Instead, we model $q_\phi(w | X)$ using a Logistic Normal distribution with mean and diagonal covariance generated by a multilayer perceptron conditioned on X . Specifically, $w \sim q_\phi(w | X) = \mathcal{LN}(\mu_\phi(X), \sigma_\phi^2(X))$, where w satisfies $0 \leq w_i \leq 1$ for $i = 1, \dots, 5$, and $\sum_{i=1}^5 w_i = 1$. This enables w to be interpreted as the estimated probabilities that X belongs to each of the 5 classes.

Importantly, samples from the Logistic Normal distribution can be deterministically generated from a Normal distribution with the same parameters. If $y \sim \mathcal{N}(\mu, \Sigma)$ is sampled from a multivariate normal distribution in $\mathbb{R}^4$, the corresponding $w \sim \mathcal{LN}(\mu, \Sigma)$ is computed as:

$$w_j = \frac{e^{y_j}}{1 + \sum_{k=1}^4 e^{y_k}}, \quad \text{for } j = 1, \dots, 4, \quad (6)$$

and

$$w_5 = \frac{1}{1 + \sum_{k=1}^4 e^{y_k}} \quad (7)$$

This approach allows us to sample $w \sim q_\phi(w | X)$ and $z \sim q_\phi(z | X, w)$ using the reparameterization trick, enabling efficient training of the Classifying VAE with SGVB.

2) *Implementation*: For the Classifying VAE, we assume the following priors on w and z_t for $t = 1, \dots, T$:

$$p_\theta(w) = \mathcal{N}(0, I) \quad p_\theta(z_t) = \mathcal{N}(0, I). \quad (8)$$

For our posterior approximations, we use:

$$q_\phi(w | X) = \mathcal{N}(\mu_{w,\phi}(X), \text{diag}[\sigma_{w,\phi}^2(X)]) \quad (9)$$

$$q_\phi(z_t | X_t, w) = \mathcal{N}(\mu_{z,\phi}(X_t, w), \text{diag}[\sigma_{z,\phi}^2(X_t, w)]). \quad (10)$$

where $\mu_{w,\phi}$ and $\sigma_{w,\phi}^2$ are implemented as the outputs of the classifier, and $\mu_{z,\phi}$ and $\sigma_{z,\phi}^2$ are implemented as the outputs of the encoder. During training we draw a single sample $w \sim q_\phi(w|X)$ as in (9) to compute $\mu_{z,\phi}$ and $\sigma_{z,\phi}^2$. Finally, we assume

$$p_\theta(X_t | w, z_t, X_{t-1}) = \mathcal{N}(\mu_\theta(w, z_t, X_{t-1}), \sigma_\theta^2(w, z_t, X_{t-1})) \quad (11)$$

where μ and σ are the output of the decoder given a single sample of w and z_t from (9) and (10), as well as the previous time step X_{t-1} . The output of the decoder network is split into two branches: one branch computes μ_θ directly, while the other computes $\log \sigma_\theta^2$ to ensure the variance remains positive.

3) *Modeling time series with a LSTM framework*: Since we are dealing with time series X_t , we autoencode each time step X_t by a classifying VAE independently. Extensions of VAEs to temporal data incorporate RNN such as LSTM networks into both the encoder and decoder networks of a VAE which allows the encoding and decoding networks to be conditioned on the sequence history at each time step.

B. local and temporal variational long short term memory (LTVLSTM)

Next we introduce the second method, LTVLSTM, which learns the local patterns from time series subsequences and the temporal dynamics among time series subsequences. In order to achieve this, we firstly encode the time series subsequence which learns captures local patterns; and the encoding of entire time series which learns temporal dynamics among time series subsequences.

