Predicting Endotracheal Intubation of Pneumonia Patients using Deep Learning

By Zachary Barrett

Advised by Dr. Michael Puthawala

A research paper submitted in partial fulfillment of the requirements for

Master of Science

Major in Mathematics-Statistics

South Dakota State University

2023

Abstract

Pneumonia is a leading cause for admission to intensive care units (ICUs) [7]. Critically ill patients suffering from lung diseases often require endotracheal intubation to assist or regulate their breathing. Equipping healthcare professionals with accurate predictions of intubation likelihood leads to better anticipation critical events, more effective allocation medical resources, and improved patient outcomes.

This paper evaluates traditional methods, such as decision trees, as well as deep learning models, including long short-term memory (LSTM) networks, in predicting the likelihood of intubation among ICU patients with pneumonia. We utilize medical data sourced from the MIMIC-III database. Specifically, we extract the vital signs recorded during the initial four hours following patients' admission to the hospital. By focusing on this critical time frame, we aim to capture early indicators and patterns that may be informative for predicting the need for intubation among pneumonia patients in the ICU.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Pneumonia and Acute Respiratory Distress Syndrome

Pneumonia is a lung infection which causes the air sacs (called alveoli) to fill with fluid, resulting in difficulty breathing [15]. Pneumonia is one of the most common causes of admission to intensive care units (ICUs). In some reports, the mortality rates associated with community-acquired pneumonia (CAP) and hospital-acquired pneumonia (HAP) requiring admission to the ICU have reached 50% [7].

One common complication of pneumonia is acute respiratory distress syndrome(ARDS), which is a severe form of respiratory failure [14]. ARDS develops when the blood doesn't have enough oxygen, has too much carbon dioxide, or both. In pneumonia patients, ARDS occurs when fluid buildup in the alveoli prevents the lungs from filling with air, and results in dangerously low blood oxygen levels (also called hypoxemia) [13].

In the United States, nearly 3 million patients annually experience ARDS which contributes to approximately 24% of patients receiving mechanical ventilation. This potentially fatal respiratory syndrome can be triggered by pulmonary causes(such as pneumonia) and is associated with high mortality rates between 35% to 46% based on disease severity at the onset [11]. Although mild cases of ARDS may respond to noninvasive ventilation, most patients require sedation, intubation, and ventilation while the underlying injury is treated [12].

It's been shown that early recognition of respiratory failure and prompt intubation can improve the outcomes in patients with CAP [6]. In the ICU setting, avoidable deaths during major airway events have been linked to gaps in care including: poor identification of at-risk patients, poor or incomplete planning, inadequate provision of skilled staff and equipment to manage these events successfully, and delayed recognition of events [3]. In this project, we aim to identify which early vital signs are associated with increased odds of endotracheal intubation to aid physicians plan for and deliver prompt life-saving treatment.

1.2 Machine Learning and Deep Learning

Traditional direct medical models look to estimate, simulate, or otherwise predict the state of a patient as a function of direct observations. Direct modeling works well for predicting the behavior of systems that are well-observed and obey known underlying mathematical/physical laws and

which don't vary significantly from patient to patient. For example, direct models could be used to predict the concentration of a drug in a patient's bloodstream as a function of time. The human body follows a known process of metabolizing drugs. Factors such as age, weight, liver and kidney function, or administration method impact the metabolizing rate of a patient. It is well-known and understood how these factors increase or decrease the metabolizing rate. Because of these well-observed behaviors and properties, direct modeling is a worthwhile approach.

There are, however, situations, such as patients in a critical care setting, where traditional methods fall short. In the ICU, observed patient data is multi-dimensional, the available features are unknown, and the features that are present vary from patient to patient. One patient might have blood tests and heart rate measurements, while another has a chest x-ray and blood pressure readings. With a direct modeling approach, it's unclear how to compare those two patients. Further, we might not know which features are relevant to the prediction task, or what laws those features follow. For critical care patients there isn't always enough time to run a suite of test in order to get enough conclusive observations to allow for a direct model. These issues, among others, pose challenges for direct modeling approaches.

Machine Learning (ML) is an indirect modeling approach that uses some guiding principals plus lots of data to uncover governing mathematical/physical laws. ML is much less constrained on the types of patterns or features that it can discover as compared to traditional models. Deep Learning (DL) is a subset of ML which leverages more sophisticated frameworks involving a lot more parameters. In medical modeling, DL has been has been successfully applied to detecting diabetic retinopathy and macular edema in patients using retinal images [5], classifying images of skin lesions as being cancerous [4], and predicting suicide risk using electronic medical records [17].

In medical prediction tasks, one of the biggest limitations is not knowing the available or relevant features for patients. This is one major reason why DL approaches are a natural choice for medical prediction tasks. We compare the performance of DL frameworks with traditional direct approaches with the goal of identifying patients that are at risk of endotracheal intubation, .

1.3 Supervised and Unsupervised Tasks

Machine learning can be split into many subcategories, two of them being supervised learning and unsupervised learning. Settings that call for an unsupervised approach typically don't have a known ground truth to begin with. For example, we are analyzing patients who visit the Emergency Room on weeknights, with the goal of identifying groups of people that display similar characteristics. There isn't a known number of groups, or set of characteristics that each group displays; that is precisely what we are hoping to learn without the help of a known ground truth.

In a supervised learning task, a computer is given a structured set of training input and output data, and tasked with determining the best way to predict the outcomes when given new input data. Each line of input data has a label, or a ground truth, which is the outcome the model aims to accurately predict for new data. For example, we have a database of past patients and their associated outcomes, namely if the patient was intubated. The ground truth is whether or not the patient was intubated and the training input is a patient's vital signs upon entering the ICU. If we input an intubated patient's vitals into a model, we desire the output to identify that person is at risk of needing intubation. The label associated with each entry gives us a simple way to assess how accurate predictions are: does the model output match the known truth?

2 The MIMIC-III Database

2.1 Introduction to MIMIC-III

The MIMIC-III database contains deidentified information on patients who stayed in critical care units at Beth Israel Medical Center located in Boston, MA between 2001-2012. The database includes demographic information, laboratory test results, medications, procedures, and vital sign measurements. The database consists of 26 tables, which are linked by patient and admission identifiers.

