A Comparison and Analysis of Explainable Clinical Decision Making Using White Box and Black Box Models

by

Liam Dingle

Lakehead University

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in Computer Science (Specialization in Artificial Intelligence)

Thunder Bay, Ontario, Canada, 2023

A Comparison and Analysis of Explainable Clinical Decision Making Using White Box and Black Box Models

by

Liam Dingle

Lakehead University

Supervisory Committee

Dr. Yimin Yang, Supervisor
(Department of Electrical and Computer Engineering, University of Western Ontario, Canada)

Dr. Ashirbani Saha, Co-Supervisor
(School of Biomedical Engineering, McMaster University, Canada)

Dr. Ruizhong Wei, Co-Supervisor
(Department of Computer Science, Lakehead University, Canada)

Dr. Thangarajah Akilan, External Examiner
(Department of Engineering, Lakehead University, Canada)

Dr. Amin Safaei, External Examiner
(Department of Computer Science, Lakehead University, Canada)

Abstract

Explainability is a crucial element of machine learning-based making in high stake scenarios such as risk assessment in criminal justice [80], climate modeling [79], disaster response [82], education [81] and critical care. There currently exists a performance tradeoff between low-complexity machine learning models capable of making predictions that are inherently interpretable (white box) to a human, and cutting-edge high complexity (black box) models are not readily interpretable.

In this thesis we first aim to assess the reliability of the predictions made by black box models. We train a series of machine learning models on an ICU (Intensive Care Unit) outcome prediction task on the MIMIC III dataset. We perform a comparison of the predictions made by white box models and their black box counterparts by contrasting explainable model feature coefficients/importances to feature importance values generated by a post-hoc SHAP (SHapley Additive exPlanation) values. We then validate our results with a panel of clinical experts. The first study shows that both black box and white box models prioritize clinically relevant variables when making outcome predictions. Higher performing models showed prioritizations to more clinically relevant variables than lower performing models. The black box models show better overall performance than the white box models.

In our second study, we aim to test the reliability of the generated explanations made by both SHAP and explainable model importances. We assess the performance impact of training machine learning models on different subsets of features created by ranking features based on their importances. The performance assessment is conducted on the same outcome prediction task, along with binary classification tasks performed on an additional three clinical datasets. The second study shows that there is a tangible performance impact between classifications made on important compared to unimportant data for a subset of higher performing models on certain datasets.

In the first study, we can conclude that black box models offer tangible performance improvements (0.0185 increase in AUROC score) while also prioritizing clinically relevant features in an outcome prediction scenario. In the second study, we can conclude that the features labelled as important through different model explanations directly impact the performance of the models, and therefore actually represent the features that are

important to the model's decision making. Higher performing models were shown to select small subsets of features (1%-5% of the total feature set) that resulted in similar (+/- 0.02 AUROC score) performance from using 100% of the dataset features.

Overall, we can conclude that the studies show evidence that implementation of black box models in high-stakes decision making can offer tangible benefits in performance. We can additionally conclude that in certain settings, black box models can provide reliable, semi-transparent predictions if proper explainability mechanisms are put in place and validated. To improve the rigor and further validate the findings, more work is needed to test additional clinical datasets in a similar framework. Furthermore, we would advocate for the further exploration of feature importances as a tool for dimensionality reduction due to the promising results shown in this work.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to extend a sincere thank you to the following individuals:

Dr. Yimin Yang, for your patience, guidance, and positivity.

Dr. Ashirbani Saha, for your flexibility, broad skillset, and the large volumes of extremely helpful feedback.

Regan MacGillivary and team, for your expertise and inhumanely fast response times.

Keiko Larocque, for your tireless encouragement and unwavering compassion.

1 Introduction

1.1 Overview & Motivation

The widespread adoption of and increased availability of robust EHR (Electronic Health Record) data has revolutionized how data driven methodologies can be applied to problem solving in the medical space [56], [67]. The quality and quantity of available data has reached a point where modern machine learning is able to be applied to a large variety of classification, forecasting, and exploratory problems. Machine learning solutions are being applied in many sub-domains of healthcare to increase the throughput and efficiency of clinical decision making [84], [85], [86]. Through the automation and/or enrichment of the decision-making process with data driven models, more informed decisions can be made in a more responsive manner which can maximize the potential for positive patient outcomes.

The application of machine learning in medicine, and other mission critical settings has had mixed responses from domain experts [54]. A common concern that supersedes the promising performance of machine learning models transparency provided by data driven models when making predictions. In a mission critical setting, it is crucial to not only provide an accurate prediction, but also be able to provide supplementary information on how that prediction was made.

Machine learning models fall under two distinct classes: Explainable (white box) or black box. White box models are easily interpretable. Their simplified structure and learning algorithm facilitate predictions that are more transparent, and easier to understand. A human is much more likely to inherently understand how an explainable model created its prediction. Black box models are created directly from data by an algorithm, meaning that humans cannot understand how variables are being combined to make predictions [55]. Even if one has context on the input features, black box predictive models can have complex mappings of the feature space from input to decision output that no human can understand.

Though it may be an easy workaround to only apply inherently explainable models to mission critical problems, black box models have been shown to outperform their explainable counterparts in most use cases where datasets are of sufficient size [19], [57], [58], [69]. This is primarily due to their ability to better fit to nonlinear decision boundaries which is common in real-world, imperfect data. This increase in performance has led to an increasing focus on the application of black box models (e.g. Deep learning Models) in clinical settings [58], [59], [60], [61].

While having very promising performances, one aspect quite lacking in the current body of research using black box models is a well-rounded assessment on how reliable the predictions are from these models, and how models prioritize features in their predictions.

In the first study, we will elucidate further the reliability of Deep learning model predictions on EHR data by first creating a lens of explainability within each black box approach using SHAP values as a post-hoc explainability framework (Chapter 4). A variety of models are trained, tested, and validated on EHR data sourced from the MIMIC III dataset. A Logistic Regression [1] classifier used as a white box machine learning model, and a series of black box machine learning models such as a Suppport Vector Machine (SVM) [9], Random Forest [5], Deep Neural Network (DNN) [13], Long Term Short Term Memory (LSTM) [21], 1D Convolutional Neural Network (1D CNN) [25] and a Deconvolution Convolution Neural Network [68] are used in a benchmarking task of mortality prediction [57] . This study contrasts explainable machine learning methods to the values of the post-hoc explainability process values of each black box model.

In the next study (Chapter 5), we seek to understand the representativeness of the explainability values (e.g. feature importances). Classification tasks are performed on four clinical datasets: MIMIC III, Sepsis Minimal Clinical Records, Diabetes 130-US Hospitals for Years 1999-2008 and the Breast Cancer Wisconsin dataset. Iterations of the datasets are modified to only include features with importance scores in the top 1%, 5%, 10%, 25% and 50%. Additional dataset iterations are created using all features not included in each respective dataset split. A set of applicable models are trained on each dataset split. Model performances for each dataset are compared to one another to indicate whether the higher importance values positively impact model performance.

1.2 Problem Description

ICU outcome prediction describes the process of using patient data to predict the outcomes of patients in the ICU. These predictions can include various clinical endpoints, such as mortality, length of ICU stay, need for mechanical ventilation, or likelihood of readmission. It is a fundamental component of critical care medicine and health informatics.

ICU outcome prediction allows for the early identification of high-risk patients, enabling healthcare providers to allocate resources more effectively and to implement preventive strategies or interventions where needed [73], [74]. ICU outcome predictions can also assist clinicians in making informed decisions about the patient's care and can aid in discussions about prognosis with patients and their families [75]. Finally, these predictions can also be useful for benchmarking and quality improvement purposes through comparing predicted outcomes with actual outcomes, hospitals can assess the quality of their ICU care and identify areas for improvement [76].

Explainability of the predicted outcomes is crucial. Clinicians are more likely to trust and therefore use predictions if they understand how the predictions are derived [77]. Additionally, explainability can facilitate clear and concise discussions with patients and their families by providing easily digestible justification and context for how a decision was made, why it was made, and what implications that has for the patient. This is crucial in high stakes decisions such as end-of-life care because something as simple as ‘the model decided’ is not sufficient or adequate justification.

Moreover, in the context of machine learning, explainable predictions can help to identify biases and errors in the model, contributing to the development of more accurate and fair prediction tools [78].

1.3 Contribution

Based on a literature review outlined in (Chapter 2.2), there are two primary areas that are lacking from the current body of research: Lack of breadth/contrast and lack of validation of the underlying feature importances (e.g. explainability frameworks). The thesis aims to contribute as the follow two aspects:

Firstly, existing research examines a single model, or small subset of models through a single explainability framework (e.g. SHAP values). This work aims to provide a more comprehensive view by contrasting explanations generated by white box models such as Logistic Regression, compared to post-hoc explanations generated on black box models. Additionally, this work analyzes performance and feature importances from black box and white box models to provide a more unified view.

Secondly, existing research validates feature importances through a single lens. This lens tends to be a subjective/qualitative evaluation from a domain expert in the task being performed (e.g. Intensivist, Surgeon). This validation is usually paired with evidence from previously published literature that supports what the model should prioritize. This work provides validation through an additional lens beyond domain expert validation, which is an analysis of the model performance on important subsets of features. This will provide validation that the features models consider important are the features that impact performance the most, and by association the subset of features that most comprehensively define the outcome of the patient.

This work will seek to provide additional insight to answer three fundamental questions:

a. Can black box models be reliably explained using explainability frameworks when being applied in a clinical setting?
b. Are black box models viable in a clinical setting despite being less interpretable?
c. Do the feature prioritizations of white box or black box machine learning models align with clinical expertise?

1.4 Organization

Chapter 1 (Introduction) provides a high-level overview of the problem this work is trying to address, the rationale behind solving the problem, and the contributions this research provides.

Chapter 2 (Background and Related Work) provides additional context on the relevant models, data processing, and evaluation metrics implemented in the methodologies for the studies covered in Chapter 4, and Chapter 5. Additionally, Chapter 2 gives an introduction and profile to of the four datasets used in our Studies. Finally, Chapter 2 also provides a Literature Review used to frame and support the gaps in research and the decisions made in forming the methodology.

Chapter 3 (Analyzing Explainable Mortality Predictions of Black Box Deep Learning Models) examines feature importances generated by white box models, and post-hoc explainability frameworks applied to black box models. It shows evaluations for model performance and assesses the consensus of the important features generated from high performing models.

Chapter 4 (Analysis of Post Hoc Explainability in Clinical Decision Making) evaluates model performance on subsets of features dictated by the importance values generated from white box models and post hoc explainability frameworks applied to black box models. This serves as a proxy assessment for the reliability of the feature importance values.

Chapter 5 (Conclusion and Future Work) highlights the tangible conclusions that can be drawn from the work highlighted in Chapters 3 and 4. Additionally, this chapter highlights potential opportunities for future work to further develop the rigor of the conclusions made, and potential additional opportunities with applying parts of the methodology to other areas of machine learning.

2 Background and Related Work

2.1 Background

In this chapter, we provide an overview of the core concepts relevant to our work. We provide conceptual overviews and mathematical definitions for each topic to provide further context to the model training, model evaluation, model explaining, and model selection used in our methodologies in the subsequent chapters.

2.1.1 Models

In this chapter, we provide an overview of each model used in our experiments, and their mathematical definitions.

2.1.1.1 Black Box Model

Black box models, on the other hand, are machine learning models that are complex and difficult for humans to interpret. These models are called "black boxes" because their

internal workings are not fully understood; you can see what goes in and what comes out, but not how the input is transformed into the output. Examples of black box models include neural networks, support vector machines, and ensemble methods such as random forests and gradient boosting.

The primary advantage of black box models is their ability to model complex, non-linear relationships in large datasets, which can result in superior predictive performance. They are often used in applications where predictive accuracy is more important than interpretability, such as image recognition or voice recognition. [53] However, a major drawback of black box models is their lack of interpretability. This makes it difficult to understand why the model is making a certain prediction, which can be problematic in applications where understanding the decision-making process is important.

2.1.1.2 White Box Model

White box models, also known as interpretable or explainable models, are machine learning models whose internal workings are understood and interpretable by humans. These models typically have transparent decision-making processes, which means that for any given input to the model, it is possible to understand how and why the model produced its output. Examples of white box models include linear regression, logistic regression, and decision trees.

The primary advantage of white box models is their interpretability and transparency [40]. This makes them particularly suitable for applications where understanding the decision-making process is crucial, such as healthcare and finance. However, a major drawback of white box models is that they may not be able to capture complex patterns or relationships in the data as effectively as some black box models. They may therefore be less accurate in certain situations.

2.1.1.3 Logistic Regression

Logistic Regression is a statistical method that was initially used in the discipline of social sciences during the 19th century. It was introduced by statistician David Cox in 1958 as an extension of regression analysis and the logistic model. Its applicability quickly expanded into various fields, including machine learning, to predict binary outcomes [1].

The logistic regression model utilizes a logistic function to model a binary dependent variable. It is based on the odds ratios, which can be interpreted as probabilities. The output is a value between 0 and 1, representing the predicted probability of the positive class. The logistic function is also known as the sigmoid function, due to its "S" shape [2].

The logistic function, also known as the sigmoid function, is defined as:

$$\sigma(z) = \frac{1}{1+e^{-z}}, \quad (1)$$

where z is a linear combination of features defined as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n, \quad (2)$$

where β and x refer to model parameters and input features respectively.

Logistic regression uses the maximum likelihood estimation method to estimate the model parameters. This process iteratively adjusts the weights of the features, attempting to find the set of weights that maximizes the likelihood of producing the observed data. The learning process often uses a method called gradient descent to find these optimal weights [3].

The goal of gradient descent is to find the set of parameters that minimizes the cost function J(β). The cost function (log loss) can be defined as:

$$J(\beta) = -\frac{1}{m}\sum_{i=1}^{m}[y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)], \quad (3)$$

where J(β), m, y, $\hat{y}$ refers to the log loss cost function, training sample, true class label, predicted probability of class label respectively.

In gradient descent, the model parameters are updated using the chain rule, defined as:

$$\beta_i := \beta_i - \alpha \frac{\partial}{\partial \beta_i} J(\beta),$$

(4)

where α, $\frac{\partial}{\partial \beta_i} J(\beta)$ refers to the learning rate and partial derivative of the cost function with respect to β_i respectively.

The logistic regression model is considered a highly interpretable or a white box model. The weights for the features are calculated in a manner that represents the log odds of the positive class, which can be easily interpreted [4].

2.1.1.4 Random Forest

The Random Forest algorithm was introduced by Leo Breiman in 2001. It is an ensemble method that builds upon the concept of decision trees, aiming to address their high variance [5].

Random Forest is an ensemble of decision trees, which are constructed by repeatedly selecting random subsets of the data and the features. The final prediction is typically the mode (classification) or mean (regression) of the individual trees' predictions. By leveraging the power of multiple decision trees, the model aims to reduce overfitting and improve prediction accuracy [6].

The ensemble of decision trees can be denoted as:

$$RF = \{T_1, T_2, \ldots, T_n\},$$

(5)

where RF and T refer to the random forest and an instance of a decision tree respectively.

Each decision tree in the Random Forest is trained independently. For each tree, a bootstrap sample is drawn from the dataset. The tree is then grown by splitting on features, chosen from a random subset of the total features.

The random subset can be denoted as:

$$F_m \subset F,$$

(6)

where F is the set of all features.

The best split is typically determined by Gini impurity for classification. This process is repeated for each tree in the ensemble [7].

The Gini impurity is defined as:

$$G(S) = 1 - \sum_{i=1}^{K} p^2{}_i,$$

(7)

where G(S), S, K and p_i refers to the Gini impurity, set of samples in a node, number of classes and fraction of samples belonging to class i in S.

The Random Forest classification prediction is made by calculating the mode from all the decision trees (also referred to as majority voting).

The calculation can be defined as:

$$C = mode\{T_1(x), T_2(x), \ldots, T_n(x)\},$$

(8)

where C, T and x refer to the classification, decision tree, and dataset sample respectively.

Random Forest is considered a relatively interpretable model, as it is based on decision trees. Feature importance can be determined based on the reduction in impurity achieved by each feature. Despite its inherent explainability, it is less transparent than simpler models like linear or logistic regression because it is an ensemble of multiple models, and understanding the collective decision of multiple trees can be challenging [8] and is considered a black box model.

2.1.1.5 Support Vector Machine

Support Vector Machines (SVMs) were introduced by Vladimir Vapnik and Alexey Chervonenkis in the 1960s as part of the development of the VC (Vapnik–Chervonenkis) theory. The current standard form of the SVM, including the use of the kernel trick, was developed during the 1990s [9].

SVMs operate by mapping the input data to a high-dimensional feature space and finding the hyperplane that maximally separates the classes.

The hyperplane can be defined as:

$$w \cdot x + b = 0,$$

(9)

where w, x, and b refer to the weight vector, feature vector and bias respectively.

This hyperplane is determined by the so-called support vectors, which are data points that lie closest to the decision boundary. The SVM can perform non-linear classification using the kernel trick, implicitly mapping inputs into high-dimensional feature spaces [10].

SVMs are trained by solving a quadratic programming problem to find the support vectors and the maximum-margin hyperplane.

The optimization problem can be solved using Lagrangian multipliers to find the minimum of the Lagrangian defined as:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w \cdot x + b) - 1],$$

(10)

where y, α are sample labels and Lagrange multipliers respectively.

This is a convex optimization problem, and therefore any local solution is also a global solution. The dual problem, which is often easier to solve, involves only the dot products of the inputs, allowing the use of the kernel trick [11].

The dual problem is defined as:

$$max_{\alpha} \sum_{i=1}^{N} \alpha_i - \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \ni \sum_{i=1}^{N} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \quad (11)$$

where y, α, x, C are sample labels, Lagrange multipliers, feature vector, and regularization parameter respectively.

While SVMs can provide a decision boundary and the support vectors, they are generally considered as less interpretable models. The high-dimensional feature space and the potentially complex decision boundaries can make it hard to interpret the model, particularly in the case of non-linear SVMs [12] and therefore, SVM is considered as a black box model.

2.1.1.6 Neural Network

The concept of artificial neural networks was introduced by Warren McCulloch and Walter Pitts in 1943. They were inspired by biological neurons and aimed to build computational models that could mimic brain function. However, the development and popularization of modern multi-layer perceptrons and backpropagation did not happen until the 1980s, due to the work of researchers like Geoffrey Hinton [13].

A neural network consists of artificial neurons, or nodes, organized into layers. Each node in a layer receives inputs from the previous layer, applies a weighted sum (dot product), and then passes this sum through a non-linear activation function. The final layer provides the output of the network. The complexity and expressivity of neural networks comes from their depth (multiple layers) and width (multiple nodes per layer) [14].

Neural networks are typically trained using a method called backpropagation and stochastic gradient descent. The weights are initialized randomly and then iteratively

updated to minimize a loss function. The backpropagation algorithm computes the gradient of the loss function with respect to the weights, and these gradients are used to update the weights [15].

Generally, neural networks are considered "black box" models. They can achieve high performance on many tasks, but the complex interactions between nodes and layers make it challenging to understand how they derive their predictions. However, research is being conducted in the field of explainable AI to improve the interpretability of these models [16].

2.1.1.6.1 Backpropagation

Backpropagation, short for "backward propagation of errors," is a method used in artificial neural networks to calculate the gradient of the loss function with respect to the weights in the network [14], and it fundamentally revolutionized the training of neural networks, making deep networks practical to train.

The process of backpropagation consists of two passes through the different layers of the network: a forward pass and a backward pass. During the forward pass, the inputs are propagated from the input layer through the network, and the output of the forward pass is used to compute the loss function.

The forward propagation of a neural network can be defined as:

$$a_j = f\left(\sum_i w_{ij} a_i + b_j\right),$$

(12)

where a_j, f, a_i, w, b refers to the current layer output, activation function, previous layer output, weight and bias respectively.

For this work, the loss function for all models is the categorical binary cross-entropy loss. The loss function can be defined as:

$$J(W,b) = -\frac{1}{m}\sum_{i=1}^{m}[y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)],$$

(13)

where W, b, m, y, and $\hat{y}$ refers to the weight matrix, bias, number of training samples, actual label and predicted probability of the label respectively.

In the backward pass, which gives the algorithm its name, the algorithm calculates the gradient of the loss function with respect to each weight in the network by propagating the gradients backward through the network, hence the term backpropagation. Specifically, the partial derivative of the loss function with respect to a weight is computed by chain rule, which involves multiplying several terms together: the derivative of the loss function with respect to the output of the neuron that the weight comes from, the derivative of that output with respect to its total input, and the derivative of the total input with respect to the weight itself. [27]

We define the gradients with respect to the weights and bias respectively as $\frac{\partial J}{\partial W_i}$, and $\frac{\partial J}{\partial b_i}$.

The gradient of the error, due to the properties of the sigmoid function, can be simplified to:

$$\delta = \hat{y} - y,$$

(14)

where δ, y and $\hat{y}$ refers to the error, actual label and predicted probability of the label respectively.

The backward propagation through the network can be defined as:

$$\delta_i = \delta_{i+1}(W_{i+1})^T * f'(z_i),$$

(15)

where δ, W, f, and z refer to error, weight, loss function, and layer output respectively.

Backpropagation is used in the training process of neural networks to minimize the loss function, usually via an optimization algorithm like stochastic gradient descent (SGD). By calculating the gradients of the loss function with respect to the weights, it allows the learning algorithm to know how to adjust the weights to improve the network's performance (minimize the loss function).

Stochastic gradient descent can be defined as:

$$\theta = \theta - \alpha \nabla_\theta J(\theta; x^i, y^i),$$

(16)

where θ, α, J, x, and y refer to parameters, learning rate, loss function, input, and output label respectively.

The update rule for stochastic gradient descent can be defined as:

$$W_i = W_i - \alpha \frac{\partial J}{\partial W_i},$$

(17)

where w, and α refer to weight and learning rate respectively.

$$b_i = b_i - \alpha \frac{\partial J}{\partial b_i},$$

(18)

where b, and α refer to bias and learning rate respectively.

Backpropagation is important because it allows neural networks to learn from mistakes. By identifying where the network is making the largest errors, backpropagation provides a way of 'directing' the learning to focus on those areas. This method is efficient compared to brute force methods because it allows the network to focus the computational effort in the areas where it is most needed.

2.1.1.7 1D Convolutional Neural Network (1D CNN)

Convolutional Neural Networks (CNNs) are a class of deep neural networks, primarily developed by Yann LeCun in 1988 for handwritten and machine-printed character recognition. The concepts behind CNNs were inspired by the visual cortex and its hierarchy of receptive fields. While the 2D version of CNNs is used extensively in image processing, the 1D version is used for sequence data, such as time-series and sentences, and it became more popular with the rise of deep learning [17].

A 1D CNN, like its 2D counterpart, utilizes convolutional layers in its architecture. However, the convolution is performed across only one spatial dimension. The convolution operation involves a filter or kernel that is convolved with the input data to produce a feature map. This operation helps to extract local features from the sequence data. These local features are then combined in further layers to understand more complex patterns [18].

A 1D convolution can be defined as:

$$C[i] = \sum_{j=0}^{m-1} K[j] \cdot X[i+j], \quad (19)$$

where C[i], K, X refer to the ith element of the output sequence, kernel of size m, and input sequence of size n (larger than m).

1D CNNs are trained in a similar way to standard neural networks, using backpropagation and gradient descent. The loss function is defined according to the task (for example, cross-entropy for classification), and the weights of the model are updated to minimize this loss. The unique aspect of the training in CNNs is the use of shared weights in the convolutional layers [19].

While CNNs can provide some level of interpretability through visualization of the filters and feature maps, they are generally considered "black box" models like other neural networks. The interactions between layers and the function of individual filters can be hard to interpret, especially in deeper networks [20].

2.1.1.8 Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) model is a type of recurrent neural network (RNN) designed to learn long-term dependencies. It was introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997. LSTM was developed to overcome the limitations of traditional RNNs, specifically the vanishing gradient problem, which made it hard for standard RNNs to learn from information more than a few steps back in the input sequence [21].

LSTMs have a similar structure to standard RNNs, but they include a 'memory cell' that can maintain information in memory for long periods. Instead of neurons, LSTM networks have memory blocks that are connected through layers. A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output. A block operates upon an input sequence and the current state of the LSTM unit, which is updated via gating units [22].

The forget gate is defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f),$$

(20)

where σ, W_f, b_f, x_t, h_{t-1} refer to the sigmoid function, weight matrix for the forget gate, bias term for the forget gate current input, and previous hidden state respectively.

The input gate and candidate cell state are defined as:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

(21)

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C),$$

(22)

where W_i, W_C, b_i, and b_C refer to the weight matrix for the input gate, weight matrix for the cell state, bias for the input gate and bias for the cell state respectively.

The cell state update is defined as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t,$$

(23)

where f_t, C_{t-1}, i_t, and $\tilde{C}_t$ refer to the forget gate, previous cell state, input gate and candidate cell state respectively.

The output gate and hidden state is defined as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

(24)

$$h_t = o_t * \tanh(C_t),$$

(25)

where W_o, b_o, and C_t refer to the weight matrix for the output gate, the bias for the output gate and the cell state respectively.

LSTM networks are trained using a variant of backpropagation, called backpropagation through time (BPTT). BPTT involves unrolling the entire sequence and then applying the standard backpropagation algorithm (). However, LSTMs use their gating mechanisms during this process to control and manage gradient flow, and thereby mitigate the vanishing gradient problem [23].

Like other neural network models, LSTM networks are generally considered black box models. Although it's possible to understand their architecture and training mechanism, the underlying decisions behind their predictions are challenging to interpret due to their complex, recursive nature [24].

2.1.1.9 2D Convolutional Neural Network (2D CNN)

The 2D Convolutional Neural Network (2D CNN) model is a significant advancement in deep learning, primarily developed by Yann LeCun in 1988 for digit recognition in zip code reading systems. CNNs have been designed to learn spatial hierarchies of features automatically and adaptively from visual data [25].

A 2D CNN, like its 1D counterpart, employs convolutional layers, where the convolution is performed across two spatial dimensions. These convolution operations involve applying filters (or kernels) to the input data to generate feature maps. Each filter is designed to recognize a specific type of feature in the input, and these features are used by the subsequent layers to understand more complex patterns. 2D CNNs are widely used in image processing tasks due to their capability of capturing spatial relationships [26].

Training a 2D CNN involves defining a loss function according to the task (for example, cross-entropy for classification) and updating the model's weights to minimize this loss. This training process involves using backpropagation and gradient descent, similar to other neural networks. One characteristic aspect of CNNs is the use of shared weights in the convolutional layers, which reduces the amount of memory required and allows the network to learn more robust features [27].

2D CNNs, like other deep learning models, are typically regarded as "black box" models due to their complex interactions and transformations. However, some methods can help visualize what the model has learned, such as plotting the learned filters or using techniques like class activation mapping. Still, their decision-making process can be hard to interpret in comparison to more transparent models [28].