1) *Encoding of time series subsequence*: Consider we have a time series subsequence $\mathbf{x}^t$, $t = 1, \dots, T$, and its corresponding latent random variable $\mathbf{z}^t$. Following the idea from [20], we train a generative model $p_\theta(x^t, z^t) = p_\theta(x^t|z^t)p_\theta(z)$. We divide latent variables z^t into L layers z_i^t , $i = 1, \dots, L$, and each layer has a sequential dependency as shown in figure 1.

Based on this, each stochastic layer is a fully factorised Gaussian distribution conditioned on the layer above:

$$\begin{aligned} p_\theta(z^t) &= p_\theta(z_L^t) \prod_{i=1}^{L-1} p_\theta(z_i^t | z_{i+1}^t) \\ p_\theta(z_i^t | z_{i+1}^t) &= \mathcal{N}(z_i^t | \mu_i^t(z_{i+1}^t), \sigma_i^t(z_{i+1}^t)), \\ p_\theta(z_L^t) &= \mathcal{N}(z_L^t | \mathbf{0}, \mathbf{I}) \\ p_\theta(X^t | z_1^t) &= \mathcal{N}(x^t | \mu_i^t(z_1^t), \sigma_i^t(z_1^t)) \end{aligned} \quad (12)$$

[20] distinguished the generative or inference distributions, but for simplicity, we adopt the same distribution among the latent

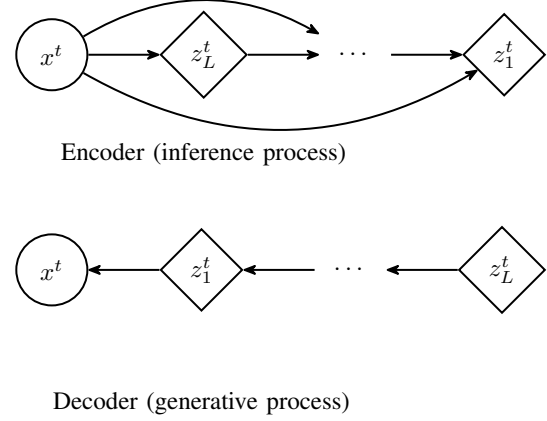


Fig. 1. The inference and generative process of subsequence X^t

random variables in the encoding and decoding processes. The generative model of the subsequence encoding is:

$$p(x^t, z^t) = p(x^t | z_1^t) p(z_L^t) \prod_{i=1}^{L-1} p(z_i^t | z_{i+1}^t) \quad (13)$$

Similar to the original VAE [12], the posterior is intractable and we will use $q(z^t|X^t)$ to estimate. Then following [20], we will obtain the inference model:

$$\begin{aligned} q(z^t | x^t) &= q(z_L^t | X^t) \prod_{i=1}^{L-1} q(z_i^t | z_{i+1}^t, x^t) \\ q(z_L^t | X^t) &= \mathcal{N}(z_L^t | \mu_L^t(X^t), \sigma_L^t(x^t)) \\ q(z_i^t | z_{i+1}^t, X^t) &= \mathcal{N}(z_i^t | \mu_i^t(z_{i+1}^t, x^t), \sigma_i^t(z_{i+1}^t, x^t)) \end{aligned} \quad (14)$$

2) *Encoding of entire time series*: Instead of creating another model to learn the entire time series, we can encode all time series subsequences $\{x^1, \dots, x^T\}$ as $\{z^1, \dots, z^T\}$ to learn the temporal dynamics of entire time series. Since each subsequence is unlikely to be independent, we impose a temporal dependency for consecutive subsequences $p(z^t, z^{t-1}) = p(z^t|z^{t-1})p(z^{t-1})$. We also use a LSTM [20] to capture long term temporal dependency with other subsequences.

