Improved Prediction of Optimal Tracheal Tube Depth in Pediatric Mechanical Ventilation

Data to Paper

January 10, 2024

Abstract

Determining the optimal tracheal tube depth (OTTD) is crucial in pediatric patients requiring mechanical ventilation. The misplacement of the tracheal tube tip can lead to severe complications, emphasizing the need for accurate depth determination methods. Current approaches, such as chest X-ray and formula-based models, have limitations in accuracy or efficiency. To address this gap, we applied machine learning models to predict OTTD based on patient features in a dataset of pediatric patients aged 0-7 years undergoing post-operative mechanical ventilation. Our Random Forest and Elastic Net models achieved root mean square errors of 1.2 cm and 1.11 cm, respectively, showing promising potential for accurate OTTD prediction. The comparison of these models also revealed a trend towards a slight difference in their performances. However, further validation on larger datasets is necessary to confirm their generalizability. By improving the accuracy and efficiency of determining OTTD, our findings have the potential to reduce complications and enhance patient outcomes in pediatric mechanical ventilation.

Results

In this section, we present the results of applying and comparing machine learning models to predict the optimal tracheal tube depth (OTTD) in pediatric patients requiring mechanical ventilation.

We first implemented and evaluated the performance of Random Forest (RF) and Elastic Net (EN) regression models. As shown in Table 1, the RF model, with hyperparameters 'max_depth' set to 10, 'min_samples_split' set to 10, and 'n_estimators' set to 100, produced a root mean square error (RMSE) of 1.2 cm. The EN model, with an 'alpha' of 0.1 and 'l1_ratio' of

0.2, displayed a slightly improved RMSE of 1.11 cm. Considering the range of the OTTD in our dataset is between 5.9 cm and 19.2 cm, these models exhibit promising potential for predicting OTTD with reasonable accuracy.

Table 1: Performance summary of two Machine-Learning models

Model	Root Mean Sq. Err.	Best Params
RF	1.2	'max_depth': 10, 'min_samples_split': 10, 'n_estimators': 100
EN	1.11	'alpha': 0.1, 'l1_ratio': 0.2

Target variable is OTTD, optimal tracheal tube depth, in cm.
Root Mean Sq. Err.: Root mean square error, in cm
Best Params: Hyperparameters that produced the smallest cross-validated RMSE
RF: Random Forest model
EN: Elastic Net model

To compare the performance of the RF and EN models and determine whether the difference in their RMSE is statistically significant, we used a paired t-test on the squared residuals of the two models. Table 2 presents the result of this test, where we found a t-statistic of 1.91 and a p-value of 0.0578. While this p-value does not meet the conventional threshold for statistical significance (0.05), this result suggests a trend toward a slight difference in the performance of the two models.

Table 2: Paired t-test results for squared residuals of RF and EN models

Statistic	Value
T-statistic	1.91
P-value	0.0578

Models compared are RF and EN. The alternative hypothesis is that the mean squared residual of the RF model is different from that of the EN model.
T-statistic: Statistic from a paired t-test of the squared residuals of the RF and EN models
P-value: Two-tailed P-value from the above t-test

In summary, the RF and EN models demonstrated their capability to accurately predict the OTTD in pediatric patients requiring mechanical ventilation with RMSEs of 1.2 cm and 1.11 cm, respectively. The paired t-test on the squared residuals of the two models indicated an observable trend in the difference of their performances, as evidenced by a nearly significant p-value. Further validation and comparison of these models on larger and

separate datasets are recommended to confirm their generalizability and robustness in predicting OTTD for pediatric mechanical ventilation.

