Reproducibility Project Instructions for CS598 DL4H in Spring 2023

Yulin Zhao and Chunyu Liu
{yulin6, chunyu3}@illinois.edu

Group ID: 42
Paper ID: 45
Code link: https://github.com/vanity-lost/Encoder-Decoder-with-Cluster-based-Attention

1 Introduction

The purpose of this project is to conduct research on the paper entitled ”Learning of Cluster-based Feature Importance for Electronic Health Record Time-series” (Aguiar et al., 2022). This paper introduces a novel approach to learning cluster-based feature importance for electronic health record (EHR) time-series data. The authors present the model CAMELOT, which combines time-series K-means with the encoder-decoder network. The efficacy of the model is validated on two real-world EHR datasets, demonstrating its superiority over existing feature selection methods in terms of robustness and interoperability. To gain a better understanding of the model, we intend to reproduce the model from scratch, replicate the experiments, test our hypotheses, and document our findings.

2 Scope of reproducibility

This paper introduces a supervised deep learning model named CAMELOT (Aguiar et al., 2022), which is proposed to classify electronic health record (EHR) data into clinically interpretable phenotypes that are relevant for patient outcome forecasting and trajectory. The performance of CAMELOT is evaluated against other methods using two EHR datasets, HAVEN and MIMIC-IV-ED (Johnson et al., 2021; Goldberger and Stanley, 2000), based on multiple metrics, which reveals that CAMELOT surpasses existing methods in both predictive accuracy and interoperability.

2.1 Addressed claims from the original paper

- Claim 1: As the paper states the CAMELOT achieves 0.72 in AUC, 0.34 in F1-score, 0.36 in Recall, and 0.11 in NMI, we want to reproduce such results to check the superiority of CAMELOT.
- Claim 2: The paper states without cluster loss function ($\beta = 0$), the performance will be worse (0.70 in AUC, 0.24 in F1-score, 0.31 in Recall, and 0.05 in NMI). We want to evaluate the influence of the cluster loss function.
- Claim 3: The paper states without distance and cluster loss function ($\alpha = \beta = 0$), the performance will be worse (0.67 in AUC, 0.30 in F1-score, 0.33 in Recall, and 0.06 in NMI). We want to evaluate the influence of the distance loss function.
- Claim 4: The paper states without the custom attention layer, the performance will be worse (0.65 in AUC, 0.24 in F1-score, 0.30 in Recall, and 0.04 in NMI). We want to evaluate the influence of the custom attention layer.

2.2 Proposed ablations from our own

- Ablation 1: As CAMELOT employs time series K-Means (TSKM) as its clustering algorithm, we would like to investigate if utilizing Gaussian Mixture Models (GMM) instead of K-Means can lead to improved performance.
- Ablation 2: The attention mechanism’s RNN layer in CAMELOT employs LSTM cells. As part of our project, we intend to assess if using GRU layers instead of LSTM in CAMELOT can lead to better performance.
- Ablation 3: The CAMELOT model is a type of encoder-decoder design. One of our goals is to investigate if adding random noise to CAMELOT can enhance its performance by increasing its robustness or accuracy.

3 Methodology

Regarding data preprocessing and the CAMELOT model, we didn’t use the existing codes to reproduce. Instead, we would develop our own

CAMELOT model based on the architecture and logic of existing code using PyTorch and scikit-learn for clustering algorithms.

3.1 Model descriptions

As Figure 1 shows, the architecture of CAMELOT is composed of the encoder, identifier, and predictor. The encoder has several LSTM layers and a custom attention layer called FeatTimeAttention. The identifier and Predictor are both MLPs with feed-forward layers, Sigmoid activation layers, and dropout layers.

During the initialization, the encoder is initialized to produce the latent representations. These latent representations are fed into the K-means model to get K clusters. Then the identifier is also initialized.

$$L_{pred}(y_{true}, y_{pred}) = -\sum_{c=1}^{C} w_c y_{true}^c log(y_{pred}^c) \tag{1}$$

$$L_{dist}(\pi) = -H(\pi_C) \tag{2}$$

$$L_{clus}(C) = -\frac{1}{K(K-1)} \sum_{i,j} ||c_i - c_j||^2 \tag{3}$$

During training, the model updates the parameters of the encoder, identifier, predictor, and latent cluster representation. To improve robustness and train the cluster representations, it introduces different loss functions as Eq. 1, Eq. 2, and Eq. 3. Thus, the encoder and the identifier are trained on $L_{pred} + \alpha L_{dist}$, the predictor is trained on L_{pred}, and the cluster representations are trained on $L_{pred} + \beta L_{clus}$. The encoder and the hidden feed-forward layers of the identifier and predictor also are applied with the mixed L1 and L2 regularizations.

3.2 Data descriptions

The MIMIC-IV-ED dataset is used to evaluate the performance of the CAMELOT, which can be accessed from PhysioNet. In order to maintain consistency and rigor in reproducing the experiments, we have chosen to use the versions of the datasets that are closest in time to the publication of the original paper. The MIMIC-IV-ED dataset a large public database of ED admissions at the Beth Israel Deaconess Medical Center between 2011 and 2019, containing 448,972 ED admissions. After the same preprocessing techniques, we have 7701 samples with 9 features and window size = 10. There are four classes, each of which has 50, 1629, 5874, and 148 samples.