A diagnosis is the result of a medical professional analyzing symptoms, running tests, and finally concluding what disease or condition explains a patient's symptoms. In the MIMIC-III database, the DIAGNOSIS field is a free-text entry field for attending clinicians to enter for each visit. For each patient, healthcare workers have the option of selecting one or multiple diagnoses from a predetermined list or enter their own text as they see fit. Chest x-rays in tandem with blood and sputum tests are conclusive in confirming a pneumonia diagnosis, leading to low rates

of false positive diagnoses.

Table 1 summarizes the frequencies (over the entire MIMIC-III database) of different combinations of diagnoses which contain pneumonia, sorted by frequency. There are over unique 200 diagnosis entries that contain "PNEUMONIA" along with combinations of other diagnoses. To avoid observing health complications that are a result of conditions unrelated to pneumonia, we focus on roughly 1500 patients that were solely diagnosed with pneumonia.

Diagnosis	**Frequency**
PNEUMONIA	1566
PNEUMONIA; TELEMETRY	44
PNEUMONIA; SEPSIS	19
PNEUMONIA; CONGESTIVE HEART FAILURE	15
PNEUMONIA; CHRONIC OBST PULM DISEASE	8
PNEUMONIA; HYPOTENSION	7

Table 1: Frequency of Pneumonia Diagnosis Combinations

For this project we focus on adults ages 18 to 65. For each patient we extracted the following vital sign metrics over the duration of their ICU visit: heart rate, blood pressure (systolic and diastolic), blood oxygen (spo2), and respiratory rate. Tables 2, 3 summarize demographic information of the 547 adult patients diagnosed with pneumonia upon their admission to the ICU.

Admission Location	**Count**	**Percent**
Emergency room	335	61.2%
Clinic referral	170	31.1%
Transfer from hospital	37	6.8%
Other	5	0.9%

Table 2: Summary Admission Location

Intubation Need	**Count**	**Percent**
Intubated	154	28%
Non-Intubated	393	72%

Table 3: Adults with Pneumonia

2.2 Pneumonia Patient Vital Readings

As discussed in [16], missingness/sparsity as well as irregular/sporadic medical recordings pose challenges for Deep Learning. In the hospital, not all vital signs are measured at regular time

intervals; they may be recorded sporadically in time depending on the underlying condition of the patient. This results in some variables being sparse compared to others, as they are measured more frequently for a given patient. Further, a patient's current condition may demand observing only a subset of the variables of interest. Figure 1 demonstrates some of these issues.

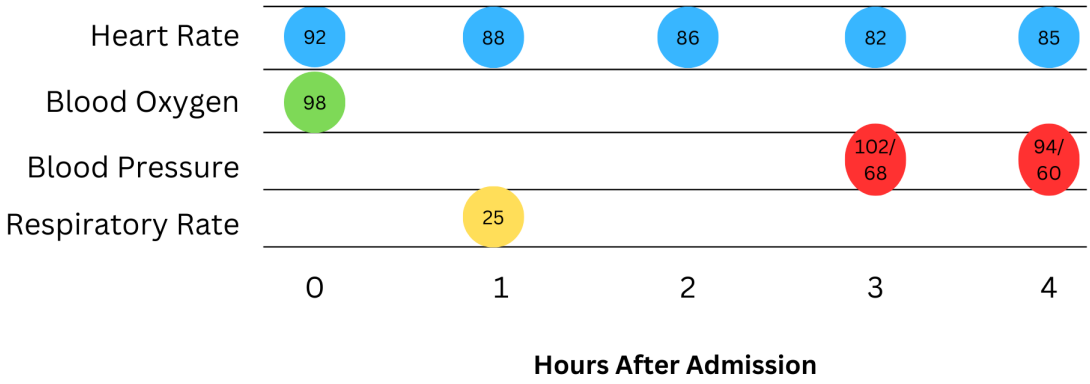

Figure 1: Sample Vital Sign Recordings

Created using canva.com

In a clinical setting, patients whose condition is at risk of quickly worsening are referred to as high-acuity patients. For pneumonia-diagnosed adults in the MIMIC-III database, acuity is highly and positively correlated with the frequency of vital sign measurements. High-acuity patients are at risk of a major medical event, so they must be monitored more closely by nurses and doctors. This results in more frequent vital sign readings compared to a patient whose condition is stable or improving. Regardless of what those measurements are, patients with more vitals recorded have an increased likelihood of negative health outcomes. This issue is further elaborated on in section 5.2.

For this project, we take the first 4 hours after admission and aggregate the recorded vital signs into hourly time intervals. Aggregating vital signs within a time interval can help alleviate some of that sparsity, but also results in some more granular data being suppressed [16]. We remove any error codes, which are recorded in the database as negative integers, and replace them with blanks. For the Deep Learning models, we normalize each feature to be normally distributed with mean 0 and standard deviation 1 across the training split(80%), and applied the same rules to the testing split(20%) to avoid data leakage. Finally, all missing values (including the error codes mentioned earlier) are replaced with 0's.

3 Model Methodologies

3.1 Multi-layer Perceptron Networks

Multi-layer perceptrons (MLPs) are a class of artificial neural networks formed by fully connected neuron stacks, commonly called layers. MLPs consist of an input layer, hidden layer(s), and an output layer in which each neuron is connected to all neurons in the following layer. There are two types of inputs into each neuron: a bias and a weighted sum of the neurons in the previous layer. The weights of the sum and the components of the bias vector are optimized (or learned) through a loss function by using the training data. The weighted sum is added to the bias and then passed into an activation function before getting passed on to the next layer. Below is a diagram outlining the structure of a MLP network.

