

# Reproducibility Project Instructions for CS598 DL4H in Spring 2022

Author1 and Author2

{satta2, bbath2}@illinois.edu

Group ID: X

Paper ID: 242, Difficulty: [hard]

Paper Link : <https://arxiv.org/pdf/1911.12216v1.pdf>

Presentation link: [https://www.youtube.com/watch?v=YbLWf1CET\\_8](https://www.youtube.com/watch?v=YbLWf1CET_8)

Code link: [https://github.com/z29/cs598\\_final\\_project](https://github.com/z29/cs598_final_project)

Power point: [https://github.com/z29/cs598\\_final\\_project/blob/master/project\\_presentation.pptx](https://github.com/z29/cs598_final_project/blob/master/project_presentation.pptx)

## 1 Introduction

Predicting the patient's clinical outcome from the historical electronic medical records (EMR) is a fundamental research problem in medical informatics. Most deep learning-based solutions for EMR analysis concentrate on learning the clinical visit embedding and exploring the relations between visits. Although those works have shown superior performances in healthcare prediction, they fail to explore the personal characteristics during the clinical visits thoroughly. Moreover, existing works usually assume that the more recent record weights more in the prediction, but this assumption is not suitable for all conditions. In this paper, we propose ConCare to handle the irregular EMR data and extract feature interrelationship to perform individualized healthcare prediction. Our solution can embed the feature sequences separately by modeling the time-aware distribution. ConCare further improves the multi-head self-attention via the crosshead decorrelation, so that the interdependencies among dynamic features and static baseline information can be effectively captured to form the personal health context. Experimental results on two real-world EMR datasets demonstrate the effectiveness of ConCare. The medical findings extracted by ConCare are also empirically confirmed by human experts and medical literature. Particularly, there are two research challenges with this study. 1) Extracting Personal health context 2) Capturing the impact of the Time Interval. To address 1, we can extract interdependencies between clinical features to learn personal health context and regenerate the feature embedding under the context, by multi-head self attention mechanism. To address 2, paper proposes a multi-channel feature embedding architecture, which learns the representation of different feature sequences via separate GRUs, and adaptively captures the effect

of time intervals between records of each feature by time-aware attention

## 2 Scope of reproducibility

We aim to conduct the mortality prediction task on real-world dataset MIMIC-III dataset and will compare it with other baseline approaches. For example, (Lee et al. 2018) proposes a medical context attention-based RNN that utilizes the derived individual information from conditional variational autoencoders. However, none of them explore the interdependencies between static baseline information and dynamic records from a global view. The proposed model in this paper, ConCare, can adaptively capture the relationship between clinical features and perform individualized prediction for patients in diverse health contexts.

Based on the MIMIC dataset provided, we should try to run the same study and compare it with at least 2 baseline methods the paper has presented and prove that ConCare outperforms. Paper has presented the comparison with GRU, TLSTM, RETAIN, MCA-RNN baseline methods by comparing AUROC, AUPRC and min(Se, P+)

## 3 Methodology

We aim to re implement what paper has implemented on <https://arxiv.org/pdf/1911.12216v1.pdf>. We will implement the same model with below steps.

### 3.1 Model descriptions

Figure 3 on paper shows the framework of the proposed Concare model. The model treats the clinical information of the patient from the perspective of features rather than visits. The model extracts the context vector of each dynamic feature and static baseline information separately. Such feature embedding vectors are then re-encoded by taking

the information of all features as healthcare context. The framework comprises of the following submodules: 1) The multi-channel time series embedding module with time-aware attention is developed to separately learn the representation of each dynamic feature. 2) The feature encoder is adopted to combine all the static information and dynamic records based on self-attention. The individualized prediction finally is obtained from all regenerated feature embeddings with an attention queried by static baseline information. `r111..rNT` are patient records during different visits

As described in the paper, ConCare follows below steps to implement the model.

1) ConCare Model first embeds the time series of each feature separately by multi-channel GRU and creates a hidden layer.  $h_{n,1}, \dots, h_{n,T} = \text{GRU}(r_{n,1}, \dots, r_{n,T})$

2) Next it defines the attention function by mapping a query and a set of key-value pairs to an output. Output is a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

$q_{\text{emb } n,T} = W_q n \cdot h_{n,T}$   $k_{\text{emb } n,t} = W_k n \cdot h_{n,t}$ , where  $q_{\text{emb } n,T}$  and  $k_{\text{emb } n,t}$  are the Query vector and the Key vector respectively

3) Then we design a time-aware attention function using query and key vector using softmax and tanh function given in paper as below.

