2017

Decision Tree and Random Forest Methodology for Clustered and Longitudinal Binary Outcomes

Jaime Lynn Speiser
Medical University of South Carolina

Decision Tree and Random Forest Methodology for Clustered and Longitudinal Binary Outcomes

Jaime Lynn Speiser

A dissertation submitted to the faculty of the Medical University of South Carolina in partial fulfillment of the requirement for the degree of Doctor of Philosophy in the College of Graduate Studies.

Department of Public Health Sciences

2017

Approved by:

Valerie Durkalski, Co-Chair

Bethany Wolf, Co-Chair

Dongjun Chung

Constantine Karvellas

David Koch

DEDICATION

To my family (Blanche, Mark and Brad) for their unwavering support and belief in me,

To my friends (near and far) for all the fun times,

To my dog (Nelson) for his companionship and enthusiasm,

To my classmate (Nathan) for making this journey bearable,

To my city (Charleston) for the sunshine and amazing food.

ACKNOWLEDGEMENTS

It takes a village. I could not have asked for a better mentor team. Thank you to my co-mentors, Drs. Valerie Durkalski and Beth Wolf, for your support, patience, time, and wisdom. I appreciate all of your effort with submitting my pre-doctoral grant and letters of recommendation. Most of all, thank you for believing in me. Thank you to Dr. Dongjun Chung for being my "picky reviewer" when reading my papers and for your advice. Thank you to Drs. Dean Karvellas and David Koch for sharing your clinical knowledge with me and for the fun times at conferences.

Several others at the Medical University of South Carolina were instrumental in my schooling. Thank you to Dr. Viswanathan Ramakrishnan for convincing me to get my PhD while on the sideline of a tennis court and for your sage advice. Thank you to Dr. Betsy Hill for your support and for encouraging me to submit the TL1 grant. Thank you to June Watson for organizing everything for students in the department and for making sure I always had food in accordance with my dietary restrictions. Thank you to Vanessa Sullivan for helping me organize travel to conferences and for answering all of my many administrative questions.

None of this would have been possible without my undergraduate mentors at Elon University. Thank you to Dr. Ayesha Delpish for instilling me with the idea that I could earn my PhD. Words cannot express how grateful I am for your guidance, encouragement and friendship. Thank you to Dr. Steven House for including me in the dean team office and for inspiring me to pursue this path. Thank you to the Elon College Fellows program for helping me hone my curiosity and for giving me an introduction to conducting research.

I would also like to acknowledge my funding sources. This dissertation was funded by the National Institute of Diabetes and Digestive and Kidney Diseases NIH/NIDDK (U01-58369) and the South Carolina Clinical and Translational Research Institute NIH/NCATS Grant (UL1-TR001450 and TL1-TR001451).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Clustered binary outcomes are frequently encountered in medical research (e.g. longitudinal studies). Generalized linear mixed models (GLMMs) typically employed for clustered endpoints have challenges for some scenarios (e.g. high dimensional data). In the first dissertation aim, we develop an alternative, data-driven method called Binary Mixed Model (BiMM) tree, which combines decision tree and GLMM. We propose a procedure akin to the expectation maximization algorithm, which iterates between developing a classification and regression tree using all predictors and developing a GLMM which includes indicator variables for terminal nodes from the tree as predictors along with a random effect for the clustering variable. Since prediction accuracy may be increased through ensemble methods, we extend BiMM tree methodology within the random forest setting in the second dissertation aim. BiMM forest combines random forest and GLMM within a unified framework using an algorithmic procedure which iterates between developing a random forest and using the predicted probabilities of observations from the random forest within a GLMM that contains a random effect for the clustering variable. Simulation studies show that BiMM tree and BiMM forest methodology offer similar or superior prediction accuracy compared to standard classification and regression tree, random forest and GLMM for clustered binary outcomes. The new BiMM methods are used to develop prediction models within the acute liver failure setting using the first seven days of hospital data for the third dissertation aim. Acute liver failure is a rare and devastating condition characterized by rapid onset of severe liver damage. The majority of prediction models developed for acute liver failure patients use admission data only, even though many clinical and

laboratory variables are collected daily. The novel BiMM tree and forest methodology developed in this dissertation can be used in diverse research settings to provide highly accurate and efficient prediction models for clustered and longitudinal binary outcomes.

