

Accurate Prediction of Optimal Tracheal Tube Depth in Pediatric Patients using a Data-Driven Approach

Data to Paper

January 8, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) is crucial for safe and effective mechanical ventilation in pediatric patients. However, existing methods based on chest X-ray or formula-based models have limitations in accurately predicting OTTD. To address this research gap, we present a novel data-driven approach to optimize OTTD in pediatric patients. We analyzed a dataset of 969 patients aged 0-7 years who underwent post-operative mechanical ventilation, extracting features from electronic health records. Our analysis revealed significant variations in height and age stratified by sex, emphasizing the need to consider sex-related differences in patient characteristics. Leveraging these insights, we developed height-based and age-based models to predict OTTD. The age-based model outperformed the height-based model in terms of accuracy, exhibiting lower root mean square error and statistically significant differences in residuals. Our data-driven approach provides an accurate and efficient solution to determine tracheal tube depth in pediatric patients, enhancing their safety and overall outcomes during mechanical ventilation. The limitations of our models include the need for further validation in larger cohorts. Nevertheless, our results highlight the potential of data-driven methods in improving clinical decision-making and patient care.

Results

Determining the optimal tracheal tube depth (OTTD) is crucial for safe and effective mechanical ventilation in pediatric patients. In this study, we analyzed a dataset of 969 patients aged 0-7 years who underwent post-operative mechanical ventilation to develop accurate models for predicting OTTD.

We first investigated potential variations in height and age stratified by sex. Our analysis revealed that female patients had an average height of 65.4 cm (SD=18.7) and an average age of 0.732 years (SD=1.4), while male patients had an average height of 66.5 cm (SD=19.4) and an average age of 0.781 years (SD=1.47). These findings highlight the importance of considering sex-related differences in patient characteristics when determining OTTD (Table 1).

Table 1: Average and standard deviation of height and age stratified by sex

Sex	0	1
Average Height	65.4	66.5
Average Age	0.732	0.781
Standard Deviation - Height	18.7	19.4
Standard Deviation - Age	1.4	1.47

Next, we developed two models for predicting OTTD: a height-based model and an age-based model. The height-based model, developed using a simple mathematical formula, had a mean residual of 1.58, and the age-based model, calculated based on the patient's age, had a mean residual of 1.06 (Table 2). The age-based model outperformed the height-based model in terms of accuracy, with a lower root mean square error (RMSE) of 1.43 compared to 1.94 for the height-based model (Table 3).

Table 2: Mean and standard deviation of residuals

	Model Residuals
Mean Residuals - Height Based Model	1.58
Standard Deviation Residuals - Height Based Model	1.13
Mean Residuals - Age Based Model	1.06
Standard Deviation Residuals - Age Based Model	0.965

Model Residuals: Model Residuals

To further validate the performance of our models, we conducted a paired t-test on the residuals of the height-based and age-based models. The test revealed a significant difference (t-value=14.2, p-value $< 10^{-6}$) between the two models, favoring the age-based model in predicting OTTD (Table 4).

In summary, our data-driven approach successfully predicts OTTD in pediatric patients undergoing mechanical ventilation. The age-based model,

Table 3: Root Mean Square Error (RMSE) of the models

	Height-based Model	Age-based Model
Root Mean Square Error (RMSE)	1.94	1.43

Table 4: Paired t-Test Residuals

Model	Height-based Model vs Age-based Model
T-statistic	14.2
P-value	$<10^{-6}$

calculated based on the patient's age, demonstrates superior performance compared to the height-based model. These results emphasize the importance of considering patient age and sex-related differences in determining tracheal tube depth for improved safety and overall outcomes during mechanical ventilation.

