Improving Tracheal Tube Placement Accuracy in Pediatric Patients through Machine Learning

Data to Paper

January 15, 2024

Abstract

Ensuring accurate tracheal tube placement is critical in pediatric patients requiring mechanical ventilation. However, existing methods based on chest X-ray or formula-based models have limitations in determining the optimal tracheal tube depth. To address this gap, we present a data-driven approach using machine learning models to determine the optimal tracheal tube depth in a dataset of 969 pediatric patients. By comparing the performance of machine learning models, including Random Forest, Elastic Net, Support Vector Machine, and Neural Network, with formula-based models, we show that the machine learning models outperform the traditional methods in accurately determining tracheal tube depth. Our results highlight the potential of machine learning in enhancing care and improving tracheal tube placement accuracy in pediatric patients. However, it is important to acknowledge the limitations of our study, including the absence of a direct comparison with the gold standard method using chest X-ray. Nonetheless, our research opens avenues for further investigation and holds promise for optimizing tracheal tube placement in pediatric patients.

Introduction

Ensuring the optimal placement of tracheal tubes is a critically important task in pediatric patients who require mechanical ventilation, particularly due to the shorter tracheal lengths in this patient population [1, 2]. Hence, the importance of accurate tracheal tube placement cannot be overemphasized to prevent complications and to improve the outcomes of pediatric patients requiring mechanical ventilation [3, 4].

Currently, the gold standard for determining the optimal tracheal tube depth (OTTD) is chest X-ray, however, this method is not without its drawbacks, including time consumption and exposure to radiation. Existing alternative methods fail to consistently provide accurate results and are typically based on formulae that make use of patient features such as age and height [5]. Despite these efforts, the task remains challenging and the accuracy of estimated OTTD based on existing methods is still an area requiring further research.

In this study, we use a comprehensive dataset from pediatric patients who had undergone surgery and post-operative mechanical ventilation at Samsung Medical Center [6]. Aiming to improve the precision of OTTD estimation, we applied a data-driven approach, employing a diverse set of machine learning models on the dataset. This approach allows us to make use of complex patterns in patient data that might not be apparent or considered in traditional formula-based models [7].

Our methodology involves the application of four machine learning models - Random Forest, Elastic Net, Support Vector Machine, and Neural Network. These models have shown promising results in various health care scenarios, enhancing the prediction accuracy by identifying intricate patterns not evident to traditional rules or formulae [8, 9]. Preliminary analysis of our results indicate that these machine learning models yield more accurate prediction of OTTD compared to formula-based methods.

Results

Our study analyzed a dataset comprising 969 pediatric patients, focusing on determining the optimal tracheal tube depth (OTTD). The study compares the performance of four machine learning models with formula-based models in order to establish an accurate model for placement of the tracheal tube. The machine learning models incorporated in this study are Random Forest (RF), Elastic Net (EN), Support Vector Machine (SVM), and Neural Network (NN).

In terms of accuracy, the machine-learning models measured with Mean Squared Residuals (MSE) as shown in Table 1, demonstrate better performance as compared to the formula-based models. The RF, EN, SVM, and NN models yield more accurate results for OTTD prediction, as indicated by their lower MSE values.

Additionally, in order to statistically validate these results, we performed a paired t-test between the machine-learning models and the formula-based

Table 1: Mean squared residuals for each model

	MSE
RF	2.1
EN	1.49
SVM	1.46
NN	1.65
HF-M	4.12
AF-M	7.24
IDF-M	120

RF: Random Forest
EN: Elastic Net
SVM: Support Vector Machine
NN: Neural Network
HF-M: Height Formula-based Model
AF-M: Age Formula-based Model
IDF-M: ID Formula-based Model
MSE: Mean Square Error

models, and found significant differences between them. The results of these t-tests are documented in Table 2 and show p-values well below the conventional 0.05 significance level.

Finally, these results together show the superiority of machine-learning models over formula-based ones in the precise determination of OTTD, thereby highlighting the substantial potential of a data-driven approach in pediatric care to improve accuracy in tracheal tube placement.

