Predicting Optimal Tracheal Tube Depth in Pediatric Patients on Mechanical Ventilation

Data to Paper

February 20, 2024

Abstract

Accurate placement of the tracheal tube is crucial for pediatric patients undergoing mechanical ventilation. Despite the potential risks of misplacement, determining the Optimal Tracheal Tube Depth (OTTD) remains challenging. Existing methods rely on chest X-ray or formula-based models but lack precision. To address this, we developed predictive models using patient features to estimate the OTTD. Our dataset comprised 969 pediatric patients aged 0-7 years who received post-operative mechanical ventilation. Leveraging Random Forest and Elastic Net algorithms, we evaluated the predictive performance based on sex, age, height, and weight. The Elastic Net model demonstrated superior accuracy in estimating the OTTD compared to the Random Forest model. Paired analysis further confirmed the superiority of the Elastic Net model. The development of accurate predictive models based on patient features offers an alternative to chest X-ray assessment for determining the OTTD. These findings have the potential to enhance patient safety and reduce complications associated with tracheal tube misplacement. However, further refinement and validation of the models are necessary for widespread implementation in clinical practice.

Results

In this section, we present the results of our analysis aimed at determining the Optimal Tracheal Tube Depth (OTTD) in pediatric patients undergoing mechanical ventilation. We utilized a dataset of 969 pediatric patients aged 0-7 years who received post-operative mechanical ventilation after undergoing surgery. Our analysis focused on developing predictive models using patient features to estimate the OTTD and reducing the reliance on chest X-ray assessment.

To achieve this, we employed two different predictive models, Random Forest and Elastic Net, to estimate the OTTD based on patient features including sex, age, height, and weight. The Random Forest model utilized a maximum number of features of "sqrt" and a number of estimators of 500, while the Elastic Net model utilized an alpha value of 0.1 and an l1 ratio of 0.1.

First, we examined the descriptive statistics of patient features and OTTD stratified by sex (Table 1). The mean and standard deviation were calculated for patient features such as age, height, weight, and OTTD. We observed potential differences in these features based on patient sex. For instance, the mean OTTD was found to be 10.1 cm (SD=1.65) for female patients and 10.3 cm (SD=1.86) for male patients. Other features such as age, height, and weight also exhibited slight variations between the two groups.

Table 1: Descriptive statistics of patient features and OTTD stratified by sex

	sex	0	1
Tube ID	**mean**	3.68	3.7
	std	0.552	0.582
Age	**mean**	0.732	0.781
	std	1.4	1.47
Height	**mean**	65.4	66.5
	std	18.7	19.4
Weight	**mean**	6.84	7.37
	std	4.57	4.94
Optimal Tracheal Tube Depth (OTTD)	**mean**	10.1	10.3
	std	1.65	1.86

Age: Patient age (in years)
Height: Patient height (in cm)
Weight: Patient weight (in kg)
Tube ID: Internal diameter of the tube (in mm)
Optimal Tracheal Tube Depth (OTTD): As determined by chest X-ray (in cm)

Next, we evaluated the performance of the Random Forest and Elastic Net models in predicting the OTTD. The models were assessed based on mean squared error (MSE) values calculated on the test set (Table 2). The Elastic Net model outperformed the Random Forest model, achieving an MSE of 1.24 compared to the Random Forest model's MSE of 1.57. These findings indicate that the Elastic Net model demonstrated superior predic-

tive performance in estimating the OTTD compared to the Random Forest model.

Table 2: Performance of Random Forest and Elastic Net models in predicting OTTD

Model	Mean Squared Error
Random Forest	1.57
Elastic Net	1.24

Finally, we conducted a paired t-test to compare the squared residuals of the two models (Table 3). The analysis revealed a significant difference between the squared residuals of the Elastic Net and Random Forest models, with a t statistic of 2.36 (p-value=0.0194). This further supports the superior performance of the Elastic Net model in estimating the OTTD.

