Using Stacking Methods to Improve Early Hospital Mortality Predictions in Coronary Care Unit Patients

Tanvi Bansal, Tyler Perez, Natasha Recoder, Lake Wang
New York University
Center for Data Science
npr264@nyu.edu

Abstract

Critical decisions about the medical care provided to incoming Intensive Care Unit (ICU) patients often rely on mortality predictions to inform the level of care needed. Allocating resources to individuals who are in the most need is of the utmost importance in ICU units and the medical field at large. Many machine learning methods have been applied to patient data to predict mortality while in the hospital, with varying results. This work aims to explore the use of stacking, a meta-learning technique, to understand if improvements can be made upon the methods used by Sadeghi et. al (2018) by following their pre-processing pipeline and incorporating their models as our base model. The meta learner used was a LASSO logistic regression model. We explore four different stacking configurations combining different uses of probability scores versus simple class binaries as meta learner input, and the inclusion or exclusion of data rescanning methods. As per Sadeghi et. al (2018), heart rate data from the MIMIC III database is broken down into 12 features as input into the base model. The results not only shine light on differing utility within differing stacking methods, but they also demonstrate the capability to improve upon any individual model using stacking. This is particularly seen when combining data rescanning with probability scores as the input into the meta learner. Additionally, the weights assigned to each model from the meta learner provide insight into which types of feature space transformations and decision boundary geometries work best for this task and within each stack configuration.

1 Introduction

While traditional score based approaches such as Acute Physiology And Chronic Health Evaluation (APACHE) are most commonly used in predicting early hospital mortality, many machine learning methods have also been applied to patient data to predict mortality while in the hospital, with varying results [1][2][3][4]. Improvements in the analysis driving this critical decision making process can have important consequences in the medical space. This work aims to explore the use of stacking, a meta-learning technique, to understand if improvements can be made upon the methods used by Sadeghi et. al (2018).

When using stacking methods, one combines various machine learning models by feeding their outputs into other machine learning models in hopes to improve upon the performance of any one model [5]. Stacking methods have been seen to improve predictions used by healthcare professionals across the medical field [6][7]. Most commonly, this is done by feeding the predictions of many models into another model known as the meta model or meta learner. In classification tasks however, it is also possible to give the meta model the probabilities that each model assigned to each class given the data [8]. Additionally, success has been achieved by feeding the initial data to the meta model, also known as rescanning, as well as the predictions from the previous models [9]. The effect of rescanning will be investigated, both classically with base model prediction as an input, and in conjunction with probability scores.

Other work using ensemble learning to predict mortality has been done with “conflicting results on the performance of different prediction tools” [9]. Studies have found varying levels of success utilizing differing machine learning techniques [10][11][12][13][14][15][16][17]. This is further seen in the work done by Sadeghi et al. (2018).

The main objectives of this study are to make improvements in early hospital mortality prediction, and to investigate stacking methods. The aforementioned work of Sadeghi et al. (2018), will be leveraged, in parallel with novel exploration of stacking methods. The research will focus on four unique stacking configurations. The methods will differ on the basis of inputting into the meta learner either the probabilities or binary predictions from the models used in Sadeghi et al., as well as whether or not to rescan the original data into the meta learner. All models will have a LASSO regression meta learner.

2 Methods

Sections 2.1, 2.2 and 2.3 outline the methods done by Sadeghi et al. (2018) as this work aims to mimic their pipeline before implementing stacking. For a more detailed understanding of the pre-processing pipeline see the original paper [1]. With the aim to replicate the pre-processing steps and results from the various machine learning algorithms implemented, their open source code was used directly.

2.1 Data Description

The MIMIC-III database consists of the records of 46,520 individuals who had been checked into an ICU. It includes, but is not limited to, vital signs, laboratory measurements, and medications [18]. The work at hand focused on a subset of 10,282 patients who had the required clinical data and vital signals. Nearly 90% of the patients suffered from cardiovascular disease, thus the chosen records were from patients admitted to the coronary care unit (CCU) which is an ICU that specializes in coronary care. The heart rate signal, measured in beats per minute was extracted from the database, as well as the binary label indicating whether or not the patient passed away while in the hospital.