2.1.1.10 Deconvolution Convolution Network

The deconvolution convolution network uses deconvolution layers to convert 1D signals into 2D data. [67] This is a departure from traditional methods that rely on human pre-processing techniques, such as using the discrete Fourier transform (DFT) to convert 1D signals into 2D arrays. After the deconvolution process, the data is processed by a 2D Convolutional Neural Network (CNN), which identifies the data.

The model, like many deep learning models, is not inherently explainable. While the transformation of 1D data to 2D data via deconvolution layers might provide some insights into the data, the subsequent processing by the 2D CNN involves complex, non-linear transformations that are not easily interpretable. Therefore, while the model may be effective and efficient, it does not provide clear, understandable rules or criteria for its predictions.

2.1.2 Binary Classification Evaluation Metrics

This chapter gives an overview of the metrics utilized for evaluating the performance of models in a high-cost binary classification setting. Both classification tasks being performed in this thesis (outcome prediction, readmission prediction and cancer diagnosis) all have a higher cost for false positives. Furthermore, we provide an overview of the loss function utilized to train all of our models.

2.1.2.1 Precision

Precision, also known as the positive predictive value, measures the proportion of correctly predicted positive observations out of the total predicted positives. It is used when the cost of false positives is high.

Precision is calculated as follows:

$$p = \frac{TP}{TP+}, \tag{26}$$

where p, TP, FP stands for precision, true positives, and false positives respectively.

In scenarios such as spam email detection or disease prediction, precision is crucial as mislabeling can have serious consequences (e.g., marking an important email as spam or predicting a healthy person as diseased). In the case of outcome prediction, false positives have a higher overall impact than false negatives. A false positive would imply that the model would predict patient expiration when the patient would expire in an instant where the patient would remain living. This has severe implications for the type of care

administered, and ultimately would impact the well-being of the patient. A precision score of 100% indicates that the model can classify every expired patient as expired.

The following drawbacks are present with this metric: Precision alone does not tell us about the model's performance in identifying all actual positive cases as it does not consider False Negatives [29].

2.1.2.2 Recall

Recall (or Sensitivity or True Positive Rate) measures the proportion of actual positives that are correctly identified. It is used when the cost of false negatives is high.

Recall is calculated as follows:

$$r = \frac{TP}{TP+FN}, \quad (27)$$

where r, TP, FN stands for recall, true positives, and false negatives respectively.

For instance, in a disease prediction model, a higher recall would be preferred to ensure fewer cases of false negatives.

The following drawbacks are present with this metric: High recall can be achieved at the expense of more false positives, which might not be desirable in all situations [30]. A

2.1.2.3 Accuracy

Accuracy is the most intuitive performance measure. It is simply a ratio of correctly predicted observations to the total observations.

Accuracy is calculated as follows:

$$a = \frac{TP+T}{TP+FP+TN+F}, \quad (28)$$

where a, TP, TN, FP, FN stands for accuracy, true positives, true negatives, false positives, and false negatives respectively.

Accuracy is suitable when the target classes are balanced, and the costs of false positives and false negatives are approximately equal.

The following drawbacks are present with this metric: Accuracy is not a good measure when the classes are imbalanced. A model could achieve high accuracy by simply predicting the majority class [31]. For mission-critical applications such as the binary classification problems explored in this thesis, accuracy is not a suitable metric to draw conclusions from. There are class imbalances in some datasets being utilized (e.g. MIMIC). Furthermore, our set of binary classification problems do not have an equal cost for false negative and false positive predictions.

2.1.2.4 AUC-ROC (Area Under the Receiver Operating Characteristic)

AUC-ROC is a performance measurement for classification problems at various threshold settings. ROC is a probability curve, and AUC represents the degree or measure of separability, e.g., how much the model is capable of distinguishing between classes.

AUC-ROC is calculated as the area under the ROC curve which plots True Positive Rate (Recall) against the False Positive Rate for different threshold values.

It can be used in binary classification or multi-class classification (using One Vs Rest or One Vs One methods). It is more useful than accuracy, especially for imbalanced classes.

The following drawback is present with this metric: It can present an overly optimistic view of the model's performance if there are imbalanced classes [32]. Generally, AUC-ROC can be used with imabalanced classes if the minority class accounts for at least 10% of the total samples.

2.1.2.5 PR-AUC (Precision-Recall Area Under Curve)

PR-AUC, like AUC-ROC, provides a way to summarize the performance of a binary classification model, but it plots Precision against Recall for different threshold values.

PR-AUC is calculated as the area under the Precision-Recall curve.

PR-AUC is useful when dealing with imbalanced datasets. It focuses on the minority class.

The following drawbacks are present with this metric: PR-AUC can't be interpreted as a probability, unlike AUC-ROC. It is less interpretable and does not account for True Negatives [34].

2.1.2.6 Youden's Index

Youden's index (also known as Youden's J statistic) is a single statistic that captures the performance of a dichotomous diagnostic test. It was introduced by W.J. Youden in 1950 to evaluate the overall discriminative power of a diagnostic test [33].

Youden's index is a measure derived from the Receiver Operating Characteristic (ROC) curve. It's calculated as the maximum vertical distance between the ROC curve and the diagonal or chance line, which is equivalent to the maximum difference between the true positive rate (sensitivity) and the false positive rate (1-specificity).

Specificity can be calculated in the following manner:

$$x = \frac{TP}{TP+FN},$$

(29)

where x, TP, FN refers to specificity, true positives, and false negatives respectively.

Sensitivity can be calculated in the following manner:

$$y = \frac{TN}{TN+FP},$$

(30)

where y, TN, FP refers to specificity, true negatives, and false positives respectively.

Youden's index is then calculated in the following manner:

$$z = x + y - 1 \,,$$

(31)

where z, x, y refers to Youden's index, sensitivity and specificity respectively.

This index ranges from 0 to 1, where 0 represents a test with no discriminative power, and 1 represents a perfect test.

Youden's index is useful for determining an optimal cutoff point or decision boundary for a diagnostic test, particularly when the costs of false positives and false negatives are roughly equal. The optimal cutoff is the one that maximizes Youden's index [35].

The main advantage of Youden's index is its simplicity and its consideration of both sensitivity and specificity in its calculation, providing a balance between these two measures. However, it assumes that the costs of false positives and false negatives are equal, which is often not the case in real-world scenarios. Moreover, it doesn't consider the prevalence of the condition [35].

Youden's index has been widely used in medical research for choosing the optimal cutoff value for various diagnostic tests or risk prediction models. For instance, it has been applied in studies related to cancer detection, cardiovascular risk prediction, and various other clinical decision-making processes [36].

2.1.3 Explainability Values

In this chapter, we provide an overview of all the relevant mechanisms and frameworks we leverage to explain the predictions made by our series of white box, and black box models discussed previously.

2.1.3.1 SHAP Values

SHAP (SHapley Additive exPlanations) is a unified measure of feature importance that allocates the contribution of each feature to the prediction for each instance. The method was proposed by Lundberg and Lee in 2017, based on the concept of Shapley values from cooperative game theory [37].

In cooperative game theory, the Shapley value is a solution concept that assigns a payout to each player depending on their contribution to the total payout.

The contribution for player i in a cooperative can be defined as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \cdot \big(v(S \cup \{i\}) - v(S)\big),$$

(32)

where ϕ_i, S, $|S|$, v(S) is the Shapley value, subset of the set of all players N excluding player i, the number of players in S, and the value function for subset S.

In the context of machine learning, SHAP values interpret the contribution of each feature to the prediction for each instance.

The contribution for feature j at sample x can be defined as:

$$\text{SHAP}_j(x) = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \cdot \big(f_x(S \cup \{j\}) - f_x(S)\big),$$

(33)

where N, S, $|S|$, and $f_x(S)$ refer to the set of all features, subset of features excluding feature j, number of features in S, and the prediction of the model using the features in S for instance x respectively.

The SHAP value for a feature is the average contribution of that feature to the prediction output, across all possible subsets of features. It calculates the difference in the output when including a feature versus not including it, averaged over all possible feature

combinations. This ensures that the feature contributions sum up to the total prediction, allowing the contributions of all features to be visually displayed together [38].

SHAP values can be used whenever you want to interpret a machine learning model at both global and local levels. It helps in understanding the model's overall behavior and the reasons for individual predictions. It's especially useful for complex models like gradient boosting and neural networks, where interpretability is often a challenge [37].

The main advantage of SHAP values is their ability to provide a consistent and locally accurate attribution for each feature. It has the property of consistency, meaning if a model changes so that it relies more on a feature, the attributed importance for that feature should not decrease.

However, the calculation of SHAP values can be computationally expensive, especially for high-dimensional datasets, as it involves iterating over all combinations of features. Approximate and model-specific methods (like TreeSHAP for tree-based models) can make the computation more tractable [39].

SHAP values have been widely used in various fields that require model interpretability. They have been applied in credit scoring, healthcare (for patient risk prediction and disease diagnosis), natural language processing, and many other areas where understanding the reasoning of the model is important [40].

2.1.3.2 Logistic Regression Feature Coefficients

Logistic regression works by using the logit link function, which is the natural log of the odds of the dependent variable occurring, to establish a relationship between the dependent and independent variables [41].

Each independent variable in a logistic regression model has a corresponding coefficient, often denoted as β. These coefficients are calculated through a method called maximum likelihood estimation (MLE). The MLE is an iterative optimization algorithm that aims to find the set of coefficients that makes the observed data most probable [42].

In the context of logistic regression, the β coefficients represent the log-odds of the outcome for a one-unit change in the predictor. Thus, if we exponentiate these coefficients, we get the odds ratios.

For instance, if the β coefficient for a variable is 0.5, the odds ratio is $\exp(0.5) \approx 1.65$, indicating that for a one-unit increase in this variable, the odds of the outcome occurring increase by a factor of 1.65, assuming all other variables are held constant [43].

The coefficients in a logistic regression model explain how changes in the independent variables affect the probability of a particular outcome. In the realm of prediction, these coefficients allow us to quantify the impact of each feature on the outcome, making logistic regression a type of white box or explainable model [44].

The sign of the coefficient (positive or negative) represents the direction of the relationship with the target variable. A positive sign indicates that as the feature value increases, the model's log-odds of predicting the positive class increase, making it more likely that the model will predict the positive class. Conversely, a negative sign indicates that as the feature value increases, the model's log-odds of predicting the positive class decrease, making it more likely that the model will predict the negative class.

To make these coefficients even more interpretable, one common practice is to calculate the marginal effects of the predictors. Marginal effects represent the change in the predicted probability of the outcome for a one-unit change in the predictor, providing a direct link between changes in the predictor and the predicted probability of the outcome [45]. It is important to note that note that the calculated coefficients and their interpretation assume that the relationships being modeled are correctly specified and all relevant predictors are included in the model.

2.1.3.3 Random Forest Feature Importances

A significant benefit of Random Forest models is their ability to measure the importance of features, thus providing some level of interpretability despite being generally classified as a black box model.

Feature importances in Random Forest are calculated based on the average impurity decrease when nodes of a particular feature are split in the trees of the forest [7]. The impurity can be measured using different criteria such as Gini impurity or entropy. When a feature is used to split data in a tree, the impurity decrease resulting from the split is calculated, and this is done across all trees in the forest for that feature. The higher the impurity decreases (or equivalently, the higher the information gain), the more important the feature is considered to be.

Features that often appear high in the trees and that significantly improve the splits are deemed important. This is because these features contribute more to increasing the homogeneity of the resultant nodes and thereby improving the predictive power of the model [46].

Random Forest feature importances can be used to explain the impact of different features on the model predictions. By comparing the feature importance scores, we can determine which features are the most influential in predicting the outcome variable. However, these importances do not tell us the direction of the effect (positive or negative), just the magnitude of the impact of the feature on prediction.

One should be cautious though while interpreting these feature importance scores. They can be biased towards preferring variables with more categories [8]. Also, correlated features might have their importance diluted since they can be substitutable for each other.

2.1.4 Data Processing Techniques & Terminology

In this chapter, we provide an overview of the data processing techniques we used to transform the raw data from our datasets into data that can be input into our models.

2.1.4.1 Imputation

Imputation is a process of substituting missing data with estimated ones. There are several techniques for imputation, with the choice depending on the nature of our data and the reason for the missingness.

The simplest method is mean imputation, where missing values of a variable are replaced with the mean of the available cases. Another method is median imputation, where the median is used instead. These methods, however, do not reflect the uncertainty created by missing data.

A more advanced technique is multiple imputation, which involves creating multiple different imputations for each missing value, reflecting the uncertainty around the right value to impute. Each of these datasets is then analyzed separately, and the results are pooled to create an overall result [47].

Imputation methods are best applied when the data are missing at random or missing completely at random. Care should be taken in situations where data is not missing at random, as imputation might introduce bias.

2.1.4.2 Standardization

Standardization is a scaling technique where the values of a feature are scaled so that they have the properties of a standard normal distribution with μ=0 and σ=1, where μ is the mean (average) and σ is the standard deviation from the mean.

Standardization is performed by subtracting the mean and dividing by the standard deviation for each value of each feature. Once the standardization is done, all the features will have a mean of zero, standard deviation of one and therefore, the same scale [48].

Mathematically, it is performed as:

$$x' = \frac{x-\mu}{\sigma}, \tag{34}$$

where x' is the standardized value, x is the feature value, μ is the dataset mean for the feature, and σ is the dataset standard deviation for the feature.

Standardization is used when we want our features to be on the same scale. This is important for many machine learning algorithms like support vector machines (SVM) and

k-nearest neighbors (KNN) that calculate the distance between two data points. If one of the features has a broad range of values, the distance will be governed by this feature.

2.1.4.3 Normalization

Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling. Here, the value of the feature is scaled down between the range 0 and 1.

Mathematically, it is performed as:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (35)$$

where x' is the normalized value, and x is the feature value.

Normalization is a good technique to use when you do not know the distribution of your data or when you know the distribution is not Gaussian (a bell curve). Normalization is useful when your data has varied scales and the algorithm you are using does not make assumptions about the distribution of your data, such as neural networks [49].

2.1.4.4 Imbalanced Dataset

Imbalanced datasets are common in many domains, including medical diagnosis, spam filtering, and fraud detection. An imbalanced dataset is one where the classes are not represented equally. In a binary classification problem, for instance, we may have 100 instances of Class A and 10,000 instances of Class B. The class with the majority of instances (Class B in this case) is often referred to as the majority class, while the other is the minority class [50].

The presence of class imbalance can severely compromise the learning process, as most machine learning algorithms and performance metrics assume balanced class distributions and equal misclassification costs. This results in models that have good accuracy in the majority class but poor accuracy in the minority class, which is often the class of interest.

When training on an imbalanced dataset, a machine learning model can become biased towards the majority class, failing to correctly classify instances from the minority class. This is because the algorithm tries to optimize overall accuracy or error rate, which can be misleadingly high if the majority class is predicted well [51].

Several techniques have been proposed to address the problem of imbalanced datasets. These techniques can be broadly grouped into two categories: data-level methods and algorithm-level methods. Data-level methods, such as oversampling the minority class or undersampling the majority class, aim to balance the class distribution. Algorithm-level methods, on the other hand, aim to adapt the learning algorithm to the imbalanced data, for example, by modifying the algorithm's loss function [52].

2.2 Related Work

The related work chapter aims to summarize the peer reviewed literature that forms key components of the methodology, including the datasets. Furthermore, it provides framing for the gaps in current research identified in (chapter 1.3).

2.2.1 Datasets

This chapter introduces the datasets utilized to train our selected machine learning models. We provide a contextual overview of each dataset, and a profiling of certain attributes relevant to the specific prediction task.

2.2.1.1 *MIMIC III*

The MIMIC-III (Medical Information Mart for Intensive Care) [56][83] is a large, single-center database comprising detailed clinical information relating to patients admitted to critical care units at a large tertiary care hospital. It includes data such as vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more.

MIMIC-III contains data associated with 53,423 distinct hospital admissions for adult patients (aged 16 years or above) admitted to critical care units between 2001 and 2012, and 7870 neonates admitted between 2001 and 2008. The data covers 38,597 distinct adult patients and 49,785 hospital admissions. The median age of adult patients is 65.8 years,

and 55.9% of the patients are male. The in-hospital mortality rate is 11.5%. The median length of an ICU stay is 2.1 days, and the median length of a hospital stay is 6.9 days.

The open nature of the data allows clinical studies to be reproduced and improved in ways that would not otherwise be possible. The primary International Classification of Diseases (ICD-9) codes from the patient discharges are listed in the database. For example, the top three codes across hospital admissions for patients aged 16 years and above were 414.01 ('Coronary atherosclerosis of native coronary artery'), accounting for 7.1% of all hospital admissions.

2.2.1.2 Sepsis Survival Minimal Clinical Records

The Sepsis Survival Minimal Clinical Records dataset is a multi-center dataset with clinical information on 110341 patients suffering from Sepsis [70]. There are two distinct cohorts within the dataset: a Norwegian cohort for training and testing, and a South Korean cohort for validation. Each sample contains the age, sex, septic episode number, and the patient outcome as features. The dataset was designed to be used for a binary outcome prediction task.

The dataset contains a class imbalance. 7.37% of patients have expired outcomes, and 92.63% patients have non-expired outcomes (denoted in Figure 1). 47.38% of patients are Male, while 52.62% of patients are female (denoted in Figure 2). The median age of patients is 68 years. 57.05% of patients only have a single septic episode on record, while the remaining sample of patients have 2 or more septic episodes (denoted in Figure 3).

Figure 1: Sepsis Minimal Clinical Records Patient Outcomes.

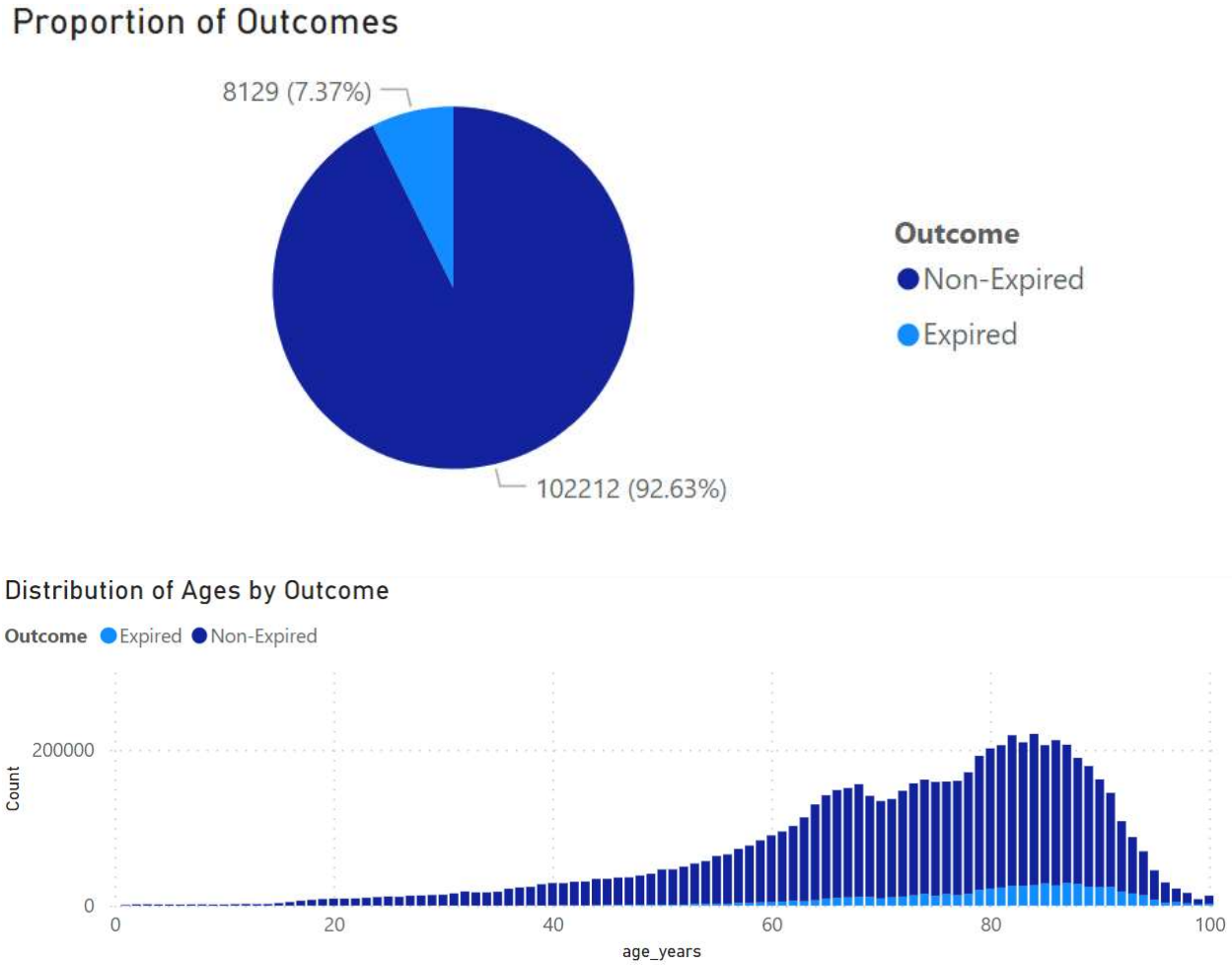


Figure 2: Sepsis Minimal Clinical Records Patient Gender.

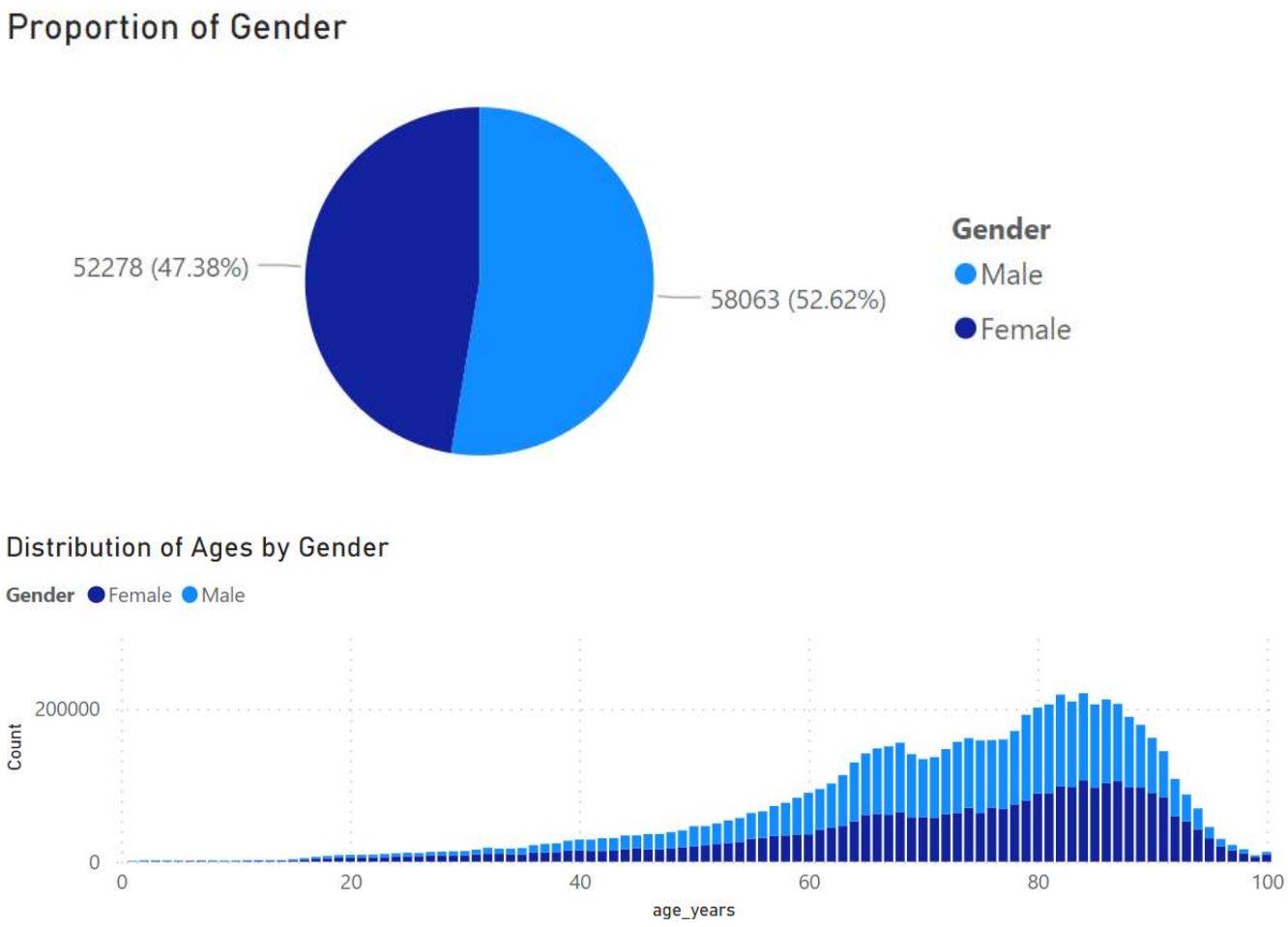
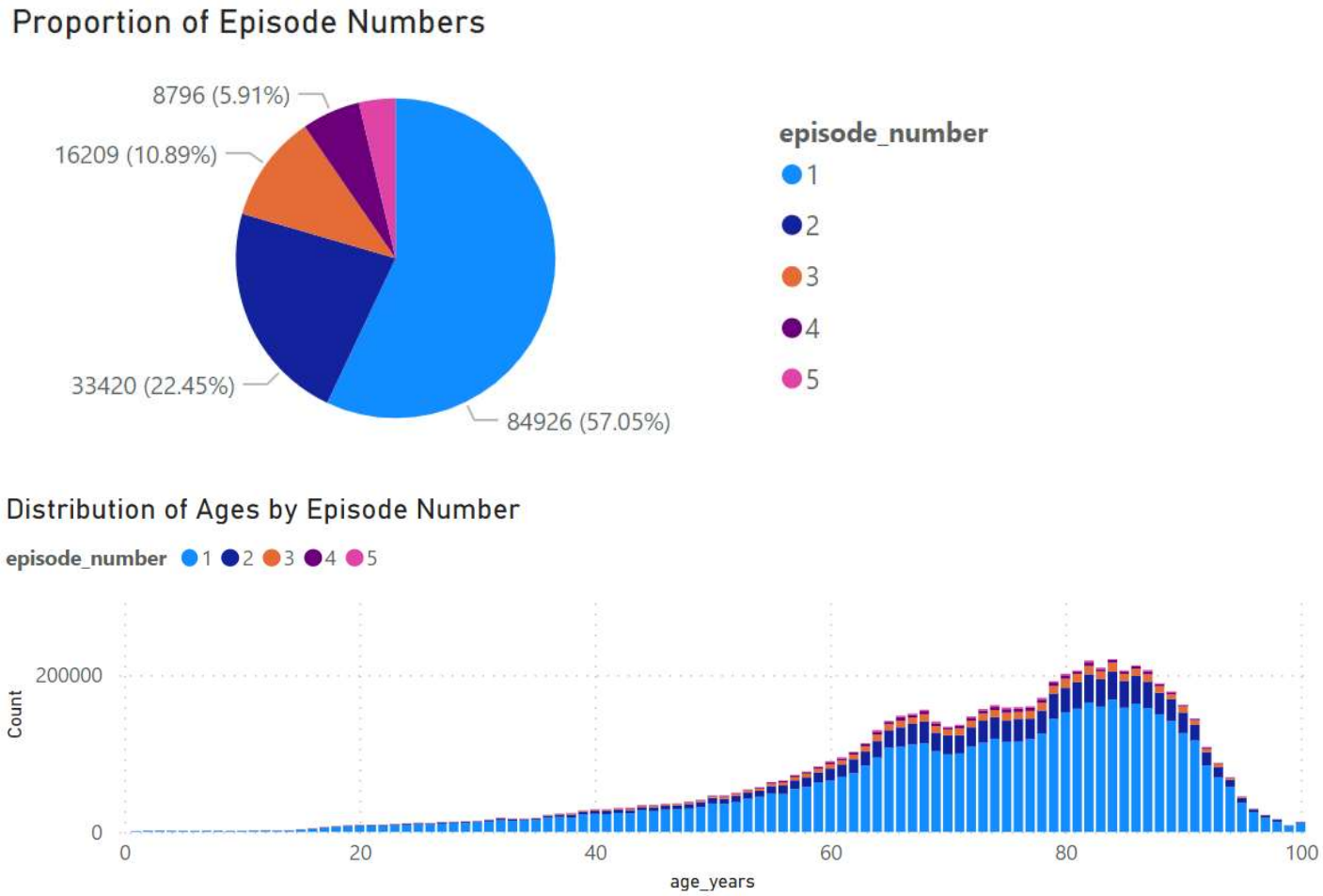


Figure 3: Sepsis Minimal Clinical Records Patient Episode Numbers



2.2.1.3 Diabetes 130-US hospitals for years 1999-2008

The Diabetes 130-US hospitals for years 1999-2008 dataset is a multi-center dataset with clinical data for 101766 hospital admissions across 130 American hospitals. Each sample represents a unique admission and contains a patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization.

