BRITS : Bidirectional Recurrent Imputation for Time Series

DAHS
석사과정 김홍범

Part 1

Introduction

Part 1

Introduction

기존의 **time series imputation methods**들은 **Linear dynamics**등의 여러 가정이 필요하다는 한계점이 존재함

하지만 저자가 제안하는 **BRITS**는 **3**가지의 장점을 가지고 있음

1. 서로 연관성을 가지는 시계열 결측치 데이터를 다루는데 특화되어 있음
2. 시계열 데이터를 특정 가정이나 분포로 가정하지 않음
3. 데이터 기반의 결측치 처리 방법을 제안하고, 일반적인 환경에서 결측치 처리가 가능함

기존 연구의 한계점

1. 결측치를 통계 혹은 **ml** 기반으로 고치는 방향으로 접근함
2. 대부분의 연구들에서 결측치에 대한 강한 가정을 필요로 함(선형성, 통계 자료, **low-rankness** 등)

Part 1

Introduction

Technical contribution

- 결측치를 채우기 위해 **Bi-RNN** 모델을 사용함, 특별한 결측치 가정을 하지 않고 일반적 사용이 가능함
- 결측치를 변수로 가정함으로써 좀 더 정확한 **loss** 계산 및 정확도를 높임
- 기존 **RNN**의 **Vanishing gradient**를 방지하기 위하여 **bi-rnn**을 사용하였고 이를 통한 **delayed gradient**의 장점이 있음
- 결측치 처리와 **classification/regression**을 작업을 동시에 진행함으로써 **backpropagation**의 오류를 줄임 **(Multi-task learning algorithm)**
- **Real world** 데이터셋인 **air quality, health-care, human activity dataset**을 사용함으로써 **imputation** 및 **classification/regression** 모두에서 **SOTA**를 달성함

Part 2

Related Work

Part 2

Related work

MICE

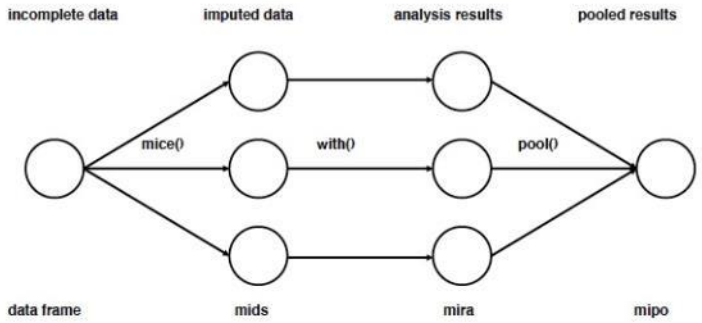


Figure 1: Main steps used in multiple imputation.

M-RNN

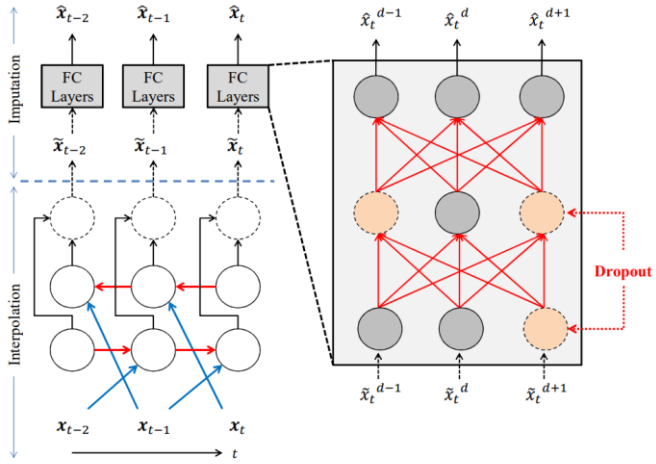


Figure 2: M-RNN Architecture. (Dropout is used for multiple imputations)

