Iowa Initiative for Artificial Intelligence
Final Report

Project title:	Machine Learning Model to Predict Extubation Success in Neonates		
Principal Investigator:	Lindsey Knake MD, MS, Danielle Rios MD, Patrick McNamara MD, James Blum MD		
Prepared by (IIAI):	Avinash Mudireddy		
Other investigators:			
Date:	11/01/2023		
Were specific aims fulfilled:		Yes	
Readiness for extramural proposal?		Yes	
If yes … Planned submission date			2 proposals submitted
Funding agency			2 proposals funded
Grant mechanism			
If no … Why not? What went wrong?			

<u>Brief summary of accomplished results:</u>

<u>Research report:</u>

Aims (provided by PI):

Background:

Mechanical ventilation in the Neonatal Intensive Care Unit (NICU) is a life-saving therapy. However, prolonged mechanical ventilation in preterm infants is associated with increased mortality, neurodevelopmental impairment, structural changes in the central nervous system, and bronchopulmonary dysplasia (BPD). Additionally, extubation failure is also associated with an increased risk of death, BPD and prolonged time on mechanical ventilation. Frequently, clinicians are weighing the risks between deciding to extubate too early and risk severe clinical decompensation that may result in brain or pulmonary hemorrhage versus waiting too long to extubate and risk lung damage or maldevelopment. Either decision may result in life-long morbidities. Thus, creating prediction models to help determine the appropriate timing for a trial of extubation in neonates is critically needed.

Previous models have attempted to predict extubation success in neonates without successful clinical adoption. Logistic regression and machine learning models have been created using static clinical and ventilator variables (with low resolution) to predict extubation success. However, there has been limited success with these models when evaluated at external institutions. The Heart Rate Characteristics index (HRCi) model has combined both static and dynamic variables in the model to predict extubation readiness. The HRCi model evaluates continuous heart rate wave form data and patient clinical characteristics but has not incorporated dynamic ventilator data into the model. We aim to incorporate high resolution ventilator data into our predictive model to enhance the accuracy and clinical utility of the model. We hypothesize that using this model to aid in the clinical decision of the best timing for a trial of extubation will result in more successful extubations with decreased morbidity in neonates.

Specific Aims:

1. To develop a predictive model for extubation success in neonates on mechanical ventilation with the primary outcome defined as the probability the patient will remain extubated for at least five days. The input variables would include static data such as laboratory data (blood gases) and patient factors (gestational age, birth weight, day of life, and weight at extubation) and high resolution dynamic clinical data (heart rate, mean airway pressure, ventilator measured lung compliance, oxygen trends, and patient desaturations).
2. To evaluate the algorithm developed in Aim 1 and for generalizability by determining performance characteristics based on hourly ventilator data that is extracted from the electronic health record (EHR).

3. To perform a subgroup analysis on different modes of mechanical ventilation or different post-extubation respiratory support to determine if these subgroups will have different predictors for extubation success.

Data for Aims:

We will plan to utilize Sickbay™ (Medical Informatics Corp. Houston, TX) for automated data capture and real-time data analysis which is already implemented into the University of Iowa NICU. This cluster-based platform interfaces with all patient-monitoring devices in the NICU and automatically records physiologic data without intervention or configuration by research or clinical personnel. It passively collects physiologic monitor data at exceptionally high resolution. These data include waveforms, vital signs, alarms, and alarm settings. Sickbay™ also provides research interfaces for self-service analytics across subject cohorts, which have been used in a variety of research projects analyzing pediatric physiologic data and has been used to support data recording and analysis in large-scale NIH-funded research projects.
Sickbay™ captures and stores both static and dynamic ventilator data parameters at 0.25 second intervals while the patient is on the ventilator and vital sign monitors. The implementation of Sickbay™ at the University of Iowa, has over 520 patients with continuous ventilator data collected and stored for research purposes. Additionally, once IRB approval is obtained, we plan to extract retrospective EHR data of the hourly documented ventilator settings from those 520 patients to test our hypothesis of Aim 2 that the prediction models will be vastly different when the resolution of the data is changed from seconds to hours.

AI/ML Approach:

<u>Input data:</u>
Clinical data- This data includes the patient(neonate) clinical parameters like weight, time, procedures, status, given medications, lab results.
Philips and capsule data- These datasets are timeseries of patient's vitals like ECG and ventilator readings.

