Optimal Tracheal Tube Depth Prediction in Pediatric Patients using Machine Learning

Data to Paper

January 9, 2024

Abstract

Mechanical ventilation in pediatric patients requires accurate positioning of the tracheal tube. However, determining the Optimal Tracheal Tube Depth (OTTD) is challenging and time-consuming. This study addresses the research gap by developing a machine learning approach to predict the OTTD in pediatric patients. We analyzed a dataset of 969 patients aged 0-7 years who underwent post-operative mechanical ventilation. By comparing the performance with a formula-based model, our Random Forest machine learning model achieved a significantly lower mean squared error (MSE) of 1.04. The analysis of residuals further validated the superior prediction accuracy of our model. These findings have important implications for improving patient outcomes and provide a reliable and efficient method for tracheal tube placement in pediatric patients.

Results

In this section, we present the results of our analysis, which aimed to develop a machine learning approach to predict the Optimal Tracheal Tube Depth (OTTD) in pediatric patients. We conducted multiple analyses comparing the performance of the machine-learning model with a formula-based model using different metrics.

We analyzed the descriptive statistics of OTTD stratified by sex to understand any potential differences. The mean OTTD was 10.1 cm (SD: 1.65) for females and 10.3 cm (SD: 1.86) for males, as shown in Table 1. Although there is a slight difference in mean OTTD between females and males, the clinical significance of this difference requires further investigation.

To evaluate the prediction accuracy of our machine-learning model, we compared it with the formula-based model using mean squared error (MSE).

Table 1: Descriptive statistics of Optimal Tracheal Tube Depth (OTTD) stratified by sex

sex	Mean OTTD	Standard Deviation
female	10.1	1.65
male	10.3	1.86

Mean OTTD: Mean Optimal Tracheal Tube Depth (cm)
Standard Deviation: Standard Deviation of OTTD (cm)

The machine-learning model, using the Random Forest algorithm, achieved a significantly lower MSE of 1.04 compared to the formula-based model with an MSE of 3.76, as shown in Table 2. This indicates that the machine-learning model provides more accurate predictions of OTTD compared to the formula-based model, which relies on patient features such as age and height.

Table 2: Performance comparison between Machine-Learning model and Formula-Based model

	model	Mean Squared Error
Random Forest Model	Random Forest	1.04
Formula-Based Model	Formula-Based	3.76

Mean Squared Error: Mean Squared Error of the model

We examined the residuals of the machine-learning model and the formula-based model to further validate their prediction accuracy. The machine-learning model had a mean residual of -0.000822, while the formula-based model had a mean residual of -1.41. The Wilcoxon test confirmed a statistically significant difference (p-value $< 10^{-6}$) in residuals between the two models, suggesting that the machine-learning model outperforms the formula-based model in predicting tracheal tube depth, as presented in Table 3. It is important to note that the mean residuals being close to zero does not necessarily indicate good prediction accuracy. Rather, it suggests that, on average, the model predictions are close to the actual OTTD values.

The analysis was conducted on a dataset of 969 pediatric patients who underwent post-operative mechanical ventilation. These results provide valuable insights into the prediction of OTTD in pediatric patients. Further validation and prospective studies are necessary to confirm and generalize

Table 3: Comparison of model residuals (prediction - target) between Random Forest model and Formula-Based model

	model	Mean Residual	P-value
Random Forest Model	Random Forest	-0.000822	$<10^{-6}$
Formula-Based Model	Formula-Based	-1.41	$<10^{-6}$

P-value: Wilcoxon test p-value of model residuals
Mean Residual: Mean Residual of the model

our findings. Additionally, the clinical implications of the difference in mean OTTD between males and females require deeper investigation.