$n_{1,1}, n_{1,2}, \dots, n_{1,T} = \text{Softmax}(n_{1,1}, n_{1,2}, \dots, n_{1,T})$   $n_{1,t} = \tanh(q_{\text{emb } n,T} \cdot k_{\text{emb } n,t} \cdot \log(e + (1(q_{\text{emb } n,T} \cdot k_{\text{emb } n,t})) \cdot t))$   $t$  - time interval since last latest record - sigmoid

function  $n$  - a featurespecific learnable parameter trained to control the influence of the time interval on the corresponding feature.

4) Finally, based on the learned weights, we can derive time aware contextual feature representation as  $fn = \sum_{i=1}^T n_{i,t} \cdot h_{n,t}$ .

5) Furthermore, the demographic baseline data is embedded into the same hidden space of  $fn$

$f_{\text{base}} = W_{\text{emb base}} \cdot \text{base}$  where  $W_{\text{emb base}}$  is an embedding matrix. Thus,

all the data of the patient can be represented by a matrix  $F$  (i.e., a sequence of vectors, where each vector represents one feature of the patient over time):  $F = (f_1, \dots, f_N, f_{\text{base}})$

6) We capture the interdependencies among dynamic features through visits as well as static baseline information, and further re-encode the feature

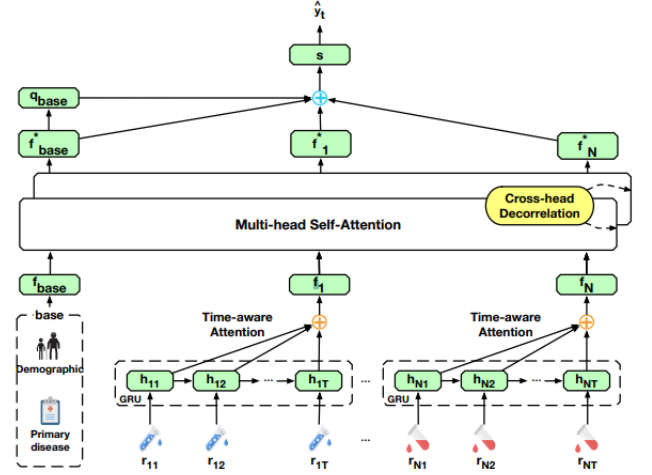


Figure 3: The Framework of ConCare.

embedding under the personal context based on self-attention. As the ConCare processes each feature, self-attention allows it to look at other features for clues that can help lead to a better encoding for this feature.. Mathematically, given current feature representations  $F$ , the refined new representations are calculated as:  $un = \text{MultiHeadAttention}(F) = [\text{head1}(fn) \text{ head2}(fn) \dots \text{headm}(fn)]WO$ ,  $\text{headm}$  is  $m$ -th attention head, is the concatenation operation  $WO$  is a linear projection matrix.

7)Heads for self-attention are expected to capture dependencies from different aspects. However,in practice, heads may tend to learn similar dependencies. To overcome this challenge, we encourage diverse or non-redundant representations by minimizing the cross-covariance of hidden activations across different heads. We utilize the crosshead decorrelation module to expand the models ability to focus on different features.

8) The final loss can be denoted as the combination of cross-entropy loss and decorrelation loss. We are using the code provided on <https://github.com/AccountableMachineIntelligence/ConCare> and able to run for subset of the data currently

### 3.2 Data descriptions

As per the reference , we have taken MIMIC benchmark data from <https://github.com/YerevaNN/mimic3-benchmarks/>.

We prepared the time series data as mentioned by the steps on the link. Data generation part generates below types of data as below.

1) Time series of events are stored in episode-timeseries.csv which counts distinct episodes 2) episode-level information (patient age, gender, ethnicity, height, weight) and outcomes (mortality, length of stay, diagnoses) are stores in episode.csv. Patients Demographic data is also available on the same link.

<https://drive.google.com/file/d/1TXn4UdtQCzfd7TdDJAo6IcnO2LUa1a/view>

### 3.3 Hyperparameters

We used below hyper parameters to generate the base and concare model.

Paramter	BaseLine	ConCare
epoch	28	28 and 100
batch size	8	256
dropout	0.8	0.1
dim(hidden layer)	16	64
time step	1	1

For concare we used epoch 100 and 28 both. Base model works best for 28 epoch as per documentation.

### 3.4 Implementation

. We found below Concare Model code and data generation code which we installed and ran to reproduce the results.