Each sample satisfies the following constraints [71]:

1. The sample is an inpatient encounter (a hospital admission).

2. The sample is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.

3. The length of stay was at least 1 day and at most 14 days.

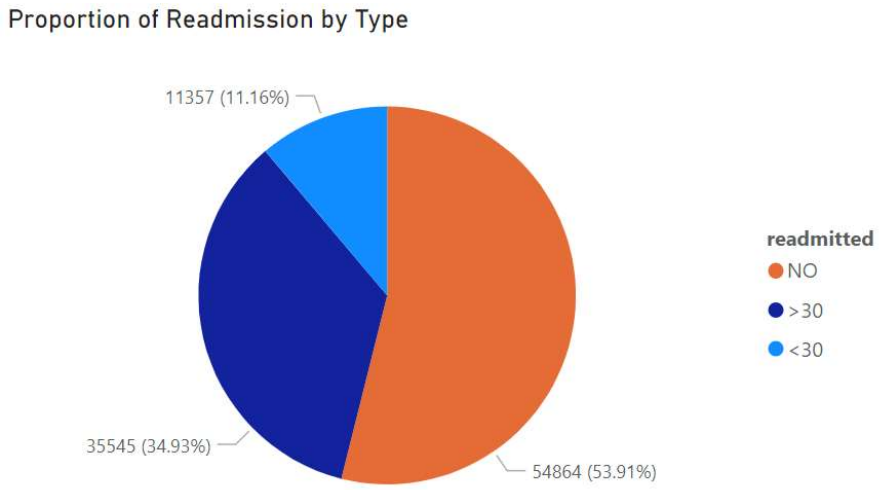
4. Laboratory tests were performed during the encounter.

5. Medications were administered during the encounter.

The dataset was designed to be used for predicting patient readmission.

53.91% of patients stays were not re-admission stays (denoted in Figure 4). 46.24% of patients are male, while 53.76% of patients are female (denoted in Figure 5). The majority (over 80%) of patient stays had patients aged 50-90 years old (denoted in Figure 6).

Figure 4: Diabetes 130-US Hospitals for Years 1999-2008 Patient Readmission Types.



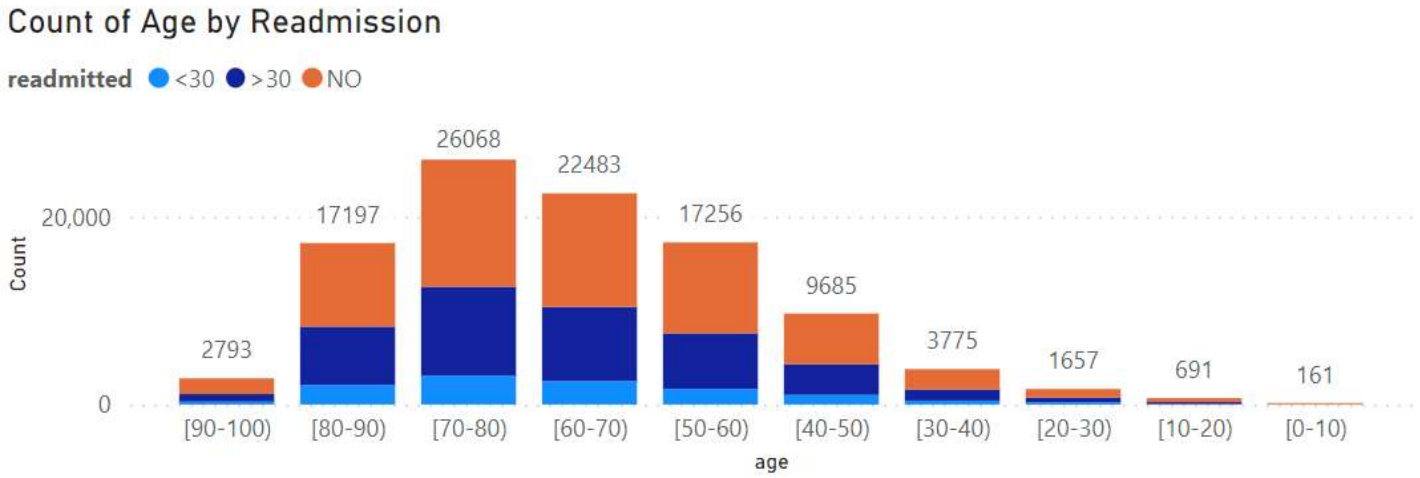
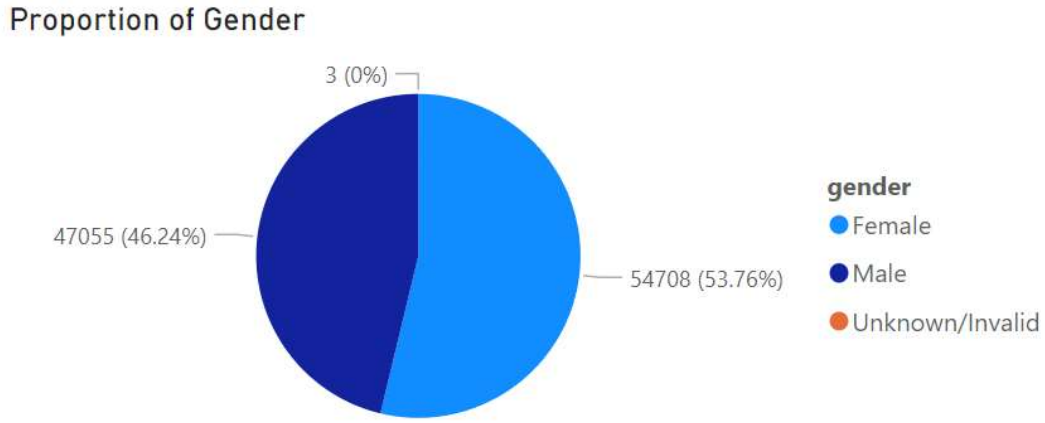


Figure 5: Diabetes 130-US Hospitals for Years 1999-2008 Patient Gender.



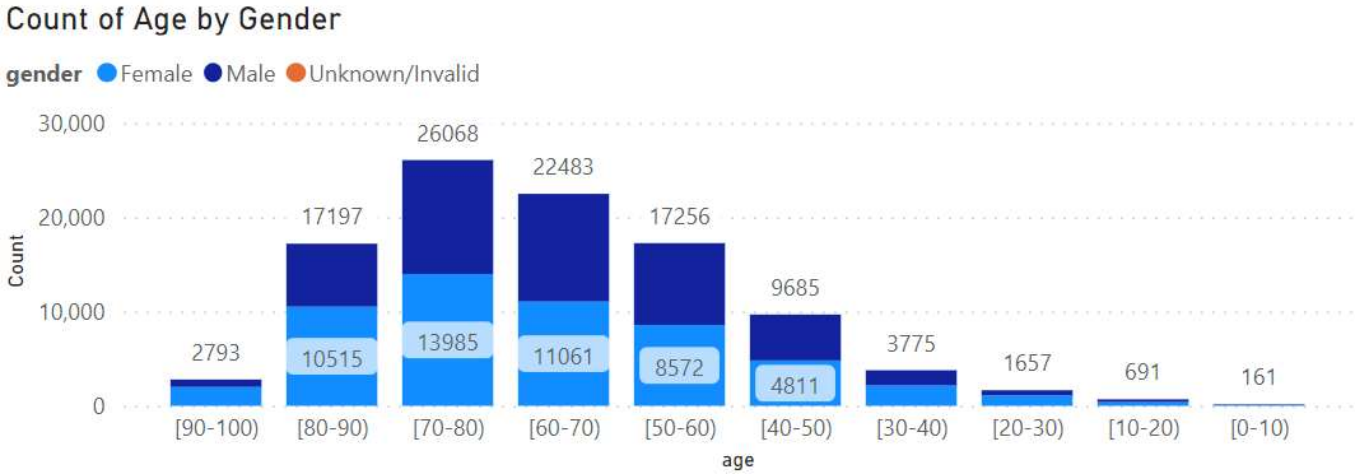
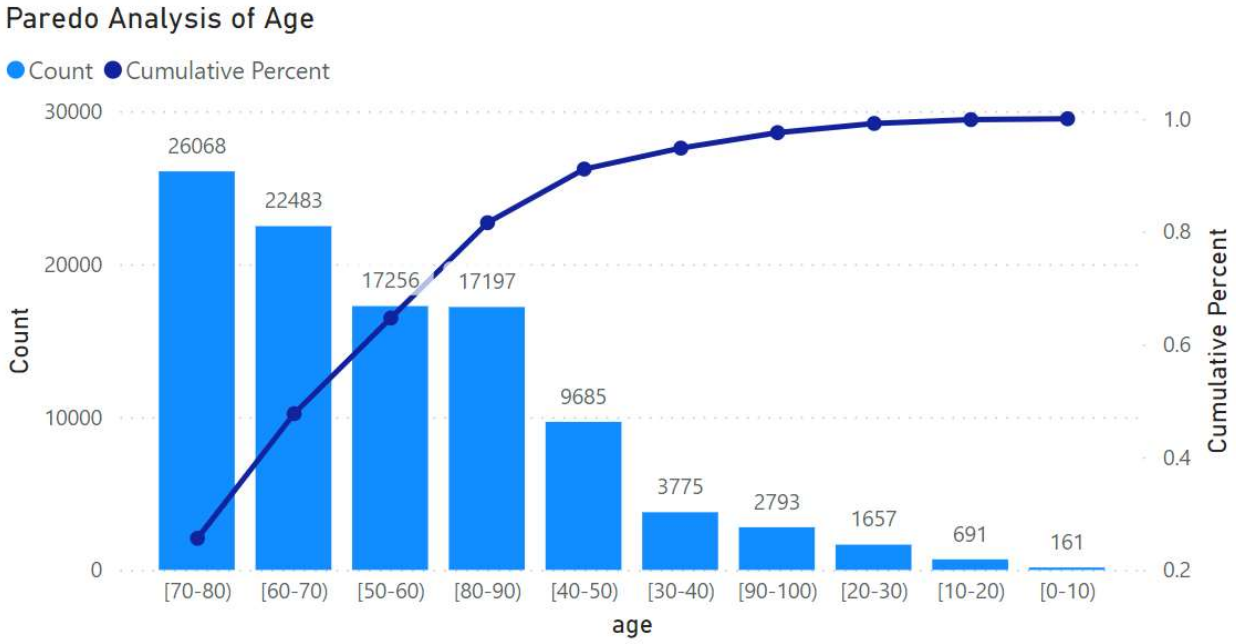


Figure 6: Diabetes 130-US Hospitals for Years 1999-2008 Patient Age.



2.2.1.4 Breast Cancer Wisconsin Dataset

The dataset primarily comprises features calculated from digitized images of a fine needle aspirate (FNA) of breast masses. These images are processed to extract detailed characteristics of cell nuclei present in the samples. [92]

Each sample in the dataset is described by 30 distinct features. These features encompass various aspects of the cell nuclei, such as their radius (mean of distances from the center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter^2 / area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, and fractal dimension ("coastline approximation" - 1). The comprehensive nature of these features makes the dataset particularly useful for fine-grained analysis in diagnostic procedures.

In terms of its structure, the dataset contains 569 samples. The classifies samples into two distinct classes: benign and malignant. Benign tumors are non-cancerous and generally considered less harmful, whereas malignant tumors are cancerous and potentially life-threatening. In the Breast Cancer Wisconsin Dataset, there's a slightly imbalanced

distribution between these two classes. The dataset contains 357 benign samples and 212 malignant samples.

2.2.2 Relevant Literature on EHR Data Processing

The objective of the paper "Multitask learning and benchmarking with clinical time series data" [57] was to propose four clinical prediction benchmarks using data derived from the publicly available Medical Information Mart for Intensive Care (MIMIC-III) database. The tasks covered a range of clinical problems including modeling risk of mortality, forecasting length of stay, detecting physiologic decline, and phenotype classification. The authors also aimed to evaluate the effect of deep supervision, multitask training, and data-specific architectural modifications on the performance of neural models.

The methodology involved compiling a subset of the MIMIC-III database containing more than 31 million clinical events corresponding to 17 clinical variables. These events covered 42,276 ICU stays of 33,798 unique patients. The four benchmark tasks defined were in-hospital mortality prediction, decompensation prediction, length-of-stay prediction, and phenotype classification. The authors developed linear regression models and multiple neural architectures for these tasks, including a basic LSTM-based neural network (standard LSTM) and a modified version (channel-wise LSTM). They performed experiments with these models and evaluated them on the test sets of the corresponding tasks.

The results showed that LSTM-based models outperformed linear models significantly across all metrics on every task. Channel-wise LSTMs performed significantly better than standard LSTMs for all four tasks, while multitasking helped for all tasks except phenotyping. Deep supervision with replicated targets did not help for in-hospital mortality prediction, but it did help for decompensation and length-of-stay prediction tasks. The best performing models for these tasks were channel-wise LSTMs with deep supervision.

From these results, the authors drew several insights. They proposed standardized benchmarks for researchers interested in clinical data problems and demonstrated that

LSTM-based models significantly outperformed linear models. They showed the advantages of using channel-wise LSTMs and learning to predict multiple tasks using a single neural model. They found that the phenotyping and length-of-stay prediction tasks were more challenging and required larger model architectures than mortality and decompensation prediction tasks. They also noted that the data in MIMIC-III, being generated within a single EHR system, might contain systematic biases, suggesting the need for future studies to explore how models trained on these benchmarks generalize to other clinical datasets.

2.2.3 Application of Black Box Models to EHR Related Prediction Tasks

2.2.3.1 LSTM

The paper "Scalable and accurate deep learning for electronic health records" [58] aims to demonstrate that deep learning models, which incorporate the entire raw electronic health record (EHR) data, can accurately predict multiple medical events from multiple centers without site-specific data harmonization.

The authors believe that using the raw EHR data in its entirety, rather than extracting and curating selected predictor variables, can enable scalable and accurate predictive models.

The methodology involves obtaining de-identified EHR data from two large US academic medical centers, including all data from 216,221 adult hospitalizations. The EHR data is represented using the Fast Healthcare Interoperability Resources (FHIR) format, which retains the raw data without harmonization. The authors developed deep learning models, including recurrent neural networks and attention-based neural networks, to predict outcomes like in-hospital mortality, 30-day readmission, length of stay, and discharge diagnoses. The prediction tasks being performed are in-hospital mortality, 30-day unplanned readmission, prolonged length of stay, and all a patient's final discharge diagnoses.

The data pre-processing steps include extracting multivariate time series features from raw datasets, normalizing all input variables to 0 mean and 1 standard deviation, using

masking to indicate which variables are missing at each time step, and calculating time interval to indicate how long each variable has been missing.

The main conclusions are that the deep learning models achieved high accuracy across sites for predicting in-hospital mortality (AUROC 0.93-0.94), 30-day readmission (AUROC 0.75-0.76), and length of stay (AUROC 0.85-0.86). The models outperformed traditional clinically used predictive models in all cases. The models were able to infer discharge diagnoses with high accuracy (Micro-F1 0.41-0.40), which could enable new clinical applications. The models were able to identify the most relevant parts of patients' raw EHR data for making predictions. This scalable approach could enable broad predictive modeling across healthcare organizations without requiring data harmonization.

2.2.3.2 RNN

The paper “Machine learning for real-time prediction of complications in critical care: a retrospective study deep learning methods” [59], aims to apply deep learning, specifically recurrent neural networks, to predict severe complications (mortality, renal failure requiring renal replacement therapy, and postoperative bleeding leading to operative revision) in real time during post-cardiosurgical care. The primary data set consisted of adult patients who underwent major open-heart surgery from 2000 to 2016 in a German tertiary care center. The predictive accuracy and timeliness of the deep learning model were measured and compared against established clinical reference tools.

Out of 47,559 intensive care admissions (which corresponded to 42,007 patients), the study included 11,492 admissions (corresponding to 9,269 patients). The deep learning models provided accurate predictions with positive predictive values (PPV) and sensitivity scores of 0.90 and 0.85 for mortality, 0.87 and 0.94 for renal failure, and 0.84 and 0.74 for bleeding. These predictions significantly outperformed the standard clinical reference tools, improving the complication prediction area under the curve (AUC) by 0.29 for bleeding, 0.24 for mortality, and 0.24 for renal failure.

The deep learning models also produced accurate predictions immediately after patient admission to the intensive care unit. Furthermore, when validated with the MIMIC-III dataset (comprising 5,898 cases), the machine learning approach demonstrated superior

performance compared to clinical reference tools, with improvements in AUC of 0.09 for bleeding, 0.18 for mortality, and 0.25 for renal failure.

The study concluded that the observed improvements in prediction for all three clinical outcomes could enhance critical care. The deep machine learning method outperformed clinical reference tools, particularly soon after admission, indicating its potential for prospective use in critical care settings to identify patients at highest risk. The study's findings are notable as they were derived solely from routinely collected clinical data, without the need for manual processing.

The paper "Recurrent Neural Networks for Multivariate Time Series with Missing Values" [60] aims to propose novel deep learning models, namely GRU-D, to effectively handle missing values in multivariate time series data and improve prediction performance. The authors believe that missing values and patterns in time series data often contain useful information for prediction tasks, especially in the healthcare domain.

The methodology is based on the Gated Recurrent Unit (GRU) to capture long-term temporal dependencies in time series. Two representations of missing patterns, masking and time interval, are incorporated into GRU to capture and utilize the missingness.

The prediction tasks being performed are time series classification, including binary classification (mortality prediction) on MIMIC-III and PhysioNet datasets, and multi-task classification (predicting multiple diagnosis codes or tasks) on MIMIC-III and PhysioNet datasets.

The data pre-processing steps include extracting multivariate time series features from raw datasets, normalizing all input variables to 0 mean and 1 standard deviation, using masking to indicate which variables are missing at each time step, and calculating time interval to indicate how long each variable has been missing.

The main conclusions are that missing values and patterns in time series data often contain useful information for prediction, especially in healthcare. The proposed GRU-D model can effectively capture and utilize the missingness by incorporating masking and time interval and outperforms baselines. GRU-D provides better performance for online prediction in the early stage and with limited training samples. Analysis of learnt decay

parameters in GRU-D gives insights into the impact of variable missingness on the prediction outcomes.

2.2.3.3 Deep Neural Network

The study "Improving Palliative Care with Deep learning" [61] aims to improve the quality of end-of-life care for hospitalized patients by using deep learning and Electronic Health Record (EHR) data to identify patients who would benefit from palliative care services. The authors believe that physicians often overestimate prognoses, leading to a mismatch between patients' wishes and the actual care they receive at the end of life. They propose that deep learning can be used to address this problem by automatically identifying patients who are likely to benefit from palliative care services.

The authors developed a deep neural network algorithm that evaluates the EHR data of admitted patients. The algorithm is trained on EHR data from previous years to predict all-cause 3-12 month mortality of patients, which serves as a proxy for identifying patients that could benefit from palliative care. The prediction task being performed by the algorithm is to predict the all-cause 3-12 month mortality of patients based on their EHR data.

The main conclusion is that the deep learning algorithm can enable the Palliative Care team to take a proactive approach in reaching out to patients who are likely to benefit from palliative care services, rather than relying on referrals from treating physicians or conducting time-consuming chart reviews of all patients. The authors also present a novel interpretation technique to provide explanations of the model's predictions.

2.2.4 Explaining Model Predictions Made on EHR Data

The paper "Comparative analysis of explainable machine learning prediction models for hospital mortality" [62] aims to construct and compare different machine learning (ML) models for predicting hospital mortality in ICU patients. The authors aim to examine the internal behavior of these models using SHapley Additive exPlanations (SHAP) values. The models were built using the same features used to calculate the APACHE IV score and were based on random forest, logistic regression, naive Bayes, and adaptive boosting algorithms.

The authors believe that machine learning holds the promise of becoming an essential tool for utilizing the increasing amount of clinical data available for analysis and clinical decision support. However, they also acknowledge that the lack of trust in these models, often due to their lack of explainability and interpretability, has limited their acceptance in healthcare. They argue that improving trust requires the development of more transparent ML methods.

The authors used the publicly available eICU database to construct the ML models. They tested several different pre-processing techniques, such as scaling of the input features, removal of patients with more than a certain number of missing values, and filling the missing values with reference values instead of mean values. The models were trained by minimizing the error with respect to the area under the receiver operating characteristic curve (AUC ROC/AUC/c-statistic). The prediction task being performed by the models is to predict hospital mortality in ICU patients.

The data pre-processing steps included scaling of the input features, removal of patients with more than a certain number of missing values and filling the missing values with reference values instead of mean values.

The authors concluded that while the four different ML models developed in the study have similar discriminative abilities, they behave quite differently. The models had similar discriminative abilities and mostly agreed on feature importance, but the calibration and impact of individual features differed considerably. The authors highlight the importance of explainable ML models and argue that understanding how models work is crucial for trust, which is essential for their implementation and use in healthcare settings. They also note that a seemingly good model does not necessarily correspond with a medically sound understanding.

The paper "Explainable machine learning to predict longterm mortality in critically ill ventilated patients, a retrospective study" [63] aims to develop an explainable machine learning model that can predict long-term mortality in critically ill ventilated patients. The model is intended to be used in the critical care field, where understanding the rationale behind decisions is crucial. The authors believe that while AI technologies have been widely applied in many fields, their adoption in the critical care field remains

uncommon due to the 'black box' issue. They argue that interpretability is substantially required in high-stake decisions, such as those in critical care.

The authors used a retrospective study design. They used the week-1 data, including comprehensive ventilatory data, to predict mortality after week-1. They used machine learning techniques such as Stochastic Gradient Boosting (SGB), LIME (Local Interpretable Model-Agnostic Explanations), and SHAP (SHapley Additive exPlanation) to develop and explain their model. They divided the data into a training dataset (80%) and a testing dataset (20%). They used the receiver operating characteristic (ROC) curve analysis, calibration curve, and decision curve analysis to determine the discrimination, accuracy, and applicability of the predictive ML models in the testing sets. The prediction tasks being performed by the model include predicting 30-day, 90-day, and 1-year mortality in critically ill ventilated patients.

The authors conclude that their model can predict short-, medium-, and long-term outcomes with interpretability among critically ill ventilated patients. They also found that the cumulative feature importance of the ventilatory domain decreased along with the prediction window, which is consistent with the clinical condition that ventilatory condition mainly reflects acute/short-term outcome. However, they acknowledge that their study and similar ones are single-center studies, and prospective multi-center studies are required to validate their findings.

The paper “Explainable Machine-Learning Model for Prediction of In-Hospital Mortality in Septic Patients Requiring Intensive Care Unit Readmission” [64] aims to develop an effective, stable, and explainable machine learning model for predicting mortality in septic patients requiring ICU readmission. The author believes in the potential of machine learning models in predicting mortality in septic patients. The author emphasizes the importance of explainability in these models.

The study uses a machine learning model, specifically a Random Forest (RF) classifier, to predict mortality. The model uses clinical features such as Glasgow Coma Scale, urine output, blood urea nitrogen, lactate, platelet, and systolic blood pressure. The model's explainability is assessed using SHapley Additive exPlanations (SHAP) values. The

prediction task involves determining the mortality risk of septic patients requiring ICU readmission. The model uses various clinical features to make these predictions.

The study concludes that parameters related to organ perfusion contribute highly to outcome prediction during ICU readmission for sepsis. The results indicate that the RF model was effective in predicting mortality in septic patients requiring ICU readmission.

The paper “Understanding Heart Failure Patients EHR Clinical Features via SHAP Interpretation of Tree-Based Machine Learning Model Predictions” aims to examine whether machine learning models, specifically the XGBoost model, can accurately predict a patient's heart failure stage based on their electronic health records (EHR). The researchers also applied the SHapley Additive exPlanations (SHAP) framework to identify informative features and their interpretations. The authors believe that machine learning models can be used to accurately monitor the disease progression of heart failure patients by continuously mining patients' EHR data. They also suggest that tailored prediction/monitoring models should be developed for different sub-populations to enhance their performance.

The authors used the XGBoost machine learning model and the SHAP framework to analyze structured data from EHRs. They also performed unsupervised clustering visualization. The main prediction task was to determine a patient's heart failure stage based on their EHR data.

The data processing steps included keeping only the drug and disease names that appear over 10,000 times in the dataset as valid features, normalizing numerical features like age, BMI, and blood pressure to exclude outliers, and setting values outside the 1% and 99% percentile to the value of 1 percentile (MIN) or 99 percentiles (MAX).

The study concluded that with the XGBoost model, SHAP interpretation, and unsupervised clustering visualization, they could predict EF score from tabular EHR data with decent performance, generate interpretations for both the XGBoost model and dataset, and classify the subgroups of heart failure. The interpretations generated were consistent with heart failure diagnosis guidelines and human intuition. The model demonstrated that variables such as gender, blood pressure, age, pulse, BMI, some

diagnoses, and medications all have an impact on heart failure stage. The authors believe that the future use of machine learning models to construct clinical decision aids related to heart failure is justifiable and feasible.

