ADAPTING A ROBUST MODEL INTO HYBRID IMPLEMENTATIONS OF MACHINE LEARNING ALGORITHMS AND STATISTICAL METHODS FOR LONGITUDINAL DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İBRAHİM HAKKI ERDURAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
STATISTICS

SEPTEMBER 2021

Approval of the thesis:

ADAPTING A ROBUST MODEL INTO HYBRID APPLICATIONS OF MACHINE LEARNING ALGORITHMS AND STATISTICAL METHODS FOR LONGITUDINAL DATA

submitted by **İBRAHİM HAKKI ERDURAN** in partial fulfillment of the requirements for the degree of **Master of Science in Statistics, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Özlem İlk Dağ
Head of the Department, **Statistics** _____

Assist. Prof. Dr. Fulya Gökalp Yavuz
Supervisor, **Statistics, METU** _____

Prof. Dr. Meral Ebegil
Co-Supervisor, **Statistics, Gazi University** _____

Examining Committee Members:

Prof. Dr. Olçay Arslan
Department of Statistics, Ankara University _____

Assist. Prof. Dr. Fulya Gökalp Yavuz
Department of Statistics, METU _____

Prof. Dr. Özlem İlk Dağ
Department of Statistics, METU _____

Date: 06.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : İbrahim Hakkı Erduran

Signature :

ABSTRACT

ADAPTING A ROBUST MODEL INTO HYBRID IMPLEMENTATIONS OF MACHINE LEARNING ALGORITHMS AND STATISTICAL METHODS FOR LONGITUDINAL DATA

Erduran, İbrahim Hakkı
Master of Science, Statistics
Supervisor : Assist. Prof. Dr. Fulya Gökalp Yavuz
Co-Supervisor: Prof. Dr. Meral Ebegil

September 2021, 68 pages

Data structures in which the same characteristics are measured repeatedly at different time points are counted among the longitudinal data types. These datasets require the use of advanced modeling techniques because of the dependency structure amongst replicates. Linear mixed models (LMM) is an advanced regression method used in the analysis of such data sets. Although the LMM method provides many flexibility and advantages, the model setup is based on a number of assumptions that are challenging to provide in real data sets. Another method for analyzing the longitudinal data could be machine learning (ML) algorithms. However, many of them desire data to be independent and identically distributed (iid) which is not applicable for longitudinal data. Because of these limitations, hybrid methods including both LMM and ML have been developed to make precise estimations for longitudinal data in models with both random and fixed effects. However, these methods have model setups based on the assumption of a normal distribution of errors, which are not robust to the presence of heavy-tailed distributed data and outlier observations. This study aims to extend and robustfy hybrid methods including LMM and ML by introducing a heavy-tailed

distribution into the model setting. While LMM performs parameter estimations related to the random effect with a robust approach; the ML algorithm performs the estimation of the fixed effect parameters with the proposed model. The model is tested on two real data sets and simulation studies with several conditions and it gives promising results in real datasets and especially in simulation trials involving heavy-tailed situations and outliers. Almost all of the results based on comparison criteria such as RMSE, AIC and BIC favor the proposed method. While this study expands one of the modern topics of statistics with a robust approach and a machine learning method; it will guide researchers who practice in this field with the open source and codes provided.

Keywords: Machine Learning, Mixed Effect Models, Heavy-tailed Data, Longitudinal Data, Hybrid Models

ÖZ

SAĞLAM BİR MODELİN MAKİNA ÖĞRENMESİ ALGORİTMALARININ VE İSTATİSTİKSEL METOTLARIN HİBRİT UYGULAMALARINA BOYLAMSAL VERİLER İÇİN UYARLANMASI

Erduran, İbrahim Hakkı
Yüksek Lisans, İstatistik
Tez Yöneticisi: Dr. Öğr. Üyesi. Fulya Gökalp Yavuz
Ortak Tez Yöneticisi: Prof. Dr. Meral Ebegil

Eylül 2021, 68 sayfa

Aynı özelliklerin farklı zaman noktalarında tekrarlı olarak ölçüldüğü veri yapıları boylamsal veri türleri arasında sayılmaktadır. Bu veri kümeleri her bir tekrar arasındaki bağımlılık yapısı nedeniyle, ileri modelleme tekniklerinin kullanılmasını gerektirmektedir. Lineer karma modeller (LMM) bu tip veri setlerinin analizinde kullanılan, ileri bir regresyon yöntemidir. LMM yöntemi sağladığı bir çok esneklik ve avantajla birlikte, model kurulumu gerçek veri setlerinde sağlanması zor olan bir takım varsayımlara dayanmaktadır. Boylamsal veri analizi için başka bir seçenek ise makine öğrenmesi (ML) algoritmaları olabilmektedir. Ancak bir çok algoritma verilerin bağımsız ve aynı dağılımlı dağılmasını zorunlu kılar ve bu varsayım boylamsal veriler için uygun değildir. Bu sınırlamalar nedeniyle, rastgele etkileri ve sabit etkileri barındıran modellerde, boylamsal veriler için hassas tahminler yapan LMM ve ML'yi birlikte içeren hibrit yöntemler geliştirilmiştir. Ancak bu yöntemlerde hataların normal dağılımı varsayımına dayalı, kalın kuyruklu dağılımlar veya aykırı gözlemlerin bulunduğu durumlara karşı sağlam olmayan, model kurulumları mevcuttur. Bu çalışma, LMM ve ML'yi içeren hibrit bir modeli

kalın kuyruklu bir dağılım ile genişletmeyi ve sağlamlaştırmayı hedeflemektedir. Önerilen model ile LMM rassal etkiye ilişkin parametre tahminlerini sağlam bir yöntem ile gerçekleştirirken; ML algoritması sabit etki parametrelerinin tahminini gerçekleştirecektir. İki ayrı gerçek veri seti ve farklı durumları içeren simülasyon çalışmaları üzerinde denenen model, gerçek veri setlerinde ve özellikle kalın kuyruklu dağılımları ve aykırı durumları içeren simülasyon denemelerinde ümit verici sonuçlar vermiştir. RMSE, AIC ve BIC gibi karşılaştırma kriterlerine dayalı sonuçların neredeyse tamamı önerilen metodun lehinedir. Bu çalışma, istatistiğin modern konularından birini sağlam bir yaklaşım ve makine öğrenmesi metodu ile genişletirken; sağlanan açık kaynak ve kodlar ile bu alanda uygulama yapan araştırmacılara yol gösterici olacaktır.

Anahtar Kelimeler: Makine Öğrenmesi, Karma Etki Modelleri, Kalın Kuyruklu Veri, Boylamsal Veri, Melez Modeller

To my lovely family and my sweet dog Vita

ACKNOWLEDGMENTS

First and foremost, I wish to express my profound gratitude to my advisor Assist. Prof. Dr. Fulya Gökalp Yavuz for her endless encouragement, patience, guidance, and support throughout my thesis process. Her discipline, professionality and mentoring have been exclusively valuable. I could not overcome obstacles and achieve without her contribution.

I wish to express my sincere thanks to my co-advisor Prof. Dr. Meral Ebegil for her guidance, caring and support.

I would also like to thank my examining committee members Prof. Dr. Özlem İlk Dağ and Prof. Dr. Olçay Arslan for their valuable comments, suggestions, and criticism.

Special thanks are owed to my best friend Orçun Oltulu for giving me a lot of motivation when I lost it, and his patience during long phone calls. Also, I would like to thank my classmates Burcu Koca, Serenay Çakar and Sevilay Doğan for their support.

I would also like to thank my colleagues Ahmet Kocatürk, Canberk Arslan, and Mihraç Küpeli for making my thesis process easier.

Lastly, I wish to express my deepest gratitude to my beloved family. Special thanks to my dear mother Asiye and my dear father Ahmet for helping me become the person I am today. Also, I wish to express my thanks to my siblings for making life happier.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
BMT	Boosted Multivariate Trees
CART	Classification and Regression Trees
CNN	Convolutional Neural Networks
DSP	Data Simulation Process
GBT	Gradient Boosting Trees
GLMM	Generalized Linear Mixed Effect Model
GMERT	Generalized Mixed Effect Regression Trees
Heavy_REEMtree	Regression Expectation Maximization Tree Under Heavy Tailed Distribution
iid	Independent and Identically Distributed
LFG	Lung Function Growth
LMM	Linear Mixed Effect Models
LSTM	Long Short-Term Memory
ME-LS-SVM	Mixed Effect Least Square Support Vector Machine
MEM	Mixed Effect Models
MEml	Mixed Effect Machine Learning
MERF	Mixed Effects Random Forest

MERT	Mixed Effects Regression Tree
MKLRE	Multiple Kernel Learning with Random Effects
MLE	Maximum Likelihood Estimate
MSE	Mean Squared Error
NLME	Nonlinear Mixed Effect Models
REEMtree	Regression Expectation Maximization Tree
RESD	Random Effects Standard Deviation
RF	Random Forest
RMSE	Root Mean Squared Error
RT	Regression Tree
SMERF	Stochastic Mixed Effects Random Forests
SMERT	Stochastic Mixed Effects Regression Tree
SREEMtree	Stochastic Random Effects Expectation Maximization Trees
SSMM	Semiparametric Stochastic Mixed Models
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Longitudinal data is measured repeatedly over time for the same individuals or objects. In general, this type of data is recorded for biological, health, educational, social, agricultural purposes, and behavioral disciplines. To illustrate, collecting medical information on patients who had Covid-19 vaccine within fixed time points to explore long term effect of the vaccine could be carried as longitudinal study. That kind of data may consist of several information such as medical history, current diseases, age, gender, ethnicity of the patience and help to track the effects of vaccine on groups of people who shows similar characteristics. Although there are some advantages of using longitudinal data such as discovering extended period relationships, finding patterns related to potential trends, and having more validated data, there are some disadvantages such as requiring considerably large data, lasting long time and having high costs.

It is not trivial to deal with longitudinal data, because of its complex framework and random error design. Since the same subject is observed more than once in the data, the dependency structure is different than a rectangular data set. As a result, the inferences include both fixed and random effect estimations. Widely used statistical methods for the implementation of longitudinal data are, linear mixed effect models (LMM) [1], generalized linear mixed effect model(GLMM) [2], semiparametric stochastic mixed models (SSMM) [3], and nonlinear mixed effect models (NLME) [4]. LMM is used when tracking the continuous response variable belongs to the same individual at different time periods. GLMM is used under the same condition with LMM, but only with a binary response instead of continuous

one. SSMM is useful to conduct when parametric assumption does not meet. If the normality of error assumption is hold, but expectation function is not linear, such as growth curve data [4], NLME method would be appropriate technique.

On the other hand, some machine learning algorithms are used to analyze repeated measures or clustered data sets such as gradient boosting trees (GBT) [5], random forest (RF) [6], support vector machine (SVM) [7], multiple kernel learning with random effects (MKLRE) [8], boosted multivariate trees (BMT) [9], deep learning models such as convolutional neural networks (CNN) [10], recurrent neural networks with long short-term memory (LSTM) [11]. However, since longitudinal data is not independent and identically distributed (iid), and machine learning algorithms requires data to be idd, it is not the best way to deal with longitudinal data.

Because of this restriction of machine learning algorithms, some hybrid methods of machine learning algorithms and statistical methods are established for longitudinal data [12], [13], [14]. The main idea of these methods is combining the estimation of fixed effect part of the model with machine learning algorithms and the estimation of random effect part of the model with mixed effect models (MEM). The alternation between these two different approaches is possible with an EM-type algorithm. However, they may not always perform well when data are not distributed normally, i.e. heavy-tailed distributed, and in the presence of outliers in the data.

The aforementioned restriction motivates the subject of this study, and a new method is proposed with the aim of developing this field. Mainly, the adapted method is inspired from the Regression Expectation Maximization Tree (REEMtree) [13] which combines Regression Tree (RT) and LMM. We prefer to use REEMtree because it is used with continuous data and it allows the nodes to split based on any explanatory variables, and different repeats for a specific subject

may appear in different nodes without causing errors to be corrupted from the longitudinal data structure [13]. Also, there is no limitation about the missing values in REEMtree method.

Instead of a classical approach, the proposed method merges RT with LMM under heavy-tailed distributions assumption [15] for robustfying the hybrid method to overcome the aforementioned restrictions. It estimates the fixed part with RT and the random part with LMM under the heavy-tailed distributed random effects and errors assumptions. The tree part is implemented with the help of classification and Regression Trees (CART) algorithm [16] and preserved as in the main article [13]. LMM part is completely changed and instead of *lme* function of the nlme package implemented by Pinheiro et al. 2009 [17], 'heavyLme' function in 'heavy' package by Osorio [18] is used for the implementation of a heavy tailed distribution assumption in the model setting. The proposed method shows improvement over REEMtree when the data is heavy-tailed and in the presence of outliers in the data set.

This thesis is formed of five chapters. Chapter 1 gives the main idea and background information. In Chapter 2, the review of previous studies from the literature is supplied. Chapter 3 gives detailed formulations related to hybrid methods and proposed methods. In Chapter 4, the analysis result of two real data sets and simulation studies are presented with comparison of methods. Lastly, Chapter 5 consists of summary and conclusion of the study and gives recommendation for future studies.

