Team H2
Member 1: Jie Feng CHAI
GTID: 903842323
gt email: jchai41

Member 2: Man Kong SIT
GTID: 903760158
gt email: msit6

1) Introduction

The paper Unified Auto Clinical Scoring (Uni-ACS) with Interpretable Machine Learning Models (Li et al., 2022) introduces the Uni-ACS framework, designed to revolutionize traditional clinical scoring systems. These conventional systems are widely used in healthcare for risk assessment, prediction, and guidance in clinical decision-making based on patients' medical conditions. However, they are limited by their reliance on manual feature selection and lack of interpretability. To address the issue of manual feature selection, the Uni-ACS framework employs machine learning to automate the creation of clinical scoring systems. By learning from healthcare datasets, such as MIMIC III and IV, the framework selects relevant features and uses ensemble models for predictions. To enhance interpretability, the Uni-ACS framework incorporates SHAP (SHapley Additive exPlanations), which explains the importance of features in the clinical scoring system.

The significant contribution of this paper lies in demonstrating that the Uni-ACS framework can be applied to various clinical outcome predictions, including mortality, readmission, and length of stay. The Uni-ACS framework is not only more accurate but also easier to generalize compared to traditional scoring systems. The interpretability it provides also makes ML-based clinical tools more trustworthy.

2) Scope of Reproducibility

The purpose of Uni-ACS is to translate ML models into clinical scoring systems. To evaluate the reproducibility and effectiveness of Uni-ACS, it's essential to understand the components of classical clinical scoring systems:

Component A - Clinical Scoring Table: This comprises risk factors with their corresponding integer score values based on a defined range. Scores for individual risk factors are aggregated to yield a final score for a patient.

Component B - Score to Risk Mapping Table: This component maps aggregated scores to their respective risk percentages or odds ratios.

<u>Hypotheses</u>
The paper hypothesizes that Uni-ACS possesses the following attributes:

1. Automatic and Model-Agnostic Translation: Uni-ACS offers an automatic, model-agnostic method to convert ML models into clinical scores, encompassing essential Components A and B.
2. Consistent Interpretations: It maintains a consistent framework for both local and global interpretations across ML models and their corresponding clinical scores.
3. Preservation of Predictive Performance: Clinical scores translated by Uni-ACS retain most of the underlying ML models' original predictive performance.

Experiment

To test these hypotheses, we will compare the performance of Uni-ACS with the state-of-the-art clinical scoring methodology, Risk SLIM, and a classical clinical score development method (baseline method). Our experiments will focus on:

1. Quantitative Model Performance Comparisons: We'll assess metrics such as Area Under the Receiver Operating Curve (AUROC), Area Under the Precision-Recall Curve (AUPRC), and accuracy, comparing Uni-ACS with the baseline method.
2. Performance Pre and Post Uni-ACS Application: We'll evaluate the AUROC and AUPRC of ML models before and after applying Uni-ACS to understand the impact of the framework on model performance.
3. Qualitative Interpretation Comparisons: We'll conduct a qualitative comparison of the clinical scores derived from Uni-ACS against current clinical literature, ensuring the practical clinical applicability of these scores.

3) Methodology

Model description

Models	**Hyperparameters**
Logistic Regression (LR)	standard scaler, liblinear
Gradient Boosting Classifier (GB)	160 boosting stages, 3 max depth
Random Forest Classifier (RF)	140 trees, 5 max depth
Neural network, MLP (NN)	standard scaler, 1 hidden layer of 100 units, Sigmoid activation function, Stochastic gradient descent

Uni-ACS models first apply calibration using 5-fold cross-validation on training data, then pick the top 10 features with the highest average SHAP value, and use these SHAP values to calculate clinical scores. Finally, the prediction is based on the predicted clinical scores and the features' probability threshold.

Dataset description

In this project, we will utilize the MIMIC-III (version 1.4) and MIMIC-IV (version 0.4) datasets. These datasets are sourced from genuine patient encounters and encompass a broad range of medical information. They are collected from the Beth Israel Deaconess Medical Center and are not synthetic or self-generated.

Datasets can be found here:
MIMIC-III Clinical Database v1.4
MIMIC-IV v0.4

Initially, we will conduct tests on a small subset of the data, primarily as a preliminary check to ensure that all our developed scripts function smoothly and that we can replicate the findings of the paper. For the final report, we will employ the complete dataset from MIMIC to provide more robust performance metrics for the final models.