2.2 Pre-processing

The MIMIC-III database collects its data from the CareVue and MetaVision clinical information systems provided by Philips and iMDSoft, respectively [18]. The tails of the data were truncated as they contained zeros or undefined values. Missing values were replaced with the previous known value. The heart rate signal was smoothed using a moving average filter with a one-hour window size.

Heart rate sampling rates were varied, thus an anti-aliasing finite impulse response (FIR) low-pass filter was passed over the signals with low sampling rate, effectively interpolating new samples.

Each heart rate signal is broken down into 12 statistical and signal-based features (Table 1). The work of Sadeghi et al. showed that high results are achievable using these simple features. All of the features except for averaged power and energy spectral density are statistical in nature.

The signal-based features were calculated from the power spectral density (PSD) of the signal over the entire frequency space. With S[n] as the signal, ρ as the sampling rate and ΔT as the interval the PSD, which is the fourier transform of the biased estimate of the autocorrelation sequence, was computed as:

$$\bar{P} = \frac{\Delta T}{N} \left| \sum_{n=0}^{N-1} S[n] e^{-i2\pi\rho} \right| \tag{1}$$

2.3 Base Model

Only a small proportion (18%) of the selected dataset passed away, thus it is imbalanced. As per Sadeghi et al. (2018), a resampling method, Adaptive Semi-Unsupervised Weighted Oversampling (A-

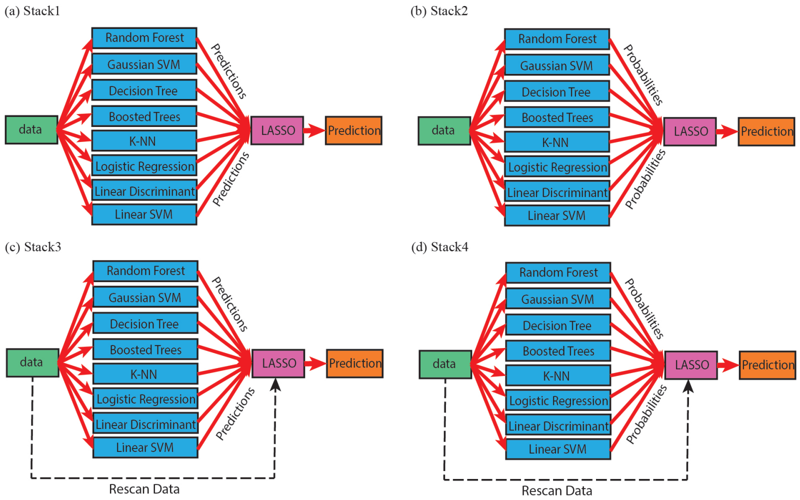


Figure 1: Four different stacking configurations explored. Stack1 feeds the logistic LASSO regression meta learner the binary classification labels from each base model. Stack2 provides the meta learned with the probability scores assigned to each class. Stack3 builds upon Stack1 by providing the meta learner with the original data through rescanning as well as the base model predictions. Stack4 similarly builds upon Stack2 by providing the meta learner with the original data as well as the probability scores.

SUWO) was used to balance the dataset. This method is less sensitive to outliers than other rebalancing methods and makes no assumptions about the distribution of samples, making it appropriate for the classification task at hand.

Eight separate machine learning models were run, as in Sadeghi et al (Table 2). These models were chosen to have a balance of transparent classifiers, such as decision tree, linear discriminant, logistic regression and support vector machine (SVM), as well as non-transparent classifiers such as K-NN, boosted tree and Gaussian SVM. The latter being less prone to interpretability. To validate the performance of the individual models as well as the fully stacked models we used a 10-fold cross validation strategy. These eight models were then used as a base model to feed into the meta model.

