A Data-driven Approach for Estimating Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 4, 2024

Abstract

Accurate placement of the tracheal tube is crucial in pediatric patients undergoing mechanical ventilation to avoid complications such as hypoxia and pneumothorax. However, determining the optimal tracheal tube depth (OTTD) remains challenging. Existing methods, such as chest X-ray and formula-based models, have limitations in accuracy and practicality. To address this gap, we developed a data-driven approach to estimate the OTTD in pediatric patients. Using a dataset of 969 patients aged 0-7 years who received post-operative mechanical ventilation, we trained machine learning models and developed formula-based models. Our results demonstrate the effectiveness of the data-driven approach in accurately estimating the OTTD, providing a valuable alternative to chest X-ray. The formula-based models also show promise, particularly the height formula-based model. However, further optimization and external validation are needed. By improving the accuracy of tracheal tube depth estimation, our findings have important implications for enhancing patient safety during mechanical ventilation in pediatric populations.

Results

In this study, our objective was to predict the Optimal Tracheal Tube Depth (OTTD) in pediatric patients undergoing mechanical ventilation using a data-driven approach. To determine the relationship between tube ID and OTTD stratified by sex, we conducted a descriptive analysis. The rationale behind exploring these variables separately for female and male patients is that pediatric patients have different anatomical characteristics, and understanding potential sex-based differences in the relationship between tube ID and OTTD could provide insights into optimal tube placement.

Table 1 presents the descriptive statistics of tube ID and OTTD for female and male patients. We observed that the mean tube ID was similar for both sexes, with females having a mean of 3.68 and males having a mean of 3.7. Regarding OTTD, we found a slight difference in mean depth, with males having a mean of 10.3 cm and females having a mean of 10.1 cm. While this difference is statistically significant, its clinical significance is less clear and requires further investigation.

Table 1: Descriptive statistics of Tube ID and OTTD stratified by sex

	Tube ID			OTTD		
	mean	min	max	mean	min	max
sex						
Female	3.68	2.5	5.5	10.1	6.6	17.7
Male	3.7	2.5	6	10.3	5.9	19.2

Tube ID: Internal diameter of the tube (mm)
OTTD: Optimal Tracheal Tube Depth as determined by chest X-ray (in cm)

To assess the accuracy of our predictive models, we performed model comparisons using different machine learning algorithms. The random forest, elastic net, support vector machine, and neural network models produced similar results, with residual sums of squares (RSS) ranging from 261 to 291 (Table 2). Although these models provided reasonable estimates of OTTD, the formula-based models exhibited comparable performance.

Table 2: Comparison of Residual Sums of Squares (RSS) of each model

	Residual Sum of Squares	p-value
Random Forest	291	0.566
Elastic Net	262	0.475
Support Vector Machine	261	0.545
Neural Network	265	0.62
height formula-based model	663	$<10^{-6}$
age formula-based model	347	0.549
id formula-based model	489	$<10^{-6}$

We further investigated the performance of formula-based models using height, age, and tube ID information. The height formula-based model yielded a higher RSS value of 663, indicating a larger deviation from the actual OTTD. The age formula-based model had an RSS value of 347, while the ID formula-based model performed better with an RSS value of 489. The higher RSS values in the formula-based models may be attributed to oversimplification in the model assumptions or unaccounted nonlinear relationships.

The machine learning models and the ID formula-based model show promise in accurately estimating the OTTD in pediatric patients. However, the clinical implications of these estimates require careful consideration. Further validation studies and refinements are necessary before incorporating these models into clinical practice.

In summary, our study demonstrates the potential of a data-driven approach and formula-based models in estimating the Optimal Tracheal Tube Depth in pediatric patients undergoing mechanical ventilation. While both machine learning models and formula-based models show promise, further research is needed to improve their accuracy and evaluate their clinical applicability. Implementing these approaches can contribute to enhancing patient safety during mechanical ventilation in pediatric populations, but careful consideration of challenges in their implementation is necessary before widespread use can be recommended.

A Data Description

Here is the data description, as provided by the user:

Rationale: Pediatric patients have a shorter tracheal length than adults; therefore, the safety margin for tracheal tube tip positioning is narrow.
Indeed, the tracheal tube tip is misplaced in 35%{50% of pediatric patients and can cause hypoxia, atelectasis, hypercarbia, pneumothorax, and even death.
Therefore, in pediatric patients who require mechanical ventilation, it is crucial to determine the Optimal Tracheal Tube Depth (defined here as `OTTD`, not an official term).

Note: For brevity, we introduce the term `OTTD` to refer to the "optimal tracheal tube depth". This is not an official term that can be found in the literature.

