Prediction of Severe Asthma Outcomes in Children on EHR Data

by

Jiaqian Liu

BS, Nanjing Agricultural University, 2019

Submitted to the Graduate Faculty of the

School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

Jiaqian Liu

It was defended on

April 20, 2023

and approved by

Lu Tang, PhD, Assistant Professor, Biostatistics, School of Public Health, University of Pittsburgh

Erick Forno, MD, MPH, Associate Professor, Pediatrics and Clinical and Translational Science, School of Medicine, University of Pittsburgh

Thesis Advisor/Dissertation Director: Ying Ding, PhD, Associate Professor, Biostatistics, School of Public Health, University of Pittsburgh

Prediction of Severe Asthma Outcomes in Children on EHR Data

Jiaqian Liu, MS

University of Pittsburgh, 2023

Background: Asthma is a leading chronic disease among children with nonnegligible numbers of Emergency Department (ED) visits and hospitalization annually. To effectively utilize real-world electronic health record (EHR) data, it is crucial to identify the best modeling approach that accounts for the unique features of EHR data. Additionally, identifying high-risk sub-populations susceptible to severe asthma outcomes can provide valuable insights for targeted interventions and improved patient medical care.

Methods: Various statistical and machine learning models, including those with random effects such as linear (generalized) mixed effects models, and mixed effects random forests, were employed to develop a prediction model for length of stay (LOS) and asthma exacerbation using EHR data. Once the optimal prediction model was identified, it was further trained on the entire dataset to identify the risk factors that significantly contribute to severe asthma outcomes.

Results: Linear mixed effects model and generalized linear mixed effects model were the top-performing models for predicting inpatient LOS and asthma exacerbation risk, with an average RMSE of 0.53 and AUC of 0.87, respectively. Notably, patient age, action plan, and the patient health history such as inpatient visit (yes or no from the last encounter) and ED visit (yes or no from the last encounter) were the strongest predictors of severe asthma outcomes. Other statistically significant predictors included having chronic diseases, belonging to a minority race group, and during the pandemic period.

Conclusion: Findings show that the inclusion of mixed effects enhances the prediction performance on EHR data. Factors such as patient age, action plan, and historical information on hospitalization and ED visits were identified as crucial predictors. The results highlight the importance of incorporating mixed effects when dealing with correlated encounter-level data and paving the way for more comprehensive EHR-based prediction models.

Public Health Significance: Understanding the association between risk factors and severe asthma outcomes will help with early intervention and precision pediatric asthma health care which can effectively improve health outcomes and reduce cost burden among children with asthma.

Table of Contents

List of Tables

List of Figures

Preface

I am immensely grateful to my advisor, Dr. Ying Ding, for her untiring support throughout my thesis analysis journey. Her guidance and mentorship began during the summer break in 2022 and extended throughout the entire research process. Her expertise in novel statistical techniques, invaluable assistance in project data analysis, and willingness to share insights on various research topics from her PhD students have been an unforgettable and enriching experience for me. I am truly thankful for her contributions to my academic growth.

I would like to extend my heartfelt appreciation to Dr. Erick Forno and Dr. Lu Tang. Dr. Forno has been instrumental in helping me understand the complex terminology and related to asthma EHR data with patience and timely advice, which has been immensely beneficial to my study. Dr. Tang's insightful suggestions and feedback greatly enriched my research and made my defense a truly enjoyable experience.

I also want to thank Xueping Zhou for her generous assistance in data cleaning and providing support for statistical analysis.

1.0 Introduction

Asthma is a long-term condition affecting both adults and children, which had a prevalence of 25 million people in the United States in 2020 according to CDC [1]. It is the most common multifactorial chronic disease which disproportionately affects children and may develop into an asthma attack (exacerbation) if the risk factors are not well controlled [2]. 42.7% of the children (Age < 18 years) with current asthma had at least one asthma attack in 2020 [3]. Acute asthma exacerbations require the patient to seek immediate care and can lead to Emergency Department (ED) encounters and hospitalizations [4-5]. Inadequate disease control of children asthma leads to over 500,000 Emergency Department (ED) visits and 80,000 hospital stays annually in the US [1]. On the other hand, children with well-managed asthma care tend to have better outcomes, including reduction in exacerbation, ED visits and hospitalizations.

1.1 EHR Data Analytics

An Electronic Health Record (EHR) is an electronic version of medical history, which contains a rich source of clinical information, including demographics, medical visit times, diagnoses, medications, procedures, vital signs, and lab tests for a large population of patients. Unlike traditional datasets attained from clinical trials or experimental studies, EHR dataset are always big, messy, high dimensional and with high missing rates [6]. To address these characteristics, machine learning (ML) approaches have become a trendy method for researchers in EHR analytics because the ML algorithms tend to have a better performance when a large

sample data is available for training. However, it is common to see the responses are not independent and the covariates are in longitudinal or clustered format in EHR datasets, which means traditional statistical methods such as Linear or Generalized Linear Model (GLM) cannot handle the clustered structure and may fail to learn the potential information behind the correlated nature of EHR data. Therefore, this research utilized statistical models and tree-based machine learning methods with mixed effects when developing prediction models.

1.2 Objectives

The main purposes of this study are to develop predictive models for severe asthma outcomes including length of stay (LOS) and exacerbation using pediatric asthma EHR data. Additionally, the study aims to identify factors that contribute to severe asthma outcomes, allowing for the identification of subgroups with higher risk to these outcomes.

To achieve these objectives, a set of prediction models were established using Linear Regression Model (LM), Generalized Linear Regression Model (GLM), and Random Forests (RF). To address the clustered structure of EHR data, Linear Mixed Effects Model (LME), Mixed Effects Random Forests (MERF) and Binary Mixed Effects Forests (BiMM) were employed for constructing the prediction models by incorporating random effects. The evaluation of the models was conducted by calculating the Root Mean Square of Error (RMSE), the Area Under Curve (AUC) of Receiver Operating Characteristic (ROC) and feature importance scores from Random Forests, depending on the type of outcomes.

1.3 Public Health Significance

Asthma is one of the main contributors of hospitalization which are particularly common in children aged < 5 years, imposing an increasingly consistent burden on health system [7]. Average annual expenses per child ranged from $3,076 to $13,612 in the United States, with a significant proportion of inpatient and ED visits [8]. Early detection of asthma severe exacerbation can provide instructions for appropriate treatment which effectively reduce the disease cost burdens. Therefore, this research may help to better characterize factors that trigger children asthma severity and aid in offering guidance on adopting intervention strategies.

2.0 Methods

2.1 Data Definition and Preparation

The primary aim of this section is to provide detailed descriptions of the predictors, outcome variables and the patients demographics that were utilized in the analysis.

2.1.1 Predictor Variable

Table 1. Predictors Definition

Variables	Definition
Sex	Female, male
Race	Race falls into the following categories: • White • Black • Asian/Multi • Others/unknown: treated as missing and excluded from the analysis.
State	Patients' location: • PA • OH • WV • Others: treated as missing due to their extremely low proportion and excluded from the analysis.
Patients Age	The age of patients is time-dependent and varies depending on the encounter date.
ED	The variable is marked as "True" if a patient was admitted to the Emergency Department during an encounter visit.
Inpatient	The variable is marked as "True" if a patient hospitalized during his/her encounter visits, and "False" otherwise.
Action Plan	Asthma action plan is a written guidance to help patients manage their asthma symptoms and provide instructions on

	what to do if they get out of control. The variable is marked as "True" if a patient was provided an action plan during an encounter visit, and "False" otherwise.
During Pandemic	The variable is marked as "True" if the encounter visit happened after the Covid-19 shutdown date of 3/15/2020 and before the date 05/01/2021, and "False" otherwise.
Asthma Assessment	The variable is marked as "True" if a patient used asthma scale table for asthma assessment during an encounter visit, and "False" otherwise.
Chronic Disease Existing	The variable "existing diseases" is created based on a range of diseases, such as chronic illnesses and respiratory-related diseases, as contributing factors: ● Bronchomalacia ● Bronchopulmonary Dysplasia ● Cardiac Disease ● Immunodeficiency ● Pulmonary Hypertension ● Sickle Cell ● Tracheomalacia The variable is marked as "True" if patients had any of diseases listed above, and "False" otherwise
Influenza Vaccine	The variable is marked as "True" if the patients had received flu shot before the encounter visit date, and "False" otherwise.
Acute Bronchospasms	The variable is marked as "True" if the patients was diagnosed with acute bronchospasms, and "False" otherwise.
Subcutaneous Emphysema	The variable is marked as "True" if the patients was diagnosed with subcutaneous emphysema, and "False" otherwise.

Table 2. Patients Demographics

Inpatient	FALSE (N=10,099)	TRUE (N= 2,901)	Overall (N=13,000)
Gender			
Female	4189 (41.5%)	1164 (40.1%)	5353 (41.2%)
Race			
Black	2169 (21.5%)	1034 (35.6%)	3203 (24.6%)
White	7426 (73.5%)	1717 (59.2%)	9143 (70.3%)
Multi/Asian	504 (5.0%)	150 (5.2%)	654 (5.0%)
State			
PA	9826 (97.3%)	2941 (97.9%)	12667 (97.4%)
OH	143 (1.4%)	34 (1.2%)	177 (1.4%)
WV	130 (1.3%)	26 (0.9%)	156 (1.2%)
Age*			
Mean (SD)	8.9 (4.7)	6.5 (4.5)	8.6 (4.7)
Median [Min, Max]	8.0[2.0, 21.0]	5.0 [2.0, 21.0]	8.0 [2.0, 21.0]

***The statistics of age visits were calculated based on patients encounters level data, which means the age changed for different encounters of a given patient.**

2.1.2 Outcome Variable

2.1.2.1 Length of Stay (LOS)

The continuous outcome variable, LOS, is defined as the number of hours patients spent in the hospital if they were admitted to the inpatient unit during an encounter visit. All encounters with length of stay greater than 7 days (168 hours) and ED = False were excluded in the final analysis. Since the distribution of LOS is highly skewed, a new outcome variable “log_LOS” was created by taking the natural logarithm transformation on the original scale LOS. Figure 1 shows the distribution of LOS before and after the log-transformation.

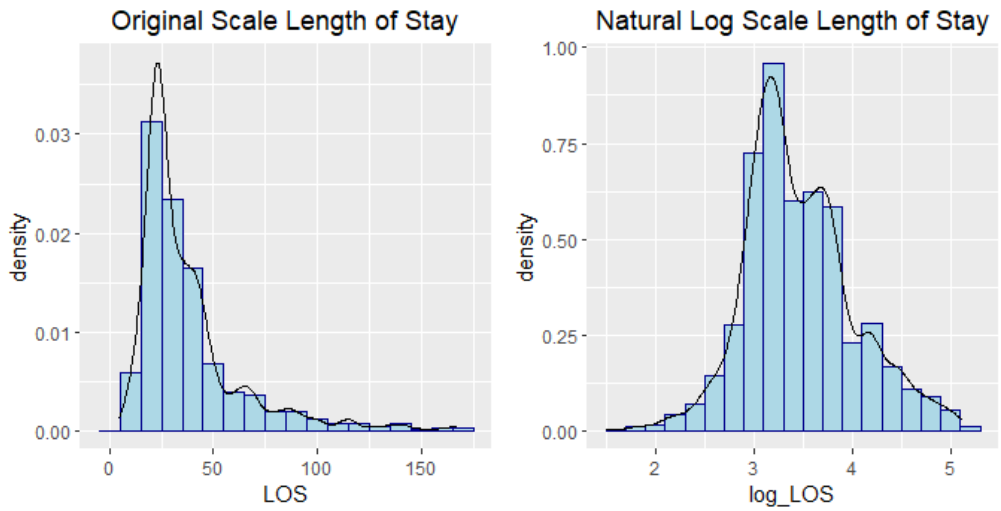


Figure 1. Distribution of Length of Stay Before and After Log-transformation

2.1.2.2 Asthma Exacerbation

Asthma Exacerbation is a binary outcome which is defined as the patients who had admitted to ED and had received albuterol dose during an encounter visit. Figure 2 shows the proportion of asthma exacerbation in the whole dataset. A total of 8,307 encounters (26.65%) has asthma exacerbation.