SAITS

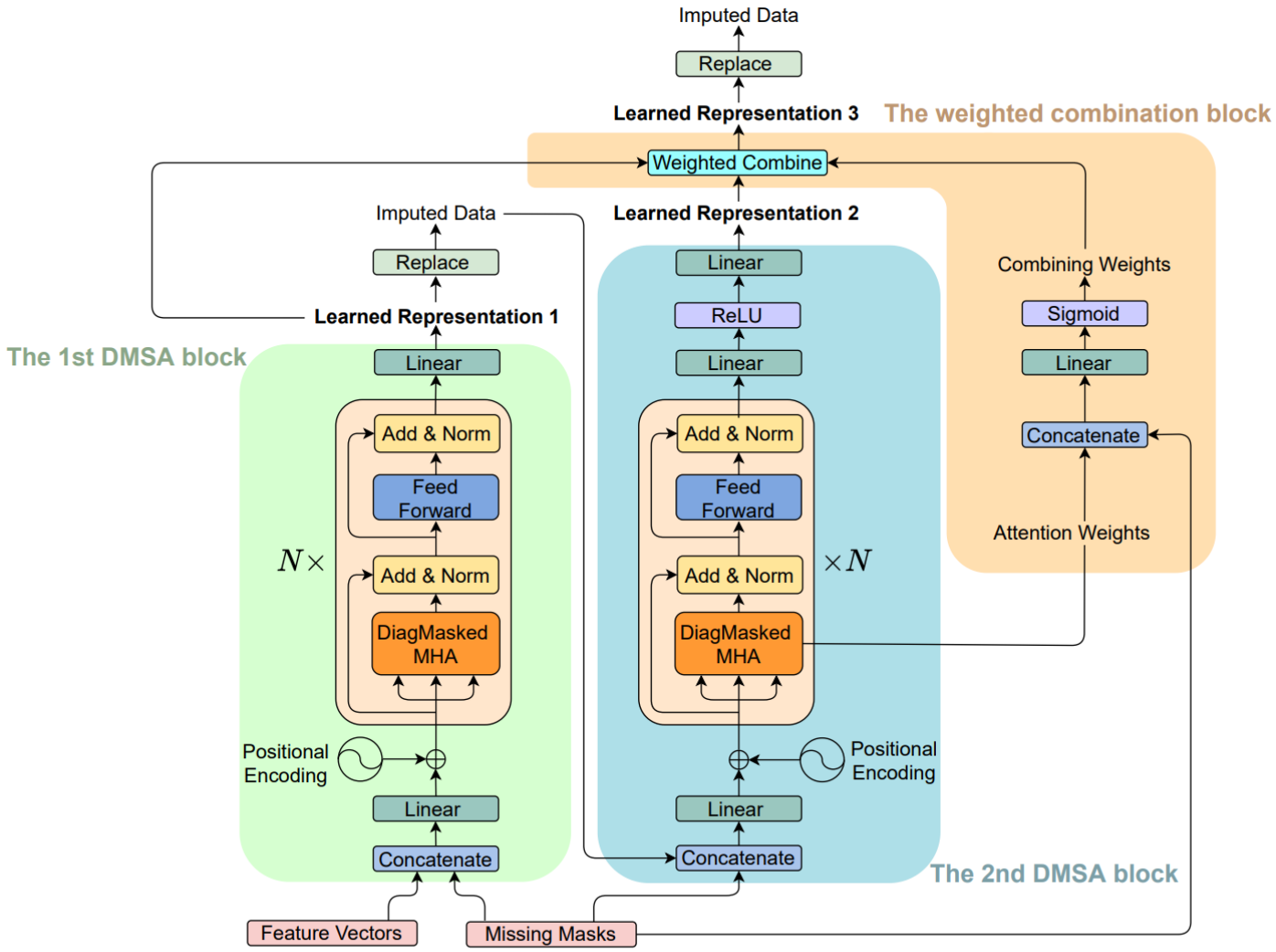


Figure 3: The SAITS model architecture.