2.4 Stacking

A logistic LASSO regression model was used as the meta model to classify the signals with regularization strength of 1. This model is interpretable, and thus gives us insight into which models contributed the most to the classification task in each different stack. Four different stacking techniques were used. Stack1 followed the most traditional stacking sequence by simply feeding the binary predictions from each base model into the meta model. Stack3 elaborates on Stack1 by additionally providing the meta model with the initial data in a process known as rescanning [9]. Similarly to Stack1, Stack2 does not rescan the data, but unlike Stack1, Stack2 feeds the meta model the probability scores that each base model assigned to each class (alive or not). Stack4 operates similarly to Stack2, however it also rescans the data (Figure 1). The relative weights assigned to each base model from each stacking configuration were calculated (Table 4).

3 Results

3.1 Input Features

A statistically significant difference between the binary classes was observed for 7 of the 12 heart rate features used as input data into the base model as determined by a Mann-Whitney U test. Maximum heart rate, minimum heart rate, mean heart rate, median heart rate, mode of heart rate, average power, and energy spectral density, all calculated from the heart rate time series data, exhibited high significance with p-values smaller than 0.00001.

Table 1: Descriptive statistics for statistical and signal based feature per group

Feature	Passed away patients	Alive patients
Maximum***	97.67	91.27
Minimum***	79.83	75.66
Mean***	87.84	81.76
Median***	87.82	81.64
Mode***	84.01	79.33
Standard deviation	2.65	2.34
Variance	15.75	12.23
Range	17.85	15.61
Kurtosis	18.98	18.28
Skewness	0.95	1.08
Averaged power***	5024.83	4381.98
Energy spectral density***	8072.99	7017.30

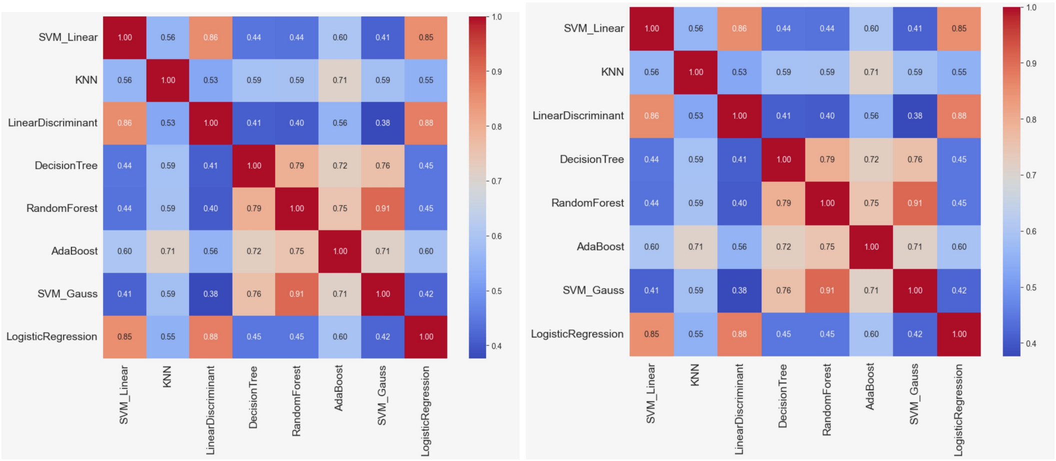
***p-value of Mann Whitney U test <0.00001

Table 2: Classification performance of CCU mortality for each base model.

Classifier	Precision	Recall	F1-score
Random forest	0.9635	0.9635	0.9635
Gaussian SVM	0.9652	0.9511	0.9511
Decision tree	0.8811	0.8810	0.8810
Boosted tree	0.8565	0.8513	0.8508
K-NN	0.7757	0.7740	0.7735
Logistic regression	0.7136	0.7113	0.7105
Linear SVM	0.7155	0.7057	0.7024
Linear discriminant	0.6935	0.6886	0.6867

3.2 Base Model

The random forest performed the best out of all the classifiers, with an F-1 score of 0.9635. The decision tree came in close second with an F-1 score of 0.9511, with the added benefit of interpretability. Linear based models performed relatively poorly achieving F-1 scores between 0.6867 and 0.7105 (Table 2).