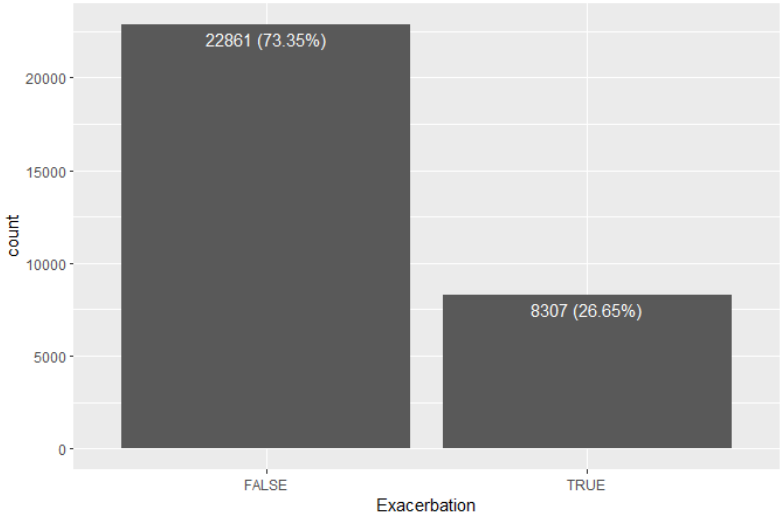


Figure 2. Proportion of Asthma Exacerbation

2.2 Models

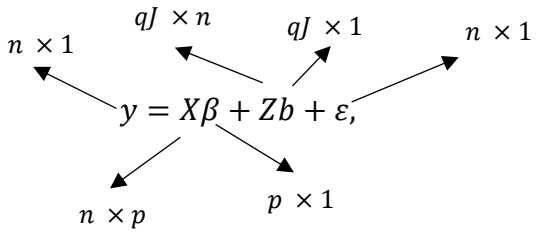
To address the clustered structure of the asthma EHR data, various mixed effects linear models were utilized, including Linear Mixed Effects Model (LME) and Generalized Linear Mixed Model (GLMM). Additionally, mixed effects tree-based methods, such as Mixed-Effect Random Forests (MERF) and Binary Mixed Model Forests (BiMM), were employed. Linear Regression, Generalized Linear Regression, and standard Random Forests were also used as modeling techniques on the data for comparisons. This section introduces all modeling methods.

2.2.1 Linear and Generalized Linear Models

The linear and generalized linear models are widely utilized in basic regression or classification problems. Since the analysis mainly focuses on using models which can handle random effects, this section will focus on LME and GLMM models.

2.2.1.1 Linear Mixed Effects Model (LME)

The linear mixed-effect model is a statistical method for analyzing longitudinal, clustered or muti-level data as it extends the simple linear models by involving both random effects and fixed effects. The following equation demonstrates how LME incorporates random effects by introducing Zb, where Z is a matrix for the q random effects of J groups, b is the $qJ \times 1$ vector of q random effects [6]. Unlike β, which is the fixed effect parameter that does not vary, b varies across different groups and serves as the additive part to the fixed effects [9].



2.2.1.2 Generalized Linear Mixed Effects Model (GLMM)

Unlike LME, GLMM can handle response variables that come from different distributions, beyond just the Gaussian distribution [10]. This is achieved through the utilization of a link function. Let η be the linear combination of fixed and random effects without the residuals, and $g(\cdot)$ represents the link function:

$$y = g^{-1}(\eta) + \varepsilon$$

where $\eta = X\beta + Zb$, $g\big(E(y)\big) = \eta$. Specifically, when dealing with binary outcomes, the logistic link function is employed in GLMM, denoted as $g(\cdot) = log_e(\frac{p}{1-p})$. GLMM can also address the count outcome and continuous outcome problems. When dealing with the continuous outcome with normal distribution, GLMM becomes the model we introduced in LME, which

indicates that GLMM has the capability to accommodate a wider range of data types and distributions compared to LME.

2.2.2 Tree-based Models

2.2.2.1 Random Forests (RF)

Random Forests is a commonly used machine learning method which is made up of multiple decision trees based on ensemble learning and bagging methods [11]. Bagging allows RF to produce uncorrelated decision trees by only considering random subsets of features, which is a key difference compared to a single decision tree. Each tree in forests comprised of a sample data extracted from whole training dataset with replacement (bootstrap). There are three important parameters that needed to be specified before training the model, which is the number of trees, node size and the number of features to consider when looking for the best split (called *mtry* in R). The RF prediction depends on the type of problem that RF is being used to solve. Figure 3 shows how the RF model is created. For the regression problems, the predictions from all the decision trees in forests are averaged. For the classification problems, the predicted outcome is decided on the most frequent class of all individual trees.

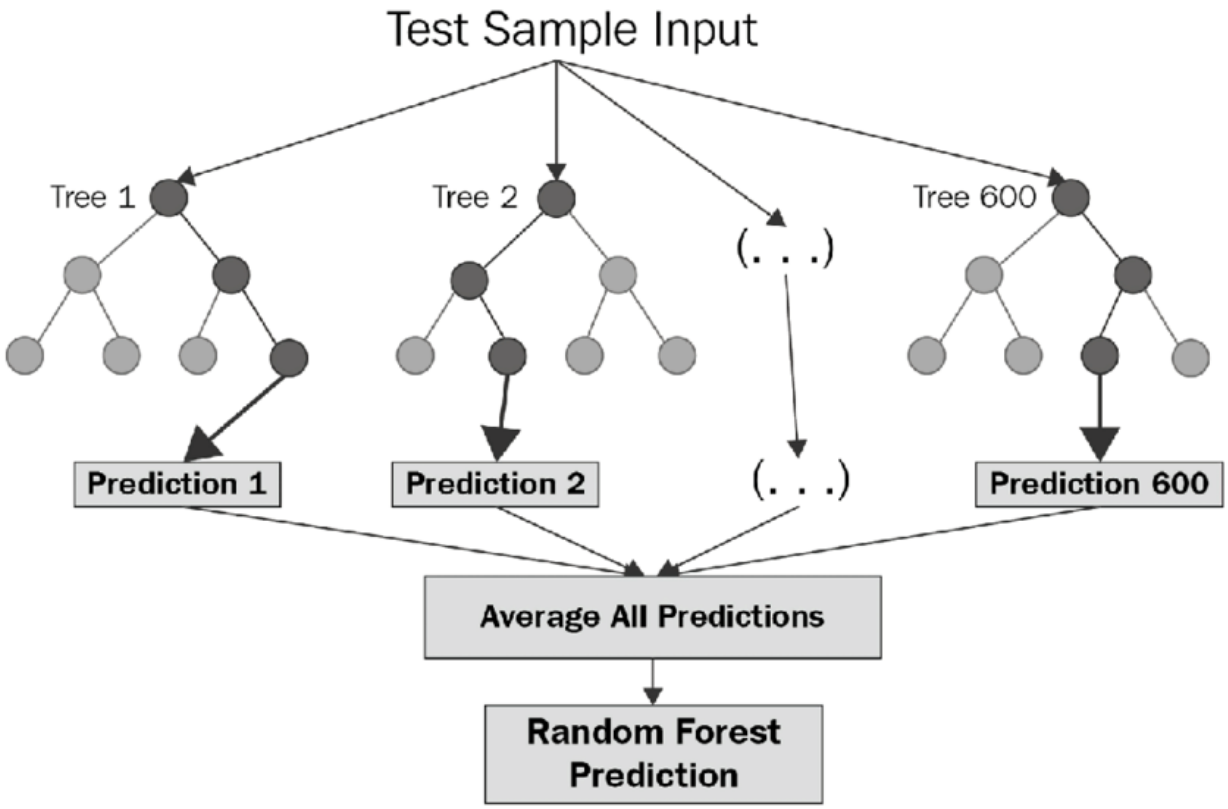


Figure 3. Random Forests Structure From Rudd, Jessica & Ray, Herman, 2020

2.2.2.2 Mixed Effects Random Forests

The Mixed Effects Random Forests (MERF) [12] is defined as follow:

$$y_i = f(X_i) + Z_i b_i + \epsilon_i,$$

$$b_i \sim N(0, D), \epsilon_i \sim N(0, R_i), i = 1, \dots, n$$

where $y_i = [y_{i1}, \dots, y_{in_i}]^T$ is the outcome variable for the n_i observations in cluster i, $X_i = [x_{i1}, \dots, x_{in_i}]^T$ is the $n_i \times p$ matrix of fixed-effects covariates, $Z_i = [z_{i1}, \dots, z_{in_i}]^{\{T\}}$ is the $n_i \times q$ matrix of random-effects covariates, $b_i = [b_{i1}, \dots, b_{iq}]^T$ is the $q \times 1$ unknown vector of random effects for cluster i. $\epsilon_i = [\epsilon_{i1}, \dots, \epsilon_{in_i}]^T$ is the $n_i \times 1$ vector of errors. D is the covariance matrix of b_i, while R_i is the covariance matrix of ϵ_i.

The MERF model equation is very similar to the LMM model, but with the fixed-effects term βX replaced by $f(X_i)$, which is estimated using Random Forests. In MERF, it is assumed that the correlation is induced solely via the between-cluster variation, making R_i diagonal ($R_i = \sigma^2 I_{n_i}$), which is suitable for dealing with large clustered datasets.

The MERF algorithm is presented in Figure 4, which consists of three steps. In step1, the algorithm initially assigns default values to $\hat{b}_i$, σ^2, and D^2. It then calculates the y_i^* by subtracting the random component in step2: y_i^* becomes updated responses value which are used in the training sets to build RF to obtain the $\hat{f}(X_i)$. Subsequently, the algorithm calculates the updated $\hat{b}_i$ based on the $\hat{f}(X_i)$ and updated estimate of random component. In step3, it updates the variance σ^2, and D^2 based on the updated estimate of residuals. The algorithm continues to iterate until it reaches convergence.

The criterion for convergence is determined by the following generalized log-likelihood (GLL) function:

$$GLL(f, b_i|y) = \sum_{i=1}^{n} \{[y_i - f(X_i) - Z_i b_i]^T R_i^{-1} [y_i - f(X_i) - Z_i b_i] + b_i^T D^{-1} b_i + \log|\mathrm{D}| + \log|\mathrm{R_i}|\}$$

When the GLL between two iterations falls below a specific threshold or the maximum number of iterations is reached, the algorithm converges.