Part 3

BRITS

Part 3

Preliminary

Definition 1 (Multivariate Time Series)

- X_T = sequence of T Observation data
- M_T = masking vector
- s_t = time gap between different timestamps
- δ_t^d = time gap from the last observation to the current timestamp

$$\mathbf{m}_t^d = \begin{cases} 0 & \text{if } x_t^d \text{ is not observed} \\ 1 & \text{otherwise} \end{cases}$$

$$\delta_t^d = \begin{cases} s_t - s_{t-1} + \delta_{t-1}^d & \text{if } t > 1, \mathbf{m}_{t-1}^d = 0 \\ s_t - s_{t-1} & \text{if } t > 1, \mathbf{m}_{t-1}^d = 1 \\ 0 & \text{if } t = 1 \end{cases}$$

Example

time series $\mathbf{X}$

31	/	/	32	27	22	
6	17	/	/	/	13	
/	107	/	87	66	90	
$\mathbf{x}_1$	$\mathbf{x}_2$	.....			$\mathbf{x}_6$	

masking vectors

1	0	0	1	1	1	
1	1	0	0	0	1	
0	1	0	1	1	1	
$\mathbf{m}_1$	$\mathbf{m}_2$	.....			$\mathbf{m}_6$	

time gaps

0	2	7	9	5	1	$d=1$
0	2	5	7	12	13	$d=2$
0	2	5	7	5	1	$d=3$
δ_1	δ_2	.....			δ_6	

$s_{1...6}$= 0, 2, 7, 9, 14, 15

Part 3

RITS-I

가장 단순하게 기본적인 **RNN**을 가정하고 **Imputation**을 진행해봄

- **RITS-I = Unidirectional Uncorrelated Recurrent Imputation**

Assumptions

1. **t**번째 **time step**에서 변수(데이터)들이 상관관계가 없음
2. 실제 값(x_t)은 그대로 검증을 위해 사용
3. 결측치들의 경우 **RNN**을 통해 나온 추정값($\hat{x}_t$)으로 대체하고, 미래의 관측값으로 검증

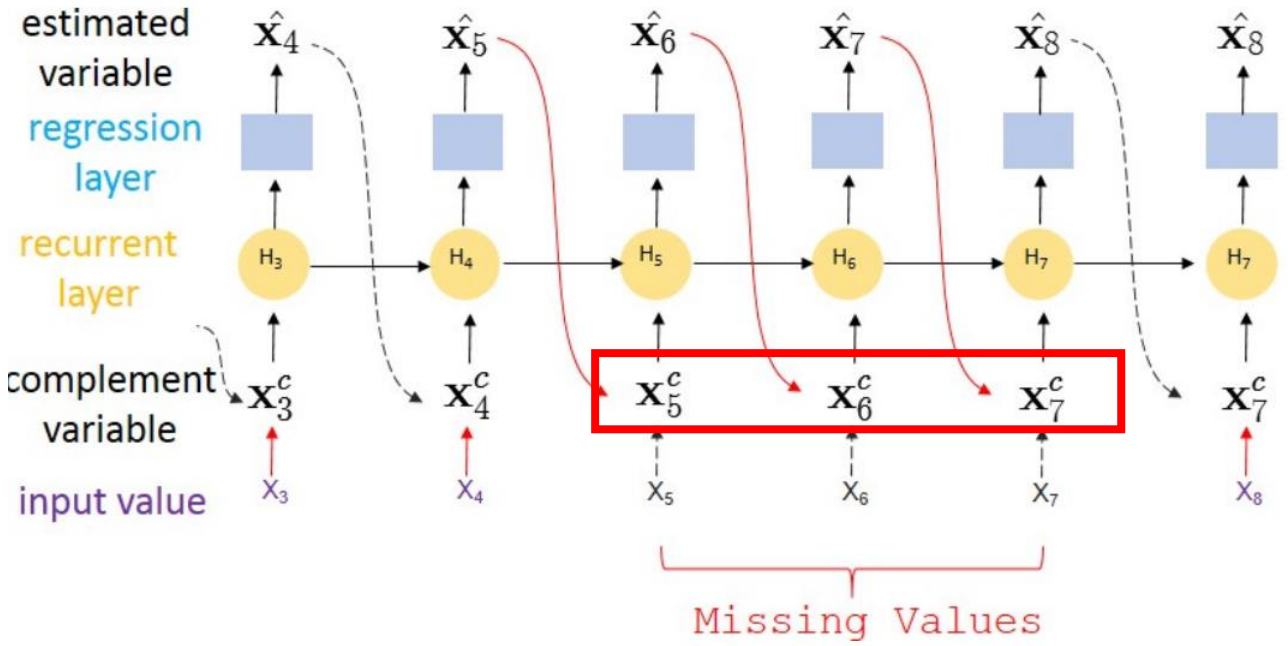


Figure 2: Imputation with unidirectional dynamics.