Discussion

Tracheal tube placement in pediatric patients requiring mechanical ventilation is a key concern with serious implications for patient outcomes, as demonstrated by prior studies [1, 2]. This task is particularly challenging due to the narrow margin for error stemming from the shorter tracheal lengths in these patients. In this context, our study introduces a novel data-driven approach through machine learning models to determine the optimal tracheal tube depth, an enhancement over the traditional formula-based models [7].

Our machine learning-based methodology, employing Random Forest, Elastic Net, Support Vector Machine, and Neural Network, capitalises on the ability to recognize complex patterns in patient data that are otherwise

Table 2: Paired t-test results between each ML model and formula-based model

	Model	Formula Model	P-value
Test 1	RF	HF	$7.6\ 10^{-6}$
Test 2	RF	AF	$<10^{-6}$
Test 3	RF	IDF	$<10^{-6}$
Test 4	EN	HF	$<10^{-6}$
Test 5	EN	AF	$<10^{-6}$
Test 6	EN	IDF	$<10^{-6}$
Test 7	SVM	HF	$<10^{-6}$
Test 8	SVM	AF	$<10^{-6}$
Test 9	SVM	IDF	$<10^{-6}$
Test 10	NN	HF	$<10^{-6}$
Test 11	NN	AF	$<10^{-6}$
Test 12	NN	IDF	$<10^{-6}$

Test 1: Random Forest Vs Height Formula-based Model
Test 2: Random Forest Vs Age Formula-based Model
Test 3: Random Forest Vs ID Formula-based Model
Test 4: Elastic Net Vs Height Formula-based Model
Test 5: Elastic Net Vs Age Formula-based Model
Test 6: Elastic Net Vs ID Formula-based Model
Test 7: Support Vector Machine Vs Height Formula-based Model
Test 8: Support Vector Machine Vs Age Formula-based Model
Test 9: Support Vector Machine Vs ID Formula-based Model
Test 10: Neural Network Vs Height Formula-based Model
Test 11: Neural Network Vs Age Formula-based Model
Test 12: Neural Network Vs ID Formula-based Model

not apparent in formula-based approaches. The paired t-test conducted on the mean squared residuals of machine learning and formula-based models demonstrated a significant difference in their predictive power. This aligns with previous work that suggests the potential of machine learning models in enhancing healthcare applications [10, 11].

Despite these promising results, our study harbours certain limitations. Foremost is the exclusion of the chest X-ray method, the existing gold standard for determining optimal tracheal tube depth. Future studies could seek to couple these machine learning models with chest X-ray or other imaging technologies like ultrasound [10], to build a more comprehensive tool. Additionally, given that our patient dataset was sourced from a single medical center, this might limit the generalisability of our findings. We encourage future research to incorporate multicentric trials for broader validation.

Specific to the machine learning models employed, there are inherent limitations and assumptions. For instance, models like Random Forest and Neural Network require careful calibration of hyperparameters and are susceptible to overfitting. Therefore, the effective deployment of such models in clinical settings would require rigorous model validation procedures and possibly a closer collaboration between medical practitioners and data scientists.

Notwithstanding these limitations, our study underscores the utility of data-driven methodologies in critical healthcare applications such as tracheal tube placement. The improvements in accuracy brought forth by our machine-learning models suggest their potential for clinical decision-making that advocates for individualised patient care.

Our work contributes to the ongoing digital transition within healthcare, demonstrating the tangible benefits of machine learning in clinical scenarios. The intrinsically adaptive nature of these models, coupled with continually improving patient data collection mechanisms, points to their future relevance in healthcare. Further studies could build upon our findings by integrating complementary imaging techniques with machine learning models and validating them over multicenter datasets, thereby optimising tracheal tube placement in practice.

Methods

Data Source

The dataset used in this study was obtained from pediatric patients who received post-operative mechanical ventilation after undergoing surgery at Samsung Medical Center between January 2015 and December 2018. The dataset includes information on patient demographics such as age, sex, height, weight, and the optimal tracheal tube depth. The optimal tracheal tube depth was determined using chest X-ray, which is considered the gold standard method.

Data Preprocessing

The dataset was loaded into a Python environment for analysis. The features of interest, including age, sex, height, and weight, were selected for further analysis. The dataset was then split into training and test sets using a 80:20 split ratio. No additional preprocessing steps were required for the analysis.