Algorithm: MERF algorithm

Input: *Clustered data:* $\{(x_{ij}, y_{ij}),\ i = 1,\ldots,N\}$

Output: *Estimated Random Forests model* $\hat{f}$ *and random effect* $\hat{b}_i$

Initialization: $r = 0$, *initialize the random effects* $\hat{b}_{i(0)} = 0$, *variance* $\hat{\sigma}^2_{(0)} = 1$ *and* $\hat{D}_{(0)} = I_q$

1 **for** $r = 1, 2, 3\ldots$ *to convergence of GLL criterion or max iterations* **do**

2 Update $y^*_{i(r)}$, $\hat{f}(X_{ij})_{(r)}$ and $b_{i(r)}$:

- Compute $y^*_{i(r)} = y_i - Z_i \hat{b}_{i(r)}$
- Build a Random Forest with $y^*_{ij(r)}$ as training set responses and obtain an estimate $\hat{f}(x_{ij})_{(r)}$ of $f(x_{ij})$
- Compute $\hat{b}_{i(r)} = \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} (y_i - \hat{f}(X_{ij})_{(r)})$, $i = 1, ..n$,where $\hat{V}^{-1}_{i(r-1)} = Z_i \hat{D}_{(r-1)} Z_i^T + \hat{\sigma}^2_{(r-1)} I_{ni}, i = 1, ..n$

3 Update $\hat{\sigma}^2_{(r)}$ and $\hat{D}_{(r)}$ using:

- $\hat{\sigma}^2_{(r)} = N^{-1} \Sigma^n_{i=1} \{\hat{\epsilon}^T_{i(r)} \hat{\epsilon}_{i(r)} + \hat{\sigma}^2_{(r-1)} [n_i - \hat{\sigma}^2_{(r-1)} trace(\hat{V}_{i(r-1)})]\}$,
- $\hat{D}_{(r)} = n^{-1} \Sigma^n_{i=1} \{\hat{b}^T_{i(r)} \hat{b}_{i(r)} + [\hat{D}_{(r-1)} - \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} Z_i \hat{D}_{(r-1)}]\}$,

4 **end**

Figure 4. MERF Algorithm Adapted From Hajjem et al., 2014

2.2.2.3 Binary Mixed Model Forests

Similar to MERF, Binary Mixed Model Forests (BiMM) combines Bayesian Generalized Linear Mixed Model (Bayesian GLMM) with Random Forests in order to address the binary outcome problems. The algorithm of BiMM also uses the EM algorithm and assumes the existence of random effects when constructing the random forest, as well as the existence of fixed effects when constructing the Bayesian GLMM.

The Binary Mixed Model Forest [13] is defined as follow:

$$logit(y_i) = \beta_0 + \beta_1 RF(X_i) + Z_i b_i,$$

where $y_i = [y_{i1}, \ldots, y_{in_i}]^T$ is the outcome variable for the n_i observations in cluster i, β_0 is the coefficient for the intercept and β_1 is the coefficient for the vector of probabilities $RF(X_i)$. $Z_i = [z_{i1}, \ldots, z_{in_i}]^{\{T\}}$ is the clustered covariates for cluster i, and $b_i = [b_{i1}, \ldots, b_{iq}]^T$ is the random effect

for cluster i. The algorithm for BiMM is presented in Figure 5, which consists of three main steps. In Step 1, the algorithm simply assumes the outcome y_i as independent and used it as training sets when building the Random Forests $RF(X_i)$. This leads to step 2, a Bayesian Generalized Linear Mixed Model (GLMM) is fitted with $RF(X_i)$, which is the known fixed effects from step1. The Bayesian GLMM model then provides predicted probabilities, which are denoted as q_{ij}. In Step 3, it computes the updated responses value y_i^* by adding the q_{ij} and applying split function to make y_i^* back to binary value. The algorithm keeps iterating until the change of the Posterior Log-likelihood of Bayesian GLMM is below a tolerance value.

There are three types of split function:

$$h_1(y_{ij} + q_{ij}) = \begin{cases} 1, \ if y_{ij} + q_{ij} > k_1 \\ 0, \ otherwise \end{cases}, \text{ where } 0 < k_1 < 1$$

Since the y_{ij} takes binary value with either 0 or 1 and q_{ij} is the probabilities ranging from 0 and 1, the value of $y_{ij} + q_{ij}$ is between 0 and 2. Using h_1 function helps maximize the sensitivity when updates the target outcome because it can only update the original outcome of 0 to 1, while preventing updates from 1 to 0.

$$h_2(y_{ij} + q_{ij}) = \begin{cases} 0, \ if y_{ij} + q_{ij} < k_2 \\ 1, otherwise \end{cases}, \text{ where } 1 < k_2 < 2$$

Similarly, using h_2 function helps maximize the specificity when updates the target outcome because it can only update the original outcome of 1 to 0, and if the original outcome value is 0, it will remain as 0.

$$h_3(y_{ij} + q_{ij}) = \begin{cases} 0, \ if y_{ij} + q_{ij} < 0.5 \\ 1, if y_{ij} + q_{ij} > 1.5 \\ 1 \ with \ probability \ q_{ij} \ if \ 0.5 < y_{ij} + q_{ij} < 1.5 \\ 0 \ with \ probability \ 1 - q_{ij} \ if \ 0.5 < y_{ij} + q_{ij} < 1.5 \end{cases}$$

h_3 function provides with a more general way to transform the $y_{ij} + q_{ij}$, that is, the original outcome value of 1 can be updated 0, and original outcome value with outcome 0 can be update to 1. Specifically, during each iteration, if the current prediction aligns with the original outcome, the updated outcome remains unchanged. However, if the current iteration does not align with the original outcome, the updated outcome value becomes 1 with probability q_{ij} or becomes 0 with probability of $1 - q_{ij}$.

Algorithm: BiMM algorithm

Input: *Clustered data:* $\{(x_{ij}, y_{ij}),\ i = 1,\ldots,N\}$

Output: *Estimated Random Forests model* $\hat{RF}$ *and the Bayesian GLMM model*

1 **for** *iteration* $= 1, 2, 3\ldots$ *to the convergence of posterior log-likelihood or max iterations* **do**

2 Compute $\hat{RF}(X_{ij})$:

- Build a Random Forest with y^*_{ij} as training set responses and obtain an estimate $\hat{RF}(x_{ij})$ of $RF(x_{ij})$

3 Fit a Bayesian GLMM model and extract the predicted probabilities from the model, denoted $q_{ij} = Pr_B GLMM(X_{ij}, Z_{ij})$:

- $logit(y^*_{i(r)}) = \beta_0 + \beta_1 \hat{RF}(x_{ij}) + Z_{ij}\hat{b}_{ij},\ i = 1, ..N$

4 Compute y^*_{ij} by adding the q_{ij} from the original y_{ij}, then applying a split function to make y^*_{ij} a binary outcome:

- $y^*_{ij} = h(y_{ij} + q_{ij})$

5 **end**

Figure 5. BiMM Algorithm Adapted From Jaime Lynn Speiser et al., 2019

2.3 Application to Asthma EHR data

The main purpose of this section is to introduce the "lag data", a newly created dataset that combines current encounter information with history encounter information, based on the original asthma electronic health record (EHR) data. Furthermore, the modeling process and evaluation metrics for the model will also be detailed.

2.3.1 Lag Data

Given the longitudinal nature of the EHR data, the lag datasets were created in order to make use of the history information for those patients who had at least 2 encounters within the whole study period (January 1st, 2019 - January 26th, 2023). Only time-varying predictors were considered lag predictors in the lag dataset. For example, a patient might not be admitted to the inpatient unit during the current visit but may have been hospitalized during the immediately preceding visit before the current one, which indicates that "Inpatient" is a time-varying predictor and can be extended to "Inpatient_lag1" predictor in the lag dataset. Figure 6 shows how we created the lag predictors.

ID	Encounter time	Age	BP
101	Monday	8	125
101	Tuesday	8	120
101	Thursday	8	119
101	Friday	8	130

Encounter time	BP
Monday	125
Tuesday	120
Thursday	119
Friday	130

ID	Encounter time	Age	BP	Encounter time_lag1	BP_lag1
~~101~~	~~Monday~~	~~8~~	~~125~~	~~NA~~	~~NA~~
101	Tuesday	8	120	Monday	125
101	Thursday	8	119	Tuesday	120
101	Friday	8	130	Thursday	119
~~NA~~	~~NA~~	~~NA~~	~~NA~~	~~Friday~~	~~130~~

Figure 6. The Procedure for Generating Lag Predictors

2.3.2 Modeling Process

For training and validation on every three datasets (lag data, non-lag data and non-lag data with the same sample size as lag data), two split methods were applied:

2.3.2.1 Random Split

The data was divided randomly into an 80% training set and a 20% test set. Next, a 5-fold cross-validation was performed on the training set, resulting in 5 distinct "fold models" since each fold model was trained on 4/5 of the folds and evaluated on the leave-out fold. Subsequently, these fold models were validated on an external 20% test set, and an average RMSE or an average AUC was calculated. This entire process was repeated 10 times for LOS prediction and 5 times for asthma exacerbation prediction.

2.3.2.2 Date Split

Unlike random split, which results in varying training and test sets with each repetition, date split divides data based solely on the registration date of the encounters. Specifically, encounters with a registration date prior to July 1st, 2022, are assigned to the training set, while encounters with a registration date on or after July 1st, 2022, are assigned to the test set. The subsequent steps remain the same as in the random split analysis.

2.3.3 Model Evaluation Metrics

2.3.3.1 Root Mean Square Error (RMSE)

The RMSE is computed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j}^{c_i}(y_{ij} - \hat{y}_{ij})}{\sum_{i}^{n} c_i}}$$

with $\hat{y}_{ij}$ being the predicted value of the jth observation in the ith cluster in the test set. c_i is the number of observations in the ith cluster and n is the number of clusters.

RMSE is employed as the evaluation metric for the LOS prediction model. Models with smaller RMSE values are considered to have better performance compared to those with larger RMSE values.

2.3.3.2 Area Under the Receiver Operating Characteristic (ROC) Curve (AUC)

The performance of asthma exacerbation prediction models was assessed and compared using the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC) as the evaluation metric. Models with higher AUC are considered to have better performance than those with lower AUC values.

2.3.3.3 Random Forests Feature Importance

For the regression problems, Random Forests features importance was measured by the percentage of increasing mean square error, referred to as "%IncMSE". This was calculated by measuring the difference in MSE before and after permuting the variable, which drops any potential relationship between the variable and the outcome.

For the classification problems, Random Forests features importance was measured by Mean Decrease Accuracy, which was calculated by measuring the difference in accuracy before and after the model excluding the variable.

3.0 Results

3.1 Descriptive Statistics

Table 3 and Table 4 present the summary statistics of lag and non-lag data for predicting LOS and asthma exacerbation, respectively. In contrast to the demographics table that is based on individual information, these two tables are derived from encounter level information.

