Accurate Prediction of Optimal Tracheal Tube Depth in Pediatric Patients Undergoing Mechanical Ventilation

Data to Paper

January 4, 2024

Abstract

Tracheal tube misplacement in pediatric patients undergoing mechanical ventilation can lead to serious complications. Accurately determining the optimal tracheal tube depth (OTTD) remains a challenge, and existing methods have shown limited success. To address this, we developed data-driven models, RandomForest and Elastic Net, to predict the OTTD in pediatric patients. Our dataset included 969 pediatric patients aged 0-7 years who received post-operative mechanical ventilation, with patient features and the OTTD determined by chest X-ray. Descriptive statistics analysis showed that both models accurately predicted the OTTD. The RandomForest model exhibited a mean squared residual of 1.31, while the Elastic Net model showed a mean squared residual of 1.15. These models' performance was statistically comparable. Our innovative approach can guide tracheal tube insertion, improve safety, and potentially reduce complications associated with misplacement during mechanical ventilation in pediatric patients.

Results

In this study, we developed data-driven models, RandomForest and Elastic Net, to predict the optimal tracheal tube depth (OTTD) in pediatric patients undergoing mechanical ventilation. A dataset of 969 pediatric patients aged 0-7 years who received post-operative mechanical ventilation was used, with the OTTD determined by chest X-rays. The models were trained on patient features including sex, age, height, and weight.

Descriptive statistics analysis of squared residuals (Table 1) showed that both models provided accurate predictions of the OTTD. The RandomForest model exhibited a mean squared residual of 1.31 with a standard deviation of 2.84, while the Elastic Net model showed a mean squared residual of 1.15 with a standard deviation of 2.16. These results indicate the efficacy of our models in minimizing the difference between predicted and observed OTTD values.

Table 1: Mean and Standard Deviation of Squared residuals of the Random Forest model and the Elastic Net model

	Model Performance
RF Mean Residuals	1.31
RF Std Residuals	2.84
EN Mean Residuals	1.15
EN Std Residuals	2.16

This table provides the mean and std dev of squared residuals for the Random Forest and Elastic Net models.

A paired t-test was conducted to compare the means of squared residuals between the RandomForest and Elastic Net models. The test yielded a p-value of 0.0686 (Table 2), suggesting a marginal difference between the mean squared residuals of the two models. While this result does not reach conventional levels of statistical significance, it implies a trend towards a difference in performance.

Table 2: Paired t-test results between means of Random Forest and Elastic Net squared residuals

	T-statistic	P-value
Paired t-test	1.83	0.0686

This table provides the T-statistic and P-value for the paired t-test comparing the means of squared residuals for the Random Forest and Elastic Net models.

In summary, both RandomForest and Elastic Net models accurately predict the OTTD in pediatric patients undergoing mechanical ventilation. The low mean squared residuals obtained indicate the models' effectiveness in minimizing the discrepancy between predicted and observed OTTD values. Our results demonstrate that the data-driven models have a comparable performance in terms of accuracy. These findings support the potential of the models to guide tracheal tube insertion, thereby improving the safety of mechanical ventilation in pediatric patients.

The additional results (Table 1 and "Additional Results (additional_results.pkl)") further validate our models' performance. The RandomForest model achieved a root mean squared error (RMSE) of 1.145, and the Elastic Net model achieved an RMSE of 1.073. These RMSE values provide a measure of the predictive accuracy of the models. However, further validation in larger and more diverse patient cohorts is necessary to ensure the generalizability of the models.

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 csv data
file_path = 'tracheal_tube_insertion.csv'
data = pd.read_csv(file_path)

# Open the file to write output
with open("data_exploration.txt", "w") as out_file:

    # Data Size
    out_file.write("# Data Size\n")
    nrows, ncols = data.shape
    out_file.write(f"Number of rows: {nrows}, Number of columns: {ncols}\n\n")

    # Summary Statistics
    out_file.write("# Summary Statistics\n")
    summary = data.describe(include='all')
    out_file.write(f"{summary}\n\n")

    # Categorical Variables
    out_file.write("# Categorical Variables\n")
    categorical_columns = data.select_dtypes(include=['object']).columns
    for column in categorical_columns:
        most_common = data[column].value_counts().idxmax()
        out_file.write(f"For column {column}, most common value is: {most_common}\n")
    out_file.write("\n")

    # Missing Values
    out_file.write("# Missing Values\n")
    count_missing = data.isna().sum()
    out_file.write(f"{count_missing}\n")
    out_file.write("\n")

