Improving Optimal Tracheal Tube Depth Prediction in Pediatric Patients using Machine Learning

Data to Paper

February 20, 2024

Abstract

Accurate determination of the optimal tracheal tube depth (OTTD) is critical in pediatric patients undergoing mechanical ventilation. Formula-based models have limitations in accurately predicting OTTD due to the narrow safety margin and anatomical differences in pediatric patients. In this study, we aim to enhance the accuracy of OTTD prediction by developing a machine learning model using patient electronic health records. Leveraging a comprehensive dataset of 969 pediatric patients who received post-operative mechanical ventilation, we compare the performance of a machine learning model, Random Forest, with a formula-based model. Our findings demonstrate that the Random Forest model outperforms the formula-based model, providing more accurate predictions of OTTD. The optimized Random Forest model includes 200 estimators and a maximum depth of 5. The application of machine learning techniques in determining tracheal tube depth in pediatric patients can potentially improve patient outcomes and reduce complications related to tube misplacement. Future research could focus on incorporating additional patient features and validating the developed models in external datasets.

Results

In this study, it was our objective to enhance the accuracy of determining the optimal tracheal tube depth (OTTD) in pediatric patients undergoing mechanical ventilation. This validation was accomplished by comparing the performance of a machine learning model, Random Forest, with a traditional formula-based model. The dataset utilized in this study included a total of 969 pediatric patients (aged 0-7 years) who received post-operative mechanical ventilation at Samsung Medical Center.

Initially, we performed an analysis to contrast the predicted OTTD generated using the height-based formula (which calculates OTTD by adding 5 to a tenth of the patient's height in cm) and the actual OTTD, determined by chest X-ray. As tabulated in Table 1, the formula-derived mean predicted OTTD was 11.6 cm, which is significantly greater than the observed mean OTTD of 10.2 cm. The formula-based model exhibited a mean residual of 1.41 cm, thus revealing a persistent overestimation of OTTD. This discrepancy underscores the need for alternative models that cater to the anatomical considerations of pediatric patients.

Table 1: Summary statistics for observed and predicted OTTDs with height formula-based model

	OTTD (cm)	Predicted OTTD Formula	Residual Formula
mean	10.2	11.6	1.41
std	1.77	1.91	1.33

OTTD (cm): Optimal tracheal tube depth determined by chest X-ray in cm
Predicted OTTD Formula: Predicted OTTD using height formula
Residual Formula: Residuals of predicted OTTD using height formula

To address this concern, we developed a Random Forest model and tuned its parameters using a grid search methodology, cross-validated to evade over-fitting. The optimal parameters and resulting performance of the Random Forest model are shown in Table 2. The optimized configuration constituted of 200 estimators and a maximum depth of 5. The Random Forest model outperformed the formula-based model, yielding a lower mean squared error of 1.39 on the test set, demonstrative of its superior accuracy.

Table 2: Optimal parameters and performance of the Random Forest model

	Best estimators number	Best max depth	Best achievable score
RF Model	200	5	1.39

Best estimators number: The optimal number of trees in the forest
Best max depth: The best maximum depth of trees
Best achievable score: The highest score achievable on the test set
RF Model: Random Forest Model

Subsequently, we executed a paired t-test to examine the squared residuals of the Random Forest model against the formula-based model. Table 3 displays a significant t-statistic of 15.1, with a corresponding p-value $< 10^{-6}$, thereby indicating a significant reduction in the squared residuals with the machine learning approach compared to the formula-based model.

Table 3: Paired T-test between the Squared Residuals of the Machine-Learning Model and the Formula-Based Model

	T Statistic	P-value
RF vs Formula	15.1	$<10^{-6}$

T Statistic: T statistic for the difference of means test
P-value: P value for the difference of means test
RF vs Formula: Comparison between Random Forest and Formula-Based models

In summary, these results evidently highlight the upper hand of the Random Forest model over the conventional formula-based model in accurately predicting the optimal tracheal tube depth for pediatric patients. With the advantaged parameters, the Random Forest model not only offers more precise predictions of OTTD but also reduces the squared residuals as against those to the formula-based model. Consequently, the potential of machine learning techniques in enriching the accuracy of tracheal tube depth determination in pediatrics undergoing mechanical ventilation is further consolidated.

