# Optimal Tracheal Tube Depth in Pediatric Patients

Data to Paper

February 20, 2024

**Abstract**

Pediatric patients undergoing mechanical ventilation often suffer complications due to misplaced tracheal tube tips. Determining the Optimal Tracheal Tube Depth (OTTD) is crucial to minimize these complications. In this study, we present a comparative analysis of a Machine Learning Model and a Height-Formula Based Model to determine OTTD in pediatric patients. The current methods, relying on chest X-rays or formula-based models, have limitations. Our analysis, based on a dataset of pediatric patients aged 0-7 years who underwent post-operative mechanical ventilation, showed that the Machine Learning Model outperformed the Height-Formula Based Model, exhibiting lower mean squared residuals and higher accuracy. However, both models have limitations in accurately predicting OTTD. These findings highlight the need for further research to develop more accurate methods for determining OTTD in pediatric patients, aiming to enhance patient safety and reduce complications associated with misplaced tracheal tube tips.

## Results

We aimed to determine the optimal tracheal tube depth (OTTD) in pediatric patients undergoing mechanical ventilation post-surgery. We compared the predictive performance of a machine learning model with that of a height-formula based model in determining the OTTD.

We conducted a comparative analysis of the machine learning model and the height-formula based model using a dataset of 969 pediatric patients aged 0-7 years. The machine learning model, based on a Random Forest regression, performed significantly better than the height-formula based model, as indicated by the lower mean squared residuals (1.6 vs 3.42) (P-value $< 10^{-6}$). The machine learning model showed a mean squared residuals of 1.6 with a 95% confidence interval of (1.116, 2.081), while the

1

height-formula based model exhibited mean squared residuals of 3.42 with a 95% confidence interval of (2.793, 4.045) (Table 1).

Table 1: Comparison of predictive powers of Machine Learning Model and Formula-Based Model.

| model | Mean ResSq. | Std ResSq. | 95% CI ResSq. | P-value |
|---|---|---|---|---|
| **RF Machine Learning Model** | 1.6 | 3.43 | (1.116, 2.081) | $<10^{-6}$ |
| **Height-Formula Based Model** | 3.42 | 4.45 | (2.793, 4.045) | $<10^{-6}$ |

The Mean ResSq. shows the average squared residuals of the models. The lower value represents a better model. The 95% CI ResSq. is the 95% Confidence Interval for the Mean ResSq. The P-value shows the statistical significance of the results. A P-value of less than 0.05 typically indicates a statistically significant result.

**Mean ResSq.**: Mean of Squared residuals
**Std ResSq.**: Standard Deviation of Squared residuals
**95% CI ResSq.**: 95% Confidence Interval for the Mean of Squared residuals

The total number of observations in our dataset was 969, providing a robust sample size for our analysis. Additionally, the machine learning model achieved an accuracy of 0.536 in predicting the OTTD, indicating its ability to capture the variability in the target variable.

In summary, our comparative analysis demonstrated that the machine learning model outperformed the height-formula based model in predicting the optimal tracheal tube depth in pediatric patients. The machine learning model exhibited lower mean squared residuals and higher accuracy, indicating its superior predictive power. These findings have important implications in enhancing the accuracy and safety of determining tracheal tube depth in pediatric patients undergoing mechanical ventilation. Such improvements are critical for minimizing complications resulting from misplaced tracheal tube tips.

# 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
1
2  import pandas as pd
3
4  # Load dataset
5  data = pd.read_csv("tracheal_tube_insertion.csv")
6
7  # Create a text file
8  output_file = open("data_exploration.txt", "w")
9
10 # Data Size
11 output_file.write("# Data Size\n")
12 output_file.write(f"Number of rows: {data.shape[0]}\n")
13 output_file.write(f"Number of columns: {data.shape[1]}\n\n")
14
15 # Summary Statistics
16 output_file.write("# Summary Statistics\n")
17 summary_stats = data.describe()
18 output_file.write(summary_stats.to_string())
19 output_file.write("\n\n")
20
21 # Categorical Variables
22 output_file.write("# Categorical Variables\n")
23 categorical_vars = data.select_dtypes(include=['object'])
24 for column in categorical_vars:
```

4