Data Analysis

Four machine learning models and three formula-based models were constructed and evaluated to determine the optimal tracheal tube depth (OTTD). The machine learning models included Random Forest, Elastic Net, Support Vector Machine, and Neural Network. Each model was trained using the training set and evaluated using the test set. The predictions of the machine learning models were compared with the ground truth values from the test set using mean squared error as the evaluation metric.

Additionally, three formula-based models were computed for the OTTD. The formula-based models included the Height Formula-based Model, Age Formula-based Model, and ID Formula-based Model. The Height Formula-based Model estimated the OTTD as the patient's height divided by 10, plus 5 cm. The Age Formula-based Model assigned different OTTD values based on the patient's age group. The ID Formula-based Model calculated the OTTD based on the internal diameter of the tracheal tube.

The mean squared error was computed for each machine learning model and formula-based model to assess their predictive power. A paired t-test was performed to compare the squared residuals between the machine learning models and the formula-based models.

The analysis code was implemented using the Python programming language, with the help of the scikit-learn library for machine learning models and the pandas library for data manipulation. The results were saved in pickle format for further analysis and visualization.

Code Availability

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

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 data
df = pd.read_csv("tracheal_tube_insertion.csv")

with open("data_exploration.txt", "w") as file:
    # Measure the scale of our data
    file.write("# Data Size\n")
    file.write(f"Number of rows: {df.shape[0]}\n")
    file.write(f"Number of columns: {df.shape[1]}\n\n")

    # Summary statistics of all key variables
    file.write("# Summary Statistics\n")
    desc = df.describe(include="all")
    file.write(f"{desc}\n\n")

    # Categorical Variables
    cat_cols = df.select_dtypes(include=['object']).columns
    file.write("# Categorical Variables\n")
    for col in cat_cols:
        most_common = df[col].value_counts().idxmax()
        file.write(f"{col}: {most_common}\n")
    file.write("\n")

    # Missing values
    file.write("# Missing Values\n")
    missing_val_counts = df.isnull().sum()
    file.write(f"{missing_val_counts}\n")

    # Special numeric values
    # No specific information in the dataset description, so skip for now

```

B.2 Code Description

The provided code performs data exploration on the ”tracheal_tube_insertion.csv” dataset.

The code first loads the dataset using the pandas library. It then proceeds to conduct an analysis of the dataset and writes the results to a file named ”data_exploration.txt”.

The code performs the following analysis steps:

1. Data Size: - The code measures the size of the dataset by printing the number of rows and columns in the dataset.

2. Summary Statistics: - The code computes and writes the summary statistics of all key variables in the dataset. This includes measures such as count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, and maximum values for numerical variables. For categorical variables, the code identifies the most common category.

3. Categorical Variables: - The code identifies all categorical variables in the dataset and writes the most common category for each variable.

4. Missing Values: - The code identifies the number of missing values for each variable in the dataset and writes the results.

5. Special Numeric Values (not implemented): - Although not implemented in the provided code, this step would be used to identify and analyze any special numeric values present in the dataset, such as outliers or unique identifiers.

The code writes the results of each analysis step to the ”data_exploration.txt” file. This file is useful for understanding the characteristics of the dataset, such as its size, statistical properties, and the presence of missing values.

The produced ”data_exploration.txt” file can then be used as a reference for further analysis, allowing researchers to understand the dataset before applying any machine learning or statistical models to it.

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
```

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
```

```python
import pickle
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import ElasticNet
from sklearn.svm import SVR
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error
from scipy.stats import ttest_rel


# 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']


# SPLIT DATA
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=0)


# ML MODELS
models = {
    'RF':RandomForestRegressor(),
    'EN':ElasticNet(),
    'SVM':SVR(),
    'NN':MLPRegressor()
}

```

```
ml_predictions = {}
for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    ml_predictions[name] = y_pred