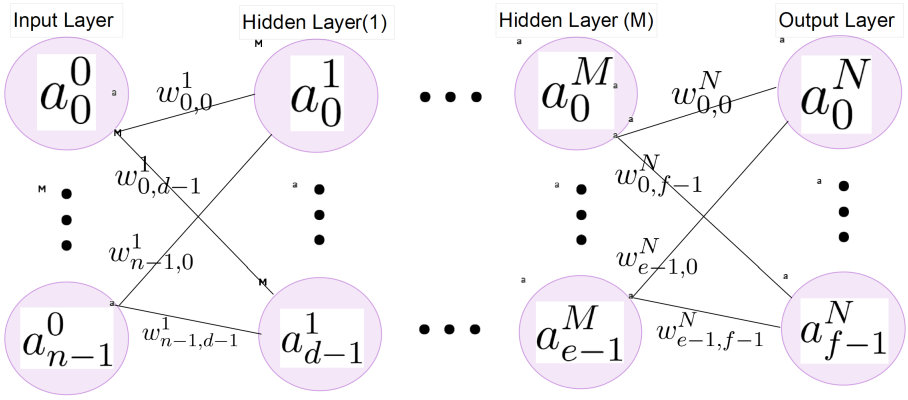

Figure 2: MLP Structure Diagram

Created using Visual Paradigm

To demonstrate the information being passed between layers of the network, we calculate the neuron values of the ith layer of a network. We express the bias as the vector as B_i, where b_j^i is the bias associated with the j^{th} neuron in layer i. The weights linking the neurons in the previous layer $(i-1)$ to the current layer (i) are arranged into the matrix $W_{k,n}^i$, where k represents the number of neurons in the current layer, and n represents the number of neurons in the previous layer. For example, the entry $w_{0,1}^i$ is the weight associated with the arc linking neuron 0 in the previous layer to neuron 1 in the current layer. We also arrange the values of the neurons in the previous layer into the vector A_i, where a_0^{i-1} is the output value of neuron 0 in the previous layer $(i-1)$. Multiplying the matrix $W_{k,n}^i$ by the vector A_i results in a vector of summed weights of

the previous layer of neurons. Summing that vector with our bias vector B_i gives us the total calculated input into the neurons of the ith layer of a network.

$$B_i = \begin{bmatrix} b_0^i \\ b_1^i \\ \vdots \\ b_{k-1}^i \end{bmatrix}, \; W_{k,n}^i = \begin{bmatrix} w_{0,0}^i & w_{1,0}^i & \cdots & w_{n-1,0}^i \\ w_{0,1}^i & w_{1,1}^i & \cdots & w_{n-1,1}^i \\ \vdots & \vdots & \ddots & \vdots \\ w_{0,k-1}^i & w_{1,k-1}^i & \cdots & w_{n-1,k-1}^i \end{bmatrix}, \; A_i = \begin{bmatrix} a_0^{i-1} \\ a_1^{i-1} \\ \vdots \\ a_{n-1}^{i-1} \end{bmatrix},$$

Using matrix multiplication, we can express the value of the neurons of the $(i+1)^{th}$ layer of a MLP as:

$$A_{i+1} = \sigma\left(W_{k,n}^i A_i + B_i\right) = R\left(\begin{bmatrix} w_{0,0}^i & w_{1,0}^i & \cdots & w_{n-1,0}^i \\ w_{0,1}^i & w_{1,1}^i & \cdots & w_{n-1,1}^i \\ \vdots & \vdots & \ddots & \vdots \\ w_{0,k-1}^i & w_{1,k-1}^i & \cdots & w_{n-1,k-1}^i \end{bmatrix} \begin{bmatrix} a_0^{i-1} \\ a_1^{i-1} \\ \vdots \\ a_{n-1}^{i-1} \end{bmatrix} + \begin{bmatrix} b_0^i \\ b_1^i \\ \vdots \\ b_{k-1}^i \end{bmatrix}\right), \quad (1)$$

where $R(x) = max\left(0, x\right)$, is the ReLU activation function.

As depicted in Figure 2, every individual neuron is connected to all of the neurons in the previous layer. The weights of the sum and the components of the bias vector are learned through training, and are optimized using a process called gradient descent. First, we define the cost function

$$C(w, b) = \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{n} \quad (2)$$

with w representing the weights of the sum, and b being the biases vector. This function evaluates how far the model's predictions y' are from the ground truth, denoted y. Using partial derivatives, MLPs identify how to adjust the weights of the sum and the components of the bias vector to minimize this cost function, or make the predictions closer to the truth.

With large amounts of data in high dimensions it becomes computationally expensive to process individual observations, update the weights and biases, and repeat. Processes like mini-batch gradient descent are commonly used to avoid this issue. In this method, the training data is randomly split into small batches. Next, one of those batches is randomly selected and the gradients within that batch are calculated. Those gradients are averaged, and that average is used to estimate the true gradient across all training datapoints. That average gradient calculation is used to update the weights and biases of the network, and then the process is repeated for another randomly selected batch.

Learning rates control how much adjusting happens to the weights of the sum and the components of the bias vector. It is common to use adaptive learning rates that larger adjustments when the gradient is steep, and smaller adjustments as the loss function nears local minimums. Adaptive learning rate methods provide faster convergence and improved performance when compared to fixed learning rates.

3.2 Long Short-Term Memory Networks

The information in the MIMIC-III database is timestamped, and so forms a timeline of a person's health across their visit. Recurrent Neural Networks (RNNs) are a class of network structures that operate well on timestamped data. In particular, Long Short-Term Memory Networks (LSTMs) are a natural choice.

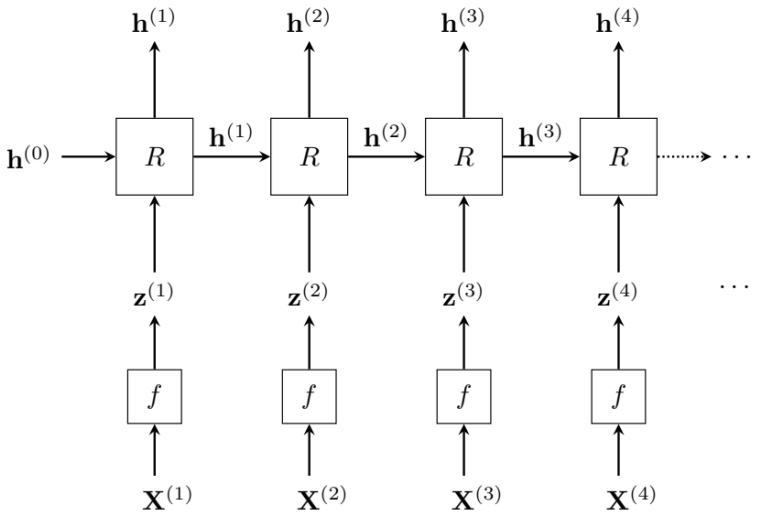

Figure 3: RNN Flow Diagram

Source: Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges [1]