```
25        output_file.write(f"For {column}, most common value:
      ↪   {data[column].mode()[0]}\n")
26  output_file.write("\n")
27
28  # Missing values
29  output_file.write("# Missing Values\n")
30  missing_values = data.isnull().sum()
31  output_file.write(missing_values.to_string())
32  output_file.write("\n\n")
33
34  output_file.close()
35
```

## B.2   Code Description

The provided code performs data exploration on the given dataset and writes
the results to a text file named "data_exploration.txt".

First, the code loads the dataset from the "tracheal_tube_insertion.csv"
file using the pandas library.

The code then proceeds to calculate and write the data size to the output
file. This includes the number of rows and columns in the dataset.

Next, the code computes and writes the summary statistics of the dataset
to the output file. These statistics include count, mean, standard deviation,
minimum, quartiles, and maximum values for each numerical column in the
dataset.

After that, the code identifies and writes the most common value for
each categorical column in the dataset to the output file.

The code also checks for missing values in the dataset and writes the
number of missing values for each column to the output file.

Overall, the code provides a comprehensive summary of the dataset,
including its size, summary statistics, most common values in categorical
variables, and information about missing values. This information can be
useful for gaining insights into the dataset, identifying any data quality issues
or patterns, and informing subsequent data analysis or modeling tasks.

## 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
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
1
2  # IMPORT
3  import pandas as pd
4  from sklearn.ensemble import RandomForestRegressor
5  from sklearn.model_selection import train_test_split
6  from sklearn.metrics import mean_squared_error
7  from scipy import stats
8  import pickle
9  import numpy as np
```

```
10
11   # LOAD DATA
12   df = pd.read_csv('tracheal_tube_insertion.csv')
13
14   # DATASET PREPARATIONS
15   # No dataset preparations are needed.
16
17   # DESCRIPTIVE STATISTICS
18   ## No descriptive statistics table is created, because
     ↪   characteristics of the data have been described in dataset
     ↪   output.
19
20   # PREPROCESSING
21   # No preprocessing is needed, because our data is already in
     ↪   numerical format.
22
23   # ANALYSIS
24   ## Table 1: "Comparison of predictive powers of RF Machine
     ↪   Learning Model and Height-Formula Based Model"
25
26   # Split data into training and test sets
27   train, test = train_test_split(df, test_size=0.2,
     ↪   random_state=42)
28
29   # Train RF model
30   rf = RandomForestRegressor(n_estimators=100, random_state=42)
31   rf.fit(train[['sex', 'age_c', 'ht', 'wt']],
     ↪   train['tube_depth_G'])
32
33   # Predict using RF model
34   test['rf_predictions'] = rf.predict(test[['sex', 'age_c',
     ↪   'ht', 'wt']])
35
36   # Predict using Height-Formula based model
37   test['formula_predictions'] = test['ht'] / 10 + 5
38
39   # Calculate squared residuals for each model
40   test['rf_squared_residuals'] = (test['tube_depth_G'] -
     ↪   test['rf_predictions']) ** 2
```

7

```python
41   test['formula_squared_residuals'] = (test['tube_depth_G'] -
     ↪  test['formula_predictions']) ** 2
42
43   # Carry out paired t-test
44   ttest_results = stats.ttest_rel(test['rf_squared_residuals'],
     ↪  test['formula_squared_residuals'])
45
46   df1 = pd.DataFrame({
47       'model': ['RF Machine Learning Model', 'Height-Formula
         ↪  Based Model'],
48       'mean_squared_residuals':
         ↪  [test['rf_squared_residuals'].mean(),
         ↪  test['formula_squared_residuals'].mean()],
49       'std_dev_squared_residuals':
         ↪  [test['rf_squared_residuals'].std(),
         ↪  test['formula_squared_residuals'].std()]
50   })
51   df1['95% CI_squared_residuals'] = df1.apply(lambda row:
     ↪  stats.norm.interval(0.95,
52
                                                                                ↪  loc=r
53
                                                                                ↪  scale
                                                                                ↪  axis=
54
55   df1['p_value'] = ttest_results.pvalue
56   df1.set_index('model', inplace=True)
57   df1.to_pickle('table_1.pkl')
58
59   # SAVE ADDITIONAL RESULTS
60   additional_results = {
61    'Total number of observations': df.shape[0],
62    'Accuracy of RF regression model': rf.score(test[['sex',
     ↪  'age_c', 'ht', 'wt']], test['tube_depth_G'])
63   }
64
65   with open('additional_results.pkl', 'wb') as f:
66    pickle.dump(additional_results, f)
67
```

8

## C.2  Code Description

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

First, the dataset is loaded and prepared for analysis.

Next, a machine learning model, specifically a Random Forest Regressor, is trained using the training set. The model is trained to predict the OTTD based on the patient's sex, age, height, and weight.

The trained model is then used to predict the OTTD for the test set of the data. Additionally, a height-formula based model is used to predict the OTTD for the same test set.

The squared residuals between the predicted OTTD and the actual OTTD values are calculated for both the machine learning model and the height-formula based model.

A paired t-test is performed to compare the squared residuals of the two models, evaluating their predictive power.