CHAPTER 2

LITERATURE REVIEW

Mixed models and machine learning approaches have been used in the literature for a long time ([19], [13], [20]). However, hybrid methods are an area where relatively new studies are carried out in this field ([13], [21], [20], [12], [22], [23], [20], [24]). Many studies in this area have proven that the use of these methods together gives better results than the use of mixed models or machine learning alone ([13], [21], [20], [12], [22], [23]). Mainly, they use machine learning techniques to estimate the fixed effects and statistical methods to estimate the random effects of the model. It is shown that estimation and classification performance improvements over both ML and statistical methods by combining advantages of them. Specifically, those algorithms are mixed effect machine learning (MEml) [21], random effects expectation maximization trees (RE-EM Trees) [13]**,** mixed effect least square support vector machine (ME-LS-SVM) [20], mixed effects regression tree (MERT) [12] , mixed effects random forest (MERF) [22], generalized mixed effect regression trees (GMERT) [14], stochastic mixed effects random forests (SMERF) [23] , stochastic mixed effects regression tree (SMERT) [23], stochastic random effects expectation maximization trees (SREEMtree) [23].

In one of the recent studies, Ngufor et al. [21] propose a new method called mixed effect machine learning (MEml) framework. The aim of this method is to capture longitudinal change in glycemic control among controlled grown-ups with type 2 diabetes by using MEml. The method predicts **binary response** which is control status of the patients. The aforementioned method uses interpretable trees [25] to

extract the fixed effects of the longitudinal model and uses GLMM to capture the random effects of the model repeatedly until convergence or until it reaches to the maximum number of iterations. The pseudo codes for this methos is located at Appendix A.

Another method for analyzing longitudinal data is random effects expectation maximization trees (RE-EM Trees) [13]. This method integrates form of mixed effects models for longitudinal data with the adaptation of tree-based approach. Parametric approaches such as linear models and LMM requires restrictive assumptions such as the normality and independency of error terms. Additionally, tree-based methods more resistant to missing values than linear models, and the overfitting problem when there are too many variables may not be a problem anymore. RE-EM models, just like any other tree models, can construct interpretable and complex models, and extract intertaction between features naturally. That makes them preferale among other models. The RE-EM algorithm concentrates on regression trees (RT) [16] application of “rpart” package [26] and expectation maximization (EM) algorithm with *lme* function in “nlme” package [19].

The mixed effect least square support vector machine (ME-LS-SVM) [20] is a comprehensive version of the least square support vector machine (LS-SVM) [27]. This hybrid method is combination of mixed effect models and regularization feature of LS-SVM. ME-LS-SVM has ability to capture random effects thanks to random intercept feature and effectively work on unbalanced data which makes it superior to LS-SVM for longitudinal data since the data generally unbalanced. ME-LS-SVM is used for multi-label classification besides binary classification since the method is possible to be extended by using different kernels for each label.

Hajjem et al. [12] introduced mixed effects regression tree (MERT) as a comprehensive version of the standard regression trees methods. The simple logic behind the MERT algorithm is using the standard regression trees to estimate fixed effect and using LMM for each node of the tree for estimating random effect by using EM algorithm. By doing so, they integrated classical approach and machine learning approach to estimate continuous outcome. MERT algorithm showed significantly better performance than other approaches when random effect is nonignorable. However, some of their implementations are conducted manually and requires higher computational time. Also, the code structure is not suitable for us to adapt a robust approach in it. The pseudo codes for this methos is located at Appendix A. Hajjem et al. [22] improved their MERT algorithm [12] by adapting the random forest method instead of the random trees and they called it as mixed effects random forest (MERF). The random forest is generated via bootstrap samples. MERF algorithm has the same idea with the MERT algorithm which estimates fixed and random effects using different algorithms until it converges. The pseudo codes for this methos is located at Appendix A. In addition to those algorithms, Hajjem et al. [14] developed the generalized mixed effects regression tree (GMERT) to predict binary response or count data after few years with similar ideas. Contemplated algorithm could work on unbalanced data. GMERT models had outstanding performance over other models such as RF, GLMM and MElog. [14]

At the Table 2.1, the hybrid methods with their algorithms, response types and which techniques are combined are summarized. While Meml algorithms can predict both binary and continuous response, other algorithms, such as MERT, RE-EM, GMERT, SMERF, can predict only one response type. In general, they are combination of trees and statistical longitudinal data analysis methods.

Table 2.1. Hybrid Methods

Method	Mixed Model	Machine Learning	Algorithm	Response Types	Reference
Meml	GLMM or LMM	Trees	PQL, EM	Binary and Continuous response	[21]
RE-EM	LMM	Tree-Based	EM	Continuous response	[13]
ME-LS-SVM	Lagrange multipliers	LS-Support Vector Machines	NLME	Binary and multi-label response	[20]
MERT	LMM	Trees	EM	Continuous response	[14]
MERF	LMM	Trees	EM	Continuous response	[22]
GMERT	GLMM	Trees	EM	Binary response	[14]
SMERT	SLME	Trees	EM	Continuous response	[23]
SMERF	SLME	Trees	EM	Continuous response	[23]
SREEMtree	SLME, LME.	Trees	EM	Continuous response	[23]

Before concluding this section, it should be noted that LMM has also been expanded in many studies using different distributions. For example, Pinherio use (multivariate) t-distribution [15], Gokalp Yavuz and Arslan use Laplace distribution [28], Chou and Tsung-I use (skew) t-distribution [29] instead of (multivariate) normal distribution in their model settings. Their studies showed that

if normality of errors does not meet, alternative methods with a robust distribution enhance the performance of the model. The model improvement performances of robust distributions demonstrated by these studies motivated our study to adapt a robust distribution to a hybrid method.

All in all, among the hybrid methods, those who use LMM make inferences over the assumptions such as normality and independency. The argument in this study is that this approach may not be well enough for some data types such as heavy-tailed ones and the ones with outliers. Therefore, the proposed approaches should be extended with robust approaches. In order to prove this claim, LMM and ML method are combined and one of the frequently used methods is extended with a robust distribution and the results are compared with the classical and hybrid approaches. We aim to pioneer robustfying these methods with some extensions in the model structures. The remaining part of this section provides a brief introduction of LMM and some of the hybrid methods.

CHAPTER 3

METHODOLOGY

As a statistical modeling technique tool, LMM allows to make inference to check out the performance of the model by introducing some assumptions. LMM has the advantages of all statistical methods which is being able to characterize the relationship in the data with statistical inference. On the other hand, it requires some assumptions, but these assumptions can be flexed with some robust methods. Conversely, ML methods mainly aim to gather the best predictions, so better predictions on the test data set are the main motivation of such models. In this study, we exploit LMM and ML to get even better models and predictions by introducing a heavy-tailed distribution into one of the hybrid models. This section starts with the brief explanations of LMM and hybrid methods mixing machine learning algorithms and statistical methods for analyzing longitudinal data, and it is finalized with the proposed robust hybrid method.

3.1 Linear Mixed Effect Models for Longitudinal Data

The data structures in which the same type of measurements of subjects are collected repeatedly over time are called as longitudinal data. We mainly use the term as longitudinal data in this study, but the proposed model can be used with clustered data types or with other hierarchical structures. These data types could be measured as balanced or unbalanced. In balanced measurements, each subject is measured in equal number of times; unbalanced cases are measured in unequal numbers.The classical LMM contains both fixed and random effect parameters which take into account of between-subject and within-subject variations.

The LMM model with N subjects, n_i repeats for each subject ($j = 1, \dots, n_i$) and p features (including intercept) could be formulated as following according to Pinheiro and Bates [30].

$$y_i = X_i\beta + Z_i b_i + \epsilon_i\,, i\ =\ 1, \dots, N. \tag{3.1}$$

where y_i is $(n_i \times 1)$ vector of response, X_i is $(n_i \times (p+1))$ design matrix for the fixed effects of the *i*-th subject, β is $(p\ x\ 1)$ unknown fixed effect parameter vector, ϵ_i is $(n_i \times 1)$ vector of within-subject error for the *i*-th subject, Z_i is $(n_i \times q)$ design matrix for the random effects of the *i*-th subject, b_i is $(q\ x\ 1)$ unknown random effect vector.

The matrix forms of the *i*-th subject is as follows:

$$y_i = \begin{pmatrix} y_{i1} \\ \vdots \\ y_{ij} \\ \vdots \\ y_{in_i} \end{pmatrix},\ \epsilon_i = \begin{pmatrix} \epsilon_{i1} \\ \vdots \\ \epsilon_{ij} \\ \vdots \\ \epsilon_{in_i} \end{pmatrix},$$

$$X_i = \begin{pmatrix} X_{i1}^{(1)} & X_{i1}^{(2)} & \cdots & X_{i1}^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ X_{in_i}^{(1)} & X_{in_i}^{(2)} & \cdots & X_{in_i}^{(p)} \end{pmatrix}, \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \vdots \\ \beta_p \end{pmatrix},$$

$$Z_i = \begin{pmatrix} Z_{i1}^{(1)} & Z_{i1}^{(2)} & \cdots & Z_{i1}^{(q)} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{in_i}^{(1)} & Z_{in_i}^{(2)} & \cdots & Z_{in_i}^{(q)} \end{pmatrix}, b_i = \begin{pmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ \vdots \\ b_{iq} \end{pmatrix}$$

where b_i~N(0, D), ϵ_i~N(0, R_i), $R_i = \sigma^2 I_{n_i}$, $(i = 1, \dots, N)$. D is the covariance matrix of b_i and R_i is the covariance matrix of ϵ_i.

LMM is also defined with t-distribution and this adaptation generates robust parameter estimations for LMM. LMM with t-distribution [15] uses multivariate t-distribution for robust estimation of fixed effect parameter, covariance matrix of

random effect and covariance matrix of within subject error instead of multivariate normal distribution in LMM. Therefore, the variance of random effect and random error depend on degrees of freedom of multivariate t-distribution, so they can alter subject to subject despite the fact that they are independent of subject in LMM. However, degrees of freedom are not granted to change for each subject in general, but they change for groups of subjects due to high cost of parameter estimations. This study mainly focuses on the aforementioned type of LMM definitions. The following sub-section explains the most common algorithm used in the implementations of LMM.

3.1.1 Expectation Maximization Algorithm in Linear Mixed Effect Models

Since LMM inferences reveal parameter estimations including other unknown parameters, iterative algorithms are used in addition to analytical solutions. Due to the hierarchical structure of LMM, EM-type algorithms are frequently used in LMM inferences which ensure convergency and keep parameter estimations in the parameter space. In this section, the EM algorithm and its steps in LMM are briefly mentioned.

Basic principle of EM algorithm [31] has the following structure:

Starting Step: Determine initial values for parameter of interest $\theta^{(0)}$. Set $k = 1$, where k is indices for loop.

Expectation Step: Use observed data and present estimate of $\theta^{(k)}$ to estimate the expectation of unobserved data.

Maximization Step: Use estimated data to calculate the maximum likelihood estimate (MLE) of $\theta^{(k+1)}$.

Convergence Step: Check if it converges or not, if not keep looping until convergence.

EM algorithm for LMM is a repetitive method that reveals the MLE of unknown parameters. It keeps iterating within steps and updating estimations until convergency. The steps of EM Algorithm for LMM are defined as follows:

Step 1: Start with initial values for unknown parameters, such as beta, the variance of random effects and the variance of random error. In general, initial values are set to zero or some of them are gathered from a basic linear model if possible.

Step 2: Estimate parameter of interest by maximizing the expected value of log-likelihood.

Step 3: Update parameter of interest with newly estimated parameters.

Keep iterating until convergence.

Laird and Ware [19] state that if one could be able to observe b_i and ϵ_i, then they can find the MLE of parameter θ based on b_i and ϵ_i , where $i = 1, \dots N$. Steps of EM algorithm for LMM to obtain $\hat{\theta}_N$, which is the MLE of θ, is as follows:

Step 1: Define appropriate initial values for $\hat{\theta}$.

Step 2: Find the following parameters in the expectation steps:

$$\hat{\sigma}^2 = \frac{\sum_1^N \varepsilon_i^T \varepsilon_i}{\sum_1^N n_i} = \frac{t_1}{\sum_1^N n_i}$$

$$\widehat{D} = N^{-1} \sum_1^N b_i b_i^T = \frac{t_2}{N}$$

where $\hat{R}_i = \hat{\sigma}^2 I_{n_i}$ is the covariance matrix of ϵ_i, t_1 and t_2 are the sufficient statistics for θ and D is the covariance matrix of b_i.

