Dynamic Modeling of Patient Vital Signs: Leveraging Markov Chain Principles with Neural Networks for Irregular Time-Series Prediction

Anonymous Submission

***Abstract*—Analyzing patient health through irregular time series vital sign data demands innovative methods beyond conventional imputation techniques. This study introduces a novel approach diverging from prevailing attention-based models to explicitly capture temporal patient evolution. We adopt a paradigm where patients are viewed as dynamic systems evolving over time, with their vital signs encapsulating the system's states. Our conceptual framework draws parallels to a Markov chain, exploring the transitions between states within a unit of time. To navigate the challenge of a vast state space, we employ a neural network to model expected transitions. Our method portrays the patient's progression within one unit of time as the system evolves from one state to another, and forecasts states into the future. We outline the training process using irregular time series data and demonstrate its efficacy through analysis on two large vital sign data sets. Comparative analysis against attention-based models emphasizes the effectiveness and efficiency of our approach. This research heralds a promising avenue for patient vital sign analysis, providing insights into temporal patient evolution without relying on imputation methods, thereby enhancing predictive accuracy and interpretability of models.**

***Index Terms*—Vital Sign Prediction, Irregular Time Series, Markov Chains**

I. Introduction

Vital signs are measurements of physiological parameters that represent a set of quantitative measures used to determine a patient's general health and viability. They significantly influence doctors' and nurses' interpretation of a patient's overall condition and subsequently impact the course of treatment for each individual [1]. Traditionally, vital signs are a crucial component of nursing assessments and serve as early warning indicators for changes in a patient's condition, particularly in emergency departments (ED). Consequently, monitoring patients' vitals is essential for patient safety, as any signs of patient deterioration can be promptly addressed, preventing potentially costly delays in response [2], [3].

Utilizing artificial intelligence and machine learning to predict future patient states has the potential to detect patient deterioration before visible signs appear, allowing for increased attention to be given to the patient [4]. However, in a typical ED, vital sign monitoring occurs at irregular intervals, contingent upon the patient's condition, the availability of medical staff, hospital policy, nursing judgment, written physician orders, and various other factors [5]. Despite being the most commonly performed task in the ED, the frequency of vital sign monitoring is usually inconsistent [1].

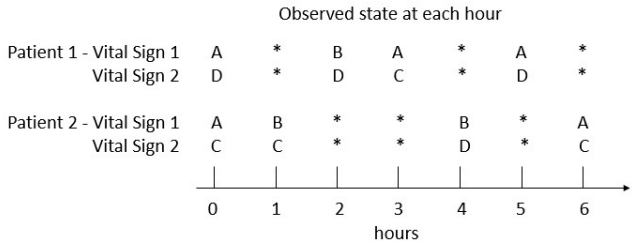


Fig. 1: An illustration of irregular time observations for two patients with two vital signs over six hours. An asterisk refers to the absence of data at a particular time point.

Hence, in vital sign analysis, a substantial amount of patient data constitutes irregular time series, where patient states are recorded at non-uniform time intervals. This presents challenges for machine learning modelling as each patient's time series data becomes diverse, complicating machine learning approaches in identifying a common underlying pattern. These data do not naturally yield a fixed-dimensional representation as required by many standard machine learning models [6]. Moreover, each time series may vary in length, posing difficulties for methods assuming fixed dimensional spaces.

Additionally, monitoring patient states involves more than just measuring one variable, resulting in multivariate time series [7]. Forecasting multivariate patient states poses an even greater challenge, as historical data for a variable not only contributes to its future value but also because interaction effects between one variable and every other variable are possible and could significantly impact its future value [8].

Consider a scenario with two patients, each characterized by two vital signs: 1 and 2. Infrequent observations are made on these two patients over a six-hour period, as depicted in Figure 1. Initially, the first patient's vital signs are observed with values `A` and `D` for vital signs 1 and 2, respectively. Subsequent observations occur at hours 2, 3, and 5, with recorded values (`B`, `D`), (`A`, `C`), and (`A`, `D`). The second patient is observed at hours 0, 1, 4, and 6, with different observations. They represent irregular time series, unlike "regular" time series where observations are consistently made at fixed frequencies.

Traditionally, the standard approach for handling such irregular data sets involves imputation [6], [9], thereby transforming the irregular dataset into a regular one. Simple statistical techniques have been employed, where missing values are filled in using methods such as zero-filling, mean substitution, moving averages, or the last observed value. However, these

methods can introduce bias and diminish accuracy [10].

Recently, there has been a development of machine learning models specifically tailored for irregular time series. As discussed in Section II, several recent models have emerged, incorporating innovative applications of Transformers (self-attention) [11]. Instead of imputation, these models are constructed solely based on available observations within the data, treating each time series as a set of observation triplets comprising time, variable, and value. These triplets are then embedded using an embedding mechanism and encode contextual information using a Transformer-based architecture with various attention mechanisms [12]. While demonstrating success in predicting the occurrence of clinical events (classification), their performance in predicting vital sign values (regression) has not been reported in the literature. Furthermore, these attention-based models inherently lack algorithmic transparency, making them challenging to explain.