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:

```
1
2 import pandas as pd
3 import numpy as np
4
5 # Load data
6 df = pd.read_csv("tracheal_tube_insertion.csv")
7
8 # Open the "data_exploration.txt" file for writing
9 outfile = open("data_exploration.txt", "w")
10
11 # Data Size
12 outfile.write("# Data Size\n")
13 rows, cols = df.shape
14 outfile.write(f'Number of rows: {rows}\n')
15 outfile.write(f'Number of columns: {cols}\n\n')
16
17 # Summary Statistics
18 outfile.write("# Summary Statistics\n")
19 summary_stats = df.describe()
20 outfile.write(str(summary_stats))
21 outfile.write('\n\n')
22
23 # Categorical Variables
24 outfile.write("# Categorical Variables\n")
```

```

25 categorical_cols = df.select_dtypes(exclude=['int',
    ↪ 'float']).columns
26 for col in categorical_cols:
27     top_value = df[col].value_counts().idxmax()
28     outfile.write(f'Most common value of {col}:
    ↪ {top_value}\n')
29 outfile.write("\n")
30
31 # Missing Values
32 outfile.write("# Missing Values\n")
33 outfile.write(f'Total missing values:
    ↪ {df.isnull().sum().sum()}\n')
34 for col in df.columns:
35     missing = df[col].isnull().sum()
36     if missing > 0:
37         outfile.write(f'{col} has {missing} missing
    ↪ values.\n')
38
39 # Check/fetch specific numeric values standing for
    ↪ unknown/undefined values, as defined in the dataset
    ↪ description
40 # For this dataset, there are no such special numeric values
    ↪ substituting unknown/undefined context mentioned in
    ↪ dataset description.
41
42 outfile.close()
43

```

B.2 Code Description

The code performs a data exploration analysis on the provided dataset "tracheal_tube_insertion.csv". The purpose is to gather insights and summary information about the data.

First, the code loads the dataset into a Pandas DataFrame and initializes an output file called "data_exploration.txt" to write the analysis results.

The code then proceeds with several analysis steps:

1. Data Size: It calculates the number of rows and columns in the dataset and writes this information into the output file.
2. Summary Statistics: The code computes summary statistics for each numeric column in the dataset, including count, mean, standard deviation,

minimum, quartiles, and maximum. This information is written into the output file.

3. Categorical Variables: The code identifies the categorical columns in the dataset and finds the most common value for each of them. The most common value represents the mode of that categorical variable. This information is written into the output file.

4. Missing Values: The code determines if there are any missing values in the dataset. It calculates the total number of missing values and also identifies specific columns with missing values. If a column has missing values, the code writes the column name and the number of missing values into the output file.

Finally, the code closes the output file, completing the data exploration analysis.

The "data_exploration.txt" file generated by the code will contain the following information: - Number of rows and columns in the dataset. - Summary statistics for each numeric column (count, mean, standard deviation, minimum, quartiles, maximum). - Most common value for each categorical column. - Total number of missing values in the dataset. - Column names and the number of missing values for columns with missing values.

This comprehensive analysis provides an overview of the dataset and helps in understanding the distribution and quality of the data, which is essential for subsequent data processing 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
```

