Enhancing Accuracy of Tracheal Tube Placement in Pediatric Patients through Data-Driven Methods

Data to Paper

January 9, 2024

Abstract

Accurate positioning of the tracheal tube tip is of paramount importance in pediatric patients undergoing mechanical ventilation, as misplaced placement can lead to severe complications. However, the current methods for determining the optimal tracheal tube depth (OTTD) suffer from limitations, highlighting the need for an accurate and efficient alternative. In this study, we propose a novel data-driven approach to address this research gap. Leveraging a comprehensive dataset of pediatric patients who underwent surgery and post-operative mechanical ventilation, we developed and trained machine learning models based on patient features extracted from electronic health records. We compared the performance of two models, random forest and elastic net, and evaluated their predictive power. Our results reveal significant associations between patient characteristics and the OTTD. Notably, age exhibited a negative correlation with the OTTD, while weight showed a positive correlation. These findings have significant clinical implications, improving the accuracy of tracheal tube positioning and reducing the risk of complications in pediatric patients requiring mechanical ventilation. However, further research is needed to validate and refine our models for broader clinical implementation, considering potential sources of bias and limitations within the dataset and methodology.

Results

In this section, we present the results of our analysis, focusing on three key aspects: descriptive statistics of age, height, weight, and OTTD stratified by sex; a comparison of the predictive power between the Random Forest and Elastic Net models; and the association of age, sex, height, and weight with the OTTD.

First, to understand the characteristics of our pediatric population and identify potential gender differences, we examined the descriptive statistics of age, height, weight, and OTTD stratified by sex, as shown in Table 1. Our results show that the mean age of both female (0.732 years) and male (0.781 years) pediatric patients were less than one year, indicating a young patient cohort. The mean heights were 65.4 cm and 66.5 cm for females and males respectively, while the respective mean weights were 6.84 kg and 7.37 kg. The OTTD, our response variable of interest, showed mean values of 10.1 cm for females and 10.3 cm for males.

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

	Female	Male
Age Mean	0.732	0.781
Age Std Dev	1.4	1.47
Height Mean	65.4	66.5
Height Std Dev	18.7	19.4
Weight Mean	6.84	7.37
Weight Std Dev	4.57	4.94
OTTD Mean	10.1	10.3
OTTD Std Dev	1.65	1.86

Age Mean: Mean age in years
Age Std Dev: Standard deviation of age
Height Mean: Mean height in cm
Height Std Dev: Standard deviation of height
OTTD Mean: Mean OTTD in cm
OTTD Std Dev: Standard deviation of OTTD
Weight Mean: Mean weight in kg
Weight Std Dev: Standard deviation of weight

Next, to identify the best predictive model for determining the OTTD, we compared the prediction performance of the Random Forest and Elastic Net models. The performance of these two models, which were chosen because of their ability to handle noisy and intercorrelated predictor variables respectively, was assessed using the mean squared residual. As shown in Table 2, the Elastic Net model performed better than the Random Forest model with a lower mean squared residual of 1.14, compared to 1.54 for the latter. The difference was found to be statistically significant with a p-value of 0.00711, suggesting superior predictive accuracy of the Elastic Net model.

Finally, to ascertain the significant factors in predicting the OTTD, we

Table 2: Comparison of predictive power between Random Forest and Elastic Net models

	Mean Squared Residual	P-value
Random Forest	1.54	0.00711
Elastic Net	1.14	-

P-value: Derived from t-test comparing the mean squared residuals of both models

performed a multiple linear regression analysis. As presented in Table 3, the results revealed significant characters associated with OTTD. Age showed a negative correlation with OTTD (coefficient = -0.159, p-value = 0.00429), while weight exhibited a strong positive correlation (coefficient = 0.244, p-value $< 10^{-6}$). However, sex and height didn't show significant correlations with the OTTD. These findings provide insights into the factors influencing the optimal tracheal tube depth in pediatric patients requiring mechanical ventilation.

Table 3: Association of age, sex, height, and weight with OTTD

	Coefficient	P-value
Intercept	7.23	$<10^{-6}$
Gender	0.0927	0.217
Age	-0.159	0.00429
Height	0.0196	0.0212
Weight	0.244	$<10^{-6}$

Gender: 0: Female, 1: Male
Age: Patient age in years
Height: Patient height in cm
Weight: Patient weight in kg

Taken together, our results show that the Elastic Net model provides a more accurate forecast of the OTTD than the Random Forest model. The regression analysis further indicates that age and weight are significant factors influencing the OTTD. The findings underscore the potential of machine learning in enhancing clinical predictions, and in this case, helping to improve the accuracy of tracheal tube placement in pediatric patients.