Table 3. Encounters Information on Inpatient Data

	Lag Data (n = 1,611)	**Non-lag Data (n = 3,693)**
	Variable = TRUE	
Chronic Disease Existing	161 (10.0%)	255 (6.9%)
Action Plan	772 (47.9%)	1695 (45.9%)
Influenza Vaccine	1039 (64.5%)	2089 (56.6%)
Acute Bronchospasms	6 (0.4%)	61 (1.7%)
Subcutaneous Emphysema	10 (0.6%)	29 (0.8%)
During Pandemic	217 (13.5%)	437 (11.8%)
AsthmaAssessment_lag1	11 (0.7%)	/
Inpatient_lag1	527 (32.7%)	/
ED_lag1	1103 (68.5%)	/
Length of Stay (LOS)		
Mean (SD)	40.7 (26.9)	37.79 (25.5)
Median [Min, Max]	32.7[4.5, 166.5]	29.2 [4.5, 167.1]

Table 4. Encounters Information on Whole Data

	Lag Data (n = 18,300)	Non-lag Data (n = 31,168)
	Variable = TRUE	
ED	3834 (21.0%)	8571 (27.5%)
Exacerbation	3683 (20.1%)	8301 (26.7%)
Inpatient	1701 (9.3%)	3915 (12.6%)
Chronic Disease Existing	1414 (7.7%)	1934 (6.2%)
Asthma Assessment	315 (1.7%)	336 (1.1%)
Action Plan	7037 (38.5%)	11856 (38.0%)
Influenza Vaccine	12697 (69.4%)	19917 (63.9%)
Acute Bronchospasms	13 (0.1%)	90 (0.3%)
Subcutaneous Emphysema	15 (0.1%)	38 (0.1%)
During Pandemic	4575 (25.0%)	6327 (20.3%)
AsthmaAssessment_lag1	264 (1.4%)	/
Inpatient_lag1	2139 (11.7%)	/
ED lag1	4414 (24.1%)	/

3.2 LOS Prediction Results

3.2.1 Model Comparisons for Predicting LOS

Linear Regression, Linear Mixed Effects model, Random Forests, and Mixed Effects Random Forests are used to predict the continuous outcome LOS. Figure 7 shows the RMSE of four modeling methods applied on lag data or non-lag data.

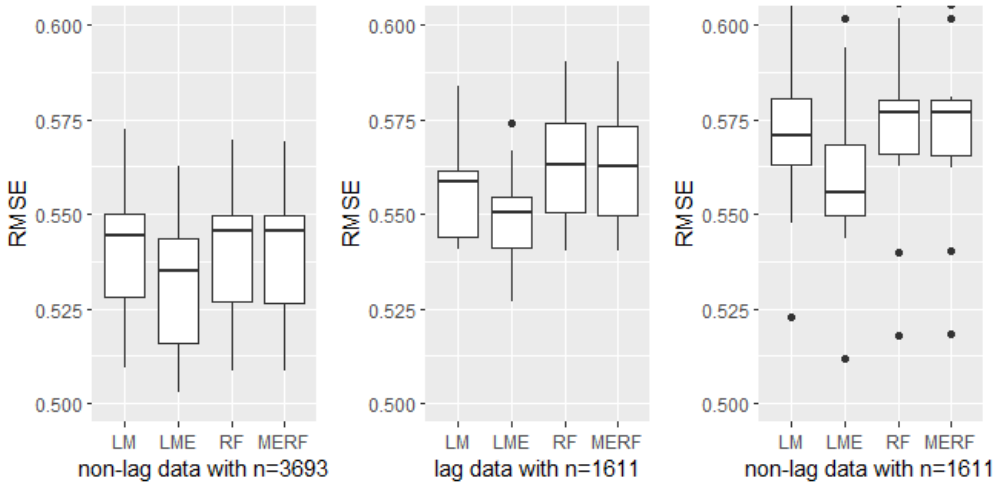


Figure 7. RMSE for 4 Modeling Methods Implemented on Data With or Without Lag Predictors Using Random Split

When comparing three scenarios, it was observed that all modeling methods had the lowest RMSE (indicating better performance) when applied to non-lag data, which might be due to the larger sample size (as compared to lag data). Additionally, when comparing data with lag predictors to data without lag predictors but with the same sample size, the models performed better with the inclusion of lag predictors, suggesting that incorporating patient health history can enhance model performance.

Furthermore, among the four modeling methods compared, LME demonstrated the best performance. RF and MERF had a very similar RMSE. The same pattern was observed in the date split result, which is shown in Figure 8.

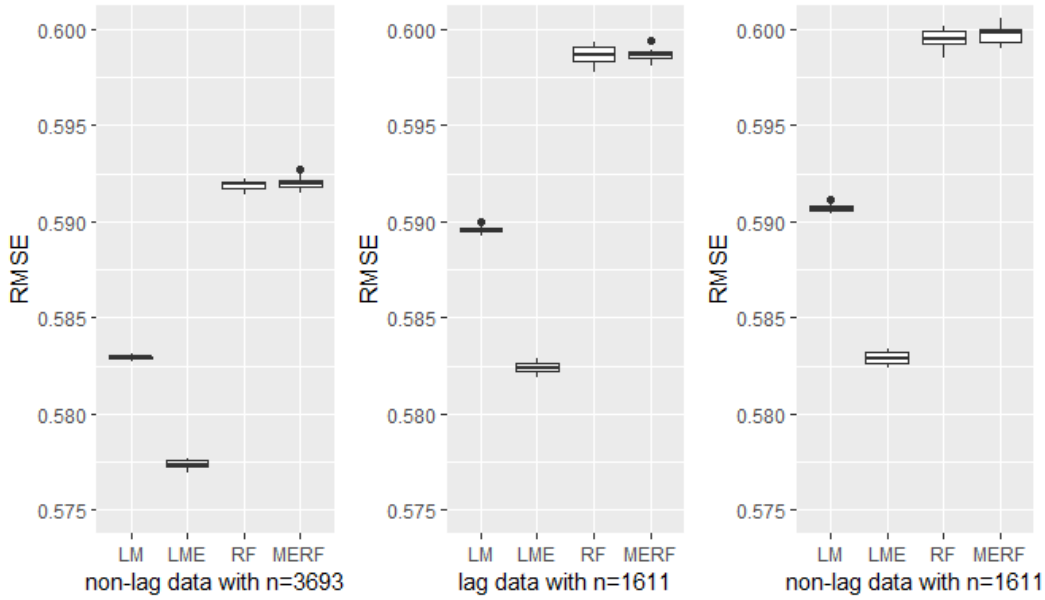


Figure 8. RMSE for 4 Modeling Methods Implemented on Data With or Without Lag Predictors Using Date Split

In contrast to random split methods, the box plot is significantly narrower when using a date split approach, as the training and test sets remain consistent throughout the entire modeling procedure.

3.2.2 Estimation of LOS Under the Best Prediction Model

Since the LME outperformed the other three modeling methods across all scenarios, the coefficients of the predictor were estimated using LME applied to the non-lag data with the full sample size (n=3,693), which is displayed in Table 5.

Table 5. LME for Predicting Log (LOS) on Non-lag Data With n=3,693

Predictor	Coefficient Estimate	95% CI	P
Sex (reference = Female)			
Male	-0.054	-0.093, -0.015	0.0060
State (reference = PA)			
OH	0.235	0.046, 0.424	0.0150
WV	0.315	0.099, 0.530	0.0042
Race (reference = White)			
Multi/Asian	-0.047	-0.132, 0.038	0.2793
Black	-0.015	-0.054, 0.025	0.4739
Chronic Disease Existing (reference = False)			
True	0.329	0.247, 0.410	<0.001
Influenza Vaccine (reference = False)			
True	0.035	-0.001, 0.072	0.0585
Patient Age	0.015	0.011, 0.020	<0.001
During Pandemic (reference = False)			
True	-0.149	-0.203, -0.096	<0.001
Action Plan (reference = False)			
True	0.188	0.154, 0.222	<0.001
Acute Bronchospasms (reference = False)			
True	0.096	-0.040, 0.232	0.1686
Subcutaneous Emphysema (reference = False)			
True	0.020	-0.174, 0.216	0.8396

When predicting LOS, several predictors were found to be significant with p-values less than 0.05, including sex, state, the existence of chronic disease, patient age, during pandemic, and action plan. Among these predictors, patient age, chronic diseases existing, during pandemic and action plan were the most significant predictors with p-values less than 0.001. The coefficient estimation indicated that older patients tended to have longer LOS when compared to other hospitalized patients. Similarly, patients with chronic diseases were also found to have longer LOS,

which is consistent with the fact that chronic diseases may result in more severe asthma symptoms and therefore lead to longer hospital stays. Additionally, there was a positive relationship between the action plan and LOS, suggesting that patients who received an action plan during their current visit tended to stay longer in the hospital. The Random Forests importance plot below (Figure 9) shows that the patient age, chronic diseases existing, and action plan are also the top predictors with high feature importance scores. The descriptive plot between the LOS and the predictors can be found in Appendix A.

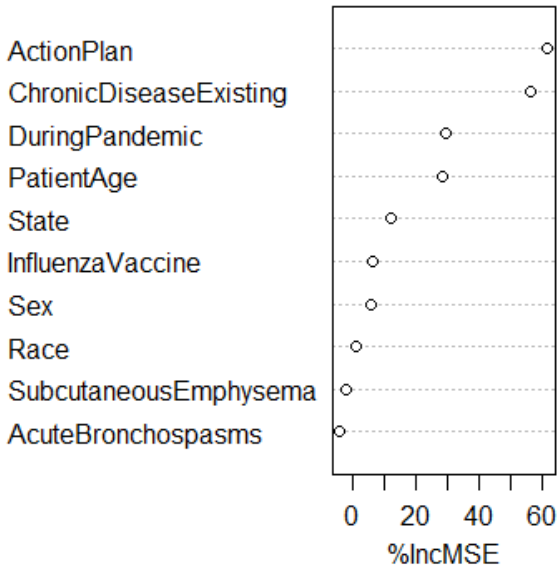


Figure 9. Random Forests Feature Importance Plot for LOS Prediction

3.2.3 Model Comparison for Predicting Asthma Exacerbation

Generalized Linear Regression, Generalized Linear Mixed Effects model, Random Forests, and Binary Mixed Model Forests are used to predict the binary outcome asthma exacerbation. Figure 10 shows the AUC of 4 modeling methods applied on lag or non-lag data.

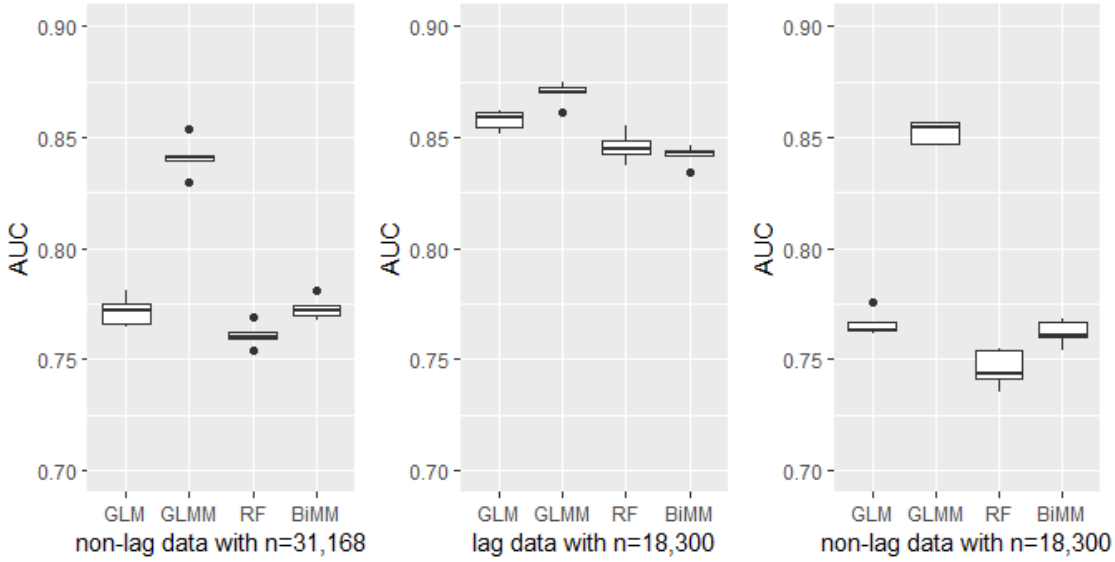


Figure 10. AUC for 4 Modeling Methods Implemented on Data With or Without Lag Predictors Using Random Split

When comparing three scenarios, it was observed that all modeling methods had the highest AUC (indicating better performance) when applied to lag data, which indicates that incorporating lag predictors can enhance model performance, even with a smaller sample size compared with non-lag data with nearly n=31,200.