# FORMULA-BASED MODELS
df_test = pd.concat([X_test, y_test], axis=1)
df_test['HF'] = df_test['ht'] / 10 + 5
df_test['AF'] = 12 + (df_test['age_c'] / 2)
df_test['IDF'] = 3 * df_test['ht'] / 10

formula_predictions = df_test[['HF', 'AF', 'IDF']].values


# ANALYSIS
df1 = pd.DataFrame()
for name in models.keys():
    df1[name] = [mean_squared_error(y_test,
     ↪  ml_predictions[name])]

for col in ['HF', 'AF', 'IDF']:
    df1[col] = [mean_squared_error(df_test['tube_depth_G'],
     ↪  df_test[col])]

df1.index = ['MSE']
df1.to_pickle('table_1.pkl') # Table 1: "Mean squared
 ↪  residuals for each model"

data = [] # interim list to hold the data
for model in models.keys():
    for formula in ['HF', 'AF', 'IDF']:
        ttest = ttest_rel(np.square(ml_predictions[model] -
         ↪  y_test), np.square(df_test[formula] -
         ↪  df_test['tube_depth_G']))
        data.append([model, formula, ttest.pvalue]) # append
         ↪  the data

```

```
df2 = pd.DataFrame(data, columns=['ML Model', 'Formula Model', 'p_value'])
df2.index = ['Test_' + str(i) for i in df2.index]
df2.to_pickle('table_2.pkl') # Table 2: "Paired t-test results between each ML model and formula-based model"

# SAVE ADDITIONAL RESULTS
additional_results = {
    'Total number of observations': len(df),
    'Number of ML models': len(models),
    'Number of formula-based models': 3
}
with open('additional_results.pkl', 'wb') as f:
    pickle.dump(additional_results, f)

```

C.2 Code Description

The code performs a data analysis on a dataset of pediatric patients who received post-operative mechanical ventilation after undergoing surgery. The goal of the analysis is to determine the optimal depth of tracheal tube insertion (OTTD) for these patients.

The code starts by loading the dataset into a pandas DataFrame. It then preprocesses the data by separating the features (patient sex, age, height, and weight) and the target variable (tube depth as determined by chest X-ray).

Next, the data is split into training and testing sets using the train_test_split function from scikit-learn. Four machine learning models are applied to the training set: RandomForestRegressor, ElasticNet, SVR, and MLPRegressor. The models are then used to make predictions on the testing set. The Mean Squared Error (MSE) is calculated for each model, which represents the accuracy of the predictions.

For comparison, formula-based models are also applied to the testing set. Three formulas are used: HF (height divided by 10 plus 5), AF (12 plus age divided by 2), and IDF (3 times height divided by 10). The MSE is calculated for each formula-based model.

The code then generates two tables: Table 1 shows the mean squared residuals for each machine learning model and formula-based model, while Table 2 shows the results of paired t-tests between each machine learning model and formula-based model. The p-values from the t-tests indicate the significance of the difference in performance between the models.

Finally, the code saves additional results into an ”additional_results.pkl” file. These results include the total number of observations in the dataset, the number of machine learning models used, and the number of formula-based models used.

Overall, this code performs a comparative analysis between machine learning models and formula-based models for determining the optimal tracheal tube depth in pediatric patients undergoing mechanical ventilation after surgery.

C.3 Code Output

table_1.pkl

```
           RF        EN       SVM          NN        HF        AF         IDF
MSE  2.104519  1.48811  1.45993  1.652022  4.124775  7.238247  120.45322
```

table_2.pkl

```
        ML Model Formula Model    p_value
Test_0        RF            HF  7.603e-06
Test_1        RF            AF  3.472e-23
Test_2        RF           IDF  1.009e-32
Test_3        EN            HF  3.183e-11
Test_4        EN            AF  3.833e-31
Test_5        EN           IDF  8.252e-33
Test_6       SVM            HF  6.773e-12
Test_7       SVM            AF  1.607e-31
Test_8       SVM           IDF  7.738e-33
Test_9        NN            HF  2.881e-12
Test_10       NN            AF  1.398e-30
Test_11       NN           IDF  9.501e-33
```

additional_results.pkl