1 INTRODUCTION

1.1 Overview

There are many statistical models which may be used to classify observations into pre-defined outcome groups. From traditional parametric models such as linear discriminant analysis to newer algorithmic techniques like random forest (RF), there is no shortage of choices for statistical classification procedures. A multitude of factors must be considered when selecting the modeling framework. The structure and distribution of variables within the dataset must be examined, model assumptions checked and missing data assessed before a classification method may be chosen. One must also consider the overall goal of the model and whether the purpose is for use as a prediction model or to assess relationships between variables and the outcome. For example, if a goal is to understand how predictor variables relate to an outcome, then all variables might be included for modeling; however, this may produce a complex model that requires collection of many variables, which may not be practical in clinical settings for obtaining predictions of outcome. Decisions must be made about assessing predictor variables and prediction accuracy, which are often opposing factors in classification problems. As model development is considered an art, so too is choosing a classification method.

Selecting a statistical method for studies in which a categorical outcome is of primary interest can be challenging in the setting of clustered and longitudinal outcomes. Since some standard models, such as linear or logistic regression, assume independence of outcome observations, alternative models must be employed (e.g. mixed models) to account for clustered outcomes, which occurs when outcomes are correlated within a group. For instance, studies which collect outcomes for several family members would be

clustered within family groups because people in the same family may have shared characteristics which make their outcome observations dependent. Another example of clustered data occurs in longitudinal or repeated measures studies, in which outcome variables are collected at multiple time points for each subject. For example, a longitudinal study may evaluate prognosis (e.g. good, moderate, or poor outcome) in a disease setting, diagnosis (e.g. if a patient has a disease or does not have a disease), or other outcomes (e.g. if a patient will be re-admitted or not re-admitted to the hospital) over time. Outcomes collected at multiple time points for each subject results in a correlation structure since values of the same variable for a subject are dependent. This should be modeled properly so that the assumption of independent observations required by some statistical methods (e.g. standard regression) is not violated.

Though many statistical methods have been extended to account for clustered binary outcomes, newer machine learning procedures such as decision trees and RFs provide unprecedented opportunities to investigate these types of outcomes. A commonly used decision tree method is classification and regression trees (CART) [1], which allows the development of predictive models using binary splits on variables which can be read like a flow chart. Gaining popularity in diverse medical fields [2, 3], CART models offer an intuitive method for predicting outcome, using processes consistent with clinical interpretation of predictors (e.g. “high” versus “low” values of a predictor). First introduced by Breiman in 2001 [4], RFs are a collection of CARTs [1] which are constructed using randomly selected training datasets and random subsets of predictor variables for prediction of a categorical or continuous outcome. CART and RF offer many benefits compared to traditionally used classification procedures. Unlike standard

regression methods, CART and RF can be used in the setting of high dimensional data and can handle interactions between predictors and nonlinear relationships between predictors and outcome without the need for user specification of such relationships [5]. A benefit of CART and RF compared to other machine learning methods is that the former methods can determine which variables are important in the model, which is especially important in clinical prediction modeling where both prediction and interpretation of predictors related to outcome are of interest.

While CART and RF can often better predict outcomes compared to other procedures such as logistic regression, discriminant analysis, and support vector machines for data captured at a single time point [6], these methods need to be thoroughly investigated for clustered and longitudinal outcomes. Methods for employing CART and RF for data with clustered outcomes thus far have largely focused on continuous outcomes, so development of the methodology for binary outcomes is warranted.