The input to an RNN can be viewed as an arbitrary number of steps through time, where at each step (t) we are provided with an input signal $X^{(t)} \in X(\Omega)$ where $X(\Omega)$ is the domain. As a concrete example, $X^{(0)}$ is a blood pressure reading for a patient upon entering the ICU, $X^{(1)}$ is the patient's blood pressure reading one hour after admission, etc. We then use an encoder function $f\left(X^{(t)}\right) = z^{(t)}$ to flatten the input into the vector $z^{(t)}$. This step is important when the input is not in a vector. For example, if the input to an RNN is a sequence of images, the matricies of RGB values must be flattened into a vector. At each step, a summary vector

$h^{(t)} = R\left(z^{(t)}, h^{(t-1)}\right)$ is computed using the update function R. The summary vector for time t is solely based on the current features and the previous summary vector; the previous step's input features are completely overwritten.

We require a model architecture that produces the same predicted likelihood of needing intubation, no matter if they come into the hospital today or a week from now. In technical terms, this describes a model that is invariant with respect to time. This symmetry, a transofmation that leaves an object or system unchanged, is satisfied by LSTMs time-shift invariant structure.

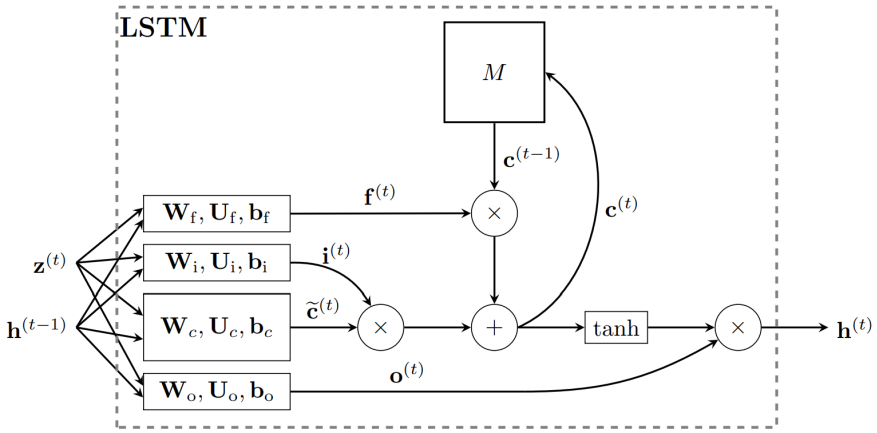

Figure 4: LSTM Flow Diagram

Source: Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges [1]

LSTM's differ from generic RNNs in their addition of memory cells that store cell state vectors and are passed between computational steps. Cell state vectors ($c^{(t)}$) are computed using the previous summary vector, the current input vector, and the previous cell state vector. Cell state vectors are not allowed to completely overwrite the memory cell, a portion of the previous cell state vector $c^{(t-1)}$ must be retained, hence the term "memory cell". For this reason, the current cell state vector is denoted $\tilde{c}^{(t)}$ and called the vector of candidate features, since not all of this vector is allowed to enter the memory cell. At each time step, the candidate features are calculated based off the current input vector and the previous summary vector. This is expressed as:

$$\tilde{c}^{(t)} = z^{(t)} \times h^{(t-1)},$$

where $\times$ is element-wise vector multiplication. The LSTM uses gating vectors, which range between $[0, 1]$ to control how much of the new signal should impact the current cell state. There are 3 gates: the input gate ($i^{(t)}$), the forget gate ($f^{(t)}$), and the output gate ($o^{(t)}$). Figure 4 visualizes

the relationships between the gating vectors and shows how they control what information reaches the memory cell (M) and the summary vector ($h^{(t)}$). The cell state vector is regulated by the input gate which controls how much of the new signal can impact the current state and the forget gate determines how much of the previous cell state should be retained. The cell state vector is a function of the candidate features regulated by the input gate, and the previous cell state vector regulated by the forget gate, which is expressed as:

$$c^{(t)} = \tilde{c}^{(t)} \times i^{(t)} + c^{(t-1)} \times f^{(t)}.$$

The output gate controls the proportion of the new cell state to use for the final summary vector. The summary vector is calculated based on the cell state vector sent through the an activation function, this diagram uses the *tanh* activation function, and regulated by the output gate. This is expressed as:

$$h^{(t)} = tanh(c^{(t)}) \times o^{(t)}.$$

3.3 Tree Models

Decision trees are one of the most popular supervised learning algorithms in classification. They are popular in part because of their ability to generate interpretable and explainable results. Each decision tree involves three main components: a root, branches, and leaves. Similar to a real tree, leaves are connected to root nodes through branches to create a hierarchical structure. At each branch, a binary decision is made based on a learned rule relating to one feature of the given data. Based on that decision, the point is then passed down to other nodes until eventually reaching a leaf node which represents a predicted outcome.

