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

Data to Paper

January 7, 2024

Abstract

Accurate determination of the optimal tracheal tube depth (OTTD) is critical for patient safety during mechanical ventilation in pediatric patients. However, existing formula-based models have limited success in predicting OTTD, leading to potential complications. In this study, we propose a novel machine learning-based approach to enhance the prediction of OTTD. Our analysis utilizes a dataset of 969 pediatric patients aged 0-7 years who underwent surgery, and compares the performance of machine learning models including Random Forest, Elastic Net, Support Vector Machine, and Neural Network with formula-based models. The results demonstrate the superiority of machine learning models, as they achieve significantly lower mean squared residuals. This study establishes the potential of machine learning in improving OTTD prediction accuracy and minimizing complications in pediatric mechanical ventilation. While considering the limitations of this single-center study, further validation on larger multi-center datasets is warranted, providing a foundation for leveraging machine learning to enhance patient outcomes in pediatric mechanical ventilation.

Results

To investigate the performance of machine learning (ML) models in predicting the optimal tracheal tube depth (OTTD) in pediatric patients, we conducted a comparative analysis using a dataset of 969 patients. First, we evaluated the predictive accuracy of ML models, including Random Forest, Elastic Net, Support Vector Machine (SVM), and Neural Network. The mean squared residuals for each ML model are summarized in Table 1. All

ML models achieved significantly lower mean squared residuals compared to the formula-based models, providing robust evidence for their superiority (t-test statistic = -3.68, p-value < 0.05). The Elastic Net model exhibited the lowest mean squared residuals (0.985), followed by the SVM model (1.03), Random Forest model (1.43), and Neural Network model (1.12). These findings highlight the potential of ML models in enhancing the accuracy of OTTD prediction beyond formula-based approaches.

Table 1: Comparison of mean squared residuals for each Machine Learning Models

	Mean Squared Residuals
Random Forest	1.43
Elastic Net	0.985
Support Vector Machine	1.03
Neural Network	1.12

Machine Learning models include Random Forest, Elastic Net, Support Vector Machine and Neural Network.

Next, we examined the performance of the formula-based models, which included the Height Model, ID Model, and Age Model. The mean squared residuals for these models are presented in Table 2. Among the formula-based models, the ID Model had the lowest mean squared residuals (2.51). However, even the best-performing formula-based model had higher mean squared residuals compared to all ML models. This emphasizes the superior predictive accuracy of ML models in OTTD prediction.

Table 2: Comparison of mean squared residuals for each Formula-Based Models

	Mean Squared Residuals
Height Model	3.76
ID Based Model	2.51
Age Model	2.05

Formula-Based models include Height Model, ID Model and Age Model.
ID Based Model: Optimal Tracheal Tube Depth (in cm) = 3 * Tube ID (in mm)

We further conducted a t-test to compare the mean squared residuals of the ML models and the formula-based models, as shown in Table 3. The t-test yielded a statistically significant difference between the two groups (t-statistic = -3.68, p-value = 0.0142). The negative t-statistic indicates that the ML models had significantly lower mean squared residuals compared to the formula-based models. These results provide additional evidence for the superiority of ML approaches in OTTD prediction.

Table 3: T-test comparing Machine Learning and Formula-based Models

	Statistic	P-value
Comparison of Machine Learning vs. Formula-based Models	-3.68	0.0142

Statistic and P-value are derived from the t-test.

In summary, our analysis demonstrated that ML models, including Random Forest, Elastic Net, SVM, and Neural Network, outperformed formula-based models in predicting OTTD in pediatric patients. The ML models achieved significantly lower mean squared residuals compared to the formula-based models, as supported by the t-test results. These findings highlight the potential of ML approaches to enhance OTTD prediction accuracy and improve patient outcomes in 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:

```python

# Import necessary packages
import pandas as pd
import numpy as np

# Load data
data = pd.read_csv("tracheal_tube_insertion.csv")

# Create a file for output
file = open('data_exploration.txt', 'w')

# Data Size
file.write('# Data Size\n')
file.write('Number of Patients : '+str(len(data))+'\n')
file.write('Number of Columns : '+str(len(data.columns))+'\n')
file.write('\n')

# Summary Statistics
file.write('# Summary Statistics\n')
desc = data.describe(include='all')
file.write(str(desc))
file.write('\n\n')

#Categorical Variables
```

```python
file.write('# Categorical Variables\n')
file.write('Sex Distribution : \n'+str(data['sex'].value_counts())+'\n')
file.write('\n')

# Missing Values
file.write('# Missing Values\n')
missing_values = data.isnull().sum()
file.write('Missing values per column: \n'+str(missing_values)+'\n')

#Close file
file.close()

```

B.2 Code Description

The code performs data exploration on the given dataset of pediatric patients who received post-operative mechanical ventilation. The main purpose of this code is to generate a text file, ”data_exploration.txt”, that provides insights into the dataset.