1.2 Gaps in the Current Literature

There are several methods available to implement CART and RF models for clustered continuous outcomes [7-16]. Laroque [12], Hajjem [10, 17] and Sela [15] proposed similar methods for implementing CART and RF models for longitudinal or clustered data with continuous outcomes. These methods incorporate mixed effects within the tree framework to account for the correlation structure within the data, using an algorithm analogous to expectation-maximization described by Wu and Zhang [18]. Though there are many CART and RF models available for continuous clustered outcomes, there is a paucity of methodology for modeling clustered binary and

categorical endpoints. For example, the R package *party* can be used to implement CART and RF models if two predictor variables are correlated, but it does not adjust for correlations resulting from clustered and longitudinal outcomes [19]. There are some techniques which circumvent the issue of adjusting for clustered outcome measures, such as summarizing variables (e.g. using averages or most frequent categorical values within the cluster) or using data from only a single time point (e.g. admission values); however, these methods have a marked loss of information since available data is summarized or partially used.

Adjusting the continuous outcome methods proposed by Laroque [12], Hajjem [10, 17] and Sela [15] for clustered categorical outcomes is non-trivial. For continuous endpoints, outcomes are updated at each iteration based on fixed and random variables using an additive effect. For categorical outcomes, the optimal method for updating outcomes is unclear because an effect cannot simply be added. Another consideration within the generalized model setting for categorical outcomes is that an iterative procedure (e.g. iterative reweighted least squares or Newton Raphson) must be used to calculate random effects of clustering variables for generalized linear mixed models (GLMMs). For complex datasets, GLMMs may not converge, or other computational issues may arise (e.g. inverting large covariance matrices) which make GLMM estimation challenging [20]. Also, if data are quasi-separated or completely separated (meaning that one or a combination of variables perfectly predicts the outcome), traditional implementations of GLMMs cannot be used [21, 22]. Thus, developing CART and RF methods for clustered and longitudinal categorical outcomes is not an easy extension from continuous methods proposed in the literature.

1.3 Motivating Example

A specific motivating example dataset for the proposed aims is the Acute Liver Failure Study Group (ALFSG) registry funded by the National Institute of Diabetes and Digestive and Kidney Diseases (U01-DK-58369-10). Affecting an estimated 2,000 people per year in the United States [23], acute liver failure (ALF) is an orphan condition. To date, the registry consists of over 3,000 patients who have been affected by ALF. Study data are collected daily over seven days following enrollment or until a transplant or hospital discharge occurs, and patients can be followed for one year. The most common cause of ALF is acetaminophen overdose, which accounts for approximately half of the cases captured in the ALFSG database. Acetaminophen is the main active ingredient in many over-the-counter pain relievers and cold/fever medications such as Tylenol, Excederin, Sudafed, and DayQuil.

Data for an example patient from the ALFSG registry is depicted in Table 1.1. The patient is a 37-year old, non-Hispanic/Latino female. She has a primary diagnosis of acetaminophen and was not wait-listed for a transplant. The patient was discharged from the hospital after seven days. Table 1.1 contains a few laboratory and clinical variables collected in the ALFSG registry. There are many other variables not depicted within Table 1.1 which are collected within the ALFSG registry. Information collected includes, but is not limited to, patients' medical history, risk factors and past medications, physical exams including neurological status, imaging, laboratory data, daily updates, vital signs, transplant status and various other clinical characteristics.

A goal of the ALFSG is to develop models to predict the outcome (poor or favorable condition) of ALF patients using daily data collected from the first week of

hospitalization. An important question for clinicians is the prompt and accurate daily prediction of the condition of ALF patients who overdosed on acetaminophen so that the decision of whether or not to list the patient for liver transplant can be made. Though many patients have high likelihood of survival with a new liver, transplantation

