Improved Prediction of Optimal Tracheal Tube Depth in Pediatric Patients Using Machine Learning

Data to Paper

February 20, 2024

Abstract

Determining the optimal tracheal tube depth is crucial for ensuring the safety and efficacy of mechanical ventilation in pediatric patients. Misplacement of the tracheal tube can lead to severe complications, such as hypoxia and pneumothorax. However, the accuracy of current methods, including formula-based models, is limited. This study aimed to improve the prediction of optimal tracheal tube depth in pediatric patients using a machine learning approach. We conducted a comparison between a random forest regression model and a formula-based model, and the random forest model outperformed the formula-based model in estimating the optimal tracheal tube depth. The most influential factor for prediction was found to be weight. The findings highlight the potential of machine learning to enhance patient safety and prevent complications in pediatric critical care. By providing accurate predictions of optimal tracheal tube depth, this approach could significantly improve clinical practice. However, further validation with larger multi-center cohorts is needed to confirm and generalize these results. Our analysis contributes to the growing body of research focused on improving prediction models for pediatric critical care.

Results

To start with determining the optimal tracheal tube depth in pediatric patients, we conducted a comparison between the machine learning model and the formula-based model. The mean squared residuals obtained from these two models were calculated to assess their prediction accuracy. As shown in Table 1, the Random Forest model outperformed the formula-based model,

with a mean squared residual of 1.41 cm^2 compared to 3.42 cm^2. This significant improvement in prediction accuracy suggests the superiority of the machine learning approach for estimating the optimal tracheal tube depth in pediatric patients.

Table 1: Comparison of Mean Squared Residuals between Machine Learning Model and Formula-based Model

Method	Mean Squared Residual (cm squared)	Standard Error (cm)
Random Forest	1.41	0.229
Formula-based	3.42	0.319

Mean Squared Residual (cm squared): Average of squared differences between predicted and actual OTTD (in cm^2)
Standard Error (cm): Standard error of the mean squared residuals (in cm)

Next, we aimed to identify the most important features influencing the prediction of optimal tracheal tube depth. Our Random Forest model allowed for the assessment of feature importance, as presented in Table 2. The feature importance analysis revealed that weight was the most influential factor in predicting the optimal tracheal tube depth, with an importance value of 0.834. Height and age also exhibited some degree of importance, with values of 0.137 and 0.0227, respectively. The sex of the patient appeared to have the least influence on the prediction, as reflected by its importance value of 0.0062.

Table 2: Feature Importances in Random Forest Model

Feature	Importance
Sex	0.0062
Age	0.0227
Height	0.137
Weight	0.834

Sex: Participant gender, 0: Female, 1: Male
Age: Participant age in years, rounded to half years
Height: Participant height in cm
Weight: Participant weight in kg
Importance: Relative importance of each feature in the Random Forest model

To further validate our findings, we performed a paired t-test on the

squared residuals of the machine learning model and formula-based model predictions. The t-test results indicated a statistically significant difference (p-value$<1.405 \times 10^{-8}$) in the mean squared residuals between the two models. This confirms that our machine learning model provides a more accurate estimation of the optimal tracheal tube depth than the formula-based model.

In summary, the results obtained from our analysis demonstrate the superiority of the machine learning model over the formula-based model in predicting the optimal tracheal tube depth in pediatric patients. The feature importance analysis revealed weight as the most influential factor for prediction. These findings highlight the potential of machine learning approaches to enhance patient safety and prevent complications in pediatric critical care.

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  from scipy import stats
5
6  # Load data
7  df = pd.read_csv('tracheal_tube_insertion.csv')
8
9  # Open the output file
10 with open('data_exploration.txt', 'w') as file:
11
12     # Data Size
13     file.write('# Data Size\n')
14     file.write(f'Number of rows: {df.shape[0]}\n')
15     file.write(f'Number of columns: {df.shape[1]}\n\n')
16
17     # Summary Statistics
18     file.write('# Summary Statistics\n')
19     description = df.describe()
20     for column in description.columns:
21         file.write(f'{column}:\n')
22         file.write(f'min: {description[column]["min"]}\n')
23         file.write(f'25%: {description[column]["25%"]}\n')
24         file.write(f'50%: {description[column]["50%"]}\n')
```

```
        file.write(f'75%: {description[column]["75%"]}\n')
        file.write(f'max: {description[column]["max"]}\n\n')

    # Categorical Variables
    file.write('# Categorical Variables\n')
    categorical_columns =
    ↪  df.select_dtypes(include=['object']).columns
    for column in categorical_columns:
        most_common = df[column].value_counts().index[0]
        file.write(f'The most common value for {column} is:
        ↪  {most_common}\n\n')

