Accurate Prediction of Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 8, 2024

Abstract

Accurate determination of the optimal tracheal tube depth (OTTD) is crucial for pediatric patients undergoing mechanical ventilation. However, existing methods, such as formula-based models and chest X-rays, have limitations in terms of accuracy and efficiency. To address this, we propose a data-driven approach to determine the OTTD in pediatric patients. Our study leverages a dataset of 969 post-operative mechanical ventilation cases in patients aged 0-7 years from Samsung Medical Center, incorporating features extracted from electronic health records. Utilizing machine learning techniques, our data-driven approach significantly outperforms the traditional height-based formula model, yielding an average squared residual of 1.49 compared to 3.48. Moreover, our approach demonstrates high precision in predicting tracheal tube depth with a mean absolute error of 0.8598. This data-driven approach has the potential to enhance patient safety by minimizing complications related to tracheal tube misplacement. While further validation in larger and more diverse patient cohorts is warranted, our findings suggest that implementing this approach could benefit pediatric patients undergoing mechanical ventilation.

Results

In this section, we present the results of our analysis based on a dataset of 969 pediatric patients aged 0 to 7 years who underwent post-operative mechanical ventilation at Samsung Medical Center.

Our first step was a descriptive analysis to gain insights into the patient characteristics. Table 1 presents the descriptive statistics of height and age, stratified by sex. The average age was approximately 0.732 years (SD=1.4) for female patients and 0.781 years (SD=1.47) for male patients. The mean

height was 65.4 cm (SD=18.7) for female patients and 66.5 cm (SD=19.4) for male patients.

Table 1: Descriptive statistics of height and age stratified by sex

	Age (years)		Height (cm)	
	mean	std	mean	std
Female	0.732	1.4	65.4	18.7
Male	0.781	1.47	66.5	19.4

Age (years): Average age rounded to half years
Height (cm): Average height

Subsequently, a comparative analysis was performed using a Random Forest model with 500 trees and a height-based formula model to predict the OTTD. The average squared residual for the Random Forest Model and the height-based formula model was evaluated. As shown in Table 2, our data-driven approach demonstrated highly effective performance with an average squared residual of 1.49, significantly outperforming the height-based formula model which yielded an average squared residual of 3.48 (t-test, p-value $< 10^{-6}$).

Table 2: Comparison of Random Forest model with Height-based model for predicting the optimal tracheal tube depth (OTTD)

Model	Average Squared Residual	P-value
Random Forest Model	1.49	$<10^{-6}$
Height-based formula model	3.48	$<10^{-6}$

To assess the precision of the models, we computed the mean absolute error (MAE). The Random Forest model, computed on the test data, had an MAE of 0.8598, indicating its high precision in predicting the tracheal tube depth. Conversely, the height-based formula model had a higher MAE of 1.549, reflecting larger deviations from the actual OTTD.

In summary, our analysis underscores the supremacy of our data-driven approach in determining the optimal tracheal tube depth. The descriptive statistics lay the foundation of understanding the patient characteristics, meanwhile, our data-driven approach validated through comparative analysis excels in predicting the optimal tracheal tube depth. This significantly

enhances patient safety by minimizing complications related to tracheal tube misplacement.

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
4  # Load data
5  df = pd.read_csv('tracheal_tube_insertion.csv')
6
7  # Open a file for output
8  with open("data_exploration.txt", "w") as out_file:
9
10     # Data size
11     out_file.write("# Data Size\n")
12     out_file.write("Number of rows: " + str(df.shape[0]) +
       ↪  "\n")
13     out_file.write("Number of columns: " + str(df.shape[1]) +
       ↪  "\n\n")
14
15     # Summary Statistics
16     out_file.write("# Summary Statistics\n")
17     out_file.write(df.describe().to_string())  # Full summary
       ↪  of every column
18     out_file.write("\n\n")
19
20     # Categorical Variables
21     out_file.write("# Categorical Variables\n")
```

```
    categorical_variables = df.select_dtypes(include=['object']).columns
    for variable in categorical_variables:
        out_file.write("\nVariable: " + variable + "\n")
        out_file.write("Most common values:\n")
        out_file.write(df[variable].value_counts().head().to_string())

    out_file.write("\n\n")