Table 1.1: Example patient data from ALFSG registry

Patient	Day	Age	Gender	Coma Grade	INR	ALT	Creatinine	Pressors
1234	1	37	Female	High	1.7	4447	0.57	Yes
1234	2	37	Female	High	1.7	6951	0.61	Yes
1234	3	37	Female	Low	1.7	5848	0.64	Yes
1234	4	37	Female	High	1.7	4437	0.61	Yes
1234	5	37	Female	Low	1.8	3474	0.68	No
1234	6	37	Female	Low	1.9	2248	0.60	No
1234	7	37	Female	Low	1.8	2106	0.61	No

for acetaminophen-induced ALF often presents significant challenges in management due to the rapidity and severity of illness, the potential for recovery without a transplant, and complex psychosocial issues, such as depression, in many patients [24, 25]. With advances in treatments available to those with ALF, patients who would have otherwise died remain alive past hospital admission. Several prognosis models are available at the time a patient is admitted to the hospital (e.g. King's College Criteria and Clichy Criteria [26, 27]); however, prediction of outcome at later time points appears less accurate [28].

There are challenges associated with the ALFSG dataset which make developing accurate prediction models difficult. Many laboratory variables collected within the ALF registry, such as INR and ionized calcium, have skewed distributions, which means that the user must specify a transformation of the predictor in order to meet linearity

assumptions for standard linear mixed models. Choosing a transformation function is sometimes challenging because several functions may satisfy linearity assumptions, and using transformations often makes interpretation of estimates for predictor variables difficult. Additionally, there may be complex interactions among predictor variables, but there is little guidance in the clinical literature about these relationships which limits *a priori* specification of interaction terms. Also, linear mixed models may be sensitive to outliers and data containing many extreme values, so these methods may perform poorly for the ALFSG data, which contain many extreme values (particularly for laboratory variables).

There are several obstacles to developing accurate models to predict the daily condition of acetaminophen-induced ALF patients which may be used both on hospital admission and post-admission. The challenges described above often associated with clinical datasets are common in many disease settings. We employ CART and RF as the statistical modeling tools because they offer several solutions to these problems: both methods naturally model nonlinear relationships and complex interactions among predictors without user specification, and can sometimes provide higher prediction accuracy compared to many other classification procedures. CART and RF offer alternative methods for some situations when traditional models (logistic regression or GLMMs) are not optimal; namely, if the number of predictor variables is greater than the number of observations or if the predictor variables contain many extreme values [5]. Additionally, CART and RF can capture nonlinear patterns between predictors and the outcome of interest, without *a priori* knowledge of a nonlinear form [5]. For these reasons, CART and RF can often better predict outcomes compared to other procedures

such as discriminant analysis and logistic regression for data captured at a single time point [6].Though CART and RF clearly have many advantages over some traditionally used models, there is no existing method for modeling clustered and longitudinal categorical endpoints.

Although our motivating example is within the acute, rare disease setting of ALF, there are many clinical settings where there is a need for methodology for modeling clustered categorical outcomes. For instance, our proposed methodology could be used to develop models for disease relapse for patients such as multiple sclerosis or lupus nephritis. Additionally, models could be developed to determine the likelihood a patient will be re-admitted into the hospital. Thus, proposed methodology to model clustered binary outcomes may be applied in myriad settings.

1.4 Overall Goal and Specific Aims

The goal of this dissertation is to develop statistical methodology which extends CART and RF framework for clustered binary outcomes, and to develop prediction models for the ALF setting which can be used throughout hospitalization. Specifically, the objectives for this dissertation are:

- Aim 1: To develop a CART method for clustered and longitudinal binary outcomes using an iterative procedure to combine CART and mixed effect models
- Aim 2: To develop a RF method for clustered and longitudinal binary outcomes using an iterative procedure to combine RF and mixed effect models
- Aim 3: To develop a prediction model for daily outcomes of acetaminophen-induced ALF patients.

2 STATISTICAL BACKGROUND

The main focus of this dissertation is to extend methodologies within the CART and RF setting for longitudinal and clustered datasets with binary outcomes. In this section, common classification methods used for non-clustered and clustered data are reviewed, with emphasis on the CART and RF methodology. An overview of CART and RF classification is presented.