The paper "Machine learning-based prediction of in-hospital mortality using admission laboratory data: A retrospective, single-site study using electronic health record data" aims to develop a model that predicts in-hospital mortality within 14 days using machine learning technology and variables of age, sex, and blood sampling test results of 21 items recorded in the electronic medical record at the time of hospitalization. The authors believe that the machine learning model developed in this study has the potential to be useful in evaluating the in-hospital mortality risk of admitted patients.

The authors used four machine learning methods: logistic regression, random forest, multilayer perceptron, and gradient boosting decision tree. The missing data was filled with multiple imputation in m (= 20) times. As a result, m (= 20) complete data sets were generated after multiple imputation. In the training phase, cross-validation was performed in the condition of k (= 5) fold. The prediction tasks being performed involved predicting in-hospital mortality within 14 days using machine learning models.

The data preprocessing involved excluding certain variables like patient's ID, hospitalization time, and alkaline phosphatase value displayed in King-Armstrong unit from the remaining variables. Because alkaline phosphatase values in IU/l were included in the data, the values with King-Armstrong unit were removed. Ultimately, 25 variables, specifically, age, sex, 21 laboratory variables, length of stay, and mortality, were considered eligible for analysis. Subsequently, cases that were missing all variables of eligible laboratory data were excluded. Finally, a training/validation data set (n = 119,160) and a test data set (n = 33,970) were obtained.

The authors concluded that they developed a model that predicts in-hospital mortality within 14 days with high predictive performance using machine learning technology and variables of age, sex, and blood sampling test results of 21 items recorded in the electronic medical record at the time of hospitalization. This machine learning model has the possibility to be useful in evaluating the in-hospital mortality risk of admitted patients.

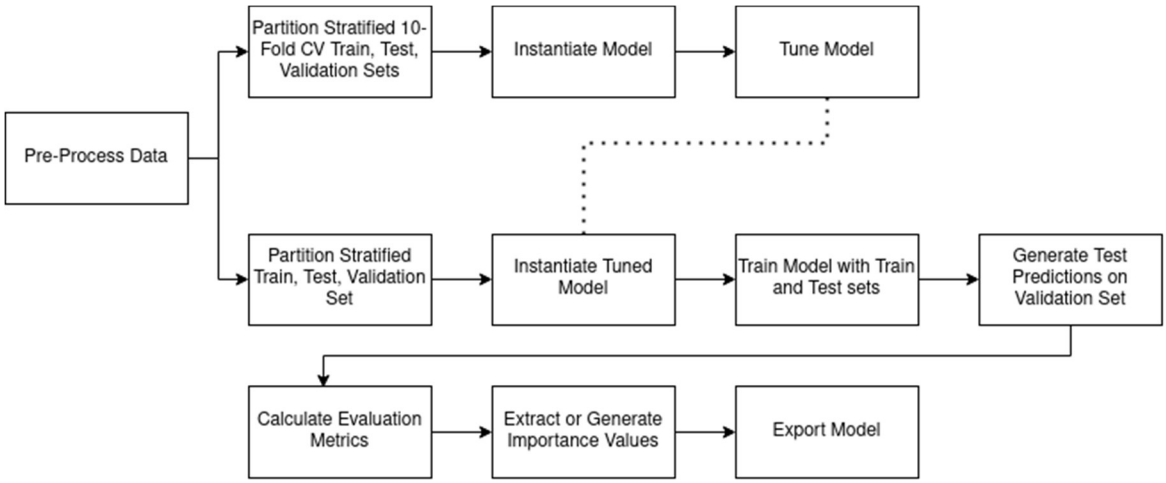
3 Analyzing Explainable Mortality Predictions of Black Box Deep Learning Models

3.1 Introduction

In this chapter we investigate the performance of both black box, and white box machine learning models in a benchmarking mortality prediction task. The white box model utilized is Logistic Regression. The black box models utilized are a Random Forest, an SVM with a Linear Kernel, LSTM, DNN, 1D CNN, and a Deconvolution Convolution Neural Network.

3.2 Methodology

Figure 7: Model Training and Importance Extraction



The MIMIC III Dataset is first processed down to a predetermined set of 21139 ICU stays. The first 48 hours of data is extracted from each stay. Each data sample is labelled as expired (True) or non-expired (False). The dataset comprising of the ICU stays is imbalanced, containing majority non-expired samples (18342) and a minority of expired samples (2797). Additionally, the dataset also contains missing features for some samples. To address missingness, imputation is used, using fixed value replacement to clinically accepted baselines [57]. One-Hot-Encoded masking columns are also added for each

imputed feature. A value of 1 indicates the value was imputed at the current timestemp. Each sample contains 17 clinical features, denoted in Table 1.

Table 1: MIMIC Features.

Features	Mimic Source Table	Feature Type
Capillary refill rate	chartevents	categorical
Diastolic blood pressure	chartevents	continuous
Fraction inspired oxygen	chartevents	continuous
Glascow coma scale eye opening	chartevents	categorical
Glascow coma scale motor response	chartevents	categorical
Glascow coma scale total	chartevents	categorical
Glascow coma scale verbal response	chartevents	categorical
Glucose	chartevents, labevents	continuous
Heart Rate	chartevents	continuous
Height	chartevents	continuous
Mean blood pressure	chartevents	continuous
Oxygen saturation	chartevents, labevents	continuous
Respiratory rate	chartevents	continuous
Systolic blood pressure	chartevents	continuous
Temperature	chartevents	continuous
Weight	chartevents	continuous
pH	chartevents, labevents	continuous

The clinical features undergo further treatment depending on their feature type (column 3 of Table 1). If a feature is categorical, it is transformed to a 'One Hot Encoded' feature. The categorical feature undergoes a transformation where each class within the feature is converted to its own True/False Boolean column. For example, the Glascow coma scale verbal response gets transformed into 6 columns, one for each possible class. Continuous variables undergo standardization before inputting into models.

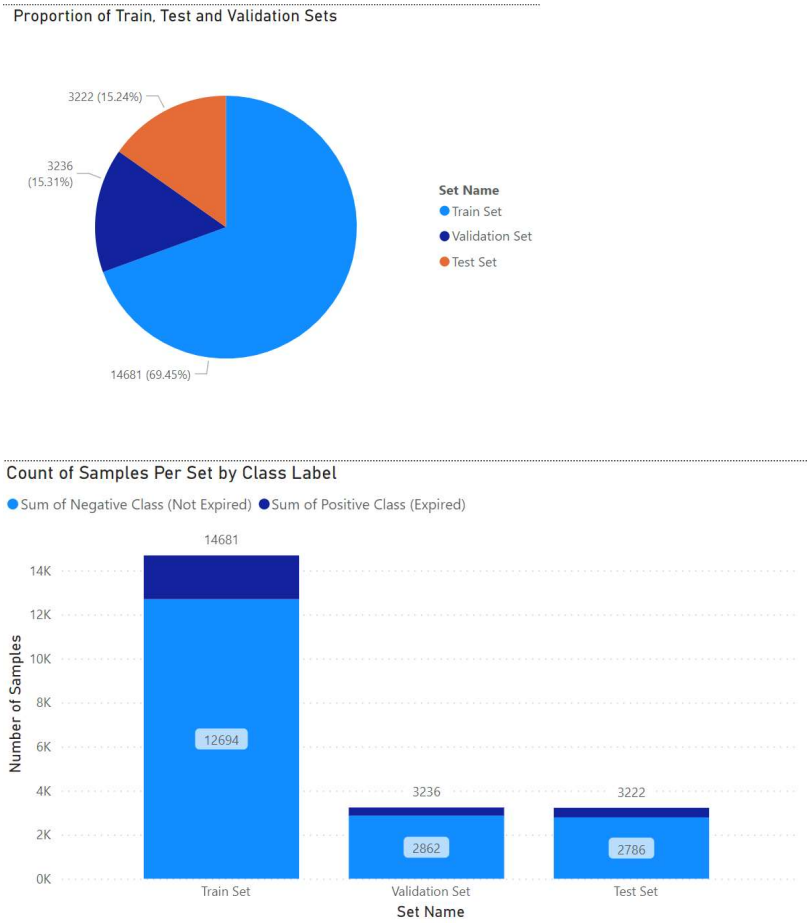
The processed dataset is then split into Train/Test sets, and a final holdout set is used for validation of the models. The dataset is split according to a pre-established baseline [57]

and is denoted in Table 2. The train, test and validation set have had their class labels evenly distributed.

Table 2: MIMIC Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (Expired)**	**Negative Class (Not Expired)**	**Set Totals**
Train Set	Training	1987	12694	14681
Test Set	Training	436	2786	3222
Validation Set	Final Evaluation	374	2862	3236
Class Totals		2797	18612	21319

Figure 7: MIMIC Dataset Class Labels.



To ensure representative performance, each model is tuned using a grid parameter search over a 10-fold stratified cross validation. Performance metrics tracked for each model are Precision, Recall, Accuracy, PR-AUROC and AUROC. The models will be ranked based on their best performing AUROC score due to the minority class of samples comprising over 10% of the training and test sets. The hyperparameters are then extracted from the best performing models out of each parameter configuration tested. Hyperparameters are extracted by calculating the most occurring values across multiple folds.

The models are trained until a change in loss is < 0.0001 or the models have exceeded 1500 epochs/training iterations. The test set is fed to the model, and the model is instructed to generate class probabilities. These class probabilities form the foundation for further post-hoc performance analysis. The final performances for each top-performing model are then evaluated using two thresholds for defining class decisions: Youden Index on the train set, and a standard fixed threshold of 0.5.

To set a proper baseline to assess the statistical significance of our results, we then configure a series of studies to represent the null hypothesis. We capture this performance baseline by performing a series of label permutation tests on the dataset. For a single permutation test the y labels of the train, test and validation sets are randomly shuffled based off a seeded value for reproducibility. Each model instance is evaluated using the permuted labels for model training and testing. The permutation test for each model is repeated 100 times each with a different set of shuffled y labels. The combined studies form a representative performance metric that establishes a numeric threshold that defines the performance achieved when the model is "randomly choosing" outcomes.

After each model has been trained, tested, and validated, black box models are then analyzed on the train set using the SHAP framework. Feature importances will be extracted via model coefficients for linear models (Logistic Regression, Linear SVM) and feature importance for tree-based models (Random Forest). The extracted feature importance will then be contrasted to the black box SHAP values for similarities and differences in magnitudes among the universal feature set.

The SHAP values are generated on a per-prediction basis. To provide a more unified view, we absolute the SHAP values, and average the values across each prediction to get an aggregated measure of overall importance for each feature.

For the linear model coefficients, we absolute each coefficient to get a non-signed view of importance. Since the Random Forest importance are already unsigned, no additional

treatment needs to be applied to make them comparable. The results from each model were reviewed by a panel of four domain experts to confirm validity and highlight areas that do not reconcile with typical expertise.

3.3 Results

In this chapter we provide a model-by-model breakdown of feature importance and performance measures. Table 3 denotes the performance metrics for each model predicting on the validation set, which was not seen during training. The rightmost column indicates the AUROC score during permutation testing of the train/test set.

Table 3: Model Performances.

Model	**Threshold**	**Threshold Value**	**Precision (Class Expired)**	**Recall (Class Expired)**	**AUROC**	**PR-AUC**	**MinPSE**	**Null Hypothesis AUROC (Mean, Min, Max)**
LR	argmax	0.5000	0.6183	0.2166	0.5996	0.4627	0.2166	(0.5000, 0.5000)
LR	youden	0.1387	0.2783	0.7299	0.7413	0.5197	0.2783	(0.4600, 0.5300)
SVM_Linear Kernel	argmax	0.5000	0.5000	0.0027	0.5012	0.3090	0.1156	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.2821	0.1663	0.2059	0.5355	0.2320	0.1663	(0.4800, 0.5200)
RF	argmax	0.5000	**0.7222**	0.1738	0.5825	0.4958	0.1738	(0.5000, 0.5000)
RF	youden	0.2217	0.3726	0.6337	0.7471	0.5243	0.3726	(0.4900, 0.5100)
1D-CNN	argmax	0.5000	0.6667	0.1604	0.5750	0.4621	0.1604	(0.5000, 0.5010)
1D-CNN	youden	0.0304	0.3027	0.7299	0.7551	0.5319	0.3027	(0.4330, 0.5270)
DECONV-CONV **Error! Reference source not found.**	argmax	0.5000	0.5035	0.3797	0.6654	0.4775	**0.3797**	(0.5000, 0.5000)
DECONV-CONV **Error! Reference source not found.**	youden	0.0602	0.3077	0.7059	0.7492	0.5238	0.3077	(0.5000, 0.5000)

DNN	argmax	0.5000	0.6111	0.2647	0.6213	0.4804	0.2647	(0.5000, 0.5010)
DNN	youden	0.0464	0.3096	0.6979	0.7473	0.5212	0.3096	(0.3960, 0.5270)
LSTM	argmax	0.5000	0.6286	0.2941	0.6357	0.5021	0.2941	(0.5000, 0.5010)
LSTM	youden	0.1209	0.2977	**0.7513**	**0.7598**	**0.5389**	0.2977	(0.4960, 0.5310)

Green = Top AUROC Argmax
Blue = Top AUROC Youden
Bold = Highest Metric

the Deconvolution Convolution shows the best AUROC
ional 0.5 decision boundary. The best performing overall AUROC score is the LSTM network when using the decision boundary defined by the Youden index. The Random Forest was able to predict outcomes with the highest level of precision, while the LSTM was able to predict most instances of mortality.

Every model had a higher AUROC score when examining the class labels generated by the Youden index decision boundary. Each recall score consistently increases, at the expense of a smaller precision score.

Most black box models (1D-CNN, DECONV CONV, DNN, and LSTM) outperformed the white box model (LR) AUROC score when applying the optimized Youden Index decision boundary. When examining from a 0.5 decision boundary, the DECONV CONV, DNN and LSTM all outperform logistic regression in the AUROC score. These models are also capable of predicting more instances of mortality than the white box Logistic regression (higher recall).

The deep learning-based models (1D-CNN, DECONV CONV, DNN, and LSTM) consistently outperform classical machine learning approaches (Random Forest, Logistic Regression and SVM) from an AUROC score perspective when labelling from a Youden index decision boundary.

Each model significantly higher performance than their baseline null hypothesis, $p < 0.01$.

Figure 9: Logistic Regression Feature Coefficients on MIMIC Dataset.

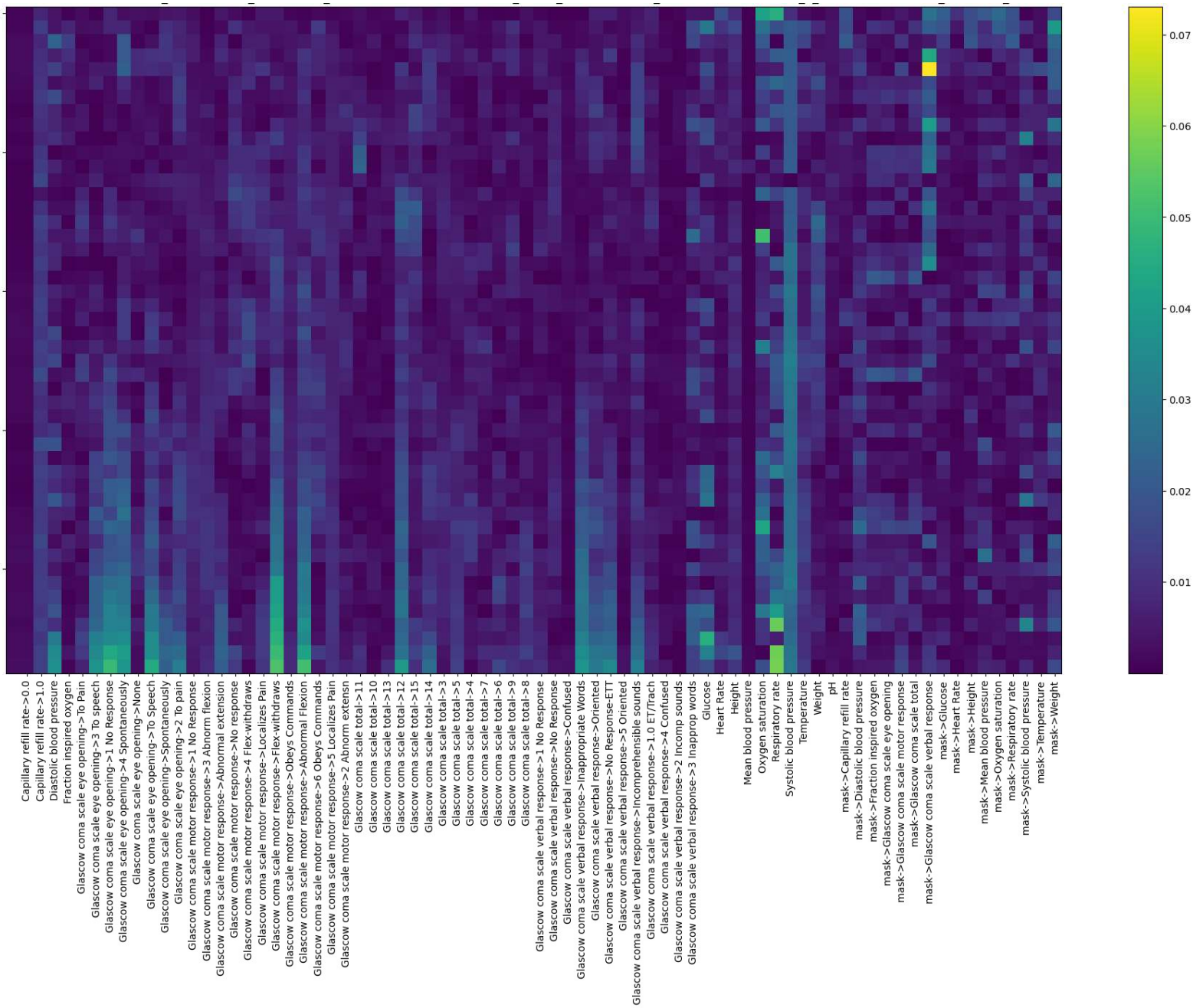


Figure 10: SVM with Linear Kernel Feature Coefficients on MIMIC Dataset.

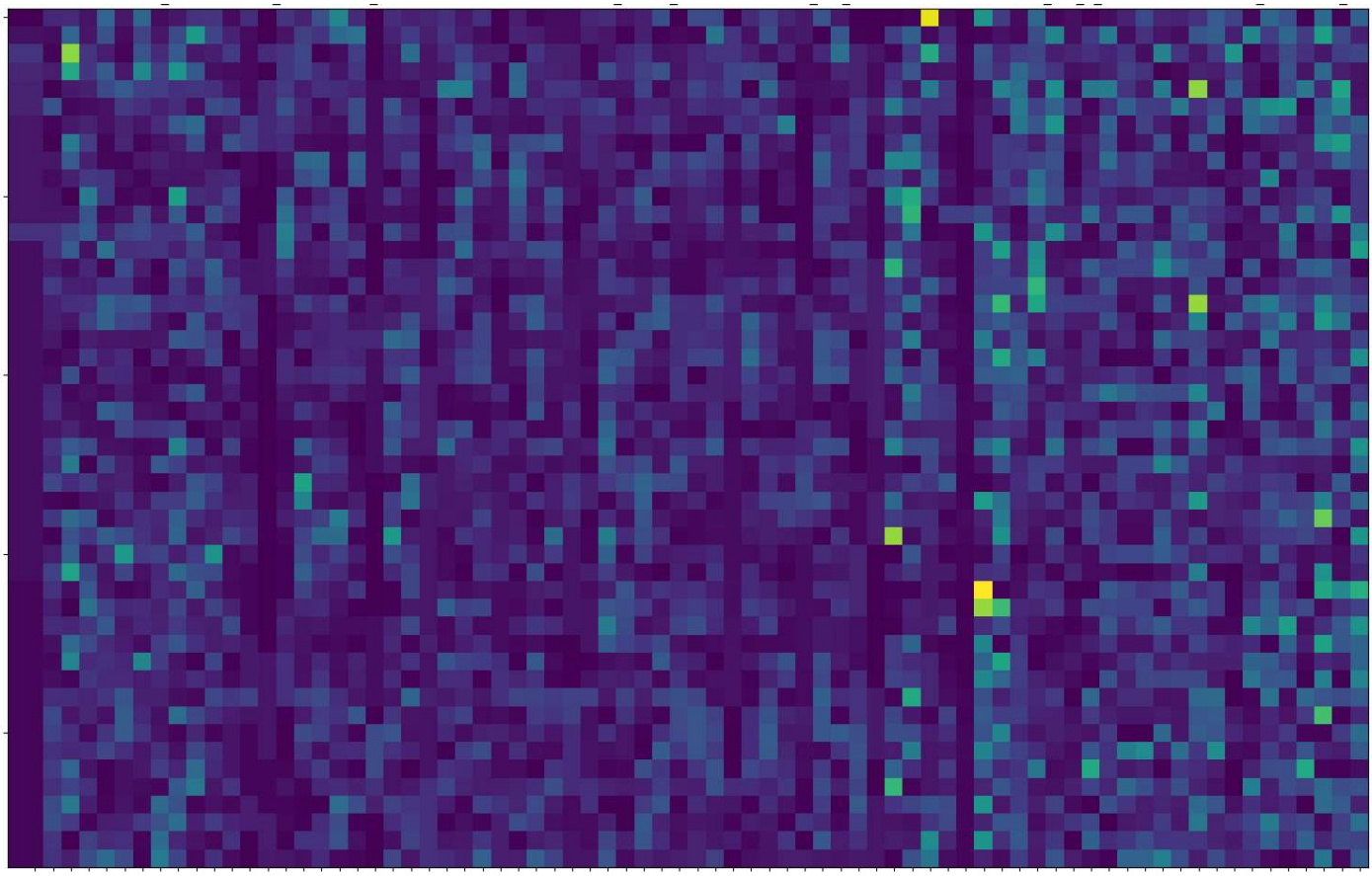


Figure 11: Random Forest Feature Importances on MIMIC Dataset.

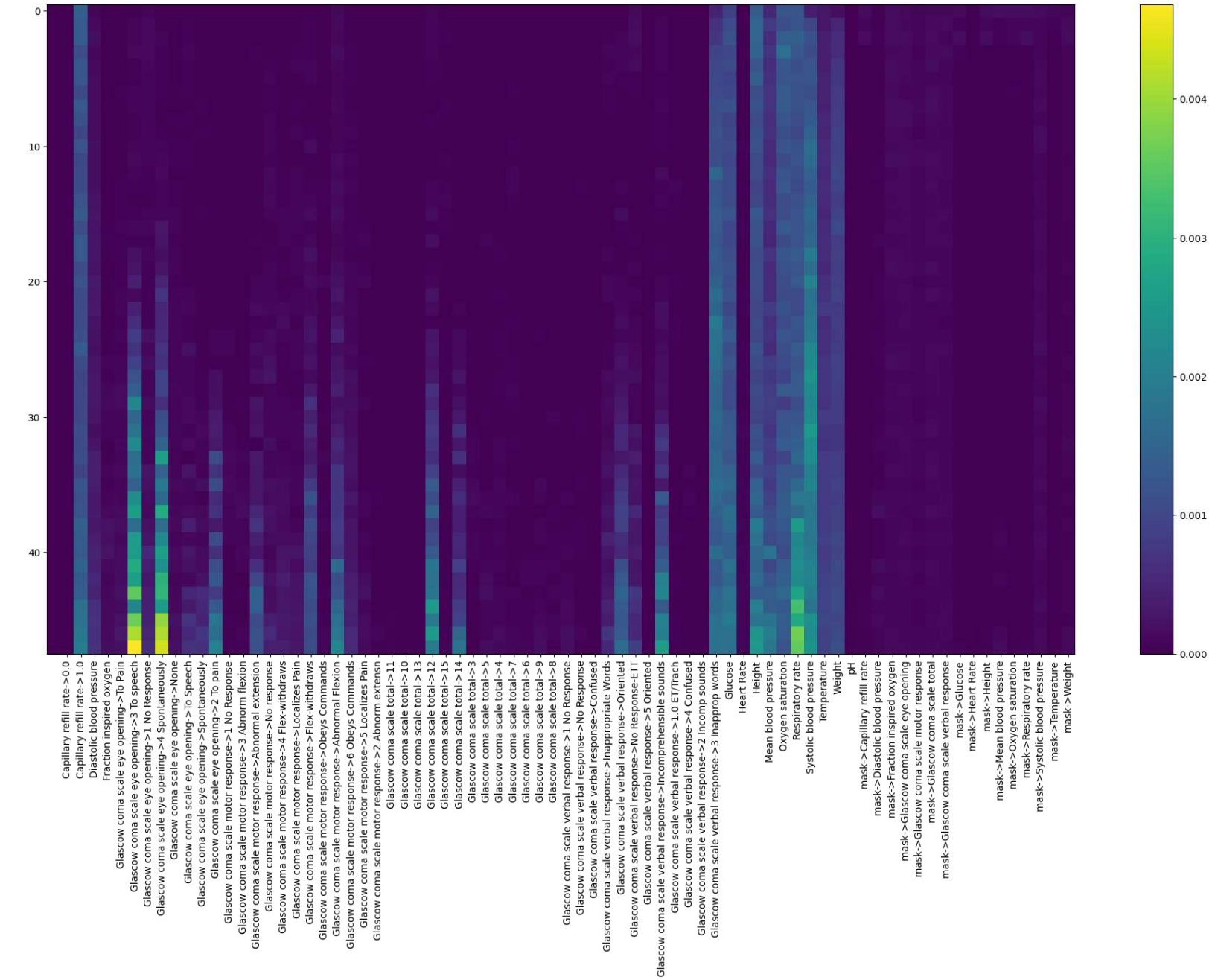


Figure 12: 1D CNN SHAP Values on MIMIC Dataset.