Existing methods: The gold standard to determine OTTD is by chest X-ray, which is time-consuming and requires radiation exposure.
Alternatively, formula-based models on patient features such as age and height are used to determine OTTD, but with limited success.

The provided dataset focus on patients aged 0-7 year old who received post-operative mechanical ventilation after undergoing surgery at Samsung Medical Center between January 2015 and December 2018.
For each of these patients, the dataset provides the OTTD determined by chest X-ray as well as features extracted from patient electronic health records.

1 data file:

"tracheal_tube_insertion.csv"
The csv file is a clean dataset of 969 rows (patients) and 6 columns:

Tube:
#1 `tube` - "tube ID", internal diameter of the tube (mm) [Included only for the formula-based model; Do not use as a machine-learning model feature]

Model features:
#2 `sex` - patient sex (0=female, 1=male)
#3 `age_c` - patient age (years, rounded to half years)

```
#4 `ht` - patient height (cm)
#5 `wt` - patient weight (kg)

Target:
#6 `tube_depth_G` - Optimal tracheal tube depth as determined by chest X-ray (in
    cm)
```

B Data Exploration

B.1 Code

The Data Exploration was carried out using the following custom code:

```python

import pandas as pd

# Load the dataset
df = pd.read_csv("tracheal_tube_insertion.csv")

# Open a file for writing
with open('data_exploration.txt', 'w') as file:

    # Data size
    file.write('# Data Size\n')
    file.write(f'Number of rows: {df.shape[0]}\n')
    file.write(f'Number of columns: {df.shape[1]}\n')
    file.write('\n')

    # Summary Statistics
    file.write('# Summary Statistics\n')
    file.write(df.describe().to_string())
    file.write('\n')

    # Categorical Variables
    file.write('\n# Categorical Variables\n')
    for col in df.select_dtypes(include=['object']).columns:
        file.write(f'{col} most common values:\n')
        file.write(df[col].value_counts().head(5).to_string())
        file.write('\n')

    # Missing Values
    file.write('\n# Missing Values\n')
    file.write(df.isnull().sum().to_string())

```

B.2 Code Description

The provided code performs data exploration on the dataset ”tracheal_tube_insertion.csv”. It conducts several analysis steps to gain insights into the dataset.

1. Data Size: The code first calculates the number of rows and columns in the dataset using the shape attribute of the DataFrame. It then writes this information into the ”data_exploration.txt” file.

2. Summary Statistics: The code calculates the summary statistics of the numerical variables in the dataset using the describe() function. This includes count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum values for each feature. The summary statistics are then written into the ”data_exploration.txt” file.

3. Categorical Variables: The code identifies the categorical variables in the dataset using the select_dtypes() function with the ’object’ parameter. It then determines the most common values for each categorical variable using the value_counts() function, and selects the top 5 values. These most common values are written into the ”data_exploration.txt” file.

4. Missing Values: The code checks for missing values in the dataset using the isnull() function, which returns a DataFrame of the same shape with True values for missing values and False values otherwise. It then calculates the sum of missing values for each feature using the sum() function. The number of missing values for each feature is written into the ”data_exploration.txt” file.

The ”data_exploration.txt” file serves as a record of the data exploration analysis, providing information on data size, summary statistics, most common categorical variable values, and missing values. These insights are crucial for understanding the characteristics and quality of the dataset, which can guide subsequent data preprocessing and modeling steps.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of rows: 969
Number of columns: 6

# Summary Statistics
        tube    sex  age_c    ht    wt  tube_depth_G
count    969    969    969   969   969           969
mean   3.694 0.5387  0.758    66 7.127         10.19
std   0.5681 0.4988   1.44 19.08 4.774         1.766
min      2.5      0      0  29.5  0.57           5.9
25%      3.5      0      0    52  3.56             9
50%      3.5      1      0  61.5   5.7           9.8
75%        4      1      1    76   9.5          11.2
max        6      1      7 135.4    31          19.2

# Categorical Variables

# Missing Values
tube            0
sex             0
age_c           0
ht              0
wt              0
tube_depth_G    0
```

C Data Analysis

C.1 Code

The Data Analysis was carried out using the following custom code:

```

# IMPORT
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

```python
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_rel
import pickle
import warnings

# To ignore warning related to more iteration needed for
↪  convergence in MLPRegressor
warnings.filterwarnings('ignore')

# LOAD DATA
df = pd.read_csv('tracheal_tube_insertion.csv')

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of height and tube_depth_G
↪  stratified by sex"
df0 = df.groupby('sex').agg({'tube': ['mean', 'min', 'max'],
↪  'tube_depth_G': ['mean', 'min', 'max']})
df0.index = df0.index.map({0:'female', 1:'male'})
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# Creating dummies for 'sex' column
df = pd.get_dummies(df, columns=['sex'], drop_first=True)

# Splitting data into train and test datasets
X_train, X_test, y_train, y_test =
↪  train_test_split(df.drop('tube_depth_G', axis=1),
↪  df['tube_depth_G'], test_size=0.2, random_state=42)

# ANALYSIS
models = [RandomForestRegressor(), ElasticNet(), SVR(),
↪  MLPRegressor(max_iter=1000)]
model_names = ["Random Forest", "Elastic Net", "Support Vector
↪  Machine", "Neural Network"]
```

```

residuals = []
names = []

for model, name in zip(models, model_names):
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    residuals.append(preds - y_test)
    names.append(name)

# Adding the tube_depth_G column to the test set
X_test['tube_depth_G'] = y_test

# Formula-based Model Calculations
# Calculating for test set only
X_test['height_formula'] = X_test['ht'] / 10 + 5
X_test['age_formula'] = X_test['age_c'].apply(lambda x: 9 if x
↪  < 0.5 else 10 if x < 1 else 11 if x < 2 else 12 + 0.5 * x)
X_test['id_formula'] = 3 * X_test['tube']

for formula in ['height_formula', 'age_formula',
↪  'id_formula']:
    residuals.append(X_test[formula] - X_test['tube_depth_G'])
    names.append(formula.replace('_formula', ' formula-based
    ↪  model'))

# Table 1: "Comparison of Residual sum of squares (RSS) of
↪  each model"
residual_sum_squares = [np.sum(np.square(res)) for res in
↪  residuals]
p_values = [ttest_rel(res, np.zeros_like(res)).pvalue for res
↪  in residuals]

df1 = pd.DataFrame({'Model': names, 'Residual Sum of Squares':
↪  residual_sum_squares, 'p-value': p_values})
df1.set_index('Model', inplace=True)
df1.index.name = None
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
```

```python
additional_results = {
     'Total number of observations': df.shape[0],
     'Total number of test observations': X_test.shape[0]
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)

```

C.2 Code Description

The code performs data analysis on a dataset of pediatric patients who received post-operative mechanical ventilation. The goal is to determine the optimal tracheal tube depth (OTTD) for these patients using various models and compare their performance.

The dataset is loaded from the ”tracheal_tube_insertion.csv” file, which contains information such as patient age, height, weight, and the chest X-ray determined OTTD.

The code starts by calculating descriptive statistics of height and OTTD stratified by patient sex and stores the results in ”table_0.pkl” file.

Next, the dataset is preprocessed by creating dummy variables for the ’sex’ column to be used in the analysis. The data is then split into train and test datasets for model evaluation.

Four different models are trained on the train dataset, namely Random Forest Regression, Elastic Net Regression, Support Vector Machine Regression, and MLP Regression. The trained models are then used to make predictions on the test dataset. The residuals (the differences between the predicted values and the actual OTTD) are calculated for each model.

Additionally, formula-based models are constructed using patient features like height, age, and tube ID. These formula-based models are evaluated on the test dataset by calculating the residuals.

A comparison of the residual sum of squares (RSS) between the different models and the formula-based models is performed, and the results are stored in ”table_1.pkl”. The p-values for each model’s residuals are also calculated.

Finally, the code saves additional results including the total number of observations in the dataset and the total number of observations in the test dataset in the ”additional_results.pkl” file.

Overall, the code performs data analysis by training models, evaluating their performance, and comparing them to formula-based models to determine the optimal tracheal tube depth for pediatric patients undergoing mechanical ventilation.

C.3 Code Output

table_0.pkl

```
            tube             tube_depth_G
            mean  min  max           mean  min   max
sex
female  3.681208  2.5  5.5      10.062416  6.6  17.7
male    3.704598  2.5  6.0      10.298276  5.9  19.2
```

table_1.pkl

```
                            Residual Sum of Squares    p-value
Random Forest                            290.823859     0.5663
Elastic Net                              261.610039     0.4752
Support Vector Machine                   260.990405     0.5446
Neural Network                           265.189515     0.6198
height formula-based model               663.264600  4.045e-34
age formula-based model                  347.300000     0.5492
id formula-based model                   489.150000  1.915e-17
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Total number of test observations': 194,
}
```

D LaTeX Table Design

D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```

# IMPORT
import pandas as pd
from my_utils import to_latex_with_note, format_p_value,
 ↪  is_str_in_df, split_mapping
from typing import Dict, Any, Optional, Tuple
```

