

# Machine Learning Refines Pediatric Intubation by Predicting Optimal Tube Depth

data-to-paper

April 25, 2024

## Abstract

Precision in pediatric tracheal intubation is vital to patient safety, with inaccuracies risking critical outcomes. Current procedural reliance on conventional formula-based estimations results in notable rates of misplacement and associated complications. This study seeks to address the significant clinical need for enhanced intubation accuracy by exploiting machine learning (ML) methodologies. We target the crucial standard of optimal tracheal tube depth (OTTD), typically corroborated by radiographic assessment, with an intent to improve patient care while mitigating exposure to diagnostic radiation. Our research utilizes a curated dataset from pediatric surgical cases, inclusive of chest X-ray determined OTTDs and patient features, to train and validate a suite of ML models. The comparative analysis reveals ML models, especially Random Forest and Support Vector Machine, significantly outperform formula-based techniques in predicting OTTD, thereby evidencing their clinical applicability. Despite the models' efficacy, we recognize the constraints of an age-restricted dataset and single-center scope, which necessitate further multi-institutional validation to generalize these promising results. Ultimately, the integration of ML in OTTD prediction stands to refine pediatric intubation practices, potentially reducing the need for confirmatory radiography and aligning with endeavors for safer, evidence-supported medical care.

## Introduction

Accurate placement of the tracheal tube in pediatric patients is a critical concern in medical scenarios involving respiratory management. This precision is of paramount importance due to the shorter tracheal length in pediatric patients as compared to adults [1]. Even slight inaccuracies in determining the Optimal Tracheal Tube Depth (OTTD) can lead to severe

complications, such as hypoxia, atelectasis, hypercarbia, pneumothorax, and in extreme cases, death [2]. Despite the severity of outcomes related to incorrect OTTD, the prevailing methodology for determining the OTTD is reliant on chest X-rays, a process that both introduces delays and exposes patients to radiation [1, 2]. Moreover, formula-based methods used as an alternative, which include age and height of the patient, often demonstrate limited success [3, 4].

Recent advancements in healthcare have witnessed significant adoption of machine learning (ML) techniques for decision support and predictive modeling in a range of contexts [5]. In fact, studies such as [6, 7] indicate superior performance of ML models over traditional logistic regression models. However, the specific application of these machine learning models in predicting OTTD in pediatric patients remains an unexplored avenue. Bridging this research gap, our study focuses on creating and evaluating a series of machine learning models for predicting OTTD, offering a comparison against traditional, formula-based methods.

The dataset employed in our study is derived from pediatric surgical cases at Samsung Medical Center from 2015 to 2018, specifically focusing on patients aged 0 to 7 years requiring post-operative mechanical ventilation [8, 9, 10]. By constraining our analysis to such an age-specific, single-center dataset, we ensure a uniform patient demographic, thereby offering more reliable and relevant insights.

In terms of methodology, we utilize a suite of machine learning models encompassing Random Forest, Elastic Net, Support Vector Machine, Neural network, and traditional formula-based models [11, 12]. Each of these models undergoes a process of hyperparameter optimization aimed at maximizing its predictive accuracy. Model performance is measured by comparing mean squared residuals of their predicted OTTDs against the gold-standard OTTDs determined by chest X-rays.

Our results provide compelling evidence of the superior accuracy of machine learning models in predicting OTTD over conventional formula-based methods. This potential translates into a meaningful clinical impact, suggesting that the application of machine learning could significantly enhance the safety and efficacy of pediatric tracheal intubation procedures.

## Results

First, to determine the predictive accuracy of machine-learning (ML) models for predicting the Optimal Tracheal Tube Depth (OTTD) compared to con-

ventional formula-based methods, we compared mean squared error (MSE) values expressed in  $\text{cm}^2$ . Analyzing a dataset comprising a total of 969 pediatric patient records, with 775 used for training and 194 for testing, all ML models demonstrated a notable improvement in precision over traditional methods. The Random Forest model yielded an MSE of 1.41, significantly lower than the height-based formula, which displayed an MSE of 3.42, with a p-value of less than  $10^{-6}$ , as detailed in Table 1.

Table 1: Comparison of mean squared residuals of predicted OTTD by various models.

