DEEP LEARNING GROUP COURSEWORK

Shubham Kamble
University of Southampton
Student ID: 34966013
srk1g23.ac.uk

Dawei He
University of Southampton
Student ID: 34885358
dh5n23@soton.ac.uk

Bowen Pei
University of Southampton
Student ID: 34919313
bp2m23@soton.ac.uk

ABSTRACT

In this report, we are reimplementing the deep reinforcement learning experiments from Yu et al. (2023) to predict Acute Kidney Injury diagnosis. This report will give a breakdown of different experiments carried out, and a comparison of how well they perform to the results obtained in the paper. This report will then conclude on an evaluation if the experiments are reproducible, and assess if our results support the conclusions drawn fromYu et al. (2023)

1 INTRODUCTION

In recent years, the application of deep reinforcement learning has received widespread attention in the medical field, especially in optimizing diagnostic strategies and reducing medical costs. One of them, a semi-model-based deep diagnostic policy optimization (SM-DDPO) framework that is compatible with end-to-end training and online learning, is very eye-catching. By dynamically selecting experimental panels to achieve a balance between diagnostic accuracy and cost, SM-DDPO can train and identify all Pareto front solutions. Original studies demonstrate the effectiveness and cost-savings of SM-DDPO in multiple clinical tasks. To verify the reproducibility of this method, we used our own acute kidney injury (AKI) dataset to compare the performance of SM-DDPO with traditional methods (random forest, logistic regression, three-layer neural network, and XGBoost).

Acute kidney injury (AKI) is when the kidneys suddenly stop working properly. It can range from mild loss of kidney function to complete kidney failure (NHS (2024-05-10)). It is a common and serious clinical problem, and early and accurate diagnosis is of great significance to the treatment of patients. Traditional machine learning methods usually require pre-selected and fixed feature sets, while SM-DDPO dynamically selects features through reinforcement learning to reduce the cost of hospital detection. Our experiments aim to evaluate the reproducibility of this method on this dataset.

2 DATA PROCESSING

Data processing also plays a large part in this article. However, in the original paperYu et al. (2023), the author uses the MIMIC-3 database. The use of this database requires application and testing. Due to limited time, we looked for another similar smaller database on GitHub, which contains 2110 ICU clinical data and AKI grade of patients.

The data processing process is as follows: First, we extract the variables needed in the paper from the original data set, such as 'lactate_max', 'ph_min', 'ph_max', 'so2_min', etc. Then we dealt with missing values and outliers.To maintain integrity of the data, we deleted all columns and rows with more than 5% missing data. After initial cleaning, the data set remained with 1786 observations and 47 variables.

For categorical data, we convert it into an int type for model identification. For example, in patient ethnicity, 0 represents "unknown", 1 represents "white", 2 represents "minorities", and 3 represents "black_african_american"

We standardized the numerical data to eliminate the differences between different feature dimensions and make model training more stable. For specific data processing content, you can view data_cleaning.ipynb in the project GitHub.

Finally, before training each model, we divided the data set into a training set (70%), a validation set (15%), and a test set (15%) for model training, hyperparameter tuning, and final evaluation.

3 METHODOLOGY

This section will describe the different methods attempted to reproduce from the original experiment. We will present our approach to constructing each method and then present the results we achieved, and compare their performance against each other.

3.1 LOGISTIC REGRESSION

Firstly, the type data in the data is converted into numerical data, and then the data is randomly divided into 30% test set and 70% training set. Then feature scaling is performed on the data so that the mean value of the features is 0 and the variance is 1. For Logistic regression, after 100 iterations, the Logistic regression model uses the trained model to predict the data on the test set.

3.2 RANDOM FOREST

For the random forest model, it randomly samples in the training set, and randomly selects a feature subset to partition nodes. In its training process, random data sets and feature subsets are used for training, and the number of trees is specified as 100 to reduce overfitting. Finally, by means of voting, the prediction results of each decision tree for the sample are counted to give the results on the test set.

3.3 DEEP NEURAL NETWORK

Deep neural networks are often used for classification tasks. The one used in this report is a three-layer perceptron that can capture non-linear relationships. The structure of the three-layer DNN is as follows:

1. The first hidden layer is a fully connected layer (fc1), which can map the input data to a 128-dimensional hidden layer. Then the batch normalization layer (bn1) performs batch normalization on the output of the first hidden layer, normalizing the output of each batch to speed up training. Then use ReLU as the activation function and randomly drop some neurons with 50% probability in the Dropout layer (dropout) to prevent overfitting.

2. The second hidden layer is the same as the first. First, the fully connected layer (fc2) and then the batch normalization layer (bn2) also use ReLU as the activation function and have 50% Dropout.

3. Finally is the output layer. This is also a fully connected layer (fc3) which maps the output of the second hidden layer to a 3-dimensional output layer, representing predictions for three categories.

We used cross entropy as the loss function of the model. And select the AdamW optimizer to update the parameters of the model. A learning rate scheduler is also set up, which can dynamically adjust the learning rate during training.