$$\begin{aligned} p(z^t | z^{<t}) &= p(z^t | z^{t-1}, h^{t-1}) \\ p(z^t | z^{t-1}, h^{t-1}) &= \mathcal{N}(z^t | \mu_i^t(z^{t-1}, h^{t-1}), \sigma_i^t(z^{t-1}, h^{t-1})) \end{aligned} \quad (15)$$

where h is the hidden state and is obtained by $h^t = \text{LSTM}(h^{t-1}, x^t)$. For the generative process $p(x^{\leq T}, z^{\leq T})$, we simplify $p(x^t | z^{\leq t})$ to $p(x^t | z^t)$ to ensure that the local patterns of x^t are mainly captured in z^t . This simplification also reduces the complexity of the reconstruction and decoding processes. Based on (15), the generative model can be expressed as:

$$p(x^{\leq T}, z^{\leq T}) = \prod_{t=1}^T p(x^t | z^t) p(z^t | z^{<t}, z^{t-1}). \quad (16)$$

The term $p(x^t | z^t, z^{<t})$ can also be written as $p(x^t | z^t, h^{t-1})$, due to the recursive structure of the GRU, where h^{t-1} is recursively computed based on $x^{<t}$.

Similarly, the inference model is derived as:

$$\begin{aligned} q(z^t | x^{<t}, z^{t-1}) &= \mathcal{N}(z^t | \mu_i^t(x^{<t}, z^{t-1}), \sigma_i^t(x^{<t}, z^{t-1})) \\ &= \mathcal{N}(z^t | \mu_i^t(h^{t-1}, x^t, z^{t-1}), \sigma_i^t(h^{t-1}, x^t, z^{t-1})). \end{aligned} \quad (17)$$

This approximated posterior of z^t accounts for the long-term dependencies conveyed by $x^{<t}$ via h^{t-1} , the neighboring relationship with z^{t-1} , and the subsequence x^t itself.

3) *Integration and joint learning*: We would like to learn both time series subsequences and the entire time series $\{(z_L^1, \dots, z_1^1), \dots, (z_L^T, \dots, z_1^T)\}$. The combining prior of (12) and (15) can be written as:

$$\begin{aligned} p(z^t | z^{t-1}, h^{t-1}) &= p(z_L^t | z_L^{t-1}, h^{t-1}) \prod_{i=1}^L p(z_i^t | z_{i+1}^t) \\ p(z_L^t | z_L^{t-1}, h^{t-1}) &= \mathcal{N}(z_L^t | \mu_L^t(z_L^{t-1}, h^{t-1}), \sigma_L^t(z_L^{t-1}, h^{t-1})) \\ p(z_i^t | z_{i+1}^t) &= \mathcal{N}(z_i^t | \mu_i^t(z_{i+1}^t), \sigma_i^t(z_{i+1}^t)). \end{aligned} \quad (18)$$

and the inference model as follow:

$$\begin{aligned} q(z^t | x^{<t}, z^{t-1}) &= q(z_L^t | x^{<t}, z^{t-1}) \prod_{i=1}^{L-1} q(z_i^t | z_{i+1}^t, x^t) \\ q(z_L^t | x^{<t}, z^{t-1}) &= \mathcal{N}(z_L^t | \mu_L^t(z_L^{t-1}, x^{<t}), \sigma_L^t(z_L^{t-1}, x^{<t})) \\ q(z_i^t | z_{i+1}^t, x^t) &= \mathcal{N}(z_i^t | \mu_i^t(z_{i+1}^t, x^t), \sigma_i^t(z_{i+1}^t, x^t)). \end{aligned} \quad (19)$$

the generative model is then:

$$\begin{aligned} p(x^t | z_1^t, x^{<t}) &= p(x^t | z_1^t, h^{t-1}) \\ &= \mathcal{N}(x^t | \mu_i^t(z_1^t, h^{t-1}), \sigma_i^t(z_1^t, h^{t-1})) \end{aligned} \quad (20)$$

