Improved Estimation of Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

January 5, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) in pediatric patients is crucial for safe and effective mechanical ventilation. However, accurate estimation of OTTD using current methods, such as chest X-ray and formula-based models, remains challenging. This study aims to improve the estimation of OTTD in pediatric patients aged 0-7 years using machine learning. We built machine learning models utilizing patient features extracted from electronic health records, and compared their performance with formula-based models. The dataset consisted of 969 patients who underwent post-operative mechanical ventilation. Our results demonstrate that the machine learning models consistently outperformed the formula-based models in estimating OTTD. The Random Forest model achieved the highest accuracy among the evaluated models. Additionally, we highlight the limitations of our study and the need for further validation and optimization. Our findings underscore the potential of machine learning to enhance the accuracy and safety of tracheal tube depth estimation in pediatric patients, contributing to improved patient outcomes in mechanical ventilation.

Results

We aimed to estimate the optimal tracheal tube depth (OTTD) in pediatric patients aged 0-7 years using machine learning models and compare their performance with formula-based models. The dataset consisted of 969 patients who underwent post-operative mechanical ventilation, and the patient attributes along with the OTTD determined by chest X-ray were extracted. Descriptive statistics of patient sex, age, height, weight, and OTTD are presented in Table 1. The mean OTTD was found to be 10.2 cm (SD=1.77

cm), emphasizing the importance of accurate estimation for safe mechanical ventilation.

Table 1: Descriptive Statistics of Patient's Sex, Age, Height, Weight and Optimal Tracheal Tube Depth.

	Sex	Age (years)	Height (cm)	Weight (kg)	Optimal Tube Depth (cm)
mean	0.539	0.758	66	7.13	10.2
std	0.499	1.44	19.1	4.77	1.77

Sex: Participant sex: 0 - Female, 1 - Male
Age (years): Patient age in years. Rounded to half years.
Height (cm): Patient height in centimeters.
Weight (kg): Patient weight in kilograms.
Optimal Tube Depth (cm): Optimal tracheal tube depth as determined by Chest X-ray (in cm).

To evaluate the performance of machine learning models and formula-based models in estimating OTTD, we employed four machine learning algorithms: Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network. Additionally, three formula-based models were considered: Height Formula, Age Formula, and ID Formula. The results, summarized in Table 2, unequivocally demonstrated the superiority of machine learning models over formula-based models in estimating OTTD.

Table 2: Performance comparison among Machine Learning models and formula-based models

	SR	p-val HF	p-val AF	p-val IDF
Random Forest	1.59	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
Elastic Net	1.91	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
Support Vector Machine	1.31	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
Neural Network	1.25	$<10^{-6}$	$<10^{-6}$	$<10^{-6}$
Height Formula	3.42	-	-	-
Age Formula	90.4	-	-	-
ID Formula	310	-	-	-

SR: Squared Residues of the model predictions.
p-val AF: p-value of the Age Formula-Based Model.
p-val HF: p-value of the Height Formula-Based Model.
p-val IDF: p-value of the ID Formula-Based Model.
ID Formula: OTTD (in cm) = 3 * (tube ID [mm])

First, the machine learning models consistently outperformed the formula-

based models, as indicated by lower squared residues. The Random Forest model achieved the lowest squared residue of 1.59, followed by Elastic Net (1.91), SVM (1.31), and Neural Network (1.25). In comparison, the formula-based models exhibited higher squared residues: Height Formula (3.42), Age Formula (90.4), and ID Formula (310). The p-values for all machine learning models were $<10^{-6}$ when compared to the formula-based models, providing statistically significant evidence of superior performance.

Our analysis included a comparative evaluation of the machine learning models. The Random Forest model, with the lowest squared residue, demonstrated the highest accuracy in estimating OTTD among the evaluated models. The performance comparison revealed that machine learning models considerably improve the accuracy of OTTD estimation compared to formula-based models. These results support the potential of machine learning techniques in enhancing patient care and safety.