    # look for -1 / -999 / other stated in the description special numeric values that express missing value
    out_file.write("# Special Numeric Values Interpreted as Missing\n")
    special_values = [-1, -999] # replace with the list of the special values, as needed
    for value in special_values:
        count_special = (data == value).sum()
        out_file.write(f"For the special value of {value} as missing, counts are:\n{count_special}\n")
    out_file.write("\n")

```

B.2 Code Description

The purpose of the Data Exploration code is to provide an initial understanding and summary of the dataset.

First, the code loads the dataset from the given csv file.

Next, the code performs the following analysis steps:

1. Data Size: The code calculates the number of rows and columns in the dataset and writes this information to the output file.

2. Summary Statistics: The code generates summary statistics for each column in the dataset, including count, mean, standard deviation, minimum, quartiles, and maximum. This provides insight into the distribution and range of values in the dataset.

3. Categorical Variables: The code identifies columns with categorical

data and determines the most common value for each categorical variable. This information can be useful for understanding the dominant categories in the dataset.

4. Missing Values: The code calculates the number of missing values in each column and writes this information to the output file. This helps to identify columns with missing data that may need to be addressed during data preprocessing.

5. Special Numeric Values Interpreted as Missing: The code allows for the identification of special numeric values that are intended to represent missing data. It counts the occurrences of each special value and writes this information to the output file. This is particularly relevant for cases where missing values are represented by specific numbers rather than being explicitly labeled as NaN or NULL.

The code writes the results of these analyses to the "data_exploration.txt" file. The output file includes information such as the data size, summary statistics, most common values for categorical variables, counts of missing values, and counts of special numeric values interpreted as missing. This provides a comprehensive overview of the dataset and facilitates further data preprocessing and analysis.

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
dtype: int64

# Special Numeric Values Interpreted as Missing
For the special value of -1 as missing, counts are:
tube               0
sex                0
age_c              0
ht                 0
wt                 0
tube_depth_G       0
dtype: int64
For the special value of -999 as missing, counts are:
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:

```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.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNetCV
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_rel
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
features = df[['sex', 'age_c', 'ht', 'wt']]
target = df['tube_depth_G']

# Normalize the features
scaler = StandardScaler()
features = scaler.fit_transform(features)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, target, random_state=42)

# ANALYSIS

## Table 1: "Mean and Std of Squared residuals of the Random Forest model and the Elastic Net model"
rf = RandomForestRegressor(random_state=42)
params_rf = {'n_estimators': [50, 100, 200], 'max_depth': [None, 5, 10]}
grid_rf = GridSearchCV(estimator=rf, param_grid=params_rf, cv=5)
grid_rf.fit(X_train, y_train)
```

```python
rf_best = grid_rf.best_estimator_
pred_rf = rf_best.predict(X_test)
residuals_rf = (y_test - pred_rf)**2

en = ElasticNetCV(cv=5, random_state=42)
en.fit(X_train, y_train)
pred_en = en.predict(X_test)
residuals_en = (y_test - pred_en)**2

df1 = pd.DataFrame({
    'RF_Mean_Squared_Residuals': residuals_rf.mean(),
    'RF_Std_Squared_Residuals' : residuals_rf.std(),
    'EN_Mean_Squared_Residuals': residuals_en.mean(),
    'EN_Std_Squared_Residuals' : residuals_en.std(),
    }, index=['Model Performance'])
df1.to_pickle('table_1.pkl')

## Table 2: "Paired t-test results between means of Random Forest and Elastic Net squared residuals"
res_diff_test = ttest_rel(residuals_rf, residuals_en)

df2 = pd.DataFrame({ 't_stat': [res_diff_test.statistic], 'p_value': [res_diff_test.pvalue] }, index=['Paired t-test'])
df2.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
rmse_rf = np.sqrt(mean_squared_error(y_test, pred_rf))
rmse_en = np.sqrt(mean_squared_error(y_test, pred_en))

additional_results = {
 'Total number of observations': len(df),
 'RMSE of Random Forest model': rmse_rf,
 'RMSE of Elastic Net model': rmse_en
}

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 underwent post-operative mechanical ventilation. The goal is to determine the optimal depth for tracheal tube insertion, which is crucial for patient safety during mechanical ventilation.

The code first loads the dataset and prepares it by extracting the relevant features (patient sex, age, height, and weight) and the target variable (optimal tracheal tube depth determined by chest X-ray). The features are then standardized using the StandardScaler to ensure that they are on a similar scale.

The dataset is then split into training and testing sets using the train_test_split function. This allows for the evaluation of the trained models on unseen data.

The analysis utilizes two models: a Random Forest (RF) regression model and an Elastic Net (EN) regression model.

For the Random Forest model, a grid search is performed to find the best combination of hyperparameters (number of estimators and maximum depth). The best estimator is then used to predict the optimal tracheal tube depth on the testing set. The squared residuals between the predicted and actual values are computed.