First, the code imports the necessary packages, including pandas and numpy, for data manipulation and analysis.

Then, it loads the dataset into a pandas DataFrame using the read_csv() function.

The code starts the analysis by providing information about the data size. It writes the number of patients and the number of columns in the dataset into the output file.

Next, the code computes summary statistics for the dataset using the describe() function. This includes measures such as count, mean, standard deviation, minimum, maximum, and quartiles for numerical variables, as well as the frequency of unique values for categorical variables. These summary statistics are written into the output file.

The code then examines the distribution of the categorical variable ’sex’, which represents the patient’s gender. It calculates the frequency of each gender category and writes it into the output file.

Finally, the code identifies missing values in the dataset using the isnull() function. It computes the count of missing values for each column and writes this information into the output file.

The ”data_exploration.txt” file serves as a record of the initial analysis of the dataset, providing a summary of the dataset's size, summary statistics, the distribution of the 'sex' variable, and the presence of missing values. This information is important for gaining insights into the dataset and planning subsequent data analysis steps.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of Patients : 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
Sex Distribution :
sex
1    522
0    447
Name: count, dtype: int64

# Missing Values
Missing values per column:
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.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from scipy.stats import ttest_ind
import warnings
import pickle

# Ignore known warnings
warnings.filterwarnings("ignore")

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

# DATASET PREPARATIONS
# No dataset preparations are needed.

# DESCRIPTIVE STATISTICS
# No descriptive statistics are needed.

# PREPROCESSING
# No preprocessing is needed, because the features are already numerical and have no missing values.

# ANALYSIS
features = df[['sex', 'age_c', 'ht', 'wt']]
target = df['tube_depth_G']

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=1)

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

models = {'Random Forest': RandomForestRegressor(),
          'Elastic Net': ElasticNet(),
          'Support Vector Machine': SVR(),
          'Neural Network': MLPRegressor()}

parameters = {'Random Forest': {'n_estimators': [100, 200, 300]},
              'Elastic Net': {'alpha': [0.1, 1, 10, 100]},
              'Support Vector Machine': {'C': [0.1, 1, 10]},
              'Neural Network': {'hidden_layer_sizes': [(50,), (50, 50)]}}

residuals_ML = []
model_names_ML = []

for model_name, model in models.items():
    grid_search = GridSearchCV(estimator=model, param_grid=parameters[model_name])
    grid_search.fit(X_train, y_train)
    y_pred = grid_search.predict(X_test)
    residuals_ML.append(np.mean((y_test - y_pred) ** 2))
    model_names_ML.append(model_name)

# Table 1: Comparison of mean squared residuals for each Machine Learning Models
df_ML = pd.DataFrame(residuals_ML, index=model_names_ML, columns=["Mean Squared Residuals"])
df_ML.to_pickle('table_1.pkl')

```

```python
ht_model_preds = df['ht'] / 10 + 5
id_model_preds = 3 * df['tube']
age_model_preds = df['age_c'].apply(lambda x: 9 if x < 0.5 else (10 if x < 1 else (11 if x < 2 else 12 + x*0.5)))

formula_models = [ht_model_preds, id_model_preds, age_model_preds]
formula_names = ['Height Model', 'ID Model', 'Age Model']

residuals_formula = []
for preds in formula_models:
    residuals_formula.append(np.mean((df['tube_depth_G'] - preds) ** 2))

# Table 2: Comparison of mean squared residuals for each Formula-Based Models
df_formula = pd.DataFrame(residuals_formula, index=formula_names, columns=["Mean Squared Residuals"])
df_formula.to_pickle('table_2.pkl')

# Perform t-test comparing mean squared residuals of the Machine Learning models to the residuals of our Formula-Based models
t_test_result = ttest_ind(residuals_ML, residuals_formula)

# Table 3: T-test comparing Machine Learning and Formula-based Models
df_ttest = pd.DataFrame({'Statistic': [t_test_result.statistic], 'P-value': [t_test_result.pvalue]}, index=['Comparison of Machine Learning vs. Formula-based Models'])
df_ttest.to_pickle('table_3.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
 'Total number of observations': df.shape[0]
}
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 received post-operative mechanical ventilation. The goal is to determine the optimal tracheal tube depth (OTTD) for these patients.

The code first loads the dataset and prepares it for analysis. It then splits the dataset into training and testing sets. The features of the dataset, which include patient sex, age, height, and weight, are standardized using a scaler.