In summary, our results confirm the enhanced performance of machine learning models in estimating OTTD compared to formula-based models. The Random Forest model exhibited the highest accuracy in predicting OTTD. These findings emphasize the potential of machine learning approaches to improve the accuracy and safety of tracheal tube depth estimation in pediatric patients aged 0-7 years, which are crucial for safe and effective 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:

```python

import pandas as pd
import numpy as np

# Load the dataset
df = pd.read_csv('tracheal_tube_insertion.csv')

# Open file to save output
with open('data_exploration.txt', 'w') as file:

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

    # Summary statistics
    file.write('# Summary Statistics\n')
    summary_stats = df.describe(include='all')
    file.write(f'{summary_stats}\n\n')

    # Categorical variables
    file.write('# Categorical Variables\n')
    categorical_cols =
    ↪  df.select_dtypes(include=['object','bool']).columns
```

```
    if len(categorical_cols) == 0:
        file.write("There are no categorical variables in the dataset.\n")
    else:
        for cat in categorical_cols:
            mode = df[cat].mode().values[0]
            file.write(f'Most common value in {cat}: {mode}\n')
    file.write('\n')

    # Missing values
    file.write('# Missing Values\n')
    missing_values = df.isnull().sum()
    if missing_values.sum() == 0:
        file.write("There are no missing values in the dataset.\n")
    else:
        file.write(f'{missing_values}\n')

# close the file
file.close()
```

B.2 Code Description

The provided code performs data exploration on the given dataset of pediatric patients who received post-operative mechanical ventilation.

The code starts by loading the dataset into a pandas DataFrame.

Next, it writes the following analyses into the "data_exploration.txt" file:

1. Data Size: It writes the number of rows and columns present in the dataset.

2. Summary Statistics: It calculates and writes the summary statistics for each column in the dataset. This includes count, unique values, top value, and frequency for categorical variables, and mean, standard deviation, minimum, maximum, and quartile information for numerical variables.

3. Categorical Variables: It identifies the categorical variables in the dataset and writes the most common value for each categorical variable. If there are no categorical variables, it writes a message stating that there are none.

4. Missing Values: It identifies and reports the number of missing values

for each column in the dataset. If there are no missing values, it writes a message stating that there are none.

The code saves all the analysis results in the "data_exploration.txt" file, providing a comprehensive overview of the dataset. These analyses help in understanding the data distribution, identifying any issues such as missing values, and provide initial insights into the dataset before further analysis or modeling is performed.

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
There are no categorical variables in the dataset.

# Missing Values
There are no missing values in the dataset.
```

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
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import ElasticNet
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_ind
import pickle
import warnings
warnings.filterwarnings('ignore')

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# Table 0: "Descriptive statistics of patient's sex, age,
↪  height, weight and Optimal Tracheal Tube Depth"
df0 = data[['sex', 'age_c', 'ht', 'wt',
↪  'tube_depth_G']].describe().loc[['mean', 'std']]
df0.to_pickle('table_0.pkl')

# PREPROCESSING
y = data['tube_depth_G']
X = data.drop(columns=['tube', 'tube_depth_G'])
X_train, X_test, y_train, y_test = train_test_split(X, y,
↪  test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# ANALYSIS
```

```
# Table 1: "Performance comparison among Machine Learning models and formula-based models"
ml_models = {
    'Random Forest': RandomForestRegressor(),
    'Elastic Net': ElasticNet(),
    'Support Vector Machine': SVR(),
    'Neural Network': MLPRegressor(max_iter=1000)
}

ml_results = {}
ml_preds = {}

for model_name, model in ml_models.items():
    model.fit(X_train_scaled, y_train)
    predictions = model.predict(X_test_scaled)
    squared_residues = mean_squared_error(y_test, predictions)
    ml_results[model_name] = squared_residues
    ml_preds[model_name] = predictions

formula_based_results = {
    'Height Formula': mean_squared_error(y_test, X_test['ht'] / 10 + 5),
    'Age Formula': mean_squared_error(y_test, X_test['age_c']),
    'ID Formula': mean_squared_error(y_test,  X_test['wt'] * 3)
}

formula_preds = {
    'Height Formula': X_test['ht'] / 10 + 5,
    'Age Formula': X_test['age_c'],
    'ID Formula': X_test['wt'] * 3
}

ml_results.update(formula_based_results)

df1 = pd.DataFrame.from_dict(ml_results, orient='index', columns=['Squared Residues'])
for ml_method, pred in ml_preds.items():
    for formula_method, formula_pred in formula_preds.items():
```

```
        ttest_res = ttest_ind(pred, formula_pred)
        df1.loc[ml_method, f'p-value vs {formula_method}'] =
        ↪  ttest_res.pvalue

df1 = df1.fillna('-')
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': len(data),
 'Total number of test observations': len(X_test)
}

with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)
```

