Prediction of Patient Outcomes after Renal Replacement Therapy (RRT) in the ICU

Harry Freitas Da Cruz
Hasso Plattner Institute (HPI)
Enterprise Platform and Integration Concepts
Potsdam, Germany
Email: Harry.FreitasDaCruz@hpi.de

Siegfried Horschig
Hasso Plattner Institute (HPI)
Enterprise Platform and Integration Concepts
Potsdam, Germany
Email: siegfried.horschig@student.hpi.de

***Abstract*—In order to compensate impairments of the renal system in the human body, artificial methods in the form of renal replacement therapy (RRT), called dialysis, have to be introduced. Many parameters of the dialysis can be adjusted and the outcome of the procedure may change with different patient characteristics. In this paper, we introduce a clinical decision support system to predict the effect of a given dialysis on a patient while in the intensive care unit (ICU).**
For this purpose, we employ two kinds of machine learning models: Bayesian Rule Lists (BRL) and Deep Neural Networks (DNN). Although the DNN may provide better accuracy, its decision making is not easily interpretable for humans. For this reason, we use mimic learning as a method to make the DNN interpretable.
Results show us that the DNN outperforms our BRL classifier as expected, but by a rather small margin. For the mimic learning process, we used a bayesian ridge regression model. Even though the regression model performs worse when training as a mimic model as opposed to drectly on the data, it provides some insight into the inner workings of the DNN.

1. Introduction

The renal system in the human body has the purpose to eliminate wastes from the body and control levels of certain substances in the blood. If this system is impaired, for example due to Acute Kidney Injury (AKI), artificial methods in the form of Renal Replacement Therapy (RRT) have to be introduced, more commonly known under the term of dialysis.
There are different options for dialysis available. One example is the hemodialysis, where the patient's blood is pumped through a dialyzer, inside of which is a liquid called dialysate. This liquid's composition determines which substances should be filtered out of the blood. The dialyzer separates the blood and the dialysate though a partially permeable membrane, allowing for the filtering of the blood through osmosis. Another example for the dialysis is the peritoneal dialysis, which uses the peritoneal cavity inside the patient as a container for the dialysate.

The dialysis outcomes are highly dependent on both the patients characteristics and the parameters as well as the type of the dialysis. So, usually, patients undergoing the peritoneal dialysis experience lesser health issues related to the dialysis than those undergoing hemodialysis, as there is less pressure on the circulatory system. On the other hand, the hemodialysis is more efficient in such a way that it needs less time for the same amount of filtration.
Especially the hemodialysis is a costly process which needs specialized equipment and therefore has many parameters to be tuned. These include, but are not limited to the duration of the process, the filtration rate and flow rates of the blood and dialysate. We want to introduce a machine learning model that, based on those parameters as input, allows to make a prediction for the health of the patient after the procedure.

Aside from usual criteria like accuracy or recall, when employing a machine learning model in the medical context one especially important factor is the interpretability of the model. This is due to the fact that the doctors want to make decisions based on those predictions and take full responsibility for the consequences, so they have to validate the decision making process. Oftentimes, models just show correlations between various parameters, and those correlations have to be manually checked for causal relations. Additionally, the European Union introduced regulations (taking effect 2018) that give consumers in any sector a "right to explanation". Essentially, this means that in any decision making process that is done by machines, be it a credit application or a diagnosis, the individual has the right to access meaningful information about the logic involved.

This way, we can roughly separate machine learning algorithms in two categories: interpretable and non-interpretable. One example for interpretable models are Bayesian Rule Lists (BRL). By presenting itself as *if...then...else* lists, it is easy for humans to comprehend both the decision making and the individual influence of each parameter on the outcome.
Deep Neural Networks (DNN) on the other hand are

non-interpretable, because the weights of the nodes in the hidden layers is everything they expose to the outside. Due to the fact that different loss and activation functions take effect when updating those weights, the abstraction to the original input data is just too large for a human to grasp.
For this reason, we want to make non-interpretable models interpretable, giving some insight into their inner workings and decision making.