Table 3: Paired t-test comparing the squared residuals of the two models

Model 1 residuals	Model 2 residuals	t statistic	p value
Random Forest	Elastic Net	2.36	0.0194

In summary, our analysis suggests that the Elastic Net model shows promise in accurately estimating the Optimal Tracheal Tube Depth in pediatric patients on mechanical ventilation. Furthermore, patient features, such as sex, age, height, and weight, may play a crucial role in determining the OTTD. The results highlight the potential of utilizing predictive models based on patient features to minimize the need for chest X-ray assessment, providing an alternative approach in clinical practice.

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 the dataset
6  df = pd.read_csv("tracheal_tube_insertion.csv")
7
8  header = ['Data Size', 'Summary Statistics', 'Categorical
   ↪  Variables', 'Missing Values']
9
10 # count of rows and columns
11 data_size = "Number of Rows: {}\nNumber of Columns:
   ↪  {}".format(df.shape[0], df.shape[1])
12
13 # summary statistics
14 summary_stats =
   ↪  df.describe(include='all').transpose().to_string()
15
16 # unique values in categorical variable
17 cat_vars = "sex: {}".format(df.sex.value_counts().to_string())
18
19 # check count of missing values in all columns
20 missing_values = df.isnull().sum().to_string()
21
```

```
exploratory_outputs = [data_size, summary_stats, cat_vars,
↪  missing_values]

with open("data_exploration.txt", 'w') as f:
    for i,section in enumerate(exploratory_outputs):
        f.write("# " + header[i] + "\n" + section + "\n\n")

```

B.2 Code Description

The provided code performs an initial data exploration on the given dataset.

First, the code reads the "tracheal_tube_insertion.csv" file using the pandas library and stores it in a dataframe called "df".

The code then proceeds to perform the following analysis steps:

1. Data Size: The code determines the number of rows and columns in the dataset, and stores this information as "Number of Rows: <number>" and "Number of Columns: <number>".

2. Summary Statistics: The code computes the summary statistics for each column in the dataset using the describe() function. It generates statistics such as count, mean, standard deviation, minimum, and quartiles for numerical columns, and provides the count, number of unique values, top value, and frequency of top value for categorical columns. The summary statistics are stored as a table in the "summary_stats" variable.

3. Categorical Variables: The code calculates the count of unique values in the 'sex' column, which represents the patient's sex. It provides the count of female patients and male patients in a formatted string.

4. Missing Values: The code checks for missing values in each column of the dataset using the isnull() function. It counts the number of missing values for each column and stores the result as a string in the "missing_values" variable.

Finally, the code writes the results of the data exploration analysis to a text file named "data_exploration.txt", using the "header" list as section titles. Each section includes the corresponding result or analysis step.

The "data_exploration.txt" file will contain the following sections:

1. Data Size: The number of rows and columns in the dataset. 2. Summary Statistics: The summary statistics for each column in the dataset. 3. Categorical Variables: The count of unique values in the 'sex' column. 4. Missing Values: The count of missing values in each column.

This information can be used to gain a preliminary understanding of the dataset and identify any potential issues or patterns before proceeding with

further analysis.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of Rows: 969
Number of Columns: 6

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

# Categorical Variables
sex: sex
1    522
0    447

# 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:

```python

# IMPORT
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNet
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_rel
import pickle
import warnings
warnings.filterwarnings('ignore')

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# Table 0: "Mean and standard deviation of patient features and tube_depth_G stratified by sex"
df0 = df.groupby('sex').agg(['mean','std']).transpose()
df0.to_pickle('table_0.pkl')