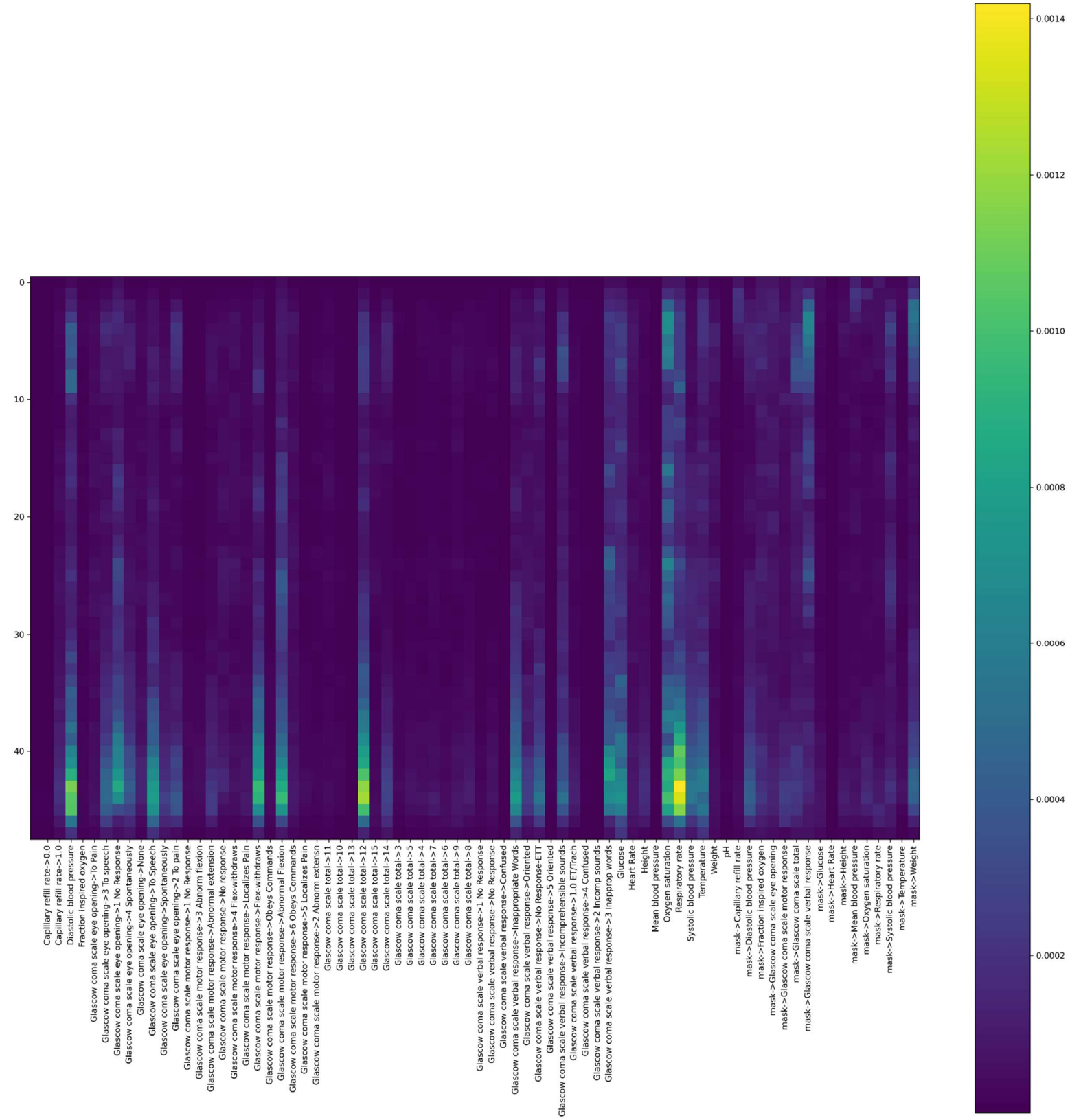


Figure 13: DECONV CONV SHAP Values on MIMIC Dataset.

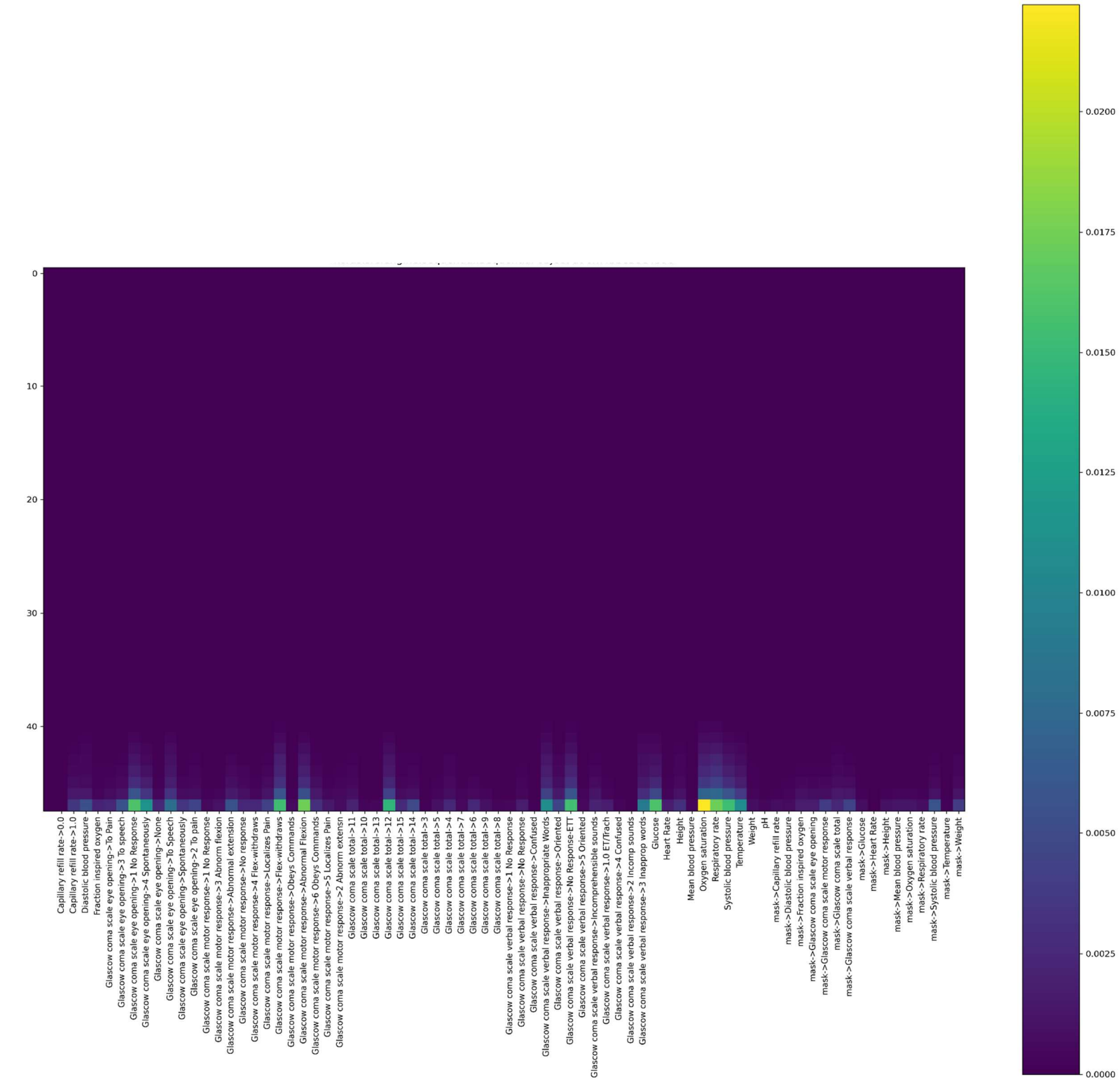


Figure 14: DNN SHAP Values on MIMIC Dataset.

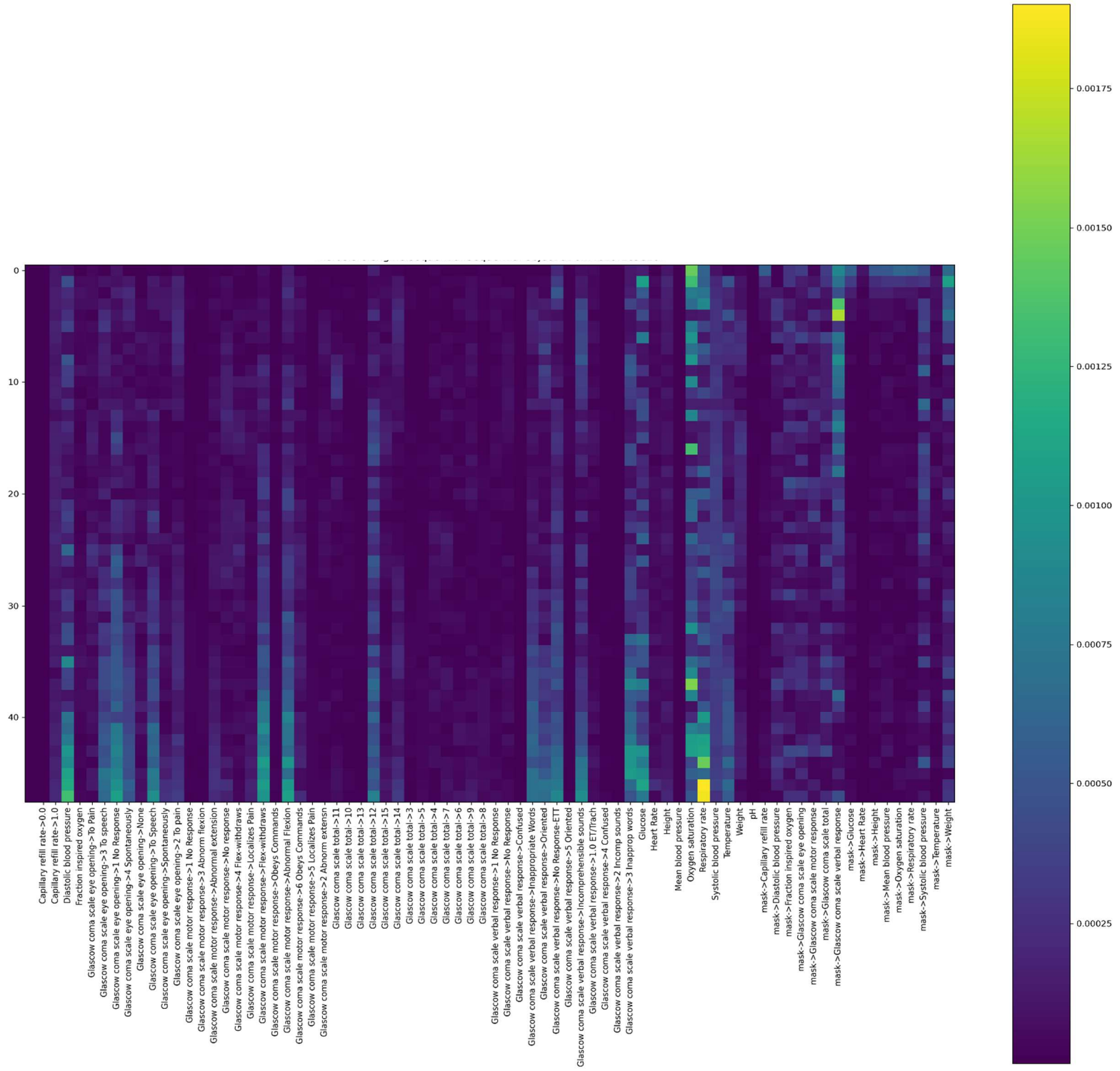
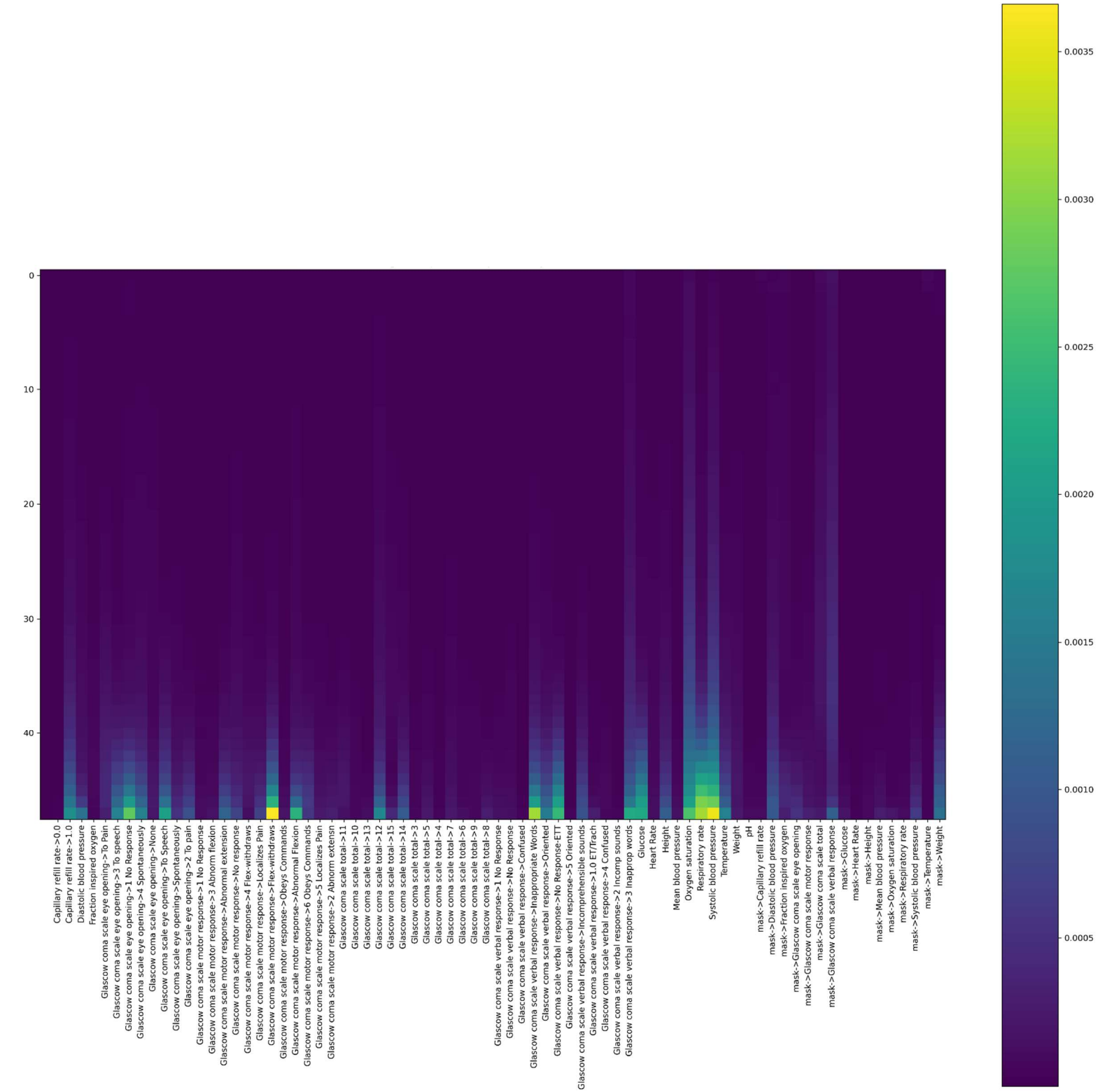


Figure 15: LSTM SHAP Values on MIMIC Dataset.



Higher performing models show a similar prioritization pattern in features. The top 2 performing models (Deconv Conv and LSTM) both show higher feature prioritization near the end of the 48-hour window. Figure 12 and Figure 14 show a gradual increase for certain features (Respiratory Rate, Systolic Blood pressure, Oxygen Saturation and Glasgow Coma scale categorical features). The Deconv Conv network initiates its gradual increase of feature importance at T=40, while the LSTM network starts the gradual increase earlier (T=30 and earlier). The more important features in the higher performing models have a higher overall score than those in the lower performing models. (e.g. total importance is distributed over more features).

Less performant models from an AUROC score perspective (LR, SVM, DNN, 1D CNN) show fewer gradual increases in feature importance over time. They also appear to show more sporadic importances that increase then suddenly decrease as the time scale increases.

3.4 Discussion

We identify higher level performance and trends by examining the results through logical groupings of the models. We will examine model performance and importance by comparing black box vs. white box models, and deep learning vs classical machine learning models.

Overall, the Youden index decision boundary performs better than the standard 0.5 decision boundary in an AUROC score. This is primarily due to a large increase in recall. The youden index calculation lowers the decision boundary by a large amount for each model. As a consequence, this creates a situation where more stays are labelled as expired, despite a lower probability of expiration. This results in an increased recall because we identify more True positives by increasing the number of positive predictions. However, this reduces each model's precision, because more false positives will be labelled as expired. In the context of mortality prediction in the ICU, a false positive would carry less cost than a false negative, as it is less harmful for a patient to receive prioritized care despite being at a lower risk for mortality than a patient who is higher risk for mortality not receiving higher priority care.

We see the precision-recall tradeoff with certain models in the argmax decision boundary as well. Models with higher precision generally have lower recall than those with lower precision. This again is since the models with higher precision are making fewer positive

predictions. In a clinical setting this may not be ideal because more True positive patients may be missed.

Overall, we see better performance amongst the black box models compared to the white box models. This is primarily due to their ability to disseminate more complex relationships than linear models.

The plots of higher performing black box models like LSTM, and DECONV CONV show overall smoother charts with higher peak importance values. This may be since they have built in mechanisms to do further feature extraction and isolation through their transformation layers. They are better able to understand and extract complex spatial relationships that a traditional neural network would not. Low performing plots would often be noisier and have feature importance distributed disparately throughout the time series. Their peak importance was lower overall than the better performing models. This can be explained by the fact that a model that is better able to classify would be better at extracting a targeted subset of meaningful features. It is also apparent that as model performance improves, the importance plots seem to superficially converge to a more "optimal" feature importance representation. Higher performing models also placed higher importance on features near the end of the 48-hour observation period. This would reconcile with understanding that the most representative readings of patient's current condition would generally be the most current, especially when examining complications.

Domain experts noted higher performing models (LSTM, Deconv Conv) have their feature prioritizations in line with what would be expected to be important in a clinical setting. It was specifically noted that the lower performing models (DNN, 1D CNN, RF, SVM, LR) lacked a necessary targeted prioritization on lower Glasgow Coma Scale Scores, which are utilized quite heavily in the ICU to assess patient mortality risk.

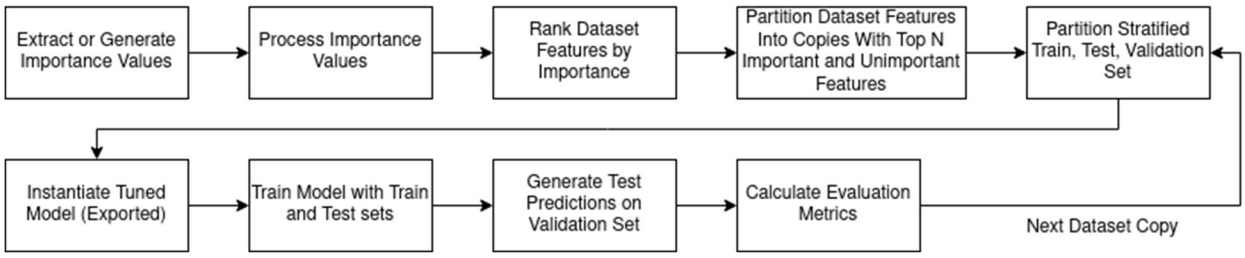
4 Analysis of Post Hoc Explainability in Clinical Decision Making

4.1 Introduction

In this chapter we investigate the performance impact of training machine learning models on different subsets of features created by ranking features based on their importances. The performance assessment is conducted on the same outcome prediction task as the previous chapter, along with additional binary classification tasks performed on an additional three clinical datasets. The first additional task is a patient outcome prediction performed on the Sepsis Survival Minimal Clinical Records dataset. The second additional task is a re-admission prediction performed on the Diabetes 130-US Hospitals for Years 1999-2008. The third task is a diagnostic task distinguishing between benign or malignant tumours in the Wisconsin Breast Cancer dataset.

4.2 Methodology

Figure 16: Importance Based Feature Selection and Model Training



Each dataset has its own set of preprocessing steps that have been determined based on each dataset's attributes and the task being performed.

The Sepsis Survival Minimal Clinical Records dataset contains 4 distinct features for each patient. The features are denoted in Table 3. Both patient sex and episode_number are converted to a numeric value through one-hot encoding. An additional categorical variable is derived from the age_years feature. The age_years feature is then standardized. The resulting dataset now contains 4 features for 110341 patients. The dataset does not contain any missing values. The dataset is imbalanced, with the majority class of "Non-Expired" being assigned to 92.63% of the samples. The outcome encoding has been

modified to assign 0 to a non-expired outcome, and 1 to an expired outcome. The train and test sets are defined using stratification.

Table 4: Sepsis Survival Minimal Clinical Records Features.

Features	Feature Type
age_years	continuous
episode_number	continous
sex	categorical
outcome	categorical

Table 5: Sepsis Survival Minimal Clinical Records Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (Expired)**	**Negative Class (Not Expired)**	**Set Totals**
Train Set	Training	4538	57175	61713
Test Set	Training	1135	14294	15429
Validation Set	Final Evaluation	2432	30630	33062
Class Totals		8129	102212	110341

The Diabetes 130-US Hospitals for Years 1999-2008 dataset contains both categorical and continuous variables with some missing values. The categorical features are all one-hot encoded. Any categorical column denoted with a missing has a distinct OHE category to represent missingness if present. The continuous features within the dataset are standardized. There are no missing values for the continuous features. The dataset is split into 3 discrete variations. The first dataset contains no readmission and < 30 readmission samples. The second variation contains no readmission and > 30 readmission samples. The final variation contains no readmission, < 30 readmission, and > 30 readmission samples.

The diag_1, diag_2 and diag_3 categorical features have respectively 848, 923, and 954 unique values. One-Hot-Encoding features would drastically increase the dimensionality of the feature space, which could affect performance in the < 30 readmission samples due to almost halving the sample. To maintain a reasonable dimensionality of the input features,

the diagnoses' ICD-9 codes are used to calculate a Charlson Comorbidity Index (CCI) for each patient stay [85]. In this manner the three categorical columns are compressed into a single continuous metric that has shown to be correlated to < 30 day hospital readmissions [87][86].

We test the combined dataset, <30-day readmission subset, <30-day readmission with CCI replacement and >30-day readmission with their own train and test sets.

Table 6: Diabetes 130-US Hospitals for Years 1999-2008 Features.

Feature Name	Feature Type	Use
encounter_id	Categorical (ID)	Dropped
patient_nbr	Categorical (ID)	Dropped
race	Categorical	Input
gender	Categorical	Input
age	Categorical	Input
weight	Categorical	Dropped
admission_type_id	Categorical	Input
discharge_disposition_id	Categorical	Input
admission_source_id	Categorical	Input
time_in_hospital	Continuous	Input
payer_code	Categorical	Dropped
medical_specialty	Categorical	Dropped
num_lab_procedures	Continuous	Input
num_procedures	Continuous	Input
num_medications	Continuous	Input
number_outpatient	Continuous	Input
number_emergency	Continuous	Input
number_inpatient	Continous	Input
diag_1	Categorical	Input
diag_2	Categorical	Input
diag_3	Categorical	Input
number_diagnoses	Continuous	Input
max_glu_serum	Categorical	Input
A1Cresult	Categorical	Input
metformin	Categorical	Input
repaglinide	Categorical	Input
nateglinide	Categorical	Input

chlorpropamide	Categorical	Input
glimepiride	Categorical	Input
acetohexamide	Categorical	Input
glipizide	Categorical	Input
glyburide	Categorical	Input
tolbutamide	Categorical	Input
pioglitazone	Categorical	Input
rosiglitazone	Categorical	Input
acarbose	Categorical	Input
miglitol	Categorical	Input
troglitazone	Categorical	Input
tolazamide	Categorical	Input
examide	Categorical	Input
citoglipton	Categorical	Input
insulin	Categorical	Input
glyburide-metformin	Categorical	Input
glipizide-metformin	Categorical	Input
glimepiride-pioglitazone	Categorical	Input
metformin-rosiglitazone	Categorical	Input
metformin-pioglitazone	Categorical	Input
change	Categorical	Input
diabetesMed	Categorical	Input
readmitted	Categorical	Output

Table 7: Diabetes 130-US Hospitals for Years 1999-2008 Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (Readmitted)**	**Negative Class (Not Readmitted)**	**Set Totals**
Train Set	Training	26264	30724	56988
Test Set	Training	6567	7681	14248
Validation Set	Final Evaluation	14071	16459	30530
Class Totals		46902	54864	101766

Table 8: Diabetes 130-US Hospitals for Years 1999-2008 (Under 30 Day Subset) Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (readmitted < 30 Day)**	**Negative Class (Not Readmitted)**	**Set Totals**
Train Set	Training	6360	30723	37083
Test Set	Training	1590	7681	9271
Validation Set	Final Evaluation	3407	16460	19867
Class Totals		11357	54864	66221

Table 9: Diabetes 130-US Hospitals for Years 1999-2008 (Over 30-Day Subset) Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (readmitted > 30 Day)**	**Negative Class (Not Readmitted)**	**Set Totals**
Train Set	Training	19904	30724	50628
Test Set	Training	4977	7681	12658
Validation Set	Final Evaluation	10664	16459	27123
Class Totals		35545	54864	90409

The Breast Cancer Wisconsin dataset contains mainly continuous features. There are no missing values. The continuous variables have been standardized. The categorical y variable (Diagnosis) has been defined through one hot encoding. A value of 1 indicates malignant, while a value of 0 indicates benign. The dataset is split into train, test, and validation sets using stratification.

Table 10: Breast Cancer Wisconsin Features.

Feature Name	Feature Type	Use
ID	Categorical	Dropped
Diagnosis	Categorical	Output
Radius Mean	Continuous	Input
Texture Mean	Continuous	Input

Perimeter Mean	Continuous	Input
Area Mean	Continuous	Input
Smoothness Mean	Continuous	Input
Compactness Mean	Continuous	Input
Concavity Mean	Continuous	Input
Concave Points Mean	Continuous	Input
Symmetry Mean	Continuous	Input
Fractal Dimension Mean	Continuous	Input
Radius SE	Continuous	Input
Texture SE	Continuous	Input
Perimeter SE	Continuous	Input
Area SE	Continuous	Input
Smoothness SE	Continuous	Input
Compactness SE	Continuous	Input
Concavity SE	Continuous	Input
Concave Points SE	Continuous	Input
Symmetry SE	Continuous	Input
Fractal Dimension SE	Continuous	Input
Radius Worst	Continuous	Input
Texture Worst	Continuous	Input
Perimeter Worst	Continuous	Input
Area Worst	Continuous	Input
Smoothness Worst	Continuous	Input
Compactness Worst	Continuous	Input
Concavity Worst	Continuous	Input
Concave Points Worst	Continuous	Input

Table 11: Breast Cancer Wisconsin Train, Test and Validation Set Sample Sizes.

Set Name	**Purpose**	**Positive Class (Malignant)**	**Negative Class (Benign)**	**Set Totals**
Train Set	Training	133	225	358
Test Set	Training	15	25	40
Validation Set	Final Evaluation	64	107	171
Class Totals		212	357	569

For each dataset, we first generate feature importances using the same general methodology described in the previous chapter for each dataset variation. We leverage a combination of the same set of models discussed in the previous dataset for MIMIC data. For the remainder of the datasets, only a DNN is used due to the tabular nature of the other datasets. The hyperparameters are chosen the same way as in the previous chapter. We then leverage the feature importances generated for each dataset to define an importance ranking for each feature in a dataset. We evaluate model performance using the top 1%, 5%, 10%, 25% and 50% of features based on their extracted feature importances. In the Sepsis example, we alter the percentages to account for even distribution amongst the very small number of features. We generate performance measures for each "IMPORTANT" subset, and then an "UNIMPORTANT" subset that contains all the features not in the "IMPORTANT" group.