C.2 Code Description

The code performs an analysis on a dataset of pediatric patients who received post-operative mechanical ventilation after surgery. The goal is to determine the optimal tracheal tube depth (OTTD) for these patients in a non-invasive and efficient manner.

The code begins by importing the necessary libraries for data analysis and model training. It then loads the dataset which contains information about patient sex, age, height, weight, and the OTTD determined by chest X-ray.

After loading the dataset, the code prepares the data for analysis. It splits the data into feature variables (X) and the target variable (y), and then splits the data into training and testing sets. The features are scaled using standardization to ensure comparable ranges.

Next, the code performs the analysis using different machine learning models and formula-based methods. The machine learning models include Elastic Net, Random Forest, Support Vector Machine, and Neural Network. The code fits each model on the training data and evaluates their performance by predicting the OTTD on the test data. The mean squared error is calculated as a measure of performance.

Additionally, the code includes formula-based methods that estimate the OTTD based on patient height, age, and weight. The mean squared error is calculated for each formula-based method. The code then compares

the performance of each machine learning model with the formula-based methods using a t-test to determine if there is a significant difference in performance.

The code saves the results of the analysis in two pickle files. "table_0.pkl" contains descriptive statistics of the patient's sex, age, height, weight, and the OTTD. "table_1.pkl" includes the performance comparison among the machine learning models and formula-based methods, including the squared residues and the p-values from the t-test.

Finally, the code saves additional results in the "additional_results.pkl" file. This includes the total number of observations in the dataset and the total number of observations used for testing.

Overall, the code performs a comprehensive analysis to determine the optimal tracheal tube depth using machine learning models and formula-based methods, providing insights into the performance of different approaches in a non-invasive manner.

C.3 Code Output

table_0.pkl

	sex	age_c	ht	wt	tube_depth_G
mean	0.538700	0.757998	66.000516	7.126687	10.189474
std	0.498758	1.440271	19.081267	4.774186	1.766052

table_1.pkl

	Squared Residues	p-value vs Height Formula	p-value vs Age Formula	p-value vs ID Formula
Random Forest	1.590379	3.739e-14	5.828e-200	1.109e-22
Elastic Net	1.913334	8.892e-19	3.513e-238	8.066e-23
Support Vector Machine	1.312018	1.055e-13	1.306e-205	1.636e-22
Neural Network	1.246383	2.692e-14	4.455e-205	1.259e-22
Height Formula	3.418890	-	-	-
Age Formula	90.393557	-	-	-

```
ID Formula                    309.559605                    -
    -                    -
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Total number of test observations': 194,
}
```

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, AbbrToNameDef

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
 'sex': ('Sex', 'Participant sex: 0 - Female, 1 - Male'),
 'age_c': ('Age (years)', 'Patient age in years. Rounded to
 ↪  half years.'),
 'ht': ('Height (cm)', 'Patient height in centimeters.'),
 'wt': ('Weight (kg)', 'Patient weight in kilograms.'),
 'tube_depth_G': ('Optimal Tube Depth (cm)', 'Optimal tracheal
 ↪  tube depth as determined by Chest X-ray (in cm).')
}

