DeepRiskICU: A Machine Learning Application for Predicing Mortality Risk in ICU Patients

Tyler Anderton

1 INTRODUCTION

The Intensive Care Unit (ICU) is a specialized hospital department that provides critical care to patients with serious, life-threatening illnesses, or injuries. Generally the fatality rate is quite high, which demands immediate, informed, and decisive care. This care in turn requires exceptionally skilled staff and scarce medical equipment that need to be managed carefully to optimize patient outcomes.

The high-stakes environment of the ICU has attracted the attention of many medical researchers, all seeking to make an impact on the lives of critical patients around the world. In addition to requiring the careful management of hospital resources, the nature of care in the ICU also demands continuous data collection and monitoring of each patient. This allows providers to detect and treat signs of deteriorating conditions as early as possible, and for researchers, it also provides a data-rich environment to support a large body of work.

In the ICU, one of the most important outcomes that can be predicted from patient data is their mortality risk, or their risk of dying while in the hospital. This prediction can be based on a wide variety of patient information, ranging from their demographic information to their medical history and current vital signs. This mortality risk prediction provides a reference point for physicians to judge their patient's condition and to determine the most appropriate plan of action. In doing so, mortality risk predictions can simultaneously assist individual patients in receiving the highest quality attention while also optimizing resource allocation across all patients in the department.

This paper presents a web application, DeepRiskICU, that offers an easy-to-use interface for obtaining early mortality risk predictions for patients in the ICU within the first 24 hours of their admission. This interface requires the user to have no prior knowledge of statistical methods and allows a provider to acquire an accurate mortality prediction for their patient within seconds. This speed and ease of use leaves the caregiver's workflow uninterrupted and enables lightning-fast decision making that can improve patient outcomes and optimize the allocation of hospital resources. The underlying models in this application have been validated on the MIMIC III dataset to achieve 0.792 AUROC with an XGBoost model and 0.809 AUROC with a neural classifier. These performance metrics are already competitive with SOTA methods, and the methodology of this paper leaves much room for further iteration and improvement.

2 RELATED WORK

2.1 EHRs and MIMIC III Dataset

Earlier methods for predicting mortality risk were both limited by the data available and by the complexity of the calculation. The implementation of Electronic Health Records (EHRs) has revolutionized this process by digitizing the storage of comprehensive patient data across departments and even across hospital networks. By digitally and often automatically recording patient information, analyses and predictions like mortality risk have become more powerful and accessible to better inform physicians' decisions and improve the efficacy and efficiency of care.

One of the most comprehensive sources of ICU data widely available to researchers is the Medical Information Mart for Intensive Care (MIMIC) III database [4]. This publicly-available dataset was collected on 38,597 adult patients and 49,785 admissions to ICUs of the Beth Israel Deaconess Medical Center in Boston, Massachusetts between the years 2001 and 2012. It includes a wide variety of patient data including admission information, patient demographics, vital sign measurements, laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality dates and times.

2.2 Mortality Prediction Methods

2.2.1 General ICU Population Methods. Even with the revolution of EHRs, many methods for mortality risk prediction in the current literature have been limited in a number of ways. Many traditional statistical methods have been hindered by their mathematical complexity and difficulty to implement properly. Machine learning (ML) models have proven themselves to be dominant over these traditional methods for two main reasons. Firstly, they are often simpler to implement by researchers and easier to use by providers, as they don't require complex statistical calculations to be performed explicitly. Secondly, ML models have demonstrated the ability to ingest large amounts of data to identify complex patterns and relationships that traditional methods cannot, thereby scoring higher on industry standard metrics.

Many researchers have utilized the aforementioned MIMIC III dataset to train and evaluate ML models for the purpose of mortality risk prediction. Liu et al. 2022 implemented a combination of LASSO regression for feature selection and logistic regression for binary classification to design their mortality prediction model [5]. After reducing the dataset by their own inclusion criteria, the authors included 9,276 patients to train and evaluate their model, with a mortality rate of 11.78%.