    # Missing Values
    file.write('# Missing Values\n')
    missing_values = df.isna().sum()
    for column in missing_values.index:
        file.write(f'Number of missing values for {column}:
        ↪  {missing_values[column]}\n\n')
```

B.2 Code Description

The code performs a data exploration on the provided dataset using pandas and numpy libraries in Python. The main goal of this code is to analyze the dataset and extract useful information about its size, summary statistics, categorical variables, and missing values.

1. Data Size: - The code determines the number of rows and columns in the dataset and writes them to the output file.

2. Summary Statistics: - The code calculates the summary statistics (minimum, 25th percentile, median, 75th percentile, and maximum) for each numerical column in the dataset using the describe() function from pandas. - It iterates through each column and writes the summary statistics to the output file.

3. Categorical Variables: - The code identifies categorical columns in the dataset using the select_dtypes() function from pandas with the parameter 'include=['object']'. - For each categorical column, it determines the most common value by counting the frequency of unique values and selecting the most frequent value. - It then writes the name of the column along with its most common value to the output file.

4. Missing Values: - The code checks for missing values in each column of

the dataset using the isna().sum() function from pandas. - For each column, it counts the number of missing values and writes the column name along with the count to the output file.

The code writes the results of the data exploration to a text file named "data_exploration.txt". The file includes information about the size of the data, summary statistics for each numerical column, the most common values for categorical columns, and the number of missing values in each column. This file provides a comprehensive overview of the dataset, allowing researchers to understand its characteristics and make informed decisions regarding data preprocessing and analysis.

B.3 Code Output

data_exploration.txt

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

# Summary Statistics
tube:
min: 2.5
25%: 3.5
50%: 3.5
75%: 4.0
max: 6.0

sex:
min: 0.0
25%: 0.0
50%: 1.0
75%: 1.0
max: 1.0

age_c:
min: 0.0
25%: 0.0
50%: 0.0
75%: 1.0
max: 7.0
```

```
ht:
min: 29.5
25%: 52.0
50%: 61.5
75%: 76.0
max: 135.4

wt:
min: 0.57
25%: 3.56
50%: 5.7
75%: 9.5
max: 31.0

tube_depth_G:
min: 5.9
25%: 9.0
50%: 9.8
75%: 11.2
max: 19.2

# Categorical Variables
# Missing Values
Number of missing values for tube: 0

Number of missing values for sex: 0

Number of missing values for age_c: 0

Number of missing values for ht: 0

Number of missing values for wt: 0

Number of missing values for tube_depth_G: 0
```

C Data Analysis

C.1 Code

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

```

# IMPORT
import pandas as pd
import numpy as np
from scipy.stats import ttest_rel, sem
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import GridSearchCV
import pickle

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# No descriptive statistics table is needed.

# PREPROCESSING
# No preprocessing is needed, because our data is already in a format suitable for analysis.

# ANALYSIS
# Table 1: "Comparison of Mean Squared Residuals, with associated Standard Errors, between Machine Learning Model and Formula-based Model."

# Split the data into training and testing sets
X = df[['sex','age_c','ht','wt']]
y = df['tube_depth_G']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```
# Hyperparameter tuning for Random Forest Regressor using GridSearchCV
parameters = {'n_estimators':[50, 100, 150, 200], 'max_depth':[None, 5, 10, 15, 20]}
rf = RandomForestRegressor(random_state=42)
clf = GridSearchCV(rf, parameters)
clf.fit(X_train, y_train)
best_params = clf.best_params_

# Train the model using the best parameters
rf = RandomForestRegressor(n_estimators=best_params['n_estimators'], max_depth=best_params['max_depth'], random_state=42)
rf.fit(X_train, y_train)

# Predict OTTD using the trained model and calculate squared residuals
y_pred_rf = rf.predict(X_test)
squared_residuals_rf = (y_test - y_pred_rf)**2

# Calculate OTTD using formula-based model and calculate squared residuals
y_pred_formula = X_test['ht'] / 10 + 5
squared_residuals_formula = (y_test - y_pred_formula)**2

# Perform paired t-test on squared residuals and preserve the results as an object for further attribute access
t_test_results = ttest_rel(squared_residuals_rf, squared_residuals_formula)

# Calculate standard errors
se_rf = sem(squared_residuals_rf)
se_formula = sem(squared_residuals_formula)

# Prepare the dataframe for Table 1
df1 = pd.DataFrame({'Method': ['Random Forest', 'Formula-based'],
                    'Mean Squared Residual': [squared_residuals_rf.mean(), squared_residuals_formula.mean()],
```

```
                    'Standard Error': [se_rf, se_formula]
                    })
df1.set_index('Method', inplace=True)

# Save the dataframe
df1.to_pickle('table_1.pkl')

# Table 2: "Feature importances from the Random Forest model."

# Extract feature importances and prepare the dataframe for Table 2
feature_importances = rf.feature_importances_
df2 = pd.DataFrame({'Feature': X.columns,
                    'Importance': feature_importances
                   })
df2.set_index('Feature', inplace=True)

# Save the dataframe
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
# Save the best parameters of RF model and p-value of the t-test to additional_results.pkl
additional_results = {
 'Best Parameters of RF Model': best_params,
 'P-value of T-test': t_test_results.pvalue
}

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

C.2 Code Description

The provided code performs an analysis on the dataset of pediatric patients who received post-operative mechanical ventilation. The goal of the analysis is to compare the performance of a machine learning model (Random Forest) with a formula-based model in determining the optimal tracheal tube depth (OTTD) for these patients.

First, the code loads the dataset and splits it into training and testing

sets. Then, it uses the GridSearchCV function to find the best hyperparameters for the Random Forest Regressor model by performing a search over different combinations of the number of estimators and maximum depth. The best parameters obtained from this search are saved.

Next, the Random Forest model is trained using the best parameters on the training set. The model is then used to predict the OTTD for the testing set. The squared residuals (the squares of the differences between the predicted and actual OTTD values) are calculated for both the Random Forest model and the formula-based model, where the formula-based model predicts the OTTD based on the patient's height.

A paired t-test is performed on the squared residuals of the two models to compare their mean squared residuals. The p-value of the t-test is also calculated and saved. The mean squared residuals and their standard errors for both models are stored in a dataframe, which is then saved as "table_1.pkl".

Additionally, the feature importances of the Random Forest model are extracted and stored in a dataframe, which is then saved as "table_2.pkl". These feature importances indicate the relative importance of each feature (sex, age, height, weight) in predicting the OTTD.

Finally, the best parameters of the Random Forest model and the p-value of the t-test are saved in the "additional_results.pkl" file.

Overall, the code performs analysis steps including model training, prediction, evaluation, and comparison, and saves the results and performance metrics for further analysis and interpretation.

C.3 Code Output

table_1.pkl

Method	Mean Squared Residual	Standard Error
Random Forest	1.414799	0.229219
Formula-based	3.418890	0.319395

table_2.pkl

Feature	Importance
sex	0.006203
age_c	0.022666