	ML MSE	Formula MSE	p-value
<b>Random Forest vs Height</b>	1.41	3.42	$<10^{-6}$
<b>Random Forest vs Age</b>	1.41	1.79	0.0214
<b>Random Forest vs ID</b>	1.41	2.52	$1.33 \cdot 10^{-5}$
<b>Elastic Net vs Height</b>	1.24	3.42	$<10^{-6}$
<b>Elastic Net vs Age</b>	1.24	1.79	0.000449
<b>Elastic Net vs ID</b>	1.24	2.52	$<10^{-6}$
<b>Support Vector Machine vs Height</b>	1.23	3.42	$<10^{-6}$
<b>Support Vector Machine vs Age</b>	1.23	1.79	0.000117
<b>Support Vector Machine vs ID</b>	1.23	2.52	$<10^{-6}$
<b>Neural Network vs Height</b>	1.36	3.42	$<10^{-6}$
<b>Neural Network vs Age</b>	1.36	1.79	0.0116
<b>Neural Network vs ID</b>	1.36	2.52	$2.42 \cdot 10^{-5}$

**ML MSE:** Mean Squared Error of Machine-Learning models

**Formula MSE:** Mean Squared Error of Formula-based models

**p-value:** p-value from the paired t-test comparing ML models with formula-based models

In further investigations of the best-performing ML models, after hyperparameter optimization, the Random Forest model exhibited an MSE of 1.405 with 'max\_depth' set to 5 and 'n\_estimators' to 50. The Support Vector Machine's most effective configuration achieved an MSE of 1.235 with the 'C' parameter set at 10. The Elastic Net model produced an MSE of 1.24 with 'alpha' at 0.1 and 'l1\_ratio' at 0. The Neural Network, implementing a 'tanh' activation function and 'hidden\_layer\_sizes' of 100, achieved an MSE of 1.355. These findings underscore the effects of hyperparameter optimization on enhancing model accuracy on the test dataset, and are further discussed in the context of Table 1.

To statistically confirm the improvement in prediction accuracy using ML models over formula-based methods, paired t-tests were conducted. The

Random Forest model exhibited significantly better performance than the age-based formula with a p-value less than 0.05, precisely 0.0214. This level of statistical significance is consistent across all ML models when compared with various formula-based predictions, affirming the ML models' enhanced capability for accurately predicting OTTD.

In summary, machine-learning algorithms have demonstrated a statistically significant edge over conventional formula-based techniques for predicting OTTD in pediatric patients. The consistently lower MSE values of the ML models suggest their potential application in clinical settings could contribute to improved safety and efficacy in pediatric tracheal intubation.

## Discussion

Our investigation utilized machine learning (ML) techniques to boost the precision in predicting the optimal tracheal tube depth (OTTD) for pediatric patients, an important factor for safe respiratory management [1, 2]. Presently, conventional estimations for determining OTTD either lean on chest X-ray examinations, which pose a delay and radiation exposure risk, or formula-based methods dependent on age and height parameters that often fail to assure optimal placements [3, 4].

Due to the successful adoption of ML models in various healthcare contexts [5], we found it valuable to scrutinize their potential for OTTD prediction. Using a pediatric surgical case dataset and employing a range of ML models, we compared the ML models' predictions against conventional formula-based methods. Along with model training and validation, an integral part of our methodologies was hyperparameter tuning, which dramatically enhanced the ML models' accuracy [7].

Upon analysis, irrespective of the individual strengths and weaknesses of different models, ML models as a group considerably outperformed the formula-based approach, consistent with findings from other studies [13, 14]. Notably optimized ML models like Random Forest and Support Vector Machine displayed particularly impressive results.

Despite our study's promising findings, we acknowledge certain limitations. The single-center, age-restricted nature of our dataset, obtained exclusively from the Samsung Medical Center, flags the potential for selection bias. Therefore, the generalizability of our findings across diverse pediatric populations might merit further exploration. Additionally, using chest X-ray-derived OTTD values as an undeniable standard introduces another layer of complexity, given inherent limitations with X-ray usage, primarily

radiation exposure. Future studies might benefit from integrating other clinically accepted OTTD assessment measures aligning with enhancing patient safety [15].