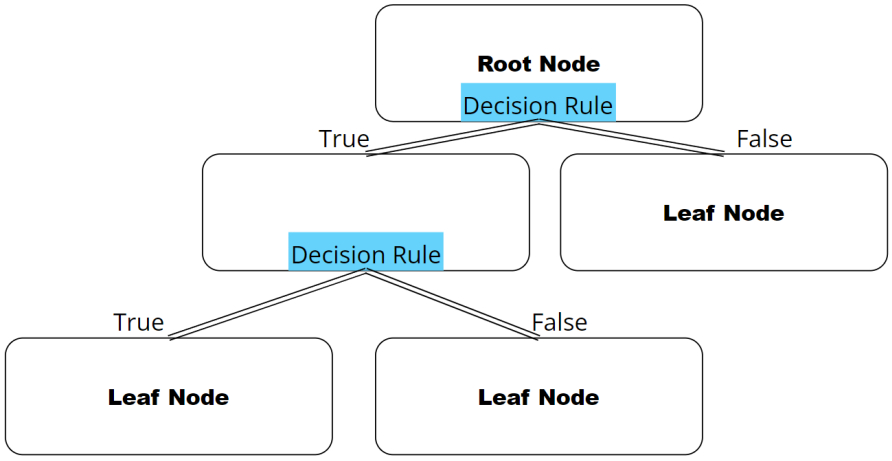

Figure 5: Tree Structure Diagram

Created using Visual Paradigm

Decision trees learn what rules are the best at separating the sample data into groups, in our case that'd mean separating out intubated and non-intubated patients. Gini index is one of the ways to optimize the rules controlling how the branches split the data. Gini index takes on a value from 0 to 1, describing how well the rule splits the two types of individuals. In the decision tree context, a higher Gini index corresponds to a more homogeneous group of people within a node. For example: if a proposed decision rule has a Gini index of 1, that means we have perfectly separated all intubated and non-intubated patients, as each group is completely homogeneous.

4 Model Evaluation

Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) statistics are common model performance metrics used in classification problems. They measure how well a model can separate between classes; in our case, how well can a model separate the intubated patients from the non-intubated.

The first step in constructing a ROC curve is rank ordering patients by their predicted probability of needing intubation. As shown in Figure 6, let the cutoff value be V. All patients with a predicted likelihood above V are predicted as requiring intubation, all patients lower than V are predicted to not need intubation. Given a particular cutoff value V, there are four different situations that can occur, which are summarized in Table 4.

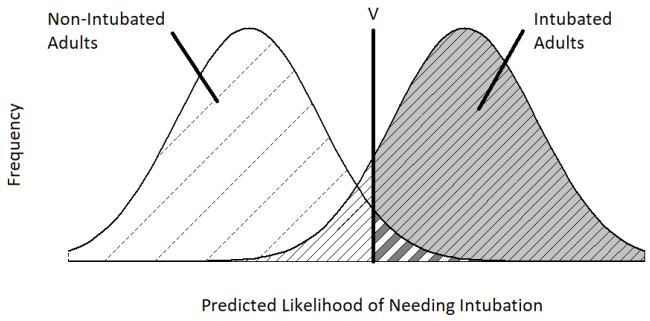

Figure 6: ROC Cutoff Illustration
Created using R

		Truth	
		Intubated	**Not-Intubated**
Prediction	$\geq V$	True Positive	False Positive
	$< V$	False Negative	True Negative

Table 4: Confusion Matrix

If a patient has a predicted likelihood of needing intubation greater than V, and actually needs intubation, we have a correct prediction. If the predicted likelihood is above V but the patient doesn't actually need intubation, we have a false positive. In figure 6, this is depicted by the region colored with grey and white stripes. The false positive rate (at a given cutoff point) is the fraction of times that the model falsely classifies a patient as needing intubation. That is:

$$\text{False Positive Rate} = \frac{\text{Number of False Positives}}{\text{Number of True Negatives} + \text{Number of False Positives}} \tag{3}$$

False negatives occur when a patient has a predicted likelihood of needing intubation that is less than V, when they actually need intubation. The false negative rate (at a given cutoff point is the fraction of times the falsely classifies a patient as not needing to be intubated. That is:

$$\text{False Negative Rate} = \frac{\text{Number of False Negatives}}{\text{Number of True Positives} + \text{Number of False Negatives}} \tag{4}$$

The compliments of the false positive rates and false negative rates are the proportion of times the model correctly assigns an intubated patient as needing intubation, and assigning non-intubated patients as not needing intubation. They can be expressed as:

$$\text{True Positive Rate} = 1 - \text{False Negative Rate} \tag{5}$$

$$\text{True Negative Rate} = 1 - \text{False Positive Rate} \tag{6}$$

To construct the ROC curve, a computer iterates through all possible cutoff values, calculating the false positive rate and true positive rate at each V. Finally, the computer plots the false positive rates versus the true positive rates.

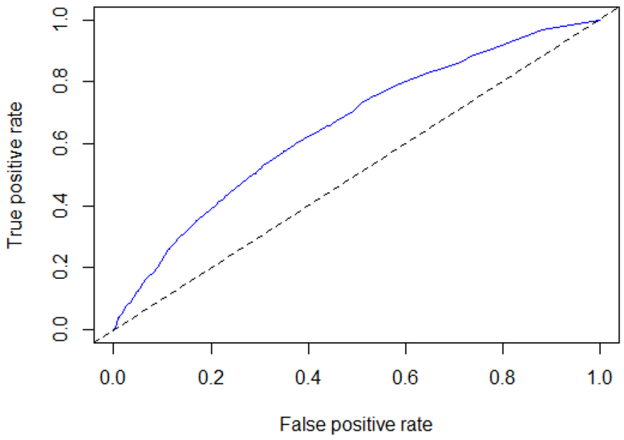

Figure 7: Example ROC Curve

Created using R

The 45-degree dashed line in the Figure 7 signifies a scenario where the model's true positive rate is equal to its false positive rate. In other words, when we classify a patient as needing intubation, we are equally likely to be correct as we are to be incorrect. The proximity of the curve to the point (0,1) indicates the model's accuracy. Achieving the point (0,1) denotes perfect identification of all intubated patients with zero false positives, signifying a perfect model. The AUC (Area Under the Curve) quantifies how closely the model approaches perfection, where an AUC of 1 represents the best possible performance.

Accuracy is another common model evaluation metric. At each cutoff point, the proportion of correct classifications is calculated. The maximum across all cutoff points is referred to as the accuracy. For example, if we found the highest proportion of correct predictions at the cutoff value of 0.43, with the 68% of patients being correctly classified, 0.68 is the accuracy of our model.

5 Results

5.1 A Simple Baseline Model

Table 3 shows that 72% of all pneumonia patients did not need intubation during their ICU admission. Hence, a natural baseline model would classify every patient as not needing intubation. Table 5 shows that based on the randomized train/test split, the accuracy on the testing data would be 64%.