# PREPROCESSING
# No preprocessing is needed, because all the features are numerical and there are no missing values.

# ANALYSIS
# Determining feature and target variables
X = df[["sex", "age_c", "ht", "wt"]]
y = df["tube_depth_G"]

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```python
# Table 1: "Performance (mean squared error) of Random Forest and Elastic Net models on the test set"
# Hyperparameter tuning for Random Forest
rf = RandomForestRegressor(random_state=42)
param_grid = {
    'n_estimators': [200, 500],
    'max_features': ['sqrt', 'log2']
}
CV_rf = GridSearchCV(estimator=rf, param_grid=param_grid, cv=2)
CV_rf.fit(X_train, y_train)
best_rf = CV_rf.best_estimator_
rf_pred = best_rf.predict(X_test)
rf_mse = mean_squared_error(y_test, rf_pred)

# Hyperparameter tuning for Elastic Net
en = ElasticNet(random_state=42)
param_grid = {
    'alpha' : [0.1, 0.5, 1, 2, 5],
    'l1_ratio' : [0.1, 0.5, 1]
}
CV_en = GridSearchCV(en, param_grid=param_grid, cv=2)
CV_en.fit(X_train, y_train)
best_en = CV_en.best_estimator_
en_pred = best_en.predict(X_test)
en_mse = mean_squared_error(y_test, en_pred)

df1 = pd.DataFrame({'Model': ['Random Forest', 'Elastic Net'], 'Mean Squared Error': [rf_mse, en_mse]})
df1.set_index('Model', inplace=True)
df1.to_pickle('table_1.pkl')

# Table 2: "Paired t-test comparing the squared residuals of the two models"
rf_res = (rf_pred - y_test)**2
en_res = (en_pred - y_test)**2
ttest_results = ttest_rel(rf_res, en_res)
```

```
df2 = pd.DataFrame({'Model 1 residuals': ['Random Forest'],
    'Model 2 residuals': ['Elastic Net'], 't statistic':
    [ttest_results.statistic], 'p value':
    [ttest_results.pvalue]})
df2.set_index('Model 1 residuals', inplace=True)
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': df.shape[0],
 'Best parameters of Random Forest model': CV_rf.best_params_,
 'Best parameters of Elastic Net model': CV_en.best_params_
}
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 require mechanical ventilation after surgery. The goal is to determine the optimal tracheal tube depth (OTTD), which is crucial for patient safety.

The code first loads the dataset and prepares it for analysis. It then calculates descriptive statistics, such as the mean and standard deviation of patient features and OTTD stratified by sex, and stores the results in a table.

Next, the code performs preprocessing on the dataset, which is not needed in this case as all the features are numerical and there are no missing values.

The code then proceeds to the analysis step. It splits the dataset into training and testing sets and defines the feature variables (patient sex, age, height, and weight) and the target variable (OTTD).

Two machine learning models, Random Forest and Elastic Net, are trained on the training set and evaluated on the testing set. The code uses grid search to find the best hyperparameters for each model. The mean squared error (MSE) is calculated for both models on the testing set, and the results are stored in a table.

To compare the performance of the two models, a paired t-test is conducted on the squared residuals of the predictions from each model. The t-statistic and p-value of the t-test are then saved in another table.

Finally, the code saves additional results, including the total number of observations in the dataset, and the best parameters found for the Random Forest and Elastic Net models.

The "additional_results.pkl" file contains a pickled dictionary with these additional results, including the total number of observations and the best parameters of each model.

Overall, the code performs data analysis on the pediatric patient dataset, determines the OTTD using machine learning models, compares the performance of the models, and saves the results for further reference.

C.3 Code Output

table_0.pkl

sex		0	1
tube	mean	3.681208	3.704598
	std	0.551846	0.582023
age_c	mean	0.731544	0.780651
	std	1.402500	1.472808
ht	mean	65.400447	66.514368
	std	18.701462	19.403722
wt	mean	6.841902	7.370556
	std	4.568146	4.935102
tube_depth_G	mean	10.062416	10.298276
	std	1.645478	1.857778

table_1.pkl

Model	Mean Squared Error
Random Forest	1.572975
Elastic Net	1.239276

table_2.pkl

Model 1 residuals	Model 2 residuals	t statistic	p value
Random Forest	Elastic Net	2.357123	0.01942