Next, features are removed from each feature subset using two main techniques. For time series data (e.g. MIMIC), in model configurations where preserving dimensionality is required, features are hidden through applying a masking value. For the LSTM models, the features are excluded by changing the feature value to 0. This can be done because the LSTM models have a masking layer implemented that skips over any feature with a 0 value. For the 1D CNN and Deconvolution-Convolution network, the values are masked with the population mean. This preserves dimensionality while suppressing the influence the feature has on the output. For the remaining models that ingest flattened vectors, the features are deleted from each sample.

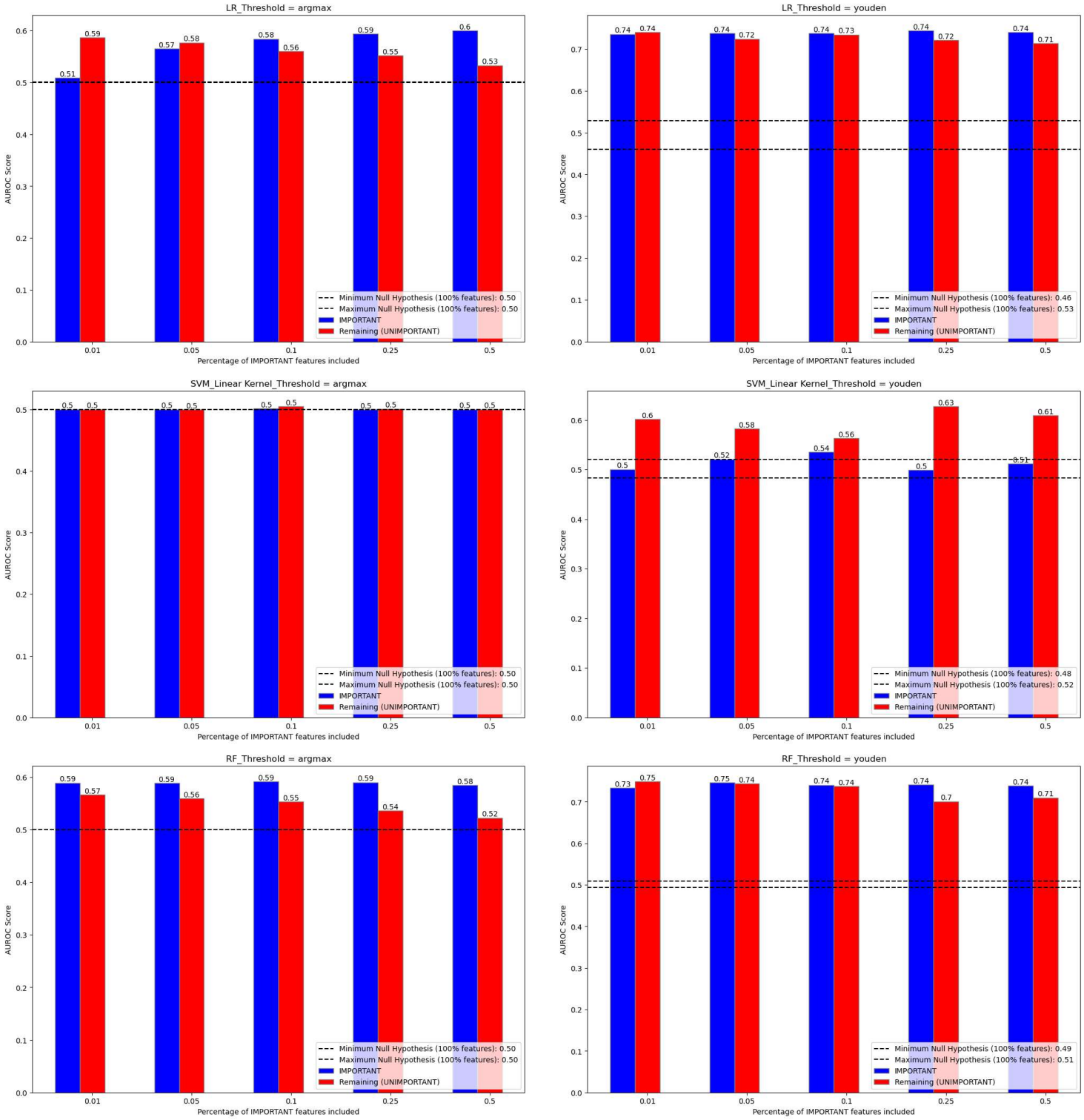
Finally, we establish the null hypothesis for this experiment using the same methodology in the previous chapter for each dataset and model. We evaluate significance by comparing the performance generated by the reduced feature set compared to the "random choosing" performance threshold of each model using 100% of the available features.

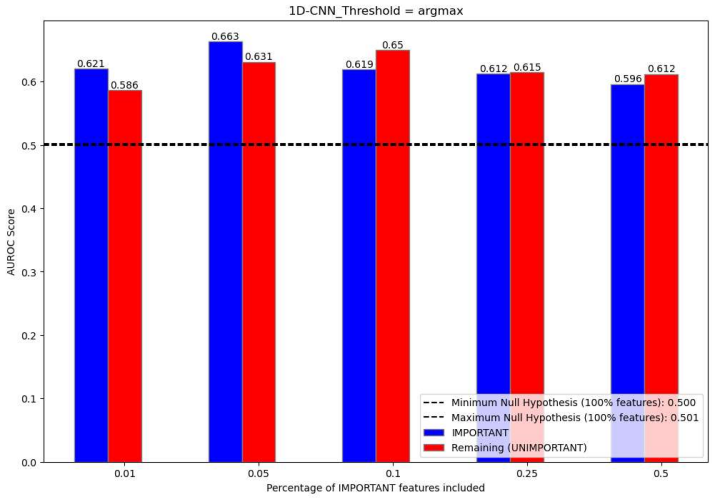
4.3 Results

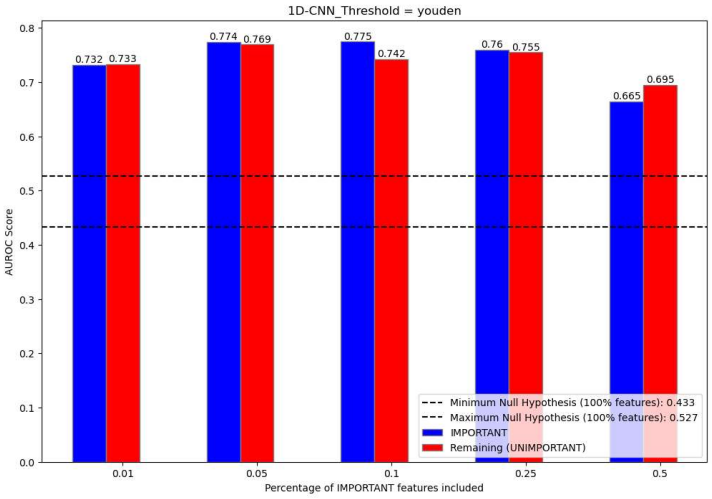
Table 12: MIMIC Dataset Performance Results (100% of Features).

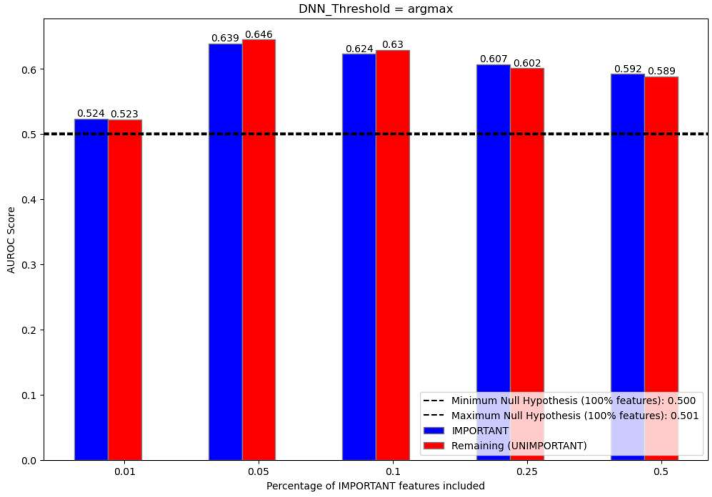
Model	**Threshold**	**Threshold Value**	**Precision (Class Expired)**	**Recall (Class Expired)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.6183	0.2166	0.5996	0.4627	(0.5000, 0.5000)
LR	youden	0.1387	0.2783	0.7299	0.7413	0.5197	(0.4600, 0.5300)
SVM_Linear Kernel	argmax	0.5000	0.5000	0.0027	0.5012	0.3090	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.2821	0.1663	0.2059	0.5355	0.2320	(0.4800, 0.5200)
RF	argmax	0.5000	0.7222	0.1738	0.5825	0.4958	(0.5000, 0.5000)
RF	youden	0.2217	0.3726	0.6337	0.7471	0.5243	(0.4900, 0.5100)
1D-CNN	argmax	0.5000	0.6667	0.1604	0.5750	0.4621	(0.5000, 0.5010)
1D-CNN	youden	0.0304	0.3027	0.7299	0.7551	0.5319	(0.4330, 0.5270)
DECONV-CONV	argmax	0.5000	0.5035	0.3797	0.6654	0.4775	(0.5000, 0.5000)
DECONV-CONV	youden	0.0602	0.3077	0.7059	0.7492	0.5238	(0.5000, 0.5000)
DNN	argmax	0.5000	0.6111	0.2647	0.6213	0.4804	(0.5000, 0.5010)
DNN	youden	0.0464	0.3096	0.6979	0.7473	0.5212	(0.3960, 0.5270)
LSTM	argmax	0.5000	0.6286	0.2941	0.6357	0.5021	(0.5000, 0.5010)
LSTM	youden	0.1209	0.2977	**0.7513**	**0.7598**	**0.5389**	(0.4960, 0.5310)

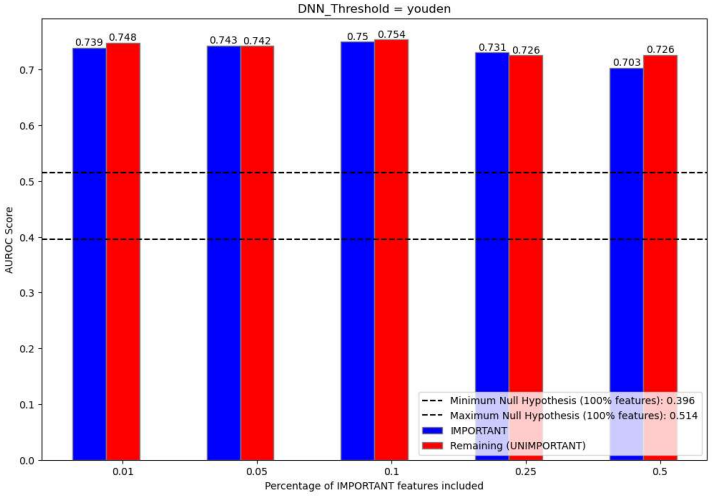
Figure 15: MIMIC Dataset Performance Results (Subset of Important Features).

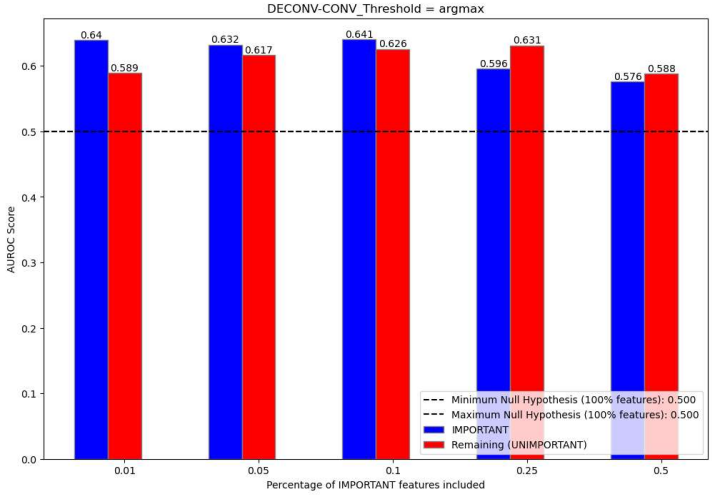


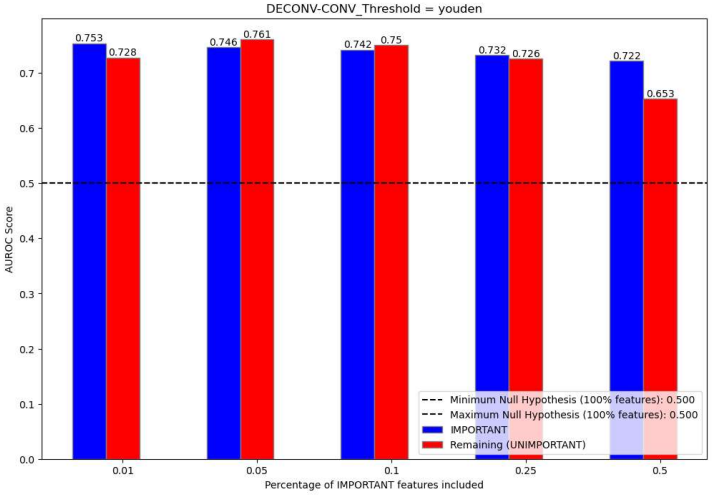

1D-CNN_Threshold = argmax
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.621
0.586
0.663
0.631
0.619
0.65
0.612
0.615
0.596
0.612
Minimum Null Hypothesis (100% features): 0.500
Maximum Null Hypothesis (100% features): 0.501
IMPORTANT
Remaining (UNIMPORTANT)

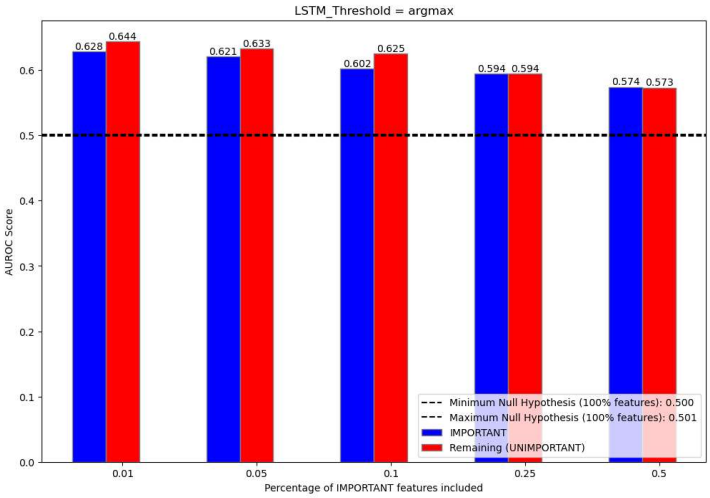

1D-CNN_Threshold = youden
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.732
0.733
0.774
0.769
0.775
0.742
0.76
0.755
0.665
0.695
Minimum Null Hypothesis (100% features): 0.433
Maximum Null Hypothesis (100% features): 0.527
IMPORTANT
Remaining (UNIMPORTANT)


DNN_Threshold = argmax
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.524
0.523
0.639
0.646
0.624
0.63
0.607
0.602
0.592
0.589
Minimum Null Hypothesis (100% features): 0.500
Maximum Null Hypothesis (100% features): 0.501
IMPORTANT
Remaining (UNIMPORTANT)


DNN_Threshold = youden
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.739
0.748
0.743
0.742
0.75
0.754
0.731
0.726
0.703
0.726
Minimum Null Hypothesis (100% features): 0.396
Maximum Null Hypothesis (100% features): 0.514
IMPORTANT
Remaining (UNIMPORTANT)


DECONV-CONV_Threshold = argmax
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.64
0.589
0.632
0.617
0.641
0.626
0.596
0.631
0.576
0.588
Minimum Null Hypothesis (100% features): 0.500
Maximum Null Hypothesis (100% features): 0.500
IMPORTANT
Remaining (UNIMPORTANT)


DECONV-CONV_Threshold = youden
AUROC Score
Percentage of IMPORTANT features included
0.01
0.05
0.1
0.25
0.5
0.753
0.728
0.746
0.761
0.742
0.75
0.732
0.726
0.722
0.653
Minimum Null Hypothesis (100% features): 0.500
Maximum Null Hypothesis (100% features): 0.500
IMPORTANT
Remaining (UNIMPORTANT)



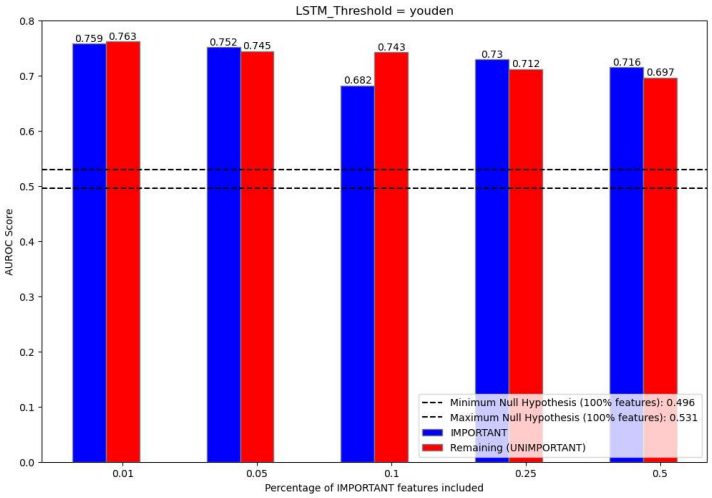
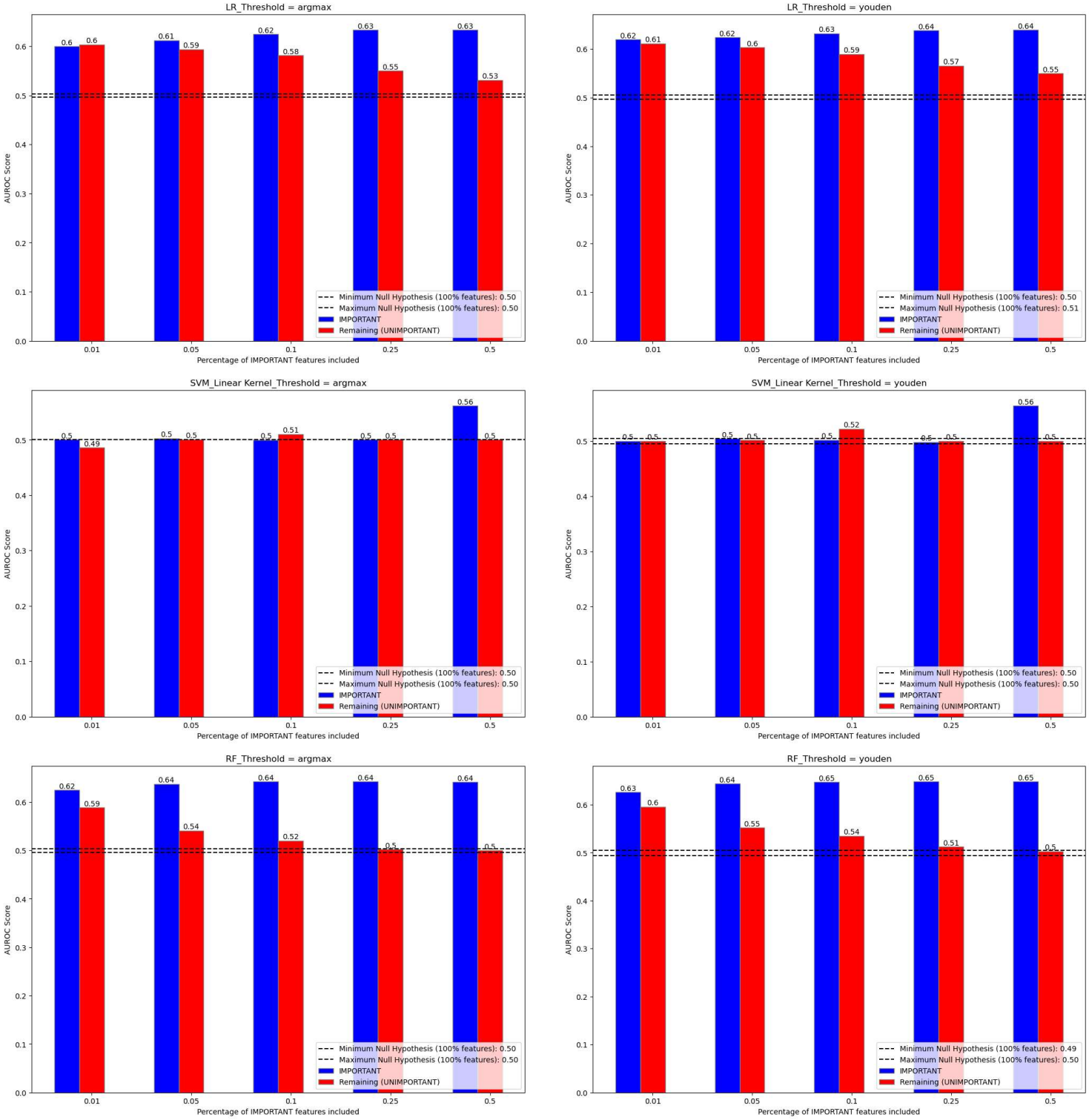
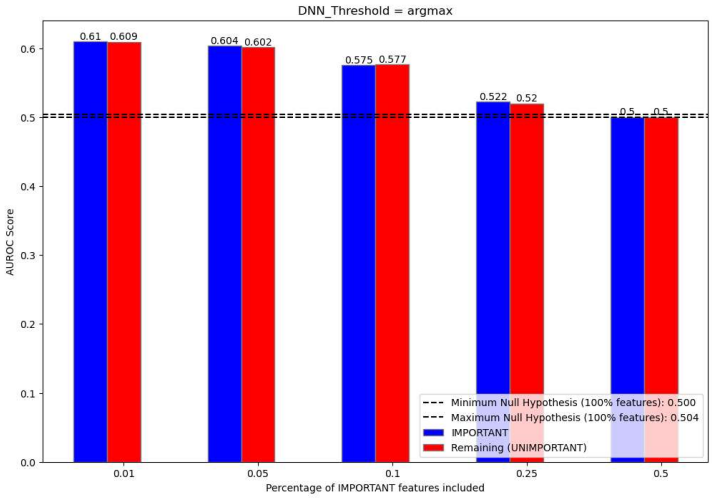


Table 13: Diabetes 130-US Hospitals for Years 1999-2008 (Categorical) Results (100% of Features).

Model	**Threshold**	**Threshold Value**	**Precision (Class Readmitted)**	**Recall (Class Readmitted)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.6462	0.5080	0.6351	0.6905	(0.4990, 0.5010)
LR	youden	0.4385	0.5970	0.6642	0.6405	0.7080	(0.4990, 0.5040)
SVM_Linear Kernel	argmax	0.5000	0.4352	0.2558	0.4860	0.5170	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.5433	0.4364	0.0017	0.4999	0.4491	(0.4980, 0.5010)
RF	argmax	0.5000	0.6503	0.5174	0.6398	0.6951	(0.4960, 0.5020)
RF	youden	0.4690	0.6186	0.6248	0.6477	0.7081	(0.4920, 0.5030)
DNN	argmax	0.5000	0.6428	0.5049	0.6325	0.6880	(0.5000, 0.5040)
DNN	youden	0.3779	0.6032	0.6285	0.6375	0.7015	(0.4700, 0.5330)

Figure 16: Diabetes 130-US Hospitals for Years 1999-2008 (Categorical) Results (Subset of Important Features).





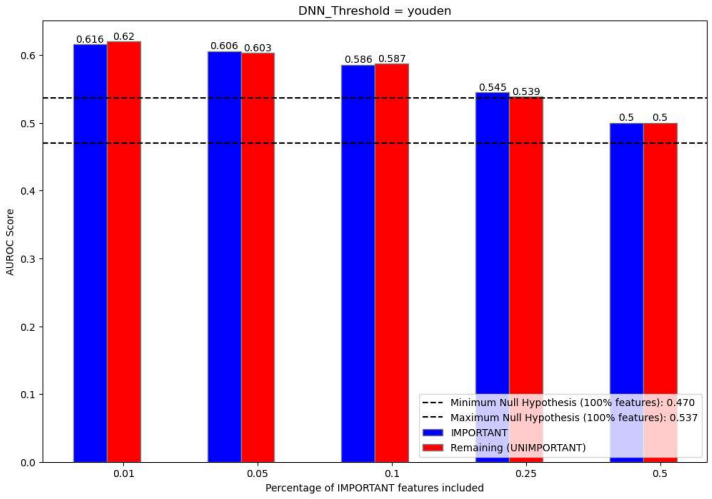
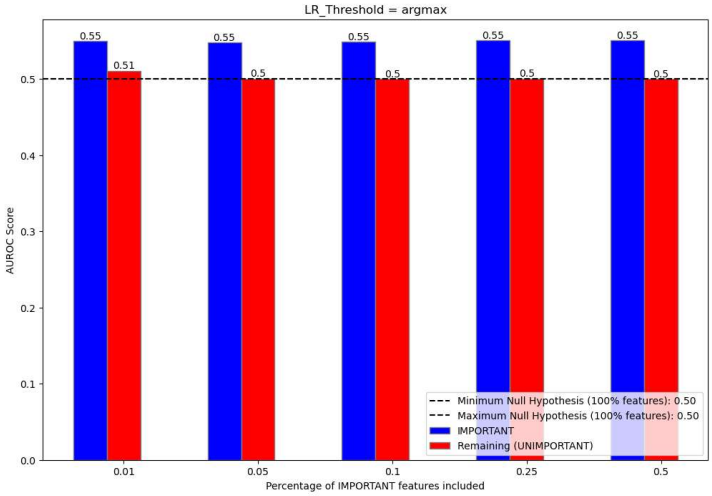
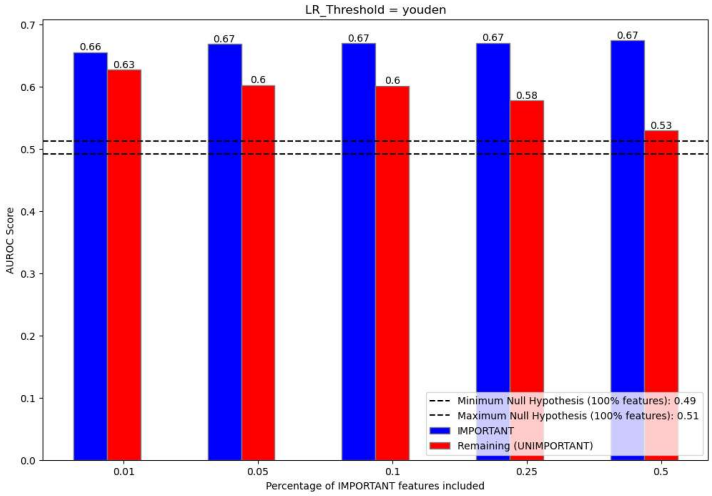


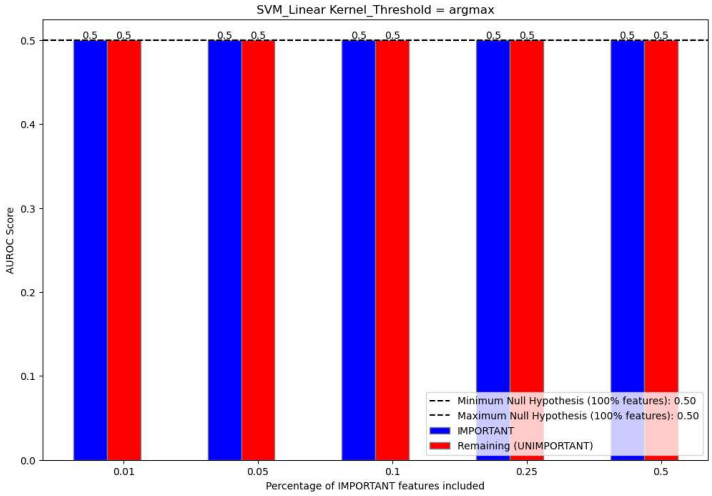
Table 14: Diabetes 130-US Hospitals for Years 1999-2008 (Under 30 Day Subset, Categorical) Results (100% of Features).