Next, several machine learning models, namely RandomForestRegressor, ElasticNet, SVR, and MLPRegressor, are trained using the training set. Grid search cross-validation is used to find the best hyperparameters for each model. The trained models are then used to make predictions on the testing set, and the mean squared residuals are calculated.

In addition to the machine learning models, the code also includes three formula-based models to predict the tube depth based on patient features. The predictions from these models are compared to the actual tube depths, and the mean squared residuals are calculated.

The code generates three tables: *Table 1* compares the mean squared residuals for each machine learning model, *Table 2* compares the mean squared residuals for each formula-based model, and *Table 3* presents the results of a t-test comparing the mean squared residuals of the machine learning models to those of the formula-based models.

Finally, the code saves additional results, including the total number of observations, in a file named *additional_results.pkl*. This file can be further analyzed or used for reporting purposes.

Overall, the code performs a comprehensive analysis of the dataset using machine learning and formula-based models to determine the optimal tracheal tube depth for pediatric patients who require mechanical ventilation.

C.3 Code Output

table_1.pkl

```
                        Mean Squared Residuals
Random Forest                         1.433434
Elastic Net                           0.984989
Support Vector Machine                1.026052
Neural Network                        1.124440
```

table_2.pkl

```
             Mean Squared Residuals
Height Model               3.758860
ID Model                   2.508184
Age Model                  2.054923
```

table_3.pkl

```
                                                   Statistic  P-value
Comparison of Machine Learning vs. Formula-based Models  -3.684654  0.01422
```

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 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

# Define mappings common to all tables
shared_mapping: AbbrToNameDef = {
 'sex': ('Sex', '0: Female, 1: Male'),
 'age_c': ('Age', 'Patient Age in years'),
 'ht': ('Height', 'Patient height in cm'),
 'wt': ('Weight', 'Patient weight in kg'),
 'tube': ('Tube ID', 'Internal diameter of the tube in mm'),
 'tube_depth_G': ('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)}
abbrs_to_names, legend = split_mapping(mapping1)
df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex
to_latex_with_note(
 df1, 'table_1.tex',
 caption="Comparison of mean squared residuals for each Machine Learning Models",
 label='table:ml_models',
 note="Machine Learning models include Random Forest, Elastic Net, Support Vector Machine and Neural Network.",
 legend=legend)

# 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 |= { 'ID Model': ('ID Based Model', 'Optimal Tracheal Tube Depth (in cm) = 3 * Tube ID (in mm)')}
abbrs_to_names, legend = split_mapping(mapping2)
df2 = df2.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex
to_latex_with_note(
 df2, 'table_2.tex',
 caption="Comparison of mean squared residuals for each Formula-Based Models",
 label='table:formula_models',
 note="Formula-Based models include Height Model, ID Model and Age Model.",
 legend=legend)

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

# FORMAT VALUES
df3['P-value'] = df3['P-value'].apply(format_p_value)

# RENAME ROWS AND COLUMNS
mapping3 = {k: v for k, v in shared_mapping.items() if is_str_in_df(df3, k)}
abbrs_to_names, legend = split_mapping(mapping3)
df3 = df3.rename(columns=abbrs_to_names, index=abbrs_to_names)

# Save as latex
to_latex_with_note(
 df3, 'table_3.tex',
 caption="T-test comparing Machine Learning and Formula-based Models",
 label='table:comparison',
 note="Statistic and P-value are derived from the t-test.",
 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{Comparison of mean squared residuals for each Machine Learning Models}
\label{table:ml_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Mean Squared Residuals \\
\midrule
\textbf{Random Forest} & 1.43 \\
\textbf{Elastic Net} & 0.985 \\
\textbf{Support Vector Machine} & 1.03 \\
\textbf{Neural Network} & 1.12 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Machine Learning models include Random Forest, Elastic Net, Support Vector
    Machine and Neural Network.
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Comparison of mean squared residuals for each Formula-Based Models}
\label{table:formula_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & Mean Squared Residuals \\
\midrule
\textbf{Height Model} & 3.76 \\
\textbf{ID Based Model} & 2.51 \\
\textbf{Age Model} & 2.05 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Formula-Based models include Height Model, ID Model and Age Model.
\item \textbf{ID Based Model}: Optimal Tracheal Tube Depth (in cm) = 3 * Tube ID
```

```
    (in mm)
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_3.tex

```
\begin{table}[h]
\caption{T-test comparing Machine Learning and Formula-based Models}
\label{table:comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lrl}
\toprule
 & Statistic & P-value \\
\midrule
\textbf{Comparison of Machine Learning vs. Formula-based Models} & -3.68 &
    0.0142 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item Statistic and P-value are derived from the t-test.
\end{tablenotes}
\end{threeparttable}
\end{table}
```