Step 3: Estimate, t_1 and t_2 as:

$$\widehat{t_1} = \mathrm{E}(\sum_1^N \varepsilon_i^T \varepsilon_i \mid y_i, \widehat{\beta}\,(\hat{\theta}), \hat{\theta}))$$

$$= \sum_1^N [\, \widehat{\varepsilon_i}(\hat{\theta})^T \widehat{\varepsilon_i}(\hat{\theta}) + tr\ var\ \{\varepsilon_i \mid y_i, \widehat{\beta}\,(\hat{\theta}), \hat{\theta}\,\}\ ]$$

$\widehat{t_2} = \mathrm{E}(\sum_1^N b_i b_i^T \mid y_i, \widehat{\beta}\,(\hat{\theta}), \hat{\theta}))$

$= \sum_1^N \{\, \widehat{b_i}(\hat{\theta})^T \widehat{b_i}(\hat{\theta}) + var\,(\, y_i, \widehat{\beta}\,(\hat{\theta}), \hat{\theta}\,)\,\}$

Step 4: Keep iterating until convergence.

Since the robust version of LMM is adapted in this study, the LMM with t-distribution and its implementation steps with EM are introduced in Section 3.6.

3.2 Classification and Regression Trees (CART) - "rpart" function

Classification and regression tree (CART) is proposed by Breiman et al. [16] in 1984. CART algorithm, is a binary tree method, uses features to split internal nodes into two nodes (groups), then finds the correct accurate terminal node for each observation. Any node that is splitted into two nodes (groups) is called internal node, and any node that is not splitted into groups is called terminal node. The first root, which is at the top of the tree, is called as root node. The process of splitting trees is repeated in successive steps. In other words, the regression tree could have several internal and terminal nodes. This is called *Binary Recursive Partitioning* since the tree could be partitioned more than one by using features of the data.

Regression tree criteria for splitting node is maximizing the following criteria:

$$SS_T - (\, SS_R + SS_L\,) \qquad (3.2)$$

where $SS_T = \sum(y_i - \bar{y})^2$ is the sum of squares for that particular node. SS_R is the sum of squares for the right node and SS_L is the sum of squares for the left node. [32]

CART algorithm runs as stated below: [33]

1. Built Trees: Building trees starts with choosing the root node. Root node includes all observations in the data set. Then, the algorithm finds the best two

nodes to split the root node after checking all possible features in the data set by using aforementioned splitting criteria (Equation 3.2). It sets rule that decides which observations goes to which node. To illustrate, say if chosen splitting feature is less than some value (if the feature is continuous) or that feature is in category 1 (if feature is binary) for that observation, it goes to left root. Otherwise, it goes to right node. It splits nodes as much as possible.

2. Stop Tree: Splitting tree stops for three reasons.

I-) if only one observation is left in a node.

II-) If all observations in a node are identically distributed based on ruling feature.

III-) Maximum number of trees, settled by user, is reached.

3. Prune Tree: The algorithm prunes the tree by starting at the bottom of the tree until it reaches to simplest tree with the most effective prediction. In other words, if a terminal node does not improve the prediction, that node goes away from the tree.

4. Select Optimal Tree: The most complex tree will be the one that estimates the best response. However, it also will be the one that overfits, and will not perform well on the test data. To overcome this problem, cross validation technique is used. The data is splitted into K parts, then K-1 parts is used for training and remaining 1 for testing the model. It is done for each of K parts. Entire CART building process is done K times. At this point, you have K sequences of trees. Then, trees which has the same number of nodes is compared. Finally, the best tree with minimum number of nodes and maximum prediction performance is selected. The selected tree will be complex enough to predict well, but neither less complex to predict badly nor more complex to overfit.

When the algorithm predicts, the new observation is placed to terminal node (one of the nodes at the bottom of the tree) by using settled rules, and it is predicted as the mean of all observations in that node.

3.3 Regression Expectation Maximization Tree (RE-EM Tree)

In general, longitudinal data structure has dissimilarities between subjects which is not solely because of features or time points, but also because of the differences between subjects. To estimate continuous response in that kind of structured data, mixed effect model (MEM) is used. MEM has two components that the random effects which take into account the dissimilarities between subjects and the fixed effects which take into account the population-level effect. Combining random effect and fixed effect in a model result in MEM. [13]

Sela and Simonoff [13] present an alternative method for estimating continuous response variable in longitudinal data which is called regression expectation maximization tree (REEMtree) method. In their method, they combine the advantages of tree-basis methodologies and MEM. They use classification and regression tree (CART) [16] algorithm since it does not require parametric assumptions such as normality, and it can handle more missing data than LMM could do. They use LMM to benefit from its random effect structure, and EM [19] for the parameter estimations in LMM. It is better to use REEMtree for the prediction out of sample response in high-dimensional longitudinal data (measurement of a subject/group repeats more than 400 times) compared to other longitudinal data analysis methods such as LMM. REEMtree shows predictive improvement in comparison to standard regression trees and provides with flexibility of estimating target variables. According to simulation results, when tree form of relationship between independent variables and response variable is built, REEMtree surpasses the tree methods that cannot account for random effects.

REEMtree can be applied in R programing language by using ‘REEMtree’ [34] package. The REEMtree package uses ‘rpart’ function under the ‘rpart’ package [26] to implement RT part, and ‘lme’ function under ‘nlme’ [35] package to implement LMM part while modeling with continuous response variable.

Description of the notation for the algorithm:

$$y_{it} = X_{it}\beta + Z_{it}b_i + \varepsilon_{it}, \quad (3.3)$$

where $i = 1, \dots, N$ (subject number), $t = 1, \dots, T$ (time), y_{it} is continuous response, X_{it} is the known matrix for features, Z_{it} is the known design matrix for random effect, ε_{it} is the error term. β is the fixed effect parameter(s), and b_i is the random effect parameters for the model.

One could fit a regression tree to estimate β by setting y_{it}-$Z_{it}\hat{b}_i$ as response value assuming random effects, b_i, are known. Also, one could estimate random effects, b_i, by using LMM if fixed effects, β are known. For this reason, the algorithm fits regression tree assuming random effects are accurate and, fits LMM assuming regression trees are accurate in EM algorithm. The algorithm runs as stated below.

1. Set estimated random effects, $\hat{b}_i$ as zero.

2. Run the following sub-steps until estimated random effects,$\hat{b}_i$, converge. (restricted likelihood function is less than some tolerance value or depending on change in the likelihood)

a) Fit regression tree to estimate β, by using y_{it}-$Z_{it}\hat{b}_i$ as response variable. Set new indicator variable, $I(X_{it} \in a_p)$, a_p is groups of nodes, where p is range for nodes, by using regression tree nodes. In other words, specify the terminal nodes for each data points to create new indicator variable. The algorithm finds distinct tree for each observation. That means different data points for the same subject could be placed into different trees.

b) Implement LMM with $y_{it} = \boldsymbol{I}(\boldsymbol{X_{it}} \in \boldsymbol{a_p})\boldsymbol{\mu_p} + Z_{it}b_i + \epsilon_{it}$. Get estimated random effects, $\hat{b}_i$, from the fitted model.

3. Use estimated $\boldsymbol{\hat{\mu}_p}$, from the LMM in 2-b instead of predicted response in each terminal node of the fitted tree.

3.4 Mixed Effect Regression Trees (MERT)

Mixed Effect Regression Trees (MERT) method is proposed by Hajjam et al. [12] in 2010 to improve the prediction performance for clustered and longitudinal data. It combines mixed effect models with regression trees (RT). RT is used to estimate fixed effects, and node-invariant linear form is used for estimation random effects within structure of EM algorithm. The proposed model for MERT is defined as:

$$y_i = f(X_i) + Z_i b_i + \varepsilon_i, i = 1, \ldots, N \quad (3.4)$$

where $b_i \sim N(0, D)$, $\varepsilon_i \sim N(0, R_i), D$ is the covariance matrix of b_i, R_i is the covariance matrix of ε_i, $f(X_i)$ is the unknown RT model that will be estimated.

Index for iteration numbers is $r = 1, \ldots\ldots, n$, and steps for MERT algorithm is as follows:

Step 0. Setting r = 1, $\hat{\sigma}^2_{(0)} = 1$, $\widehat{D}_{(0)} = I_q$.

Step 1. Letting r = r + 1. Estimating $y^*_{i(r)}$, $\widehat{f}(X_i)_{(r)}$, $\widehat{b}_{i_{(r)}}$ by using

i. $y^*_{i(r)} = y_i - Z_i \widehat{b}_{i_{(r-1)}}, i = 1, \ldots\ldots, N.$ (3.5)

ii. Estimate $\widehat{f}(X_i)_{(r)}$ by using RT and use $y^*_{i(r)}$ as response, X_i as features.

iii. $\widehat{b}_{i_{(r)}} = \widehat{D}_{(r-1)} Z_i^T \widehat{V}^{-1}_{i_{(r-1)}} (y_i - \widehat{f}(X_i)_{(r)}), i = 1, \ldots\ldots N.$ (3.6)

where $V_{i_{(r-1)}} = Z_i \widehat{D}_{(r-1)} Z_i^T + \hat{\sigma}^2_{(r-1)} I_{n_i}, i = 1, \ldots\ldots N.$

Step 2. Update $\hat{\sigma}^2_{(r)}$ and $\widehat{D}_{(r)}$ with:

$$\hat{\sigma}^2_{(r)} = \frac{1}{N} \sum_{i=1}^{N} (\widehat{\epsilon}^T_{i_{(r)}} \widehat{\epsilon}_{i_{(r)}} + \hat{\sigma}^2_{(r-1)} [n_i - \hat{\sigma}^2_{(r-1)} \text{trace}(\widehat{V}_{i_{(r-1)}})]),$$

$$\widehat{D}_{(r)} = \frac{1}{N} \sum_{i=1}^{n} (\widehat{b}_{i_{(r)}} \widehat{b}^T_{i_{(r)}} \widehat{\epsilon}_{i_{(r)}} + [\widehat{D}_{(r-1)} - \widehat{D}_{(r-1)} Z_i^T \widehat{V}^{-1}_{i_{(r-1)}} Z_i \widehat{D}_{(r-1)}]). \quad (3.7)$$

Step 3. Repeating Step 1 and Step 2 until it converges.

where $\widehat{\epsilon}_{i(r)} = y_i - \widehat{f}(X_i)_{(r)} - Z_i \widehat{b}_{i(r)}$,

V_i is the covariance matrix of vector of observations in y_i in subject i.

$Cov(y_i) = Z_i D Z_i^T + R_i$,

$V = Cov(y) = diag(V_1, \ldots . V_n)$.

3.5 Mixed Effect Random Forest (MERF)

Mixed Effect Random Forest (MERF) is a progressive version of MERT since this algorithm simply changes the RT in MERT algorithm with RF. In other words, every time it iterates, building forest is redone. Notation in this section is comparable with MERT's notation.

Steps for MERF algorithm are as follows:

Step 0. Setting $r = 0$, $\widehat{\sigma^2}_{(0)} = 1$, $\widehat{D}_{(0)} = I_q$.

Step 1. Letting $r = r + 1$. Estimating $y^*_{i(r)}$, $\widehat{f}(X_i)_{(r)}$, $\widehat{b}_{i(r)}$ by using

i. $y^*_{i(r)} = y_i - Z_i \widehat{b}_{i(r-1)}$, $i = 1, \ldots, N$. (3.8)

ii. Built RF and use $y^*_{i(r)}$ as response, X_i as features, use bootstrap sampling method to build forests.

iii. Estimate $\widehat{f}(X_i)_{(r)}$ by using trees in RF.

iv. $\widehat{b}_{i(r)} = \widehat{D}_{(r-1)} Z_i^T \widehat{V}^{-1}_{i(r-1)} (y_i - \widehat{f}(X_i)_{(r)})$, $i = 1, \ldots . N$. (3.9)

where $\widehat{V}_{i(r-1)} = Z_i \widehat{D}_{(r-1)} Z_i^T + \sigma^2_{(r-1)} I_q$.

Step 2. Calculating $\widehat{\sigma^2}_{(r)}$ and $\widehat{D}_{(r)}$ by using:

$$\widehat{\sigma^2}_{(r)}=\frac{1}{N}\sum_{i=1}^{N}\{\ \widehat{\epsilon}^T{}_{i(r)}\ \widehat{\epsilon}_{i(r)}\ +\ \hat{\sigma}^2_{(r-1)}\Big[\ n_i\ -\ \hat{\sigma}^2_{(r-1)}\text{trace}\Big(\ \widehat{V}_{i(r-1)}\Big)\ \Big]\},\tag{3.10}$$

$$\widehat{D}_{(r)}=\frac{1}{N}\sum_{i=1}^{N}\{\ \widehat{b}_{i(r)}\ \widehat{b}^T{}_{i(r)}\ \widehat{\epsilon}_{i(r)}\ +\ [\widehat{D}_{(r-1)}\ -\ \widehat{D}_{(r-1)}Z_i^T\ Z_i^T\ \widehat{V}^{-1}{}_{i(r-1)}\widehat{D}_{(r-1)}\ ]\ \ \}.\tag{3.11}$$

Step 3. Iterate Step 1 and Step 2 until it converges.