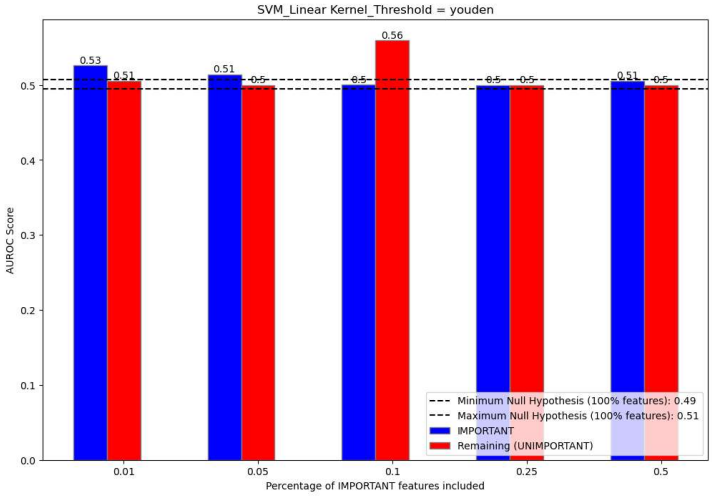
Model	**Threshold**	**Threshold Value**	**Precision (Class Readmitted)**	**Recall (Class Readmitted)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.6083	0.1162	0.5504	0.4380	(0.5000, 0.5000)
LR	youden	0.1667	0.3125	0.6451	0.6757	0.5093	(0.4920, 0.5100)
SVM_Linear Kernel	argmax	0.5000	0.0000	0.0000	0.5000	0.5857	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.2464	0.1772	0.1224	0.5024	0.2251	(0.4910, 0.5110)
RF	argmax	0.5000	0.9123	0.0152	0.5075	0.5482	(0.5000, 0.5000)
RF	youden	0.2064	0.3545	0.5365	0.6672	0.4853	(0.4930, 0.5120)
DNN	argmax	0.5000	0.5616	0.1861	0.5780	0.4436	(0.5000, 0.5000)
DNN	youden	0.0634	0.3271	0.6560	0.6884	0.5211	(0.4940, 0.5130)

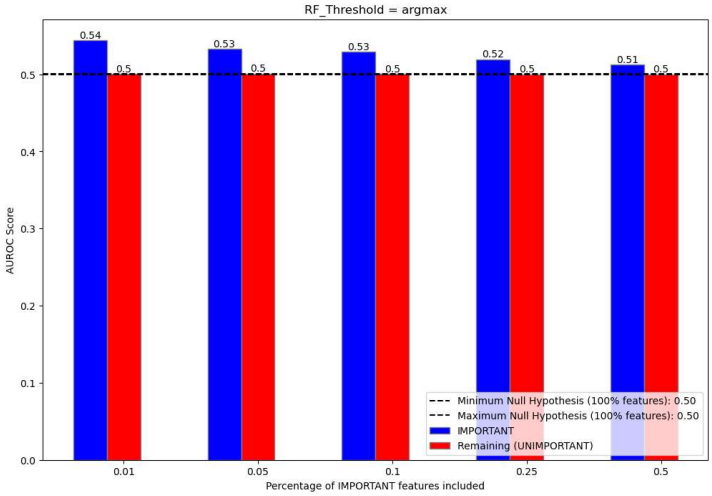
Figure 17: Diabetes 130-US Hospitals for Years 1999-2008 (Under 30 Day Subset, Categorical) Results (Subset of Important Features).

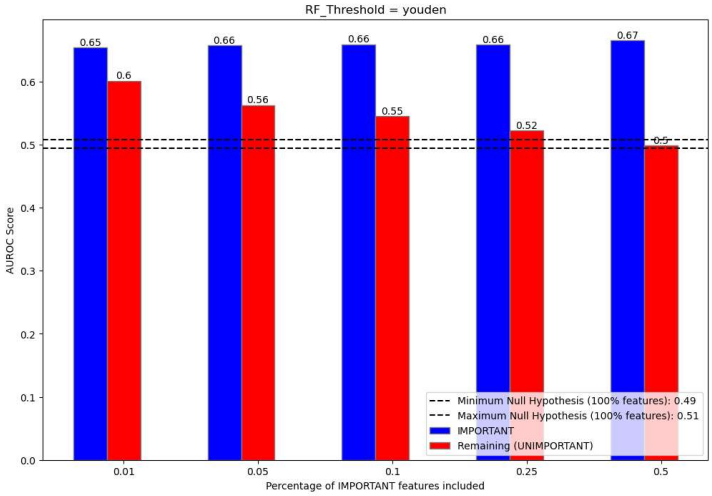


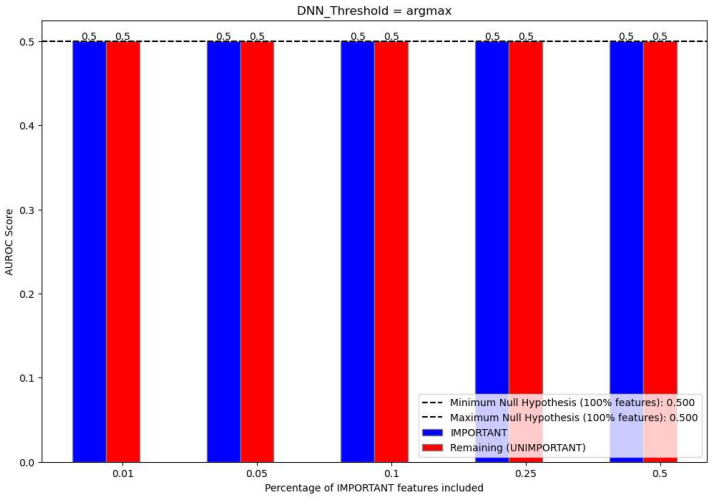












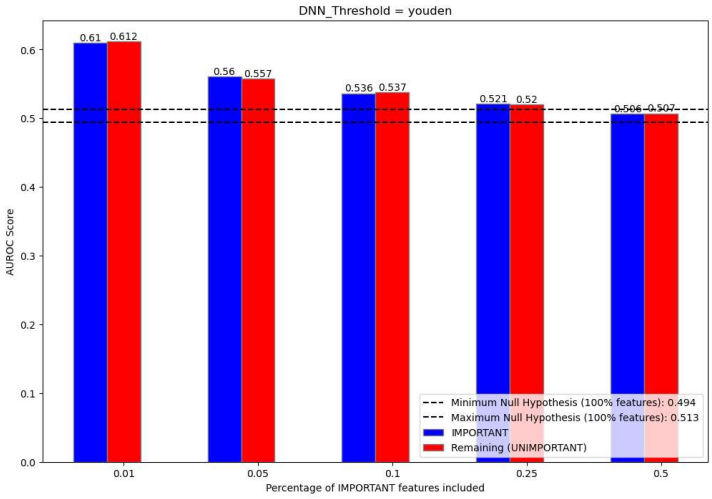
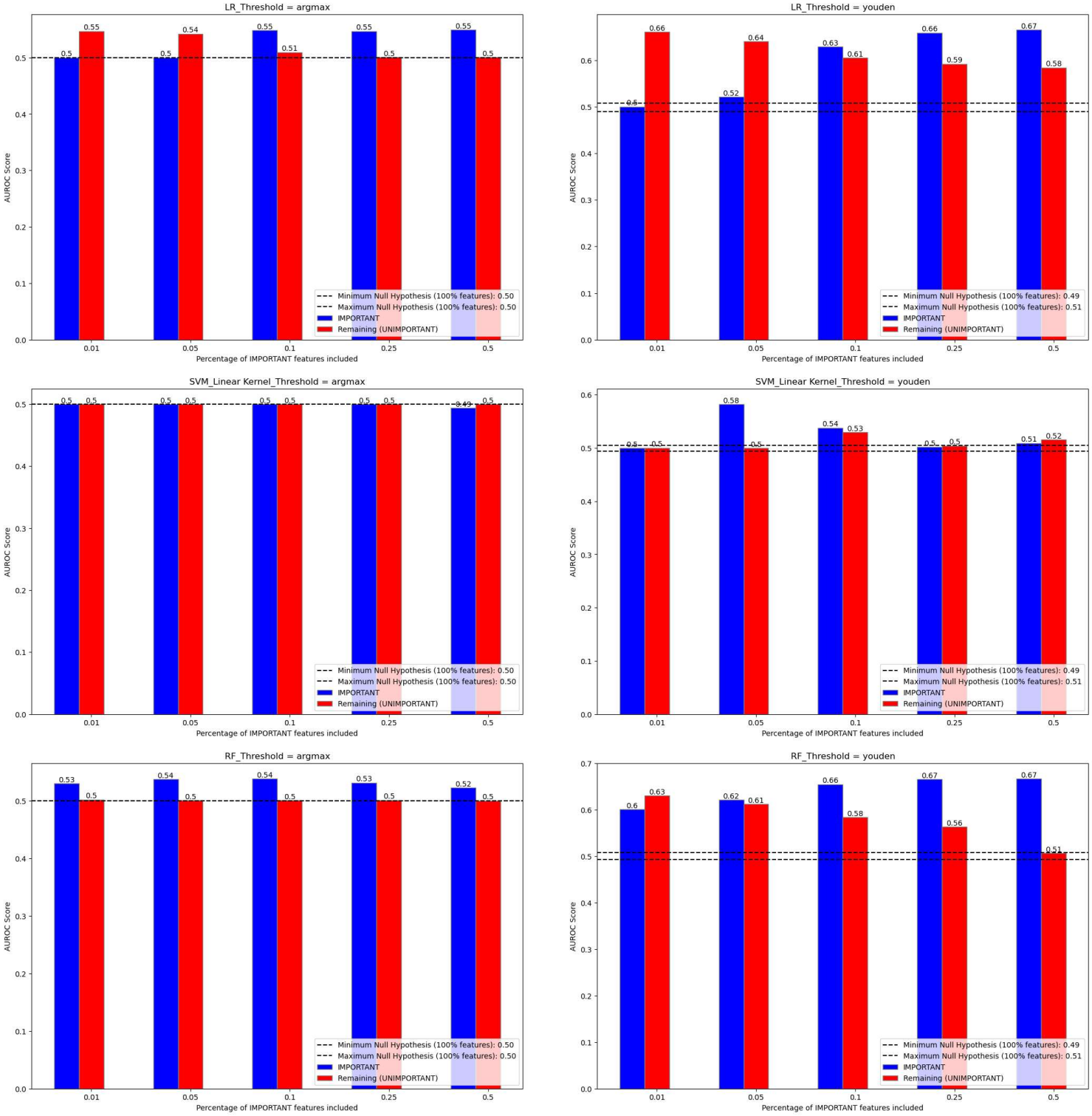
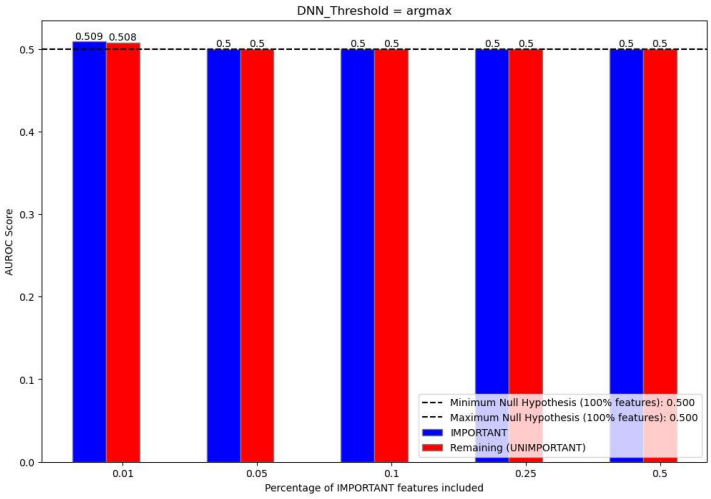


Table 15: Diabetes 130-US Hospitals for Years 1999-2008 (Under 30 Day Subset, CCI Score) Results (100% of Features).

Model	**Threshold**	**Threshold Value**	**Precision (Class Readmitted)**	**Recall (Class Readmitted)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.6037	0.1136	0.5491	0.4347	(0.5000, 0.5000)
LR	youden	0.1576	0.2941	0.6586	0.6657	0.5056	(0.4920, 0.5100)
SVM_Linear Kernel	argmax	0.5000	0.0000	0.0000	0.5000	0.5857	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.2561	0.0000	0.0000	0.4999	0.0857	(0.4910, 0.5110)
RF	argmax	0.5000	0.7961	0.0355	0.5168	0.4985	(0.5000, 0.5000)
RF	youden	0.1894	0.3193	0.6014	0.6680	0.4945	(0.4930, 0.5120)
DNN	argmax	0.5000	0.5688	0.1479	0.5624	0.4314	(0.5000, 0.5000)
DNN	youden	0.0615	0.3258	0.5929	0.6694	0.4942	(0.4940, 0.5130)

Figure 18: Diabetes 130-US Hospitals for Years 1999-2008 (Under 30 Day Subset, CCI Score) Results (Subset of Important Features).





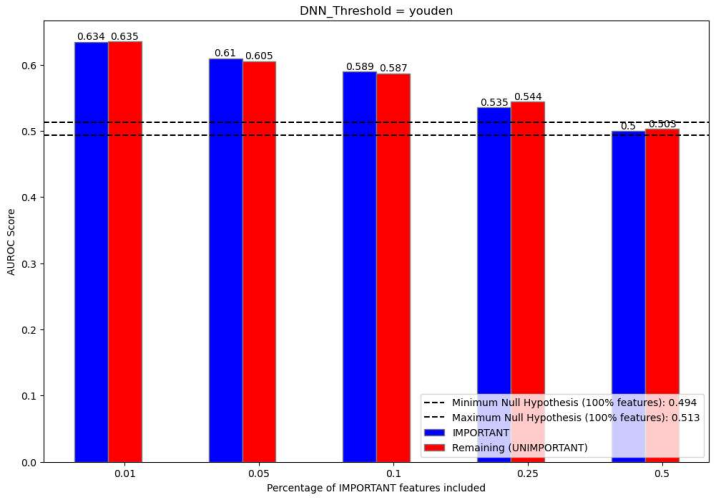
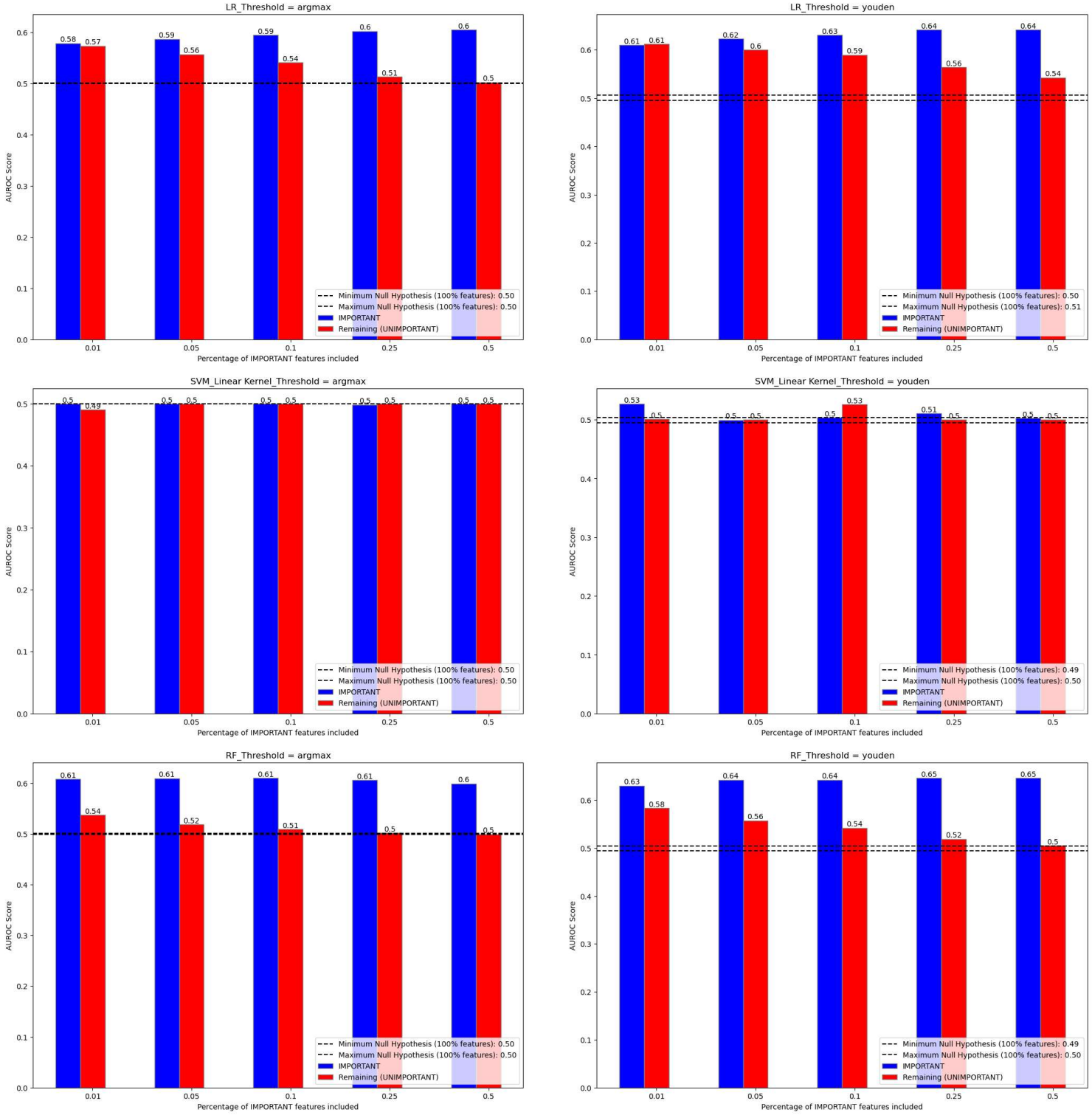
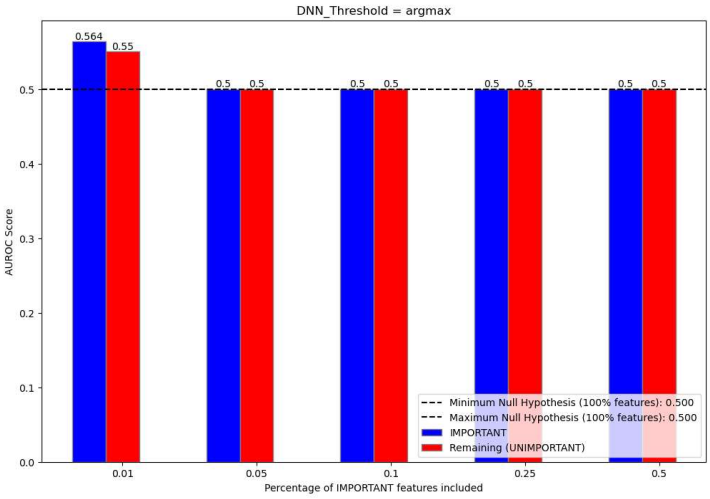


Table 16: Diabetes 130-US Hospitals for Years 1999-2008 (Over 30-Day Subset, Categorical) Results (100% of Features).

Model	**Threshold**	**Threshold Value**	**Precision (Class Readmitted)**	**Recall (Class Readmitted)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.6291	0.3445	0.6065	0.6157	(0.5000, 0.5000)
LR	youden	0.3833	0.5348	0.6525	0.6424	0.6619	(0.4920, 0.5100)
SVM_Linear Kernel	argmax	0.5000	0.0000	0.0000	0.5000	0.6966	(0.5000, 0.5000)
SVM_Linear Kernel	youden	0.3966	0.3960	0.7517	0.5044	0.6227	(0.4910, 0.5110)
RF	argmax	0.5000	0.6612	0.2762	0.5922	0.6110	(0.5000, 0.5000)
RF	youden	0.4168	0.5650	0.5819	0.6458	0.6556	(0.4930, 0.5120)
DNN	argmax	0.5000	0.6220	0.4370	0.6324	0.6402	(0.5000, 0.5000)
DNN	youden	0.3075	0.5457	0.6913	0.6592	0.6792	(0.4940, 0.5130)

Figure 19: Diabetes 130-US Hospitals for Years 1999-2008 (Over 30-Day Subset, Categorical) Results (Subset of Important Features).





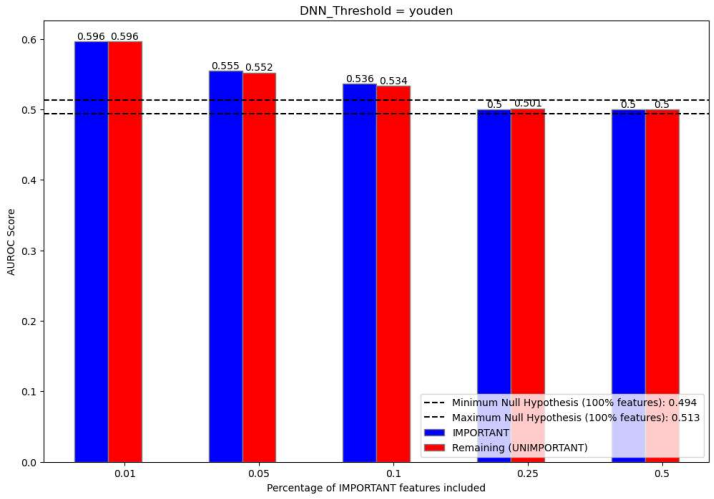
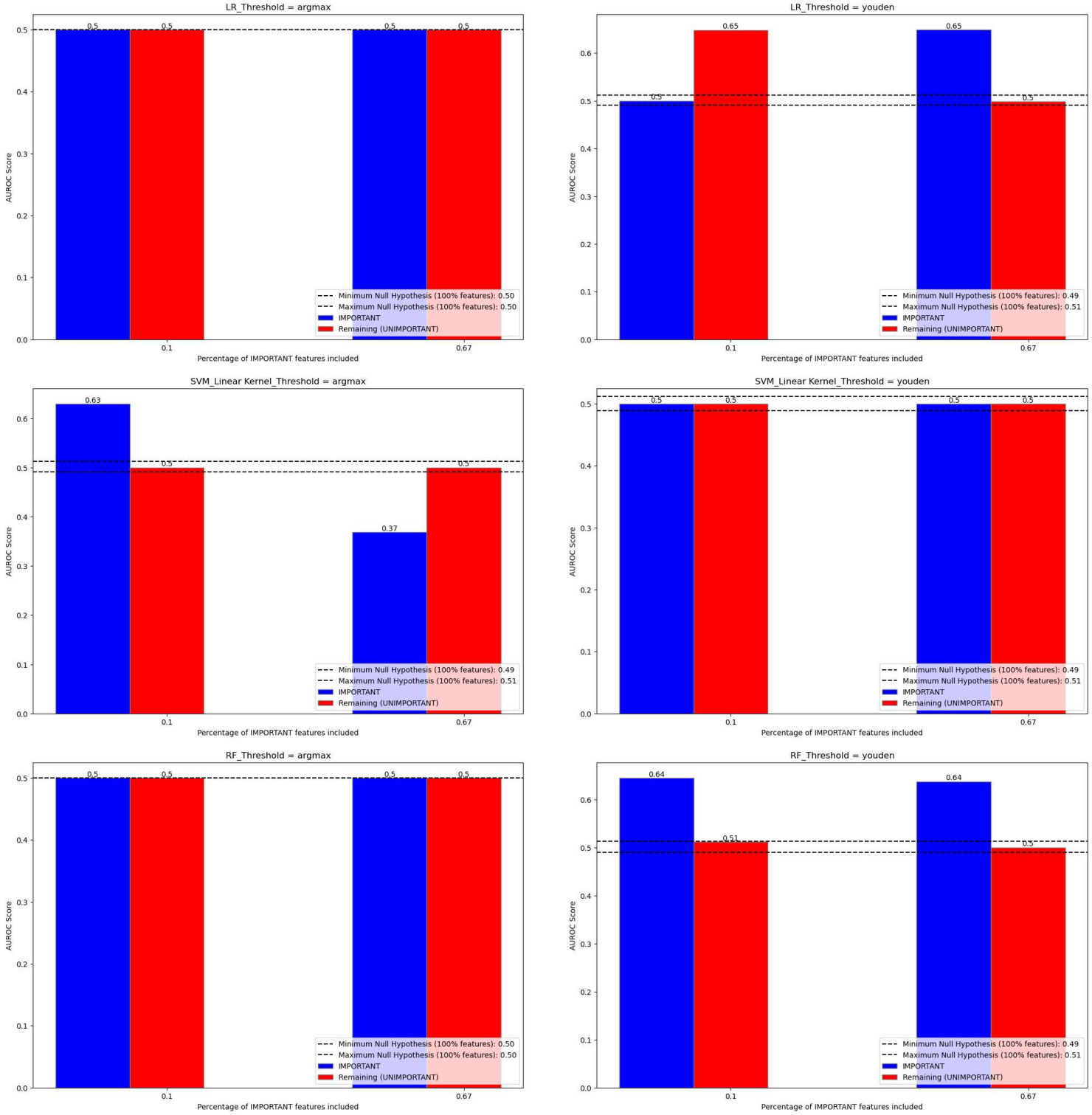
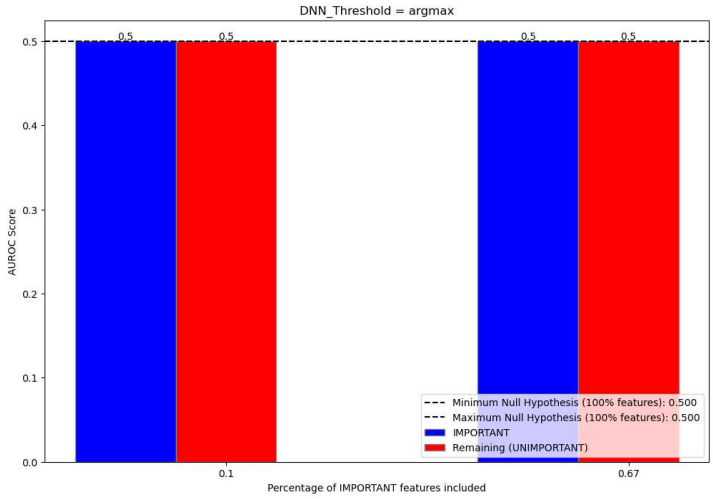


Table 17: Sepsis Survival Minimal Clinical Records Results (100% of Features).