The main purpose of this paper is to develop a clinical decision support system to determine patient-specific outcomes after RRT in the ICU and evaluate the interpretability of different models used in this process. We employ two different models, BRL as the interpretable one and DNN as the non-interpretable one. Afterwards, we show performance comparison of both those models and try to give some insight into the decision-making process of the DNN using mimic learning.

2. Related Work

In the field of prediction concerning the renal system, a lot of work has already been done. In [1] mortality of patients after RRT in the ICU was observed and set in relation to the method of dialysis, proposing a new cost-efficient using a single-pass batch system. The dosage of dialysis was discussed and set in correlation with blood values in [2].
Oftentimes, mortality is used as a target outcome in the renal context [3] [4] [5]. When it comes to model selection, [6] found a bayes algorithm suitable, while [4] compared regression, random forest and artificial neural networks and proposed the latter for better performance and accuracy. Other models include decision trees [7] and support vector machines [8] [9].
The term *interpretability* in the context of machine learning is defined and its extent discussed in [10]. In [11] methods of getting interpretability into models in the medical context is researched. Especially for the prediction in the ICU, [12] proposes mimic learning as a technique to be used for interpretability.

In the context of this work, we try to unify both the prediction of patient outcomes in the renal context as well as the interpretability of the models used. We compare the results of mimic learning as described in [12] with the results of an interpretable classifier, Bayesian Rule Lists as proposed in [13].

3. Tools

For quick prototyping, we used *RapidMiner* [14], allowing us to prepare data, develop and cross-validate first models. For deeper model development, we implemented them with the *scikit-learn* library [15] in Python.
The Data we used was provided by the *MIMICIII* dataset [16] stored in a *HANA* database [17].

4. Data

The *MIMICIII* dataset contains hospital admission data for patient collected over an eleven-year period in a Boston hospital. As seen in figure 4, out of the approximately 46,000 patients present in the dataset, we extracted 925 relevant patients for us, totaling to approximately 3,000 dialysis procedures we can train our models upon.

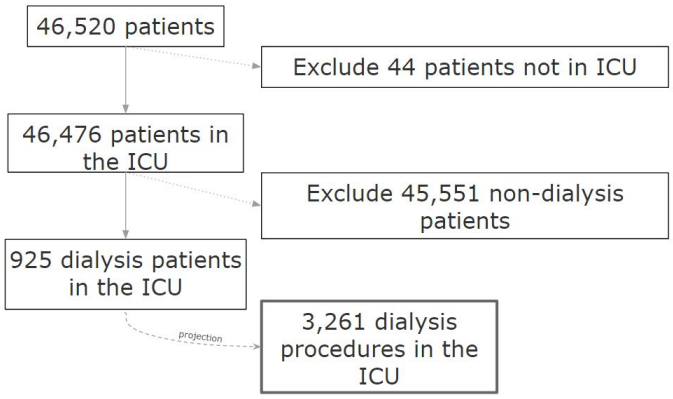

Figure 1. Cohort selection of the dialysis procedures based on the patients.

For each procedure, we have about 80 features defining it. Those include patient demographics such as age or BMI, dialysis parameters such as the duration of the procedure, comorbidities as well as lab values. Additionally, we include patient vitals and measurements on how ”well” the patient is doing such as 90-day mortality, the number of days he/she spent without mechanical ventilation and in the ICU in general.

4.1. Missing Data

Due to the manually curated nature of the *MIMICIII* dataset, aside from the inconsistencies within, a lot of data is missing. *TODO: Retrieve numbers!!!* As the scikit-learn models need a complete dataset for training, we decided to impute the missing values. We evaluated both mean/median imputation and K-Nearest-Neighbour imputation and decided on the latter, giving us an increase of about 2% in accuracy.

5. Models

In order to compare performance of both interpretable and non-interpretable models, we trained one model for each category. In the following section, we describe the models and strategies used as well as the parameters chosen for training.