The application of ML models into the healthcare setting presents its own set of challenges. Their relative superiority in prediction does not negate the necessity for computational resources, which can be considerable in a real-world setting. Furthermore, ML models' black-box nature could hinder widespread acceptance amongst clinicians, underlining the crucial aspect of model interpretability in a clinical context.

In consideration of the limitations and implications of our study, broader multi-institutional studies could serve to validate our results. Engagement with various computational and visual tools may aid in making complex ML models more accessible and interpretable to clinicians. Moreover, future research could explore ML model training with additional patient parameters, potentially contributing towards a more robust, versatile predictive tool.

In conclusion, our research navigates the largely uncharted territory of using ML models for predicting OTTD in pediatric intubation [7]. We demonstrated that ML models, especially when optimized appropriately, significantly outperform traditional formula-based methods. By reducing the reliance on undesirable procedural factors such as unnecessary radiation exposure, these results have significant implications for clinical practice. As we tread forward, it's imperative to carry these promising, yet initial, findings through further research avenues to facilitate the safe and effective implementation of such models into routine care.

## Methods

### Data Source

The study leverages a dataset obtained from pediatric surgical cases necessitating post-operative mechanical ventilation. This data encompasses records from patients aged 0 to 7 years, treated over a period spanning from January 2015 to December 2018. It contains pertinent details such as the optimal tracheal tube depth assessed via chest X-ray and other patient characteristics—age, sex, height, and weight—that hold relevance for predicting the optimal tracheal tube depth.

## Data Preprocessing

Our analysis initiated without the need for additional data preprocessing, as the dataset was previously curated and presented in a clean, ready-to-analyze state. All records were numerical, requiring no transformation or encoding prior to being fed into our machine-learning algorithms.

## Data Analysis

Dividing the dataset into training and testing subsets facilitated the evaluation of our predictive models in a manner reflective of their performance in unseen cases. The machine-learning models we assessed included Random Forest, Elastic Net, Support Vector Machine, and Neural Network. Each model underwent hyperparameter optimization to ensure peak predictive capability.

Concurrently, we computed three formula-based predictions for the optimal tracheal tube depth using height, age, and internal diameter of the tube obtained from the dataset. The accuracy of each formula-based model was then assessed by comparing their output with the gold-standard optimal depths determined by chest X-ray on the test data.

The final stage of our analysis involved comparing the mean squared residuals of the predicted optimal tracheal tube depths from each machine-learning model against those of the formula-based models. This comparison was quantified through paired t-tests, assessing the significance of the differences between the predictions made by machine-learning versus traditional formula-based methods. The overarching aim of these analyses was to ascertain which methodology demonstrated superior performance in estimating the optimal tracheal tube depth, with the goal of enhancing precision in pediatric intubation practice.

## Code Availability

Custom code used to perform the data preprocessing and analysis, as well as the raw code outputs, are provided in Supplementary Methods.

## References

- [1] M. Kollef, Edward J. Legare, and M. Damiano. Endotracheal tube misplacement: Incidence, risk factors, and impact of a quality improvement program. *Southern Medical Journal*, 87:248–254, 1994.

- [2] P. Monnier. Pediatric airway surgery: Management of laryngotracheal stenosis in infants and children. 2010.
- [3] E. Mariano, C. Ramamoorthy, L. Chu, Michael I. Chen, and G. Hammer. A comparison of three methods for estimating appropriate tracheal tube depth in children. *Pediatric Anesthesia*, 15, 2005.
- [4] K. Takita, Y. Morimoto, and O. Kemmotsu. The height-based formula for prediction of left-sided double-lumen tracheal tube depth. *Journal of cardiothoracic and vascular anesthesia*, 17 3:412–3, 2003.
- [5] Alistair E. W. Johnson, M. Ghassemi, S. Nemati, K. Niehaus, D. Clifton, and G. Clifford. Machine learning and decision support in critical care. *Proceedings of the IEEE. Institute of Electrical and Electronics Engineers*, 104:444 – 466, 2016.
- [6] Wentao Bao, K. Ho, and Yucai Hong. Machine learning for the prediction of volume responsiveness in patients with oliguric acute kidney injury in critical care. *Critical Care*, 23, 2019.
- [7] A. Meyer, Dina Zverinski, B. Pfahringer, J. Kempfert, T. Kuehne, S. Sndermann, C. Stamm, Thomas Hofmann, V. Falk, and Carsten Eickhoff. Machine learning for real-time prediction of complications in critical care: a retrospective study. *The Lancet. Respiratory medicine*, 6 12:905–914, 2018.
- [8] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and V. Osmani. Benchmarking machine learning models on multi-centre eicu critical care dataset. *PLoS ONE*, 15, 2019.
- [9] S. A. Ingelse, H. Wieggers, J. Calis, J. V. van Woensel, and R. A. Bem. Early fluid overload prolongs mechanical ventilation in children with viral-lower respiratory tract disease\*. *Pediatric Critical Care Medicine*, 18:e106e111, 2017.
- [10] P. Gupta, M. Robertson, B. Beam, J. Gossett, M. Schmitz, C. Carroll, J. Edwards, J. Fortenberry, and W. Butt. Relationship of ecmo duration with outcomes after pediatric cardiac surgery: a multi-institutional analysis. *Minerva anesthesiologica*, 81 6:619–27, 2015.
- [11] Hamdan O. Alanazi, Abdul Hanan Abdullah, and K. Qureshi. A critical review for developing accurate and dynamic predictive models using machine learning methods in medicine and health care. *Journal of Medical Systems*, 41:1–10, 2017.