In summary, our findings demonstrate that the machine-learning model, based on the Random Forest algorithm, outperforms the formula-based model in predicting the Optimal Tracheal Tube Depth in pediatric patients. The machine-learning model achieved a lower mean squared error and demonstrated superior prediction accuracy, as evidenced by the comparison of residuals in our analysis. These results have important implications for improving patient outcomes by providing a reliable and efficient method for tracheal tube placement in pediatric patients. Further research is needed to validate these findings and explore their clinical significance in larger cohorts of 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:

```
1
2  # Import necessary libraries
3  import pandas as pd
4  import numpy as np
5
6  # Load data
7  data = pd.read_csv('tracheal_tube_insertion.csv')
8
9  # Open file for writing
10 f = open('data_exploration.txt', 'w')
11
12 # Write size of data
13 f.write('# Data Size \n')
14 f.write('Number of Rows: {}\n'.format(data.shape[0]))
15 f.write('Number of Columns: {}\n'.format(data.shape[1]))
16 f.write('\n')
17
18 # Write summary statistics
19 f.write('# Summary Statistics \n')
20 desc = data.describe(include='all')
21 f.write('{}\n'.format(desc))
22 f.write('\n')
23
24 # Categorical Variables
```

```
f.write('# Categorical Variables \n')
categorical = data.select_dtypes(include=['object',
↪  'category'])
categorical_columns = categorical.columns.tolist()
for column in categorical_columns:
    f.write('Most Common Value in {}: {}\n'.format(column,
    ↪  data[column].mode()[0]))
f.write('\n')

# Check for missing values
f.write('# Missing Values \n')
missing_values = data.isna().sum()
f.write('{}\n'.format(missing_values))
f.write('\n')

# Close file
f.close()

```

B.2 Code Description

The code performs data exploration and writes the results into a file named "data_exploration.txt".

B.3 Data Loading

The code first loads the dataset from the "tracheal_tube_insertion.csv" file using the pandas library, assigning it to the variable "data".

B.4 Data Size

The code writes the size of the data into the file. It includes the number of rows and the number of columns in the dataset.

B.5 Summary Statistics

The code calculates the summary statistics of the dataset and writes them into the file. It provides descriptive statistics such as count, mean, standard deviation, minimum, quartiles, and maximum for all numerical columns in the dataset.

B.6 Categorical Variables

The code identifies the categorical variables in the dataset and writes the most common value for each categorical column into the file. This information helps to understand the distribution and prevalence of each category.

B.7 Missing Values

The code checks for missing values in the dataset and writes the number of missing values present in each column into the file. This information is important for determining the completeness of the dataset and possible strategies for handling missing data.

B.8 Output File

The code writes all the exploration results into the file "data_exploration.txt". The file includes the data size, summary statistics, most common values for categorical variables, and information about missing values. This file can be used to gain insights into the dataset and to assist in further analysis and modeling.

B.9 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
dtype: int64
```

C Data Analysis

C.1 Code

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

```
1
2  # IMPORT
3
4  import pandas as pd
5  import numpy as np
6  from scipy import stats
7  from statsmodels.stats.weightstats import DescrStatsW
8  from sklearn.ensemble import RandomForestRegressor
9  from sklearn.model_selection import GridSearchCV
10 from sklearn.metrics import mean_squared_error
11 import pickle
12
13 # LOAD DATA
14
15 data = pd.read_csv("tracheal_tube_insertion.csv")
16
17 # DATASET PREPARATIONS
18
19 # No dataset preparations are needed
20
21 # DESCRIPTIVE STATISTICS
22
```

```python
## Table 0: "Descriptive statistics of tracheal tube depth stratified by sex"

df0 = data.groupby("sex").tube_depth_G.agg(['mean', 'std'])
df0.index = df0.index.map({0: 'female', 1: 'male'})
df0.to_pickle('table_0.pkl')

