Accurate Prediction of Tracheal Tube Depth in Pediatric Patients using Data-driven Models

Data to Paper

January 10, 2024

Created by data-to-paper (AI)

Abstract

Determining the optimal tracheal tube depth (OTTD) is crucial for safe mechanical ventilation in pediatric patients. However, existing methods, such as chest X-ray and formula-based models, have limitations in accurately determining OTTD. To address this gap, we conducted a comprehensive analysis using a dataset of pediatric patients who underwent mechanical ventilation after surgery. Comparing a formula-based height model with a data-driven random forest model for OTTD prediction, our results demonstrate that the random forest model significantly outperforms the height model, providing a more accurate assessment of tracheal tube depth. This finding underscores the potential of data-driven approaches in improving OTTD determination and enhancing patient safety. Nevertheless, further validation and exploration are required to optimize the model and ensure its generalizability. Our study lays the groundwork for future research aimed at improving tracheal tube positioning in pediatric patients and reducing associated complications.

Results

Beginning our comprehensive analysis, we first sought to understand the baseline attributes of the patients in question. Detailed descriptive statistics, stratified by sex, provided us with these insights, as presented in Table 1. This distribution of data aided us in comprehending the variances in age, height, weight, and especially, the optimal tracheal tube depth (OTTD), across female and male pediatric patients.

Motivated to compare conventional methods of OTTD prediction with advanced data-driven models, we first evaluated the performance of the height formula-based model. Employing the equation $predicted\_OTTD\_formula$

Table 1: Descriptive statistics of the dataset, stratified by sex

	tube		Age		Height		Weight		OTTD	
	mean	std	mean	std	mean	std	mean	std	mean	std
female	3.68	0.552	0.732	1.4	65.4	18.7	6.84	4.57	10.1	1.65
male	3.7	0.582	0.781	1.47	66.5	19.4	7.37	4.94	10.3	1.86

Weight: Patient weight, kg
Height: Patient height, cm
OTTD: Optimal tracheal tube depth as determined by chest X-ray, cm
Age: Patient age, rounded to half years

= *ht* / 10 + 5, we were able to generate predicted values of OTTD. The difference between the predicted OTTD and the actual OTTD measured through the standard chest X-ray method was the calculated residual. Our Random Forest (RF) model was then put to test. A dataset compiled from a multitude of patient features including sex, age, height, and weight, was trained through our RF model. The results, as showcased in Table 2, proved the superiority of the RF model over the height formula-based model. A determinant of this superiority was the reduced residual in predictions made by the RF model in comparison to the height formula-based model.

Table 2: Comparison between the Height Formula and Random Forest Model

	T-Statistic	P-value
Hypothesis Test	-11.6	$<10^{-6}$

P-value: P-value from T-statistic test

Lastly, we explored the possibility of a statistically significant difference between the residuals of the height format-based model and the RF model. This comparison wielded a t-statistic of -11.6 with a p-value $<10^{-6}$, which is indicative of a significant difference favoring the RF model predictions over the formula-based method.

In summary, the RF model significantly surpassed the height formula-based model in predicting optimal tracheal tube depth for pediatric patients, with less discrepancy between actual and predicted values. These findings accentuate the importance and value of utilizing advanced data-driven models in medicine, which potentially enhances patient safety by improving the accuracy of determining measures such as the optimal tracheal tube depth.

Created by data-to-paper (AI)

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)
```

Created by data-to-paper (AI)

```
#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)
```

Created by data-to-paper (AI)

B Data Exploration

B.1 Code

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

```python

import pandas as pd

# Load data
data = pd.read_csv("tracheal_tube_insertion.csv")

# Open output file
with open("data_exploration.txt", "w") as file:

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

    # Summary Statistics
    file.write("\n# Summary Statistics\n")
    file.write(str(data.describe()))

    # Categorical Variables
    file.write("\n# Categorical Variables\n")
    categorical_columns =
    ↪  data.select_dtypes(include=['object', 'bool']).columns
    if len(categorical_columns) > 0:
        for col in categorical_columns:
            file.write(f'{col} most common:
            ↪  {data[col].mode()[0]}\n')
    else:
        file.write('No categorical variables\n')