    # Missing Values
    out_file.write("# Missing Values\n")
    out_file.write(df.isnull().sum().to_string())  # Count of missing values

    out_file.write("\n\n")

    # Other measures of interest:
    # for example, let's count the number of unique values per column
    out_file.write("# Unique values per column\n")
    out_file.write(df.nunique().to_string())

    out_file.write("\n\n")

```

B.2 Code Description

The purpose of the provided code is to perform data exploration on the given dataset and generate a summary report in the "data_exploration.txt" file.

The code first loads the dataset using pandas' 'read_csv' function.

Next, it examines the size of the dataset by writing the number of rows and columns to the "data_exploration.txt" file.

Then, the code calculates and writes the summary statistics of the dataset, including count, mean, standard deviation, minimum, quartiles, and maximum values for each column. These summary statistics provide insights into the central tendency and spread of the data.

The code identifies the categorical variables in the dataset using the 'select_dtypes' function and writes the most common values for each cate-

gorical variable in the "data_exploration.txt" file. This helps to understand the distribution and frequency of different categories present in the dataset.

The code then determines and writes the count of missing values for each column. This information is crucial for data quality assessment and helps identify potential issues or gaps in the dataset.

Finally, the code calculates and writes the number of unique values present in each column. This measure provides an understanding of the uniqueness and variability of data within each column.

Overall, the code provides a comprehensive exploration of the dataset by generating summary statistics, analyzing categorical variables, identifying missing values, and assessing the uniqueness of values in each column. The output file, "data_exploration.txt", serves as a reference for further analysis and interpretation of the dataset.

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

# Unique values per column
tube            9
sex             2
age_c           9
ht            302
wt            320
tube_depth_G   94
```

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 pickle
6  from sklearn.model_selection import train_test_split
7  from sklearn.ensemble import RandomForestRegressor
8  from sklearn.metrics import mean_squared_error
9  from scipy.stats import ttest_rel
10
11 # LOAD DATA
12 df = pd.read_csv('tracheal_tube_insertion.csv')
13
14 # DATASET PREPARATIONS
15 # No dataset preparations are needed.
16
17 # DESCRIPTIVE STATISTICS
18 # Table 0: "Descriptive statistics of height and age
   ↪  stratified by sex"
```

```python
df0 = df.groupby('sex').agg({'age_c': ['mean', 'std'], 'ht':['mean', 'std']})
df0.index = ['Female', 'Male']
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, because the data is ready for analysis.

# ANALYSIS
## Table 1: "Comparison of Random Forest model with Formula-based model for predicting OTTD"
# Train-Test split on original dataset
X_train, X_test, y_train, y_test = train_test_split(df[['sex', 'age_c', 'ht', 'wt']], df['tube_depth_G'], test_size=0.3, random_state=42)

# ML MODEL
rf = RandomForestRegressor(n_estimators=500, max_depth=10, random_state=42)
rf.fit(X_train, y_train)
predictions_rf = rf.predict(X_test)
residuals_rf = (y_test - predictions_rf)**2

# FORMULA-BASED MODEL
predictions_formula = (X_test['ht']/10) + 5
residuals_formula = (y_test - predictions_formula)**2

# Perform paired t-test on the squared residuals of the two models
p_value = ttest_rel(residuals_rf, residuals_formula).pvalue

# Create dataframe for Table 1
df1 = pd.DataFrame({'Model': ['Random Forest', 'Height-based model'],
                    'Average Squared Residual': [residuals_rf.mean(), residuals_formula.mean()],
                    'p-value': [p_value, p_value]})
df1.set_index('Model', inplace=True)
```