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  import scipy as sp
5
6  # Load the data
7  data = pd.read_csv("tracheal_tube_insertion.csv")
8
9  # Open the file "data_exploration.txt" to write
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: {data.shape[0]}\n")
15     file.write(f"Number of columns: {data.shape[1]}\n\n")
16
17     # Summary statistics
18     file.write("# Summary Statistics\n")
19     summary_stats = data.describe().transpose()
20     summary_stats.to_string(file)  # Output summary stats to
       ↪  the file
21     file.write("\n\n")
22
23     # Categorical Variables
```

```
    file.write("# Categorical Variables\n")
    categorical_columns =
    ↪  data.select_dtypes(include=['object',
    ↪  'bool']).columns.tolist()
    for column in categorical_columns:
        file.write(f"For the variable '{column}', the most
        ↪  common value is: {data[column].mode()[0]}\n")
    if not categorical_columns:  # If there are no categorical
    ↪  columns
        file.write("There are no categorical variables in the
        ↪  dataset.\n")
    file.write("\n")

    # Missing Values
    file.write("# Missing Values\n")
    missing_values = data.isnull().sum()
    missing_values.to_string(file)  # Output missing values to
    ↪  the file
    if missing_values.sum() == 0:  # If there are no missing
    ↪  values
        file.write("There are no missing values in the
        ↪  dataset.\n")

```

B.2 Code Description

The provided code performs an initial data exploration on the "tracheal_tube_insertion.csv" dataset. This dataset contains information about pediatric patients who underwent surgery and received post-operative mechanical ventilation.

1. First, the code loads the dataset using the pandas library and assigns it to the variable "data".

2. Next, the code opens a file named "data_exploration.txt" to write the results of the data exploration.

3. The code then proceeds to perform several analysis steps:

a. Data Size: - The code determines the number of rows and columns in the dataset using the "shape" attribute of the pandas DataFrame. - It writes these values to the "data_exploration.txt" file.

b. Summary Statistics: - The code calculates summary statistics for the numerical variables in the dataset using the "describe" method. - It transforms the summary statistics into a transpose format using the

"transpose" method. - The code writes the summary statistics to the "data_exploration.txt" file.

c. Categorical Variables: - The code identifies categorical variables in the dataset by selecting columns with the data types "object" or "bool". - For each categorical variable, the code determines the most common value using the "mode" method. - The code writes the most common value for each categorical variable to the "data_exploration.txt" file. - If there are no categorical variables in the dataset, the code writes a message stating so.

d. Missing Values: - The code checks for missing values in the dataset using the "isnull" method. - It calculates the sum of missing values for each column. - The code writes the number of missing values for each column to the "data_exploration.txt" file. - If there are no missing values, the code writes a message stating so.

4. Finally, the code closes the "data_exploration.txt" file.

The "data_exploration.txt" file contains the following information: - Data size: the number of rows and columns in the dataset. - Summary statistics: descriptive statistics for the numerical variables in the dataset. - Categorical variables: the most common value for each categorical variable. - Missing values: the number of missing values for each column.

This data exploration process provides researchers with an initial understanding of the dataset, including its size, distribution of numerical variables, prevalence of categorical variables, and presence of missing values. This information can guide further analysis and data preprocessing steps in the research study.

B.3 Code Output

data_exploration.txt

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

# Summary Statistics
            count   mean    std  min  25%  50%  75%   max
tube          969  3.694 0.5681  2.5  3.5  3.5    4     6
sex           969 0.5387 0.4988    0    0    1    1     1
age_c         969  0.758   1.44    0    0    0    1     7
ht            969     66  19.08 29.5   52 61.5   76 135.4
wt            969  7.127  4.774 0.57 3.56  5.7  9.5    31
```

```
tube_depth_G    969  10.19  1.766  5.9    9  9.8 11.2  19.2

# Categorical Variables
There are no categorical variables in the dataset.

# Missing Values
tube            0
sex             0
age_c           0
ht              0
wt              0
tube_depth_G    0There are no missing values in the dataset.
```