One key point of the MERT algorithm is that it uses bootstrap sampling after removing random effects at Step 1-ii, since observations are not independent from each other. To clarify, when observations are independent, bootstrap sampling method would be appropriate, but MERT algorithms is used for observations that are dependent to each other. Hajjem et al. [22] assumed that removing random effects discharges intra-cluster correlation. They compared MERT with four other algorithms, MERF, RF, RT, LME, LM and found out that MERT outperforms to all other algorithms, especially when random effects are present.

3.6 Proposed Method: Heavy-Tailed Regression Expectation Maximization Trees (Heavy_REEMtree)

One can use classical statistical approaches or machine learning algorithms for handling longitudinal data. However, in most cases, it is difficult to satisfy normality assumption intendent for real data sets. Additionally, machine learning techniques require data to be independently and identically distributed, but the nature of longitudinal data reserves dependency structures within subjects. Furthermore, longitudinal data sets often have missing values or outliers, and they are often high dimensional. For some data sets, using solely LMM or ML algorithms may not be sufficient to overcome these limitations. Proposed method adapts RT to use its advantage of tolerating missing values and being easy to explain, and LMM **under heavy-tailed distribution** to use its advantage of including the random effect structure for non-normal data sets. The adapted model

is mainly motivated by REEMtree [13] and robustified LMM [15]. In the proposed method, we adapt REEMtree, since it is able to estimate continuous data, uses explanatory variables to split nodes, and different measures for same subject can be placed into different nodes despite the longitudinal data structure and it can extract interpretable trees. Additionally, its complexity is well enough to adapt a new method in R. The adapted model uses LMM under heavy-tailed distribution introduced by Pinheiro et al. [15] instead of classical LMM used in REEMtree. Algorithm works with the same logic as REEMtree. For the ease of understanding, algorithms will be provided by bolding the adjustments.

1. Set estimated random effects, $\hat{b}_i$ to zero.

2. Run the following sub-steps until the differences between restricted likelihood functions of two adjacent steps is less than some tolerance value:

a) Fit regression tree to estimate β, by using y_{it}-$Z_{it}\hat{b}_i$ as response variable. Set new indicator variable, $I(X_{it} \in a_p)$, a_p is groups of nodes, where p is range for nodes, by using regression tree nodes. In other words, specify the terminal nodes for each data points to create new indicator variable. The algorithm finds distinct tree for each observation. That means different data points for the same subject could be placed into different trees.

b) Implement LMM **under heavy-tailed distribution** with $y_{it} = \boldsymbol{I}(\boldsymbol{X_{it}} \in \boldsymbol{a_p})\boldsymbol{\mu_p} + Z_{it}b_i + \epsilon_{it}$. Get estimated random effects, $\hat{b}_i$, from the fitted model.

3. Use estimated $\boldsymbol{\hat{\mu}_p}$, from the LMM **under heavy-tailed distribution** in 2-b instead of predicted response in each terminal node of the fitted tree.

The model is extended from using 'REEMtree' package [34]. 'REEMtree' estimates the response variable with the help of 'rpart' function [26] and 'lme' function [35]. The adapted robust model is modified from source codes of 'REEMtree' by substituting 'lme' with 'heavyLme' function under 'heavy' package [18] in R programming and making necessary adjustments. 'heavyLme'

function is the computational version of efficient algorithms for robust estimation in LMM using the multivariate t-distribution proposed by Pinheiro et al. [15]. The function is implemented by Osorio [18].

Multivariate-t LMM by Pinheiro et al. [15] is defined as follows. Please note that $y_i,\ X_i, \beta,\ Z_i,\ b_i\ and\ \epsilon_i$ represent the same model components as LMM defined at (3.1).

$$y_i = X_i\beta + Z_i b_i + \epsilon_i,\ i\ =\ 1, \dots, N, \tag{3.12}$$

$$b_i \sim t_q(0, \Psi, \nu_i),$$

$$\epsilon_i \sim t_{n_i}(0, \Lambda_i, \nu_i),$$

where Ψ is the covariance matrix of b_i, Λ_i is the covariance matrix of ε_i. ν_i stands for multivariate-t distribution degrees of freedom for i_{th} observation. b_i and ϵ_i are uncorrelated.

$$var(b_i) = \frac{\nu_i}{\nu_i - 2}\ \Psi\ ,\ var(\epsilon_i) = \frac{\nu_i}{\nu_i - 2}\ \Lambda_i, (\nu_i > 2),$$

The joint distribution of y_i and b_i is written as:

$$\begin{bmatrix} y_i \\ b_i \end{bmatrix} \overset{ind}{\sim} t_{ni+q}\left(\begin{bmatrix} X_i\beta \\ 0 \end{bmatrix}, \begin{bmatrix} Z_i\Psi Z_i^T + \Lambda_i & Z_i\Psi \\ \Psi Z_i^T & \Psi \end{bmatrix}, \nu_i \right)\ , \text{i=1,…..N,} \tag{3.13}$$

$$y_i \sim t_{ni}\left(X_i\beta, Z_i\Psi Z_i^T + \Lambda_i, \nu_i\right), \tag{3.14}$$

where $b_i \sim$ N (0, Ψ), $\varepsilon_i \sim$ N (0, Λ_i).

3.6.1 Expectation Maximization Algorithm in Linear Mixed Effect Models with multivariate t-distribution

EM for LMM with multivariate t-distribution [15] is easy to implement due to its hierarchical structure. The marginal distribution of $[y_i', b_i']'$ can be expressed in the following hierarchical structure:

$$\begin{bmatrix} y_i \\ b_i \end{bmatrix} | \tau_i \overset{ind}{\sim} N_{ni+q} \left(\begin{bmatrix} X_i\beta \\ 0 \end{bmatrix}, \frac{1}{\tau_i} \begin{bmatrix} Z_i \Psi Z_i^T + \Lambda_i & Z_i\Psi \\ \Psi Z_i^T & \Psi \end{bmatrix}, \nu_i \right),$$

$$\tau_i \overset{ind}{\sim} Gamma\left(\frac{\nu_i}{2}, \frac{\nu_i}{2}\right), \; i = 1, \ldots N \tag{3.15}$$

or

$$y_i | b_i, \tau_i \overset{ind}{\sim} N(X_i\beta + Z_i b_i, \; \frac{1}{\tau_i}\Lambda_i),$$

$$b_i | \tau_i \overset{ind}{\sim} N\left(0, \frac{1}{\tau_i}\Psi\right),$$

$$\tau_i \overset{ind}{\sim} Gamma\left(\frac{\nu_i}{2}, \frac{\nu_i}{2}\right), \; i = 1, \ldots N. \tag{3.16}$$

ML estimation with ECME [36] with unknown degrees of freedom is found by letting:

$$\hat{\tau}_i = E[\tau_i | \theta = \hat{\theta}, y], \; \hat{b}_i = E[b_i | \theta = \hat{\theta}, y], \; \hat{\Omega}_i = \hat{\tau}_i cov[b_i | \theta = \hat{\theta}, y],$$

$$\hat{\Omega}_i = \hat{\Psi} - \hat{\Psi} Z_i^T (Z_i \hat{\Psi} Z_i^T + \hat{\sigma}^2_{h(i)} R_i)^{-1} Z_i \hat{\Psi} = (\hat{\Psi}^{-1} + \frac{1}{\hat{\sigma}^2_{h(i)}} Z_i^T R_i^{-1} Z_i)^{-1} \tag{3.17}$$

See [15] for details.

Finally, ECM algorithm for LMM with multivariate t-distribution is defined as follows:

E- Step: Calculate $\hat{b}_i, \; \hat{\tau}_i, \; \hat{\Omega}_i$ for $i = 1, \ldots N$ using Equations 3.16 and 3.17.

Constrained Maximization Step 1: Set $\hat{\sigma}^2_{h(i)} = \sigma^2_{h(i)}$, i=1,…,N and update $\hat{\beta}$, where

$$\hat{\beta} = (\sum_1^N \frac{\hat{\tau}_i}{\hat{\sigma}^2_{h(i)}} X_i^T R_i^{-1} X_i)^{-1} \sum_1^N \frac{\hat{\tau}_i}{\hat{\sigma}^2_{h(i)}} X_i^T R_i^{-1} (y_i - Z_i \hat{b}_i) \tag{3.18}$$

Constrained Maximization Step 2: Set $\beta = \hat{\beta}$ and update $\hat{\sigma}^2_{h(i)}$, where i=1,….N

$$\hat{\sigma}_j^2 = \sum_{i:h(i)=j}[\ \hat{\tau}_i(y_i - X\hat{\beta} - Z_i\hat{b}_i)^T R_i^{-1}(y_i - X\hat{\beta} - Z_i\hat{b}_i) + trace(\hat{\Omega}_i Z_i^T R_i^{-1} Z_i) \ / \ \sum_{i:g(i)=j} n_i \quad (3.19)$$

Constrained Maximization Step 3: Update $\hat{\Psi}$ using:

$$\hat{\Psi} = \frac{1}{N}\sum_1^N(\hat{\tau}_i \hat{b}_i \hat{b}_i^T + \hat{\Omega}_i\ ) \quad (3.20)$$

Constrained Maximization Step 4: Update $\hat{\nu}$ using:

$$\hat{\nu}_j = \arg\max_{\nu} \sum_{i:g(i)=j}(\frac{\nu}{2}\{\ln\left(\frac{\nu}{2}\right) + E[\ln(\tau_i)|y, \hat{\theta}] - \hat{\tau}_i\} - \ln\left[\Gamma\left(\frac{\nu}{2}\right)\right] \quad (3.21)$$

Keep iterating until convergence.

According to the study of Pinheiro et al. [15], the LMM model with multivariate t-distribution gives better parameter estimations than the classical LMM according to simulations and data analysis examples while having outliers in the data set even with small number of them. The results of Pinheiro et al. [15], motivate the use of a heavy-tailed distribution model setup in this study, as well.

CHAPTER 4

DATA ANALYSIS AND SIMULATION STUDIES

Data analysis part covers analysis of two real data sets which are lung function growth and alcohol usage of youth people, and simulation study extended from the work of Hajjem et al. [37]. In each part, after getting familiarized with the data set and providing with the summary statistics, lattice graphs; the related comparison measurements are calculated for LMM, REEMTree and the proposed method (Heavy_REEMTree).

4.1 Introduction to Datasets

This section contains two separate real datasets to which hybrid algorithms are applied. The first data set related to the study on the lung function growth, called "*fev1"*, is from the R package called "*ALA*" [38].

The second data set is about alcohol consumption of youth people [39]. This data set is referred as Alcohol Usage of Youth People.

4.1.1 Lung Function Growth

This study is carried out on 13,379 children who born after 1966. The study is designed to find out factors that affect the lung function growth (LFG). The children located on different states of the USA, and they are the participant of six different communities. Most of the children was participated in the study between age of six and seven. The researchers examined the children and measured their pulmonary function every year until they graduated from high school or drop out of the study. In addition to this, their age and height are recorded. The data set in R is a subsample of the main study. The subsample includes 1,994 observations, 300

female students who lives in Topeka, Kansas with at least one measurement and maximum twelve measurements. The *age0* and *height0* stand for the first data point of the measurement of age and height variables. The response, *logFEV1* stands for the logarithm of lung function. Back transformation is applied to find the original measurement values and named as *fev*, so it is used as response variable. For testing the performance of the proposed robust method, a slight perturbation is applied to the response variable to make sure it has some outliers. "10" is added to the each response values of the first and the last observation of the dataset.

4.1.1.1 Explanatory Analysis for Lung Function Growth

To get insight about the data, Table 4.1 is examined.

Table 4.1: Summary Statistics for Lung Function Growth

	age	**height**	**age0**	**height0**	**fev**
Minimum	6.434	1.110	6.434	1.110	0.500
Median	12.595	1.540	7.781	1.260	2.385
Mean	12.566	1.497	8.030	1.276	2.391
Maximum	18.691	1.790	14.067	1.720	13.220

Minimum/maximum values, mean and median are seen at Table 4.1. The maximum value of the response variable *fev* seems outlier, but it is result of the perturbation. It is intentionally changed to an outlier to be able to observe its effect on the methods being compared.

There are 300 subjects in the datasets. For the ease of tracking, lattice graphics (Figure 4.1) for randomly selected 54 subjects is added in this part. Lattice graphics of the remaining observations can be found at https://github.com/hakkierduran/lattice. The lattice graphics represent the relationship between age and fev(response variable) and how it changes for each subject. The horizontal line stands for age whereas the vertical line stands for LFG,

and each box represents a specific subject with their subject numbers above. It can be observed from the first look at the graph that the number of measurements for each observation are not the same.