Summary of Train Split		
Total	438	
Intubated	114	26%
Non-Intubated	324	74%

Summary of Test Split		
Total	111	
Intubated	40	36%
Non-Intubated	71	64%

Table 5: Test and Train Data Split Summaries

5.2 One-depth Decision Tree Performance

As described in section 2.2, sparsity of vital sign readings is a strong indicator of patient acuity. To demonstrate this, we have created a one-depth decision tree using the train/test data as summarized in table 5. The optimal split found by the decision tree was on the feature hour 1 respiratory rate, with branches created at the value of 13. Table 6 summarizes the distribution of respiratory rate values for patients in the testing split.

Hour 1 Resp Rate	**Frequency**
0	90
12	3
13	1
14-16	5
20-24	4
25-33	7

Table 6: Distribution of Respiratory Rate in Testing Split

Recall that a 0 in this column is equivalent to either an error value or no recorded respiratory

rate during the first hour. Hence, this decision tree is separating the testing data into two groups based on the presence, or absence, of a respiratory rate reading during the first hour of admission (with the exception of 4 people who had a low reading and were grouped in with the people who had no reading). The one-depth decision tree produced a prediction accuracy of 65.5% and an AUC of 0.562.

5.3 Decision Tree Performance

We now expand the single-depth decision tree to include more leaves and branches. For this model, we set the maximum tree depth to 3 and the minimum observations in each leaf to 7. Both of these decisions were made to avoid over-fitting, and retain the ability to interpret how those splits fit into the context of the problem. A brief study was conducted to determine the impact of adjusting these parameters on the accuracy of the resulting model. The results of this study are listed in Table 9 in Appendix A. Figure 10 in Appendix A shows a diagram of the resulting decision tree. This model has an accuracy of 70.9%, and an AUC of 0.64.

5.4 MLP Performance

We used the ReLU activation function for a three hidden layer MLP network. Each of the 3 layers are made up of 128 neurons, with the output being a number between 0 and 1. To optimize the weights and biases of the network, we used the ADAM solver [8]. This is a stochastic gradient-based optimizer, meaning gradients are calculated on small batches rather than the entire training set. For our model we set the size of each mini-batch to have 32 patients. For this dataset it probably wasn't necessary to use mini-batches due to the manageable number of patients. However, for future applications with significantly more patient data it is necessary to lower the computational workload in each training iteration. With these parameters, the MLP model achieved an accuracy of 68.2% and an AUC of 0.58.

5.5 LSTM Performance

A brief study was conducted to determine the optimal number of neurons and layers in our LSTM. We found that having two layers, each made up of 32 neurons, resulted in the highest prediction accuracy. At the end of those two layers, we used a fully-connected layer with the

sigmoid activation function, which can be expressed as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

The LSTM had an accuracy of 73.6% on the testing data, and an AUC of 0.634.

5.6 Model Comparison

Table 7 shows the performance metrics for each of the 5 models we evaluated. The top two performing models are the LSTM and the Decision Tree. These two models perform similarly when evaluated on AUC, but the LSTM has more accurate predictions compared to the Decision Tree. The next best performing model is the MLP, followed by the last two traditional models.

Model Structure	**AUC**	**Accuracy**
LSTM	0.634	73.6%
Decision Tree	0.64	70.9%
MLP	0.58	68.2%
One-depth Decision Tree	0.562	65.5%
Simple Baseline	N/A	64%

Table 7: Summary of Model Performance, Ordered by Accuracy

As described in Section 4, one of the first steps in plotting a ROC curve is rank-ordering the predictions. We interpret the patients who are represented on the left side of the plot as being most likely to need intubation, as predicted by the model. When we compare the ROC curves of each model depicted in Figure 8, we notice that the deep learning models (MLP and LSTM) out-perform both of the traditional methodologies. The deep learning models are better at picking out patients who are at high risk of needing intubation.

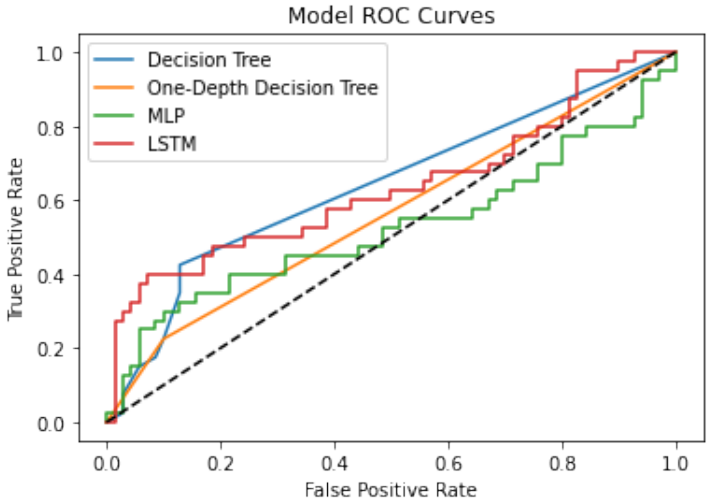

Figure 8: ROC Curve Comparison

On the right side of the graph are the patients who are least likely to need intubation, as predicted by each model. The LSTM is the most accurate at identifying these patients. Of the 4 models plotted on the ROC curve, the MLP model is the least accurate at labeling these individuals. Patients towards the middle of the x-axis are most accurately categorized by the decision tree model.

6 Discussion

6.1 Exploring Model Performance

The comparative ROC curve, depicted in Figure 8, shows that the deep learning models (MLP and LSTM) outperform traditional approaches in identifying patients who are most likely to require intubation. Additionally, the LSTM outperforms all other models in accurately identifying patients who are least likely to need intubation. We present two patient scenarios where the LSTM's memory-retention allows for the accurate identification of individuals with both high and low probabilities of requiring intubation.

Patient A is hooked up to a respiratory monitoring machine upon admission to the ICU. After two hours of close monitoring, doctors determine that their condition is improving enough to stop monitoring that vital sign. Traditional methods would classify this patient with all the other at-risk patients who have respiratory readings upon admission. However, this wouldn't be

accounting for the improvement in condition signaled by the absence of respiratory readings at hours three and four.