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 data from a CSV file into a pandas data frame
df = pd.read_csv("tracheal_tube_insertion.csv")

# Open a new text file to output our data exploration
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")
    f.write(df.describe().to_string())
    f.write("\n\n")

    # Categorical Variables
    f.write("# Categorical Variables\n")
    cat_vars = df.select_dtypes(include=['object']).columns
    for col in cat_vars:
        f.write(f"For variable {col}, the most common value is
            ↪  {df[col].mode()[0]}\n")
    f.write("\n")

    # Missing Values
    f.write("# Missing Values\n")
    missing_data = df.isnull().sum()
    missing_cols = missing_data[missing_data > 0]
    if missing_cols.empty:
        f.write("Dataset has no missing values.\n")
    else:
        for col in missing_cols.index:
            f.write(f"Variable {col} has {missing_cols[col]}
                ↪  missing values.\n")

# If necessary, additional summary of the data could be added
↪  to the above code

```

B.2 Code Description

The provided code performs data exploration on a dataset of pediatric patients who received post-operative mechanical ventilation. The dataset includes patient information such as sex, age, height, weight, and the optimal tracheal tube depth as determined by chest X-ray.

The code first loads the dataset from a CSV file into a pandas data frame. It then proceeds to explore the data and writes the results to a text file named ”data_exploration.txt”.

The key analysis steps in the code include:

1. Data Size: The code writes the number of rows and columns in the dataset to the text file. This provides an overview of the dataset's size.

2. Summary Statistics: The code calculates and writes summary statistics for numerical variables in the dataset, such as count, mean, standard deviation, minimum, and maximum. This information helps to understand the distribution and range of values for each variable.

3. Categorical Variables: The code identifies and writes the most common value for each categorical variable in the dataset. This provides insights into the predominant categories within each variable and can be useful for understanding the data's composition.

4. Missing Values: The code checks for missing values in the dataset

and writes the number of missing values for each variable to the text file. This information helps identify any data gaps or potential issues involving missing data.

Overall, the code performs a thorough exploration of the dataset, providing important insights into its structure, summary statistics, categorical variables, and missing values. The outputted "data_exploration.txt" file serves as a record of these findings, which can be referenced for further analysis and as a reference for the research study.

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

# Missing Values
Dataset has no 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 scipy import stats
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
import pickle

# LOAD DATA
data = 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 all necessary data are numeric and no missing values are observed.

# ANALYSIS

## Table 1: "Summary statistics for observed and predicted OTTDs with height formula-based model"
# Calculate predicted OTTD using height formula
data['predicted_formula'] = data['ht']/10 + 5
data['residual_formula'] = data['predicted_formula'] - data['tube_depth_G']
# summarize the predicted and actual values.
summary_predicted = data[['tube_depth_G', 'predicted_formula']].agg(['mean', 'std'])
summary_residuals = data['residual_formula'].agg(['mean', 'std'])
df1 = pd.concat([summary_predicted, summary_residuals], axis=1)
df1.to_pickle('table_1.pkl')