In this study, we adopt a different approach. Rather than relying on imputation to filling in missing values or using attention mechanisms to encode time, we explicitly model *time progression* of each patient. Essentially, we consider a patient as a *dynamic system* evolving over time, with the patient's vital signs characterize the system's *states*. Our aim is to represent the patient system as a Markov Chain, conceptualizing a transition matrix that specifies the probabilities of the system moving from one state to another with a unit time. This transition matrix enables us to describe how a patient progresses over time. Given the large state space, instead of explicitly solving for this transition matrix, we employ a neural network $F : S \mapsto S$ that maps between states (S) with parameters θ, describing expectations of state transitions. Effectively, given a patient in state s at time n,

$$E[Z_{n+1}|Z_n = s] \approx F(s;\theta)$$

describes the expected state transition in one unit time; and

$$E[Z_{n+t}|Z_n = s] \approx F^{(t)}(s;\theta)$$

describe system states in t units time. (Note that $F^{(t)}(s;\theta)$ denotes applying F recursively t-times.) In this work, we elaborate on how such an F can be trained from irregular time series data by finding the optimal θ and demonstrate its effectiveness in two large vital sign data sets, in comparison with attention based approaches.

In summary, the contributions of our work are as follows:

1) Conceptualization of a "patient state", which consists of the measurements of vital signs at a point in time.
2) Using neural networks with trainable weights and biases to model state transition.
3) Novel training method for the neural network to accommodate irregular time-series data.

The rest of this paper is organized as follows. Section II describes existing works related to irregular time modeling, while Section III describes our approach using Markov state transition as well as other methods for comparison. The data sets, experiments conducted, and the results are covered in Section IV, and finally, concluded in Section V.

II. Related Work

Given the sequential nature of time-series data, recurrent neural networks (RNNs) such as long short-term memory (LSTM) models, gated recurrent units (GRUs), and transformers have gained popularity. In the realm of irregular time series, [9] provides a systematic review of methods employing "gated RNNs." These models, based on RNN architectures with dedicated connections indicating missing values, have found applications across various domains including medical research [13]–[15], traffic monitoring [16], [17], and environmental monitoring [18], [19].

Imputation encompasses a broad spectrum of methods to handle irregular time series data, with numerous techniques outlined in the literature. These methods include replacement strategies [20], interpolation techniques [21], re-sampling methodologies [22], and approaches leveraging Gaussian processes [23]–[25].

Additionally, neural network models have been designed with specific structures to identify missing data. One approach involves masking missing values using "missing data indicators" like NaN values, allowing models to utilize this indicator set and skip data points lacking valid observations [15], [26]. For example, [27] introduced a time-aware long short-term memory (LSTM) modification that adjusts the hidden state to accommodate time gaps. Another model, the GRE-D proposed by [26], adapts the gated recurrent unit (GRU) cell to decay inputs and hidden states across unobserved time intervals.

Although both imputation and missing data identification have displayed some success, they can incur excessive computations and introduce unnecessary noise, particularly when dealing with high missing rates [12]. Consequently, a recent trend has emerged where attention mechanisms [28] are employed to handle irregular time series data more effectively. We elaborates on two notable recent works, STraTS [12] and PrimeNet [29], which are state-of-the-art methods and have exhibited significant success in predicting irregular time-series data, particularly vital signs, in comparison with all aforementioned methods. These models hence serve as the benchmarks for our experimental comparisons later in this study, shown in Section IV.

A. STraTS

Tipirneni and Reddy [12] introduced the Self-supervised Transformer for Time-Series (STraTS) model, treating time-series as a collection of observation triplets. They employ a "continuous value embedding" approach to encode continuous time and variable values without discretization. The model utilizes a Transformer component with multi-head attention layers to learn contextual embedding for these triplets. The resultant time series embedding is then merged with demographic information, embedded via a separate feedforward network, and forwarded into a prediction head to forecast a target value for each time series. Their experimental focus has been on predicting ICU mortality, achieving ROC-AUC scores of 0.891 and 0.839 on the MIMIC-III [30] and PhysioNet [31] data sets, respectively.

B. PrimeNet

Chowdhury et al. [29] introduced the PrimeNet model, utilizing a learnable time representation known as Time Embedding [32]. This Time Embedding incorporates trainable weights and biases to transform time into a vector representation, comprising linear and periodic terms. These Time Embedding vectors form the query and key vectors for Time-Feature Attention (TFA), combined with feature values as the value vector. Feature-Feature Attention (FFA) subsequently employs self-attention on the output from TFA, followed by residual and feedforward layers. The resulting outputs can be directed to task-specific layers, enabling adaptation to various downstream prediction tasks, including interpolation, regression, and classification. Notably, on the MIMIC-III and PhysioNet datasets, the model achieved ROC-AUC scores of 0.838 and 0.842, respectively, in predicting patient mortality.

III. Method

A Markov chain is a stochastic model that describes a sequence of events where the probability of transitioning from one state to another depends solely on the current state and not on the sequence of events that preceded it, encapsulating the Markov property [33]. It consists of a set of states and transition probabilities, forming a discrete-time or continuous-time process, applicable in various fields from physics to finance and natural language processing. The chain's memoryless property allows for the prediction of future states based solely on the current state, enabling the analysis of random processes and system behaviors over time, essential in modeling systems with probabilistic dynamics and understanding state-dependent probabilistic relationships.