In binary classification tasks like this, especially those that involve heavy class imbalance, the area under the receiver operating curve (AUROC) is the industry standard metric to understand the performance of one's model. The receiver operating curve (ROC) plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across a range of classification thresholds. The area under the curve (AUC) measures the overall ability of the model to discriminate between the positive and negative classes, independent of the classification threshold, with 0.0 being the worst possible score, 0.5 representing random chance, and 1.0 indicating perfect performance. A key feature of AUROC is its insensitivity to class imbalance, making it ideal for measuring model performance

in the task of mortality prediction, where the positive class (fatality) is almost always heavily out-weighted by the negative class (survival).

Liu et al. 2022 achieved 0.821 AUROC with their model, which is near state of the art (SOTA) on this task of predicting mortality across all patients in the ICU. While the authors' model performed quite well, they propose a nomogram as their final interface by which providers can produce mortality predictions. A nomogram is a 2-dimensional graphic calculator, often printed on a piece of paper or implemented in a simple GUI. Each feature is assigned a point value, and the points are added up to calculate the final prediction. Nomograms offer benefits over traditional statistical methods in that they are easy to use, they can be based on quite performant ML models, and they are very interpretable. This interpretability is a key feature, allowing providers and patients to understand exactly which features weigh heavily into the mortality prediction. However, depending on how they are implemented, they might lose some of the performance of the underlying ML model, and they can take a small but significant amount of time and effort to operate.

2.2.2 ICU Subpopulation Methods. Ashrafi et al. 2024a also utilized the MIMIC III dataset to train and evaluate ML models for mortality prediction [1]. These authors focused on adult patients diagnosed with heart failure, as identified by ICD-9 codes, which reduced their dataset to include only 1,177 patients. This group sought to compare and evaluate a variety of ML methods including Logistic Regression, Support Vector Machine (SVM), Random Forest, LightGBM, and XGBoost. XGBoost is a robust gradient boosting algorithm that has proven itself to be effective in a wide variety of ML tasks across industries, especially those that require a lightweight solution for processing large amounts of complex structured data. Out of the models tested, Ashrafi et al. 2024a found XGBoost to outperform the other methods by a wide margin, achieving 0.923 AUROC. This performance is incredible yet strictly limited to patients suffering from heart failure.

In recent years, neural networks have demonstrated their efficacy in a wide range of tasks, particularly excelling where traditional ML models might fail to extract critical relationships from extremely large amounts of data. As the name would suggest, neural networks function by scaling rather simple units, called "neurons", into incredibly deep and complex networks. Each neuron represents a simple linear function of its input values to produce an output. These outputs are then fed through a special "activation layer" that captures nonlinear relationships before sending the signal to the next neuron. Besides a few other tricks, these neurons are arranged in layers that can scale on the order of hundreds or thousands. Each layer then feeds into the next until the final layer outputs the desired calculation.

Ashrafi et al. 2024b recently demonstrated a simple fully connected neural network (FCN) can perform better than traditional ML models, including XGBoost, in the task of mortality prediction [2]. These authors also used the MIMIC III dataset, but focused their research on patients who required the assistance of mechanical ventilation. This left them with 16,499 patients in their dataset. After selecting 12 features, they achieved 0.879 AUROC with their neural network, outperforming XGBoost, the best of their baseline ML models, which scored 0.854 AUROC.

2.2.3 Present Study. In the present study, I aim to demonstrate the efficacy of XGBoost and neural classifiers in the task of mortality prediction using the MIMIC III dataset. Liu et al. already demonstrated the efficacy of a simpler Logistic Regression model for this task on the same dataset. Given the excellent performance of the more advanced XGBoost and neural models on the specific heart failure and mechanical ventilation groups, this study seeks to extend that work to the general ICU population.