5.1. Interpretable - Bayesian Rule Lists

For the interpretable model, we chose the existing Python 2 implementation of Bayesian Rule Lists (BRL) as

described in [13]. They pose itself as a direct competitor to decision tree approaches, as they have a high accuracy for classification while still being easy to read for humans.
This algorithm tries to find *if...then...else* statements over a dataset with the important criteria of them being sparse for better human readability. It achieves this goal by mining antecendents from the data and afterwards computing the posterior distribution over the antecedent lists.
The current implementation of BRL has the shortcoming of only being able to classify binary targets. Thus, we have to adjust the target features accordingly.

5.1.1. Parameters. The sole adjustable parameter in the used implementation is the maximum number of iterations. Multiple adjustments to this parameter - including changes by factor 10 - did not result in a significant change, neither for the runtime nor for the accuracy. For the evaluation, we chose a value of 50,000 maximum iterations.

5.2. Non-Interpretable - Deep Neural Network

As non-interpretable model, we chose the powerful and widely used Deep Neural Network (DNN). Specifically, the scikit-learn implementation Multi-Layer Perceptron (MLP). Just as other implementations, this network consists of multiple layers of so-called "neurons": one input layer with as many neurons as there are inputs, one output layer with the size of the number of target features and hidden layers varying in size and quantity. The weights of each neuron is updated after each iteration of training, optimizing the log-loss function.
Scikit-learn offers implementations for both regression and classification tasks.

5.2.1. Parameters. Neural networks have a wealth of parameters to be adjusted. Doing a grid search over some of the parameters, we found the default ones from the library to perform the best.
This means the learning rate - determining the speed and accuracy of convergence - is set to 0.001. The activation function, determining the output of the neurons in the hidden layer, is the rectifier linear unit "relu". The network consists of one hidden layer with 100 neurons. The maximum number of iterations before convergence is set to 200.

5.3. Interpretablity Approach - Mimic Learning

The large amount of neurons in the DNN and the many parameters influencing their weights and output make it very difficult - if not impossible - for a human to understand the influence of each feature on the training.
That is why we wanted to give some insight into the workings of the DNN by applying a method called *mimic learning*. Orientated on the works of [12], we train an interpretable model - the so-called "mimic model" - on the outputs of the non-interpretable model. The mimic model takes the same input features as the non-interpretable model. For classification tasks, the output of the non-interpretable model are called "soft scores", because as they are probabilites, they are continuous variables, coming close to the actual target features. In theory, the training of the mimic model on the soft scores allows to create a much smaller, thus understandable, faster but still equally accurate model. Using the principle of knowledge distillation, it is even possible for the mimic model to perform better than the non-interpretable model.
In our case, the DNN is the non-interpretable model. For the mimic model, we need a model, which is able to predict continuous scores. We decided on Bayesian Ridge Regression.

5.3.1. Bayesian Ridge Regression. The Bayesian Ridge Regression, like common linear regression, tries to find coefficients for each input feature so that they map to the target feature, minimizing loss. In addition to common linear regression, it includes regularization parameters to control the growth of the coefficients. Therefore, this model is less prone to overfitting while still being as fast as linear regression.

6. Performance Metrics

In order to compare the models' performance to each other, we have to specify performance metrics, in this case for binary classification. All of them are working with the terms of true and false positives and negatives. The terms *positives* and *negatives* refer to the prediction of the model, while the terms of *true* and *false* refer to the fact if the prediction of the model was correct. Additionally, all targets that are true are *relevant* elements.
Sensitivity / Recall specifies the number of relevant items that have been selected. This means the number of true positives divided by the number of all true targets.
Precision specifies the number of relevant instances in the result set. This means, out of all as positively classified targets, how many have actually been positive.
Specificity specifies the number of correctly classified negative instances.
The **Diagnostic Odds Ratio** (DOR) is defined as the ratio of the odds of the prediction being positive if the target is positive against the odds that the prediction is positive if the target is negative. It range from zero to infinity, with a DOR of 1 meaning that the prediction is equally likely to give a positive prediction no matter the true status. A higher value indicates a better prediction.
The **Receiver Operating Characteristics curve** (ROC) plots the true positive rate against the false positive rate at various thresholds. The important measure for this plot is the **Area Under the Curve** (AUC), a value ranging from zero to one, with an AUC of 0.5 describing a random classifier. Higher values indicate better classification performance.