Part 3

RITS-I

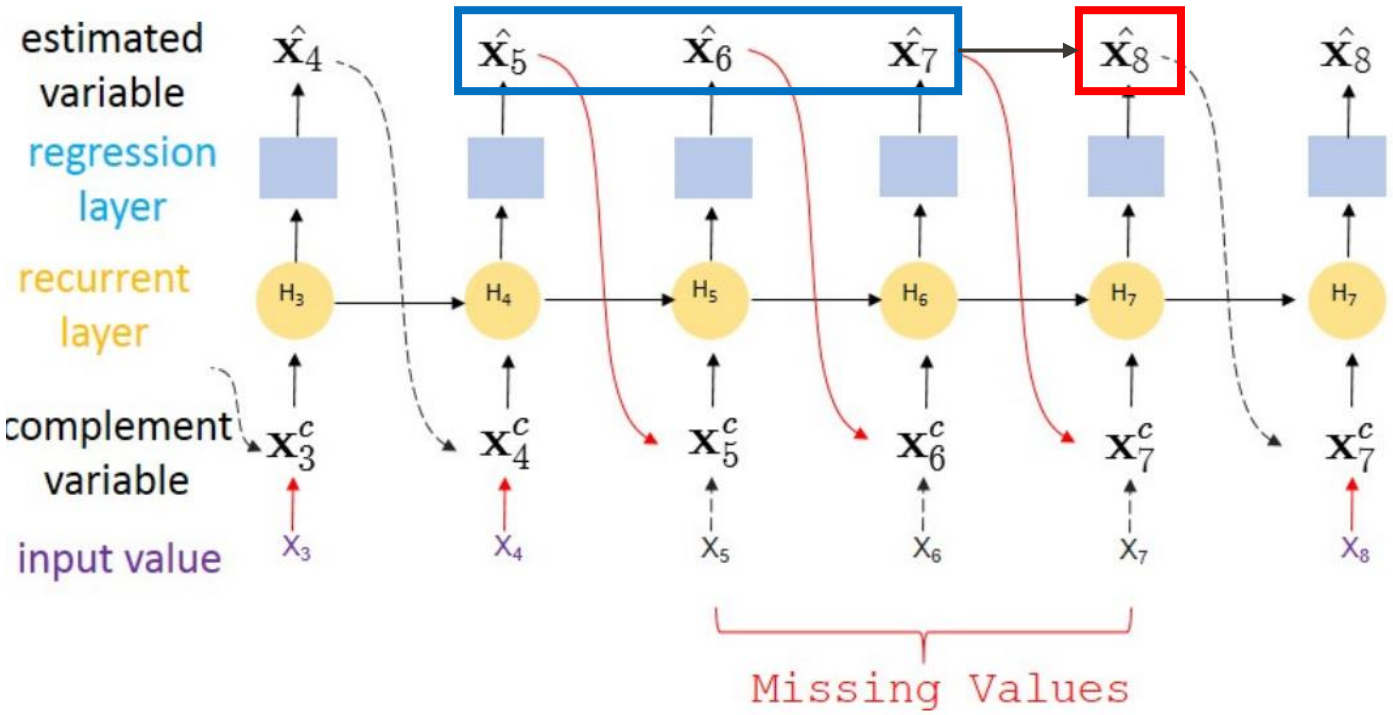


Figure 2: Imputation with unidirectional dynamics.

Estimation error(t_1 ~ t_4 , t_8) : loss function(loss(x, $\hat{x}$))
Estimation error(t_5 ~ t_7) : $\hat{x}_5$ ~ $\hat{x}_7$의 값들은 $\hat{x}_8$에 depend함
-> "delayed" error를 8번째에서 계산 가능함

Part 3

RITS-I

Algorithm

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{U}_h \mathbf{x}_t + \mathbf{b}_h),$$ ⟶ 기존 **RNN**과 동일

σ : **sigmoid function,** W_h, U_h, b_h : **parameters** h_t, : **hidden state of previous time steps**