C Data Analysis

C.1 Code

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

```
1
2  # IMPORT
3  import pandas as pd
4  import numpy as np
5  import pickle
6  from sklearn.model_selection import train_test_split,
   ↪  GridSearchCV
7  from sklearn.ensemble import RandomForestRegressor
8  from sklearn.linear_model import ElasticNet
9  from sklearn.metrics import mean_squared_error
10 from scipy import stats
11
12 # LOAD DATA
13 data = pd.read_csv("tracheal_tube_insertion.csv")
14
15 # DATASET PREPARATIONS
16 # No dataset preparations are needed.
17
18 # DESCRIPTIVE STATISTICS
19 # No descriptive statistics table is needed for this analysis.
```

```
20
21  # PREPROCESSING
22  # No preprocessing is needed, because data is already
    ↪  preprocessed and ready for analysis.
23
24  # ANALYSIS
25
26  # Table 1: "Performance summary of Random Forest and Elastic
    ↪  Net models"
27  features = data[["sex", "age_c", "ht", "wt"]]
28  target = data["tube_depth_G"]
29  X_train, X_test, y_train, y_test = train_test_split(features,
    ↪  target, test_size=0.2, random_state=42)
30
31  rf = RandomForestRegressor(random_state=42)
32  params_rf = {'n_estimators': [50, 100, 200], 'max_depth':
    ↪  [None, 10, 30], 'min_samples_split': [2, 5, 10]}
33  grid_rf = GridSearchCV(estimator=rf, param_grid=params_rf,
    ↪  cv=3)
34  grid_rf.fit(X_train, y_train)
35  rf_pred = grid_rf.predict(X_test)
36  rf_sq_resid = (rf_pred - y_test) ** 2
37
38  en = ElasticNet(random_state=42)
39  params_en = {'alpha': [0.1, 0.5, 1], 'l1_ratio': [0.2, 0.5,
    ↪  0.8]}
40  grid_en = GridSearchCV(estimator=en, param_grid=params_en,
    ↪  cv=3)
41  grid_en.fit(X_train, y_train)
42  en_pred = grid_en.predict(X_test)
43  en_sq_resid = (en_pred - y_test) ** 2
44
45  model_performace = pd.DataFrame({
46      "Model": ["Random Forest", "Elastic Net"],
47      "RMSE": [np.sqrt(mean_squared_error(y_test, rf_pred)),
        ↪  np.sqrt(mean_squared_error(y_test, en_pred))],
48      "Best Params": [str(grid_rf.best_params_),
        ↪  str(grid_en.best_params_)]
49  })
50  model_performace.set_index('Model', inplace=True)
```

```
model_performace.to_pickle('table_1.pkl')

# Table 2: "Paired t-test results for squared residuals of Random Forest and Elastic Net models"
t_test_results = stats.ttest_rel(rf_sq_resid, en_sq_resid)
t_test_df = pd.DataFrame({
    "Statistic": ["T-statistic", "p-value"],
    "Value": [t_test_results.statistic, t_test_results.pvalue]
})
t_test_df.set_index('Statistic', inplace=True)
t_test_df.to_pickle('table_2.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': data.shape[0],
    'Ideal tube_depth_G range': [data.tube_depth_G.min(), data.tube_depth_G.max()],
}

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

```

C.2 Code Description

The code performs an analysis to determine the optimal tracheal tube depth (OTTD) in pediatric patients who require mechanical ventilation. The analysis is conducted using two models: Random Forest and Elastic Net.

First, the code loads the dataset, which contains information about the patients' sex, age, height, weight, and the OTTD determined by chest X-ray.

Next, the code prepares the dataset by selecting the relevant features (sex, age_c, ht, wt) and the target variable (tube_depth_G). It then splits the dataset into training and testing sets.

For the Random Forest model, the code uses grid search cross-validation to find the optimal hyperparameters (number of estimators, maximum depth, and minimum samples split). It fits the model on the training data and makes predictions on the testing data. The root mean squared error (RMSE) is calculated to evaluate the model's performance. The best parameters and RMSE are stored in a table (Table 1).

Similarly, for the Elastic Net model, the code performs grid search cross-

validation to find the best values for the alpha and l1 ratio hyperparameters. It fits the model, makes predictions, and calculates the RMSE. The best parameters and RMSE are also stored in Table 1.