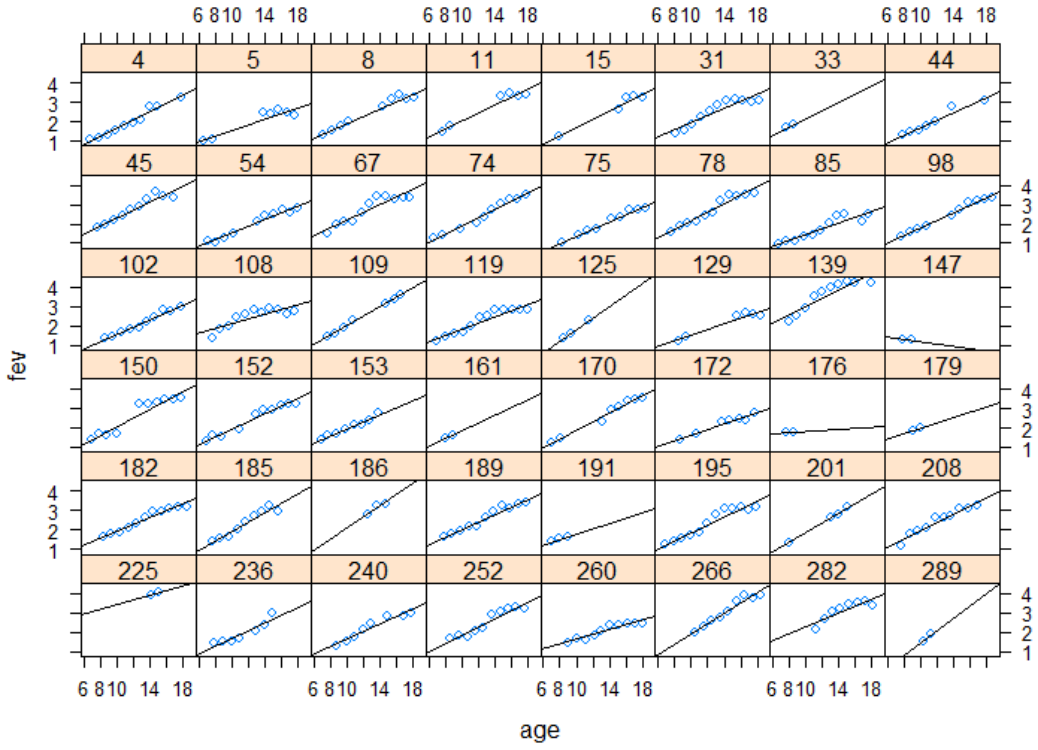

Figure 4.1: Lattice Graphics for Randomly Selected 54 Subjects from the Data

Figure 4.1 indicates that there is an increasing trend over time. In other words, lung function increases as age increases. However, the starting point and the slope of the changes are different in each observation. This encourages the use of a model that can show the variability in each observation, rather than a generalized model.

4.1.1.2 Analysis for Lung Function Growth

In this section, three different models taking into account the differences between subjects are used to model the response variable, LFG. “Age” and “height” are used as independent/explanatory variables, and “id” is used as grouping variables

for all tree models. The first model is LMM proposed by Verbeke and Lesaffre [1], and "*lme*" function under "*nlme*" package in R is used to build the model. The second model is REEMtree [13] model proposed by Sela and Simonoff. The function "*REEMtree*" under REEMtree package in R is used to construct the model. The third model is the Regression Expectation Maximization Tree under heavy-tailed distribution (Heavy_REEMtree), which is proposed in this study. Estimated variances of errors, log-likelihood values, root mean squared error (RMSE), mean squared error (MSE), random effects standard deviation (RESD), Akaike information criterion (AIC), Bayesian information criterion (BIC) values for each model are located at Table 4.2.

Table 4.2: Comprehension of Models Used to Estimate LFG.

	LMM	**REEMtree**	**Heavy_REEMtree**
Estimated variance of errors	0.132632	0.207443	**0.004363**
Log likelihood	-1003.671	-1419.876	**-670.830**
RMSE	**0.3455504**	0.4342631	0.4028607
MSE	**0.1194051**	0.1885844	0.1622967
RESD	0.2338404	0.0641993	**0.0368498**
AIC	2017.343	2847.752	**1347.672**
BIC	2045.325	2870.14	**1364.454**

According to Table 4.2, proposed method outperforms the classical approach and REEMtree. Heavy_REEMtree gives better performance measurements in terms of both estimated variance of errors and log-likelihood when compared to other models. Heavy_REEMtree model shows improvement over REEMtree model in terms of Root Mean Squared Error (RMSE) and Mean Squared Error (MSE). Additionally, Heavy_REEMtree model can be chosen as the best model among others, since it has the smallest values of AIC, BIC and RESD. However, Heavy_REEMtree model could not perform well in terms of RMSE and MSE when compared to LMM.

The decision trees for two hybrid models are located at Figure 4.2 and Figure 4.4.

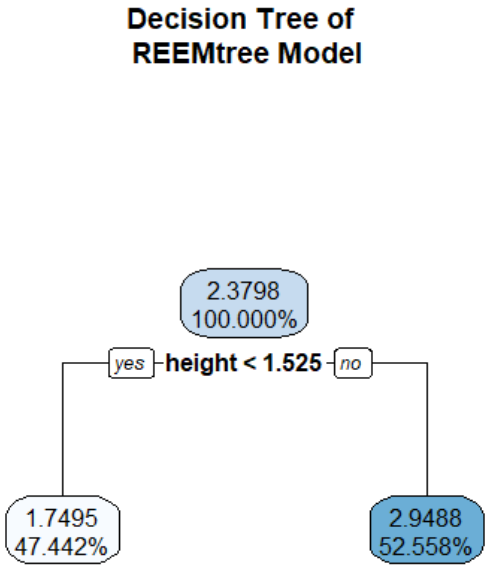

Figure 4.2: Decision Tree of REEMtree Model

Figure 4.2 depicts interpretable decision tree used to construct REEMtree model. According to Figure 4.2, lung function of observations whose height is less than 1.525 is predicted as 1.749 and lung function of observations whose height is greater than 1.525 is predicted as 2.948.

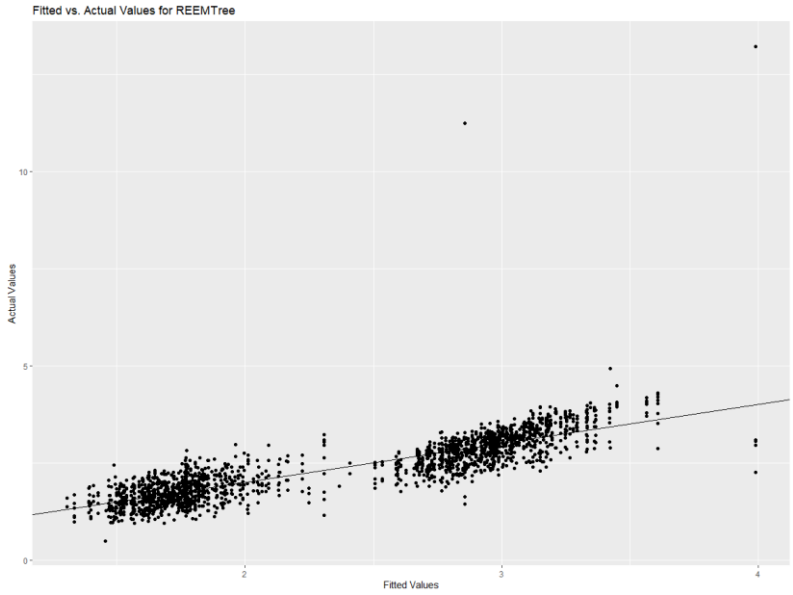

Figure 4.3: Fitted vs. Actual Values for REEMTree Model of LFG

According to Figure 4.3, fitted values for REEMTree model are spread around the diagonal line with noticeable gaps. Even though the model does not fit perfectly, it seems to be a reasonable model for LFG.

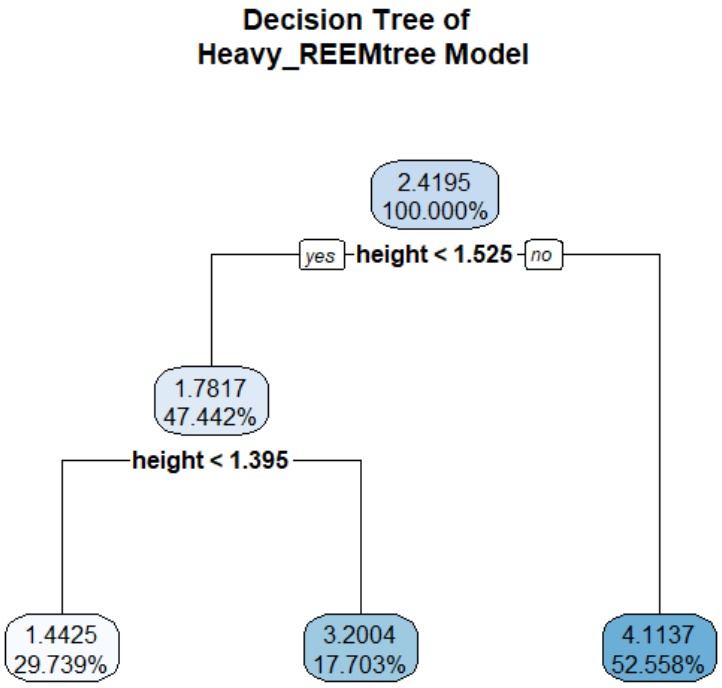

Figure 4.4: Decision Tree of Heavy_REEMtree Model

Figure 4.4 represents interpretable decision tree used to build Heavy_REEMtree. According to Figure 4.4, lung function of observations whose height is less than 1.396 is predicted as 1.443, lung function of observations whose height is between 1.395 and 1.525 is predicted as 3.2004, lung function of observations whose height is greater than 1.525 is predicted as 4.113.

When it is compared to the tree of REEMtree model, the proposed model gives more detailed and accurate estimations.

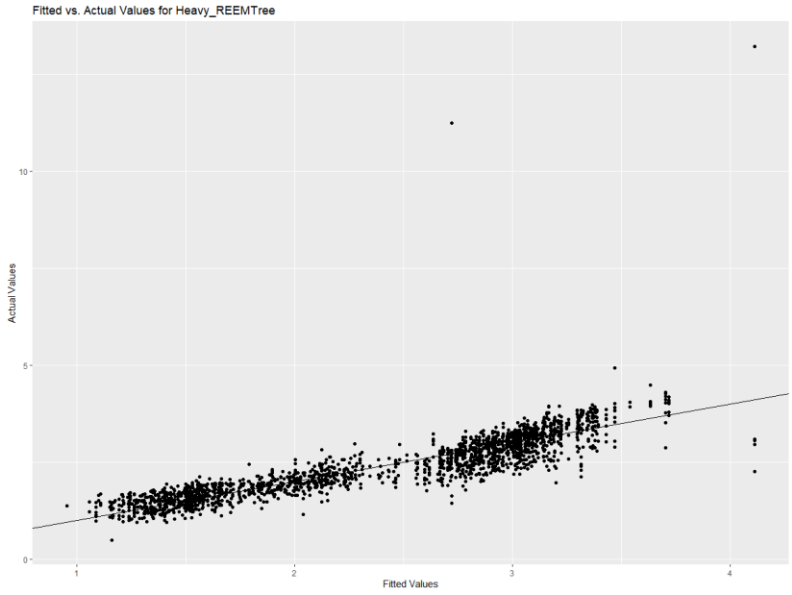

Figure 4.5: Fitted vs. Actual Values for Heavy_REEMTree Model of LFG

According to Figure 4.5, fitted values for Heavy_REEMTree model are spread around the diagonal line with smaller gaps than REEMTree model. The proposed robust model outperforms REEMTree model since fitted values of Heavy_REEMTree model are closer to actual values when compared to REEMTree.

4.1.2 Alcohol Usage of Youth People

The aim of alcohol usage of youth people study is to investigate which effects are important to explain the amount of alcohol usage among youth people. This study is conducted with 82 young people in 1997. They started to measure when they were 14 years old and repeated measurements three years in a row. The data set consist of 246 observations and nine variables which are *id, age, coa* (child of an alcoholic parent, cases as 1 = yes, 0 = no), *male, age_14, alcuse* (response variable, alcohol usage), peer (alcohol usage of youth's associate), peer (centered peer), ccoa (centered coa). At the beginning of the study, youths are asked about alcohol usage

of their friend and wanted to give score between ‘0’ and ‘6’. ‘0’ means none of their friends uses alcohol, and ‘6’ means all of their friends use alcohol. For testing the performance of the proposed robust method, a slight perturbation is applied to the response variable to make sure it has some outliers. “10” is added to the response value of the first observation of the dataset.

4.1.2.1 Explanatory Analysis of Alcohol Usage of Youth People

Table 4.3 and Table 4.4 give descriptive statistics and cross-tables, respectively.

Table 4.3 : Summary Statistics Alcohol Usage of Youth People

	age	**peer**	**alcuse**
Minimum	14	0	0
Median	15	0.8944	1.0000
Mean	15	1.0176	0.9626
Maximum	16	2.5298	11.7321

In consonance with Table 4.3, maximum value of *alcuse* might be seen as outlier, but it is result of the perturbation.

Table 4.4: Cross Table for Alcohol Usage of Youth People

		Gender	
		Female	Male
Child of an alcoholic parent	Yes	51	60
	No	69	66

As reported by Table 4.4, 111 of youth people stated that their parents use alcohol whereas 135 of them stated that their parents don't use alcohol. Additionally, 40 of 82 youth is female while 42 of them is male.