The timeseries data is converted to the (Batch, #of seconds of history in the past, features)
The non-timeseries data is converted to (Batch, features)

Our research contains two types of models.

Timeseries-models:
1. 7.5 minute interval of GRU-based NN
2. Same model without Capsule data (no vent data)
3. Same model without gestational age
4. Same model without all clinical parameters to see if the clinical parameters make a big difference

Non-timeseries based models:

5. Logistic regression model with all variables – to do this we would need to transform each feature of our time series data into mean, std, min_val, max_val, last_valid_entry, slope, intercept
6. Random Forest

<u>Timeseries based Model design:</u>
The provided model is a deep learning architecture designed for predicting Extubation success or failure based on

Input Layers: The model takes three types of input data: Clinical data, Philips data (time series), and Capsule data time series).
Processing Philips and Capsule Data: Both Philips and Capsule data undergo a series of operations:
1. Convolutional Neural Network (CNN) layers: These are used for feature extraction from the time series data. The CNN layers help capture local patterns and dependencies in the input sequences.
2. Max-pooling: Max-pooling layers reduce the spatial dimensions of the CNN outputs while preserving the most important information.

3. GRU layers: Gated Recurrent Unit (GRU) layers are employed to capture temporal dependencies and patterns within the time series data. GRU is a type of recurrent neural network that is effective for sequence modeling.

Attention Mechanisms: After the GRU layers for Philips and Capsule data, multi-headed attention mechanisms are applied. These attention mechanisms are used to weight and focus on important temporal features within the data. This step enhances the model's ability to learn complex dependencies in the time series data.

Concatenation: The outputs from the attention mechanisms and the processed clinical data are concatenated together. This creates a fused representation of information from all three input sources.

Dense Layers: Additional dense layers with rectified linear unit (ReLU) activations and dropout are used for further feature extraction and non-linearity. These layers help the model learn complex relationships between the input features.

Output Layer: The model ends with a single neuron output layer with a sigmoid activation function. This configuration is appropriate for binary classification tasks, such as predicting extubation success or failure.

Rationale for Using GRU:

The use of GRU layers in the model is motivated by the nature of the input data, which includes time series information (Philips and Capsule data). GRU is a type of recurrent neural network that is well-suited for modeling sequences and capturing temporal dependencies.

Here's why GRU is a good choice:

Temporal Dependency: GRU units are designed to capture and model sequential data, making them suitable for time series analysis. They maintain an internal state that allows them to remember past observations and learn dependencies in the data.

Efficiency: GRUs are computationally efficient compared to other recurrent architectures like Long Short-Term Memory (LSTM) units, making them a good choice when dealing with large datasets.

Gradient Flow: GRUs are designed to mitigate the vanishing gradient problem, which can be an issue in deep networks, especially when dealing with long sequences.

In this model, the GRU layers are used to extract and model temporal patterns in the Philips and Capsule data, helping the model make predictions based on the historical information provided by these time series.

Experimental methods, validation approach:

The train: validation: test split is 60:20:20.

We used Adam with following configuration:
learning_rate=1e-4, decay= 1e-5
loss = binary_crossentropy