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 CSV data into a DataFrame
df = pd.read_csv('tracheal_tube_insertion.csv')

# Create a text file and write data exploration summary into it
with open('data_exploration.txt', 'w') as f:

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

    # Summary Statistics
    f.write("# Summary Statistics\n")
    desc = df.describe(include='all')
    for col in desc.columns:
        f.write(f"{col}\n{desc[col]}\n\n")

    # Categorical Variables
    f.write("# Categorical Variables\n")
```

```python
    categorical_cols = df.select_dtypes(include=['object','bool']).columns
    for col in categorical_cols:
        f.write(f"{col} - most common values:\n{df[col].value_counts().head().to_string()}\n\n")

    # Missing Values
    f.write("# Missing Values\n")
    missing_vals = df.isnull().sum()
    for key, val in missing_vals.items():
        if val > 0:
            f.write(f"{key}: {val} missing values\n")

    # Addition could be done here for exploring other data metrics

```

B.2 Code Description

The provided code performs a data exploration on the dataset "tracheal_tube_insertion.csv" and writes the exploration summary into a text file called "data_exploration.txt".

The code first loads the CSV data into a DataFrame using the pandas library.

Then, it performs several analysis steps to explore the dataset:

1. Data Size: The code calculates the number of rows and columns in the dataset and writes these values into the output file.

2. Summary Statistics: The code computes summary statistics for each column in the dataset using the describe() function. It includes statistical measures like count, mean, standard deviation, minimum, quartiles, and maximum for numerical columns, and count, unique values, top value, and frequency for categorical columns. These summary statistics are then written into the output file.

3. Categorical Variables: The code identifies the categorical variables in the dataset and writes the most common values for each categorical variable into the output file. This provides information on the distribution of the categorical variables.

4. Missing Values: The code checks for missing values in each column of the dataset using the isnull() function. If any column has missing values, it writes the column name and the number of missing values into the output file.

The code provides a structured summary of the dataset, including information on data size, summary statistics, common values for categorical variables, and missing values. This exploration serves as an initial step in understanding the dataset before further analysis or modeling.

The "data_exploration.txt" file generated by the code contains all the information mentioned above, allowing researchers to easily refer to the key insights and characteristics of the dataset.

B.3 Code Output

data_exploration.txt

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

# Summary Statistics
tube
count      969
mean     3.694
std     0.5681
min        2.5
25%        3.5
50%        3.5
75%          4
max          6
Name: tube, dtype: float64

sex
count      969
mean    0.5387
std     0.4988
min          0
25%          0
50%          1
75%          1
max          1
Name: sex, dtype: float64

age_c
```

```
count     969
mean    0.758
std      1.44
min         0
25%         0
50%         0
75%         1
max         7
Name: age_c, dtype: float64

ht
count     969
mean       66
std     19.08
min      29.5
25%        52
50%      61.5
75%        76
max     135.4
Name: ht, dtype: float64

wt
count     969
mean    7.127
std     4.774
min      0.57
25%      3.56
50%       5.7
75%       9.5
max        31
Name: wt, dtype: float64

tube_depth_G
count     969
mean    10.19
std     1.766
min       5.9
25%         9
50%       9.8
75%      11.2
```

```
max       19.2
Name: tube_depth_G, dtype: float64

# Categorical Variables
# Missing Values
```

C Data Analysis

C.1 Code

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

```python

# IMPORT
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNetCV
from scipy import stats
import statsmodels.formula.api as smf
import pickle

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

# DATASET PREPARATIONS

# No dataset preparations are needed

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of age, height, weight, and OTTD stratified by sex"
df0 = data.groupby('sex').agg({'age_c': ['mean', 'std'], 'ht': ['mean', 'std'], 'wt': ['mean', 'std'], 'tube_depth_G': ['mean', 'std']})
df0.index = ['Female', 'Male']
```

```
df0.columns = ['_'.join(col).strip() for col in df0.columns.values]
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, because the data are already in suitable format for machine learning.

# ANALYSIS
features = ['sex', 'age_c', 'ht', 'wt']
target = 'tube_depth_G'
X = data[features]
y = data[target]

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Train and predict with Random Forest
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
rf_y_pred = rf.predict(X_test)

# Train and predict with Elastic Net
en = ElasticNetCV()
en.fit(X_train, y_train)
en_y_pred = en.predict(X_test)

# Calculate squared residuals
rf_squared_residuals = (rf_y_pred - y_test)**2
en_squared_residuals = (en_y_pred - y_test)**2

## Table 1: "Comparison of predictive power between Random Forest and Elastic Net models"
t_test_results = stats.ttest_rel(rf_squared_residuals, en_squared_residuals)

df1 = pd.DataFrame({
    'Mean Squared Residual': [np.mean(rf_squared_residuals), np.mean(en_squared_residuals)],
```

```
    'P-value': [t_test_results.pvalue, '-']
}, index=['Random Forest', 'Elastic Net'])
df1.to_pickle('table_1.pkl')

model = smf.ols(formula='tube_depth_G ~ sex + age_c + ht + wt', data=data)
results = model.fit()

## Table 2: "Association of age, sex, height, and weight with OTTD"
df2 = pd.DataFrame({
    'Coefficient': results.params,
    'P-value': results.pvalues
}, index=['Intercept', 'sex', 'age_c', 'ht', 'wt'])
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(data),
 'Root Mean Square Error of Random Forest': np.sqrt(mean_squared_error(y_test, rf_y_pred)),
 'Root Mean Square Error of Elastic Net': np.sqrt(mean_squared_error(y_test, en_y_pred)),
 'R-squared of Multiple linear regression': results.rsquared
}
with open('additional_results.pkl', 'wb') as f:
 pickle.dump(additional_results, f)