Model	**Threshold**	**Threshold Value**	**Precision (Class Expired)**	**Recall (Class Expired)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.0000	0.0000	0.5000	0.5368	(0.5000, 0.5000)
LR	youden	0.0688	0.1161	0.7652	0.6513	0.4493	(0.4920, 0.5110)
SVM_Linear Kernel	argmax	0.5000	0.0000	0.0000	0.5000	0.5368	(0.4910, 0.5110)
SVM_Linear Kernel	youden	0.0752	0.0806	0.3590	0.5169	0.2434	(0.4910, 0.5110)
RF	argmax	0.5000	0.0000	0.0000	0.5000	0.5368	(0.5000, 0.5000)
RF	youden	0.0731	0.1129	0.7315	0.6378	0.4321	(0.4900, 0.5120)
DNN	argmax	0.5000	0.0000	0.0000	0.5000	0.5368	(0.5000, 0.5000)
DNN	youden	0.0130	0.1184	0.6982	0.6426	0.4194	(0.3520, 0.5480)

Figure 20: Sepsis Survival Minimal Clinical Records Results (Subset of Important Features).





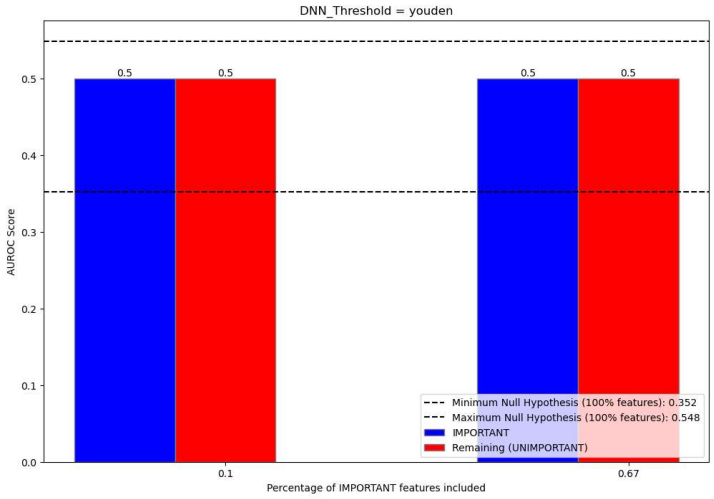
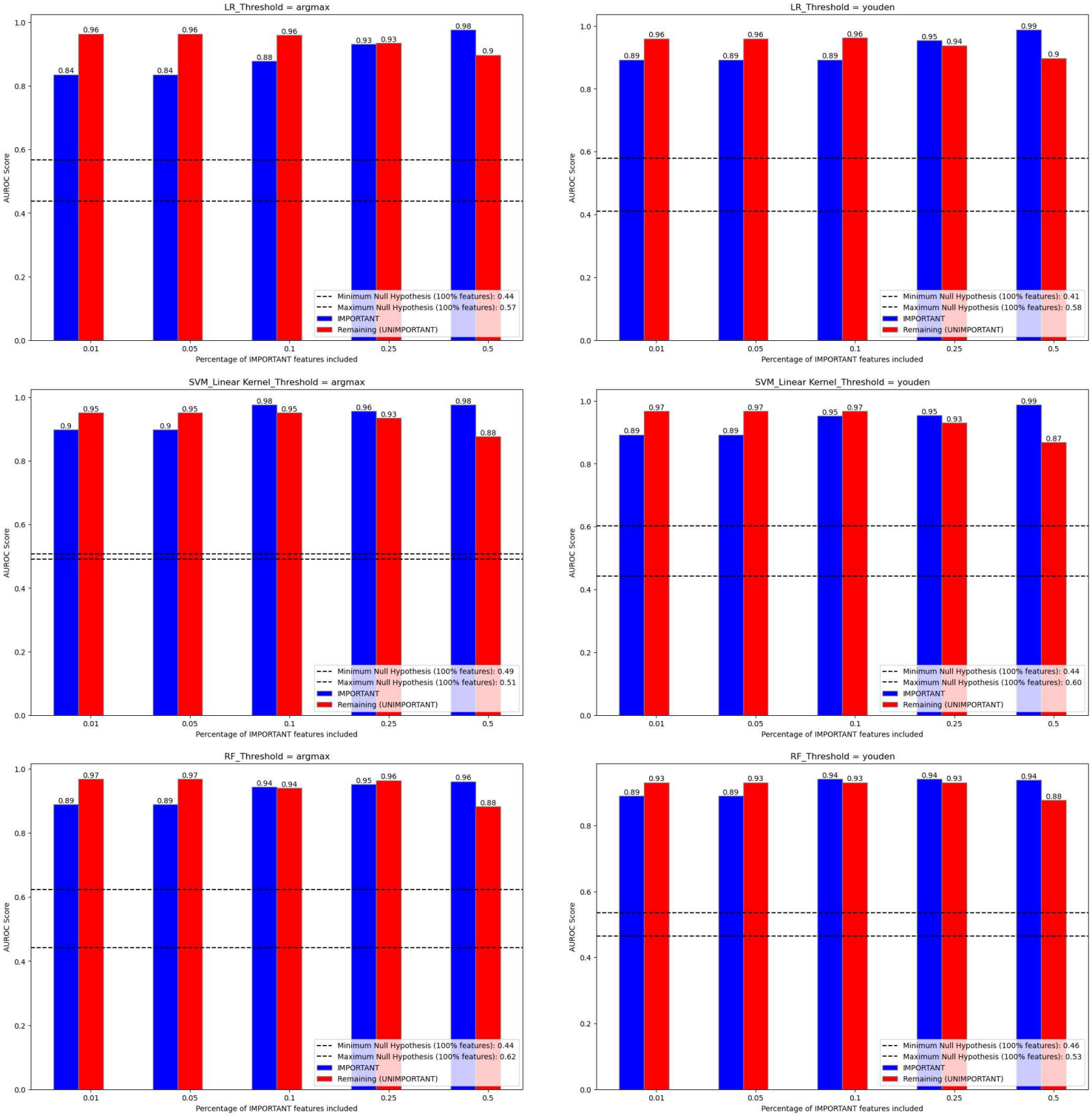
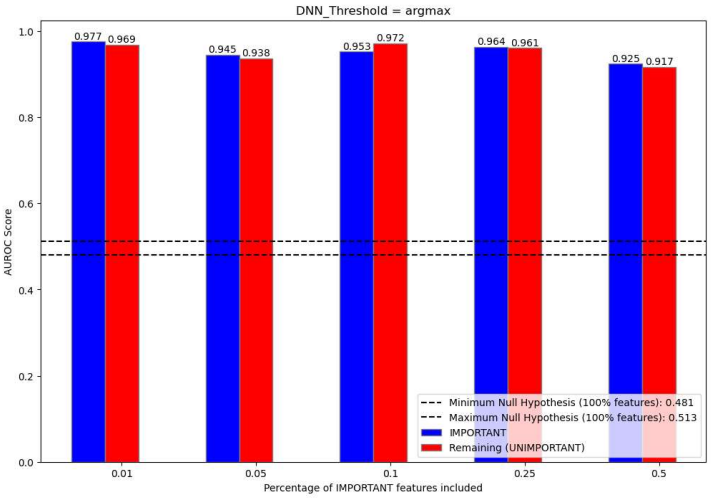


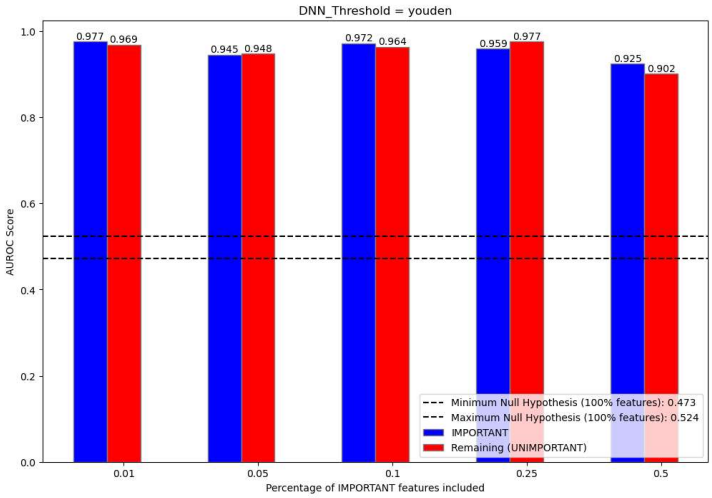
Table 18: Breast Cancer Wisconsin Results (100% of Features).

Model	**Threshold**	**Threshold Value**	**Precision (Class Malignant)**	**Recall (Class Malignant)**	**AUROC**	**PR-AUC**	**Null Hypothesis AUROC (Min, Max)**
LR	argmax	0.5000	0.9839	0.9531	0.9718	0.9773	(0.4400, 0.5710)
LR	youden	0.4788	0.9682	0.9531	0.9672	0.9695	(0.4120, 0.5810)
SVM_Linear Kernel	argmax	0.5000	0.9830	0.9062	0.9484	0.9622	(0.4900, 0.5100)
SVM_Linear Kernel	youden	0.3706	0.9682	0.9531	0.9672	0.9695	(0.4410, 0.6030)
RF	argmax	0.5000	1.000	0.9375	0.9688	0.9804	(0.4430, 0.6220)
RF	youden	0.5800	1.000	0.8906	0.9453	0.9658	(0.4610, 0.5300)
DNN	argmax	0.5000	1.000	0.9688	0.9844	0.9902	(0.4810, 0.5130)
DNN	youden	0.2223	1.000	0.9844	0.9922	0.9951	(0.4730, 0.5240)

Figure 21: Breast Cancer Wisconsin Results (Subset of Important Features).







The results shown between different datasets are mixed. In the first dataset example (MIMIC), we applied feature importance prioritizations to the same subset of models analyzed in the previous chapter. The models that have higher AUROC scores show significantly better performances than the null hypothesis baseline. Furthermore, the more performant models can distinguish a smaller subset of features that result in higher AUROC scores compared to the lower performance models. The opposite is true for poorer performing models. The SVM has the worst AUROC performance. It also happens to have scores with all feature splits that do not exceed the null hypothesis (random guessing baseline).

The top performing deep learning (DECONV CONV for 0.5 threshold, LSTM for youden) models also have the top performances for the subset of features tested on. This statement is true for 1% of features, 5% of features and 10% of features. The deep learning models with built in data transformations (DECONV-CONV, 1D CNN) to further isolate and extract features have superior performances on the 1% and 5% split of important features vs the respective 99% and 95% of respective unimportant features.

The white box model has overall poorer performance than the more performant black box models (DECONV-CONV, LSTM, 1D-CNN) on smaller importance splits (up to 25% important features). The poorer performing deep learning model (DNN) has less performance separation between the Important and Unimportant groups for all partitions.

In all the datasets where deep learning performed well (MIMIC and Wisconsin Breast Cancer), higher performance than classical machine learning/white box models can be

achieved with fewer features using SHAP values rather than the coefficients/importances. In the examples where deep learning models perform poorer than the white box/classical machine learning approaches, the feature selection does not result in any improvement, and reduces AUROC scores (DNN in the Diabetes Dataset, and Sepsis Dataset) to be closer to the null hypothesis. The classical machine learning importances and the white box coefficients result in feature selections that result in higher AUROC scores, and higher performance separation between the important, and unimportant features.

4.4 Discussion

Despite mixed results, there are indicators that do show the promise feature prioritizations based black box model SHAP values have. The main caveat we find is that the more performative the model through the AUROC metric, the better the feature prioritizations look. We base this assumption primarily on the fact that the more performative models show higher performance AUROC scores with as few as 1% - 25% of overall dataset features. This makes intuitive sense, since the better the model is at classifying ICU outcomes, the more it would be able to prioritize features relevant to the true label output.

In the situations where SHAP values performed poorly when selecting features, the underlying model used to generate the SHAP values also had poorer performance than the classical machine learning/white box models (e.g. Diabetes dataset). This is may not be a result of the SHAP values as feature selectors themselves but the model's ability to disseminate important from unimportant features. A more performant model (as indicated by the CPU models) can improve the selection of important vs unimportant features. We see related results with the SVM linear coefficients. The SVM did not fit well to all datasets except the Breast Cancer Wisconsin dataset (as indicated by its lower AUROC score). The feature prioritizations generated by this model in turn also resulted in poorer performance with no improvements and aligned more with the null hypothesis baseline of "random guessing". This principle of "Higher Performance = Better Feature Selection" seems to be universal among white box feature coefficients, black box SHAP values, and black box model importance values.

We noticed for all datasets, though more performant in some (MIMIC, Wisconsin Breast Cancer) than others (Diabetes dataset), the deep neural network had weaker performance separations between Good and Bad scores from an AUROC perspective. This can be primarily attributed to the lack of further feature transformation functionality, that the

other models (1D CNN, DECONV CONV, LSTM) possess through convolution, deconvolution, and LSTM layers. As a result, these models may be able to further distill more complex spatial or spatiotemporal relationships in the data that may make a feature more deterministic of the output than with no transformation.

5 Conclusion & Future Work

To conclude, we have conducted a series of exploratory experiments on a variety of clinical datasets to evaluate the applicability of post hoc explained black box models in clinical settings, and the applicability of model coefficients, importances and SHAP values as a feature selection tool. Based on the insights uncovered in the first series of experiments, there is evidence in the results to suggest that deep learning and black box models can make outcome predictions that prioritize correct, clinically relevant features when examined under the lense of post-hoc SHAP values. Some black box models had overall higher AUROC scores than the white box Logistic regression. This should be framed more as an initial view and would not replace common best practices of proper problem understanding, consulting experts, simplified model design, assessment, and validation. Our assessment in the first analysis was limited to a single dataset used for an outcome prediction task. To expand further, it would be advisable to expand the analysis to more patient datasets, and potentially more commonly performed clinical tasks (e.g. phenotyping, length of stay prediction, readmission prediction) to add additional rigor to the initial findings.

The second series of experiment's objective was to further validate the reliability of SHAP values and assess the application of SHAP values and model coefficients/importances as feature selectors. In higher performing models for certain datasets there was a performance distinction between important vs unimportant features for SHAP values and model coefficients/importances. In datasets where deep learning models had high AUROC scores, selecting features via their SHAP value magnitude showed higher performances with less features than with selecting features using model coefficients/model feature importances. The performances also exceeded a random guessing baseline. We would advocate further explore how generalizable the approach of selecting features based on post-hoc explain ability and white box model coefficients/model importances. A hybrid

ensembled approach combining both white box model coefficients, model feature importances, and black box post-hoc SHAP values may limit the drawbacks found on each individual approach. This hybrid approach can apply multiple approaches, and merge through pooling, or statistical aggregation. We would advocate for further model testing in both a tabular format, and with timeseries datasets, extending beyond the clinical use cases initially explored. Broader exploration may uncover further drawbacks to the methodology proposed that may not be readily apparent through a narrower scope.

References

[1] Cox, D. R. (1958). The regression analysis of binary sequences (with discussion). Journal of the Royal Statistical Society B, 20, 215-242.

[2] Pregibon, D. (1981). Logistic regression diagnostics. Annals of Statistics, 9, 705-724.

[3] Hosmer, D. W., & Lemeshow, S. (2000). Applied logistic regression (2nd ed.). New York: Wiley.

[4] D. G. Kleinbaum and M. Klein (2010). Logistic Regression: A Self-Learning Text (Statistics for Biology and Health).

[5] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32.

[6] Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18–22.

[7] Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. BMC Bioinformatics, 9, 307.

[8] Vapnik, V. N. (1999). An overview of statistical learning theory. IEEE transactions on neural networks, 10(5), 988-999.

[9] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[10] Schölkopf, B., Smola, A. J., & Müller, K. R. (1999). Kernel principal component analysis. In ICA (pp. 583-588). Springer.

[11] Duan, K. B., Keerthi, S. S., & Poo, A. N. (2005). Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing, 68, 296-311.

[12] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin of Mathematical Biophysics, 5(4), 115–133.

[13] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.

[14] LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (2012). Efficient backprop. In Neural networks: Tricks of the trade (pp. 9-48). Springer, Berlin, Heidelberg.

[15] Gunning, D. (2017). Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web, 2.

[16] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1(4), 541–551.

[17] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

[18] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[19] Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In European Conference on Computer Vision (pp. 818-833). Springer.

[20] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

[21] Olah, C. (2015). Understanding LSTM Networks.

[22] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation, 12(10), 2451–2471.

[23] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. Proceedings of the National Academy of Sciences, 116(44), 22071–22080.

[24] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1(4), 541–551.

[25] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems 25 (pp. 1097-1105). Curran Associates, Inc.

[26] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[27] Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In European Conference on Computer Vision (pp. 818-833). Springer.

[28] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).

[29] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[30] Japkowicz, N., & Shah, M. (2011). Evaluating Learning Algorithms: A Classification Perspective. Cambridge University Press.

[31] Fawcett, T. (2006). An

[32] Youden, W. J. (1950). Index for rating diagnostic tests. Cancer, 3(1), 32-35.

[33] Fluss, R., Faraggi, D., & Reiser, B. (2005). Estimation of the Youden Index and its associated cutoff point. Biometrical Journal: Journal of Mathematical Methods in Biosciences, 47(4), 458-472.

[34] Hajian-Tilaki, K. (2013). Receiver Operating Characteristic (ROC) curve analysis for medical diagnostic test evaluation. Caspian journal of internal medicine, 4(2), 627.

[35] Ruopp, M. D., Perkins, N. J., Whitcomb, B. W., & Schisterman, E. F. (2008). Youden Index and optimal cut-point estimated from observations affected by a lower limit of detection. Biometrical Journal: Journal of Mathematical Methods in Biosciences, 50(3), 419-430.

[36] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in neural information processing systems (pp. 4765-4774).

[37] Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... & Lee, S. I. (2020). From local explanations to global understanding with explainable AI for trees. Nature machine intelligence, 2(1), 56-67.

[38] Chen, J., Song, L., Wainwright, M. J., & Jordan, M. I. (2018). Learning to explain: An information-theoretic perspective on model interpretation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8830-8839).

[39] Molnar, C. (2020). Interpretable machine learning. Lulu. com.

[40] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression, 3rd Edition. Wiley.

[41] Menard, S. (2002). Applied logistic regression analysis, 2nd Edition. Sage.

[42] Agresti, A. (2013). Categorical Data Analysis, 3rd Edition. Wiley.

[43] Zhang, Z. (2017). Logistic regression in medical research: beyond the basics. Annals of Translational Medicine, 5(16), 332.

[44] Molenberghs, G., & Verbeke, G. (2005). Models for Discrete Longitudinal Data (Springer Series in Statistics). Springer.

[45] Díaz-Uriarte, R., & De Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. BMC Bioinformatics, 7(1), 3

[46] Rubin, D. B. (1987). Multiple Imputation for Nonresponse in Surveys. John Wiley & Sons.

[47] Jolliffe, I. (2002). Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed. Springer, NY.

[48] Patro, S. G. K., & Sahu, K. K. (2015). Normalization: A Preprocessing Stage. ArXiv:1503.06462 [Cs]. Retrieved from http://arxiv.org/abs/1503.06462

[49] He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263–1284.

[50] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357.

[51] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence, 5(4), 221–232.

[52] Castelvecchi, D. (2016). Can we open the black box of AI? Nature News, 538(7623), 20.

[53] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206-215.

[54] Rudin, C., & Radin, J. (2019). Why Are We Using Black Box Models in AI When We Don't Need To? A Lesson From an Explainable AI Competition. *Harvard Data Science Review*, *1*(2).

[55] Johnson, A., Pollard, T., Shen, L. *et al.* MIMIC-III, a freely accessible critical care database. *Sci Data* 3, 160035 (2016).

[56] Harutyunyan, H., Khachatrian, H., Kale, D.C. *et al.* Multitask learning and benchmarking with clinical time series data. *Sci Data* 6, 96 (2019).

[57] Rajkomar, A., Oren, E., Chen, K. *et al.* Scalable and accurate deep learning with electronic health records. *npj Digital Med* **1**, 18 (2018).

[58] Meyer A, Zverinski D, Pfahringer B, Kempfert J, Kuehne T, Sündermann SH, Stamm C, Hofmann T, Falk V, Eickhoff C. Machine learning for real-time prediction of complications in critical care: a retrospective study. Lancet Respir Med. 2018 Dec;6(12):905-914.

[59] Che, Z., Purushotham, S., Cho, K. *et al.* Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Sci Rep* **8**, 6085 (2018).

[60] Avati, A., Jung, K., Harman, S. *et al.* Improving palliative care with deep learning. *BMC Med Inform Decis Mak* 18 (Suppl 4), 122 (2018)

[61] Stenwig, E., Salvi, G., Rossi, P.S. *et al.* Comparative analysis of explainable machine learning prediction models for hospital mortality. *BMC Med Res Methodol* 22, 53 (2022).

[62] Chan MC, Pai KC, Su SA, Wang MS, Wu CL, Chao WC. Explainable machine learning to predict long-term mortality in critically ill ventilated patients: a retrospective study in central Taiwan. BMC Med Inform Decis Mak. 2022 Mar 25;22(1):75.

[63] Hu, C., Li, L., Li, Y. *et al.* Explainable Machine-Learning Model for Prediction of In-Hospital Mortality in Septic Patients Requiring Intensive Care Unit Readmission. *Infect Dis Ther* **11**, 1695–1713 (2022).

[64] Lu S, Chen R, Wei W, Belovsky M, Lu X. Understanding Heart Failure Patients EHR Clinical Features via SHAP Interpretation of Tree-Based Machine Learning Model Predictions. AMIA Annu Symp Proc. 2022 Feb 21;2021:813-822.

[65] Seki T, Kawazoe Y, Ohe K. Machine learning-based prediction of in-hospital mortality using admission laboratory data: A retrospective, single-site study using electronic health record data. PLoS One. 2021 Feb 5;16(2):e0246640.

[66] Pollard, T., Johnson, A., Raffa, J. *et al.* The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Sci Data* **5**, 180178 (2018).

[67] Y. Yang, W. Zhang, J. Wu, W. Zhao, and A. Chen, "Deconvolution-and-convolution networks," arXiv.org, https://arxiv.org/abs/2103.11887.

[68] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118 Chicco,Davide and Jurman,Giuseppe. (2023). Sepsis Survival Minimal Clinical Records. UCI Machine Learning Repository.

[69] Clore,John, Cios,Krzysztof, DeShazo,Jon, and Strack,Beata. (2014). Diabetes 130-US hospitals for years 1999-2008. UCI Machine Learning Repository.

[70] Antal,Balint and Hajdu,Andras. (2014). Diabetic Retinopathy Debrecen. UCI Machine Learning Repository.

[71] Knaus, W. A., Draper, E. A., Wagner, D. P., & Zimmerman, J. E. (1985). APACHE II: a severity of disease classification system. Critical care medicine, 13(10), 818-829.

[72] Zimmerman, J. E., Kramer, A. A., McNair, D. S., & Malila, F. M. (2006). Acute Physiology and Chronic Health Evaluation (APACHE) IV: hospital mortality assessment for today's critically ill patients. Critical care medicine, 34(5), 1297-1310.

[73] Ramsay, P., Haneuse, S., Walkey, A., & Wiener, R. (2020). Informing ICU care using observational data. Current Opinion in Critical Care, 26(5), 484-491.

[74] Knaus, W. A., Wagner, D. P., Draper, E. A., Zimmerman, J. E., Bergner, M., Bastos, P. G., Sirio, C. A., Murphy, D. J., Lotring, T., Damiano, A., & Harrell, F. E. (1993). The APACHE III prognostic system. Risk prediction of hospital mortality for critically ill hospitalized adults. Chest, 104(6), 1833-1859.

[75] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1721-1730).

[76] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).

[77] Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat (2019). Deep learning and process understanding for data-driven Earth system science. Nature, 566(7743), 195–204.

[78] Angwin, J., Larson, J., Mattu, S., & Kirchner, L. (2016). Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. ProPublica.

[79] Xie, J., Sun, J., Li, X., & Wu, H. (2018). Review of explainable machine learning-based pedagogical strategies for education. In Proceedings of the 8th International Conference on Educational Data Mining.

[80] McQuarrie, S., Dumitriu, D., & Reich, B. J. (2020). Predicting emergency department visits for respiratory illness using machine learning. arXiv preprint arXiv:2004.03661.

[81] Johnson, Alistair, et al. "MIMIC-III Clinical Database" (version 1.4). PhysioNet (2016), https://doi.org/10.13026/C2XW26.

[82] Yu, C., Liu, J., Nemati, S., & Yin, G. (2021). Reinforcement learning in healthcare: A survey. ACM Computing Surveys (CSUR), 55(1), 1-36.

[83] Zhang, Z., & Sejdić, E. (2019). Radiological images and machine learning: trends, perspectives, and prospects. Computers in biology and medicine, 108, 354-370.

[84] Sidey-Gibbons, J., Sidey-Gibbons, C. Machine learning in medicine: a practical introduction. BMC Med Res Methodol 19, 64 (2019). https://doi.org/10.1186/s12874-019-0681-4

[85] Van Walraven, Carl, et al. "Derivation and validation of an index to predict early death or unplanned readmission after discharge from hospital to the community." Cmaj 182.6 (2010): 551-557.

[86] Voskuijl T, Hageman M, Ring D. Higher Charlson Comorbidity Index Scores are associated with readmission after orthopaedic surgery. Clin Orthop Relat Res. 2014 May;472(5):1638-44.

[87] Sheng S, Xu FQ, Zhang YH, Huang Y. Charlson Comorbidity Index is correlated with all-cause readmission within six months in patients with heart failure: a retrospective cohort study in China. BMC Cardiovasc Disord. 2023 Mar 6;23(1):111.

[88] Wolberg,William, Mangasarian,Olvi, Street,Nick, and Street,W.. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository.

A List of Abbreviations

DNN	Deep Neural Network
1D CNN	1Dimensional Convolutional Neural Network
LSTM	Long-Term Short-Term Memory
DECONV CONV	Deconvolutional Convolutional Network
AUROC	Area Under Receiver Operating Characteristic Curve
PR AUC	Area Under Precision Recall Curve
EHR	Electronic Health Records
ICU	Intensive Care Unit