(a) Correlation between prediction labels

(b) Correlation between probability scores

Figure 2: Correlation matrices between base models' (a) prediction labels and (b) probability scores

Table 3: Classification performance of CCU mortality for each stacking configuration.

Classifier	Rescan	Meta model input	Precision	Recall	F1-score
Stack1	No	Model predictions	0.9709	0.9552	0.9629
Stack2	No	Probability scores	0.9648	0.9562	0.9604
Stack3	Yes	Model predictions	0.9683	0.9614	0.9647
Stack4	Yes	Probability scores	**0.9672**	**0.9637**	**0.9654**

3.3 Correlation Between Base Model Outputs

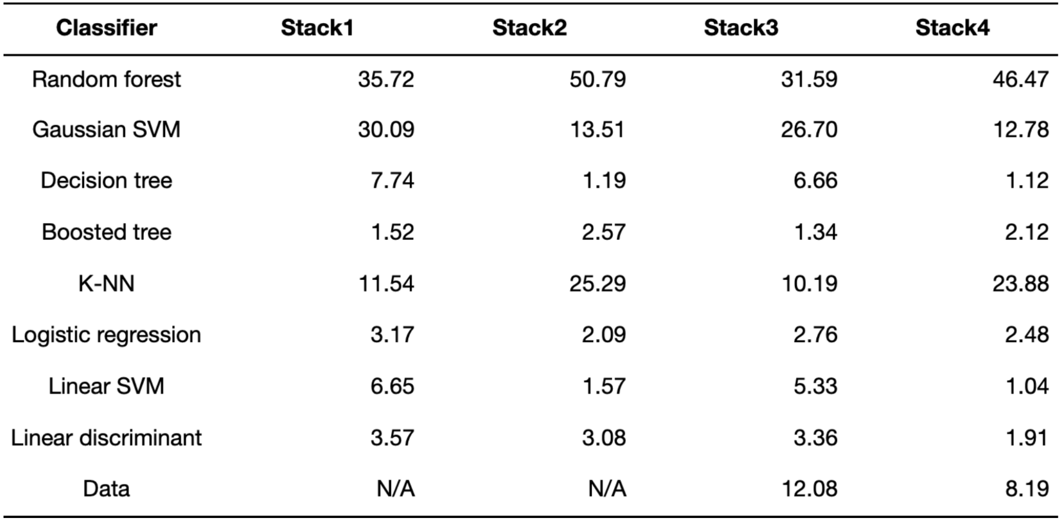
High correlation was observed between tree based models for both binary label classifier outputs and probability score outputs (Figures 2a,2b). Between random forest and decision tree correlation values of 0.79 for predictions and 0.87 for probability scores were observed. High correlation was additionally observed between linear models such as logistic regression and linear SVM, with correlation of 0.85 for predictions and 0.89 for probability scores. The highest correlation was observed between the two highest performing models, random forest and gaussian SVM with correlation of 0.91 for predictions and 0.94 for probability scores (Figures 2a,2b).

3.4 Stacking

Both the highest performing classifiers according to the metric of F-1 score rescanned the data (Table 3). The highest performing classifier was Stack4, where the original data was rescanned and probability scores were fed into the logistic LASSO regression meta learner. Stack4 outperformed the highest performing single classifier, the random forest in all metrics. Stack3, where the data was rescanned and prediction labels were fed to the meta learner outperformed the random forest in all metrics except recall.