Equation

$$\begin{aligned}
\hat{\mathbf{x}}_t &= \mathbf{W}_x \mathbf{h}_{t-1} + \mathbf{b}_x, && (1)\\
\mathbf{x}_t^c &= \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{x}}_t, && (2)\\
\gamma_t &= \exp\{-\max(0, \mathbf{W}_\gamma \delta_t + \mathbf{b}_\gamma)\}, && (3)\\
\mathbf{h}_t &= \sigma(\mathbf{W}_h [\mathbf{h}_{t-1} \odot \gamma_t] + \mathbf{U}_h [\mathbf{x}_t^c \circ \mathbf{m}_t] + \mathbf{b}_h), && (4)\\
\ell_t &= \langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) \rangle . && (5)
\end{aligned}$$

Part 3

RITS-I

Equation

$$\hat{\mathbf{x}}_t = \mathbf{W}_x \mathbf{h}_{t-1} + \mathbf{b}_x, \quad (1)$$

Hidden state를 추정값($\hat{x}_t$)으로 대체하는 **regression component**임

$$\mathbf{x}_t^c = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{x}}_t, \quad (2)$$

결측값을 **(1)**에서 구한 값으로 대체함

$$\gamma_t = \exp\{-\max(0, \mathbf{W}_\gamma \delta_t + \mathbf{b}_\gamma)\}, \quad (3)$$

앞서 정의한 델타 값을 이용한 **missing pattern**으로서 **hidden state**를 줄여주는 역할을 하며, 추후 결측치의 패턴을 파악하는데 도움을 줌

$$\mathbf{h}_t = \sigma(\mathbf{W}_h[\mathbf{h}_{t-1} \odot \gamma_t] + \mathbf{U}_h[\mathbf{x}_t^c \circ \mathbf{m}_t] + \mathbf{b}_h), \quad (4)$$

(3)의 값을 이용하여 다음 **hidden state**를 예측함

$$\ell_t = \langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) \rangle . \quad (5)$$

최종적으로 추정값과 실제값을 이용하여 **loss**를 계산함, L_e **= mean absolute error**

Part 3

RITS-I

Key point : RNN 이후 **classification/regression**을 추가로 진행함으로써 **loss**값을 계산함

$$\hat{\mathbf{y}} = f_{out}(\sum_{i=1}^{T} \alpha_i \mathbf{h}_i),$$

Accumulated loss

Min $\frac{1}{T}\sum_{t=1}^{T} \ell_t + \mathcal{L}_{out}(\mathbf{y}, \hat{\mathbf{y}})$, $\alpha_i = \frac{1}{T}$