Lattice graphics (Figure 4.7) are drawn for randomly selected 20 subjects, lattice graphics for the remaining subjects are added to https://github.com/hakkierduran/lattice. The lattice graphics show how relationship between alcohol usage and age vary for each subject. The x-axis represents age while y-axis represents the alcohol usage (response variable), and each box represents a different subject.

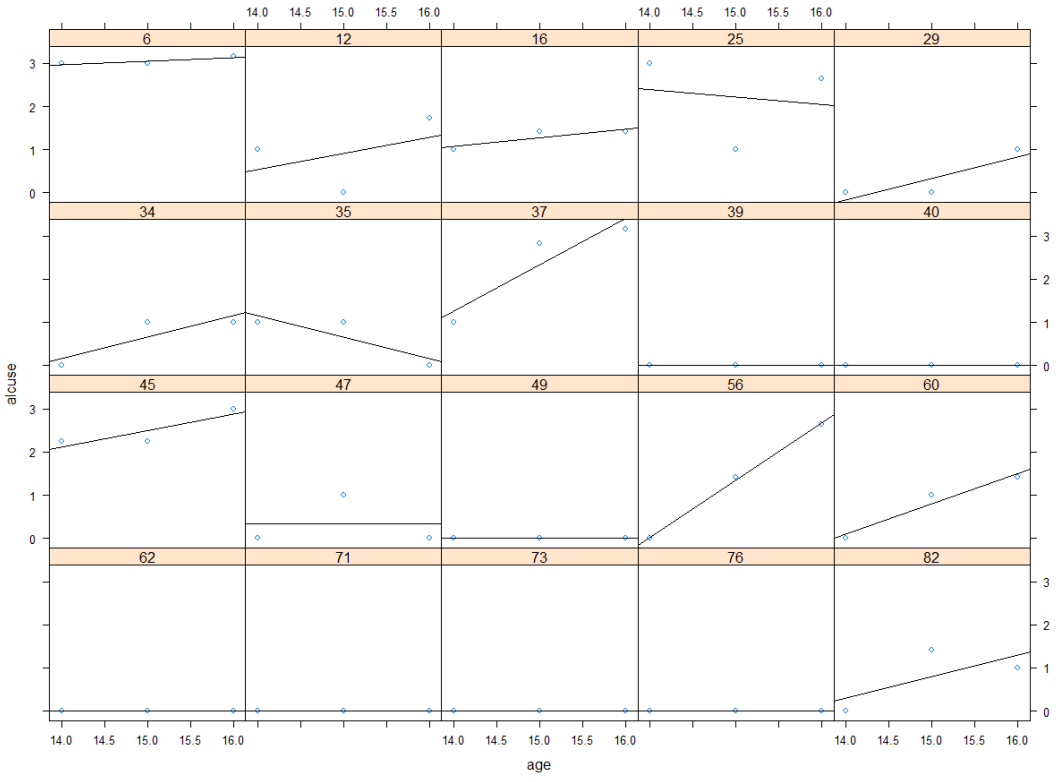

Figure 4.6: Lattice Graphics for Randomly Selected 20 subjects.

According to Figure 4.6, some of the youth people never use alcohol during the study time period, some of them had increased alcohol usage over the years, and some of them has reduced. Therefore, all subjects have individual trend slopes.

4.1.2.2 Analysis for Alcohol Usage of Youth People

In this section, response variable *alcuse* is estimated by using tree different models. Independent variables are *id* (as grouping variable), *age, coa, male* and *peer*. R programing language is used for running algorithms. To compare the proposed robust approach with classical methods, "LMM", "REEMtree" [13] and "Heavy_REEMtree" models are used. The results of the models can be seen at Table 4.5 below.

Table 4.5: Comprehension of Models Used to Estimate Alcohol Usage

	LMM	**REEMtree**	**Heavy_REEMtree**
Estimated variance of errors	0.90804	0.94646	**0.10558**
Log likelihood	-373.8367	-389.443	**-344.458**
RMSE	0.8541	0.8553	**0.8168**
MSE	0.7294	0.7316	**0.6673**
RESD	0.6160226	0.66177	**0.47617**
AIC	761.6734	784.8866	**695.0164**
BIC	786.0669	795.3903	**705.4332**

Table 4.5 indicates that proposed method, Heavy_REEMtree shows better performance than other two models in terms of all criterions since it has the biggest log-likelihood, the smallest estimated variance of errors, RMSE, MSE, AIC, BIC and RESD.

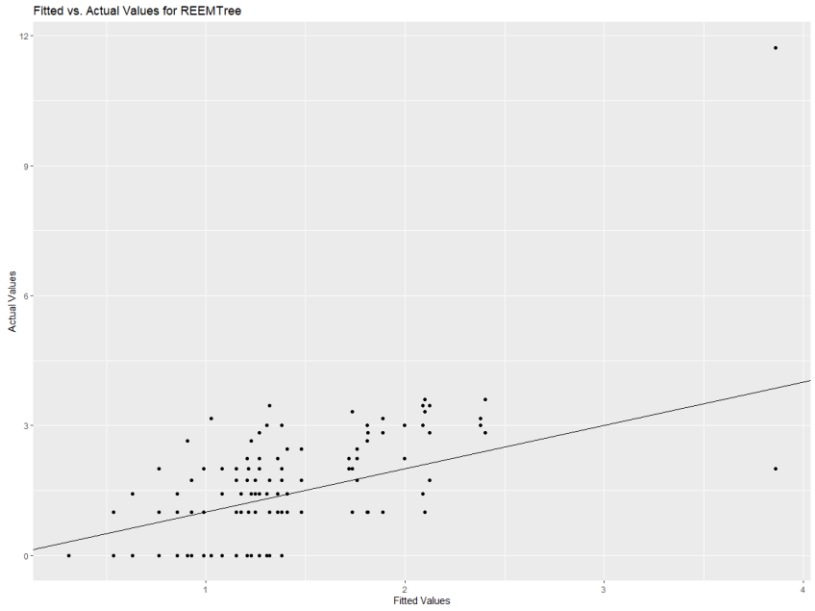

Figure 4.7: Fitted vs. Actual Values for REEMTree Model of Alcohol Usage of Youth People

According to Figure 4.7, fitted values for REEMTree model does not seem to fit well for Alcohol Usage of Youth People. Most probably, it is because this data set does not include high number of observations and repetition, and also increasing and decreasing trends together. Also, there are too many zero records of response variable.

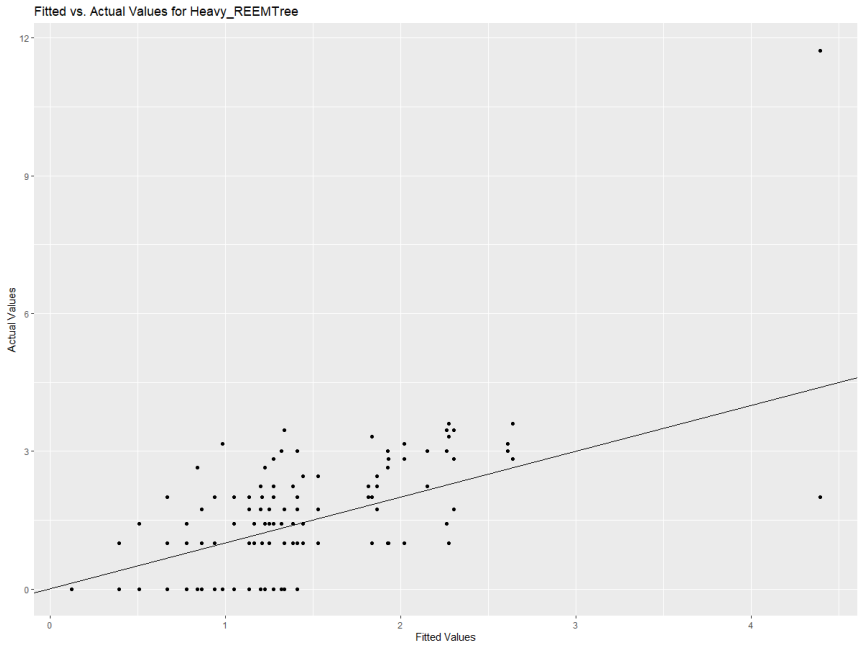

Figure 4.8: Fitted vs. Actual Values for Heavy_REEMTree Model of Alcohol Usage of Youth People

According to Figure 4.8, the Heavy_REEMTree model for Alcohol Usage of Youth People does not fit as well. However, we observe slight improvement in terms of the fitted values of the proposed method.

4.1.3 Simulation Study

The data simulation process (DSP) used in this section is extended from the simulation study of Hajjem et al. [37] for the ease of comparisons. In this study, 14 DSP is used to create the set of trees based on the following rules;

$$y_{ij} = \mu_1 + z_i^T b_i + \epsilon_{ij} \; if \; X_{1_{ij}} \leq 5 \; and \; X_{2_{ij}} \leq 5$$

$$y_{ij} = \mu_2 + z_i^T b_i + \epsilon_{ij} \; if \; X_{1_{ij}} \leq 5 \; and \; X_{2_{ij}} > 5$$

$$y_{ij} = \mu_3 + z_i^T b_i + \epsilon_{ij} \; if \; X_{1_{ij}} > 5 \; and \; X_{3_{ij}} \leq 5$$

$$y_{ij} = \mu_4 + z_i^T b_i + \epsilon_{ij} \; if \; X_{1_{ij}} > 5 \; and \; X_{3_{ij}} > 5$$

Therefore, there is a data set with 4 leaves and 4 different means. The number of observations is 100, (where $i = 1, \ldots 100$), and the number of repeated measurements is 50, (where $j = 1, \ldots .50$). b_i and ϵ_i are generated from the multivariate normal and multivariate t-distributions separately to see the performance of the proposed method for a heavy-tailed data set. Additionally, outliers are added to the multivariate normal case simulations. These outliers are generated by adding "100" to the last 5 observations of all DSP. b_i is generated from a normal distribution with mean 0 and variance D; whereas ϵ_i is generated from a normal distribution with mean 0 and variance I. Then, b_i is generated from a multivariate t distribution with mean 0 and variance D; whereas ϵ_i is generated from a t-distribution with mean 0, variance I and degrees of freedom 4. Predictors $X_1, X_2,$ and X_3 are generated from uniform distributions with the interval [0,10]. Design of the simulation data is in Table 4.6 below.

Table 4.6: Design of the Simulation Study

DSP	Data Structure							
	Fixed Effect					Random Effect		
	Effect	μ_1	μ_2	μ_3	μ_4	Design	d_{11}	d_{22}	d_{12}
1	Large	-20	-10	10	20	No random	0.00	0.00	0.00
2	Small	10	11	12	13	effect	0.00	0.00	0.00
3	Large	-20	-10	10	20	Random	0.25	0.00	0.00
4						intercept	0.50	0.00	0.00
5	Small	10	11	12	13		0.25	0.00	0.00
6							0.50	0.00	0.00
7	Large	-20	-10	10	20	Random	0.25	0.25	0.00
8						intercept and	0.50	0.50	0.00
9	Small	10	11	12	13	covariate X_1	0.25	0.25	0.00
10						with 0	0.50	0.50	0.00
						correlation			
11	Large	-20	-10	10	20	Random	0.25	0.25	0.125
12						intercept and	0.50	0.50	0.25
13	Small	10	11	12	13	covariate	0.25	0.25	0.125
14						X_1 with 0.5	0.50	0.50	0.25
						correlation			

Simulation results for each DSP are presented in following tables and interpreted at the end of the tables.