```
49
50 df1.to_pickle('table_1.pkl')
51
52 # SAVE ADDITIONAL RESULTS
53 additional_results = {'Total number of observations':
   ↪  df.shape[0],
54                       'Mean Absolute Error of RF model':
                         ↪  np.mean(np.abs(y_test -
                         ↪  predictions_rf)),
55                       'Mean Absolute Error of Height-based
                         ↪  model': np.mean(np.abs(y_test -
                         ↪  predictions_formula))}
56 with open('additional_results.pkl', 'wb') as f:
57     pickle.dump(additional_results, f)
58
```

C.2 Code Description

The code performs data analysis to evaluate the performance of a Random Forest model compared to a formula-based model for predicting the Optimal Tracheal Tube Depth (OTTD) in pediatric patients. The analysis is based on a dataset of 969 patients, containing features such as patient sex, age, height, weight, and the OTTD determined by chest X-ray.

First, descriptive statistics of height and age stratified by sex are computed and stored in 'table_0.pkl'. This provides insights into the characteristics of the patient population.

Next, the dataset is split into training and testing sets, and the Random Forest (RF) model is trained using the training set. The RF model is then used to make predictions on the testing set, and the squared residuals (the differences between the predicted and actual OTTD values, squared) are computed.

As a comparison, a formula-based model is also used to predict OTTD based on the patient's height. The squared residuals of this formula-based model are also computed.

A paired t-test is performed on the squared residuals of the RF model and the formula-based model to determine if there is a significant difference between their performance. The p-value from the t-test is stored in 'table_1.pkl', along with the average squared residuals for each model.

Additionally, the code saves some additional results in 'additional_results.pkl'. These include the total number of observations in the dataset, and the mean

absolute error of predictions from both the RF model and the height-based model. These additional results provide further insights into the performance of the models.

Overall, the code enables the comparison of the RF model and the formula-based model for predicting OTTD in pediatric patients. It provides statistical measures and insights to evaluate the performance of the models and their suitability for clinical applications.

C.3 Code Output

table_0.pkl

	age_c		ht	
	mean	std	mean	std
Female	0.731544	1.402500	65.400447	18.701462
Male	0.780651	1.472808	66.514368	19.403722

table_1.pkl

Model	Average Squared Residual	p-value
Random Forest	1.493187	2.491e-09
Height-based model	3.483762	2.491e-09

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Mean Absolute Error of RF model': 0.8598            ,
    'Mean Absolute Error of Height-based model': 1.549             ,
}
```

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 typing import Dict, Optional, Any, Tuple
```

```python
from my_utils import to_latex_with_note, format_p_value, is_str_in_df, split_mapping
from my_utils import AbbrToNameDef

# PREPARATION FOR ALL TABLES
SharedMapping: AbbrToNameDef = {
    'sex': ('Sex', '0: Female, 1: Male'),
    'age_c': ('Age (years)', None),
    'ht': ('Height (cm)', None),
    'wt': ('Weight (kg)', None),
    'tube_depth_G': ('OTTD (cm)', 'Optimal tracheal tube depth, as determined by chest X-ray')
}

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in SharedMapping.items() if is_str_in_df(df, k)}
mapping.update({
    'age_c': ('Age (years)', 'Average age rounded to half years'),
    'ht': ('Height (cm)', 'Average height'),
})

abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, level=0)

# Save as latex:
to_latex_with_note(
 df,
'table_0.tex',
 caption="Descriptive statistics of height and age stratified by sex",
 label='table:statistics',
 legend=legend)

# TABLE 1
df = pd.read_pickle('table_1.pkl')
```

```

# FORMAT P-VALUES
df['p-value'] = df['p-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping = {
    'Average Squared Residual': ('Average Squared Residual', None),
    'p-value': ('P-value', None),
    'Random Forest': ('Random Forest Model', None),
    'Height-based model': ('Height-based formula model', None),
}

abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex:
to_latex_with_note(
    df,
    'table_1.tex',
    caption="Comparison of Random Forest model with Height-based model for predicting the optimal tracheal tube depth (OTTD)",
    label='table:comparison',
    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 height and age stratified by sex}
\label{table:statistics}
```

```
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrr}
\toprule
 & \multicolumn{2}{r}{Age (years)} & \multicolumn{2}{r}{Height (cm)} \\
 & mean & std & mean & std \\
\midrule
\textbf{Female} & 0.732 & 1.4 & 65.4 & 18.7 \\
\textbf{Male} & 0.781 & 1.47 & 66.5 & 19.4 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Age (years)}: Average age rounded to half years
\item \textbf{Height (cm)}: Average height
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Comparison of Random Forest model with Height-based model for
    predicting the optimal tracheal tube depth (OTTD)}
\label{table:comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Average Squared Residual & P-value \\
Model &  &  \\
\midrule
\textbf{Random Forest Model} & 1.49 & $<$1e-06 \\
\textbf{Height-based formula model} & 3.48 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
```

```
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```