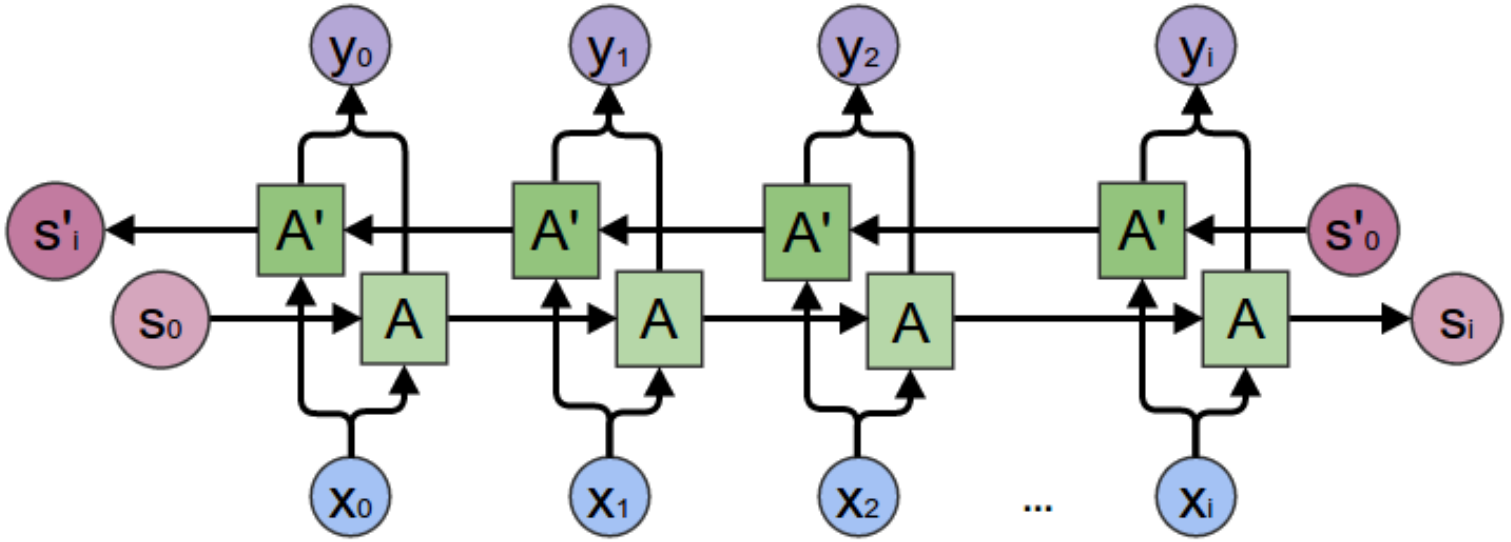
Original loss

Output loss(specific task)

Part 3

BRITS-I

RITS-I의 단점
결측치의 추정된 에러는 다음 **Epoch**전까지는 수정되지 않는다.
-> 모델의 수렴을 느리게 하고, 학습의 비효율을 불러일으킴 또한 **bias exploding** 문제를 일으킴



RITS-I에 **Bi-rnn**
구조만 추가함

해당 문제를 **bi-rnn**구조를 통해 해결함 -> **variable**은 **forward direction**뿐만 아니라 **backward direction** 두가지의 동시 영향을 받는다.

$$\ell_t^{cons} = \text{Discrepancy}(\hat{\mathbf{x}}_t, \hat{\mathbf{x}}'_t)$$

Loss = forward loss + backward loss + consistency loss

Part 3

BRITS-I

BRITS-I의 단점

RITS-U, BRITS-I : 같은 **Time** 시점에서 관측치들이 서로 **uncorrelated**하다고 가정을 하고 진행한다.
하지만 실제 **real world**에서는 **feature간에 correlation**이 존재하고 이를 반영해야 한다.

BRITS-I

$\hat{x}_t$ -> history - based estimation

BRITS

$\hat{z}_t$ -> feature - based estimation

LOSS

$$\ell_t = \langle \mathbf{m}_t, \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) \rangle$$

Loss

$$\ell_t = \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) + \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{z}}_t) + \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{c}}_t).$$

Metric

$$\mathtt{MAE} = \frac{\sum_i |\mathtt{pred}_i - \mathtt{label}_i|}{N}, \quad \mathtt{MRE} = \frac{\sum_i |\mathtt{pred}_i - \mathtt{label}_i|}{\sum_i |\mathtt{label}_i|}.$$

Part 3

BRITS

BRITS의 Equation

$$\hat{\mathbf{z}}_t = \mathbf{W}_z \mathbf{x}_t^c + \mathbf{b}_z, \quad (7)$$

$W_Z, b_Z \rightarrow correspoding\ parameters$

$$\beta_t = \sigma(\mathbf{W}_\beta [\gamma_t \circ \mathbf{m}_t] + \mathbf{b}_\beta) \quad (8)$$

마스킹 여부(**temporal decay** 동시 고려)

$$\hat{\mathbf{c}}_t = \beta_\mathbf{t} \odot \hat{\mathbf{z}}_t + (1 - \beta_\mathbf{t}) \odot \hat{\mathbf{x}}_t. \quad (9)$$

추정값 대체 여부 판단

$$\mathbf{c}_t^c = \mathbf{m}_t \odot \mathbf{x}_t + (1 - \mathbf{m}_t) \odot \hat{\mathbf{c}}_t \quad (10)$$

$$\mathbf{h}_t = \sigma(\mathbf{W}_h [\mathbf{h}_{t-1} \odot \gamma_t] + \mathbf{U}_h [\mathbf{c}_t^c \circ \mathbf{m}_t] + \mathbf{b}_h). \quad (11)$$

Hidden state 추정

BRITS의 loss

$$\ell_t = \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{x}}_t) + \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{z}}_t) + \mathcal{L}_e(\mathbf{x}_t, \hat{\mathbf{c}}_t).$$

Loss - (first step)

Loss - (second step)

Loss - (Third step)