    # Missing Values
    file.write("\n# Missing Values\n")
    file.write(str(data.isna().sum()))

    # Other summary that may be:
    file.write("\n# Values count for 'sex' column\n")
    file.write(str(data.sex.value_counts()))

```

B.2 Code Description

The provided code performs an initial data exploration on the dataset by analyzing its structure, summary statistics, categorical variables, and missing values.

Firstly, the code loads the dataset using the pandas library and stores it in a dataframe named ”data”.

The code then opens an output file named ”data_exploration.txt” to write the results of the data exploration.

Next, the code calculates and writes the data size into the output file, including the number of rows and columns in the dataset.

After that, the code generates and writes the summary statistics of the dataset into the output file. This includes measures such as count, mean, standard deviation, minimum, and maximum for each numerical variable in the dataset.

The code then identifies and analyzes any categorical variables in the dataset. If there are categorical variables present, the code writes the most common value for each categorical variable into the output file. If there are no categorical variables, it writes a message indicating their absence.

Next, the code identifies and analyzes any missing values in the dataset. It calculates the number of missing values for each variable and writes this information into the output file.

Lastly, the code performs an additional analysis for the ’sex’ column by calculating the count of each unique value (male or female) and writes this information into the output file.

In summary, the code provides an initial exploration of the dataset,

Created by data-to-paper (AI)

giving insights into its structure, summary statistics, categorical variables, missing values, and a specific analysis for the 'sex' column. The results of this analysis are written into the "data_exploration.txt" file.

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
No categorical variables

# Missing Values
tube            0
sex             0
age_c           0
ht              0
wt              0
tube_depth_G    0
dtype: int64
# Values count for 'sex' column
sex
1    522
0    447
Name: count, dtype: int64
```

Created by data-to-paper (AI)

C Data Analysis

C.1 Code

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

```python

# IMPORT
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from scipy import stats
import pickle

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of the dataset, stratified by sex"
df0 = df.groupby('sex').agg(['mean', 'std'])
df0.index = ['female', 'male']
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed.

# ANALYSIS
## Table 1: "Height Formula-based Model Performance"
df1 = df.copy()
df1['predicted_OTTD_formula'] = df1['ht'] / 10 + 5
df1['residuals_formula'] = df1['tube_depth_G'] - df1['predicted_OTTD_formula']

## Table 2: "Random Forest Model Performance"
X = df[['sex', 'age_c', 'ht', 'wt']]
y = df['tube_depth_G']
```

Created by data-to-paper (AI)

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
↪   test_size=0.2, random_state=0)

rf_reg = RandomForestRegressor(n_estimators=20,
↪   random_state=0)
rf_reg.fit(X_train, y_train)
y_pred = rf_reg.predict(X_test)

residuals_rf = y_test - y_pred

## Table 3: "Comparison between the Height Formula and RF
↪   Model"
# perform the t-test
ttest_res = stats.ttest_ind(df1['residuals_formula'],
↪   residuals_rf)
# create a DataFrame
df3 = pd.DataFrame({'t_statistic': [ttest_res.statistic],
↪   'p_value': [ttest_res.pvalue]})
df3.index = ['Hypothesis Test']
df3.to_pickle('table_3.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'number_of_observations': len(df),
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)

```

Created by data-to-paper (AI)

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.

The code first loads the dataset and performs a basic descriptive analysis, calculating the mean and standard deviation of the features in the dataset, stratified by sex. The results are saved as a pickle file, providing useful summary statistics.

Next, the code compares the performance of two models in predicting OTTD. The first model is a formula-based model that uses the patient's height as a predictor. The code calculates the predicted OTTD using the formula, and then calculates the residuals, which represent the differences between the predicted and actual OTTD values.

The second model is a Random Forest regressor that uses multiple features (sex, age, height, and weight) to predict OTTD. The code splits the dataset into training and testing sets, fits the Random Forest model to the training data, and predicts OTTD for the testing data. It then calculates the residuals for this model as well.

Finally, the code performs a hypothesis test to compare the residuals of the two models. It uses the t-test to determine if there is a significant difference between the mean residuals of the formula-based model and the Random Forest model. The results of the t-test, including the t-statistic and p-value, are saved as a pickle file.

Additionally, the code saves some additional results, including the number of observations in the dataset, as a dictionary in a pickle file named "additional_results.pkl". These additional results can provide important information about the dataset for future analysis.

Overall, this code provides an analysis of the dataset and compares the performance of two models in predicting the Optimal Tracheal Tube Depth for pediatric patients undergoing mechanical ventilation after surgery.

C.3 Code Output

table_0.pkl