In Stack1, where the model predictions from all of our base models without rescan were fed to the meta learner, superior performance to the random forest model was achieved in precision, but inferior performance was achieved in recall and F-1 score (Table 3). It achieved lower precision than the highest performing model in terms of precision, the gaussian SVM. The poorest performing stack in terms of F-1 score was Stack2, where it only outperformed the random forest in terms of recall.

Table 4: Percentage weight attribution to each base model from each stacking configuration.



Classifier	Stack1	Stack2	Stack3	Stack4
Random forest	35.72	50.79	31.59	46.47
Gaussian SVM	30.09	13.51	26.70	12.78
Decision tree	7.74	1.19	6.66	1.12
Boosted tree	1.52	2.57	1.34	2.12
K-NN	11.54	25.29	10.19	23.88
Logistic regression	3.17	2.09	2.76	2.48
Linear SVM	6.65	1.57	5.33	1.04
Linear discriminant	3.57	3.08	3.36	1.91
Data	N/A	N/A	12.08	8.19

3.5 Weights

The percentage contribution from each base model to each stack configuration was calculated by taking the absolute value of all the regression coefficients, and dividing the regression coefficient assigned to each model by the sum of all regression coefficients (Table 4). All of the stack configurations assigned highest weight to the random forest model with a range of 31.59% weight from Stack3, up to 50.79% weight for Stack2. Stacks 1 & 3, which used labels, not probability scores, assigned similarly high weights to Gaussian SVM, with weights of 30.09 and 26.70% respectively. Stacks 2 & 4, which used probability scores, assigned high weights to K-NN instead, with weights of 25.29 and 23.88% respectively. The lowest weights were assigned to the boosted tree model, with weights ranging from 1.34 to 2.12%, and linear models with weights ranging from 1.04 to 6.65%. The data was assigned weights of 12.08 and 8.19% for stacks 3 & 4 respectively.

4 Discussion

The two primary goals of this research were to see if stacking could improve performance on any singular base model and to gain understanding of how different stacking permutations affect results. The below sections address these goals.

4.1 Rescanning and Model Improvements

The goal of improving performance on any singular base model was accomplished by Stack4 (Table 3). The best performing base model, random forest, already achieved extremely high performance, and thus left little room for improvement. Although the improvements were minute, slight improvements in predictive power can have substantial effects to save lives in the medical field. Additionally, the training and testing data sets were kept consistent between stacks to ensure that any observed improvements were a result of the stack configuration alone. Stack4 utilized both the probability scores output by the base models, as well as a rescanning of the original data (Figure 1). The utility of using probability scores is based on the added information it gives compared to a simple binary prediction. Use of probability scores allows the meta learner to utilize the patient's distance to the decision boundary. An individual who is predicted to have a .99 likelihood of passing away would be believed to have different circumstances than someone assigned a .52 likelihood despite the fact the binary classification would assign them the same prediction.

Rescanning leads to improved results as this allows the model to recapture and incorporate the relationships within the covariate space [9]. Our LASSO logistic regression meta learner only allows the model to recapture linear relationships within the covariate space. This is a potential downside of the stacking architecture laid out, however we chose to prioritize interpretability of our meta learner. Further research into the use of non-linear meta learners could yield superior stacking models.

Differing use cases could lead to different model choices, if one was to prioritize precision they may choose Stack1, the most generic form of stacking. However within this context recall seems to be more consequential as this is the percentage of patients that passed away that the model was able to correctly classify, thus alerting health care providers to the severity of the situation.

4.2 Probability Scores and Model Improvements

Although we would expect the stacking configurations that utilized probability scores to outperform those that utilized binary labels, our results were conflicting. In the configurations that included rescanned data, the use of probability scores improved the meta learners performance (by F-1 score). However in the configurations that excluded rescanned data, the use of probability scores worsened performance (Table 3). A hypothesis for this behavior is that the linear base models introduce more error than explained variance which is amplified when using probability scores (continuous scale) as opposed to labels (discrete scale). This effect is dampened when rescanning is introduced by reducing the relative importance of the linear models, allowing the meta learner to recover some improvement from the linear relationships captured in the input data. This hypothesis is supported by the larger proportional weight drops in the linear base models when rescanned data is included as compared to nonlinear base models (Stack 1 & Stack 3, or Stack 2 & Stack 4), however further experimentation with additional stacking configurations and permutations is needed to test this hypothesis.