Furthermore, all of the aforementioned studies included data from the entirety of a patient's stay in the ICU. This identifies a potential oversight in their methods. In order to urgently develop effective treatment plans, inference models must be able to make accurate predictions as early as possible. If the model in question has been trained on data that would not be available this early, it will be making out-of-domain predictions, which will likely result in dangerously inaccurate predictions in the real world. Therefore, these models must only be trained on data that will be available at the time of inference. To ensure the robust utility of the proposed models, the dataset in this study was restricted to include only information available within the first 24 hours after a patient's admission to the ICU.

Finally, after training and validation, the models proposed in this study have been packaged into a easy-to-use web interface that allows providers to obtain mortality predictions within seconds.

3 METHODOLOGY

3.1 Dataset

This study uses the Medical Information Mart for Intensive Care (MIMIC) III database [4]. Initially, 58,976 admission rows and 46,520 patient rows were found in the source tables.

Newborn and elective admission types were filtered out to only include emergency and urgent ICU admissions, leaving 43,407 admissions. The admission location and insurance type from the admissions table were kept as categorical features. The patients' gender, age at the time of admission, and ethnicity were retained as demographic features. The ethnicity was categorized into five buckets: white, black, hispanic, asian, and other, with white making up the vast majority (70%) of patients. When calculating the age, ostensibly due to the time-shifted nature of the MIMIC III dataset, several patients had invalid or unrealistic age values. These patients were dropped, leaving the final dataset with 40,900 admissions and 31,663 distinct patients. The categorical features gender, ethnicity, admission location, and insurance type were encoding using one hot encoding, and the age was standardized across the entire dataset. After determining the final set of admissions and patients, the label was determined by the presence of a valid death time in the admissions table, with 12.35% of patients in the positive (fatality) class.

With the requirement of restricting data to only the first 24 hours after the time of admission, several data were inadmissible, including procedures and diagnoses that did not have timestamp information. Instead, information on these events was extracted from the notes and prescriptions source tables. First, notes were filtered

to exclude marked errors, missing timestamps, and timestamps more than 24 hours after their respective admission times, leaving 229,486 notes in the dataset. The same was done to the prescriptions data, resulting in 1,265,014 valid entries with drug names, doses, and start times. These notes and prescriptions were then each concatenated into single strings for each admission, and embeddings were extracted for each of these concatenated entries. Having been fine-tuned on a corpus of scientific and medical texts, ClinicalBERT was chosen to extract these embeddings [3]. ClinicalBERT has a rather small input limit of only 512 tokens, so after tokenization, each concatenated notes or prescriptions entry was chunked and fed into ClinicalBERT in batches of 256. The chunk embeddings were then averaged into a single 768-dimension array for each of the notes and prescriptions for each admission ID. This process was performed on an Nvidia RTX 4070 Super with 12GB RAM and took took 68 minutes to extract a total of 36,181 note embeddings and 7 minutes to produce 39,618 prescription embeddings.

Finally, data from all four source tables were merged into a final feature set. Many admissions did not have matching notes or prescriptions, so these entries were filled with 768-dimension zero-arrays to match the embedding dimension and type. Principle Component Analysis (PCA) was applied to each of the embedding features with 99% variance retained for each. The notes embedding PCA retained 110 components, and the prescriptions embedding PCA retained 39 components. Combined with the standardized age and the encoded categorical features, a total of 168 features were included in the final dataset.

3.2 Model Training

3.2.1 Logistic Regression Baseline. To develop a baseline of performance, I trained and evaluated a Logistic Regression binary classification model. The hyperparameters were tuned with Scikit Learn's RandomizedSearchCV over five folds. This tuning ran for 10 iterations and optimized the sklearn LogisticRegression's inverse regularization strength C between 0.01 and 100, penalty between l1 and l2, solver between liblinear and saga, and maximum iterations between 100 and 500. The best model was then evaluated on a 20% test split of the dataset.