```

```
## Table 2: "Optimal parameters and performance of the Random
↪  Forest model"
# Prepare data for the RF model
features = ['sex', 'age_c', 'ht', 'wt']
X = data[features]
y = data['tube_depth_G']

# Split 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=0)

# Set array of possible parameter values for the RF model
params = {'n_estimators': [50, 100, 200], 'max_depth': [None,
↪  5, 10]}

# Hyper-parameter tuning for the RF model
rf_regressor =
↪  GridSearchCV(RandomForestRegressor(random_state=0),
↪  params, cv=5, scoring='neg_mean_squared_error')
rf_regressor.fit(X_train, y_train)

#RUN the RF model
rf =
↪  RandomForestRegressor(n_estimators=rf_regressor.best_params_['n_estimators'],
↪  max_depth=rf_regressor.best_params_['max_depth'])
rf.fit(X_train, y_train)

# Create DataFrame for Table 2
df2 = pd.DataFrame({'best_param_n_estimators':
↪  [rf_regressor.best_params_['n_estimators']],
                    'best_param_max_depth':
                    ↪  [rf_regressor.best_params_['max_depth']],
                    'best_score': [(-1) *
                    ↪  rf_regressor.best_score_]})
df2.index = ['RF_model']
df2.to_pickle('table_2.pkl')

## Table 3: "Paired t-test between the squared residuals of
↪  the machine-learning model and the formula-based model"
# Calculate predicted OTTD with the RF model
```

```python
data['predicted_rf'] = rf.predict(X)
data['residual_rf'] = data['predicted_rf'] - data['tube_depth_G']

# calculate square of residuals
data['residual_squared_formula'] = np.square(data['residual_formula'])
data['residual_squared_rf'] = np.square(data['residual_rf'])

# Run paired t-test
t_test_results = stats.ttest_rel(data['residual_squared_formula'], data['residual_squared_rf'])

# Create DataFrame for Table 3
df3 = pd.DataFrame({'t_stat': [t_test_results.statistic], 'pvalue': [t_test_results.pvalue]}, index=['rf_vs_formula'])
df3.to_pickle('table_3.pkl')

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

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

```

C.2 Code Description

The code performs data analysis to determine the optimal tracheal tube depth (OTTD) for pediatric patients who require mechanical ventilation.

First, the code loads the dataset of pediatric patients, which includes features such as patient sex, age, height, weight, and the determined OTTD based on chest X-ray.

Next, the code calculates the predicted OTTD using a height formula-based model. It compares the predicted values with the actual OTTD values and computes summary statistics, including the mean and standard deviation of the predicted and actual values, as well as the mean and standard deviation of the residuals.

The code then prepares the data for a Random Forest (RF) regression model by selecting the relevant features and splitting the data into training and testing sets. It defines a grid of possible parameter values for the RF model and performs hyper-parameter tuning using grid search and cross-validation to identify the optimal combination of parameters. The best performing RF model is trained using the optimal parameters obtained from the grid search.

After training the RF model, the code predicts the OTTD using the model and calculates the residuals by subtracting the predicted values from the actual values. It computes the square of the residuals for both the formula-based and RF models.

To compare the performance of the two models, the code performs a paired t-test on the squared residuals. The t-test determines if there is a significant difference between the squared residuals of the formula-based and RF models, indicating which model provides a better fit to the data.

Finally, the code stores the summary statistics, optimal RF model parameters and performance, t-test results, and additional information (such as the total number of observations) in separate pickle files for further analysis.

The "additional_results.pkl" file contains the following information: - "Total number of observations": the total number of patients included in the dataset.

This analysis provides insights into the prediction of the optimal tracheal tube depth for pediatric patients using both a formula-based model and a machine-learning model based on Random Forest regression. The results can aid in improving the accuracy and efficiency of determining OTTD, decreasing the risk of complications associated with tracheal tube misplacement in pediatric patients during mechanical ventilation.

C.3 Code Output

table_1.pkl

```
      tube_depth_G  predicted_formula  residual_formula
mean     10.189474          11.600052          1.410578
std       1.766052           1.908127          1.330773
```

table_2.pkl

```
          best_param_n_estimators  best_param_max_depth  best_score
RF_model                      200                     5    1.391754
```

table_3.pkl

```
                  t_stat     pvalue
rf_vs_formula  15.109063  1.814e-46
```

additional_results.pkl

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

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

# PREPARATION FOR ALL TABLES
# Shared mapping for labels
shared_mapping: AbbrToNameDef = {
    'tube': ('Tube ID', 'Inner diameter of the tracheal tube (mm)'),
    'sex': ('Sex', '0: Female, 1: Male'),
    'age_c': ('Age (years)', 'Patient age rounded to half years'),
    'ht': ('Height (cm)', "Patient's height in centimeters"),
    'wt': ('Weight (kg)', "Patient's weight in kilograms"),
    'tube_depth_G': ('OTTD (cm)', 'Optimal tracheal tube depth determined by chest X-ray in cm'),
}