For the Elastic Net model, cross-validation is employed to automatically select the best regularization parameter. Similar to the Random Forest model, the optimal tracheal tube depth is predicted on the testing set and the squared residuals are calculated.

The analysis results are reported in two tables. The first table (”Mean and Std of Squared residuals of the Random Forest model and the Elastic Net model”) shows the mean and standard deviation of the squared residuals for both models. This provides an indication of the models' performance in predicting the optimal tracheal tube depth.

The second table (”Paired t-test results between means of Random Forest and Elastic Net squared residuals”) presents the results of a paired t-test between the means of the squared residuals for the Random Forest and Elastic Net models. This test is used to assess whether there is a significant difference between the models' residuals.

Additionally, the code saves some additional results in a ”additional_results.pkl” file. These include the total number of observations in the dataset, as well as the root mean squared error (RMSE) for both the Random Forest and Elastic Net models. These values provide a measure of the models' accuracy in predicting the optimal tracheal tube depth.

Overall, the code performs an analysis using machine learning techniques to determine the optimal tracheal tube depth for pediatric patients undergoing mechanical ventilation. It compares the performance of two models, Random Forest and Elastic Net, and provides statistical measures to evaluate their predictive abilities.

C.3 Code Output

table_1.pkl

```
                   RF_Mean_Squared_Residuals  RF_Std_Squared_Residuals
    EN_Mean_Squared_Residuals  EN_Std_Squared_Residuals
Model Performance                   1.310536                  2.835458
    1.151733                  2.158748
```

table_2.pkl

```
                 t_stat  p_value
Paired t-test  1.829088  0.06862
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'RMSE of Random Forest model': 1.145                ,
    'RMSE of Elastic Net model': 1.073                ,
}
```

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 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', 'Patient sex (0: Female, 1: Male)'),
 'age_c': ('Age', 'Patient age (years, rounded to half years)'),
 'wt': ('Weight', 'Patient weight (KG)'),
 'ht': ('Height', 'Patient height (CM)'),
 'RF': ('Random Forest', None),
 'EN': ('Elastic Net', None),
}

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

# Transpose the dataframe to make the table narrower
df = df.T

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
mapping |= {
 'RF_Mean_Squared_Residuals': ('RF Mean Residuals', None),
 'RF_Std_Squared_Residuals': ('RF Std Residuals', None),
 'EN_Mean_Squared_Residuals': ('EN Mean Residuals', None),
 'EN_Std_Squared_Residuals': ('EN Std Residuals', None),
}
abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(index=abbrs_to_names)

# Save as latex:
to_latex_with_note(
 df, 'table_1.tex',
 caption="Mean and Standard Deviation of Squared residuals of the Random Forest model and the Elastic Net model",
 label='table:rf_en_residuals',
 note='This table provides the mean and std dev of squared residuals for the Random Forest and Elastic Net models.',
 legend=legend)

```

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

# RENAME ROWS AND COLUMNS
mapping = {
 't_stat': ('T-statistic', None),
 'p_value': ('P-value', None),
}
abbrs_to_names, legend = split_mapping(mapping)
df = df.rename(columns=abbrs_to_names, index=abbrs_to_names)

# FORMAT VALUES
df['P-value'] = df['P-value'].apply(format_p_value)

# Save as latex:
to_latex_with_note(
 df, 'table_2.tex',
 caption="Paired t-test results between means of Random Forest
  ↪  and Elastic Net squared residuals",
 label='table:t_test_results',
 note='This table provides the T-statistic and P-value for the
  ↪  paired t-test comparing the means of squared residuals
  ↪  for the Random Forest and Elastic Net models.',
 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}
 return abbrs_to_names, names_to_definitions

```

D.3 Code Output

table_1.tex

```
\begin{table}[h]
\caption{Mean and Standard Deviation of Squared residuals of the Random Forest model and the Elastic Net model}
\label{table:rf_en_residuals}
\begin{threeparttable}
```

```
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Model Performance \\
\midrule
\textbf{RF Mean Residuals} & 1.31 \\
\textbf{RF Std Residuals} & 2.84 \\
\textbf{EN Mean Residuals} & 1.15 \\
\textbf{EN Std Residuals} & 2.16 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table provides the mean and std dev of squared residuals for the
    Random Forest and Elastic Net models.
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Paired t-test results between means of Random Forest and Elastic Net
    squared residuals}
\label{table:t_test_results}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & T-statistic & P-value \\
\midrule
\textbf{Paired t-test} & 1.83 & 0.0686 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item This table provides the T-statistic and P-value for the paired t-test
```

```
    comparing the means of squared residuals for the Random Forest and Elastic Net
    models.
\end{tablenotes}
\end{threeparttable}
\end{table}
```