Part 4

Experiments

Part 4

Dataset

Air Quality Data

- **2014/05/01 ~ 2015/04/30 :** 베이징시의 공기 오염도를 측정함**(13.3% missing value)**
- **Test data : 3,6,9**월
- **Training data :** 나머지 데이터 사용

Health care Data

- **MIMICE**의 **ICU Data** 사용 **: 35 feature (up to 78% missing value)**
- **Data: first 48 hours after patient's admission to ICU**
- **Imputation** 뿐만 아니라 **classification(death prediction)**을 동시에 진행함

Human Activity data

- **Walking, falling, sitting down** 등 **11**가지의 활동 데이터**(5**명**)**
- **4**가지의 센서들**(**왼**,** 오른 발목 등**)**에서 데이터를 수집함**(30,917 time series data, 10% missing value)**
- **Imputation + classification task**를 동시에 진행함
- 랜덤하게 **10% ground truth**를 생성한 후에 **imputation** 성능 측정을 진행함

Part 4

Results

Table 1: Performance Comparison for Imputation Tasks (in MAE(MRE%))

Method		Air Quality	Health-care	Human Activity
Non-RNN	Mean	55.51 (77.97%)	0.461 (65.61%)	0.767 (96.43%)
	KNN	29.79 (41.85%)	0.367 (52.15%)	0.479 (58.54%)
	MF	27.94 (39.25%)	0.468 (67.97%)	0.879 (110.44%)
	MICE	27.42 (38.52%)	0.510 (72.5%)	0.477 (57.94%)
	ImputeTS	19.58 (27.51%)	0.390 (54.2%)	0.363 (45.65%)
	STMVL	12.12 (17.40%)	/	/
RNN	GRU-D	/	0.559 (77.58%)	0.558 (70.05%)
	M-RNN	14.05 (20.16%)	0.445 (61.87%)	0.248 (31.19%)
Ours	RITS-I	12.45 (17.93%)	0.385 (53.41%)	0.240 (30.10%)
	BRITS-I	11.58 (16.66%)	0.361 (50.01%)	0.220 (27.61%)
	RITS	12.19 (17.54%)	0.292 (40.82%)	0.248 (31.21%)
	BRITS	**11.56 (16.65%)**	**0.278 (38.72%)**	**0.219 (27.59%)**

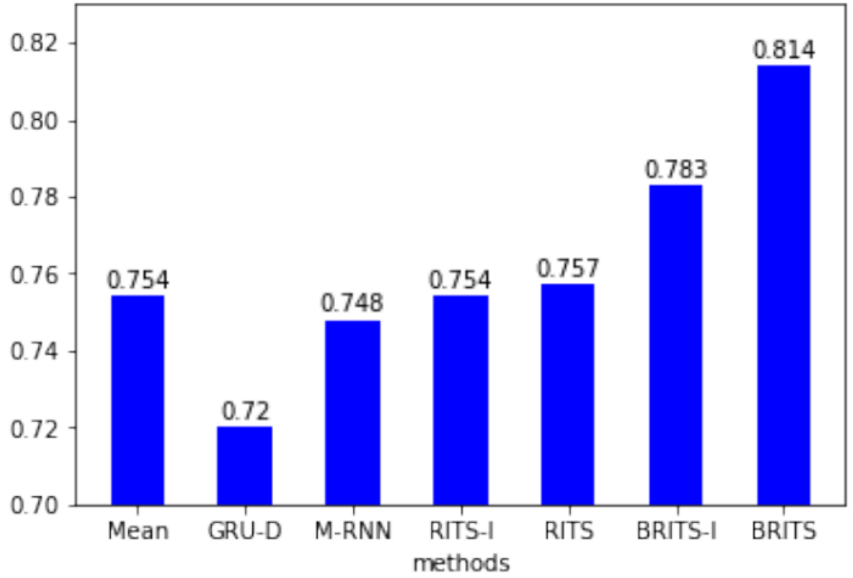
Evaluation : 10% non missing values를 validation data로 사용