Among the four modeling methods compared, GLMM demonstrated the best performance, especially when applied on the non-lag data. The same pattern can be observed in the date split result, which is shown in Figure 11.

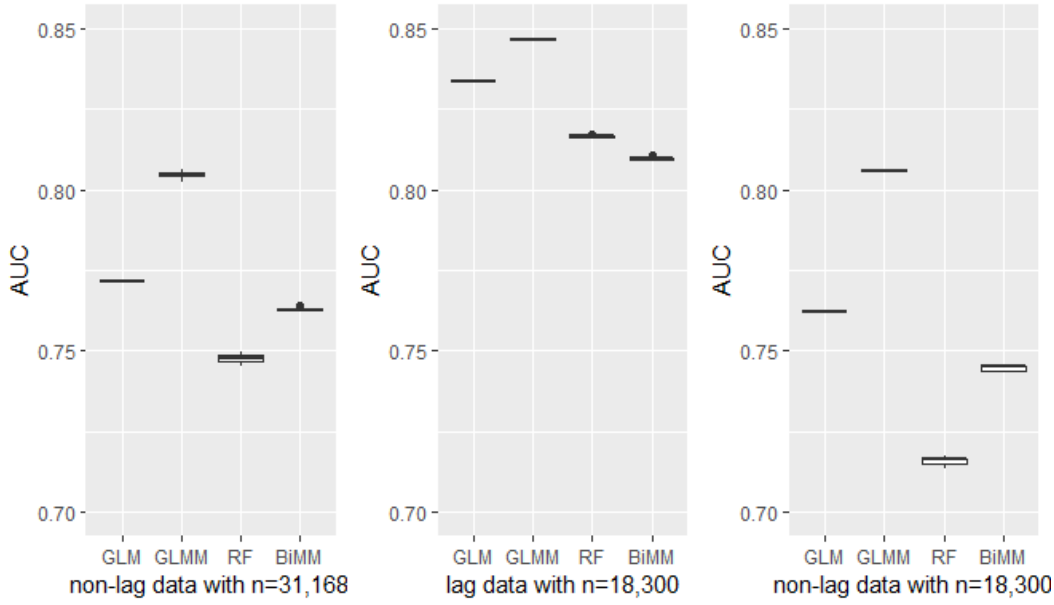


Figure 11. AUC for 4 Modeling Methods Implemented on Data With or Without Lag Predictors Using Date Split

3.2.4 Estimation of Asthma Exacerbation Under the Best Prediction Model

Since the GLMM outperformed the other three modeling methods across all scenarios, the predictor coefficients were estimated using GLMM applied to the lag data with the full sample size (n=18,300), which is displayed in Table 6.

Table 6. GLMM for Predicting Asthma Exacerbation on Lag Data with n=18,300

Predictor	Coefficient Estimate	95% CI	P
Sex (reference = Female)			
Male	-0.055	-0.174, 0.063	0.3595
State (reference = PA)			
OH	-2.371	-3.560, -1.143	<0.001
WV	-0.518	-1.280, 0.244	0.1826
Race (reference = White)			
Multi/Asian	0.429	0.171, 0.688	0.0011

Black	1.537	1.391, 1.682	<0.001
Chronic Disease Existing (reference = False)			
True	-0.054	-0.302, 0.193	0.6667
Influenza Vaccine (reference = False)			
True	-0.406	-0.520, -0.293	<0.001
Patient Age	-0.076	-0.089, -0.062	<0.001
During Pandemic (reference = False)			
True	-0.530	-0.661 -0.340	<0.001
Action Plan (reference = False)			
True	-1.241	-1.357, -1.124	<0.001
Acute Bronchospasms (reference = False)			
True	2.651	0.803, 4.498	0.0049
Subcutaneous Emphysema (reference = False)			
True	3.487	1.994, 4.979	<0.001
AsthmaAssessment_lag1 (reference = False)			
True	-0.333	-0.916, 0.250	0.2636
Inpatient_lag1 (reference = False)			
True	-0.429	-0.578, -0.281	<0.001
ED_lag1 (reference = False)			
True	2.209	2.078 2.340	<0.001

Among all the predictors, race, during pandemic, patient age, influenza vaccine, action plan, subcutaneous emphysema, ED_lag1, and Inpatient_lag1 were found to be the most significant predictors with p-values less than 0.001. The coefficient estimation revealed that younger patients had a higher probability of having asthma exacerbation. Similarly, patients who did not receive the influenza vaccine, patients from minority races, or patients with subcutaneous emphysema were also more likely to experience exacerbations. However, patients who had been previously hospitalized tended to have a lower probability of experiencing exacerbations, possibly due to the appropriate treatments or care they received during their previous hospital stay. Furthermore, patients who were given an action plan or had their encounter visit during the pandemic were less

likely to experience exacerbations, suggesting that timely intervention and proactive measures may have helped in managing their asthma condition, and patients were more cautious about visiting ED due to the fear of the pandemic. The Random Forests importance plot below (Figure 12) shows that the two lag predictors (previous ED visit and inpatient visit) are among the top four predictors with high feature importance scores. The descriptive plot between the asthma exacerbation and the most significant predictors can be found in Appendix A.

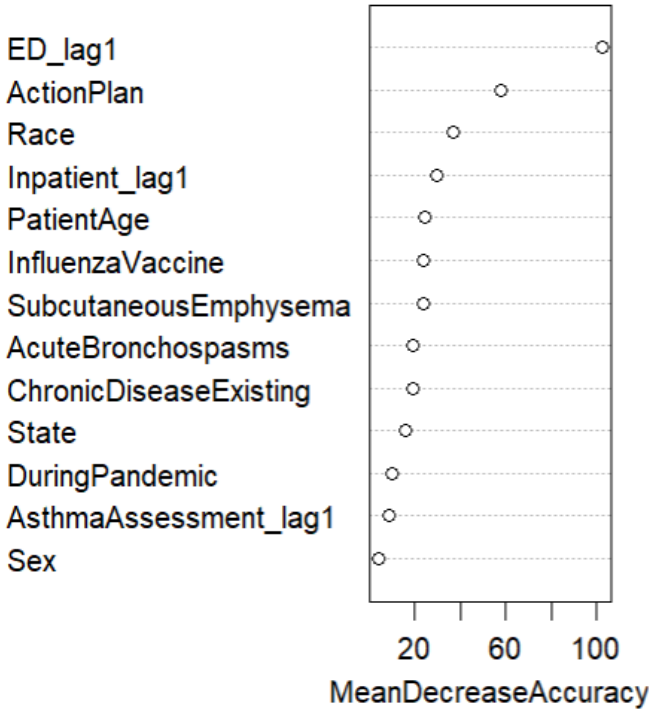


Figure 12. Random Forests Feature Importance Plot for Asthma Exacerbation Prediction

4.0 Discussion

The primary goals of this research are to develop predictive models for severe asthma outcomes, including inpatient length of stay and asthma exacerbation, and to identify risk factors contributing to severe outcomes using the EHR data from the Children's Hospital of Pittsburgh.

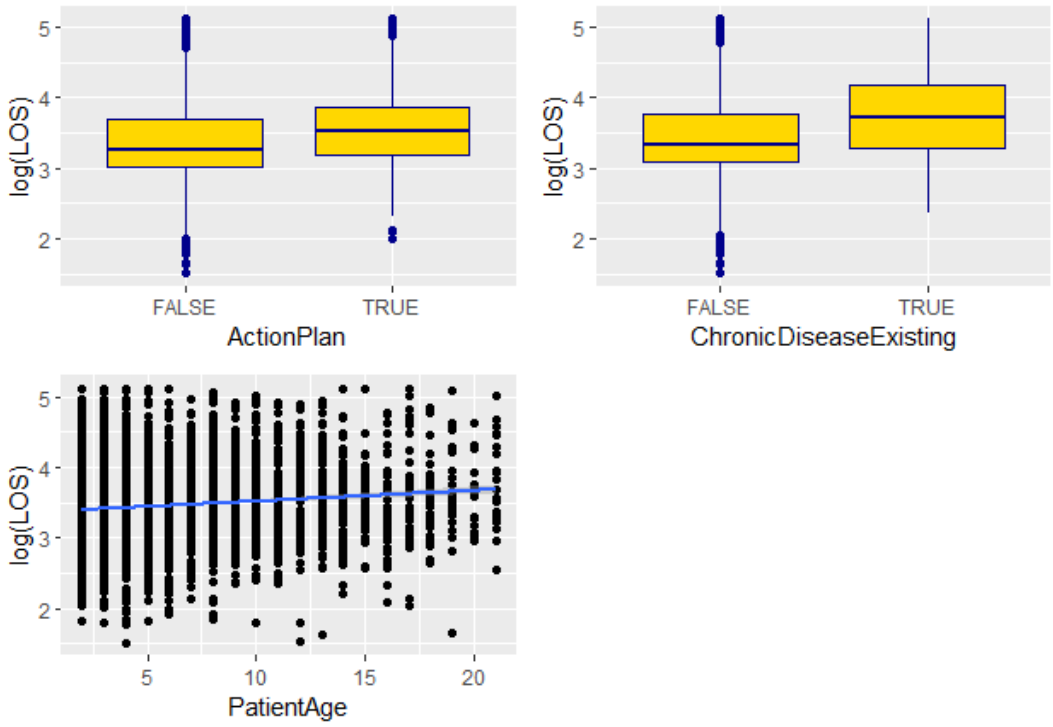
Four different modeling methods, particularly those incorporating mixed effects, were employed for each outcome. Overall, mixed effects methods showed superior performance compared to fixed effects methods. Linear mixed effects modeling methods had better performance than tree-based mixed effects modeling methods, potentially due to the high proportion of binary predictors in the data. Although the Random Forests algorithm is able to split the nodes by calculating the Gini impurity score of the continuous features, it always splits the nodes based on True or False in this context, which means the high proportion of the binary predictors may limit the performance of RF. Furthermore, it should be noted that in comparison to the sample size of the data, particularly when predicting asthma exacerbation, the number of predictors was relatively small. RF typically is more powerful when the number of predictors is large, as it can randomly select a subset of predictors when constructing trees. However, in this case, the number of predictors was small as compared to the sample size, which may have limited the potential advantages of using RF in the modeling process.

The inclusion of lag predictors, such as Inpatient_lag1 and ED_lag1, significantly improved the performance of the models, both in terms of LOS prediction and asthma exacerbation prediction. This suggests that patient health history information from the EHR data is useful for predicting future outcomes and incorporating them is recommended when constructing predictive models using EHR data. There are limitations to the approach of creating lag predictors.