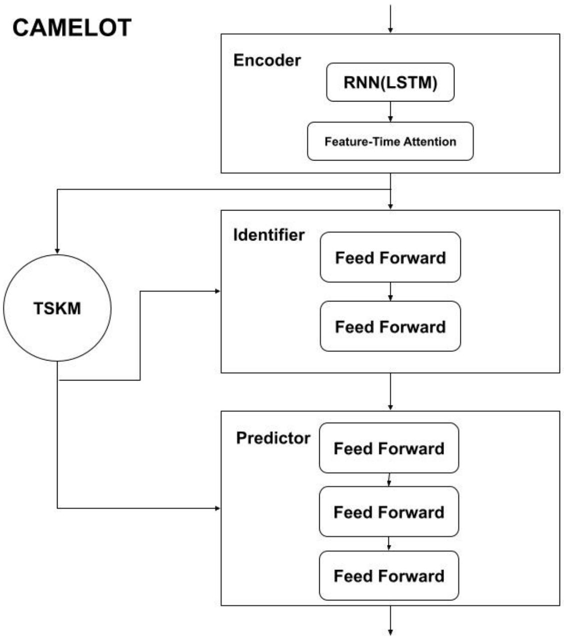

Figure 1: CAMELOT Architecture

3.3 Hyperparameters

The hyperparameters are: *alpha, beta, latent dimension, number of clusters, learning rate, dropout rate, hidden dimension of attention, hidden dimension of MLP, learning rate decay, and batch size*. Some of these hyperparameters are described in Table A.7 of the paper, but their performance is not consistent on our end (i.e. not the best). Thus, we did our grid search to find the best combination of hyperparameters.

3.4 Implementation

For both data preprocessing pipelines and the CAMELOT model, we did NOT reuse the existing codes (i.e. the authors' codes). We wrote our own code for the CAMELOT from scratch using Pytorch and scikit-learn. All the codes are available on our GitHub repo.

3.5 Computational requirements

We developed our model with 1 Intel® Core™ i9-9900K and 1 RTX 2080Ti. For debugging, fine-tuning, and testing phrases, we used 2 Intel(R) Xeon(R) CPU @ 2.20GHz and 2 Tesla P100 GPUs provided by Kaggle.

Table 1: Grid search

(latent_dim,α, β)	AUC
(128, 0.01, 0.001)	0.520996031858463
(128, 0.01, 0.01)	0.5284037417571888
(128, 0.01, 0.005)	0.5284037417571888
(128, 0.05, 0.001)	0.5161637759401114
(128, 0.05, 0.01)	0.5161637759401114
(128, 0.05, 0.005)	0.5161637759401114
(128, 0.005, 0.001)	0.5266069332453946
(128, 0.005, 0.01)	0.5266069332453946
(128, 0.005, 0.005)	0.5266069332453946
(64, 0.01, 0.001)	0.5975889148689542
(64, 0.01, 0.01)	0.5975889148689542
(64, 0.01, 0.005)	0.5975889148689542
(64, 0.05, 0.001)	0.6178344960186388
(64, 0.05, 0.01)	0.6220521540966879
(64, 0.05, 0.005)	0.6220521540966879
(64, 0.005, 0.001)	0.6077363400299416
(64, 0.005, 0.01)	0.6077363400299416
(64, 0.005, 0.005)	0.6077363400299416

4 Results

4.1 Result of addressed claims

4.1.1 Result 1

As Table 1 shows, we did a small grid search on α, β, and latent dimensions. The best combination of these hyperparameters is $\alpha = 0.05, \beta = 0.005, latent\_dim = 64$, achieving 0.62 in AUC, 0.02 in F1-score, 0.34 in Recall, and 0.02 in NMI.

This result is 0.1 lower than the AUC stated in the paper, 0.3 lower than the F1-score stated in the paper, 0.02 lower than the recall stated in the paper, and 0.18 lower than the NMI stated in the paper. This probably shows there are still bugs or malfunctions in our implementation. We can see that current our model performs worse in the F1-score, and thus a possible reason is that the class weight does not perform as desired. We plan to spend another week reviewing and debugging the code, and if needed, we may contact the authors of the paper for help.

4.1.2 Result 2

After removing the cluster loss by setting $\beta = 0$, our model achieves 0.62 in AUC, 0.02 in F1-score, 0.37 in Recall, and 0.02 in NMI.

4.1.3 Result 3

After removing the distance and cluster loss by setting $\alpha = \beta = 0$, our model achieves 0.61 in AUC, 0.02 in F1-score, 0.35 in Recall, and 0.02 in NMI.

4.1.4 Result 4

After removing the attention layer by replacing a feed-forward layer, our model achieves 0.61 in AUC, 0.02 in F1-score, 0.37 in Recall, and 0.02 in NMI.

4.2 Result of proposed ablations

4.2.1 Result 1

TODO

4.2.2 Result 2

TODO

4.2.3 Result 3

TODO

5 Discussion

TODO

5.1 What was easy

TODO

5.2 What was difficult

TODO

5.3 Recommendations for reproducibility

TODO

6 Communication with original authors

TODO

References

Henrique Aguiar, Mauro Santos, Peter Watkinson, and Tingting Zhu. 2022. Learning of cluster-based feature importance for electronic health record time-series. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 161–179. PMLR.

L. Amaral L. Glass J. Hausdorff P. C. Ivanov R. Mark J. E. Mietus G. B. Moody C. K. Peng Goldberger, A. and H. E. Stanley. 2000. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation [Online]*, 101(23):e150–e220.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Leo Anthony Celi, Roger Mark, and Steven Horng. 2021. Mimic-iv-ed.