2.1 Classification Methods for Non-Clustered Data

There are a number of statistical procedures which may be used to model categorical outcomes when observations are not longitudinal or clustered. In this section, we focus on methods which may be employed when predictor variables are collected at a single time point (i.e. the data does not contain longitudinal measures or clustering variables). Commonly used classification methods include linear procedures, support vector machines, neural networks, and decision trees.

A popular linear classifier is logistic regression, which may be used if the outcome of interest is binary (e.g. alive or dead). Easily applied to datasets within virtually any statistical computing platform, logistic regression is simple and offers straight-forward interpretations when there are only a small number of predictors (e.g. less than 20). The odds of the outcome are modeled as a linear function of predictor variables, using the logit link. Potential pitfalls of logistic regression include: model assumptions may not be appropriate in all applications (e.g. linearity of the logit), interactions between several predictor variables complicate interpretation of model parameters, and lack of applicability for complex datasets (e.g. when the number of predictor variables is greater than the number of observations) [29]. Logistic regression

may also provide poor fit if the event rates of the outcome are at the extremes, either very small or large.

Another parametric classification method is linear discriminant analysis, in which outcome classes are separated using linear decision boundaries. Several assumptions made in linear discriminant analysis include: common covariance across classes, classes are linearly separable using hyperplanes, and predictor variables follow a multivariate normal distribution. Though violations of the assumption of common covariance matrix of classes may be addressed using quadratic discriminant analysis, the parametric nature of discriminant analysis may be limiting in some settings (e.g. for applications of ALFSG registry data since variables are not normally distributed). Another limitation of discriminant analysis is that the number of parameters which need to be estimated is often quite large, so this procedure may not be applicable for datasets which have many predictor variables and for outcomes which have many categories [29-31].

Support vector machines (SVMs) extend the method of linear discriminants, through construction of nonlinear boundaries for classes. While SVMs often provide high classification accuracy for complex training datasets, there may be difficulty determining an appropriate kernel distribution, and there is no way to interpret and assess variables included in the model. Furthermore, SVMs are prone to overfitting, so accuracy may decrease for testing or validation datasets. For this reason, SVMs are not ideal in many clinical settings, where assessing predictor variables is an essential part of model-building. Another limitation of SVMs is that they can be sensitive to outliers and data containing many extreme values, which is the case for many important clinical variables within the ALF setting [32].

Another machine learning classification procedure is a neural network, in which linear combinations of predictor variables are constructed (called hidden layers) to model the outcome of interest. Though neural networks offer a multi-stage modeling strategy in which latencies in the data are considered, there are several disadvantages of the method. Neural networks are sensitive to starting values and the scale of predictor variables, and are prone to overfitting [29]. Moreover, users must determine how many hidden layers to include, and choosing various amounts of hidden layers may lead to different models. Initial neural network models were created for binary outcomes only, and extensions for multi-class outcomes recently developed are computationally intensive and have low accuracy if classes are not balanced. Like SVMs, neural networks also lack interpretability of predictor variables [33-35].

2.1.1 Decision Trees for Non-Clustered Data

Decision trees are alternate classification methods which require fewer assumptions compared to parametric methods such as logistic regression and discriminant analysis. Read like a flow chart, decision trees develop predictive models using splits on variables. While there are many different ways to develop decision trees [36], one of the most common procedures implemented is classification and regression tree (CART) modeling. Developed by Breiman [1], the nonparametric nature of CART offers results which are simple to use (e.g. does not require calculation or use of an application) and interpret (e.g. low versus high variable values). These aspects of CART are advantageous compared to logistic regression and many other classification methods, where calculations may be cumbersome (e.g. plugging in numbers and exponentiation requires a calculator or application) and interpretation of results may be unclear (e.g. if there are

interactions between two or more predictors or if predictor variables cannot be readily and easily analyzed). CART also allows for inclusion of complex interactions among predictor variables. Modern advances in computing have provided efficient implementation of CART modeling using many standard statistical software platforms.