- [12] Duncan Shillan, J. Sterne, A. Champneys, and B. Gibbison. Use of machine learning to analyse routinely collected intensive care unit data: a systematic review. *Critical Care*, 23, 2019.
- [13] L. Fleuren, T. Klausch, Charlotte L. Zwager, L. Schoonmade, Tingjie Guo, Luca F Roggeveen, E. Swart, A. Girbes, P. Thorald, A. Ercole, M. Hoogendoorn, and P. Elbers. Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive Care Medicine*, 46:383 – 400, 2020.
- [14] Jing Xiao, R. Ding, Xiulin Xu, Haochen Guan, Xinhui Feng, Tao Sun, Sibozhu, and Zhibin Ye. Comparison and development of machine learning tools in the prediction of chronic kidney disease progression. *Journal of Translational Medicine*, 17, 2019.
- [15] Verena Schning, E. Liakoni, C. Baumgartner, A. Exadaktylos, W. Hautz, A. Atkinson, and F. Hammann. Development and validation of a prognostic covid-19 severity assessment (cosa) score and machine learning models for patient triage at a tertiary hospital. *Journal of Translational Medicine*, 19, 2021.



## 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.

Goal: Create and hyperparameter optimize different machine-learning models to predict OTTD, and to evaluate their performance compared to formula-based methods.

Dataset: For this study, we curated a dataset for creating, hyperparameter tuning, and evaluating machine-learning models to predict OTTD.

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:

```
import pandas as pd
import numpy as np

# Load dataset
df = pd.read_csv("tracheal_tube_insertion.csv")

# Calculate summary statistics
data_size = df.shape
summary_stats = df.describe()
missing_values = df.isnull().sum()

# Find out unique counts for categorical variables
categorical_vars = df.select_dtypes(include='object') #
    ↳ Assuming that categorical variables are of object data
    ↳ type
categorical_counts = categorical_vars.apply(lambda x: x.
    ↳ value_counts().idxmax())

# Saving output to txt file

with open('data_exploration.txt', 'w') as f:
    f.write('# Data Size\n')
    f.write(f'Number of Rows: {data_size[0]}, Number of Columns
        ↳ : {data_size[1]}\n\n')
```

```

f.write('# Summary Statistics\n')
f.write(str(summary_stats)+'\n\n')

if not categorical_vars.empty:
    f.write('# Categorical Variables\n')
    for var, val in categorical_counts.items():
        f.write(f'Most common value for {var}: {val}\n')
    f.write('\n')
else:
    f.write('# Categorical Variables\n')
    f.write('# Not Applicable\n\n')

f.write('# Missing Values\n')
for var, val in missing_values.items():
    if val > 0:
        f.write(f'Number of missing values for {var}: {val}
        ↪ }\n')
mismatch_vals = df[(df.age_c<0) | (df.ht<0) | (df.wt<0) | (
    ↪ df.tube_depth_G<0)]
if len(mismatch_vals)>0:
    f.write(f'Rows with non-logical negative values: {len(
    ↪ mismatch_vals)}\n')
else:
    f.write('No missing, unknown or illogical values found
    ↪ .\n')
f.close()

```

## B.2 Code Description

The provided code performs data exploration on the dataset "tracheal\_tube\_insertion.csv". It aims to gain insights into the dataset by calculating summary statistics, identifying missing values, and determining the most common values for categorical variables.