According to [20], our objective function (ELBO) will be as follow:

$$\begin{aligned} \mathcal{L}_{\text{enc}} &= \sum_{t=1}^T \left\{ \mathbb{E}_{q(z^t | x^{<t}, z_1^{t-1})} \log p(x^t | h^{t-1}, z_1^t) \right. \\ &\quad - \sum_{i=1}^{L-1} \text{KL}(q(z_i^t | z_{i+1}^t, x^t) \| p(z_i^t | z_{i+1}^t)) \\ &\quad \left. - \text{KL}(q(z_L^t | x^{<t}, z_1^{t-1}) \| p(z_L^t | x^{<t}, z_1^{t-1})) \right\}. \end{aligned} \quad (21)$$

The initial component of $\mathcal{L}_{\text{enc}}$ represents the reconstruction error for LTVLSTM, capturing the discrepancy between the input x^t and the reconstructed output x^t generated using z_1^t and h^{t-1} . The subsequent terms act as regularization components, ensuring that the latent random variables are encoded in a manner that effectively represents the local structures of individual subsequences while also modeling the temporal dependencies across the entire time series. The expected value of $\mathcal{L}_{\text{enc}}$ is computed using Monte Carlo sampling, and the final value is obtained as the average of $\mathcal{L}_{\text{enc}}$ across all time series samples.

For the final time series prediction, we utilize h^t and z^t , where the predicted output, $\hat{y}$, is computed as $\psi(h^t, z^t)$. Here, $\psi(\cdot)$ represents a single-layer fully connected neural network. The forecasting error is quantified by (23):

$$\mathcal{L}_{\text{pred}} = \text{Err}(y, \hat{y}) \quad (22)$$

$$\text{Err}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (23)$$

Then the overall objective function minimises the negative ELBO of LTVLSTM and the forecasting loss:

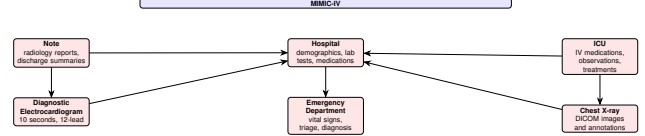
$$\mathcal{L} = -\mathcal{L}_{\text{enc}} + \mathcal{L}_{\text{pred}} \quad (24)$$

III. DATE AND MODEL

A. Data Description

The "Medical Information Mart for Intensive Care (MIMIC-IV)" [21] is a modern electronic health record dataset that spans a decade of patient admissions from 2008 to 2019. MIMIC-IV offers comprehensive data for retrospective clinical studies and critical care operations, featuring high granularity on aspects such as re-admissions, length of stay, prescriptions, caregivers, and diagnoses and procedures (ICD-9). As shown in Fig.2, MIMIC-IV separates patient information in the Emergency Department (ED) and Intensive Care Unit (ICU). We focus on exclusively on ICU data, supported by with demographic data, as ICU data contains longer time-series inputs and less missing data (further details are provided in the following subsections).

Fig. 2. structure of MIMIC-IV constructed in this paper



B. Data preprocessing

Gupta et al. [22] contains extensive data processing pipeline for MIMIC-II, yielding optimal result with MIMIC-IV 2.2. Consequently, this paper adhered to this version of MIMIC-IV.

1) *Outlier removal*: During the preprocessing, we cleaned the original dataset by removing outliers, allowing us to impute missing entries. Following the approach in [22], we identified statistically extreme values, unlike the method in [23], where only values labeled 999999 and the negative (infeasible) values were removed as in [17]. We customized the two-tailed threshold to 95th percentile, which means all extreme values outside this percentile range should be removed.

2) *Missing values*: Missing data can lead to issues such as distorted patterns, trends, and biases in parameter estimation [24]. We adopt the approach of [25], excluding variables with more than 20% missing data (table I. For variables with less than 5% missing data, missing values in continuous variables that exhibit a normal distribution were imputed using the mean specific to the patient group. For continuous variables with skewed distributions, the median was used as replace missing values [26]. For variables with more than 5% missing data, we implemented a method adapted from [9], a benchmark from MIMIC-III. This approach imputed missing values using the most recent measurement when available, or a predefined 'normal' value otherwise (for further details, see [9]).