```

C.2 Code Description

The provided code performs an analysis on a dataset containing information on pediatric patients who received post-operative mechanical ventilation. The aim of the analysis is to determine the Optimal Tracheal Tube Depth (OTTD) for these patients, which is crucial for the safety and efficacy of the ventilation procedure.

The code initially imports necessary libraries for data manipulation, preprocessing, statistical analysis, and machine learning. The dataset is then loaded into a pandas DataFrame.

Descriptive statistics are calculated to summarize the age, height, weight, and OTTD of the patients, stratified by sex. The results are saved in a pickle file named 'table_0.pkl'.

After preparing the dataset by selecting the relevant features and target variable, it is split into training and testing sets. Two machine learning models, Random Forest and Elastic Net, are trained on the training data and used to predict the OTTD for the testing data. The squared residuals are calculated for each model.

A t-test is performed to compare the predictive power of the Random Forest and Elastic Net models based on the squared residuals. The results, including the mean squared residuals and the t-test p-value, are saved in 'table_1.pkl'.

A multiple linear regression model is fitted using the statsmodels library to assess the association of age, sex, height, and weight with the OTTD. The coefficients and p-values are extracted and saved in 'table_2.pkl'.

The code also saves additional results in 'additional_results.pkl'. These include the total number of observations, the root mean square error of the Random Forest and Elastic Net models, and the R-squared value of the multiple linear regression model.

Overall, the code performs descriptive statistics, machine learning modeling, statistical analysis, and saves the results of these analyses in pickle files for further investigation and reporting.

C.3 Code Output

table_0.pkl

```
        age_c_mean  age_c_std    ht_mean     ht_std   wt_mean    wt_std
    tube_depth_G_mean  tube_depth_G_std
Female    0.731544   1.402500  65.400447  18.701462  6.841902  4.568146
    10.062416          1.645478
Male      0.780651   1.472808  66.514368  19.403722  7.370556  4.935102
    10.298276          1.857778
```

table_1.pkl

```
               Mean Squared Residual   P-value
Random Forest               1.537771  0.007111
Elastic Net                 1.139353         -
```

table_2.pkl

	Coefficient	P-value
Intercept	7.228066	3.562e-77
sex	0.092701	0.2174
age_c	-0.158746	0.004289
ht	0.019552	0.02123
wt	0.244348	1.652e-12

additional_results.pkl


```
{
    'Total number of observations': 969,
    'Root Mean Square Error of Random Forest': 1.24                ,
    'Root Mean Square Error of Elastic Net': 1.067               ,
    'R-squared of Multiple linear regression': 0.5733            ,
}
```


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 my_utils import to_latex_with_note, format_p_value, is_str_in_df, split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES

# Common dictionary for column (row) labels that are common across the tables
shared_mapping: AbbrToNameDef = {
    'sex': ('Gender', '0: Female, 1: Male'),
    'age_c': ('Age', 'Patient age in years'),
    'ht': ('Height', 'Patient height in cm'),
    'wt': ('Weight', 'Patient weight in kg'),
    'tube_depth_G': ('OTTD', 'Optimal tracheal tube depth in cm'),
```