The code begins by loading the dataset into a pandas dataframe using the 'pd.read\_csv()' function. It then proceeds to perform the following analysis steps:

1. **Calculate Summary Statistics:** The 'describe()' function is used to calculate summary statistics for each numerical column in the dataframe. This includes count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum values. These statistics provide an overview of the distribution and range of values for each numerical variable.
2. **Identify Unique Counts for Categorical Variables:** The code identifies the categorical variables in the dataframe and retrieves the most frequent value for each categorical variable. This is done using the 'se-

lect\_dtypes()' function to select columns with object data type, and then applying the 'value\_counts().idxmax()' function to determine the most common value for each categorical variable.

**3. Identify Missing Values:** The code examines each variable in the dataframe and counts the number of missing values for each variable using the 'isnull().sum()' function. The results are then written to the output file.

**4. Identify Non-Logical Negative Values:** Following the check for missing values, the code performs an additional check to identify rows with non-logical negative values. This includes values less than zero for the age, height, weight, and tube depth variables. Any rows with such values are recorded and written to the output file.

The code writes the results of the data exploration analysis to a text file named "data\_exploration.txt". The output file includes the following information:

- Data size: The number of rows and columns in the dataset.
- Summary statistics: Detailed statistics (count, mean, standard deviation, etc.) for each numerical variable.
- Categorical variables: The most common value for each categorical variable.
- Missing values: The number of missing values for each variable, as well as any rows with non-logical negative values.

Overall, this data exploration code provides a comprehensive summary and analysis of the dataset, allowing for a better understanding of its characteristics and potential issues.

## B.3 Code Output

### data\_exploration.txt

```
\# Data Size
Number of Rows: 969, Number of Columns: 6

\# Summary Statistics
count      tube      sex  age\_c      ht      wt      tube\_depth\_G
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
\# Not Applicable
```

```
\# Missing Values  
No missing, unknown or illogical values found.
```

## C Data Analysis

### C.1 Code

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

```
# IMPORT  
import pandas as pd  
import numpy as np  
from sklearn.model_selection import GridSearchCV  
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  
from sklearn.metrics import mean_squared_error  
from scipy.stats import ttest_rel  
import warnings  
import pickle  
  
# Ignore 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 table is needed.  
  
# PREPROCESSING  
# No preprocessing is needed, because all data columns are  
  ↳ numerical and do not require transformation.  
  
# ANALYSIS  
  
## Table 1: "Comparison of mean squared residuals of predicted  
  ↳ OTTD by each machine-learning model and each formula-  
  ↳ based model"  
  
# Prepare features and target
```

```

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

# Split data
X_train, X_test, y_train, y_test = train_test_split(features,
    ↪ target, test_size=0.2, random_state=42)

# Create formula-based predictions
df_test = df.iloc[X_test.index]
df_test['ht_formula'] = df_test['ht'] / 10 + 5
df_test['age_formula'] = np.select([df_test['age_c'] < 0.5,
    ↪ df_test['age_c'] < 1, df_test['age_c'] < 2, df_test['
    ↪ age_c'] >= 2],
    [9, 10, 11, 12 + df_test['
    ↪ age_c'] * 0.5])
df_test['id_formula'] = df_test['tube'] * 3

# Compute residuals for formula-based models
residuals = {'height_formula': (df_test['ht_formula'] - y_test)
    ↪ **2,
    'age_formula': (df_test['age_formula'] - y_test)
    ↪ **2,
    'id_formula': (df_test['id_formula'] - y_test)**2}

# Create machine learning models
models = {'Random Forest': (RandomForestRegressor(), {'
    ↪ n_estimators': [10, 50, 100], 'max_depth': [None, 5, 10,
    ↪ 15, 20]}),
    'Elastic Net': (ElasticNet(), {'alpha': [0.1, 0.5, 1,
    ↪ 2, 5], 'l1_ratio': [0, 0.5, 1]}),
    'Support Vector Machine': (SVR(), {'C': [0.1, 1, 10,
    ↪ 100]}),
    'Neural Network': (MLPRegressor(max_iter=500), {'
    ↪ hidden_layer_sizes': [(50, 50), (100,)], '
    ↪ activation': ['tanh', 'relu']})
}

# Hyperparameter tuning and prediction for each model
for name, (model, params) in models.items():
    gs = GridSearchCV(model, params)
    gs.fit(X_train, y_train)
    predictions = gs.predict(X_test)
    residuals[name] = (predictions - y_test) ** 2
    models[name] = (gs.best_estimator_, gs.best_params_,
    ↪ mean_squared_error(y_test, predictions))

# Running t-tests for residuals
t_test_results = {}
for mdl in models.keys():