TABLE I
PERCENTAGE OF MISSING VALUES FOR VARIOUS VARIABLES

Variables	Percentage of missing values
Heart rate	4.23%
Respiratory rate	4.89%
MBP	4.39%
SBP	7.81%
DBP	7.81%
AOS	10.93%
Glucose	12.68%
Temperature	19.54%

3) *Time-series representation*: The following continuous data were extracted from MIMIC-IV ICU: heart rate (HR), systolic blood pressure (SBP), diastolic blood pressure (DBP), mean blood pressure (MBP), respiratory rate (RR), body temperature, Arterial O2 Saturation (AOS) and Glucose. Next, we applied the pipeline from [22], these variables were organized into uniform 4-hour intervals for consistency. The time-series data is typically recorded every 4 hours, so we opted for a 4-hour interval for consistency. If a patient has multiple entries within a 4-hour period, the mean value was computed and replaced by those entries with a single value for consistency. Conversely, if there are no measurements within a 4-hour slot, we addressed this as a missing value issue and resolved it using the method outlined in [9]. Since the average stay in the ICU is 11 days (with a standard deviation of 13), we limited the selection to patients who stayed in the ICU for more than 10 days. For each binned time interval, the pipeline assigned a feature value of 1 if it is recorded during that interval, and 0 otherwise. This preprocessing resulted in a matrix of size $60 \times 860 \times 860 \times 8$ for each stay, with records exceeding 10 days excluded, leaving a total of 51,080 stay records. The 10-day threshold was chosen to focus on longer ICU stays, which provide more reliable temporal data. Additionally, since the data were extracted from department where all subjects have relatively severe health issues, facing a survivorship bias problem [27]. This means the inference model may show higher accuracy for severe patients.

C. Model Implementation

We selected 3 previously proposed methods for clinical time-series forecasting to compare with our approach: the classical statistical model [8], the RNN-based method [10],

and the VAE+RNN-based method [14]. Since none of these methods have been evaluated on the MIMIC-IV dataset, we retrained them on the new dataset using carefully chosen parameters, which are explained in detail below.

The models were implemented with T4 GRU, with all temporal features re-scaled in value between -1 and 1. The training took approximately 3 hours to complete for CVLSTM and 6.5 hours for LTVLSTM. We search the optimal number of lag (past time series values) from 1 to 10, and use the same strategy to search optimal (the number of past observations) and (the size of moving average window) for ARIMA, with the optimal differencing degree searched from 0 to 4. For VRNN, we search the optimal batch size from $\{32, 64, 128\}$ and set the maximum iteration to be 50 epoches. The learning rate is searched from $\{0.001, 0.01, 0.1\}$. The dimension of the LSTM hidden states are searched from $\{8, 16, 32, 64, 128\}$, and the number of layers is no more than 3.

1) *Implementation detail of CVLSTM*: Table II shows the implementation details of CVLSTM, where X and Z denoted the dimension of x_t and z_t respectively. This table also includes details on the number of hidden layers, batch size and epoch. The batch size selection was guided by prior research [15]. For the choice of the number of hidden layers and their size, we experimented with many combinations, including setting from previous work[28]. The final choice of the hidden size, number of layers and number of epochs were based on test set performance. However, the optimal values produced the best result in the test set was different from any of the settings in [12][19][28]. There are about

TABLE II
IMPLEMENTATION DETAILS OF CVLSTM

Model	X	Z	No. Layers	Hidden	Batch	Epoch
CVLSTM	8	2	2	60	128	50

13000 unique disease diagnosis in MIMIC-IV. To simplify the calculation and prevent overfitting, we group all disease into 5 main categories (matching with the dimensions in II-A1): heart disease, blood pressure disease, kidney disease, respiratory disease, and Others. Therefore, the latent classifier variable $w \in \mathbb{R}^5$. This provided the model with priori information on potential patient pattern as described earlier.