Part 4

Results

Table 2: Performance Comparison for Classification Tasks

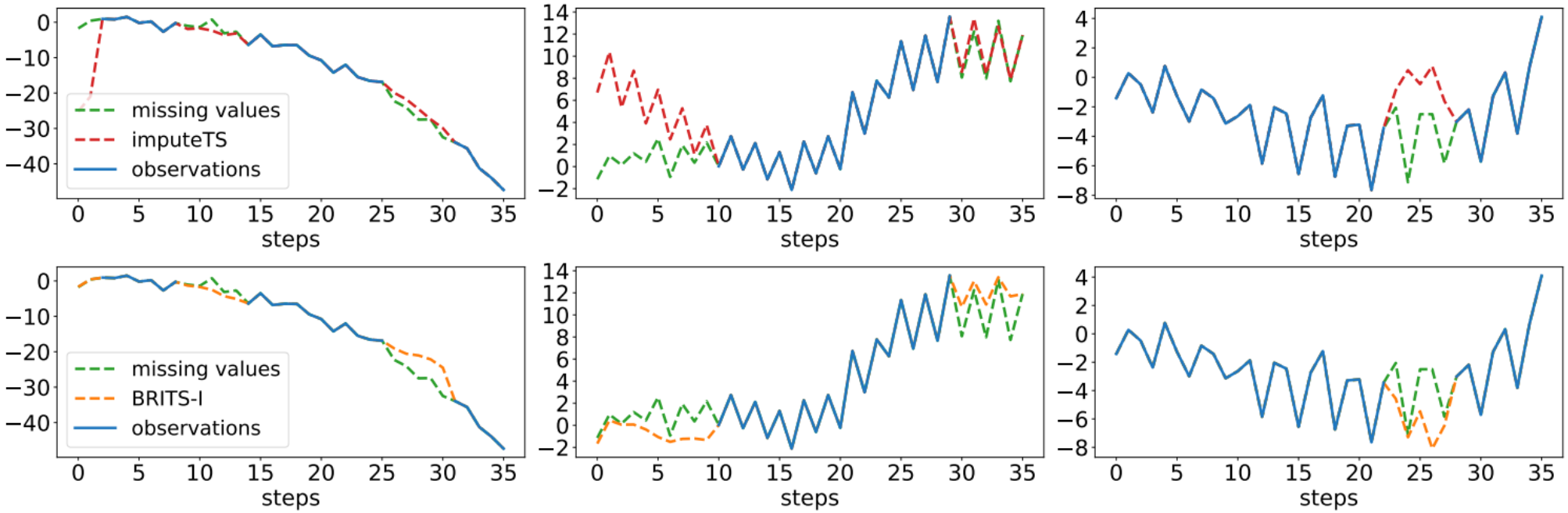
Method	Health-care (AUC)	Human Activity (Accuracy)
GRU-D	0.834 ± 0.002	0.940 ± 0.010
M-RNN	0.817 ± 0.003	0.938 ± 0.010
RITS-I	0.821 ± 0.007	0.934 ± 0.008
BRITS-I	0.831 ± 0.003	0.940 ± 0.012
RITS	0.840 ± 0.004	0.968 ± 0.010
BRITS	$\mathbf{0.850 \pm 0.002}$	$\mathbf{0.969 \pm 0.008}$



Part 4

Results

Appendix



- 가상의 데이터 생성 후 추가 성능 검증 진행

Part 4

Conclusion

1. 특정 분포를 가정하지 않고 **bi-rnn** 구조를 가지고 **missing value**를 학습 후 대체한다는 장점을 가지고 있는 모델임
2. **Bi-rnn** 구조에서 **missing value**를 **constant**가 아닌 **variables**로 취급함으로써 **delayed gradients** 효과를 불러일으키고 정확도를 상승시키는 효과를 냄
3. **Imputation**과 **classification/regression task**를 동시에 진행함으로써 각각의 **task**에서 높은 **sota** 성능을 보임

의문점

1. **Imputation**과 **classification/regression**을 동시에 진행하였는데, 해당 방법론에 대한 명확한 근거가 제시되어 있지 않고, **SOTA** 성능을 동시 달성했다고만 언급됨

References

https://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66
http://arxiv.org/pdf/2202.08516v2.pdf (SAITS)
https://arxiv.org/pdf/1805.10572v1.pdf (BRITS)
https://arxiv.org/pdf/1711.08742v1.pdf (M-RNN)
https://onlinelibrary.wiley.com/doi/epdf/10.1002/sim.4067 (MICE)

THANK YOU