To compare the performance of the two models, the code conducts a paired t-test on the squared residuals (the difference between the predicted and actual OTTD values). The t-statistic and p-value are calculated and stored in a table (Table 2).

In addition to the model performance and t-test results, the code also saves additional results in the "additional_results.pkl" file. These additional results include the total number of observations in the dataset and the range of the ideal tube_depth_G values.

The code provides a comprehensive analysis of the relationship between patient characteristics and the optimal tracheal tube depth. It compares the performance of two models and reports their respective RMSE values. Furthermore, it conducts a statistical test to determine if there is a significant difference between the models in terms of the squared residuals. The results of the analysis can be used to improve the determination of OTTD in pediatric patients requiring mechanical ventilation.

C.3 Code Output

table_1.pkl

```
                RMSE
    Best Params
Model
Random Forest  1.203488  {'max_depth': 10, 'min_samples_split': 10,
    'n_estimators': 100}
Elastic Net    1.113168                                  {'alpha': 0.1,
    'l1_ratio': 0.2}
```

table_2.pkl

```
                Value
Statistic
T-statistic  1.908963
p-value       0.05775
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Ideal tube_depth_G range': [5.9, 19.2],
}
```

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 Dict, Any, Optional, Tuple
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 = {
    'RMSE': ('Root Mean Sq. Err.', 'Root mean square error, in
    ↪ cm'),
    'Best Params': (None, 'Hyperparameters that produced the
    ↪ smallest cross-validated RMSE')
}

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

# FORMAT VALUES
# No value formatting needed.

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if
↪ is_str_in_df(df, k)}
mapping |= {
    'Random Forest': ('RF', 'Random Forest model'),
    'Elastic Net': ('EN', 'Elastic Net 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_1.tex',
 caption="Performance summary of two Machine-Learning models",
 label='table:performance_summary',
 note="Target variable is OTTD, optimal tracheal tube depth, in cm.",
 legend=legend)

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

# FORMAT VALUES
# 'T-statistic' row is not a p-value, hence we should only apply format_p_value to the 'p-value' row.
df.loc['p-value', 'Value'] = format_p_value(df.loc['p-value', 'Value'])

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df, k)}
mapping |= {
    'T-statistic': (None, 'Statistic from a paired t-test of the squared residuals of the RF and EN models'),
    'p-value': ('P-value', 'Two-tailed P-value from the above t-test')
}
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="Paired t-test results for squared residuals of RF and EN models",
 label='table:t_test_results',
```

```
 note="Models compared are RF and EN. The alternative hypothesis is that the mean squared residual of the RF model is different from that of the EN model.",
 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_1.tex

```
\begin{table}[h]
\caption{Performance summary of two Machine-Learning models}
\label{table:performance_summary}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Root Mean Sq. Err. & Best Params \\
Model &  &  \\
\midrule
\textbf{RF} & 1.2 & {'max\_depth': 10, 'min\_samples\_split': 10,
    'n\_estimators': 100} \\
\textbf{EN} & 1.11 & {'alpha': 0.1, 'l1\_ratio': 0.2} \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Target variable is OTTD, optimal tracheal tube depth, in cm.
\item \textbf{Root Mean Sq. Err.}: Root mean square error, in cm
\item \textbf{Best Params}: Hyperparameters that produced the smallest cross-
    validated RMSE
\item \textbf{RF}: Random Forest model
\item \textbf{EN}: Elastic Net model
\end{tablenotes}
\end{threeparttable}
```

```
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Paired t-test results for squared residuals of RF and EN models}
\label{table:t_test_results}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{ll}
\toprule
 & Value \\
Statistic &  \\
\midrule
\textbf{T-statistic} & 1.91 \\
\textbf{P-value} & 0.0578 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Models compared are RF and EN. The alternative hypothesis is that the mean
    squared residual of the RF model is different from that of the EN model.
\item \textbf{T-statistic}: Statistic from a paired t-test of the squared
    residuals of the RF and EN models
\item \textbf{P-value}: Two-tailed P-value from the above t-test
\end{tablenotes}
\end{threeparttable}
\end{table}
```