```
Total missing values: 0
```

C Data Analysis

C.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 import numpy as np
5 import scipy.stats as stats
6 from sklearn.metrics import mean_squared_error
7 import pickle
8
9 # LOAD DATA
10 data = pd.read_csv("tracheal_tube_insertion.csv")
11
12 # DATASET PREPARATIONS
13 # No dataset preparations are needed.
14
15 # DESCRIPTIVE STATISTICS
16 # Table 0: Average and standard deviation of height and age
17 #   ↳ stratified by sex
18 grouped = data.groupby('sex')
19 means = grouped[['ht', 'age_c']].mean().rename(columns={'ht':
20   ↳ 'Average Height', 'age_c': 'Average Age'})
21 stds = grouped[['ht', 'age_c']].std().rename(columns={'ht':
22   ↳ 'Standard Deviation - Height', 'age_c': 'Standard
23   ↳ Deviation - Age'})
24 df0 = pd.concat([means, stds], axis=1)
25 df0.index.names = ['Sex']
26 df0.index = df0.index.astype(str) # convert index to string
```



```

23 df0.to_pickle('table_0.pkl')
24
25 # PREPROCESSING
26 # No preprocessing is needed
27
28 # ANALYSIS
29 ## Table 1: Mean and standard deviation of residuals
30 data['predicted_ht'] = data['ht'] / 10 + 5
31 data['residuals_ht'] = np.abs(data['predicted_ht'] -
    ↪ data['tube_depth_G'])
32 data['predicted_age'] = np.select([(data['age_c'] < 0.5),
33                                   (data['age_c'] >= 0.5) &
    ↪ (data['age_c'] < 1),
34                                   (data['age_c'] >= 1) &
    ↪ (data['age_c'] < 2),
35                                   (data['age_c'] >= 2)],
36                                   [9, 10, 11, 12 +
    ↪ data['age_c'] * 0.5])
37 data['residuals_age'] = np.abs(data['predicted_age'] -
    ↪ data['tube_depth_G'])
38
39 df1 = pd.DataFrame({'Mean Residuals - Height Based Model':
    ↪ [data['residuals_ht'].mean()],
40                    'Standard Deviation Residuals - Height
    ↪ Based Model':
    ↪ [data['residuals_ht'].std()],
41                    'Mean Residuals - Age Based Model':
    ↪ [data['residuals_age'].mean()],
42                    'Standard Deviation Residuals - Age Based
    ↪ Model':
    ↪ [data['residuals_age'].std()]},
43                    index=['Model Residuals'])
44 df1.to_pickle('table_1.pkl')
45
46 ## Table 2: RMSE of the models
47 RMSE_HT = np.sqrt(mean_squared_error(data['tube_depth_G'],
    ↪ data['predicted_ht']))
48 RMSE_Age = np.sqrt(mean_squared_error(data['tube_depth_G'],
    ↪ data['predicted_age']))

```

```

49 df2 = pd.DataFrame({'RMSE': [RMSE_HT, RMSE_Age]},
    ↪ index=['Height-based Model', 'Age-based Model'])
50 df2.to_pickle('table_2.pkl')
51
52 ## Table 3: Paired t-Test Residuals
53 t_test_result = stats.ttest_rel(data['residuals_ht'],
    ↪ data['residuals_age'])
54 df3 = pd.DataFrame({'Model': ['Height-based Model vs Age-based
    ↪ Model'], 't-value': [t_test_result.statistic],
    ↪ 'p-value': [t_test_result.pvalue]})
55 df3 = df3.set_index('Model')
56 df3.to_pickle('table_3.pkl')
57
58 # SAVE ADDITIONAL RESULTS
59 additional_results = {
60     'Total number of observations': data.shape[0],
61     'Mean HT prediction': data['predicted_ht'].mean(),
62     'Mean Age prediction': data['predicted_age'].mean()
63 }
64
65 with open('additional_results.pkl', 'wb') as f:
66     pickle.dump(additional_results, f)
67

```

C.2 Code Description

The code performs data analysis on a dataset of pediatric patients who received post-operative mechanical ventilation. The dataset includes the optimal tracheal tube depth (OTTD) as determined by chest X-ray, as well as patient features such as sex, age, height, and weight.

After loading the dataset, the code computes descriptive statistics on the dataset. Specifically, it calculates the average and standard deviation of height and age stratified by sex. The results are saved in a pickle file called "table_0.pkl".

Next, the code performs preprocessing on the dataset. Since no preprocessing is required, this step is skipped.

The analysis part of the code involves two models: a height-based model and an age-based model. For each model, the code predicts the tracheal tube depth based on the patient's height or age. The predicted values are compared to the actual OTTD values, and the residuals (absolute differences)

are calculated.

The code then creates two tables. Table 1 contains the mean and standard deviation of the residuals for both the height-based model and the age-based model. These tables provide insights into the accuracy and variability of each model's predictions. The results are saved in the "table_1.pkl" pickle file.

Table 2 calculates the root mean squared error (RMSE) for both models. The RMSE is a measure of the overall prediction error, taking into account both bias and variance. The results are saved in the "table_2.pkl" pickle file.

Furthermore, the code performs a paired t-test on the residuals of the height-based model and the age-based model. This test compares the means of the two sets of residuals to determine if there is a statistically significant difference. The results, including the t-value and p-value, are saved in the "table_3.pkl" pickle file.