In our context, we consider *discrete-time Markov chains with continuous state spaces*, the system's evolution occurs at distinct time steps while the potential states span a continuous set. This framework extends the Markov chain concept to scenarios where the system's state variables exist within a continuum. The principles of the Markov property persist, dictating that future states depend solely on the present state, allowing for the modeling and understanding of complex systems with continuous state variables across time.

In this paper, we will use Markov chains to model the space of patients' vital signs. For presentation simplicity, we assume that the state space S has an enumerable approximation with cardinality $N \in \mathbb{N}$, i.e., we can write

$$S = \{s_1, s_2, \ldots, s_N\}$$

to denote the space of all possible patient states. Then, we consider chains defined on this state space S. That is,

$$(Z) = \langle Z_1, Z_2, \ldots, Z_z \rangle,$$

in which $z \in \mathbb{N}$ and $Z_i \in S$. Most importantly, we consider Z having the Markov property in the sense that for any i, the probability distribution on Z_{i+1} depends only on the state Z_i at time i, and not on previous values of Z. In other words, state transitions are memory-less. Formally,

$$\begin{aligned}\Pr(Z_{n+1} = s'_{i+1} | Z_n = s'_i, \ldots, Z_1 = s'_1) \\ = \Pr(Z_{n+1} = s'_{i+1} | Z_n = s'_i).\end{aligned}$$

Furthermore, we consider time-homogeneous Markov chains [34]. In other words, the probability of transition does not depend on time. Formally, for any $n, m \in \mathbb{N}$, it holds that

$$\Pr(Z_{n+1} = s'_{i+1} | Z_n = s'_i) = \Pr(Z_{m+1} = s'_{i+1} | Z_m = s'_i).$$

At the core of a Markov model is the Markov transition matrix P. For a state space with N states, the matrix is an N-by-N square matrix, defining the probabilities of transitioning from one state to another in a single time step, encapsulating the dynamic behaviour of the system.

$$P = \begin{bmatrix} P_{11} & P_{12} & \ldots & P_{1N} \\ P_{21} & P_{22} & \ldots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \ldots & P_{NN} \end{bmatrix}$$

Each entry in the matrix represents the probability of transitioning from a current state to a future state, e.g.,

$$P_{i,j} = \Pr(Z_j | Z_i)$$

denotes the probability of transition from Z_i to Z_j. From the Markov transition matrix, we can represent the expected value of the transition from the current state s_n to its next state s_{n+1} as:

$$E[Z_{n+1} | Z_n = s_i] = \sum_{j=1}^{N} s_j \cdot P_{ij}. \tag{1}$$

In other words, Equation 1 describes the expected outcome of a single step state transition from a given state s_i. Thus, with all states s_i identified from a data set, if we can compute P, then we will be able to describe and predict how a patient's vital signs progress through time using

$$E[Z_{n+t} | Z_n = s_i] = \sum_{j=1}^{N} s_j \cdot P_{ij}^{(t)}, \tag{2}$$

in which $P_{ij}^{(t)}$ is t-th power of P, describing the transition probabilities for moving from Z_n to all possible states after t time steps.

However, since N is large, it is not feasible to estimate P directly. In this work, we provide an alternative approach to compute $E[Z_{n+1} | Z_n = s_i]$ by constructing a neural network regression model $F : S \mapsto S$ that maps between states directly, such that

$$F(s_i; \theta) \approx \sum_{j=1}^{N} s_j \cdot P_{ij}, \tag{3}$$

in which θ is the parameters (weights and biases) of F. With this, we can effectively compute the vital sign progression with

$$E[Z_{n+t} | Z_n = s_i] \approx F^{(t)}(s_i; \theta). \tag{4}$$

In words, given a patient's vital signs s_i at time i, we can estimate the vital signs at time $i+t$ by feeding s_i into the regression model F exactly t times, recursively, such that the output of one iteration is used as the input for the next.

To train the regression model F from data, we take several steps. Firstly, we consider a data set D with patient vital signs collected from τ patients. Data from each patient forms a chain Z,

$$D = \{(Z^1), \ldots, (Z^\tau)\}.$$

The state space S will contain the union of all states from all patients in D,

$$\bigcup_{i=1}^{\tau} (Z^i) \subseteq S.$$

Then, we stratify D into m partitions $D^1, \ldots, D^m$ such that:

1) Each D^i is a set of pairs, $D^i = \{(x_1^i, y_1^i), (x_2^i, y_2^i), \ldots\}$;
2) Each pair (x_j^i, y_j^i) is drawn from a chain (Z), such that $x_j^i = Z_k$ and $y_j^i = Z_{k+i}$; and
3) For each pair (x_j^i, y_j^i) in D^i, there is no Z_a in any chain (Z) such that for $x_j^i = Z_k$, $y_j^i = Z_{k+i}$, it is the case that $k < a < k+i$.

This stratification ensures that (1) all pairs are extracted from the data set D, (2) pairs in the same partition D^i are data points that are exactly i-steps apart from each other; and (3) pairs represent observations that are closest to each other.

TABLE I: A data set containing two patients with irregular observations crossing 5 time steps.