Specifically, we did not take into consideration the specific time interval between the previous and current visits. For example, even though patients may have had multiple encounters during the entire study period, the time gap between these visits could be large. In that case, the information from the previous encounter may be less relevant to the current visit, resulting in less accurate predictions. Besides, the lag dataset itself excluded those patients who visited the hospital only one time, which may lead to a biased target population in the analysis. In the future, a time-to-event type of analysis (e.g., time-to-exacerbation) with possible recurrent events may be desired to further investigate the dynamic prediction of asthma outcomes.

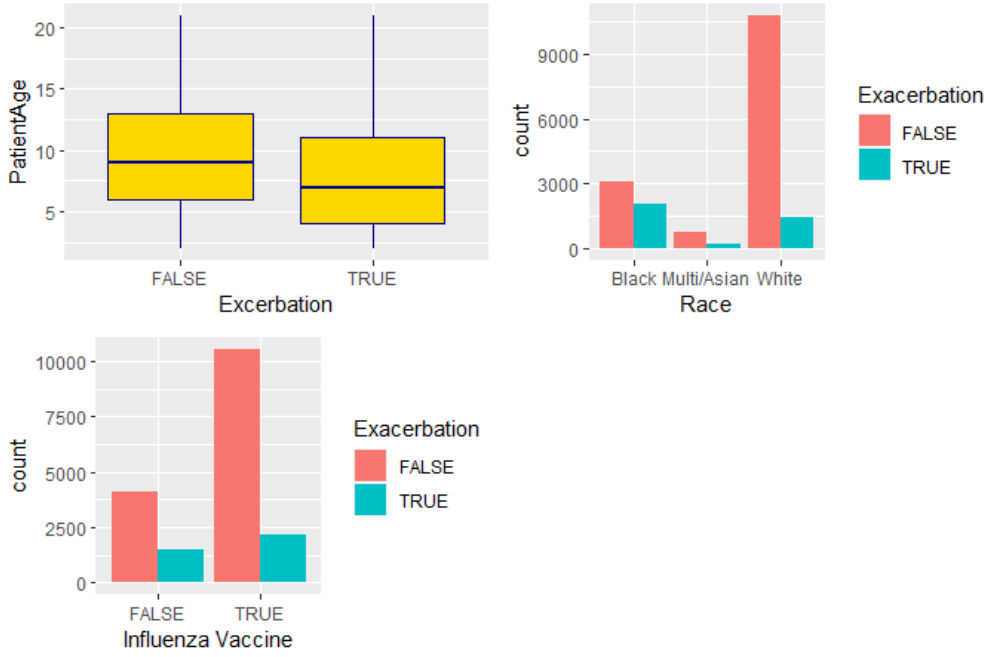
The analysis is also limited by a large amount of missing data in the EHR, which prevents the inclusion of potential useful predictors such as medication doses and lab results. For the predictors used in this study, action plan (yes or no) is a controversial predictor since the coefficient of it appeared to be opposite compared between LOS prediction and asthma exacerbation prediction. This discrepancy may be due to the fact that the action plan variable in the dataset reflects the asthma condition during the current visit, rather than an intervention plan with long-term impact. Despite these limitations, the action plan variable remains an important predictor of interest in our analysis. Furthermore, “during pandemic”, which is one of the most significant predictors, may not be suitable for future predictions as the current pandemic has ended and the value of this predictor would no longer exhibit variability in predicting future asthma outcomes.

Appendix A Descriptive Plot



Appendix Figure 1. The Descriptive Plots Between the log(LOS) And the Most Significant Predictors

Appendix Figure 2. The Descriptive Plots Between Asthma Exacerbation And the Most Significant Predictors



Appendix Figure 3. The Descriptive Plots Between Asthma Exacerbation And the Most Significant Predictors