Table 4.7: DSP 1

	LMM	**REEMtree**	**Heavy_REEMtree**
Estimated variance of errors (Normal)	**8.146**	66.370	11.865
Log likelihood(Normal)	-17589	-7101.776	**-4981.634**
RMSE(Normal)	8.143	**0.998**	7.235
AIC(Normal)	35190	14215.550	**9969.273**
BIC(Normal)	35229.090	14254.650	**9988.820**
Estimated variance of errors(Multivariate T)	**8.230**	67.741	344.933
Log likelihood(Multivariate T)	-17640.270	-9577.250	**-8839.983**
RMSE(Multivariate T)	8.226	**1.636**	7.9860
AIC(Multivariate T)	35292.540	19166.500	**17685.970**
BIC(Multivariate T)	35331.640	19205.600	**17705.520**
Estimated variance of errors (Normal with Outlier)	8.775	10.267	**8.300**
Log likelihood(Normal with Outlier)	-17983.800	-13000.208	**-5251.806**
RMSE(Normal with Outlier)	8.739	**3.177**	17.488
AIC(Normal with Outlier)	35979.600	26012.420	**10509.620**
BIC(Normal with Outlier)	36018.690	26051.510	**10529.160**

Table 4.8: DSP 2

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	1.138	0.997	**0.915**
Log likelihood(Normal)	-7758.653	**-7099.030**	-7522.228
RMSE(Normal)	1.136	0.998	**0.989**
AIC(Normal)	15529.310	**14210.060**	15050.460
BIC(Normal)	15568.400	**14249.160**	15070.010
Estimated variance of errors(Multivariate T)	2.957	2.682	**1.290**
Log likelihood(Multivariate T)	-9825.211	-9574.908	**-8789.181**
RMSE(Multivariate T)	1.716	1.635	**1.622**
AIC(Multivariate T)	19662.420	19161.820	**17584.370**
BIC(Multivariate T)	19701.520	19200.910	**17603.910**
Estimated variance of errors (Normal with Outlier)	3.265	11.586	**4.857**
Log likelihood(Normal with Outlier)	-13099.310	-13294.119	**-9754.305**
RMSE(Normal with Outlier)	3.238	3.377	**3.369**
AIC(Normal with Outlier)	26210.630	26594.240	**19514.620**
BIC(Normal with Outlier)	26249.730	26613.790	**19534.160**

Table 4.9: DSP 3

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	67.227	**0.999**	1042.500
Log likelihood(Normal)	-17622.610	**-7178.557**	-11282.750
RMSE(Normal)	8.193	**0.991**	17.190
AIC(Normal)	35257.230	**14369.110**	22571.510
BIC(Normal)	35296.330	**14408.210**	22591.060
Estimated variance of errors(Multivariate T)	68.099	**2.086**	67.468
Log likelihood(Multivariate T)	-17695.930	-9131.825	**-7765.008**
RMSE(Multivariate T)	8.201	1.429	**6.768**
AIC(Multivariate T)	35403.860	18275.650	**15536.020**
BIC(Multivariate T)	35442.960	18314.750	**15555.570**
Estimated variance of errors (Normal with Outlier)	8.199	**10.188**	96.366
Log likelihood(Normal with Outlier)	-18009.700	-12985.436	**-9806.393**
RMSE(Normal with Outlier)	8.781	**3.164**	8.135
AIC(Normal with Outlier)	36031.410	25982.870	**19618.790**
BIC(Normal with Outlier)	36070.500	26021.970	**19638.340**

Table 4.10: DSP 4

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	67.228	**0.999**	223.433
Log likelihood(Normal)	-17629.750	**-7233.760**	-9352.849
RMSE(Normal)	8.182	**0.989**	6.789
AIC(Normal)	35271.510	**14479.520**	18711.700
BIC(Normal)	35310.610	**14518.620**	18731.250
Estimated variance of errors(Multivariate T)	68.099	**2.086**	26.149
Log likelihood(Multivariate T)	-17697.200	-9134.331	**-7654.980**
RMSE(Multivariate T)	8.200	**1.429**	25.095
AIC(Multivariate T)	35406.410	18280.660	**15315.970**
BIC(Multivariate T)	35445.500	18319.760	**15335.510**
Estimated variance of errors (Normal with Outlier)	**8.819**	10.188	69.440
Log likelihood(Normal with Outlier)	-18013.910	-12993.998	**-9541.060**
RMSE(Normal with Outlier)	8.777	**3.163**	4.917
AIC(Normal with Outlier)	36039.820	26000	**19088.130**
BIC(Normal with Outlier)	36078.920	26039.100	**19107.670**

Table 4.11: DSP 5

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	1.351	0.999	**0.917**
Log likelihood(Normal)	-7927.868	**-7178.563**	-7559.835
RMSE(Normal)	1.153	0.991	**0.989**
AIC(Normal)	15867.740	**14369.130**	15125.680
BIC(Normal)	15906.830	**14408.230**	15145.220
Estimated variance of errors(Multivariate T)	2.426	2.185	**1.482**
Log likelihood(Multivariate T)	-9507.743	-9243.588	**-8575.773**
RMSE(Multivariate T)	1.542	**1.463**	**1.463**
AIC(Multivariate T)	19027.490	18497.180	**17157.550**
BIC(Multivariate T)	19066.580	18529.760	**17177.100**
Estimated variance of errors (Normal with Outlier)	**3.261**	11.586	5.170
Log likelihood(Normal with Outlier)	-13098.220	-13297.827	**-9821.307**
RMSE(Normal with Outlier)	**3.234**	3.376	3.369
AIC(Normal with Outlier)	26208.440	26601.660	**19648.620**
BIC(Normal with Outlier)	26247.540	26621.210	**19668.170**

Table 4.12: DSP 6

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	1.351	0.999	**0.915**
Log likelihood(Normal)	-7981.216	**-7233.760**	-7586.200
RMSE(Normal)	1.151	0.989	**0.989**
AIC(Normal)	15974.430	**14479.520**	15178.410
BIC(Normal)	16013.530	**14518.620**	15197.950
Estimated variance of errors(Multivariate T)	2.426	2.185	**1.482**
Log likelihood(Multivariate T)	-9510.215	-9246.072	**-8578.495**
RMSE(Multivariate T)	1.542	**1.463**	**1.463**
AIC(Multivariate T)	19032.430	18502.150	**17162.990**
BIC(Multivariate T)	19071.530	18534.730	**17182.540**
Estimated variance of errors (Normal with Outlier)	**3.261**	11.586	5.146
Log likelihood(Normal with Outlier)	-13106.710	-13306.079	**-9834.154**
RMSE(Normal with Outlier)	**3.233**	3.375	3.369
AIC(Normal with Outlier)	26225.410	26618.160	**19674.310**
BIC(Normal with Outlier)	26264.510	26637.710	**19693.860**

Table 4.13: DSP 7

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	69.080	**3.081**	141.974
Log likelihood(Normal)	-17783.150	-10153.616	**-8670.609**
RMSE(Normal)	8.237	**1.737**	3.729
AIC(Normal)	35578.300	20319.230	**17347.220**
BIC(Normal)	35617.390	20358.330	**17366.770**
Estimated variance of errors(Multivariate T)	73.863	**8.065**	1086.655
Log likelihood(Multivariate T)	-18005.120	-12573.807	**-11114.790**
RMSE(Multivariate T)	8.509	**2.810**	8.540
AIC(Multivariate T)	36022.240	25159.620	**22235.590**
BIC(Multivariate T)	36061.340	25198.710	**22255.130**
Estimated variance of errors (Normal with Outlier)	8.928	**12.358**	617.165
Log likelihood(Normal with Outlier)	-18137.940	-13558.295	**-12718.867**
RMSE(Normal with Outlier)	8.849	**3.480**	7.531
AIC(Normal with Outlier)	36287.890	27128.590	**25443.740**
BIC(Normal with Outlier)	36326.990	27167.690	**25463.290**

Table 4.14: DSP 8

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	70.378	**4.605**	210.764
Log likelihood(Normal)	-17859.990	-11171.896	**-9105.432**
RMSE(Normal)	8.309	**2.123**	5.981
AIC(Normal)	35731.980	22355.790	**18216.870**
BIC(Normal)	35771.080	22394.890	**18236.420**
Estimated variance of errors(Multivariate T)	71.154	**6.105**	447.041
Log likelihood(Multivariate T)	-17887.220	-11862.635	**-11512.257**
RMSE(Multivariate T)	8.354	**2.445**	8.736
AIC(Multivariate T)	35786.430	23737.270	**23030.520**
BIC(Multivariate T)	35825.530	23776.370	**23050.070**
Estimated variance of errors (Normal with Outlier)	**8.991**	13.728	52.167
Log likelihood(Normal with Outlier)	-18204.950	-13852.242	**-8630.168**
RMSE(Normal with Outlier)	8.905	**3.667**	6.372
AIC(Normal with Outlier)	36421.910	27716.480	**17266.340**
BIC(Normal with Outlier)	36461	27755.580	**17285.890**

Table 4.15: : DSP 9

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	3.424	**3.081**	3.689
Log likelihood(Normal)	-10417.740	-10153.291	**-9695.359**
RMSE(Normal)	1.831	**1.737**	**1.744**
AIC(Normal)	20847.470	20318.580	**19396.720**
BIC(Normal)	20886.570	20357.680	**19416.270**
Estimated variance of errors(Multivariate T)	8.393	8.295	**5.696**
Log likelihood(Multivariate T)	-12677.560	-12641.383	**-10733.760**
RMSE(Multivariate T)	2.867	**2.851**	**2.853**
AIC(Multivariate T)	25367.130	25290.770	**21473.530**
BIC(Multivariate T)	25406.230	25316.830	**21493.070**
Estimated variance of errors (Normal with Outlier)	**3.576**	13.586	6.819
Log likelihood(Normal with Outlier)	-13649.290	-13788.731	**-10619.941**
RMSE(Normal with Outlier)	**3.540**	**3.650**	**3.648**
AIC(Normal with Outlier)	27310.590	27583.460	**21245.890**
BIC(Normal with Outlier)	27349.690	27603.010	**21265.430**

Table 4.16: : DSP 10

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	4.922	**4.603**	5.148
Log likelihood(Normal)	-11341.220	-11170.752	**-10462.578**
RMSE(Normal)	2.195	**2.123**	**2.124**
AIC(Normal)	22694.450	22353.510	**20931.160**
BIC(Normal)	22733.550	22392.600	**20950.710**
Estimated variance of errors(Multivariate T)	6.374	**6.097**	**11.243**
Log likelihood(Multivariate T)	-11974.390	-11859.413	**-11330.348**
RMSE(Multivariate T)	2.498	**2.443**	**2.469**
AIC(Multivariate T)	23960.780	23730.830	**22666.700**
BIC(Multivariate T)	23999.880	23769.930	**22686.250**
Estimated variance of errors (Normal with Outlier)	**3.758**	14.619	9.611
Log likelihood(Normal with Outlier)	-13927.780	-14004.315	**-11153.914**
RMSE(Normal with Outlier)	3.719	3.785	**3.785**
AIC(Normal with Outlier)	27867.560	28014.630	**22313.830**
BIC(Normal with Outlier)	27906.660	28034.180	**22333.380**

Table 4.17: : DSP 11

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	69.080	**3.081**	141.974
Log likelihood(Normal)	-17783.150	-10153.616	**-8670.609**
RMSE(Normal)	8.237	**1.737**	3.729
AIC(Normal)	35578.300	20319.230	**17347.220**
BIC(Normal)	35617.390	20358.330	**17366.770**
Estimated variance of errors(Multivariate T)	73.863	**8.065**	1086.655
Log likelihood(Multivariate T)	-18005.120	-12573.807	**-11114.790**
RMSE(Multivariate T)	8.509	**2.810**	8.509
AIC(Multivariate T)	36022.240	25159.620	**22255.130**
BIC(Multivariate T)	36061.340	25198.710	**22255.130**
Estimated variance of errors (Normal with Outlier)	8.906	**12.358**	561.397
Log likelihood(Normal with Outlier)	-18127.090	-13558.295	**-13472.162**
RMSE(Normal with Outlier)	8.828	**3.480**	7.531
AIC(Normal with Outlier)	36266.190	27128.590	**26950.330**
BIC(Normal with Outlier)	36305.290	27167.690	**26969.880**

Table 4.18: : DSP 12

	LMM	**REEMtree**	**Heavy_REEMtree**
Estimated variance of errors (Normal)	73.418	**7.687**	738.780
Log likelihood(Normal)	-17990.750	-12456.576	**-10912.563**
RMSE(Normal)	8.483	**2.744**	6.603
AIC(Normal)	35993.500	24925.150	**21831.130**
BIC(Normal)	36032.600	24964.250	**21850.680**
Estimated variance of errors(Multivariate T)	75.504	**9.807**	230.531
Log likelihood(Multivariate T)	-18080.310	-13074.334	**-11219.357**
RMSE(Multivariate T)	8.602	**3.099**	12.727
AIC(Multivariate T)	36172.620	26160.670	**22444.720**
BIC(Multivariate T)	36211.720	26199.770	**22464.270**
Estimated variance of errors (Normal with Outlier)	**8.906**	12.374	774.644
Log likelihood(Normal with Outlier)	-18127.090	-13562.581	**-11980.207**
RMSE(Normal with Outlier)	8.828	**3.482**	7.0880
AIC(Normal with Outlier)	36266.190	27137.160	**23966.420**
BIC(Normal with Outlier)	36305.290	27176.260	**23985.970**

Table 4.19: : DSP 13

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	3.424	**3.081**	3.689
Log likelihood(Normal)	-10417.740	-10153.291	**-9695.359**
RMSE(Normal)	1.831	**1.737**	**1.744**
AIC(Normal)	20847.470	20318.580	**19396.720**
BIC(Normal)	20886.570	20357.680	**19416.270**
Estimated variance of errors(Multivariate T)	8.393	8.295	**5.696**
Log likelihood(Multivariate T)	-12677.560	-12641.383	**-10733.760**
RMSE(Multivariate T)	2.867	**2.851**	**2.853**
AIC(Multivariate T)	25367.130	25290.770	**21473.530**
BIC(Multivariate T)	25406.230	25316.830	**21493.070**
Estimated variance of errors (Normal with Outlier)	**3.576**	13.586	6.819
Log likelihood(Normal with Outlier)	-13649.290	-13788.731	**-10619.941**
RMSE(Normal with Outlier)	3.540	3.650	**3.648**
AIC(Normal with Outlier)	27310.590	27583.460	**21245.890**
BIC(Normal with Outlier)	27349.690	27603.010	**21265.430**

Table 4.20: : DSP 14

	LMM	REEMtree	Heavy_REEMtree
Estimated variance of errors (Normal)	8.012	9.655	**5.713**
Log likelihood(Normal)	-12564.060	-13035.258	**-10552.960**
RMSE(Normal)	2.801	3.075	**2.791**
AIC(Normal)	25140.120	26078.520	**21111.930**
BIC(Normal)	25179.210	26104.580	**21131.470**
Estimated variance of errors(Multivariate T)	10.122	10.057	**7.996**
Log likelihood(Multivariate T)	-13157.850	-13134.935	**-11115.771**
RMSE(Multivariate T)	**3.148**	**3.139**	**3.140**
AIC(Multivariate T)	26327.700	26277.870	**22237.550**
BIC(Multivariate T)	26366.800	26303.940	**22257.090**
Estimated variance of errors (Normal with Outlier)	**3.575**	13.556	6.803
Log likelihood(Normal with Outlier)	-13647.890	-13784.290	**-10589.865**
RMSE(Normal with Outlier)	3.539	3.646	**3.644**
AIC(Normal with Outlier)	27307.770	27574.580	**21185.740**
BIC(Normal with Outlier)	27346.870	27594.130	**21205.280**

According to simulation results, for multivariate normal, multivariate t-distribution and multivariate normal with outliers cases, Heavy_REEMtree gives the best results in terms of log-likelihood, AIC and BIC for almost all DSP with the optimal degrees of freedom since the degrees of freedom is setted accordingly. In addition to this, Heavy_REEMtree gives the best results for many cases in terms of RMSE results, especially for the multivariate-t and multivariate normal with outliers cases. The model performance of Heavy_REEMtree gets better when covariance effect is added to the random effect structure. In other words, the model performance of Heavy_REEMtree shows improvement over other models for between DSP1 and DSP14. Changing the degrees of freedom effects, the result significantly changes, so the degrees of freedom can be examined in detail for further studies.