Patient ID	t_1	t_2	t_3	t_4	t_5
1	s_1	NA	s_2	NA	s_3
2	s_4	s_5	NA	s_6	s_7

For instance, consider a data set containing two patients as shown in Table I. This data set forms two partitions with

$$D^1 = \{(s4, s5), (s6, s7)\},$$
$$D^2 = \{(s1, s2), (s2, s3), (s5, s6)\}.$$

On this stratified data $D^1, \ldots, D^m$, with a loss function L, the model training can be concisely expressed as

$$\theta^* = \arg\min_\theta \sum_{k=1}^{m} \frac{1}{|D^k|} \left(\sum_{i=1}^{|D^k|} L(y_i^k, F^{(k)}(x_i^k; \theta)) \right), \quad (5)$$

where $|D^k|$ is the cardinality of D^k.

To train the model F by computing θ^* in Equation 5, we take an iterative approach by training F with respect to each D^k separately. Thus, θ^* is updated after each pass to D^k, for each k. Training stops once some termination condition is met.

We discuss the training in details as follows. Firstly, it is easy to see that for D^1, this is nothing but standard gradient descent training,

$$\theta^* = \arg\min_\theta \sum_{i=1}^{|D^1|} L(y_i^1, F(x_i^1; \theta)),$$

which is solved with backpropagation.

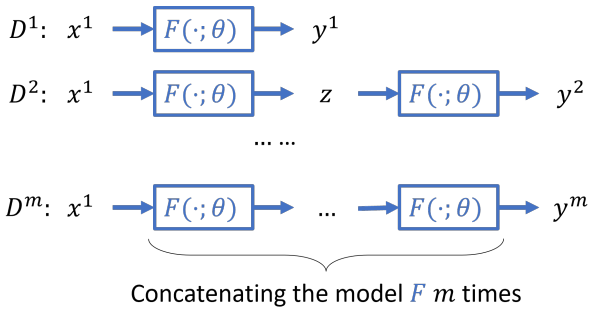


Fig. 2: Training the prediction model can be viewed as training several concatenated models sharing the same parameters θ.

For $D^k, k > 1$, we can still use backpropagation with accumulated gradients. Effectively, as illustrated in Figure 2, we can consider $F^{(k)}(x, \theta)$ as k copies of $F(x, \theta)$ concatenated with one another, all with the shared weights and biases θ. We illustrate the case $k = 2$, with the Mean Squared Error (MSE) loss,

$$L(\theta) = \frac{1}{2} ||F(F(x; \theta)) - y||^2, \quad (6)$$

as follows. We compute the first application of F with

$$z = F(x; \theta),$$

and the second application of F

$$\hat{y} = F(z; \theta).$$

In the backward pass, we first compute the gradient of the loss with respect to the second application of F:

$$\delta_2 = (\hat{y} - y) \odot F'(z; \theta),$$

where $\odot$ denotes the element-wise multiplication and F' is the gradient of F with respect to its parameters θ. Then we compute the gradient of the loss with respect to the first application of F:

$$\delta_1 = F'(x; \theta) \odot (F'(z; \theta)^T \delta_2).$$

The overall gradient of the loss with respect to both applications is the sum of the two:

$$\delta = \delta_1 + \delta_2.$$

Then in each iteration, θ is updated using the delta rule

$$\theta \leftarrow \theta - \alpha\delta,$$

with some learning rate α.

It is easy to see that the above training process generalizes to any number of repeated applications of F as we just need to keep accumulating the loss δ. Moreover, we see that this process shares the same convergence properties as backpropagation training in general, i.e., it is guaranteed to converge to a local minimum and known optimization techniques such as Adaptive Moment Estimation (Adam) [35] can be applied.

The overall model training algorithm is summarized in Algorithm 1. The data is first formatted into the inputs, target outputs, and the number of intervals between the inputs and

Algorithm 1 Markov Chain Model Training

$learningRate \leftarrow 0.01$; $\gamma \leftarrow 0.1$
for $interval$ from 1 to $maxIntervalSize$ **do**
  $d \leftarrow \emptyset$
  for $timeSeries$ in $dataset$ **do**
    for row in $timeSeries$ **do**
      if row is not the last in $timeSeries$ **then**
        ($intervalSize$ of row) $\leftarrow$ $\frac{\text{time difference between current and next row}}{\text{time of one interval}}$
        ($inputs$ of row) $\leftarrow$ patient state of the current row
        ($targets$ of row) $\leftarrow$ patient state of the next row
        if ($intervalSize$ of row) = $interval$ **then**
          $d \leftarrow d \cup row$
        end if
      end if
    end for
  end for
  update model weights with gradient descent on d
  $learningRate \leftarrow learningRate \times \gamma$
end for

the target. Then, the model is trained on data of increasing interval sizes sequentially. All gradient computation is done via autograd implemented by PyTorch [36]. As the interval size increases, the learning rate decreases by a factor γ. Both learning rate and γ values were determined through random hyperparameter tuning.

To visualize how the Markov chain model models the progression of a patient's vital signs over time, Figure 3 shows a radar chart of the predictions made by the Markov chain model for one patient for the next 4 hours at hourly intervals given the current patient state at $t = 0$. Based on the predictions, values for Urine decrease while the values for SysABP, DiasABP, MAP increase, and HR remain stable.

IV. Experiments

A. Datasets

The datasets used are Physionet Challenge 2012 [31] and MIMIC-III [30], [37]. Both contain intensive care unit (ICU) records containing measurements of various physiological variables at irregular time points.

1) Physionet: The following vital signs are used.