Appendix B Analysis Executed in R

```
 # lag data create function

 createlag<-function(df,lag_col=c("Inpatient")){
   df<-df[order(df$PersonID,df$RegistrationDT3 ),]
lag<-data.frame(matrix(nrow = nrow(df),
     ncol = ncol(df)+length(lag_col)),stringsAsFactors=FALSE)

   lag_col_name <- paste(lag_col,"lag1",sep="_")
   colnames(lag)<-c(colnames(df),lag_col_name)

   prev_id <-df$PersonID[1]
   begin_row_idx <- 1
   end_row_idx <- 1
   v <- data.frame(matrix(nrow=1,ncol=length(lag_col)), stringsAsFactors=FALSE)
   colnames(v)<-lag_col
   for(j in 1:nrow(df)){
     current_id <- df$PersonID[j]

     if(current_id == prev_id){
       end_row_idx <- j
     } else {
       # slice data frame of the same person id
       data <- df[begin_row_idx:end_row_idx,]

       # remove some columns
       data <- subset(data,select = lag_col)

       #combine rows
       data<-rbind.data.frame(v,data) # add the first NA row
       data<-data[-nrow(data),] # remove the last row
   lag[begin_row_idx:end_row_idx,]<-
                 as.matrix(cbind.data.frame(df[begin_row_idx:end_row_idx,],data))

       # next group
       begin_row_idx <- j
       end_row_idx <- j
     }

     if(j == nrow(df)){
       # slice data frame of the same Person id
       data <- df[begin_row_idx:end_row_idx,]
```

```
            # remove some columns
            data <- subset(data,select = lag_col)

            #combine rows
            data<-rbind.data.frame(v,data) # add the first NA row
            data<-data[-nrow(data),] # remove the last row
            lag[begin_row_idx:end_row_idx,]<-
as.matrix(cbind.data.frame(df[begin_row_idx:end_row_idx,],data))
          }
          prev_id <- current_id
          if (j%%1000 ==0){
            print(j)
          }

        }
        return(lag)
      }


      # MERF function from the author of reference [12]

      MERF <- function(
        xnam
        ,MERF.lDB
        ,ni
        ,Zi
        ,Yi
        ,ntree
        ,mtry
        ,nodesize

        ,sigmasqzero = NULL
        ,Dzero = NULL
        ,bizero = NULL

        ,F.niter
        ,max.niter
        ,smallest.Jump.allowed

        ,verbose = TRUE

      ){

        ####################
        ####STEP 0####
```

```
####################

#Memory Allocation and initialization:

#Parameters values
n <- length(ni)
N <- sum(ni)
q <- dim(Zi[[1]])[2] # q=1 in random intercept case


#Initial values of sigmasqzero, Dzero, and bizero
if( is.null(sigmasqzero) ) sigmasqzero <- 1
else sigmasqzero <- sigmasqzero

if( is.null(Dzero) ){
  Dzero <- diag(0.01 ,nrow=q,ncol=q)
}
else Dzero <- Dzero

if( is.null(bizero) ){
  bizero <- list(); length(bizero)<- n
  for(i in 1:n) bizero[[i]] <- matrix(0,nrow=q,ncol=1)
}
else bizero <- bizero

#iter number
r <- 1
if (verbose)
  message("MERF iter no: ", r)

#transformed outcome, star.Yi[[r]][[i]], initialized with the original values
star.Yi <- list()
for(i in 1:n){
  star.Yi[[i]] <- Yi[[i]] - Zi[[i]] %*% bizero[[i]]
}

#one STD random forest
######################

MERF.lDB$star.Yi <- unlist(star.Yi)
rm(star.Yi) ;  gc(verbose=FALSE)

fit.rf.formula <-  as.formula(paste("star.Yi ~ ", paste(xnam, collapse= "+")))

fit.rf <- randomForest(
  formula=fit.rf.formula
```

```
  ,data=MERF.lDB
  ,ntree=ntree
  ,mtry = mtry
  ,replace=TRUE
  ,nodesize = nodesize
  ,proximity=FALSE
)

#fixed part
#as vector
MERF.lDB$f.pred  <- predict(fit.rf, type="response")#!!! use the out-of-bag predictions

#in matrix format
fixed.pred <- list()
fixed.pred <- split(MERF.lDB, MERF.lDB$cluster.id)
for(i in 1:n)fixed.pred[[i]] <- as.matrix(subset(fixed.pred[[i]] ,select=f.pred), ncol=1)

#randompart
#############
#random effects parameters in list format
bi <- list(list()) ; length(bi) <- r
for(i in 1:n)bi[[r]][[i]] <- bizero[[i]]
#print("bizero");print(bizero)
rm(bizero) ; gc(verbose=FALSE)

#level-1 variance component
#residuals
epsili <- list()
for(i in 1:n)
  epsili[[i]] <- Yi[[i]] - fixed.pred[[i]] - Zi[[i]] %*% bi[[r]][[i]]

sigma.sq <- vector(mode="numeric") ;length(sigma.sq) <- r
sigma.sq[r] <- sigmasqzero
#print("sigmasqzero") ;print(sigmasqzero)
rm(sigmasqzero) ; gc(verbose=FALSE)
#message("sigmasq of current micro iter", sigma.sq[r] )

#level-2 variance component
D <- list() ;length(D) <- r
D[[r]] <- Dzero#!!!Dzero <- diag(x=0.01, nrow=q, ncol = q)
#print("Dzero") ;print(Dzero)
rm(Dzero) ; gc(verbose=FALSE)
#message("D of current micro iter: ", D[[r]] )

#level-1 and level-2 variance components (or typical or total variance)
Vi <- list()
```

```
        inv.Vi <- list(list()) ; length(inv.Vi) <- r

        for(i in 1:n){
          Vi[[i]] <- Zi[[i]] %*% D[[r]] %*% t(Zi[[i]]) + sigma.sq[r]*diag(x = 1, nrow=ni[i], ncol
= ni[i])
          if(q==1)
            inv.Vi[[r]][[i]] <-
              (1/sigma.sq[r]) * (diag(rep(1,ni[i]))
                                 -
((as.numeric(D[[r]])/sigma.sq[r])/(1+ni[i]*(as.numeric(D[[r]])/sigma.sq[r])))*matrix(rep(1,(ni[i])
^2)
                                                                                    , ncol=ni[i],
nrow=ni[i]) )
          else inv.Vi[[r]][[i]] <- solve(Vi[[i]])
        }

        Vi <- list(NULL)
        #inv.Vi[[r-1]] <- list(NULL) #not to run at step 0

        #the generalized log-likelihood (GLL)
        GLL <- vector(mode="numeric") ; length(GLL) <- r
        term <- vector(mode="numeric",length=n)
        for(i in 1:n)
          term[i]<-t(epsili[[i]]) %*% solve(sigma.sq[r]*diag(x=1,nrow=ni[i],ncol=ni[i])) %*%
epsili[[i]]
        + t(bi[[r]][[i]]) %*% solve(D[[r]]) %*% bi[[r]][[i]]
        + log(abs(D[[r]]))
        + log(abs(sigma.sq[r]*diag(x=1, nrow=ni[i], ncol = ni[i])))
        GLL[r] <- sum(term)
        rm(term)
        gc(verbose=FALSE)

        #convergence criterion
        Jump <- rep(NA,r) #at this first iteration Jump = NA
        convergence.iter<- rep(NA,r) #at this first convergence.iter = NA

        ####################
        ####STEP 1####
        ####################

        #update iteration number r
        r <- r+1
        if (verbose)
          message("MERF iter no: ", r)

        #update the length of the different lists
```

```
length(sigma.sq) <- r
length(D) <- r
length(inv.Vi) <- r
length(bi) <- r
length(GLL) <- r

length(Jump) <- r
length(convergence.iter) <- r

#update the transformed outcome, star.Yi
star.Yi <- list()
for(i in 1:n){
  star.Yi[[i]] <- Yi[[i]] - Zi[[i]] %*% bi[[r-1]][[i]]
}

#one STD random forest
######################

MERF.lDB$star.Yi <- unlist(star.Yi)
rm(star.Yi) ;  gc(verbose=FALSE)


fit.rf <- randomForest(
  formula=fit.rf.formula
  ,data=MERF.lDB
  ,ntree=ntree
  ,mtry = mtry
  ,replace=TRUE
  ,nodesize = nodesize
  ,proximity=FALSE
)

#fixed part
#as vector
MERF.lDB$f.pred  <- predict(fit.rf, type="response")

#in matrix format
fixed.pred <- list()
fixed.pred <- split(MERF.lDB, MERF.lDB$cluster.id)
for(i in 1:n)fixed.pred[[i]] <- as.matrix(subset(fixed.pred[[i]] ,select=f.pred), ncol=1)

#randompart
############
for(i in 1:n)
```

```
    bi[[r]][[i]] <- D[[r-1]]%*%t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]] %*% (Yi[[i]] -
fixed.pred[[i]])

  bi[r-1] <- list(NULL)

  ####################
  ####STEP 2####
  ####################

  #level-1 variance component
  #residuals
  epsili <- list()
  for(i in 1:n)
    epsili[[i]] <- Yi[[i]] - fixed.pred[[i]] - Zi[[i]] %*% bi[[r]][[i]]


  term <- vector(mode="numeric",length=n)
  for(i in 1:n)
    term[i] <- crossprod(epsili[[i]]) +
    sigma.sq[r-1] * (ni[i] - sigma.sq[r-1]* sum(diag(inv.Vi[[r-1]][[i]])))
  sigma.sq[r] <- (1/N)*(sum(term))
  rm(term) ;gc(verbose=FALSE)
  #message("sigmasq of current micro iter", sigma.sq[r] )

  #level-2 variance component
  term <- list()
  term[[1]] <- tcrossprod(bi[[r]][[1]]) +
    (D[[r-1]] -
       D[[r-1]] %*% t(Zi[[1]])%*% inv.Vi[[r-1]][[1]] %*% Zi[[1]] %*% D[[r-1]]
    )
  for(i in 2:n)
    term[[i]] <- term[[i-1]]+ tcrossprod(bi[[r]][[i]]) +
    (D[[r-1]] -
       D[[r-1]] %*% t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]]%*% Zi[[i]]%*% D[[r-1]]
    )
  term <- term[[n]]
  D[[r]] <- (1/n)*term
  rm(term) ;gc(verbose=FALSE)
  #message("D of current micro iter: ", D[[r]] )

  #level-1 and level-2 variance components (or typical or total variance)
  inv.Vi[[r]] <-list()
  for(i in 1:n){
    Vi[[i]] <- Zi[[i]] %*% D[[r]] %*% t(Zi[[i]])+sigma.sq[r]*diag(x = 1, nrow=ni[i], ncol
= ni[i])
    if(q==1)
```

```
            inv.Vi[[r]][[i]] <-
              (1/sigma.sq[r]) * (diag(rep(1,ni[i]))
                                 -
((as.numeric(D[[r]])/sigma.sq[r])/(1+ni[i]*(as.numeric(D[[r]])/sigma.sq[r])))*matrix(rep(1,(ni[i])
^2)
                                                                                , ncol=ni[i],
nrow=ni[i]) )
             else inv.Vi[[r]][[i]] <- solve(Vi[[i]])
            }
            Vi <- list(NULL)
            inv.Vi[[r-1]] <- list(NULL) #not to run at step 0


            #the generalized log-likelihood (GLL)
            term <- vector(mode="numeric",length=n)
            for(i in 1:n)
              term[i]<-t(epsili[[i]]) %*% solve(sigma.sq[r]*diag(x=1,nrow=ni[i],ncol=ni[i])) %*%
epsili[[i]]
            + t(bi[[r]][[i]]) %*% solve(D[[r]]) %*% bi[[r]][[i]]
            + log(abs(D[[r]]))
            + log(abs(sigma.sq[r]*diag(x=1, nrow=ni[i], ncol = ni[i])))
            GLL[r] <- sum(term)
            rm(term)
            gc(verbose=FALSE)

            #update the value of the Jump in GLL
            Jump[r] <- abs( (GLL[r]- GLL[r-1])/GLL[r] )

            if(Jump[r] < smallest.Jump.allowed | Jump[r] == smallest.Jump.allowed) {
              convergence.iter[r] <- r
              if (verbose) message("Converg. at iter no: ", r)
            }


            ####################
            ####STEP 3####
            ####################


            ######################################################
            #Iterating "F.niter" times to avoid early stopping#
            ######################################################

            for(I in 1:F.niter){#repeat step 1 and 2

              ####################
```

```
####STEP 1####
####################

#update iteration number r
r <- r+1
if (verbose)
  message("MERF iter no: ", r)

#update the length of the different lists

length(sigma.sq) <- r
length(D) <- r
length(inv.Vi) <- r
length(bi) <- r
length(GLL) <- r

length(Jump) <- r
length(convergence.iter) <- r

#update the transformed outcome, star.Yi
star.Yi <- list()
for(i in 1:n){
  star.Yi[[i]] <- Yi[[i]] - Zi[[i]] %*% bi[[r-1]][[i]]
}

#one STD random forest
######################

MERF.lDB$star.Yi <- unlist(star.Yi)
rm(star.Yi) ;  gc(verbose=FALSE)

fit.rf <- randomForest(
  formula=fit.rf.formula
  ,data=MERF.lDB
  ,ntree=ntree
  ,mtry = mtry
  ,replace=TRUE
  ,nodesize = nodesize
  ,proximity=FALSE
)

#fixed part
#as vector
MERF.lDB$f.pred  <- predict(fit.rf, type="response")

#in matrix format
```

```
        fixed.pred <- list()
        fixed.pred <- split(MERF.lDB, MERF.lDB$cluster.id)
        for(i in 1:n)fixed.pred[[i]] <- as.matrix(subset(fixed.pred[[i]] ,select=f.pred), ncol=1)

        #randompart
        ############
        for(i in 1:n)
          bi[[r]][[i]] <- D[[r-1]]%*%t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]] %*% (Yi[[i]] -
fixed.pred[[i]])
        bi[r-1] <- list(NULL)


        ####################
        ####STEP 2####
        ####################

        #level-1 variance component
        #residuals
        epsili <- list()
        for(i in 1:n)
          epsili[[i]] <- Yi[[i]] - fixed.pred[[i]] - Zi[[i]] %*% bi[[r]][[i]]


        term <- vector(mode="numeric",length=n)
        for(i in 1:n)
          term[i] <- crossprod(epsili[[i]]) +
          sigma.sq[r-1] * (ni[i] - sigma.sq[r-1]* sum(diag(inv.Vi[[r-1]][[i]])))
        sigma.sq[r] <- (1/N)*(sum(term))
        rm(term) ;gc(verbose=FALSE)
        #message("sigmasq of current micro iter", sigma.sq[r] )

        #level-2 variance component
        term <- list()
        term[[1]] <- tcrossprod(bi[[r]][[1]]) +
          (D[[r-1]] -
             D[[r-1]] %*% t(Zi[[1]])%*% inv.Vi[[r-1]][[1]] %*% Zi[[1]] %*% D[[r-1]]
          )
        for(i in 2:n)
          term[[i]] <- term[[i-1]]+ tcrossprod(bi[[r]][[i]]) +
          (D[[r-1]] -
             D[[r-1]] %*% t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]]%*% Zi[[i]]%*% D[[r-1]]
          )
        term <- term[[n]]
        D[[r]] <- (1/n)*term
        rm(term) ;gc(verbose=FALSE)
        #message("D of current micro iter: ", D[[r]] )
```

```
            #level-1 and level-2 variance components (or typical or total variance)
            inv.Vi[[r]] <-list()
            for(i in 1:n){
              Vi[[i]] <- Zi[[i]] %*% D[[r]] %*% t(Zi[[i]])+sigma.sq[r]*diag(x = 1, nrow=ni[i], ncol
= ni[i])
              if(q==1)
                inv.Vi[[r]][[i]] <-
                  (1/sigma.sq[r]) * (diag(rep(1,ni[i]))
                                     -
((as.numeric(D[[r]])/sigma.sq[r])/(1+ni[i]*(as.numeric(D[[r]])/sigma.sq[r])))*matrix(rep(1,(ni[i])
^2)
                                                                                    , ncol=ni[i],
nrow=ni[i]) )
              else inv.Vi[[r]][[i]] <- solve(Vi[[i]])
            }
            Vi <- list(NULL)
            inv.Vi[[r-1]] <- list(NULL) #not to run at step 0


            #the generalized log-likelihood (GLL)
            term <- vector(mode="numeric",length=n)
            for(i in 1:n)
              term[i]<-t(epsili[[i]]) %*% solve(sigma.sq[r]*diag(x=1,nrow=ni[i],ncol=ni[i])) %*%
epsili[[i]]
            + t(bi[[r]][[i]]) %*% solve(D[[r]]) %*% bi[[r]][[i]]
            + log(abs(D[[r]]))
            + log(abs(sigma.sq[r]*diag(x=1, nrow=ni[i], ncol = ni[i])))
            GLL[r] <- sum(term)
            rm(term)
            gc(verbose=FALSE)

            #update the value of the Jump in GLL

            Jump[r] <- abs( (GLL[r]- GLL[r-1])/GLL[r] )

            if(Jump[r] < smallest.Jump.allowed | Jump[r] == smallest.Jump.allowed) {
              convergence.iter[r] <- r
              if (verbose) message("Converg. at iter no: ", r)
            }

          }
          #end for (I in 1: F.niter)
          ############################

          ########################################
```

```
#Iterating "max.niter" times at most #
#######################################

while( r < (2 + F.niter + max.niter) ){

  if(Jump[r] > smallest.Jump.allowed){ #repeat step 1 and 2

    ####################
    ####STEP 1####
    ####################

    #update iteration number r
    r <- r+1
    if (verbose)
      message("MERF iter no: ", r)

    #update the length of the different lists

    length(sigma.sq) <- r
    length(D) <- r
    length(inv.Vi) <- r
    length(bi) <- r
    length(GLL) <- r

    length(Jump) <- r
    length(convergence.iter) <- r

    #update the transformed outcome, star.Yi
    star.Yi <- list()
    for(i in 1:n){
      star.Yi[[i]] <- Yi[[i]] - Zi[[i]] %*% bi[[r-1]][[i]]
    }

    #one STD random forest
    ######################

    MERF.lDB$star.Yi <- unlist(star.Yi)
    rm(star.Yi) ;  gc(verbose=FALSE)

    fit.rf <- randomForest(
      formula=fit.rf.formula
      ,data=MERF.lDB
      ,ntree=ntree
      ,mtry = mtry
      ,replace=TRUE
      ,nodesize = nodesize
```

```
  ,proximity=FALSE
 )

 #fixed part
 #as vector
 MERF.lDB$f.pred  <- predict(fit.rf, type="response")

 #in matrix format
 fixed.pred <- list()
 fixed.pred <- split(MERF.lDB, MERF.lDB$cluster.id)
 for(i in 1:n)fixed.pred[[i]] <- as.matrix(subset(fixed.pred[[i]] ,select=f.pred), ncol=1)

 #randompart
 ############
 for(i in 1:n)
  bi[[r]][[i]] <- D[[r-1]]%*%t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]] %*% (Yi[[i]] -
fixed.pred[[i]])
 bi[r-1] <- list(NULL)


 ####################
 ####STEP 2####
 ####################

 #level-1 variance component
 #residuals
 epsili <- list()
 for(i in 1:n)
  epsili[[i]] <- Yi[[i]] - fixed.pred[[i]] - Zi[[i]] %*% bi[[r]][[i]]


 term <- vector(mode="numeric",length=n)
 for(i in 1:n)
  term[i] <- crossprod(epsili[[i]]) +
  sigma.sq[r-1] * (ni[i] - sigma.sq[r-1]* sum(diag(inv.Vi[[r-1]][[i]])))
 sigma.sq[r] <- (1/N)*(sum(term))
 rm(term) ;gc(verbose=FALSE)
 #message("sigmasq of current micro iter", sigma.sq[r] )

 #level-2 variance component
 term <- list()
 term[[1]] <- tcrossprod(bi[[r]][[1]]) +
  (D[[r-1]] -
     D[[r-1]] %*% t(Zi[[1]])%*% inv.Vi[[r-1]][[1]] %*% Zi[[1]] %*% D[[r-1]]
  )
 for(i in 2:n)
```

```
  term[[i]] <- term[[i-1]]+ tcrossprod(bi[[r]][[i]]) +
  (D[[r-1]] -
     D[[r-1]] %*% t(Zi[[i]]) %*% inv.Vi[[r-1]][[i]]%*% Zi[[i]]%*% D[[r-1]]
  )
term <- term[[n]]
D[[r]] <- (1/n)*term
rm(term) ;gc(verbose=FALSE)
#message("D of current micro iter: ", D[[r]] )

#level-1 and level-2 variance components (or typical or total variance)
inv.Vi[[r]] <-list()
for(i in 1:n){
  Vi[[i]] <- Zi[[i]] %*% D[[r]] %*% t(Zi[[i]])+sigma.sq[r]*diag(x = 1, nrow=ni[i],
ncol = ni[i])
  if(q==1)
    inv.Vi[[r]][[i]] <-
      (1/sigma.sq[r]) * (diag(rep(1,ni[i]))
                         -
((as.numeric(D[[r]])/sigma.sq[r])/(1+ni[i]*(as.numeric(D[[r]])/sigma.sq[r])))*matrix(rep(1,(ni[i])
^2)
                                                                  , ncol=ni[i],
nrow=ni[i]) )
  else inv.Vi[[r]][[i]] <- solve(Vi[[i]])
}
Vi <- list(NULL)
inv.Vi[[r-1]] <- list(NULL) #not to run at step 0


#the generalized log-likelihood (GLL)
term <- vector(mode="numeric",length=n)
for(i in 1:n)
  term[i]<-t(epsili[[i]]) %*% solve(sigma.sq[r]*diag(x=1,nrow=ni[i],ncol=ni[i]))
%*% epsili[[i]]
+ t(bi[[r]][[i]]) %*% solve(D[[r]]) %*% bi[[r]][[i]]
+ log(abs(D[[r]]))
+ log(abs(sigma.sq[r]*diag(x=1, nrow=ni[i], ncol = ni[i])))
GLL[r] <- sum(term)
rm(term)
gc(verbose=FALSE)

#update the value of the Jump in GLL

Jump[r] <- abs( (GLL[r]- GLL[r-1])/GLL[r] )

if(Jump[r] < smallest.Jump.allowed | Jump[r] == smallest.Jump.allowed) {
  convergence.iter[r] <- r
```

```
    if (verbose) message("Converg. at iter no: ", r)
   }


  }
  #end if(Jump[r] > smallest.Jump.allowed) and STOP repeating step 1 and 2

  else break
  #end while( r < (2 + F.niter + max.niter) )


  ############################
}###END OF STEP 3####
############################


#output to be returned (MERF model is the one at the last iteration)
######################


output <- list(
  Jump[r]
  ,GLL
  ,convergence.iter
  ,fit.rf
  ,bi[[r]]
  ,sigma.sq[r]
  ,D#,D[[r]]
)

names(output) <- c(
  "Jump[r]"
  ,"GLL"
  ,"convergence.iter"
  ,"fit.rf"
  ,"bi[[r]]"
  ,"sigma.sq[r]"
  ,"D"#,"D[[r]]"
)

#clean memory
#############
rm(
  xnam

  ,MERF.lDB
```

```
    ,ni,n,N
    ,Zi,q
    ,Yi

    ,ntree
    ,mtry
    ,nodesize
    ,fit.rf.formula
    ,fit.rf
    ,fixed.pred ,epsili

    ,sigma.sq,D,bi,Vi,inv.Vi

    ,F.niter ,max.niter
    ,smallest.Jump.allowed ,GLL ,Jump ,convergence.iter
    ,r,i,I

    ,verbose
   )
   gc(verbose=FALSE)


   #return
   #######
   output

  }


  # BiMM Forests function from the author of reference [13]

  library(blme)
  library(rpart)
  library(randomForest)
  bimmForest<-
function(formula,random,traindata,seed=8636,method="1iter",h1c=0.5,h2c=1.5,ErrorTolerance=
0.5,MaxIterations=100,verbose=TRUE){
    #rename the dataset
    data=traindata
    #parse formula
    Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
    TargetName<-formula[[2]]
    Target<-data[,toString(TargetName)]
    initialRandomEffects=rep(0,length(data[,1]))
    #set up variables for loop
```

```
ContinueCondition<-TRUE
iterations<-0
#initial values
AdjustedTarget<-as.numeric(Target)-initialRandomEffects
oldlik<- -Inf
# Make a new data frame to include all the new variables
newdata <- data

while(ContinueCondition){
  # Current values of variables
  newdata[,"AdjustedTarget"] <- AdjustedTarget
  iterations <- iterations+1
  #build forest
  set.seed(seed)
  forest <- randomForest(formula(paste(c("factor(AdjustedTarget)", Predictors),collapse
= "~")),
                         data = data, mtry=5, ntree=500,method = "class")
  forestprob<-predict(forest,type="prob")[,2]
  if(verbose){
    print(paste(c("Iteration:",iterations)))
    print(forest)
  }
  ## Estimate New Random Effects and Errors using BLMER
  options(warn=-1)
  lmefit                                                                    <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),"forestprob"),      collapse="~"),
"+(1|random)",sep=""))),

data=data,family=binomial,control=glmerControl(optCtrl=list(maxfun=20000))),error=function(
cond)"skip")
  if(verbose){
    print(paste(c("Iteration:",iterations)))
    print(lmefit)
  }
  # Get the likelihood to check on convergence
  if(!(class(lmefit)[1]=="character")){
    newlik <- logLik(lmefit)
    ContinueCondition <- (abs(newlik-oldlik)>ErrorTolerance & iterations <
MaxIterations)
    oldlik <- newlik
    # Extract random effects to make the new adjusted target
    logit<-forestprob
    logit2<-exp(predict(lmefit))/(1+exp(predict(lmefit)))
    AllEffects <- logit2
    #1 iteration, ignore adjusted target
    if(method=="1iter") {
```

```
            ContinueCondition<-FALSE
          }
          if(method=="h1"){
            AdjustedTarget <- ifelse(as.numeric(AdjustedTarget) + AllEffects-1>h1c,1,0)
          }
          if(method=="h2"){
            AdjustedTarget <- ifelse(as.numeric(AdjustedTarget) + AllEffects-1<h2c,0,1)
          }
          if(method=="h3"){
            for(k in 1:length(AllEffects)){
              if(as.numeric(Target[k])+AllEffects[k]-1<.5){AdjustedTarget[k]=0}
              else if(as.numeric(Target[k])+AllEffects[k]-1>1.5){AdjustedTarget[k]=1}
              else{
                #generate random probability coin flip based on AllEffects (q notation in paper)
                AdjustedTarget[k]<-rbinom(1,1,AllEffects[k])
              }
            }
          }
          #check to see if updated outcomes are the same, if so get out of loop
          if(min(AdjustedTarget)==max(AdjustedTarget)){
            ContinueCondition<-FALSE
            shouldpredict=FALSE
            print("Error: updates are all for one group")
          }
        }
        if((class(lmefit)[1]=="character")){
          ContinueCondition<-FALSE
          print("Error: Bayesian GLMM did not converge")
        }
      }

      #return stuff

return(list(Forest=forest,EffectModel=lmefit,Iterations=iterations,PostLogLike=logLik(lmefit),returndata=data))

    }
```