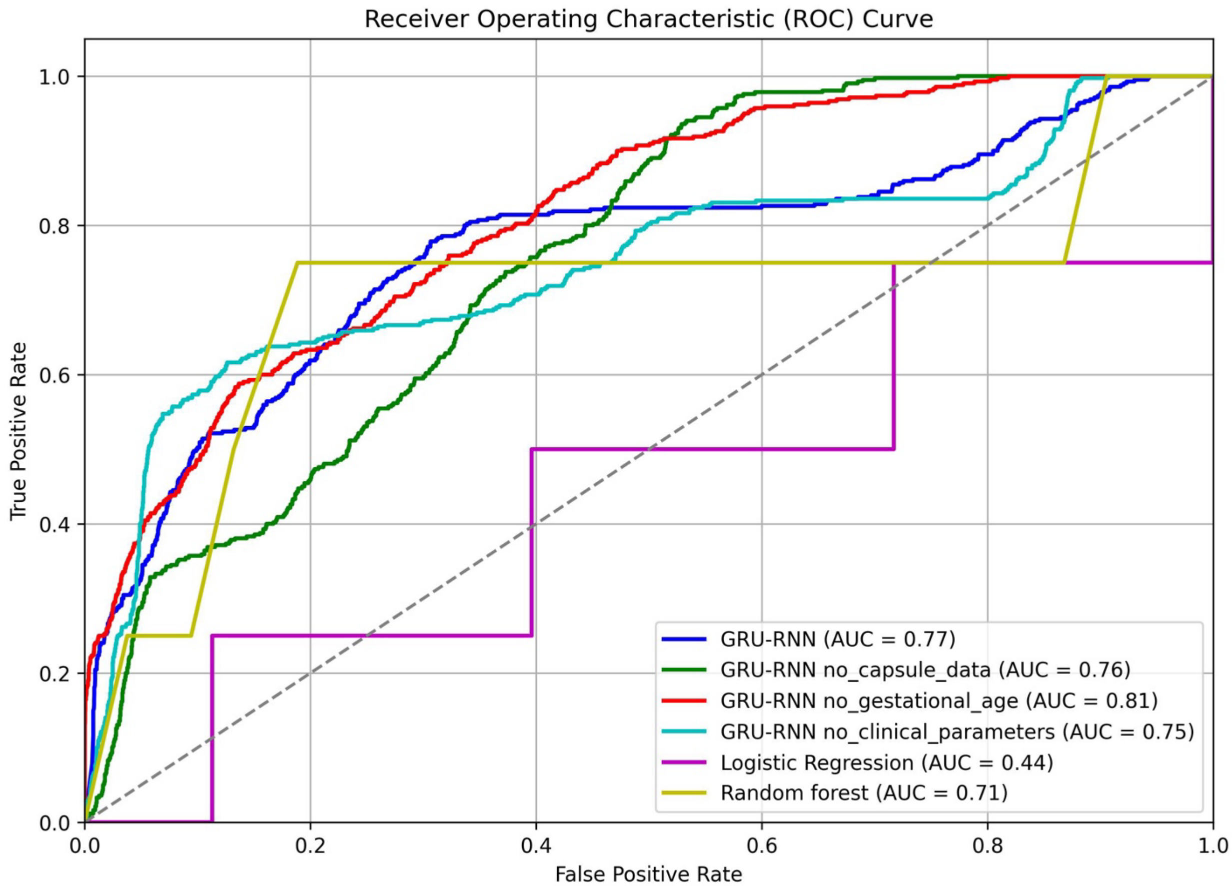
All the models are trained for 30 epochs with 512 as batch size.

Sample model architecture:

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_2 (InputLayer)           [(None, 450, 6)]     0           []

 conv1d (Conv1D)                (None, 449, 512)     6656        ['input_2[0][0]']

 conv1d_1 (Conv1D)              (None, 448, 1024)    1049600     ['conv1d[0][0]']

 max_pooling1d (MaxPooling1D)   (None, 44, 1024)     0           ['conv1d_1[0][0]']

 gru (GRU)                      (None, 44, 128)      443136      ['max_pooling1d[0][0]']

 layer_normalization (LayerNorm  (None, 44, 128)     256         ['gru[0][0]']
 alization)

 attention (Attention)          (None, 44, 128)      1           ['layer_normalization[0][0]',
                                                                  'layer_normalization[0][0]']

 input_1 (InputLayer)           [(None, 234)]        0           []

 flatten (Flatten)              (None, 5632)         0           ['attention[0][0]']

 dense_1 (Dense)                (None, 64)           15040       ['input_1[0][0]']

 dense (Dense)                  (None, 256)          1442048     ['flatten[0][0]']

 dropout_1 (Dropout)            (None, 64)           0           ['dense_1[0][0]']

 dropout (Dropout)              (None, 256)          0           ['dense[0][0]']

 concatenate (Concatenate)      (None, 320)          0           ['dropout_1[0][0]',
                                                                  'dropout[0][0]']

 dense_2 (Dense)                (None, 1)            321         ['concatenate[0][0]']

==================================================================================================
Total params: 2,957,058
Trainable params: 2,957,058
Non-trainable params: 0
```

All the other models have similar layers with their respective inputs.

Results:



Receiver Operating Characteristic (ROC) Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
GRU-RNN (AUC = 0.77)
GRU-RNN no_capsule_data (AUC = 0.76)
GRU-RNN no_gestational_age (AUC = 0.81)
GRU-RNN no_clinical_parameters (AUC = 0.75)
Logistic Regression (AUC = 0.44)
Random forest (AUC = 0.71)