- Diastolic arterial blood pressure (DiasABP) (mmHg)
- Mean arterial blood pressure (MAP) (mmHg)
- Systolic arterial blood pressure (SysABP) (mmHg)
- Heart rate (HR) (bpm)
- Urine output (Urine) (mL)

Set A was used for training and Set B was used for model evaluation as specified in the Challenge. Data points with missing values in any of these features were dropped from the datasets. Clinically impossible outliers where DiasABP, MAP, SysABP had 0 values and Urine values greater than 1,000 were removed. Patient who had multiple ICU records were also removed. As the majority of the remaining data were in intervals of 1 hour, discretization was employed to ensure the intervals in the data were all in multiples of 1 hour. Furthermore, models were trained on intervals of nine hours or less but were also evaluated on intervals beyond nine hours. This is to test the ability of the models to forecast patient states at time intervals further than those seen in the training set.

The post-processed training dataset has 1,768 patients, 1,006 male and 762 female, with ages ranging from 16 to 90. The post-processed testing dataset has 1,743 patients, 1,012 male and 731 female, with ages ranging from 16 to 90. The minimum, maximum, and mean values across the combined training and testing dataset as well as the standard deviation for each variable within each patient are shown in Table II.

TABLE II: Physionet: Descriptive statistics of the dataset.

	DiasABP	MAP	SysABP	HR	Urine
Minimum	3.0	4.0	10.0	14.0	0.0
Maximum	272.0	297.0	285.0	200.0	1000.0
Mean	59.5	79.7	120.0	87.4	95.5
SD	7.3	9.7	13.7	8.1	72.1

2) MIMIC-III: The dataset was preprocessed by following the steps taken in [12], with the following vital signs:

- Diastolic arterial blood pressure (DBP) (mmHg)
- Mean arterial blood pressure (MBP) (mmHg)
- Systolic arterial blood pressure (SBP) (mmHg)
- Heart rate (HR) (bpm)
- O2 Saturation (O2) (%)

Stratified sampling based on the patients' gender and age was employed to select a smaller data subset for experimentation to reduce computation times. As MIMIC-III had more data with intervals shorter than an hour than PhysioNet, the data was discretized to 15-minute intervals.

The post-processed dataset has 983 patients, 492 female and 491 male, with ages ranging from 18 to 89. The minimum, maximum, and mean values across the entire dataset as well as the standard deviation are shown in Table III.

TABLE III: MIMIC-III: Descriptive statistics of the dataset.

	DBP	HR	MBP	O2	RR	SBP
Minimum	14.0	43.0	10.0	80.0	1.0	46.0
Maximum	179.0	169.0	186.0	100.0	57.0	240.0
Mean	59.8	87.9	78.9	97.9	18.8	119.7
SD	7.6	6.3	9.6	1.3	3.2	13.6

Min-max scaling from 0 to 1 was applied to each variable, with minimum and maximum values derived from the training data. For experiments on MIMIC-III, the results are averaged from 120 iterations of random 70-30 train-test splits.

B. Methods for Comparison

1) Naive Forecast: In the case where no models are applied, we can naively assume that the best prediction for patient states in the future is the most recent input. We use the latest time point available as the naive forecast for the target.

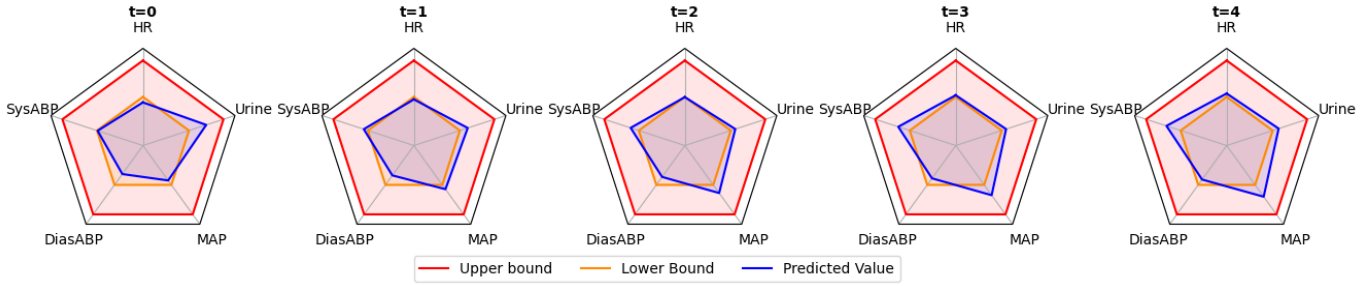


Fig. 3: Illustration of how a patient's state evolves through time. Predictions from Markov Chain model for a patient for the next 4 hours in the Physionet dataset. The upper bound represents the upper limit of what would be considered normal for an average person for that variable, while the lower bound represents the lower limit of what would be considered normal. The upper and lower bounds respectively for each variable are as follows: Heart rate (HR): 100, 60. Urine: 130, 30. Mean arterial blood pressure (MAP): 100, 60. Diastolic arterial blood pressure (DiasABP): 90, 60. Systolic arterial blood pressure (SysABP): 140, 90. The patient values are scaled according to the upper and lower bound values.

2) STraTS: We adopt STraTS [12] for vital sign prediction, keeping its self-supervision component. To forecast at various intervals into the future, the time values are encoded as values relative to the target forecasting time before embedding of time values. Hyperparameter tuning was performed to determine the dimension of the embedding layers, the number of transformer layers, the batch size, and the number of training iterations.