```python
}

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

# FORMAT VALUES
# Not applicable in this case

# RENAME ROWS AND COLUMNS
mapping0 = {k: v for k, v in shared_mapping.items() if is_str_in_df(df0, k)}
mapping0 |= {
    'age_c_mean': ('Age Mean', 'Mean age in years'),
    'age_c_std': ('Age Std Dev', 'Standard deviation of age'),
    'ht_mean': ('Height Mean', 'Mean height in cm'),
    'ht_std': ('Height Std Dev', 'Standard deviation of height'),
    'tube_depth_G_mean': ('OTTD Mean', 'Mean OTTD in cm'),
    'tube_depth_G_std': ('OTTD Std Dev', 'Standard deviation of OTTD'),
    'wt_mean': ('Weight Mean', 'Mean weight in kg'),
    'wt_std': ('Weight Std Dev', 'Standard deviation of weight')
}
abbrs_to_names0, legend0 = split_mapping(mapping0)
df0 = df0.rename(index=abbrs_to_names0, columns=abbrs_to_names0)

# Transpose the DataFrame to make the table narrow
df0 = df0.T

# Convert DataFrame to LaTeX and save it in a .tex file
to_latex_with_note(
    df0, 'table_0.tex',
    caption="Descriptive statistics of age, height, weight, and OTTD, stratified by sex",
    label='table:table_0',
    note=None,
    legend=legend0)

```

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

# FORMAT VALUES
df1['P-value'] = df1['P-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping1 = {'P-value': ('P-value', 'Derived from t-test
↪   comparing the mean squared residuals of both models')}
abbrs_to_names1, legend1 = split_mapping(mapping1)
df1 = df1.rename(index=abbrs_to_names1,
↪   columns=abbrs_to_names1)

# Convert DataFrame to LaTeX and save it in a .tex file
to_latex_with_note(
    df1, 'table_1.tex',
    caption="Comparison of predictive power between Random
    ↪   Forest and Elastic Net models",
    label='table:table_1',
    note=None,
    legend=legend1)

# TABLE 2:
df2 = pd.read_pickle('table_2.pkl')

# FORMAT VALUES
df2['P-value'] = df2['P-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping2 = {k: v for k, v in shared_mapping.items() if
↪   is_str_in_df(df2, k)}
mapping2 |= { 'Intercept': ('Intercept', None) }
abbrs_to_names2, legend2 = split_mapping(mapping2)
df2 = df2.rename(index=abbrs_to_names2,
↪   columns=abbrs_to_names2)

# Convert DataFrame to LaTeX and save it in a .tex file
to_latex_with_note(
    df2, 'table_2.tex',
```

```
    caption="Association of age, sex, height, and weight with
    ↪ OTTD",
    label='table:table_2',
    note=None,
    legend=legend2)

```

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):
```

```python
    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

```latex
\begin{table}[h]
\caption{Descriptive statistics of age, height, weight, and OTTD, stratified by
    sex}
\label{table:table_0}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & Female & Male \\
\midrule
\textbf{Age Mean} & 0.732 & 0.781 \\
\textbf{Age Std Dev} & 1.4 & 1.47 \\
\textbf{Height Mean} & 65.4 & 66.5 \\
\textbf{Height Std Dev} & 18.7 & 19.4 \\
\textbf{Weight Mean} & 6.84 & 7.37 \\
\textbf{Weight Std Dev} & 4.57 & 4.94 \\
\textbf{OTTD Mean} & 10.1 & 10.3 \\
\textbf{OTTD Std Dev} & 1.65 & 1.86 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Age Mean}: Mean age in years
\item \textbf{Age Std Dev}: Standard deviation of age
\item \textbf{Height Mean}: Mean height in cm
```

```
\item \textbf{Height Std Dev}: Standard deviation of height
\item \textbf{OTTD Mean}: Mean OTTD in cm
\item \textbf{OTTD Std Dev}: Standard deviation of OTTD
\item \textbf{Weight Mean}: Mean weight in kg
\item \textbf{Weight Std Dev}: Standard deviation of weight
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Comparison of predictive power between Random Forest and Elastic Net
    models}
\label{table:table_1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Mean Squared Residual & P-value \\
\midrule
\textbf{Random Forest} & 1.54 & 0.00711 \\
\textbf{Elastic Net} & 1.14 & - \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{P-value}: Derived from t-test comparing the mean squared residuals
    of both models
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Association of age, sex, height, and weight with OTTD}
```

```
\label{table:table_2}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Coefficient & P-value \\
\midrule
\textbf{Intercept} & 7.23 & $<$1e-06 \\
\textbf{Gender} & 0.0927 & 0.217 \\
\textbf{Age} & -0.159 & 0.00429 \\
\textbf{Height} & 0.0196 & 0.0212 \\
\textbf{Weight} & 0.244 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Gender}: 0: Female, 1: Male
\item \textbf{Age}: Patient age in years
\item \textbf{Height}: Patient height in cm
\item \textbf{Weight}: Patient weight in kg
\end{tablenotes}
\end{threeparttable}
\end{table}
```