CHAPTER 5

CONCLUSION

Real-life longitudinal data sets mostly consist of missing values, they are often unbalanced, and it is difficult to meet assumptions of statistical methods for these type of data sets. For instance, random errors might not be distributed normally and data sets may have outliers originating from within- or between-observations. Moreover, since longitudinal data sets are not *iid* by its nature, using machine learning methods instead of statistical methods may not be the best option to handle longitudinal data. Therefore, hybrid methods which are integrating machine learning algorithms and statistical methods are alternatively used in sort of situations. For this reason, this study focused on machine learning algorithms and statistical methods for longitudinal data and their hybrid applications.

The aim of this study is adapting a robust model into hybrid applications of machine learning algorithms and statistical methods for longitudinal data to be able to handle with the heavy-tailed distributed data and the outliers. Proposed robust model is synthesizing RT with LMM under heavy-tailed distribution. Motivation of the robust model arising from proposing hybrid robust model when data are not distributed normally or appearance of outliers by using help of proven robust models, such as multivariate t-distribution by Pinheiro et al. [15], using Laplace distribution by Gokalp Yavuz and Arslan [28], using (skew) t-distribution by Chou and Tsung-I [29], and hybrid method of mixture of RT and LMM which is called RE-EM trees [13].

The proposed method, Heavy REEMTree is implemented via R programming language with the combination of package ‘REEMtree’ and ‘heavyLme’ function under ‘heavy’ package [18]. The related codes are provided at https://github.com/hakkierduran/heavycodes.

In the first part of the data analysis section, two real data sets are used to conduct LMM, REEMtree and Heavy_REEMtree. Results indicated that;

- Heavy_REEMtree shows better performance over REEMtree in terms of all performance measurements.
- Heavy_REEMtree imparts more detailed RT and precise estimation compared to REEMtree.
- Heavy_REEMtree presents improvements over LMM in terms of some performance measurements, such as log-likelihood, AIC and BIC
  LMM performs well in terms of some performance measurements, whereas RMSE, for one case. , such as

Simulation study is set according to the case of hybrid methods are needed. To clarify, 14 different scenarios simulated the large volume of data sets based on tree rules for normally distributed, multivariate t-distributed and normally distributed with outliers random terms. Degrees of freedom for Heavy_REEMtree is tuned while estimation process. Simulation results revealed that;

- Heavy_REEMTree surpass both LMM and REEMTree nearly in all simulation scenarios with regard to performance measurements.
- Tuning degrees of freedom for Heavy_REEMTree made huge impact on estimation results.

Real data analysis and simulation studies proves the superiority of the proposed method Heavy_REEMtree over REEMTree in the presence of heavy tailed data and the outliers. But it should be borne in mind that the effect of the degrees of freedom for Heavy_REEMtree estimation results require further investigation. Existing programming packages are not designed to optimize degrees of freedom. In future studies, it is planned to perform this optimization and add it to the R package. Also, as mentioned in this study, there are many hybrid algorithms as we use. Considering this study as a starting point, it is aimed to consolidate it in other methods in subsequent studies.

REFERENCES

[1] G. Verbeke and E. Lesaffre, "A Linear Mixed-Effects Model With Heterogeneity in the Random-Effects Population," *Journal of the American Statistical Association,* pp. 217-221, 1996.

[2] A. Cnaan, N. M. Laird and P. Slasor, "Using the general linear mixed model to analyse unbalanced repeated measures and longitudinal data," *Statistics in Medicine,* pp. 2349-2380, 1997.

[3] D. Zhang, X. Lin, J. Raz and M. Sowers, "Semiparametric Stochastic Mixed Models for Longitudinal Data," *Journal of the American Statistical Association,* pp. 710-719, 1998.

[4] M. J. Lindstrom and D. M. Bates, "Nonlinear Mixed Effects Models for Repeated Measures Data," *Biometrics,* pp. 673-687, 1990.

[5] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics,* pp. 1189-1232, 2001.

[6] L. Breiman , "Random Forests," *Machine Learning,* pp. 5-32, 2001.

[7] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, NY, 1995.

[8] C. Tianle, D. Zeng and Y. Wang, "Multiple Kernel Learning with Random Effects for PredictingLongitudinal Outcomes and Data Integration," *Biometrics,* pp. 918-928, 2015.

[9] A. Pand, L. Li, R. Jeevanantham, J. Ehrlinger and H. Ishwaran, "Boosted multivariate trees for longitudinal data," *Machine Learning,* pp. 277-305,

2017.

[10] Y. LeCun, G. Hinton and . Bengio, "Deep Learning," *Nature,* pp. 436-444, 2015.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* pp. 1735-1780, 1997.

[12] A. Hajjem, F. Bellavance and D. Larocque, "Mixed effects regression trees for clustered data," *Statistics and Probability Letters,* pp. 451-459, 2011.

[13] R. J. Sela and J. S. Simonoff, "RE-EM trees: a data mining approach for longitudinal," *Machine Learning ,* pp. 169-207, 2012.

[14] A. Hajjem, D. Larocque and F. Bellavance, "Generalized mixed effects regression trees," *Statistics and Probability Letters,* pp. 114-118, 2017.

[15] J. C. Pinheiro, C. Liu and Y. N. Wu, "Efficient algorithms for robust estimation in linear mixed-effects models using the multivariate t distribution," *Journal of Computational and Graphical,* vol. 10, pp. 249-276, 2001.

[16] L. Breiman, J. H. Friedman, R. . A. Olshen and C. . J. Stone, Classification and regression trees, Monterey: Brooks/Cole Publishing, 1984.

[17] J. Pinheiro, D. Bates, D. Sarkar, S. Heisterkamp, B. V. Willigen and J. Ranke, "nlme: Linear and Nonlinear Mixed Effects Models," [Online]. Available: https://cran.r-project.org/web/packages/nlme/index.html. [Accessed 20 April 2021].

[18] F. Osorio, "heavy: Robust Estimation Using Heavy-Tailed Distributions," [Online]. Available: https://cran.r-project.org/web/packages/heavy/index.html. [Accessed 20 April 2021].

[19] N. M. Laird and J. . H. Ware, "Random-Effects Models for Longitudinal Data," *International Biometric Society,* pp. 963-974, December 1982.

[20] J. Luts, G. Molenberghs, G. Verbeke, S. V. Huffel and J. A. Suykens, "A mixed effects least squares support vector machine model for classification of longitudinal data," *Computational Statistics and Data Analysis,* pp. 611-628, 2012.

[21] C. Ngufor, H. V. Houten, B. Caffo, N. Shah and R. McCoy, "Mixed effect machine learning: A framework for predicting longitudinal change in hemoglobin A1c," *Journal of Biomedical Informatics,* pp. 56-67, 2019.

[22] A. Hajjem, F. Bellavance and D. Larocque, "Mixed-effects random forest for clustered data," *Journal of Statistical Computation and Simulation,* pp. 1313-1328, 2014.

[23] L. Capitaine, R. Genuer and R. Thiebaut, "Random forests for high-dimensional," *Statistical Methods in Medical Research,* pp. 166-184, 2021.

[24] C. Spanbauer and R. Sparapani, "Nonparametric machine learning for precision medicinewith longitudinal clinical trials and Bayesian additiveregression trees with mixed models," *Statistics in Medicine,* pp. 2665-2691, 2021.

[25] H. Deng, "Interpreting Tree Ensembles with inTrees," *International Journal of Data Science and Analytics,* pp. 277-287, 2019.

[26] T. Therneau and B. Atkinson, *rpart: recursive partitioning. R port by Brian Ripley,* Cran Project, 2010.

[27] J. A. Suykens , T. V. Gestel, J. D. Brabanter, B. . D. Moor and J. Vandewalle, Least Squares Support Vector Machines, World Scientific, 2012.

[28] F. Gokalp Yavuz and O. Arslan , "Linear mixed model with Laplace

distribution (LLMM)," *Statistical Papers,* vol. 59, pp. 271-289, 2018.

[29] H. H. Chou and T.-I. Lin, "Robust linear mixed models using the skew t distribution with application to schizophrenia data," *Biometrical Journal,* vol. 52, no. 4, pp. 449-469, 2010.

[30] J. C. Pinheiro and D. M. Bates, Mixed-Effects Models in S and S-PLUS, Springer, 2000.

[31] T. K. Moon, "The Expectation Maximization Algorithms," *IEEE Signal processing magazine,* pp. 47-60, 1996.

[32] T. Therneau and B. Atkinson, "rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15," [Online]. Available: https://CRAN.R-project.org/package=rpart. [Accessed 28 March 2021].

[33] R. J. Lewis, "An Introduction to Classification and Regression Tree (CART) Analysis," in *Annual Meeting of the Society for Academic Emergency Medicine*, California, 2000.

[34] R. Sela, J. Simonoff and W. Jing, "REEMtree: Regression Trees with Random Effects for Longitudinal (Panel) Data," [Online]. Available: https://cran.r-project.org/web/packages/REEMtree/index.html. [Accessed 20 April 2021].

[35] R.-c. Team, *Package 'nlme',* Cran Project, 2021.

[36] C. Liu and D. B. Rubin, "The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence," *Biometrika,* vol. 84, no. 4, pp. 633-648, 1994.

[37] A. Hajjem, F. Bellavance and D. Larocque, "Mixed effects regression trees for clustered data," *Statistics & Probability Letters,* vol. 81, no. 4, pp. 451-459, 2011.

[38] G. Fitzmaurice, N. Laird and J. Ware, Applied Longitudinal Analysis, John Wiley & Sons, 2011.

[39] J. D. Singer and J. B. Willett, Applied Longitudinal Data Analysis: Modeling, Oxford University Press, 2003.

[40] C. R. Henderson, "Analysis of Covariance in the Mixed Model: Higher-Level, Nonhomogeneous, and Random," *International Biometric Society,* pp. 623-640, 1982.

[41] F. Osorio, "heavy: Robust Estimation Using Heavy-Tailed Distributions," 20 October 2019. [Online]. Available: https://cran.r-project.org/web/packages/heavy/. [Accessed 10 April 2021].

APPENDICES

A. Example Pseudo Codes

1. Pseudo Codes for MEml:

Use convergence criterion or max iteration.

Assume random effect is known and set b_i to zero.

Step 1: After setting the response as response minus random effect, estimate the fixed effect components by using gradient boosting machine (GBM), RF, model-based recursive partitioning (MOB) or conditional inference tree (Ctree) and weights for each observation.
Step 2: Find indicator variables by using tree algorithms above.

Step 3: Construct GLMM model to estimate random effect $\widehat{b}_i$.

Repeat until convergence.

2. Pseudo Codes for MERT:

Use expectation-maximization algorithm. Assume, random effect is zero.

Step 1: Recalculate modified response, fixed effect part and random effect part.

a) Set new(modified) response as actual response minus currently estimated random effects.
b) Estimate fixed effect part by using random tree.
c) Estimate random effect part using currently modified response.

Step 2: Find covariance matrix of random effects and error.

Repeat first two steps until convergence.

3. Pseudo Codes for MERF:

All steps are the same as MERT, but in step1-b, the algorithm uses random forest with bootstrap sampling met.