Patient B comes into the ICU with a sore throat and dry cough, and has exploratory lab tests run. Two hours after admission, their condition quickly worsens and requires strict monitoring by hospital staff. They are hooked up to machines that continuously monitor blood pressure, pulse, oxygen saturation, and respiratory rate. Traditional methods might group this patient with other individuals who don't have many vital sign recordings within the first two hours after admission. However, this wouldn't be accounting for the rapid decline in condition this patient is experiencing.

In both of these situations, the memory-retention structure of LSTM's allows this model to identify the progression or regression of a patient's health state. This is a distinct advantage in healthcare applications, since medical staff are interested in the history and progression of a patient's condition.

6.2 Imputation

In section 5.2, we stated that the single best predictor of intubation status is the presence of a respiratory rate reading within the first hour after admission. Our findings revealed that patients who had respiratory rate recordings were more likely to need intubation. Upon further analysis, we find this trend to be true with respiratory rate readings taken at all four hours. Table 8 shows the conditional probabilities of intubation based on whether a respiratory rate reading was present or absent within the first four hours of admission.

	P(intubated | recording)	**P(not intubated | no recording)**
Hour 1 Resp Rate	0.5175	0.7806
Hour 2 Resp Rate	0.5272	0.7459
Hour 3 Resp Rate	0.6571	0.7441
Hour 4 Resp Rate	0.5	0.739

Table 8: Conditional Probabilities of Intubation Based on Respiratory Rate Recording

We constructed tables of conditional probabilities for other vital signs and observed similar patterns. As discussed in Section 2.2, patients with a higher frequency of vital sign recordings

tend to have worse health outcomes.

Clinically appropriate data collection often does not occur on a regular schedule but rather is guided by patient condition and clinical or administrative requirements. Thus, electronic health record data are often only available at irregular intervals for selected variables that vary between patients and type of data. While the absence of recorded data may be clinically appropriate, machine learning algorithms' performance typically suffers from biased and incomplete data[9].

In an attempt to mitigate this issue, we conducted a review of imputation literature and concluded that the success of any of these methods would require a larger amount of data than what was available to us. The frequency of missing vital readings within each patient's records was exceptionally high. For instance, some patients had no respiratory rate readings during their visit. Given the sparsity and high dimensionality of the data, the imputation techniques we explored proved to be unsuitable for addressing these challenges.

6.3 Data Aggregation

The decision to aggregate vital signs on an hourly basis could be seen as a significant limitation of this project. Patients in critical care settings often experience rapid and substantial changes in their conditions over short periods of time. Aggregating vital signs into hourly intervals removes insights into the fluctuations and changes in a patient's condition within each time interval. However, for this project, the loss of information due to aggregation was fairly minimal. This was primarily because the majority of patients in the MIMIC-III database had, at most, three instances of the same vital sign reading within a given hour. In fact, many patients had either only one or no vital sign readings within each hour.

The choice to aggregate the data was made for practical reasons, aiming to establish a standardized framework for analyzing vital sign readings across all patients. However, we acknowledge that this approach may have led to the omission of valuable temporal information that could have provided insightful into the dynamic physiological changes experienced by the patients.

6.4 Limitations of the MIMIC-III Database

Utilizing the MIMIC-III database for research purposes is prevalent within the medical community; however, its use does have inherent limitations. Although Pneumonia is one of the most frequently diagnosed diseases in the MIMIC-III database, the number of adults diagnosed with

this condition is limited to just 547.

As highlighted in Section 1.2, deep learning algorithms rely on a substantial amount of data to uncover intricate relationships between input features. Unfortunately, due to the limited number of observations in our study, the complexity of patterns that can be revealed is considerably constrained.

A low patient count also restricts the depth of decision trees that can be constructed without risking overfitting. Decision trees are prone to overfitting when they become too complex relative the available data. With a limited number of patients, it becomes challenging to build deeper decision trees that capture more nuanced relationships and provide accurate predictions that generalize to larger populations.

The MIMIC-III database also presents a challenge due to the coexistence of two different standards for disease documentation. The data collection period for the MIMIC-III database coincided with the transition in the United States from the International Classification of Diseases, 9th revision (ICD-9), to the 10th revision (ICD-10). The presence of mismatched standards poses challenges when it comes to analyzing and extracting patient information from the MIMIC-III database. Despite these difficulties, it was not viable to exclude patients recorded using the ICD-9 standard due to the already low volume of patients. Consequently, patients documented under the ICD-9 standard were included in this analysis, even though their documentation does not align with current practices.

Additionally, we were unable to precisely determine when a patient was intubated during their visit. This lack of information introduces bias into our models since a patient's vital signs undergo significant changes once they are intubated. If we had access to the exact timing of intubation, we could also modify the prediction target to determine whether a patient needs intubation at that very moment based on current and past readings. This would increase the practical value of our models, as they would effectively aid in crucial decision-making during the timeframe between patient admission and the 4-hour mark.

6.5 Future Works

The models developed in this project exclusively rely on vital sign readings as the input features. However, there is potential for future research to incorporate additional factors into the models, such as laboratory results and patient demographic information. Existing research has concluded

racial and socioeconomic disparities influence the outcomes of hospitalized pneumonia patients [10, 2]. By incorporating an analysis of these disparities, as well as other relevant factors, into future modeling efforts, we can gain insights into the intricate relationships between patient characteristics, clinician interventions, and the outcomes of pneumonia patients. This knowledge can serve as a foundation for developing targeted interventions and strategies aimed at addressing disparities and improving the overall quality of patient care.

7 Acknowledgements

I would like to thank my advisor Dr. Michael Puthawala for his invaluable mentorship and unwavering support throughout my master's degree journey. His profound knowledge and consistent encouragement played a vital role in the completion of this research project. I would also like to thank Dr. Christine Puthawala for her guidance throughout the course of this project. Her clinical expertise played a pivotal role in shaping the scope of this research project to focus on a medically relevant area of study.

References

[1] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[2] Deron C Burton, Brendan Flannery, Nancy M Bennett, Monica M Farley, Ken Gershman, Lee H Harrison, Ruth Lynfield, Susan Petit, Arthur L Reingold, William Schaffner, et al. Socioeconomic and racial/ethnic disparities in the incidence of bacteremic pneumonia among us adults. *American journal of public health*, 100(10):1904–1911, 2010.