```

```

for formula in ['height_formula', 'age_formula', '
↳ id_formula']:
    t_test_result = ttest_rel(residuals[mdl], residuals[
↳ formula])
    t_test_results[f'{mdl} vs {formula}'] = [residuals[mdl
↳ ].mean(), residuals[formula].mean(),
↳ t_test_result.pvalue]

# Prepare DataFrame for Table 1
df1 = pd.DataFrame(t_test_results, index=['ML Mean Sq Residual'
↳ , 'Formula Mean Sq Residual', 'p-value']).T
df1.to_pickle('table_1.pkl')

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': len(df),
    'Observations used for training': len(X_train),
    'Observations used for testing': len(X_test),
}

for name, (model, params, mse) in models.items():
    additional_results[name] = {
        'best_hyperparameters': params,
        'model_mse': mse
    }

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

```

## C.2 Code Description

The provided code performs data analysis to predict the Optimal Tracheal Tube Depth (OTTD) in pediatric patients who require mechanical ventilation. The code focuses on creating and hyperparameter optimizing different machine learning models to predict OTTD, and evaluates their performance compared to formula-based methods.

The analysis steps are as follows:

1. The code first loads the dataset, which includes 969 patient records with features such as patient sex, age, height, weight, and the actual OTTD determined by chest X-ray.
2. Dataset preparations and descriptive statistics are not required in this analysis.
3. Preprocessing is not needed as all data columns are numerical and do not require transformation.

4. The code then splits the dataset into training and testing sets, with 80% of the data used for training and 20% for testing.

5. Formula-based predictions are computed by calculating the OTTD using pre-defined formulas based on patient features such as height, age, and tube ID.

6. Residuals, representing the difference between the actual OTTD and the predicted OTTD by formula-based models, are computed for each formula-based model.

7. Machine learning models including Random Forest, Elastic Net, Support Vector Machine, and Neural Network are created and hyperparameter tuned using GridSearchCV. For each model, the optimal hyperparameters and mean squared error (MSE) on the testing set are computed.

8. Residuals are computed for each machine learning model by calculating the difference between the actual OTTD and the predicted OTTD using the trained machine learning model.

9. T-tests are performed to compare the mean squared residuals between the formula-based models and each machine learning model. The p-values indicate the statistical significance of differences in prediction performance between the models.

10. The results of the t-tests are stored in a DataFrame, which is then saved to a pickle file named "table\_1.pkl".

11. Additional results such as the total number of observations, the number of observations used for training and testing, the best hyperparameters for each machine learning model, and the MSE of the models are stored in a dictionary and saved to a pickle file named "additional\_results.pkl".

In summary, the code creates and hyperparameter optimizes machine learning models to predict OTTD in pediatric patients. It compares the performance of these models with formula-based models using statistical tests and provides additional evaluation metrics for each model. The code outputs tables with the results of the statistical tests and stores additional results in pickle files for further analysis.