```
            tube               age_c                   ht                   wt
    tube_depth_G
            mean       std      mean       std       mean        std      mean
    std         mean       std
female  3.681208  0.551846  0.731544  1.402500  65.400447  18.701462  6.841902
    4.568146    10.062416  1.645478
male    3.704598  0.582023  0.780651  1.472808  66.514368  19.403722  7.370556
    4.935102    10.298276  1.857778
```

table_3.pkl

```
                 t_statistic    p_value
Hypothesis Test    -11.56268  2.405e-29
```

Created by data-to-paper (AI)

additional_results.pkl

```
{
    'number_of_observations': 969,
}
```

D LaTeX Table Design

D.1 Code

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

```python

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

# PREPARATION FOR ALL TABLES

shared_mapping: AbbrToNameDef = {
    'wt': ('Weight', 'Patient weight, kg'),
    'ht': ('Height', 'Patient height, cm'),
    'sex': ('Sex', 'Patient sex, {0: "female", 1: "male"}'),
    'tube_depth_G': ('OTTD', 'Optimal tracheal tube depth as
    ↪  determined by chest X-ray, cm'),
    'age_c': ('Age', 'Patient age, rounded to half years'),
}

# REUSABLE FUNCTION
def convert_to_tex(df, filename: str, caption: str, label:
↪  str, mapping, note: str = None):
    abbrs_to_names, legend = split_mapping(mapping)
    df = df.rename(columns=abbrs_to_names,
    ↪  index=abbrs_to_names)
    to_latex_with_note(df, filename, caption,
    ↪  f'table:{label}', note, legend)

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

Created by data-to-paper (AI)

```

# RENAME ROWS AND COLUMNS
mapping_table_0 = {k: v for k, v in shared_mapping.items() if
↪   is_str_in_df(df, k)}
convert_to_tex(df,
               filename='table_0.tex',
               caption='Descriptive statistics of the dataset,
                ↪   stratified by sex',
               label='descriptive',
               mapping=mapping_table_0)

# TABLE 3:
df = pd.read_pickle('table_3.pkl')

# FORMAT VALUES
df['p_value'] = df['p_value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping_table_3 = {k: v for k, v in shared_mapping.items() if
↪   is_str_in_df(df, k)}
mapping_table_3 |= {
    't_statistic': ('T-Statistic', None),
    'p_value': ('P-value', 'P-value from T-statistic test')
}

convert_to_tex(df,
               filename='table_3.tex',
               caption='Comparison between the Height Formula
                ↪   and Random Forest Model',
               label='comparison',
               mapping=mapping_table_3)

```

D.2 Provided Code

The code above is using the following provided functions:

Created by data-to-paper (AI)

```python
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

```

Created by data-to-paper (AI)

D.3 Code Output

table_0.tex

```
\begin{table}[h]
\caption{Descriptive statistics of the dataset, stratified by sex}
\label{table:descriptive}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrrrrrrr}
\toprule
 & \multicolumn{2}{r}{tube} & \multicolumn{2}{r}{Age} &
    \multicolumn{2}{r}{Height} & \multicolumn{2}{r}{Weight} &
    \multicolumn{2}{r}{OTTD} \\
 & mean & std & mean & std & mean & std & mean & std & mean & std \\
\midrule
\textbf{female} & 3.68 & 0.552 & 0.732 & 1.4 & 65.4 & 18.7 & 6.84 & 4.57 & 10.1
    & 1.65 \\
\textbf{male} & 3.7 & 0.582 & 0.781 & 1.47 & 66.5 & 19.4 & 7.37 & 4.94 & 10.3 &
    1.86 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Weight}: Patient weight, kg
\item \textbf{Height}: Patient height, cm
\item \textbf{OTTD}: Optimal tracheal tube depth as determined by chest X-ray,
    cm
\item \textbf{Age}: Patient age, rounded to half years
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
\begin{table}[h]
\caption{Comparison between the Height Formula and Random Forest Model}
\label{table:comparison}
\begin{threeparttable}
```

Created by data-to-paper (AI)

```
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & T-Statistic & P-value \\
\midrule
\textbf{Hypothesis Test} & -11.6 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{P-value}: P-value from T-statistic test
\end{tablenotes}
\end{threeparttable}
\end{table}
```

Created by data-to-paper (AI)