```

# PREPARATION FOR ALL TABLES
shared_mapping = {
    'tube': ('Tube ID', 'Internal diameter of the tube (mm)'),
    'age_c': ('Age', 'Patient age (years, rounded to half
    ↪   years)'),
    'ht': ('Height', 'Patient height (cm)'),
    'wt': ('Weight', 'Patient weight (kg)'),
    'tube_depth_G': ('OTTD', 'Optimal Tracheal Tube Depth as
    ↪   determined by chest X-ray (in cm)'),
}

# TABLE 0:
df0 = pd.read_pickle('table_0.pkl')

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
↪   is_str_in_df(df0, k)}

# Here, we specify the labels for male and female as given in
↪   the description of the dataset
mapping.setdefault('female', ('Female', None))
mapping.setdefault('male', ('Male', None))
abbrs_to_names, legend = split_mapping(mapping)
df0.rename(columns=abbrs_to_names, index=abbrs_to_names,
↪   inplace=True)

# Save as latex:
to_latex_with_note(
 df0, 'table_0.tex',
 caption="Descriptive statistics of Tube ID and OTTD
 ↪  stratified by sex",
 label='table:descriptive_statistics',
 note=None,
 legend=legend)

# TABLE 1:
df1 = pd.read_pickle('table_1.pkl')

# FORMAT VALUES
```

```
df1['p-value'] = df1['p-value'].apply(format_p_value)

# RENAME ROWS
mapping = {k: v for k, v in shared_mapping.items() if
↪  is_str_in_df(df1, k)}
abbrs_to_names, legend = split_mapping(mapping)
df1.rename(index=abbrs_to_names, inplace=True)

# Save as latex:
to_latex_with_note(
 df1, 'table_1.tex',
 caption="Comparison of Residual Sums of Squares (RSS) of each
 ↪  model",
 label='table:model_comparison',
 note=None,
 legend=legend)

```

D.2 Provided Code

The code above is using the following provided functions:

```
def to_latex_with_note(df, filename: str, caption: str, label:
↪  str, note: str = None, legend: Dict[str, str] = None,
↪  **kwargs):
 """
 Converts a DataFrame to a LaTeX table with optional note and
↪   legend added below the table.

 Parameters:
 - df, filename, caption, label: as in `df.to_latex`.
 - note (optional): Additional note below the table.
 - legend (optional): Dictionary mapping abbreviations to full
↪   names.
 - **kwargs: Additional arguments for `df.to_latex`.

 Returns:
 - None: Outputs LaTeX file.
 """

```

```
def format_p_value(x):
 returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"

def is_str_in_df(df: pd.DataFrame, s: str):
 return any(s in level for level in getattr(df.index,
  ↪  'levels', [df.index]) + getattr(df.columns, 'levels',
  ↪  [df.columns]))

AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]

def split_mapping(abbrs_to_names_and_definitions:
 ↪  AbbrToNameDef):
 abbrs_to_names = {abbr: name for abbr, (name, definition) in
  ↪  abbrs_to_names_and_definitions.items() if name is not
  ↪  None}
 names_to_definitions = {name or abbr: definition for abbr,
  ↪  (name, definition) in
  ↪  abbrs_to_names_and_definitions.items() if definition is
  ↪  not None}
 return abbrs_to_names, names_to_definitions

```

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of Tube ID and OTTD stratified by sex}
\label{table:descriptive_statistics}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrrr}
\toprule
 & \multicolumn{3}{r}{Tube ID} & \multicolumn{3}{r}{OTTD} \\
 & mean & min & max & mean & min & max \\
sex &  &  &  &  &  &  \\
\midrule
\textbf{Female} & 3.68 & 2.5 & 5.5 & 10.1 & 6.6 & 17.7 \\
\textbf{Male} & 3.7 & 2.5 & 6 & 10.3 & 5.9 & 19.2 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Tube ID}: Internal diameter of the tube (mm)
\item \textbf{OTTD}: Optimal Tracheal Tube Depth as determined by chest X-ray
    (in cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Comparison of Residual Sums of Squares (RSS) of each model}
\label{table:model_comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Residual Sum of Squares & p-value \\
\midrule
\textbf{Random Forest} & 291 & 0.566 \\
\textbf{Elastic Net} & 262 & 0.475 \\
\textbf{Support Vector Machine} & 261 & 0.545 \\
\textbf{Neural Network} & 265 & 0.62 \\
\textbf{height formula-based model} & 663 & $<$1e-06 \\
\textbf{age formula-based model} & 347 & 0.549 \\
\textbf{id formula-based model} & 489 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```