3.4 XGBOOST

The XGBoost algorithm is an efficient implementation of the gradient boosting framework. Gradient boosting is a powerful machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. The model can achieve this by training each model to minimize the gradient of the residual errors of the previous model.

For our implementation, once the data has been split into the training and test set, and feature scaling has been performed, we made a decision tree and fit the data to this model. This is done by splitting the data based on the features. To find the best split, we calculated the gain of each split based on the gradient and Hessians and chose the optimum one. The tree can then be built recursively and predict values for the training set.

The XGBoost classifier implements the gradient boosting algorithm using the DecisionTree class as the weak learner. It trains the model by iteratively adding decision trees to fit the residuals of the previous trees. Then, predictions can be made by traversing through the final concatenated tree

3.5 SM-DDPO

The main algorithm suggested by the paper Yu et al. (2023) was to use a semi-model-based modular approach to train the panel selector and classifier concurrently. Different components are used to build this model.

1. Imp Model: The model consists of a linear layer that maps the input to a hidden representation. This model transforms the input state into a representation that can be used by the classifier and the policy. This is crucial for capturing relevant features from the input data.

2. Classifier: This model consists of a linear layer that maps the embedded state to the number of output classes. This model makes predictions by mapping the embedded state to a probability distribution over the labels. The classifier is trained by directly minimizing the cross-entropy loss from collected data.

3. Policy: This model selects the appropriate diagnostic actions or panels based on the embedded state. This panel selector model is trained by using proximal policy updates. The policy is optimized using reinforcement learning to maximize a reward signal, which reflects the performance of the diagnostic actions

The full algorithm updates the panel selector and classifier concurrently. The policy is updated based on the rewards obtained from the RL environment. The classifier is refined using mini-batches of data to minimize the classification loss.

This algorithm can do end-to-end training. It can start with no knowledge or a small amount of information and it can keep improving on new incoming patient data by querying test panels and finetuning the state encoder. Yu et al. (2023)

4 RESULT

Table 1 below shows the results we obtained from our experiments.

Table 1: Model performance

Models	F_1	AUC	$Accuracy$
LR	0.5931	0.5434	0.6996
RF	0.5922	0.5567	0.7108
three-layers DNN	0.5987	0.5430	0.7095
XGBoost	0.4210	0.4835	0.5810
SM-DDPO	0.6522	0.47717	0.7570

Our implemented models show reasonably high accuracy values, indicating that they are capable of generalizing well with the provided data and making accurate predictions. The accuracy levels are also not excessively high, which suggests that the models are not overfitting to the training data, thus maintaining their generalization capability to unseen data.

Among the models evaluated, the SM-DDPO implementation emerged as the best performer. It effectively learns and fine-tunes its parameters, leading to improved accuracy and robustness in predictions. The XGBoost implementation performed the worst out of all the models. It's performance can be hampered by the quality and relevance of the features provided.

5 COMPARISON WITH ORIGINAL PAPER

The original paperYu et al. (2023) used the MIMIC-3 database. By contrast, we use a stripped-down data set that only about 1700 samples, which was not as representative or as comprehensive as the MIMIC-3 database.
There are also slightly differences of the data standardization between the original paper and our techniques.
Additionally, the original paper uses different methods to dynamically extract features from the original data.
The differences in data set size and quality, and the methodology of processing data and feature selection, may account for differences in model performance between the reproduction process and the original paper. Overall, our models seem to perform just as well, with the results from the original paperYu et al. (2023)

Table 2: Model performance in the original paper

Models	F_1	AUC
LR	0.452	0.797
RF	0.439	0.764
3-layers DNN	0.494	0.802
XGBoost	0.410	0.781
SM-DDPO	0.495	0.795

6 CONCLUSION

Our models use a smaller and less comprehensive dataset, in contrast to the MIMIC-3 database used in the original paper. Our results show that although SM-DDPO performed best in these test models, with an F1 score is 0.6522, AUC is 0.47717, and an accuracy is 75.70%, there is still a significant difference in performance between the results and the original paper. These differences are mainly because of the variations of dataset size, dataset quality, and processing methods.

The smaller dataset used in our project likely limits the diversity and representativeness of the training data, which can affect the model's ability to generalize to new cases.

In conclusion, our study reconfirms the potential of deep reinforcement learning in the medical diagnostic task area, The results highlight the effectiveness of the SM-DDPO model even with a smaller dataset, indicating that such models can be robust under varying conditions. All of our implementations were reproducible, however, they do need a of resources and time to run. Additionally, this report also highlights the critical importance of data preprocessing and feature extraction in developing effective diagnostic tools.

REFERENCES

NHS. Acute kidney injury. 2024-05-10. URL https://www.nhs.uk/conditions/acute-kidney-injury/.

Zheng Yu, Yikuan Li, Joseph Chahn Kim, Kaixuan Huang, Yuan Luo, and Mengdi Wang. Deep reinforcement learning for cost-effective medical diagnosis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0WVNuEnqVu.