2) *Implementation detail of LTVLSTM*: For layer size and subsequence length are searched from $\{2, 4, 6, 8, 10\}$ and $\{10, 20, 30, 40, 50, 60\}$ respectively. When the subsequence length is 60, we actually have the whole time series. We run each method 50 times and report the average accuracy as the final results.

D. Performance Metric

We use the oftenly used Root Mean Square Error (RMSE) as our evaluation metric. This can be written as

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^M (y_i - \hat{y}_i)^2} \quad (25)$$

where M denotes the size of the test dataset; y_i and $\hat{y}_i$ are the vectors representing the true and predicted values of all eight temporal features for the i th patient.

IV. RESULTS

In this section, we first report the average Root Mean Square Error (RMSE) on the test dataset. Even-though we only have the result for one step ahead prediction due to training time of LTVLSTM model, I believe it still could demonstrate our model's robustness. We have present the result in table III.

TABLE III
RMSE WITH ROUNDED STANDARD DEVIATIONS ON ONE AHEAD FORECASTING TASKS ON TEST DATA

Step Size	AR	ARIMA	LSTM
1	0.02220 ± 0.0139	0.01914 ± 0.0143	0.01420 ± 0.0068
Step Size	VRNN	CVLSTM	LTVLSTM
1	0.01220 ± 0.0040	0.01032 ± 0.0024	0.00962 ± 0.0023

ARIMA shows lower RMSE than AR, which is expected since ARIMA extends AR by incorporating differencing to account for non-stationary and integrating moving average terms. However, both AR and ARIMA perform worse than LSTM because they assume a linear relationship in the data and lack the capacity to capture complex non-linear temporal dependencies. In contrast, LSTM, being a recurrent neural network architecture, is specifically designed to learn long-term dependencies in sequential data, making it significantly more effective for time series forecasting tasks where such patterns are present.

VRNN further improves upon LSTM by introducing a probabilistic framework, which accounts for uncertainty in the latent space. This enables VRNN to model the variability in the data more effectively, leading to lower RMSE than LSTM. Despite this advantage, VRNN still falls short of CVLSTM models because additional latent classifier helps the model to learn the local patterns better. LTVLSTM shows a slight improvement over CVLSTM by explicitly learning local patterns, which enables the model to capture more local details. However, according to table IV, we see that the improvement of LTVLSTM to CVLSTM is not significant. Nevertheless, the comparisons involving CVLSTM models demonstrate statistically significant differences. The p-value of 0.050 for VRNN versus CVLSTM shows that CVLSTM's ability to explicitly model local patterns through the additional latent classifier leads to a significant performance gain. Furthermore, the p-values of 0.032 for VRNN versus LTVLSTM, respectively, highlight the impact of explicitly learning local patterns in LTVLSTM. It shows that our new model CVLSTM and LTVLSTM outperforms the previous model on the new dataset MIMIC-IV.

Furthermore, the standard deviations for AR and ARIMA are larger, indicating these models are less consistent across different test instances. In contrast, LSTM, VRNN, CVLSTM and LTVLSTM exhibit much smaller standard deviations,

TABLE IV
MANN-WHITNEY U TESTS ARE PERFORMED TO FIND IF THE ERROR DIFFERENCES ARE SIGNIFICANT USING STANDARD $\alpha = 0.05$. THE RESULTING P-VALUES FOR BOTH STATISTICAL TESTS ARE PRESENTED.