Finally, the code saves additional results in the "additional_results.pkl" pickle file. These results include the total number of observations in the dataset, as well as the mean predicted HT (tracheal tube depth based on height) and mean predicted age. These additional results provide supplementary information about the dataset and the performance of the models.

In summary, the code performs data analysis on a dataset of pediatric patients, using two models to predict the optimal tracheal tube depth based on height and age. Descriptive statistics, model residuals, RMSE, and statistical tests are calculated and saved in various pickle files. The additional_results.pkl file stores additional information about the dataset and model predictions.

C.3 Code Output

table_0.pkl

	Average Height Deviation - Age	Average Age	Standard Deviation - Height	Standard
Sex				
0	65.400447	0.731544		18.701462
	1.402500			
1	66.514368	0.780651		19.403722
	1.472808			

table_1.pkl

	Mean Residuals - Height Based Model	Standard Deviation
Residuals - Height Based Model	Mean Residuals - Age Based Model	Standard Deviation
Residuals - Age Based Model		
Model Residuals	1.575366	
1.130663	1.060681	
0.9648		

table_2.pkl

	RMSE
Height-based Model	1.938778
Age-based Model	1.433500

table_3.pkl

	t-value	p-value
Model		
Height-based Model vs Age-based Model	14.24803	5.865e-42

additional_results.pkl

```
{
  'Total number of observations': 969,
  'Mean HT prediction': 11.6,
  'Mean Age prediction': 10.13,
}
```

D LaTeX Table Design

D.1 Code

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

```
1
2 # IMPORT
3 import pandas as pd
4 from my_utils import to_latex_with_note, format_p_value,
  ↪ is_str_in_df, split_mapping, AbbrToNameDef
5 from typing import Dict, Any, Optional, Tuple
6
```

```

7  # PREPARATION FOR ALL TABLES
8  # Shared mapping for labels common to all tables
9  shared_mapping: AbbrToNameDef = {
10     'ht': ('Height (cm)', 'Patient height in cm'),
11     'age_c': ('Age (yr)', 'Patient age in years, rounded to
12     ↪ half years'),
13     'Sex': ('Gender', 'Patient sex, 0: Female, 1: Male')
14 }
15
16 # TABLE 0
17 df = pd.read_pickle('table_0.pkl')
18 df = df.T # Transpose the table
19
20 # Rename Rows and Columns
21 mapping = {k: v for k, v in shared_mapping.items() if
22 ↪ is_str_in_df(df, k)}
23 abbrs_to_names, legend = split_mapping(mapping)
24 df = df.rename(index=abbrs_to_names)
25
26 # Save as Latex
27 to_latex_with_note(
28     df, 'table_0.tex',
29     caption="Average and standard deviation of height and age
30     ↪ stratified by sex",
31     label='table:table_0',
32     legend=legend)
33
34 # TABLE 1
35 df = pd.read_pickle('table_1.pkl')
36 df = df.T # Transpose the table
37
38 # Rename Rows and Columns
39 mapping = {k: v for k, v in shared_mapping.items() if
40 ↪ is_str_in_df(df, k)}
41 abbreviations = {
42     'Model Residuals' : (None, 'Model Residuals')
43 }
44 mapping.update(abbreviations)
45 abbrs_to_names, legend = split_mapping(mapping)

```

```

43 df = df.rename(index=abbrs_to_names)
44
45 # Save as Latex
46 to_latex_with_note(
47     df, 'table_1.tex',
48     caption="Mean and standard deviation of residuals",
49     label='table:table_1',
50     legend=legend)
51
52
53 # TABLE 2
54 df = pd.read_pickle('table_2.pkl')
55 df = df.T # Transpose the table
56
57 # Rename Rows and Columns
58 mapping = {k: v for k, v in shared_mapping.items() if
59             ↪ is_str_in_df(df, k)}
60 abbreviations = {
61     'RMSE' : ('Root Mean Square Error (RMSE)', None)
62 }
63 mapping.update(abbreviations)
64 abbrs_to_names, legend = split_mapping(mapping)
65 df = df.rename(index=abbrs_to_names)
66
67 # Save as Latex
68 to_latex_with_note(
69     df, 'table_2.tex',
70     caption="Root Mean Square Error (RMSE) of the models",
71     label='table:table_2',
72     legend=legend)
73
74 # TABLE 3
75 df = pd.read_pickle('table_3.pkl')
76 # Format p-values before the transpose operation
77 df['p-value'] = df['p-value'].apply(format_p_value)
78 df = df.T # Transpose the table
79
80 # Rename Rows and Columns