Data statistics

MIMIC-III (data available from original paper github repo)
4555 rows, 113 features, 70/30 train test split
label ("28 Day Death", binary): 38.09%
missing values: 169592

MIMIC-IV (data available from original paper github repo)
7690 rows, 84 features, 70/30 train test split
label (icu admission, binary): 26.879%
label (mortality, binary): 6.957%
missing values: 0

Data processing

4 post-processing steps are required to prepare data for risk SLIM & baseline.
One hot encode all categorical variables.
IterativeImputer is trained using the X_train and employed to impute missing values in both the X_train and X_test.
SelectKBest selects the top 10 features based on the chi-squared statistic, reducing the number of features.
The feature data in both X_train and X_test is normalized to a range of [0, 1].

Computational implementation

PyTorch and Scikit-Learn will serve as our fundamental coding frameworks. An Nvidia GPU will be utilized to accelerate computation. However, at this stage, we are only testing our code using a small subset of the data.

Code

Code can be found at github and instructions to run can be found in the README: https://github.com/cjiefeng/BD4H-H2

The required Python libraries are SHAP, PyGAM, CSAPS, Scikit-Learn, Pandas, and Numpy.

We have developed more robust preprocessing pipelines to speed up testing time. Medical explainer is referred from the original paper resources with fixes to make sure it is able to run smoothly.

4) Results

The results are generally similar to those that are reported from the paper, with the exception of the Uni-ACS NN model. This is due to computational complexity with kernel SHAP and hence it was only trained with a smaller subset of the data and yielded a poor performance.

Original model performance

Dataset	Model	AUROC	AUPRC	Accuracy
MIMICIII (Mortality)	LR	0.781	0.707	0.724
	GB	0.855	0.806	0.773
	RF	0.822	0.770	0.738
	NN	0.784	0.710	0.728
MIMIC-IV (Mortality)	LR	0.852	0.354	0.927
	GB	0.864	0.353	0.929
	RF	0.874	0.350	0.932
	NN	0.875	0.375	0.935
MIMIC-IV (ICU)	LR	0.901	0.809	0.864
	GB	0.911	0.827	0.871
	RF	0.904	0.817	0.853
	NN	0.906	0.817	0.864

Uni-ACS model performance

Dataset	Model	AUROC	AUPRC	Accuracy
MIMICIII (Mortality)	LR	0.692	0.566	0.659
	GB	0.724	0.641	0.679
	RF	0.754	0.678	0.649
	NN	0.511	0.395	0.612
MIMIC-IV (Mortality)	LR	0.798	0.231	0.931
	GB	0.842	0.290	0.931
	RF	0.843	0.309	0.932
	NN	0.5	0.069	0.069
MIMIC-IV (ICU)	LR	0.816	0.704	0.830
	GB	0.882	0.780	0.848
	RF	0.879	0.771	0.835
	NN	0.505	0.272	0.732

5) Discussion

The paper seems to be reproducible. We believe further improvements can be made to enhance the Uni-ACS models' performance (specifically, minimizing the performance drop after conversion to clinical scores).

The easy part of the reproduction process is that we use a small subset of the original datasets, allowing us to run most of our code without GPU acceleration or GPU-supported libraries. We anticipate difficulties in handling the original datasets (~20-40 GBs).

The only model that has difficulty running is the Uni-ACS NN model. This is due to the computationally expensive nature of calculating kernel SHAP for the NN model. As a workaround, an even smaller subset of data is used for this particular model, resulting in a significant difference in the Uni-ACS model performance metrics compared to the original NN model.

The original author could provide a list of dependencies and their respective versions. Some deprecated calls in the provided code broke the original code execution.

References

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and R Mark IV. Mimic-iv (version 0.4). PhysioNet, 2020.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. Scientific data, 3(1):1–9, 2016.

Li, A., Ong, M.L., Oei, C.W., Lian, W., Phua, H.P., Htet, L.H., & Lim, W.Y.. (2022). Unified Auto Clinical Scoring (Uni-ACS) with Interpretable ML models. 182:26-53 Available from https://proceedings.mlr.press/v182/li22a.html