7. Results

8. Discussion

9. Conclusion

10. Future Work

References

[1] V. Schwenger, M. a. Weigand, O. Hoffmann, R. Dikow, L. P. Kihm, J. Seckinger, N. Miftari, M. Schaier, S. Hofer, C. Haar, P. P. Nawroth, M. Zeier, E. Martin, and C. Morath, "Sustained low efficiency dialysis using a single-pass batch system in acute kidney injury - a randomized interventional trial: the REnal Replacement Therapy Study in Intensive Care Unit PatiEnts," *Critical Care*, vol. 16, no. 4, p. R140, 2012. [Online]. Available: http://ccforum.com/content/16/4/R140

[2] Z. Ricci, R. Bellomo, and C. Ronco, "Dose of dialysis in acute renal failure." *Clinical journal of the American Society of Nephrology : CJASN*, vol. 1, no. 3, pp. 380–388, 2006.

[3] B. J. Barrett, P. S. Parfrey, J. Morgan, P. Barré, A. Fine, M. B. Goldstein, S. P. Handa, K. K. Jindal, C. M. Kjellstrand, A. Levin *et al.*, "Prediction of early death in end-stage renal disease patients starting dialysis," *American journal of kidney diseases*, vol. 29, no. 2, pp. 214–222, 1997.

[4] K. R. Lakshmi, Y. Nagesh, and M. Veerakrishna, "Performance Comparison of Three Data Mining Techniques for Predicting Kidney Dialysis Survivability," *International Journal of Advances in Engineering & Technology*, vol. 7, no. 1, pp. 242–254, 2014.

[5] A. Kusiak, B. Dixon, and S. Shah, "Predicting survival time for kidney dialysis patients: A data mining approach," *Computers in Biology and Medicine*, vol. 35, no. 4, pp. 311–327, 2005.

[6] P. S. Baby and P. Vital, "Statistical Analysis and Predicting Kidney Diseases using Machine Learning Algorithms," *International Journal of Engineering Research & Technology*, vol. 4, no. 07, pp. 206–210, 2015.

[7] R. Greco, T. Papalia, D. Lofaro, S. Maestripieri, D. Mancuso, and R. Bonofiglio, "Decisional Trees in Renal Transplant Follow-up," *Transplantation Proceedings*, vol. 42, no. 4, pp. 1134–1136, may 2010. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0041134510003490

[8] S. Vijayarani and S. Dhayanand, "Kidney Disease Prediction Using Svm and Ann," *International Journal of Computing and Business Research (IJCBR)*, vol. 6, no. 2, 2015.

[9] P. Sinha and P. Sinha, "Comparative Study of Chronic Kidney Disease Prediction using KNN and SVM," vol. 4, no. 12, pp. 608–612, 2015.

[10] Z. C. Lipton, "The Mythos of Model Interpretability," no. Whi, 2016. [Online]. Available: http://arxiv.org/abs/1606.03490

[11] G. J. Katuwal and R. Chen, "Machine Learning Model Interpretability for Precision Medicine," 2016. [Online]. Available: http://arxiv.org/abs/1610.09045

[12] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Interpretable Deep Models for ICU Outcome Prediction." *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2016, pp. 371–380, 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/28269832

[13] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[14] "Rapidminer," https://rapidminer.com/, accessed: 2018-03-11.

[15] "scikit-learn," http://scikit-learn.org/stable/index.html, accessed: 2018-03-11.

[16] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific data*, vol. 3, 2016.

[17] F. Färber, S. K. Cha, J. Primsch, C. Bornhövd, S. Sigg, and W. Lehner, "SAP HANA database," *ACM SIGMOD Record*, 2012.