```
ht          0.137397
wt          0.833734
```

additional_results.pkl

```
{
    'Best Parameters of RF Model': {'max_depth': 5, 'n_estimators': 150},
    'P-value of T-test': 1.405e-08,
}
```

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 Optional, Dict
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 gender, 0: Female, 1: Male'),
 'age_c': ('Age', 'Participant age in years, rounded to half years'),
 'ht': ('Height', 'Participant height in cm'),
 'wt': ('Weight', 'Participant weight in kg'),
}

# 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)}
mapping |= {
```

```
 'Mean Squared Residual': ('Mean Squared Residual (cm
 ↪ squared)', 'Average of squared differences between
 ↪ predicted and actual OTTD (in cm^2) '),
 'Standard Error': ('Standard Error (cm)', 'Standard error of
 ↪ the mean squared residuals (in cm)'),
}
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="Comparison of Mean Squared Residuals between Machine
 ↪ Learning Model and Formula-based Model",
 label='table:comparison',
 note=None,
 legend=legend)

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
 ↪ is_str_in_df(df, k)}
mapping |= {
 'Importance': ('Importance', 'Relative importance of each
 ↪ feature in the Random Forest model'),
}
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_2.tex',
 caption="Feature Importances in Random Forest Model",
 label='table:importance',
 note=None,
 legend=legend)


```

D.2 Provided Code

The code above is using the following provided functions:

```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}
```

```
26  return abbrs_to_names, names_to_definitions
27
```

D.3 Code Output

table_1.tex

```
\begin{table}[h]
\caption{Comparison of Mean Squared Residuals between Machine Learning Model and
    Formula-based Model}
\label{table:comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrr}
\toprule
 & Mean Squared Residual (cm squared) & Standard Error (cm) \\
Method &  &  \\
\midrule
\textbf{Random Forest} & 1.41 & 0.229 \\
\textbf{Formula-based} & 3.42 & 0.319 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Mean Squared Residual (cm squared)}: Average of squared
    differences between predicted and actual OTTD (in cm\textasciicircum{}2)
\item \textbf{Standard Error (cm)}: Standard error of the mean squared residuals
    (in cm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Feature Importances in Random Forest Model}
\label{table:importance}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
```

```
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Importance \\
Feature &  \\
\midrule
\textbf{Sex} & 0.0062 \\
\textbf{Age} & 0.0227 \\
\textbf{Height} & 0.137 \\
\textbf{Weight} & 0.834 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Sex}: Participant gender, 0: Female, 1: Male
\item \textbf{Age}: Participant age in years, rounded to half years
\item \textbf{Height}: Participant height in cm
\item \textbf{Weight}: Participant weight in kg
\item \textbf{Importance}: Relative importance of each feature in the Random
    Forest model
\end{tablenotes}
\end{threeparttable}
\end{table}
```