4.3 Stacking Model Weights and Relations Between Base Models

We suspect that the relationships between the input features are nonlinear in nature as there are many hidden factors dictating biological systems that may be confounding the relationship between our input features and output as the true data generating process underlying survival involves uncountable biological factors. It is hypothesized that this led to poor performance of linear models. For example, linear discriminant could have achieved poor performance because of the assumption that different groups of data are generated from separate Gaussian distributions (Table 2). This is likely untrue, as is seen from kurtosis and skewness values that are not equal to zero, as would be expected for true gaussian distributions (Table 1). Another possibility is that the data is not linearly separable, which would explain the poor performance of the linear SVM as well. Given the data involves the combination of uncountable factors it is unlikely that the data would be linearly separable.

Generalization and the risk of overfitting are pervasive concerns within any machine learning task. Such concerns were considered greatly within this research. The methods outlined here are assumed to generalize well due to two poignant reasons. The first of which is the usage of regularization which inherently decreases overfitting to the training data. Additionally the employment of 10 fold cross validation furnishes the necessary aspect of generalization. More research is needed to further optimize these models, specifically for the hyperparameters that are crucial in regularization. This can lead to finding a balance between increasing predictive power while not outputting a model that could be prone to overfitting. Hyperparameter tuning can be optimized in such a way to maximize certain metrics such as F-1 Score or Recall. Packages such as sklearn can be programmed in order to optimize specific metrics.

The usage of a regularized LASSO logistic meta-learner with a regularization strength of 1, provides increased interpretation of the individual base models' contributions. As expected the high performing random forest corresponded to the highest weight across all the stack configurations. The weights assigned to each base model from the meta learner provide insight into which types of feature space transformations and decision boundary geometries work best for this task and within each stack configuration. This can be seen through the relatively high weight assigned to the K Nearest Neighbors classifier (Table 4). The model provided relatively mediocre results within the base model as it ranked 5th in each accuracy metric out of the eight models, however consistently was one of the most important models for each stack configuration (Tables 2, 4). This illustrates the hypothesis that K-NN captures differing aspects of the data when compared to the random forest, even if the random forest does a substantially better job of capturing the true data generating process. Theoretically the geometry of the decision boundaries of these two architectures differ greatly. The K-NN model is specifically determined by the distances between the data points and the k-nearest neighbors, which yields a Voroni Diagram of polygonal shapes. This architecture relies heavily on local behavior. While the random forest decision boundary is a combination of the decision boundaries of the individual decision trees which yields a quite complex architecture that globally considers the entire feature space. Additionally, one can witness the fact that K-NN has heightened importance in configurations which utilize probability scores, while Gaussian SVM takes its place in ones that incorporate the binary classifications (Table 4). One hypothesis for this observation is due to the differing natures of the respective models' decision boundaries, as support vector machines' boundaries are determined by their kernel. A Gaussian kernel is used yielding a quasi-elipses boundary. When utilizing probability scores, the data points will be closer together in space, necessitating a more complex decision boundary such as that by K-NN.

4.4 Feature Research into Feature Importance

Although stacking yielded superior models and a greater understanding of each base model, it is not without its trade-offs. By stacking, one loses some interpretability of the original feature space, particularly when including some less interpretable models such as random forest. Professionals need to understand the reasoning behind the model outputs. This is particularly poignant within the medical sphere where health care providers need to do their best at saving lives within the context of the limited resources of ICU beds and infrastructure. The drastic nature of this reality was seen during the COVID-19 pandemic when healthcare workers were forced to decide who to allocate limited hospital resources to. It is therefore important for the professionals to understand why a model is outputting differing results for different individuals. One pathway for future research could be exploring black box explainer metrics such as SHAP or LIME on the stacked model [19]. Such explainers can be applied to any machine learning algorithm. These can be applied for understanding the feature importance in individual decisions, as well as at a model level. Additionally these methods focus on fidelity in a model agnostic manner to understand what features contribute to the model globally. With the inclusion of SHAP or LIME, stakeholders of the models could even see what specific features led to any specific decision as well as understanding what features have the most influence on the models at large.