3.2.2 XGBoost. The XGBoost model was tuned and evaluated similarly. Here tuning ran for 30 iterations and the hyperparameters tuned were: maximum depth between 3 and 10, learning rate between 0.01 and 0.2, number of estimators between 100 and 500, minimum child weight between 1 and 10, gamma between 0 and 0.5, subsample between 0.6 and 0.4, column subsample by tree between 0.6 and 0.4, regularization parameter alpha between 0 and 1, and regularization parameter lambda between 1 and 10. All other tuning parameters were identical to Logistic Regression, and the best XGBoost model was also evaluated on the same 20% test split.

3.2.3 Neural Classifier. The neural classifier was implemented with Pytorch and designed with a relatively simple fully connected architecture. Three sizes of this architecture were implemented and tested (Fig 1). During training, early stopping and learning rate scheduling were implemented, and the Adam optimizer was used. Two loss functions were tested. BCEWithLogitsLoss (Binary Cross Entropy Loss) was used with the positive weight parameter set

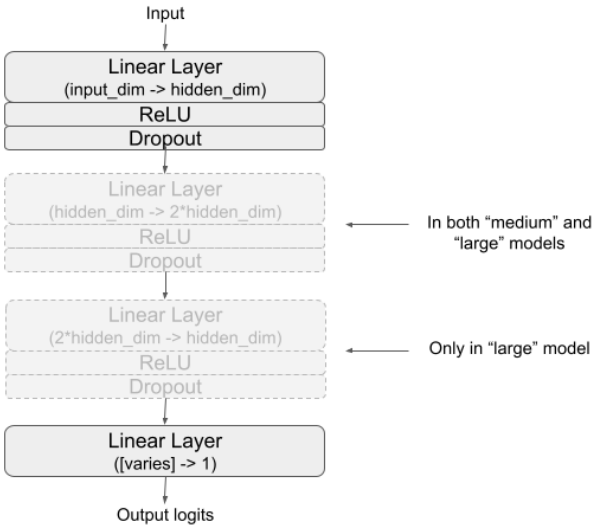


Figure 1: The "small" version of the model consisted of a single Linear layer expanding to a tuned hidden dimension, a ReLU activation layer, a Dropout layer, then a final Linear output. In addition to the "small" version, "medium", and "large" versions were also tested in the hyperparameter tuning process. The "medium" version included another Linear->ReLU->Dropout module that expanded to twice the hidden dimension before the output layer, and the "large" version included yet another module that reduced the dimensionality back down to the hidden dimension before the output layer.

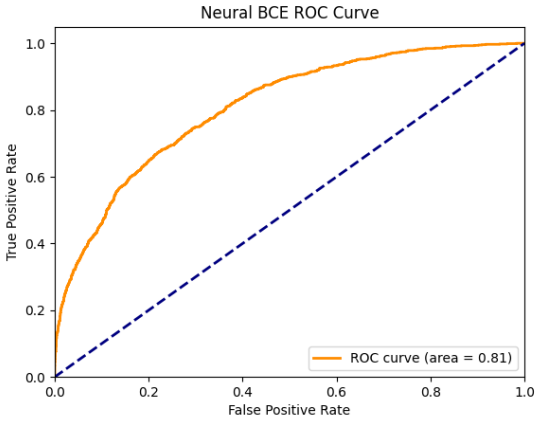
to account for class imbalance. To further account for this imbalance, Focal Loss was also implemented, but this did not improve performance.

The model architecture versions were optimized by hand along with the hyperparameters. Future work could automate the hyperparameter tuning process for further optimization. The other hyperparameters that were tuned included: hidden dimension from 64 to 1024, learning rate from 1e-3 to 1e-2, learning rate scheduler factor from 0.1 to 0.5, batch size from 64 to 8192, dropout rate from 0 to 0.5, and early stopping patience from 3 to 10. All models in this study were evaluated with a fixed classification threshold of 0.5.