Step Size	ARIMA < LSTM	LSTM < VRNN	VRNN < CVLSTM	VRNN < LTVLSTM	CVLSTM < LTVLSTM
1	0.078	0.24	0.050	0.032	0.10

reflecting more stable and robust prediction. This further underscores the limitations of traditional statistical models, as their assumptions about data linearity and stationary often do not hold in complex time series datasets. Figure 3

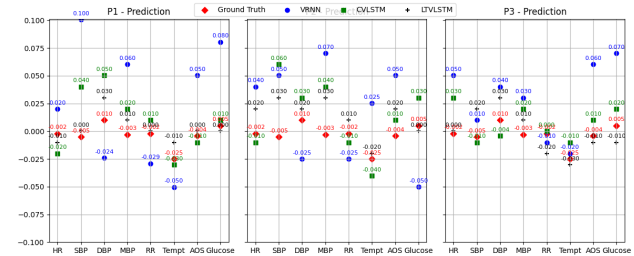


Fig. 3. one step predicted values for all 8 temporal variables from three randomly chosen patient

shows the one-step-ahead prediction of VRNN, CVLSTM, and LTVLSTM for three representative patients. The ground-truth series for each of the eight features (HR, SBP, DBP, MBP, RR, Tempt, AOS, and Glucose) is plotted in red. As can be observed, CVLSTM and LTVLSTM outperform VRNN on most of the features—particularly on MBP, SBP, Glucose, and AOS—across all three patients. LTVLSTM, in particular, produces extremely close prediction to the ground truth, which appears at first glance to contradict the average results shown in Table III. However, this discrepancy arises because these three patients happen to be cases in which LTVLSTM's prediction align especially well with the true values. By including additional patient-diagnosis information, the CVLSTM framework already benefits from more contextual data, and LTVLSTM adds yet another layer of temporal modeling that can further boost performance. Overall, the plots indicate that LTVLSTM may slightly outperform CVLSTM on these specific patients, consistent with the notion that richer temporal modeling yields more accurate forecasts.

V. CONCLUSION

This paper introduces two innovative models, CVLSTM and LTVLSTM, to improve clinical time series forecasting by integrating disease classification and hierarchical variational modeling. CVLSTM outperforms traditional models by incorporating disease class information, while LTVLSTM further refines prediction by explicitly learning local subsequences and their temporal dynamics. Extensive experiments on the MIMIC-IV dataset confirm the robustness of our methods, with LTVLSTM achieving the best overall performance in

terms of RMSE. Our findings highlight the value of incorporating latent class information and hierarchical variational structures to address the complexity of clinical data, paving the way for more accurate and actionable prediction in healthcare applications. Future work could explore multi-step forecasting to extend the applicability of these models to longer-term predictions. Additionally, applying these methods to diverse clinical datasets could further validate their effectiveness and generalizability in improving patient care and hospital efficiency.