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

# RENAME ROWS AND COLUMNS
mapping: AbbrToNameDef = {k: v for k, v in
 ↪  shared_mapping.items() if is_str_in_df(df, k)}
```

```python

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's Sex, Age, Height, Weight and Optimal Tracheal Tube Depth.",
 label='table:descriptive_statistics',
 legend=legend)

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

# Format the DataFrame and p-values
df.rename(columns={col: col.replace('_', ' ') for col in df.columns}, inplace=True)
df.rename(columns={
    'Squared Residues': 'SR',
    'p-value vs Height Formula': 'p-val HF',
    'p-value vs Age Formula': 'p-val AF',
    'p-value vs ID Formula': 'p-val IDF'
    }, inplace=True)

for col in df.columns:
    if 'p-val' in col:
        df[col] = df[col].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping: AbbrToNameDef = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
mapping |= {
    'SR': ('SR', 'Squared Residues of the model predictions.'),
    'p-val AF': ('p-val AF', 'p-value of the Age Formula-Based Model.'),
    'p-val HF': ('p-val HF', 'p-value of the Height Formula-Based Model.'),
```

```
    'p-val IDF': ('p-val IDF', 'p-value of the ID Formula-Based Model.'),
    'ID Formula': ('ID Formula', 'OTTD (in cm) = 3 * (tube ID [mm])')
}

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

# Save as latex
to_latex_with_note(
 df, 'table_1.tex',
 caption="Performance comparison among Machine Learning models and formula-based models",
 label='table:performance_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.
 """
```

```python

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 Patient's Sex, Age, Height, Weight and
    Optimal Tracheal Tube Depth.}
\label{table:descriptive_statistics}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrrrr}
\toprule
 & Sex & Age (years) & Height (cm) & Weight (kg) & Optimal Tube Depth (cm) \\
\midrule
\textbf{mean} & 0.539 & 0.758 & 66 & 7.13 & 10.2 \\
\textbf{std} & 0.499 & 1.44 & 19.1 & 4.77 & 1.77 \\
```

```
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Sex}: Participant sex: 0 - Female, 1 - Male
\item \textbf{Age (years)}: Patient age in years. Rounded to half years.
\item \textbf{Height (cm)}: Patient height in centimeters.
\item \textbf{Weight (kg)}: Patient weight in kilograms.
\item \textbf{Optimal Tube Depth (cm)}: Optimal tracheal tube depth as
    determined by Chest X-ray (in cm).
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Performance comparison among Machine Learning models and formula-based
    models}
\label{table:performance_comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrlll}
\toprule
 & SR & p-val HF & p-val AF & p-val IDF \\
\midrule
\textbf{Random Forest} & 1.59 & $<$1e-06 & $<$1e-06 & $<$1e-06 \\
\textbf{Elastic Net} & 1.91 & $<$1e-06 & $<$1e-06 & $<$1e-06 \\
\textbf{Support Vector Machine} & 1.31 & $<$1e-06 & $<$1e-06 & $<$1e-06 \\
\textbf{Neural Network} & 1.25 & $<$1e-06 & $<$1e-06 & $<$1e-06 \\
\textbf{Height Formula} & 3.42 & - & - & - \\
\textbf{Age Formula} & 90.4 & - & - & - \\
\textbf{ID Formula} & 310 & - & - & - \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{SR}: Squared Residues of the model predictions.
```

```
\item \textbf{p-val AF}: p-value of the Age Formula-Based Model.
\item \textbf{p-val HF}: p-value of the Height Formula-Based Model.
\item \textbf{p-val IDF}: p-value of the ID Formula-Based Model.
\item \textbf{ID Formula}: OTTD (in cm) = 3 * (tube ID [mm])
\end{tablenotes}
\end{threeparttable}
\end{table}
```