|  |  |  |
| --- | --- | --- |
| **Fall 2024** | **Report #3 – 09/17/2024** | **Dingyi Nie** |

**——————————————————————————————————————**

**I. Task Achieved Last Week**

**——————————————————————————————————————**

* + Comprehensive experiment with XGBoost classifier on P12
  + Ablation study on input data composition and imputation strategy
  + Compare results with DL SotA, providing baselines for GL

**——————————————————————————————————————**

**II. Feedback and Interaction**

**——————————————————————————————————————**

* **Prof. Kuo’s Feedback**
  + Last week we did experiment on synthetic datasets in attempts to prove the conjecture that XGBoost’s internal ability to handle missing values (NaN’s) can possibly be beneficial for modeling biomedical time series data. Professor Kuo suggested we should further verify the conjecture by doing experiments on more real-world biomedical datasets. He also suggested that if missing patterns are somehow truly meaningful and can be discriminant, we may want to find the balance in the trade-off of performance gain between that brought by imputation and that brought by missing pattern.

**——————————————————————————————————————**

**III. Report**

**——————————————————————————————————————**

**A Brief Overview of PhysioNet 2012 and Its Preparations**

[PhysioNet Challenge 2012 (P12)](https://www.physionet.org/content/challenge-2012/1.0.0/) is a real-world ICU tabular dataset consists of ~12,000 episodes. Each episode patient has 37 time series feature variables (very irregularly sampled), 5 outcome labels and 5 static demographic information fields associated with the patient. The variables generally have high missing rates: the least missing feature has a missing rate of ~23% and 22 out of 37 variables have a > 90% missing rate. It was used as one of the benchmarks by a series of DL time series modeling works, specifically the in-hospital mortality prediction task (binary classification). The SotA of DL methods on the task so far is 85.1% AUROC by [ViTST](https://arxiv.org/abs/2303.12799).

To prepare the dataset, we follow the practice of previous DL works such as [RAINDROP](https://arxiv.org/abs/2110.05357) and ViTST. We group the observations at the same time as a feature vector associated to that time stamp, thus forming a feature array along with the time stamp vector for each episode. We also follow [GRU-D](https://www.nature.com/articles/s41598-018-24271-9) in providing 2 more arrays of the same shape as the feature array: mask (1 for true observations and 0 for missing values) and interval (time since last true observation of the same feature). We then get rid of 12 outlier episodes, split the dataset into 8:1:1 train-valid-test ratio and calculate the mean and standard deviation on train set. We drop 1 specific feature (*MechVent*) which has 0 variation. Next, we use the stats info on the train set to normalize all feature arrays in all 3 splits so that globally all features have a mean of 0 and standard deviation of 1. We take the in-hospital mortality label as our supervision label. Finally, we pad all arrays in the beginning to have the same sequence length.

This gives us 9590 episodes in train set, 1199 episodes in valid and test sets, with the remaining 36 feature columns and a maximum sequence length of 215. Hence, in train set, the unified feature array itself is (9590, 215, 36) in shape.

**XGBoost Classifier on P12**

We experiment a simple XGBoost classifier (100 estimators, max depth=3, learning rate=0.1, log loss) on different combinations of the 3 input components (same definition as in GRU-D): variables **x**, masks **m** and time intervals **d**. To feed the data into the XGBoost, the input arrays are flattened to have 2 axes. This time the data is not imputed, all missing values are still there represented by NaN, so we are basically relying on XGBoost’s internal sparsity design. We fit the classifier on train set and report the AUROC on valid set.

The results are show in table 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AUROC | x | x-m-d | x-m | x-d | m-d |
| No imputation | 80.65% | 80.98% | 80.92% | 81.15% | 69.05% |

Table 1. XGBoost on raw P12 data.

The results are as expected. Since the missing values in x are still NaN’s, m is actually redundant. The best performance is on the combination x-d (variables + time intervals) which is 81.15% in AUROC.