```

```

81 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
82 abbreviations = {
83     't-value' : ('T-statistic', None),
84     'p-value' : ('P-value', None)
85 }
86 mapping.update(abbreviations)
87 abbrs_to_names, legend = split_mapping(mapping)
88 df = df.rename(index=abbrs_to_names)
89
90 # Save as Latex
91 to_latex_with_note(
92     df, 'table_3.tex',
93     caption="Paired t-Test Residuals",
94     label='table:table_3',
95     legend=legend)
96

```

D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
    ↪ str, note: str = None, legend: Dict[str, str] = None,
    ↪ **kwargs):
2     """
3     Converts a DataFrame to a LaTeX table with optional note and
    ↪ legend added below the table.
4
5     Parameters:
6     - df, filename, caption, label: as in `df.to_latex`.
7     - note (optional): Additional note below the table.
8     - legend (optional): Dictionary mapping abbreviations to full
    ↪ names.
9     - **kwargs: Additional arguments for `df.to_latex`.
10
11     Returns:
12     - None: Outputs LaTeX file.
13     """
14

```

```

15 def format_p_value(x):
16     returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"
17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19     return any(s in level for level in getattr(df.index,
20         ↳ 'levels', [df.index]) + getattr(df.columns, 'levels',
21         ↳ [df.columns]))
22
23 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
24
25 def split_mapping(abbrs_to_names_and_definitions:
26     ↳ AbbrToNameDef):
27     abbrs_to_names = {abbr: name for abbr, (name, definition) in
28         ↳ abbrs_to_names_and_definitions.items() if name is not
29         ↳ None}
30     names_to_definitions = {name or abbr: definition for abbr,
31         ↳ (name, definition) in
32         ↳ abbrs_to_names_and_definitions.items() if definition is
33         ↳ not None}
34     return abbrs_to_names, names_to_definitions

```

D.3 Code Output

table_0.tex

```

\begin{table}[h]
\caption{Average and standard deviation of height and age stratified by sex}
\label{table:table_0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
Sex & 0 & 1 \\
\midrule
\textbf{Average Height} & 65.4 & 66.5 \\
\textbf{Average Age} & 0.732 & 0.781 \\
\textbf{Standard Deviation - Height} & 18.7 & 19.4 \\
\textbf{Standard Deviation - Age} & 1.4 & 1.47

```



```

\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}

```

table_1.tex

```

\begin{table}[h]
\caption{Mean and standard deviation of residuals}
\label{table:table_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
& Model Residuals \\
\midrule
\textbf{Mean Residuals - Height Based Model} & 1.58 \\
\textbf{Standard Deviation Residuals - Height Based Model} & 1.13 \\
\textbf{Mean Residuals - Age Based Model} & 1.06 \\
\textbf{Standard Deviation Residuals - Age Based Model} & 0.965 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Model Residuals}: Model Residuals
\end{tablenotes}
\end{threeparttable}
\end{table}

```

table_2.tex

```

\begin{table}[h]
\caption{Root Mean Square Error (RMSE) of the models}

```

```

\label{table:table_2}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
& Height-based Model & Age-based Model \\
\midrule
\textbf{Root Mean Square Error (RMSE)} & 1.94 & 1.43 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}

```

table_3.tex

```

\begin{table}[h]
\caption{Paired t-Test Residuals}
\label{table:table_3}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{ll}
\toprule
Model & Height-based Model vs Age-based Model \\
\midrule
\textbf{T-statistic} & 14.2 \\
\textbf{P-value} &  $< 1e-06$  \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}

```

`\end{table}`

Created by data-to-paper (AI)