Bibliography

[1] United States, Department of Health and Human Services, Centers for Disease Control and Prevention. "Most Recent National Asthma Data." Centers for Disease Control and Prevention, 13 Dec. 2022. https://www.cdc.gov/asthma/most_recent_national_asthma_data.htm.

[2] Dharmage, Shyamali C et al. “Epidemiology of Asthma in Children and Adults.” Frontiers in pediatrics vol. 7 246. 18 Jun. 2019, doi:10.3389/fped.2019.00246

[3] Lang, Jason E et al. “Well-Child Care Attendance and Risk of Asthma Exacerbations.” Pediatrics vol. 146,6 (2020): e20201023. doi:10.1542/peds.2020-1023.

[4] Johnson, Laurie H et al. “Asthma-related emergency department use: current perspectives.” Open access emergency medicine: OAEM vol. 8 47-55. 13 Jul. 2016. doi:10.2147/OAEM.S69973

[5] Al-Muhsen, Saleh et al. “Poor asthma education and medication compliance are associated with increased emergency department visits by asthmatic children.” Annals of thoracic medicine vol. 10,2 (2015): 123-31. doi:10.4103/1817-1737.150735

[6] Wu, H., Yamal, J.M., Yaseen, A., & Maroufy, V. (Eds.). (2020). Statistics and Machine Learning Methods for EHR Data: From Data Extraction to Data Analytics (1st ed.). Chapman and Hall/CRC. doi: 10.1201/9781003030003

[7] Ferrante, Giuliana, and Stefania La Grutta. “The Burden of Pediatric Asthma.” Frontiers in pediatrics vol. 6 186. 22 Jun. 2018, doi:10.3389/fped.2018.00186.

[8] Perry, Richard et al. “The Economic Burden of Pediatric Asthma in the United States: Literature Review of Current Evidence.” PharmacoEconomics vol. 37,2 (2019): 155-167. doi:10.1007/s40273-018-0726-2.

[9] Bates, D., M. Mächler, B. Bolker, and S. Walker. “Fitting Linear Mixed-Effects Models Using Lme4”. Journal of Statistical Software, vol. 67, no. 1, Oct. 2015, pp. 1-48, doi:10.18637/jss.v067.i01.

[10] Breslow, N. E., and D. G. Clayton. “Approximate Inference in Generalized Linear Mixed Models.” Journal of the American Statistical Association, vol. 88, no. 421, 1993, pp. 9–25. JSTOR, doi:10.2307/2290687.

[11] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). doi: 10.1023/A:1010933404324.

[12] Ahlem Hajjem, François Bellavance & Denis Larocque. “Mixed-effects random forest for clustered data”, Journal of Statistical Computation and Simulation, 84:6, 1313-1328 (2014), doi: 10.1080/00949655.2012.741599.

[13] Speiser, Jaime Lynn et al. “BiMM forest: A random forest method for modeling clustered and longitudinal binary outcomes.” Chemometrics and intelligent laboratory systems: an international journal sponsored by the Chemometrics Society vol. 185 (2019): 122-134. doi:10.1016/j.chemolab.