Concare model code	<a href="https://github.com/Accountable-Machine-Intelligence/ConCare">https://github.com/Accountable-Machine-Intelligence/ConCare</a>
Data generation code	<a href="https://github.com/YerevaNN/mimic3benchmarks">https://github.com/YerevaNN/mimic3benchmarks</a>

We generated MIMIC III Data based on the guidelines provided and Readme.MD provides all instructions to set up environment, installation and execution. Data and Readme.md file is available in git repo link provided at top.

data	/mimic3-benchmarks-master/data
Execution	README.md

We implemented all baseline methods by modifying existing code and implemented Logistic, RETAIN, GRU and TLSTM baseline models with code as below.

We wrote down the code to generate baseline models as below which is available at below path . We modified exisiting LSTM method to add attention for TLSTM. We wrote the code for GRU and RETAIN.

Model	Code	Output
Retain	mimic3models/retain.py	Retain <sub>output.txt</sub>
GRU	mimic3models/grualpha.py	gru <sub>28epoch.txt</sub>
TLSTM	mimic3models/tlstm.py	lstm <sub>28epoch.txt</sub>
ConCare	concare-notebook.ipynb	concareoutput.txt

### 3.5 Computational requirements

Current paper has used below configuration.

CPU	Intel Xeon E5-2630
RAM	256 GB
GPU	Nvidia Titan

To run 28 epoch, it took significant longer but we could run all models in 2 hours using below configuration.

CPU	Inteli5-7400@3.00GHz 3.00 GHz
RAM	12 GB
28 epochs	2 hours
100 epochs	6 hours

## 4 Results

Based on below table, we can see that TLSTM preforms the best among baseline methods with 0.8625 AUROC. Concare for same epochs is with 0.8566 AUROC but gains and improves to 0.8669 AUROC. As suggested by MIMIC II dataset guidelines, it performs best up to 28 epochs and if you see the all outputs uploaded on git, it starts performing bad as number of epoch increases. On the other hand Concare is keep improving and performs best with area under curve(AUROC), Area precision/recall(AUPRC) and min(se,p+).

Method	AUROC	AUPRC	Min(Se, P+)
GRU	0.8568	0.5493	0.5091
RETAIN	0.8479	0.5077	0.5092
TLSTM	0.8625	0.5599	0.5
Regression	0.8484	0.4744	0.4601
ConCare(28 epoch)	0.8566	0.4985	0.5183
ConCare(100 epoch)	0.8669	0.5736	0.5481

It also matches with paper Concare and other methods output as below.

GRU	0.8628	0.4949	0.5026
RETAIN	0.8313	0.4790	0.4721
TLSTM	0.8617	0.4964	0.4977
MCA <sub>R</sub> NN	0.8587	0.5003	0.4932
Transformer	0.8535	0.4917	0.5000
SAnd	0.8382	0.4545	0.4885
ConCare	0.8702	0.5317	0.5082

Based on this, we can say that Concare performs best with the prediction task as it uses demographic information apart from patient's visit information to create attention model which other methods is not using it.

#### 4.1 Additional results not present in the original paper

Paper did not compare it with plain regression but we tried to run it and it does not perform better than that. we tried to split data based on gender and run it but we could not find any significant difference. We wanted to distribute based on ethnicity and wanted to check if it does matter but we could not complete a split data based on ethnicity within given time period.

## 5 Discussion

We believe that we reproduce the original paper with matching results and compared it successfully with baseline models which we could implement. We could not get a sufficient time to implement MCA RNN and Transformer method but based on the current result which match exactly with number, we can say Concare makes a difference by using static demo graphic information along with Patients time series visits and diagnosis information.

We could not get time to do analysis to study decay rate and cross feature dependency described in paper either.

### 5.1 What was easy

Easy part was paper was well documented and proven. We could understood the topic well. Also as we found code , we did not had to build it from scratch. Data generation part was also provided with detail instruction.

### 5.2 What was difficult

Understanding the current code and running it was difficult. BaseModels is implemented with keras API and we have not learned it in this course. We

had to go through multiple online videos and tutorials to understand it. after that we could modified LSTM.py to add attention to it and create GRU and Retain model.

Also Keras and Tensorflow API are not backword compatible and we had to create multiple environment in Anaconda to run it with compatible versions for given method.

### 5.3 Recommendations for reproducibility

Provided Code with given instructions makes it easy to reproduce but setting up correct environment with compatible version of libraries is the key. If you follow instructions provided in our ReadME.MD file then it would be easier for you to reproduce.

## 6 Communication with original authors

We did not do any communication with original author.

## 7 References

Hands-on Machine Learning with Scikit-Learn Keras and TensorFlow second edition - O'Reilly by Aurelien Geron

Ma, Liantao et al. "ConCare: Personalized Clinical Feature Embedding via Capturing the Healthcare Context." AAAI (2020). url: <https://arxiv.org/abs/1911.12216>