This work did a brief dive into feature differences between the binary of passed away patients and surviving patients, finding statistical significance between 7 of the 12 features with a Mann-Whitney U test (Table 1). The usage of the Mann-Whitney test was catalyzed by the fact that data of the underlying population was neither normally nor evenly distributed between the two classes. This expanded on the previous research by Sadeghi et al. (2018) which simply listed the average values for each group, however our research output slightly different results. One can hypothesize these subtle differences being due to package updates or slight variations between the open source code and that used in the study. Furthermore, future research can expand on the Sadeghi et al. (2018) study while delving into the feature space/selection. In turn one can possibly only include the seven significant statistics between the two classes. Within this sphere, exploring the dependencies and relationships between the features themselves can provide even more information to further finetune models and ensemble learning methods. This could be combined with meta-learning on the input space directly, rather than just the individual features, which could increase predictive power in parallel with explainability as shown in other research [20].

4.5 Conclusion

It has been shown that stacking methods have the ability to increase predictive power and accuracy when compared to the best individual performing base model. The subtle yet impactful improvements are not trivial. Their importance is not only seen from a machine learning sense but in addition, their employment has the ability to save lives and help optimize the finite resources within an ICU unit and the medical sphere. Stacking methods additionally offer key insights into the model space. Given the stakeholders will not always have a machine learning background, future research within interpretability of the feature space is of utmost importance when the matters are truly life or death.

5 References

[1] Sadeghi, R., Banerjee, T., & Romine, W. (2018). Early hospital mortality prediction using vital signals. *Smart Health*, *9–10*, 265–274. https://doi.org/10.1016/j.smhl.2018.07.001

[2] Knaus, W. A., Draper, E. A., Wagner, D. P., & Zimmerman, J. E. (1986). APACHE II-A Severity of Disease Classification System. *Critical Care Medicine*, *14*(8), 755. https://doi.org/10.1097/00003246-198608000-00028

[3] Luca Citi, & Barbieri, R. (2012). PhysioNet 2012 Challenge: Predicting mortality of ICU patients using a cascaded SVM-GLM paradigm. *Computing in Cardiology*, 257–260.

[4] Ribeiro, C. M., Beserra, B. T. S., Silva, N. G., Lima, C. L., Rocha, P. R. S., Coelho, M. S., Neves, F. de A. R., & Amato, A. A. (2020). Exposure to endocrine-disrupting chemicals and anthropometric measures of obesity: a systematic review and meta-analysis. *BMJ Open*, *10*(6), e033509. https://doi.org/10.1136/bmjopen-2019-033509

[5] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259. https://doi.org/10.1016/s0893-6080(05)80023-1

[6] Byeon, H. (2021). Exploring Factors for Predicting Anxiety Disorders of the Elderly Living Alone in South Korea Using Interpretable Machine Learning: A Population-Based Study. *International Journal of Environmental Research and Public Health*, *18*(14), 7625. https://doi.org/10.3390/ijerph18147625

[7] Wang, P., Zhang, W., Wang, H., Shi, C., Li, Z., Wang, D., Luo, L., Du, Z., & Hao, Y. (2024). Predicting the incidence of infectious diarrhea with symptom surveillance data using a stacking-based ensembled model. *BMC Infectious Diseases*, *24*(1). https://doi.org/10.1186/s12879-024-09138-x

[8] Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Using Stacking to Average Bayesian Predictive Distributions (with Discussion). Bayesian Analysis, 13(3), 917–1007. https://doi.org/10.1214/17-ba1091