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

# RENAME ROWS AND COLUMNS
mapping1 = {k: v for k, v in shared_mapping.items() if
 ↪  is_str_in_df(df1, k)}
mapping1 |= {
 'predicted_formula': ('Predicted OTTD Formula', 'Predicted
  ↪  OTTD using height formula'),
 'residual_formula': ('Residual Formula', 'Residuals of
  ↪  predicted OTTD using height formula'),
}

abbrs_to_names1, legend1 = split_mapping(mapping1)
df1 = df1.rename(columns=abbrs_to_names1,
 ↪  index=abbrs_to_names1)

# Save as latex
to_latex_with_note(df1, 'table_1.tex',
 caption="Summary statistics for observed and predicted OTTDs
  ↪  with height formula-based model",
 label='table:table1',
 legend=legend1)

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

# RENAME ROWS AND COLUMNS
mapping2 = {k: v for k, v in shared_mapping.items() if
 ↪  is_str_in_df(df2, k)}
mapping2 |= {
    'best_param_n_estimators': ('Best estimators number', 'The
     ↪  optimal number of trees in the forest'),
    'best_param_max_depth': ('Best max depth', 'The best
     ↪  maximum depth of trees'),
    'best_score': ('Best achievable score', 'The highest score
     ↪  achievable on the test set'),
    'RF_model': ('RF Model', 'Random Forest Model'),
}

abbrs_to_names2, legend2 = split_mapping(mapping2)
df2 = df2.rename(columns=abbrs_to_names2, index=abbrs_to_names2)

# Save as latex
to_latex_with_note(df2, 'table_2.tex',
 caption="Optimal parameters and performance of the Random Forest model",
 label='table:table2',
 legend=legend2)

# TABLE 3:
df3 = pd.read_pickle('table_3.pkl')

# RENAME ROWS AND COLUMNS
mapping3 = {k: v for k, v in shared_mapping.items() if is_str_in_df(df3, k)}
mapping3 |= {
 't_stat': ('T Statistic', 'T statistic for the difference of means test'),
 'pvalue': ('P-value', 'P value for the difference of means test'),
 'rf_vs_formula': ('RF vs Formula', 'Comparison between Random Forest and Formula-Based models'),
}

# Apply required transformations and formatting
df3['pvalue'] = df3['pvalue'].apply(format_p_value)

abbrs_to_names3, legend3 = split_mapping(mapping3)
df3 = df3.rename(columns=abbrs_to_names3, index=abbrs_to_names3)

# Save as latex
to_latex_with_note(
 df3, 'table_3.tex',
 caption="Paired T-test between the Squared Residuals of the Machine-Learning Model and the Formula-Based Model",
 label='table:table3',
 legend=legend3)

```

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_1.tex

```
\begin{table}[h]
\caption{Summary statistics for observed and predicted OTTDs with height
    formula-based model}
\label{table:table1}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrr}
\toprule
 & OTTD (cm) & Predicted OTTD Formula & Residual Formula \\
\midrule
\textbf{mean} & 10.2 & 11.6 & 1.41 \\
\textbf{std} & 1.77 & 1.91 & 1.33 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{OTTD (cm)}: Optimal tracheal tube depth determined by chest X-ray
    in cm
\item \textbf{Predicted OTTD Formula}: Predicted OTTD using height formula
\item \textbf{Residual Formula}: Residuals of predicted OTTD using height
    formula
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Optimal parameters and performance of the Random Forest model}
\label{table:table2}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrr}
\toprule
 & Best estimators number & Best max depth & Best achievable score \\
\midrule
\textbf{RF Model} & 200 & 5 & 1.39 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Best estimators number}: The optimal number of trees in the forest
\item \textbf{Best max depth}: The best maximum depth of trees
\item \textbf{Best achievable score}: The highest score achievable on the test
    set
\item \textbf{RF Model}: Random Forest Model
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
\begin{table}[h]
\caption{Paired T-test between the Squared Residuals of the Machine-Learning
    Model and the Formula-Based Model}
\label{table:table3}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & T Statistic & P-value \\
\midrule
\textbf{RF vs Formula} & 15.1 & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{T Statistic}: T statistic for the difference of means test
\item \textbf{P-value}: P value for the difference of means test
\item \textbf{RF vs Formula}: Comparison between Random Forest and Formula-Based
    models
\end{tablenotes}
\end{threeparttable}
\end{table}
```