The mean squared residuals and standard deviations of the squared residuals for both models are calculated. Additionally, the 95% confidence interval for the mean squared residuals of each model is computed.

The results, including the mean squared residuals, standard deviations, 95% confidence intervals, and p-value from the t-test, are saved in a DataFrame. This DataFrame is then saved as a pickle file named "table_1.pkl".

Furthermore, additional results are calculated, including the total number of observations in the dataset and the accuracy of the machine learning model. These additional results are saved in a dictionary and written to a pickle file named "additional_results.pkl".

The code provides an analysis of the predictive power of the machine learning model compared to the height-formula based model, providing insights into the accuracy and effectiveness of each model in determining the optimal tracheal tube depth in pediatric patients.

## C.3  Code Output

**table_1.pkl**

```
                        mean_squared_residuals  std_dev_squared_residuals
    95% CI_squared_residuals    p_value
model
RF Machine Learning Model              1.598727                   3.430614
    (1.116          , 2.081          )  2.321e-07
```

9

```
Height-Formula Based Model              3.418890                4.448652
     (2.793          , 4.045       )  2.321e-07
```

**additional_results.pkl**

```
{
    'Total number of observations': 969,
    'Accuracy of RF regression model': 0.536         ,
}
```

# D   LaTeX Table Design

## D.1   Code

The LaTeX Table Design was carried out using the following custom code:

```
1
2  # IMPORT
3  import pandas as pd
4  from my_utils import to_latex_with_note, format_p_value,
   ↪  is_str_in_df, split_mapping
5  from typing import Dict, Any, Tuple, Optional
6
7  AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
8
9  # PREPARATION FOR ALL TABLES
10 shared_mapping: AbbrToNameDef = {
11     'mean_squared_residuals': ("Mean ResSq.", "Mean of Squared
       ↪  residuals"),
12     'std_dev_squared_residuals': ("Std ResSq.", "Standard
       ↪  Deviation of Squared residuals"),
13     '95% CI_squared_residuals': ('95% CI ResSq.', "95%
       ↪  Confidence Interval for the Mean of Squared
       ↪  residuals"),
14     'p_value': ('P-value', None)
15 }
16
17 # TABLE 1:
18 df1 = pd.read_pickle('table_1.pkl')
19
```

10

```
20   # FORMAT VALUES
21   df1['p_value'] = df1['p_value'].apply(format_p_value)
22
23   # RENAME ROWS AND COLUMNS
24   mapping = {k: v for k, v in shared_mapping.items() if
     ↪  is_str_in_df(df1, k)}
25
26   abbrs_to_names, names_to_definitions = split_mapping(mapping)
27
28   df1.rename(index=abbrs_to_names, columns=abbrs_to_names,
     ↪  inplace=True)
29
30   # Save as latex:
31   to_latex_with_note(
32       df1,
33       'table_1.tex',
34       caption='Comparison of predictive powers of Machine
         ↪  Learning Model and Formula-Based Model.',
35       label='table:comparison_of_predictive_powers',
36       note=
37       """
38       The Mean ResSq. shows the average squared residuals of the
     ↪  models.
39       The lower value represents a better model.
40       The 95% CI ResSq. is the 95% Confidence Interval for the
     ↪  Mean ResSq.
41       The P-value shows the statistical significance of the
     ↪  results. A P-value of less than 0.05 typically indicates a
     ↪  statistically significant result.
42       """,
43       legend=names_to_definitions)
44
```

## 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}
    return abbrs_to_names, names_to_definitions
```

## D.3 Code Output

**table_1.tex**

```
\begin{table}[h]
\caption{Comparison of predictive powers of Machine Learning Model and Formula-
    Based Model.}
\label{table:comparison_of_predictive_powers}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrrll}
\toprule
 & Mean ResSq. & Std ResSq. & 95\% CI ResSq. & P-value \\
model &  &  &  & \\
\midrule
\textbf{RF Machine Learning Model} & 1.6 & 3.43 & (1.116, 2.081) & $<$1e-06 \\
\textbf{Height-Formula Based Model} & 3.42 & 4.45 & (2.793, 4.045) & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item
    The Mean ResSq. shows the average squared residuals of the models.
    The lower value represents a better model.
    The 95\% CI ResSq. is the 95\% Confidence Interval for the Mean ResSq.
    The P-value shows the statistical significance of the results. A P-value of
    less than 0.05 typically indicates a statistically significant result.

\item \textbf{Mean ResSq.}: Mean of Squared residuals
\item \textbf{Std ResSq.}: Standard Deviation of Squared residuals
\item \textbf{95\% CI ResSq.}: 95\% Confidence Interval for the Mean of Squared
    residuals
\end{tablenotes}
\end{threeparttable}
\end{table}
```