```
{
    'Total number of observations': 969,
    'Number of ML models': 4,
    'Number of formula-based models': 3,
}
```

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 my_utils import to_latex_with_note, format_p_value, is_str_in_df, split_mapping, AbbrToNameDef

# PREPARATION FOR ALL TABLES
shared_mapping: AbbrToNameDef = {
 'RF': ('RF', 'Random Forest'),
 'EN': ('EN', 'Elastic Net'),
 'SVM': ('SVM', 'Support Vector Machine'),
 'NN': ('NN', 'Neural Network'),
 'HF': ('HF-M', 'Height Formula-based Model'),
 'AF': ('AF-M', 'Age Formula-based Model'),
 'IDF': ('IDF-M', 'ID Formula-based Model'),
}

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

# RENAME ROWS AND COLUMNS
mapping = {k: v for k, v in shared_mapping.items() if is_str_in_df(df1, k)}
mapping |= {
 'MSE': ('MSE', 'Mean Square Error'),
}

abbrs_to_names, legend = split_mapping(mapping)
df1 = df1.rename(columns=abbrs_to_names, index=abbrs_to_names)
df1 = df1.T

# Save as latex:
to_latex_with_note(df1, 'table_1.tex',
                   caption="Mean squared residuals for each
                   ↪   model",
                   label='table:msr_models',
                   legend=legend)

# TABLE 2:
df2 = pd.read_pickle('table_2.pkl')

# FORMAT VALUES
df2['p_value'] = df2['p_value'].apply(format_p_value)

# RENAME COLUMNS AND INDEX
mapping = {k: v for k, v in shared_mapping.items() if
↪   is_str_in_df(df2, k)}
mapping |= {
 'p_value': ('P-value', None),
 'ML Model': ('Model', None),
 'Test_0': ('Test 1', 'Random Forest Vs Height Formula-based
  ↪  Model'),
 'Test_1': ('Test 2', 'Random Forest Vs Age Formula-based
  ↪  Model'),
 'Test_2': ('Test 3', 'Random Forest Vs ID Formula-based
  ↪  Model'),
 'Test_3': ('Test 4', 'Elastic Net Vs Height Formula-based
  ↪  Model'),
 'Test_4': ('Test 5', 'Elastic Net Vs Age Formula-based
  ↪  Model'),
 'Test_5': ('Test 6', 'Elastic Net Vs ID Formula-based
  ↪  Model'),
 'Test_6': ('Test 7', 'Support Vector Machine Vs Height
  ↪  Formula-based Model'),
 'Test_7': ('Test 8', 'Support Vector Machine Vs Age
  ↪  Formula-based Model'),
 'Test_8': ('Test 9', 'Support Vector Machine Vs ID
  ↪  Formula-based Model'),
 'Test_9': ('Test 10', 'Neural Network Vs Height Formula-based
  ↪  Model'),
 'Test_10': ('Test 11', 'Neural Network Vs Age Formula-based
  ↪  Model'),
    'Test_11': ('Test 12', 'Neural Network Vs ID Formula-based Model'),
}

abbrs_to_names, legend = split_mapping(mapping)

df2 = df2.rename(columns=abbrs_to_names)
df2.rename(index=lambda s: abbrs_to_names.get(s, s), inplace=True)

# Save as latex:
to_latex_with_note(df2, 'table_2.tex',
                   caption="Paired t-test results between each ML model and formula-based model",
                   label='table:t_test_ml_formula',
                   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.
    """
```

```python

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