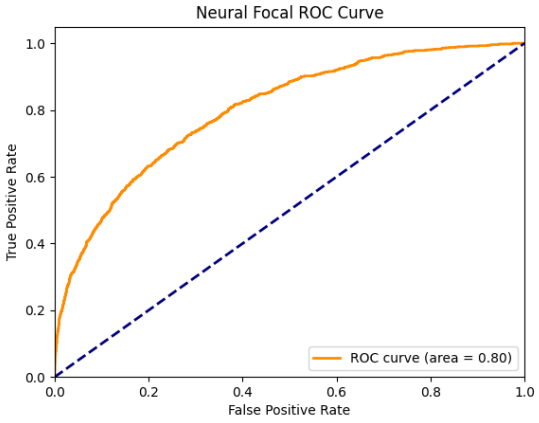
3.2.4 Application. After validating the models, the DeepRiskICU web interface was developed using the Django framework. The interface allows providers and administrators to login, submit patient information to the database, and retrieve mortality risk predictions when needed. Furthermore, each prediction request stores the input data and the model's predictions so that (with proper data handling strategies) the data can be used to monitor and improve model performance over time. All data can be stored securely wherever the app is deployed without violating any laws or regulations around privacy and security of patients' information.

4 RESULTS

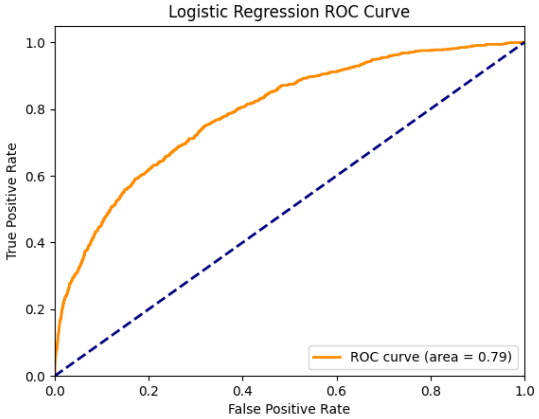
Ultimately all models performed very similarly, but both of the neural models achieved slightly higher AUROC scores than the



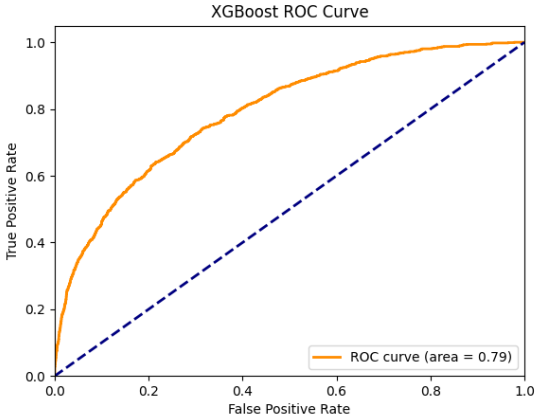
(a) Neural (BCE Loss)



(b) Neural (Focal Loss)



(c) Logistic Regression



(d) XGBoost

Figure 2: ROC curves for all four models.

traditional ML models. The neural model trained with BCE Loss performed the best out of all models, achieving 0.809 AUROC (Fig 2a) with 0.38 positive-class F1, while Focal Loss produced very similar results with 0.801 AUROC (Fig 2b) and 0.33 positive-class F1.

Both traditional ML models performed nearly identically in terms of AUROC. The best performing Logistic Regression model achieved 0.791 AUROC (Fig 2c), but only 0.25 positive-class F1. XGBoost failed to improve on this baseline AUROC score by a significant margin, yielding only 0.792 AUROC (Fig 2d), but returned a greater 0.41 positive-class F1 score.

Ultimately, these results validate XGBoost and neural models for the purpose of predicting mortality in the general ICU population and support the efficacy of the DeepRiskICU application.