3) PrimeNet: In [29], pre-training of PrimeNet was used before fine-tuning on the downstream prediction tasks. When forecasting patient vital signs, we observed that pre-training did not improve performance. Thus, models were directly trained in a supervised manner on both datasets. Hyperparameter tuning was also performed to determine the dimension of the time embedding layer, the hidden size of dense layers, the batch size, and the number of training iterations.

4) LSTM Model: We also compared the Markov Chain model with existing models for regular time series prediction, namely a standard LSTM model with missing value interpolation [13]. Two LSTM layers of 100 units each, followed by a dense layer to output predictions for each vital sign are used.

The number of trainable model parameters for each method is tabulated in Table IV.

TABLE IV: Number of parameters for each method. MC is two orders of magnitudes smaller than the next smallest model.

	Physionet	MIMIC-III
MC	30	42
Naive	0	0
STraTS	6,029	6,102
PrimeNet	97,637	98,022
LSTM	123,305	123,806

C. Performance

We evaluate the performance of the MC model based on the following criteria. noitemsep

1) Short-term forecasting performance
2) Overall forecasting performance
3) Number of parameters

From Table V, we observed that the MC model had the best forecasting performance for patient states one interval into the future even with the least number of trainable parameters (Table IV); the Markov Chain model had the lowest root mean square error (RMSE) for all variables except MAP, in which it is second to STraTS. In Table VI, the MC model also shows the best forecasting performance, displaying the lowest RMSE for 4 of the 6 output variables, and a close second to LSTM for the remaining 2 variables.

TABLE V: Physionet: Average RMSE of forecasting one interval into the future. Models with the best RMSE are shown in bold while the second-best RMSEs are italicized.

	HR	SysABP	DiasABP	MAP	Urine
Naive	*0.0403*	*0.0637*	*0.0672*	0.0486	0.0887
MC	**0.0394**	**0.0597**	**0.0639**	*0.0398*	**0.0843**
STraTS	0.0459	0.0665	0.0699	**0.0369**	*0.0844*
Prime Net	0.0616	0.1272	0.1314	0.0402	0.0961
LSTM	0.1361	0.1155	0.1098	0.0687	0.1091

TABLE VI: MIMIC-III: Average RMSE of forecasting one interval into the future. Models with the best RMSE are shown in bold while the second-best RMSEs are italicized.

	DBP	HR	MBP	O2	RR	SBP
Naive	0.0566	*0.0465*	0.0603	0.1004	0.0740	0.0855
MC	*0.0540*	**0.0460**	**0.0576**	**0.0948**	**0.0698**	*0.0813*
STraTS	0.0596	0.0626	0.0638	0.1096	0.0728	0.0870
Prime Net	0.0705	0.0745	0.0684	0.1170	0.0822	0.0997
LSTM	**0.0535**	0.0494	*0.0581*	*0.1002*	*0.0707*	**0.0799**

Evaluating model performances on forecasting intervals greater than one, the MC model performs well on the Physionet dataset. In Table VII, MC has the lowest RMSE for 3 variables. On MIMIC-III (Table VIII), MC also performs well, with RMSE values within the top two for 5 variables.

Figures 4a and 4b show the average forecasting RMSE as interval size increases for Physionet and MIMIC-III respectively. For Physionet, the RMSEs for the first 12 intervals, each representing 1 hour, are shown, while for MIMIC-III, the first 8 intervals, each representing 15 minutes, are shown. In Figure 4a, the RMSE generally gradually increases as interval size increases. In Figure 4b, there is an increasing trend in RMSE for HR and RR, but for the remaining 4 variables, the RMSE fluctuates after the 4th interval, especially for the

TABLE VII: Physionet: Average RMSE across all test data points. Models with the best RMSE are shown in bold while the second-best RMSEs are italicized.

	HR	SysABP	DiasABP	MAP	Urine
Naive	*0.0643*	0.0807	0.0884	0.0572	0.1182
MC	**0.0617**	**0.0745**	**0.0830**	0.0519	0.1072
STraTS	0.0701	*0.0768*	*0.0875*	*0.0501*	*0.1057*
Prime Net	0.0701	0.1337	0.1317	**0.0478**	**0.1012**
LSTM	0.1347	0.1185	0.1125	0.0832	0.1144

TABLE VIII: MIMIC-III: Average RMSE across all test data points. Models with the best RMSE are shown in bold while the second-best RMSEs are italicized.

	DBP	HR	MBP	O2	RR	SBP
Naive	0.0668	*0.0751*	0.0758	0.1219	0.0927	0.1026
MC	**0.0636**	**0.0745**	*0.0703*	**0.1089**	*0.0921*	0.0992
STraTS	*0.0642*	0.0842	0.0711	*0.1170*	**0.0897**	**0.0941**
Prime Net	0.0690	0.0875	**0.0690**	0.1185	0.0925	*0.0977*
LSTM	0.0667	0.0913	0.0727	0.1407	0.1022	0.0996

variables related to blood pressure. Looking at the naive graph, the changes in the variable values after 6 intervals and 8 intervals are smaller than 1 interval before, which perhaps is due to some periodic nature of these variables in the dataset.