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Best parameters of Random Forest model': {'max_features': 'sqrt',
    'n_estimators': 500},
    'Best parameters of Elastic Net model': {'alpha': 0.1, 'l1_ratio': 0.1},
}
```

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, Any, Tuple, Optional
from my_utils import to_latex_with_note, format_p_value,
 ↪  is_str_in_df, split_mapping

# PREPARATION FOR ALL TABLES
shared_mapping: Dict[str, Tuple[Optional[str], Optional[str]]]
 ↪  = {
 'sex': ('Sex', 'Patient sex (0 for female, 1 for male)'),
 'age_c': ('Age', 'Patient age (in years)'),
 'ht': ('Height', 'Patient height (in cm)'),
 'wt': ('Weight', 'Patient weight (in kg)'),
 'tube': ('Tube ID', 'Internal diameter of the tube (in mm)'),
 'tube_depth_G': ('Optimal Tracheal Tube Depth (OTTD)', 'As
 ↪  determined by chest X-ray (in cm)'),
}

# TABLE 0
df = 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(df, k)}
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_0.tex',
 caption="Descriptive statistics of patient features and OTTD stratified by sex",
 label='table:patient_features_ottd_by_sex',
 legend=legend)

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
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="Performance of Random Forest and Elastic Net models in predicting OTTD",
 label='table:model_performance',
 legend=legend)

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

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

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

```

```
# Save as latex
to_latex_with_note(
 df, 'table_2.tex',
 caption="Paired t-test comparing the squared residuals of the two models",
 label='table:paired_ttest',
 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]]]
```

```python

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 patient features and OTTD stratified by sex}
\label{table:patient_features_ottd_by_sex}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llrr}
\toprule
 & sex & 0 & 1 \\
\midrule
\multirow[t]{2}{*}{\textbf{Tube ID}} & \textbf{mean} & 3.68 & 3.7 \\
\textbf{} & \textbf{std} & 0.552 & 0.582 \\
\cline{1-4}
\multirow[t]{2}{*}{\textbf{Age}} & \textbf{mean} & 0.732 & 0.781 \\
\textbf{} & \textbf{std} & 1.4 & 1.47 \\
\cline{1-4}
\multirow[t]{2}{*}{\textbf{Height}} & \textbf{mean} & 65.4 & 66.5 \\
\textbf{} & \textbf{std} & 18.7 & 19.4 \\
\cline{1-4}
\multirow[t]{2}{*}{\textbf{Weight}} & \textbf{mean} & 6.84 & 7.37 \\
\textbf{} & \textbf{std} & 4.57 & 4.94 \\
\cline{1-4}
\multirow[t]{2}{*}{\textbf{Optimal Tracheal Tube Depth (OTTD)}} & \textbf{mean}
```

```
    & 10.1 & 10.3 \\
\textbf{} & \textbf{std} & 1.65 & 1.86 \\
\cline{1-4}
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Age}: Patient age (in years)
\item \textbf{Height}: Patient height (in cm)
\item \textbf{Weight}: Patient weight (in kg)
\item \textbf{Tube ID}: Internal diameter of the tube (in mm)
\item \textbf{Optimal Tracheal Tube Depth (OTTD)}: As determined by chest X-ray
    (in cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Performance of Random Forest and Elastic Net models in predicting OTTD}
\label{table:model_performance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Mean Squared Error \\
Model &  \\
\midrule
\textbf{Random Forest} & 1.57 \\
\textbf{Elastic Net} & 1.24 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
```

```
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Paired t-test comparing the squared residuals of the two models}
\label{table:paired_ttest}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llrl}
\toprule
 & Model 2 residuals & t statistic & p value \\
Model 1 residuals &  &  &  \\
\midrule
\textbf{Random Forest} & Elastic Net & 2.36 & 0.0194 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
\end{tablenotes}
\end{threeparttable}
\end{table}
```