[9] Taghizadeh-Mehrjardi R, Schmidt K, Amirian-Chakan A, Rentschler T, Zeraatpisheh M, Sarmadian F, Valavi R, Davatgar N, Behrens T, Scholten T. Improving the Spatial Prediction of Soil Organic Carbon Content in Two Contrasting Climatic Regions by Stacking Machine Learning Models and Rescanning Covariate Space. Remote Sensing. 2020; 12(7):1095. https://doi.org/10.3390/rs12071095

[10] Awad, A., Bader-El-Den, M., McNicholas, J., & Briggs, J. (2017). Early hospital mortality prediction of intensive care unit patients using an ensemble learning approach. *International Journal of Medical Informatics*, *108*, 185–195. https://doi.org/10.1016/j.ijmedinf.2017.10.002

[11] Wojtusiak, J., Elashkar, E., Nia, R. M., (2017). C-Lace: Computational Model to Predict 30-Day Post-Hospitalization Mortality. In HEALTHINF, pp. 169–177.

[12] Ribas, V. J., López, J. C., Ruiz-Sanmartín, A., Ruiz-Rodríguez, J. C., Rello, J., Wojdel, A., & Vellido, A. (2011). Severe sepsis mortality prediction with relevance vector machines. In Engineering in Medicine and Biology Society, EMBC, 2011 annual international conference of the IEEE, pp. 100–103.

[13] Kim, S., Kim, W., & Park, R. W. (2011). A comparison of intensive care unit mortality prediction models through the use of data mining techniques. Healthcare Informatics Research, 17(4), 232–243.

[14] Purushotham, S., Meng, C., Che, Z., & Liu, Y. (2017). Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets. arXiv preprint arXiv:1710.08531.

[15] Avati, A., Jung, K., Harman, S., Downing, L., Ng, A., & Shah, N. H. (2017). Improving palliative care with deep learning. arXiv preprint arXiv:1711.06402.

[16] Beaulieu-Jones, B. K., Orzechowski, P., & Moore, J. H. (2017). Mapping patient trajectories using longitudinal extraction and deep learning in the MIMIC-III critical care database. bioRxiv, 177428.

[17] Song, H., Rajan, D., Thiagarajan, J. J., & Spanias, A. (2017). Attend and Diagnose: Clinical Time Series Analysis using Attention Models. arXiv preprint arXiv:1711.03905.

[18] Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. Scientific data, 3, 160035. https://doi.org/10.1038/sdata.2016.35

[19] Lewis, F., Butler, A., & Gilbert, L. (2010). A unified approach to model selection using the likelihood ratio test. Methods in Ecology and Evolution, 2(2), 155–162. https://doi.org/10.1111/j.2041-210x.2010.00063.

[20] Iwata, T., & Kumagai, A. (2020). Meta-learning from Tasks with Heterogeneous Attribute Spaces. Neural Information Processing Systems, 33, 6053–6063.

6 Appendix

Evaluation Metrics

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F 1 − score} = \frac{2 \times (\text{ Precision } \times \text{ Recall })}{\text{Precision } + \text{ Recall}} \tag{4}$$

Precision, recall and F1-score were used to evaluate each model as per Sadeghi et al. (2018). A positive label was used to indicate mortality in the hospital, thus a true positive (TP) indicates a patient who was labeled as passing away by the model and who truly passed away. A false positive is a patient labeled as a positive who did not pass away in the hospital (FP). A true negative (TN) is a correctly identified record of a patient who survived and a false negative is a patient incorrectly identified as surviving (FN).

The precision metric was calculated as per (2). This metric will be smaller when more patients are incorrectly labeled as passing away. The recall metric, calculated as per (3), represents the proportion of patients that correctly predicted as passing away, out of all of the patients who passed away. The F1 score was calculated as per (4), as it is able to convey information about the model's classification ability for both the passed away and living patients.