# PREPROCESSING

# Create dummy variables for categorical variables (sex).
data["is_male"] = (data["sex"] == 1).astype(int)

# ANALYSIS

## Table 1: "Performance comparison between Machine-Learning model and Formula-Based model"

# Prepare for model training
X = data[["is_male", "age_c", "ht", "wt"]]
y = data["tube_depth_G"]

# Machine-Learning Model (Random Forest)
param_grid = {'n_estimators': [50, 100, 150], 'max_depth': [5, 10, 15]}
rf = RandomForestRegressor()
grid_search = GridSearchCV(estimator = rf, param_grid = param_grid)
grid_search.fit(X, y)
best_grid = grid_search.best_estimator_

# Prediction by Machine-Learning Model (Random Forest)
y_pred_rf = best_grid.predict(X)

# Formula-Based model
y_pred_fb = data.ht / 10 + 5

# Compare performance
mse_rf = mean_squared_error(y, y_pred_rf)
mse_fb = mean_squared_error(y, y_pred_fb)

```

```python
df1 = pd.DataFrame({
    "model": ["Random Forest", "Formula-Based"],
    "mse": [mse_rf, mse_fb]
}, index=['Model 1', 'Model 2'])

df1.to_pickle('table_1.pkl')

## Table 2: "Comparison of residuals between Random Forest model and Formula-Based model"

# Calculate residuals
residuals_rf = y - y_pred_rf
residuals_fb = y - y_pred_fb

# Perform paired Wilcoxon test
res_test = stats.wilcoxon(residuals_rf, residuals_fb)

df2 = pd.DataFrame({
    "model": ["Random Forest", "Formula-Based"],
    "mean_residual": [np.mean(residuals_rf), np.mean(residuals_fb)],
    "p_value": [res_test.pvalue] * 2
}, index=['Model 1', 'Model 2'])

df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS

additional_results = {
 'Total number of observations': len(data),
 'Best parameters for Random Forest model': best_grid.get_params(),
}

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

```

C.2 Code Description

The code performs a data analysis on a dataset of pediatric patients who underwent post-operative mechanical ventilation. The aim is to determine the Optimal Tracheal Tube Depth (OTTD) for these patients using two different approaches: a machine-learning model (Random Forest) and a formula-based model.

First, the code loads the dataset and performs some initial data preparations. Descriptive statistics of the tracheal tube depth stratified by sex are then calculated and stored in a pickle file named "table_0.pkl".

Next, the code preprocesses the data by creating dummy variables for the categorical variable 'sex'.

The code then proceeds to the analysis. It trains a Random Forest regressor using grid search for parameter tuning to predict the tracheal tube depth based on patient features: sex, age, height, and weight. The best hyperparameters are selected based on the mean squared error (MSE) as the evaluation metric.

The prediction results from the Random Forest model and a formula-based model (using a simple equation involving patient height) are compared in terms of MSE. The comparison is stored in a DataFrame named "df1" and saved as "table_1.pkl".

Additionally, the code calculates the residuals (the differences between the actual tracheal tube depth and the predicted values) for both models. The residuals are then subjected to a paired Wilcoxon test to compare the performance of the two models. The mean residuals and p-value from the Wilcoxon test are stored in a DataFrame named "df2" and saved as "table_2.pkl".

Finally, the code saves additional results, including the total number of observations in the dataset and the best parameters found for the Random Forest model, in a pickle file named "additional_results.pkl".

In summary, this code performs a data analysis to determine the Optimal Tracheal Tube Depth for pediatric patients using both a machine-learning model and a formula-based model. It compares the performance of these models and provides additional results for further analysis.

C.3 Code Output

table_0.pkl

	mean	std
sex		

```
female  10.062416  1.645478
male    10.298276  1.857778
```

table_1.pkl

```
                 model       mse
Model 1  Random Forest  1.040126
Model 2  Formula-Based  3.758860
```

table_2.pkl