In Figure 4a, the MC model displayed more consistent performance across all 5 variables as compared to PrimeNet, which performed especially poorly on SysABP and DiasABP. On MIMIC-III, the MC model has the best performance for variables HR and O2, and a comparable performance to other models for the other variables in Figure 4b, even with its small number of parameters (Table IV). MC performs well consistently for the first 4 intervals (up until 1 hour from the input data) across the 6 variables. We observe that the RMSE increases sharply for after the first 4 intervals, suggesting that changes in patient state in the first hour may have a different trend as compared to the subsequent hour.

V. Conclusion

We introduce a novel approach to the analysis of patient vital sign data characterized by irregular time series. Our approach considers patient states as dynamic systems evolving over time, bypassing the need for imputation by explicitly modeling temporal evolution. By conceptualizing patients' vital signs as indicative of system states and leveraging a neural network-based approach to model transitions, we have demonstrated the effectiveness of capturing temporal evolution without relying on fixed transition matrices or attention mechanisms. The results from experiments conducted on MIMIC-III and PhysioNet, two large vitals data sets, affirm the promise of our approach. With models that are two orders of magnitudes smaller than competitors, our approach outperform state-of-the-art models in the literature. This research prepares for future explorations in patient state analysis, encouraging the adoption of innovative methodologies that emphasize temporal evolution modeling over traditional approaches. By advancing the understanding of temporal patient states, our approach contributes significantly to the development of more accurate and efficient predictive models in healthcare.

References

[1] K. D. Johnson, C. Winkelman, C. J. Burant, M. Dolansky, and V. Totten, "The factors that affect the frequency of vital sign monitoring in the emergency department," *Journal of Emergency Nursing*, vol. 40, no. 1, pp. 27–35, 2014.

[2] W. Q. Mok, W. Wang, and S. Y. Liaw, "Vital signs monitoring to detect patient deterioration: An integrative literature review," *International journal of nursing practice*, vol. 21, pp. 91–98, 2015.

[3] B. D. Winters, S. J. Weaver, E. R. Pfoh, T. Yang, J. C. Pham, and S. M. Dy, "Rapid-response systems as a patient safety strategy: a systematic review," *Annals of internal medicine*, vol. 158, no. 5_Part_2, pp. 417–425, 2013.

[4] T. Shaik, X. Tao, N. Higgins, L. Li, R. Gururajan, X. Zhou, and U. R. Acharya, "Remote patient monitoring using artificial intelligence: Current state, applications, and challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 13, no. 2, p. e1485, 2023.

[5] S. L. Javan, M. M. Sepehri, M. L. Javan, and T. Khatibi, "An intelligent warning model for early prediction of cardiac arrest in sepsis patients," *Computer methods and programs in biomedicine*, vol. 178, pp. 47–58, 2019.

[6] S. C.-X. Li and B. Marlin, "Learning from irregularly-sampled time series: A missing data perspective," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5937–5946.

[7] L. C. Cancio, A. I. Batchinsky, W. L. Baker, C. Necsoiu, J. Salinas, A. L. Goldberger, and M. D. Costa, "Combat casualties undergoing lifesaving interventions have decreased heart rate complexity at multiple time scales," *Journal of critical care*, vol. 28, no. 6, pp. 1093–1098, 2013.

[8] A. Sengupta, A. Prathosh, S. N. Shukla, V. Rajan, and C. K. Reddy, "Prediction and imputation in irregularly sampled clinical time series data using hierarchical linear dynamical models," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2017, pp. 3660–3663.

[9] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, 2021.

[10] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[11] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[12] S. Tipirneni and C. K. Reddy, "Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 6, pp. 1–17, 2022.

[13] M. Nguyen, N. Sun, D. C. Alexander, J. Feng, and B. T. Yeo, "Modeling alzheimer's disease progression using deep recurrent neural networks," in *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*. IEEE, 2018, pp. 1–4.

[14] Y.-J. Kim and M. Chi, "Temporal belief memory: Imputing missing data during rnn training." in *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018)*, 2018.

[15] Z. C. Lipton, D. C. Kale, R. Wetzel *et al.*, "Modeling missing data in clinical time series with rnns," *Machine Learning for Healthcare*, vol. 56, no. 56, pp. 253–270, 2016.

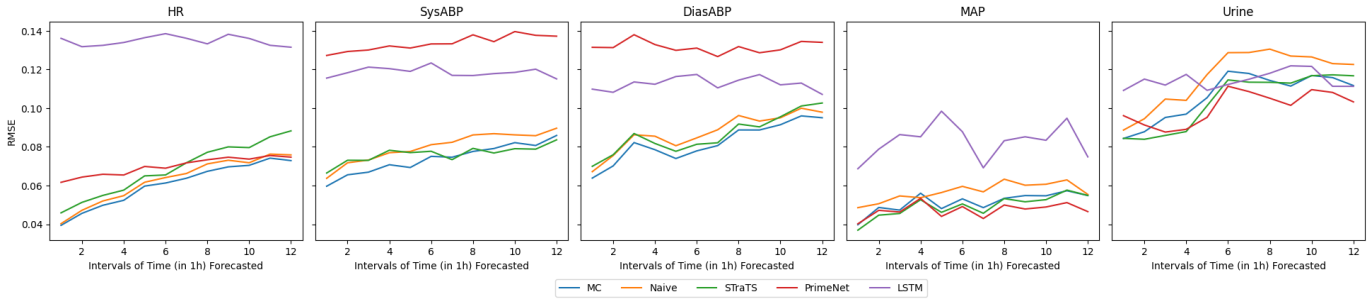
[16] J. J. Dabrowski and A. Rahman, "Sequence-to-sequence imputation of missing sensor data," in *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*. Springer, 2019, pp. 265–276.