Next, we experimented the same classifier on the same task, same settings. But this time the x part of the input is imputed with one of the 3 imputers: zero, forward, linear. Note that x is normalized to mean=0, so a zero imputer is equal to a mean imputer. The results are shown in table 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AUROC | x | x-m-d | x-m | x-d | m-d |
| Zero-imputer | 80.92% | 80.46% | 81.30% | 81.84% | - |
| Forward-imputer | 86.35% | 86.10% | 86.54% | **86.92**% | - |
| Linear-imputer | 85.98% | 86.15% | 86.52% | 86.12% | - |

Table 2. XGBoost on imputed P12 data.

The results are also interesting. We know that m and d are not completely independent, because m can basically be derived from d (once a new true observation comes in, at next time step the time interval from last observation will be reset to a small value). In this case the x-d combination with forward imputer achieves a 86.92% AUROC on valid set, which is the highest. In fact, if we compare the performance with DL models reported by ViTST (same settings), it outperforms every single model reported. (Actually, this is not even fair because in ViTST itself and most baselines have utilized the static demographic data by embedding them via a text embedding model.)

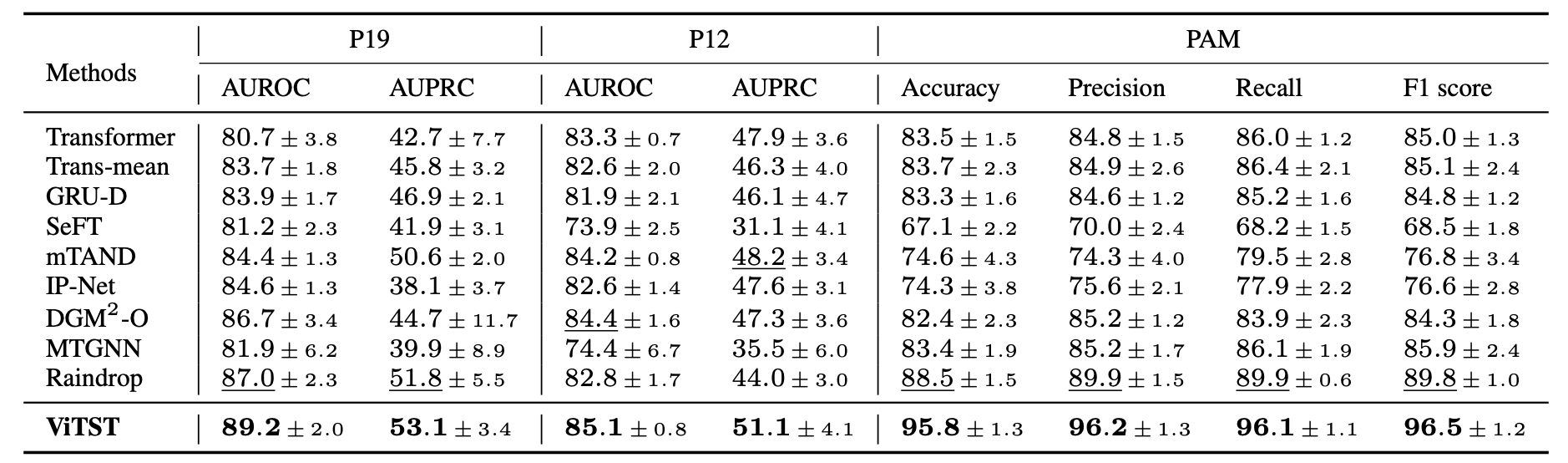


Table 3. DL baselines reported by ViTST.

**——————————————————————————————————————**

**IV. Next Steps**

——————————————————————————————————————

* Keep working on feature engineering with GL tools and try to beat the XGBoost baseline.

**——————————————————————————————————————**

**V. Milestone**

——————————————————————————————————————

* Obtained all baseline data for P12.
* Codes are available here: <https://github.com/d9sus4/GL-TS>