### C.3 Code Output

**table\_1.pkl**

```
ML Mean Sq Residual
Formula Mean Sq
Residual p-
value
```



Random Forest vs height\_formula	1.405
3.419 3e-09	
Random Forest vs age\_formula	1.405
1.79 0.0214	
Random Forest vs id\_formula	1.405
2.521 1.33e-05	
Elastic Net vs height\_formula	1.24
3.419 3.24e-11	
Elastic Net vs age\_formula	1.24
1.79 0.000449	
Elastic Net vs id\_formula	1.24
2.521 1.28e-07	
Support Vector Machine vs height\_formula	1.235
3.419 6.24e-12	
Support Vector Machine vs age\_formula	1.235
1.79 0.000117	
Support Vector Machine vs id\_formula	1.235
2.521 3.89e-08	
Neural Network vs height\_formula	1.355
3.419 7.66e-09	
Neural Network vs age\_formula	1.355
1.79 0.0116	
Neural Network vs id\_formula	1.355
2.521 2.42e-05	

#### additional\_results.pkl

```
{
  'Total number of observations': 969,
  'Observations used for training': 775,
  'Observations used for testing': 194,
  'Random Forest': {'best\_hyperparameters': {'max\_depth':
    5, 'n\_estimators': 50}, 'model\_mse': 1.405
  },
  'Elastic Net': {'best\_hyperparameters': {'alpha': 0.1, 'l1
    \_ratio': 0}, 'model\_mse': 1.24
  },
  'Support Vector Machine': {'best\_hyperparameters': {'C':
    10}, 'model\_mse': 1.235
  },
  'Neural Network': {'best\_hyperparameters': {'activation':
    'tanh', 'hidden\_layer\_sizes': (100,)}, 'model\_mse':
    1.355
  },
}
```

## 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 my_utils import to_latex_with_note, is_str_in_df,
    ↪ split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
    'ML Mean Sq Residual': ('ML MSE', 'Mean Squared Error of
        ↪ Machine-Learning models'),
    'Formula Mean Sq Residual': ('Formula MSE', 'Mean Squared
        ↪ Error of Formula-based models'),
    'p-value': ('p-value', 'p-value from the paired t-test
        ↪ comparing ML models with formula-based models'),
}

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

# RENAME ROWS AND COLUMNS
# Shorten the row names: replace '_' with ' ' and make
    ↪ abbreviations for formula names
df1 = df1.rename(index=lambda x: x.replace("_", " ").replace("
    ↪ vs height formula", " vs Height").replace(" vs age
    ↪ formula", " vs Age").replace(" vs id formula", " vs ID"))

# Use shared_mapping
mapping1 = dict((k, v) for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df1, k))

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="Comparison of mean squared residuals of predicted
        ↪ OTTD by various models.",
    label='table:comparison',
    note=None,
    legend=legend1)

```

## 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'.
    """

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

```

\% This latex table was generated from: 'table\_1.pkl'
\begin{table}[h]
\caption{Comparison of mean squared residuals of predicted OTTD
by various models.}
\label{table:comparison}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{\%
\begin{tabular}{lllll}
\toprule
\& ML MSE \& Formula MSE \& p-value \&
\midrule

```

```

\textbf{Random Forest vs Height} \& 1.41 \& 3.42 \& \<math>\$1e-06</math> \\
\textbf{Random Forest vs Age} \& 1.41 \& 1.79 \& 0.0214 \\
\textbf{Random Forest vs ID} \& 1.41 \& 2.52 \& 1.33e-05 \\
\textbf{Elastic Net vs Height} \& 1.24 \& 3.42 \& \<math>\$1e-06</math> \\
\textbf{Elastic Net vs Age} \& 1.24 \& 1.79 \& 0.000449 \\
\textbf{Elastic Net vs ID} \& 1.24 \& 2.52 \& \<math>\$1e-06</math> \\
\textbf{Support Vector Machine vs Height} \& 1.23 \& 3.42 \& \<math>\$1e-06</math> \\
\textbf{Support Vector Machine vs Age} \& 1.23 \& 1.79 \& 0.000117 \\
\textbf{Support Vector Machine vs ID} \& 1.23 \& 2.52 \& \<math>\$1e-06</math> \\
\textbf{Neural Network vs Height} \& 1.36 \& 3.42 \& \<math>\$1e-06</math> \\
\textbf{Neural Network vs Age} \& 1.36 \& 1.79 \& 0.0116 \\
\textbf{Neural Network vs ID} \& 1.36 \& 2.52 \& 2.42e-05 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{ML MSE}: Mean Squared Error of Machine-Learning models
\item \textbf{Formula MSE}: Mean Squared Error of Formula-based models
\item \textbf{p-value}: p-value from the paired t-test comparing ML models with formula-based models
\end{tablenotes}
\end{threeparttable}
\end{table}

```