[17] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "Lstm-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

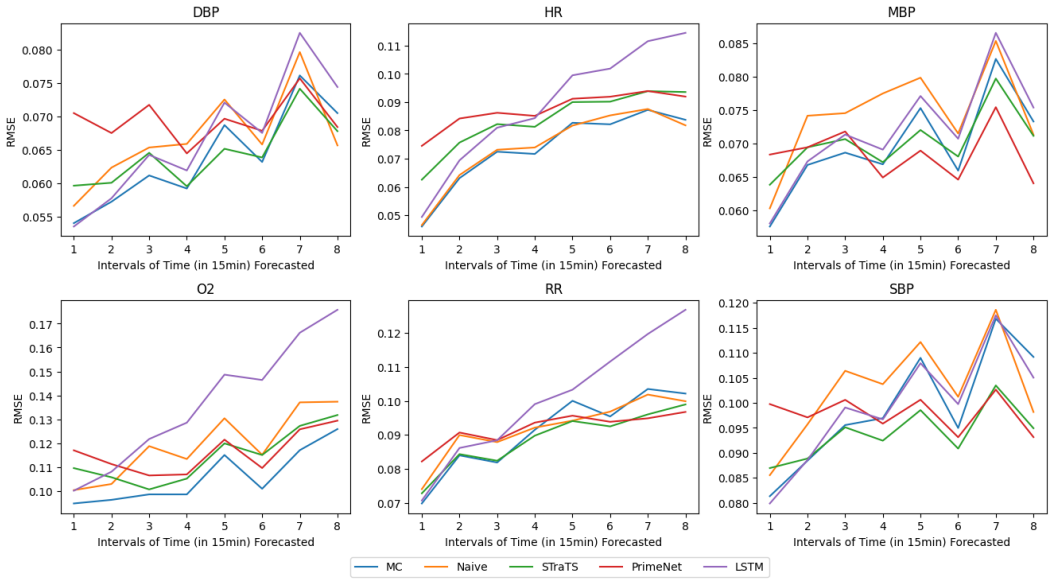
[18] J. Zhou and Z. Huang, "Recover missing sensor data with iterative imputing network," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[19] Y.-F. Zhang, P. J. Thorburn, W. Xiang, and P. Fitch, "Ssim—a deep learning approach for recovering missing time series sensor data," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6618–6628, 2019.

[20] Y. Bengio and F. Gingras, "Recurrent neural networks for missing or asynchronous data," *Advances in neural information processing systems*, vol. 8, 1995.



(a) Physionet



(b) MIMIC-III

Fig. 4: Subplots showing the average root mean squared error (RMSE) of each forecasting method as forecasting interval increases for each output variable. The blue, orange, green, red, and purple line graphs depict the performances of the Markov chain (MC), naive, STraTS, PrimeNet, and LSTM models respectively.

[21] M. Lepot, J.-B. Aubin, and F. H. Clemens, "Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment," *Water*, vol. 9, no. 10, p. 796, 2017.

[22] F. Cismondi, A. S. Fialho, S. M. Vieira, S. R. Reti, J. M. Sousa, and S. N. Finkelstein, "Missing data in medical databases: Impute, delete or classify?" *Artificial intelligence in medicine*, vol. 58, no. 1, pp. 63–72, 2013.

[23] V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt, "Gp-vae: Deep probabilistic time series imputation," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 1651–1661.

[24] Z. Lu, T. K. Leen, Y. Huang, and D. Erdogmus, "A reproducing kernel hilbert space framework for pairwise time series distances," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 624–631.

[25] S. C.-X. Li and B. M. Marlin, "Classification of sparse and irregularly sampled time series with mixtures of expected gaussian kernels and random features." in *UAI*, 2015, pp. 484–493.

[26] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[27] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 65–74.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] R. R. Chowdhury, J. Li, X. Zhang, D. Hong, R. K. Gupta, and J. Shang, "Primenet: Pre-training for irregular multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[30] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160035, May 2016. [Online]. Available: https://doi.org/10.1038/sdata.2016.35

[31] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012," in *2012 Computing in Cardiology*. IEEE, 2012, pp. 245–248.

[32] S. N. Shukla and B. M. Marlin, "Multi-time attention networks for irregularly sampled time series," *arXiv preprint arXiv:2101.10318*, 2021.

[33] J. R. Norris, *Markov chains*. Cambridge university press, 1998, no. 2.

[34] L. Wan, W. Lou, E. Abner, and R. J. Kryscio, "A comparison of time-homogeneous markov chain and markov process multi-state models," *Communications in Statistics: Case Studies, Data Analysis and Applications*, vol. 2, no. 3-4, pp. 92–100, 2016.

[35] W. K. Newey, "Adaptive estimation of regression models via moment restrictions," *Journal of Econometrics*, vol. 38, no. 3, pp. 301–339, 1988.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019.

[37] A. Johnson, T. Pollard, and R. Mark, "Mimic-iii clinical database (version 1.4)," *PhysioNet*, vol. 10, no. C2XW26, p. 2, 2016.