[3] T.M. Cook, N Woodall, J Harper, Jonathan Benger, and Fourth National Audit Project. Major complications of airway management in the uk: results of the fourth national audit project of the royal college of anaesthetists and the difficult airway society. part 2: intensive care and emergency departments. *British journal of anaesthesia*, 106(5):632–642, 2011.

[4] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

[5] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[6] Sami Hraiech, Julie Alingrin, Stéphanie Dizier, Julie Brunet, Jean-Marie Forel, Bernard La Scola, Antoine Roch, Laurent Papazian, and Vanessa Pauly. Time to intubation is associated with outcome in patients with community-acquired pneumonia. *PLoS One*, 8(9):e74937, 2013.

[7] J Karhu, TI Ala-Kokko, P Ylipalosaari, P Ohtonen, JJ Laurila, and H Syrjälä. Hospital and long-term outcomes of icu-treated severe community-and hospital-acquired, and ventilator-associated pneumonia patients. *Acta anaesthesiologica scandinavica*, 55(10):1254–1260, 2011.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Yuan Luo. Evaluating the state of the art in missing data imputation for clinical data. *Briefings in Bioinformatics*, 23(1):bbab489, 2022.

[10] Pius E Ojemolon, Valeria P Trelles-Garcia, Daniela Trelles-Garcia, Asim Kichloo, Sairam Raghavan, Abdulrahman I Abusalim, Precious Eseaton, and Precious O Eseaton. Racial disparities in outcomes of adults hospitalized for viral pneumonia. *Cureus*, 12(12), 2020.

[11] Vibhu Parcha, Rajat Kalra, Surya P Bhatt, Lorenzo Berra, Garima Arora, and Pankaj Arora. Trends and geographic variation in acute respiratory failure and ards mortality in the united states. *Chest*, 159(4):1460–1472, 2021.

[12] Aaron Saguil and Matthew V Fargo. Acute respiratory distress syndrome: diagnosis and management. *American family physician*, 85(4):352–358, 2012.

[13] Cleveland Clinic Staff. Acute respiratory distress syndrome (ards). https://my.clevelandclinic.org/health/diseases/15283-acute-respiratory-distress-syndrome-ards, 2020. Accessed: 2023-04-19.

[14] Johns Hopkins Staff. Pneumonia. https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia. Accessed: 2023-04-19.

[15] Mayo Clinic Staff. Ards- symptoms and causes. https://www.mayoclinic.org/diseases-conditions/ards/symptoms-causes/syc-20355576, 2022. Accessed: 2023-04-19.

[16] Sindhu Tipirneni and Chandan K Reddy. Self-supervised transformer for sparse and irregularly sampled multivariate clinical time-series. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(6):1–17, 2022.

[17] Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via emr-driven nonnegative restricted boltzmann machines (enrbm). *Journal of biomedical informatics*, 54:96–105, 2015.

A Appendix

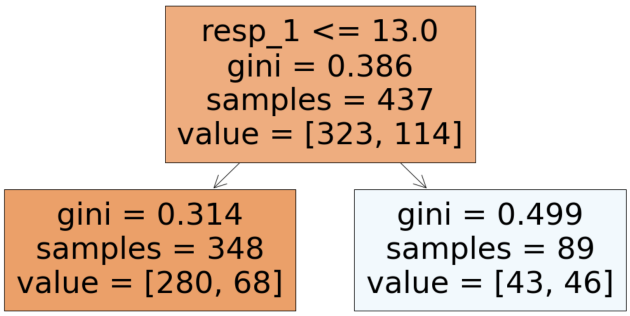

Figure 9: Summary of One-Depth Decision Tree

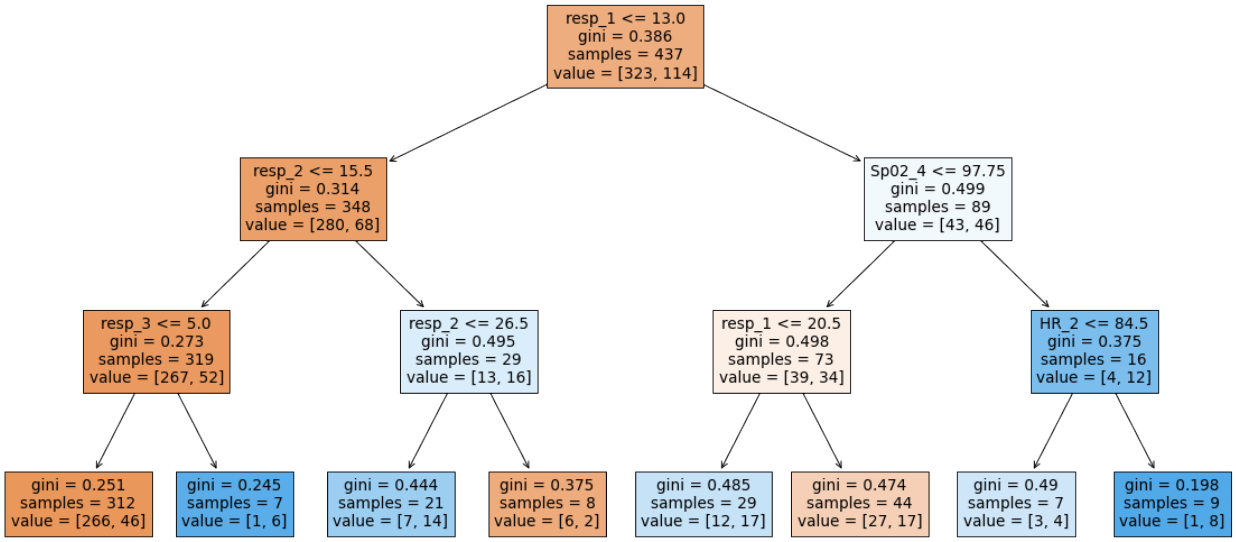

Figure 10: Summary of Tree Decisions

Max Depth	**Min Samples Per Leaf**	**Accuracy**
3	7	70.9%
2	9	69.1%
3	16	69.1%
4	9	69.1%
3	8	67.3%

Table 9: Accuracy of Tree Model by Pre-Pruning Parameters