REFERENCES

- [1] World Health Organization. Health systems financing: the path to universal coverage. 2010.
- [2] Yip W, Hafez R, Organization WH. Improving health system efficiency: reforms for improving the efficiency of health systems: lessons from 10 country cases. World Health Organization. 2015.
- [3] Mutter RL, Rosko MD, Wong HS. Measuring hospital inefficiency: the effects of controlling for quality and patient burden of illness. *Health Serv Res.* 2008 Dec;43(6):1992-2013. doi: 10.1111/j.1475-6773.2008.00892.x.
- [4] Shickel B, Tighe PJ, Bihorac A, Rashidi P. Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis. *IEEE J Biomed Health Inform.* 2018 Sep;22(5):1589-1604. doi: 10.1109/JBHI.2017.2767063.
- [5] Choi E, Schuetz A, Stewart WF, Sun J. Medical Concept Representation Learning from Electronic Health Records and its Application on Heart Failure Prediction. *arXiv.* 2016 Feb; 45. [Online]. Available: <http://arxiv.org/abs/1602.03686>.
- [6] Miotto R, Li L, Kidd BA, Dudley JT. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific reports.* 2016;6(no. April):26094.
- [7] Maddox TM, Rumsfeld JS, Payne PRO.. Questions for artificial intelligence in health care. *JAMA* 2019; 321 (1): 31–2
- [8] Junior, Paulo Rotela, Fernando Luiz Riêra Salomon, and Edson de Oliveira Pamplona. "ARIMA: An applied time series forecasting model for the Bovespa stock index." *Applied Mathematics* 5.21 (2014): 3383.
- [9] Z.D. Akşehir, E. Kiliç, How to handle data imbalance and feature selection problems in CNN-based stock price forecasting, *IEEE Access* 10 (2022) 31297–31305.
- [10] R. He, Y. Liu, Y. Xiao, X. Lu, S. Zhang, Deep spatio-temporal 3D densenet with multiscale ConvLSTM-Resnet network for citywide traffic flow forecasting, *Knowl.-Based Syst.* 250 (2022) 109054.
- [11] D. Quang, X. Xie, DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences, *Nucleic Acids Res.* 44 (11) (2016) e107.
- [12] D. P. Kingma and M. Welling, Auto-encoding variational bayes, *arXiv preprint arXiv:1312.6114*, 2013.
- [13] W. Chen, L. Tian, B. Chen, L. Dai, Z. Duan, M. Zhou, Deep variational graph convolutional recurrent network for multivariate time series anomaly detection, in: *International Conference on Machine Learning*, PMLR, 2022, pp. 3621–3633.
- [14] U. Ullah, Z. Xu, H. Wang, S. Menzel, B. Sendhoff, T. Bäck, Exploring clinical time series forecasting with meta-features in variational recurrent models, in: *International Joint Conference on Neural Networks*, IEEE, 2020, pp. 1–9.
- [15] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.
- [16] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *arXiv*, 1506.02216, 2015.
- [17] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pages 2199-2207, 2016.
- [18] Hennig, Jay A., Akash Umakantha, and Ryan C. Williamson. A classifying variational autoencoder with application to polyphonic music generation. *arXiv preprint arXiv: 1711.07050* (2017).
- [19] Fraccaro, Marco, et al. "Sequential neural models with stochastic layers." *Advances in neural information processing systems* 29 (2016).
- [20] Sønderby, Casper Kaae, et al. "Ladder variational autoencoders." *Advances in neural information processing systems* 29 (2016).
- [21] Johnson, Alistair, et al. "Mimic-iv." *PhysioNet*. Available online at: <https://physionet.org/content/mimiciv/1.0/> (accessed August 23, 2021) (2020): 49-55.
- [22] Gupta, M., Gallamozza, B., Cutrona, N., Dhakal, P., Poulain, R., & Beheshti, R. (2022, November). An extensive data processing pipeline for mimic-iv. In *Machine Learning for Health* (pp. 311-325). PMLR.
- [23] Aishwarya Mandyam, Elizabeth C Yoo, Jeff Soules, Krzysztof Laudanski, and Barbara E Engelhardt. COP-E-CAT: cleaning and organization pipeline for ehr computational and analytic tasks. In *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 1-9, 2021.
- [24] I. Pratama, A. E. Permanasari, I. Ardiyanto and R. Indrayani, A review of missing values handling methods on time-series data, 2016 *International Conference on Information Technology Systems and Innovation (IC-ITSI)*, Bandung, Indonesia, 2016, pp. 1-6, doi: 10.1109/IC-ITSI.2016.7858189.
- [25] Sun, Yiwu & He, Zhaoyi & Ren, Jie & Wu, Yifan. Prediction model of in-hospital mortality in intensive care unit patients with cardiac arrest: a retrospective analysis of MIMIC - database based on machine learning. 2023. 10.21203/rs.3.rs2551943/v1.
- [26] Zhang Z. Missing data imputation: focusing on single imputation. *Ann Transl Med.* 2016 Jan;4(1):9. doi:

- 10.3978/j.issn.23055839. 2015.12.38. PMID: 26855945; PMCID: PMC4716933.
- [27] Elston DM. Survivorship bias. *J Am Acad Dermatol.* 2021 Jun 18:S0190-9622(21)01986-1. doi: 10.1016/j.jaad.2021.06.845. Epub ahead of print. PMID: 34153385.
- [28] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, pp. 2980-2988, 2015.