5 CONCLUSION

This paper proposes two candidates for machine learning models to predict mortality in ICU patients. Both XGBoost and the neural classifier with BCE Loss met or exceeded the baseline performance of the previously-established Logistic Regression model. The rather low positive-class F1 scores across the board represent a necessity to tune the classification thresholds, but the high AUROC scores promise this tuning should prove successful.

While both XGBoost and neural models performed similarly in this study, each has tradeoffs to consider in the context of this task. Neural models in general may have an edge in some contexts where the feature set is too large for traditional ML methods to identify key patterns and relationships. Admittedly, given the comparative performance on this particular dataset, that is not the case here. While the neural models outperformed XGBoost by a very slight margin, this gain may not translate to real-world ICUs. The neural models were limited by overfitting at all sizes, which suggests that this feature set was not rich enough to warrant the relatively complex neural model. Future plans include the integration of a wider variety of features from the MIMIC III dataset and even lab images into a multimodal classification model. Such a dataset would likely benefit from – and even require – the power of a complex neural model.

Besides the ability to manage large and complex datasets, the advantages of neural models end there. Neural models are difficult to deploy, as they require large amounts of memory and compute, and depending on serving requirements, they might need GPUs to batch large volumes of requests. XGBoost models on the other hand are relatively lightweight, and could easily be deployed on site at hospitals, or even on edge devices in some special cases.

Neural models as well are often referred to as "black boxes" insofar as the inner workings of the model are a mystery. This is undesirable in medical practice, where caregivers and patients alike generally benefit from understanding what factors contribute to the model's predictions. XGBoost on the other hand allows for some degree of interpretability, as the importance weights of each feature can be extracted from the model. In the case of the XGBoost model in this study, the top five features were two notes embedding components, patient age, ethnicity, and Medicare status (which is likely correlated with the patient's age). At least with this information, the attending physician could know to look in the patient's notes to see what patterns the model might be identifying. Considering these advantages, the production version of DeepRiskICU serves the XGBoost model by default, and future iterations of the app will include these feature importances as a helpful guide to users.

In addition to expanding the feature set, future work should also look to remove detrimental and collinear features. Liu et al. 2022 used an initial LASSO regression model to select significant predictors of mortality. Similarly, Ashrafi et al. 2024a calculated the Variance Inflation Factors (VIFs) from a preliminary XGBoost model to identify collinearities, and they further culled their feature set through ablation studies. Both groups identified these measures as key steps toward optimizing their models, and future work should include rigorous feature selection strategies such as these.

REFERENCES

[1] Negin Ashrafi, Armin Abdollahi, Jiahong Zhang, and Maryam Pishgar. 2024. Optimizing Mortality Prediction for ICU Heart Failure Patients Leveraging XGBoost and Advanced Machine Learning with the MIMIC-III Database. *arXiv preprint arXiv:2409.01685* (2024). https://arxiv.org/abs/2409.01685

[2] Negin Ashrafi, Yiming Liu, Xin Xu, Yingqi Wang, Zhiyuan Zhao, and Maryam Pishgar. 2024. Deep learning model utilization for mortality prediction in mechanically ventilated ICU patients. *Informatics in Medicine Unlocked* 49 (2024), 101562. https://doi.org/10.1016/j.imu.2024.101562

[3] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. In *Proceedings of the ACM Conference on Health, Inference, and Learning (CHIL) Workshop*. ACM, Toronto, ON, Canada, 9. https://arxiv.org/abs/1904.05342

[4] Alistair E. W. 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. 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data* 3 (2016), 160035. https://doi.org/10.1038/sdata.2016.35

[5] Ran Liu, Haiwang Liu, Ling Li, Zhixue Wang, and Yan Li. 2022. Predicting in-hospital mortality for MIMIC-III patients: A nomogram combined with SOFA score. *Medicine* 101, 42 (2022), e31251. https://doi.org/10.1097/MD.0000000000031251