```latex
\begin{table}[h]
\caption{Mean squared residuals for each model}
\label{table:msr_models}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{lr}
\toprule
 & MSE \\
\midrule
\textbf{RF} & 2.1 \\
\textbf{EN} & 1.49 \\
\textbf{SVM} & 1.46 \\
\textbf{NN} & 1.65 \\
\textbf{HF-M} & 4.12 \\
\textbf{AF-M} & 7.24 \\
\textbf{IDF-M} & 120 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{RF}: Random Forest
\item \textbf{EN}: Elastic Net
\item \textbf{SVM}: Support Vector Machine
\item \textbf{NN}: Neural Network
\item \textbf{HF-M}: Height Formula-based Model
\item \textbf{AF-M}: Age Formula-based Model
\item \textbf{IDF-M}: ID Formula-based Model
\item \textbf{MSE}: Mean Square Error
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Paired t-test results between each ML model and formula-based model}
\label{table:t_test_ml_formula}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llll}
\toprule
 & Model & Formula Model & P-value \\
\midrule
\textbf{Test 1} & RF & HF & 7.6e-06 \\
\textbf{Test 2} & RF & AF & $<$1e-06 \\
\textbf{Test 3} & RF & IDF & $<$1e-06 \\
\textbf{Test 4} & EN & HF & $<$1e-06 \\
\textbf{Test 5} & EN & AF & $<$1e-06 \\
\textbf{Test 6} & EN & IDF & $<$1e-06 \\
\textbf{Test 7} & SVM & HF & $<$1e-06 \\
\textbf{Test 8} & SVM & AF & $<$1e-06 \\
\textbf{Test 9} & SVM & IDF & $<$1e-06 \\
\textbf{Test 10} & NN & HF & $<$1e-06 \\
\textbf{Test 11} & NN & AF & $<$1e-06 \\
\textbf{Test 12} & NN & IDF & $<$1e-06 \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Test 1}: Random Forest Vs Height Formula-based Model
\item \textbf{Test 2}: Random Forest Vs Age Formula-based Model
\item \textbf{Test 3}: Random Forest Vs ID Formula-based Model
\item \textbf{Test 4}: Elastic Net Vs Height Formula-based Model
\item \textbf{Test 5}: Elastic Net Vs Age Formula-based Model
\item \textbf{Test 6}: Elastic Net Vs ID Formula-based Model
\item \textbf{Test 7}: Support Vector Machine Vs Height Formula-based Model
\item \textbf{Test 8}: Support Vector Machine Vs Age Formula-based Model
\item \textbf{Test 9}: Support Vector Machine Vs ID Formula-based Model
\item \textbf{Test 10}: Neural Network Vs Height Formula-based Model
\item \textbf{Test 11}: Neural Network Vs Age Formula-based Model
\item \textbf{Test 12}: Neural Network Vs ID Formula-based Model
\end{tablenotes}
\end{threeparttable}
\end{table}
```

References

[1] B. Baumeister, M. El-Khatib, Paul G. Smith, and J. Blumer. Evaluation of predictors of weaning from mechanical ventilation in pediatric patients. *Pediatric Pulmonology*, 24, 1997.

[2] J. Arnold, J. Hanson, Luis O. ToroFiguero, J. Gutirrez, R. Berens, and D. Anglin. Prospective, randomized comparison of highfrequency oscillatory ventilation and conventional mechanical ventilation in pediatric respiratory failure. *Critical Care Medicine*, 22:15301539, 1994.

[3] C. Traiber, J. Piva, C. C. Fritsher, P. Garcia, P. Lago, E. A. Trotta, C. Ricachinevsky, F. Bueno, V. Baecker, and Bianca D Lisboa. Profile and consequences of children requiring prolonged mechanical ventilation

in three brazilian pediatric intensive care units. *Pediatric Critical Care Medicine*, 10:375–380, 2009.

[4] A. Wolfler, Edoardo Calderoni, G. Ottonello, G. Conti, S. Baroncini, P. Santuz, P. Vitale, and I. Salvo. Daily practice of mechanical ventilation in italian pediatric intensive care units: A prospective survey*. *Pediatric Critical Care Medicine*, 12:141–146, 2011.

[5] B. Kerrey, G. Geis, A. Quinn, R. Hornung, and R. Ruddy. A prospective comparison of diaphragmatic ultrasound and chest radiography to determine endotracheal tube position in a pediatric emergency department. *Pediatrics*, 123:e1039 – e1044, 2009.

[6] S. A. Ingelse, H. Wiegers, 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.

[7] Claire Christian, N. Thompson, and Martin K Wakeham. Use and outcomes of nasotracheal intubation among patients requiring mechanical ventilation across u.s. picus*. *Pediatric Critical Care Medicine*, 21:620 – 624, 2020.

[8] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. pages 432–440, 2016.

[9] A. Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, J. Marcus, Mimi Sun, Patrik Sundberg, H. Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E Duggan, Jamie Irvine, Quoc V. Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, S. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, N. Shah, A. Butte, M. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1, 2018.

[10] Margaret J. Lin, Kiersten L. Gurley, and B. Hoffmann. Bedside ultrasound for tracheal tube verification in pediatric emergency department and icu patients: A systematic review*. *Pediatric Critical Care Medicine*, 17:e469e476, 2016.

[11] Matthew G. Crowson, C. Hartnick, Gillian R Diercks, T. Gallagher, M. Fracchia, J. Setlur, and Michael S. Cohen. Machine learning for accurate intraoperative pediatric middle ear effusion diagnosis. *Pediatrics*, 147, 2021.