```
                 model  mean_residual     p_value
Model 1  Random Forest      -0.000822  1.832e-159
Model 2  Formula-Based      -1.410578  1.832e-159
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Best parameters for Random Forest model': {'bootstrap': True, 'ccp_alpha':
    0.0, 'criterion': 'squared_error', 'max_depth': 5, 'max_features': 1.0,
    'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0,
    'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0,
    'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': 0,
    'verbose': 0, 'warm_start': False},
}
```

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

# PREPARATION FOR ALL TABLES
```

```python

shared_mapping: AbbrToNameDef = {
 'mean': ('Mean OTTD', 'Mean Optimal Tracheal Tube Depth (cm)'),
 'std': ('Standard Deviation', 'Standard Deviation of OTTD (cm)'),
 'mse': ('Mean Squared Error', 'Mean Squared Error of the model'),
 'p_value': ('P-value', 'Wilcoxon test p-value of model residuals'),
 'sex': ('Sex', 'Gender of patient 0: Female, 1: Male'),
 'age_c': ('Age', 'Age of patient in years'),
 'ht': ('Height', 'Height of patient in cm'),
 'wt': ('Weight', 'Weight of patient in kgs'),
 'tube_depth_G': ("Observed OTTD", "Optimal Tracheal Tube Depth as determined by chest X-ray (cm)"),
 'mean_residual': ('Mean Residual', 'Mean Residual of the model')
}

# TABLE 0:

df = pd.read_pickle('table_0.pkl')

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 Optimal Tracheal Tube Depth (OTTD) stratified by sex',
 label='table:statistics_by_sex',
 note=None,
 legend=legend)

# TABLE 1:

```

```python
df = pd.read_pickle('table_1.pkl')

model_mapping: AbbrToNameDef = {
 'Model 1': ('Random Forest Model', None),
 'Model 2': ('Formula-Based Model', None)
}

mapping = {k: v for k, v in shared_mapping.items() if
 ↪  is_str_in_df(df, k)}
mapping |= model_mapping
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 comparison between Machine-Learning
 ↪  model and Formula-Based model',
 label='table:performance_model',
 note=None,
 legend=legend)

# TABLE 2:

df = pd.read_pickle('table_2.pkl')

mapping = {k: v for k, v in shared_mapping.items() if
 ↪  is_str_in_df(df, k)}
mapping |= model_mapping
abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Format P-values
df['P-value'] = df['P-value'].apply(format_p_value)

# Save as latex:
to_latex_with_note(
 df, 'table_2.tex',
 caption='Comparison of model residuals (prediction - target)
 ↪  between Random Forest model and Formula-Based model',
```

```
    label='table:comparison_model',
    note=None,
    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 Optimal Tracheal Tube Depth (OTTD) stratified
    by sex}
\label{table:statistics_by_sex}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & Mean OTTD & Standard Deviation \\
sex &  &  \\
\midrule
\textbf{female} & 10.1 & 1.65 \\
\textbf{male} & 10.3 & 1.86 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Mean OTTD}: Mean Optimal Tracheal Tube Depth (cm)
\item \textbf{Standard Deviation}: Standard Deviation of OTTD (cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_1.tex

```
\begin{table}[h]
\caption{Performance comparison between Machine-Learning model and Formula-Based
    model}
\label{table:performance_model}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llr}
\toprule
 & model & Mean Squared Error \\
\midrule
\textbf{Random Forest Model} & Random Forest & 1.04 \\
\textbf{Formula-Based Model} & Formula-Based & 3.76 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Mean Squared Error}: Mean Squared Error of the model
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Comparison of model residuals (prediction - target) between Random
    Forest model and Formula-Based model}
\label{table:comparison_model}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llrl}
\toprule
 & model & Mean Residual & P-value \\
\midrule
\textbf{Random Forest Model} & Random Forest & -0.000822 & $<$1e-06 \\
\textbf{Formula-Based Model} & Formula-Based & -1.41 & $<$1e-06 \\
```

```
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{P-value}: Wilcoxon test p-value of model residuals
\item \textbf{Mean Residual}: Mean Residual of the model
\end{tablenotes}
\end{threeparttable}
\end{table}
```