Binary partitions of data, selected with an exhaustive search, are used to construct CART models. For each predictor variable, every possible combination of sets (groupings of the values) is evaluated using a measure of impurity based on the distribution of the outcome in the node (splitting point). For categorical predictor variables, all combinations of the categories are assessed. For example, if a variable consisted of three groups, $\{1, 2, 3\}$, the possible splits are $\{1\}$ versus $\{2, 3\}$, $\{1, 2\}$ versus $\{3\}$, and $\{1, 3\}$ versus $\{2\}$. All combinations of the ordered values are assessed for continuous predictor variables. For instance, if a variable consisted of the values $\{2.2, 5.6, 4.5\}$, the possible sets for splits include $\{2.2\}$ versus $\{4.5, 5.6\}$ and $\{2.2, 4.5\}$ versus $\{5.6\}$. Notice that these splits could correspond to a number of non-unique cutoffs. Using the same example set, a split achieving the set $\{2.2\}$ versus $\{4.5, 5.6\}$ could use 3 as a cutoff, but any number between 2.2 and 4.5 would result in the same split. The set which minimizes the sum of the impurities of the outcome variable in the two resulting (sometimes called child or daughter) nodes is selected at each step of the CART algorithm.

CARTs are developed using the following algorithm (Figure 2.1). First, the predictor variable that optimally separates outcome groups is selected from the root node (containing all data), and a binary split is made (e.g. bilirubin < 2) which splits data at the

parent (present) node into two daughter nodes. Next, from both of these subgroups, another variable is selected with replacement, which best predicts outcome and binary splits are made. These splits are made recursively until stopping criteria are reached (e.g. when the threshold for the relative decrease in node impurity is reached), in which case a terminal node occurs. Each terminal node yields the outcome prediction for the specific subset of the data, along with the proportion of observations contained in each outcome class. Trees can become very complex; thus, it is advisable to prune CART models, a process in which unimportant variable splits are removed. Pruning not only simplifies tree models, but also makes them more generalizable to external datasets. CARTs are often pruned by selecting the complexity parameter that minimizes the cross-validated relative error rate [36, 37].

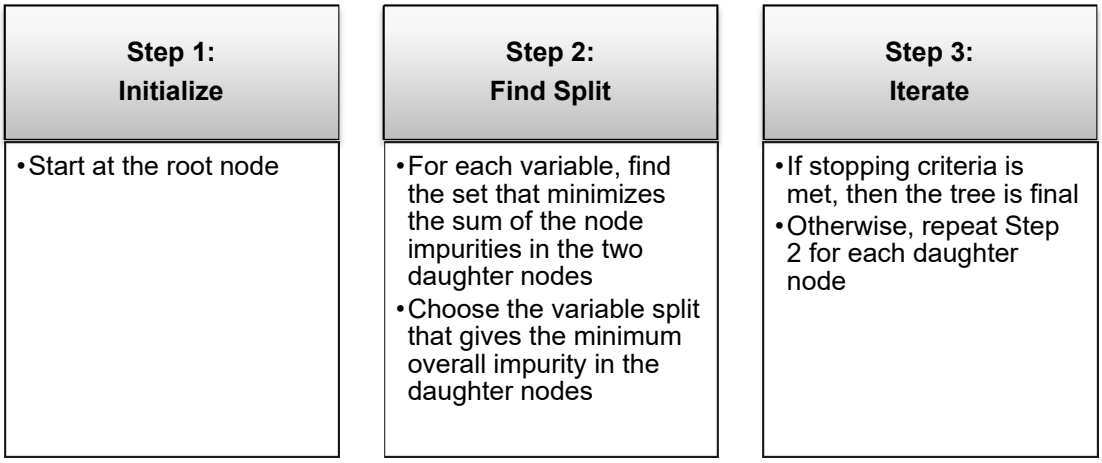
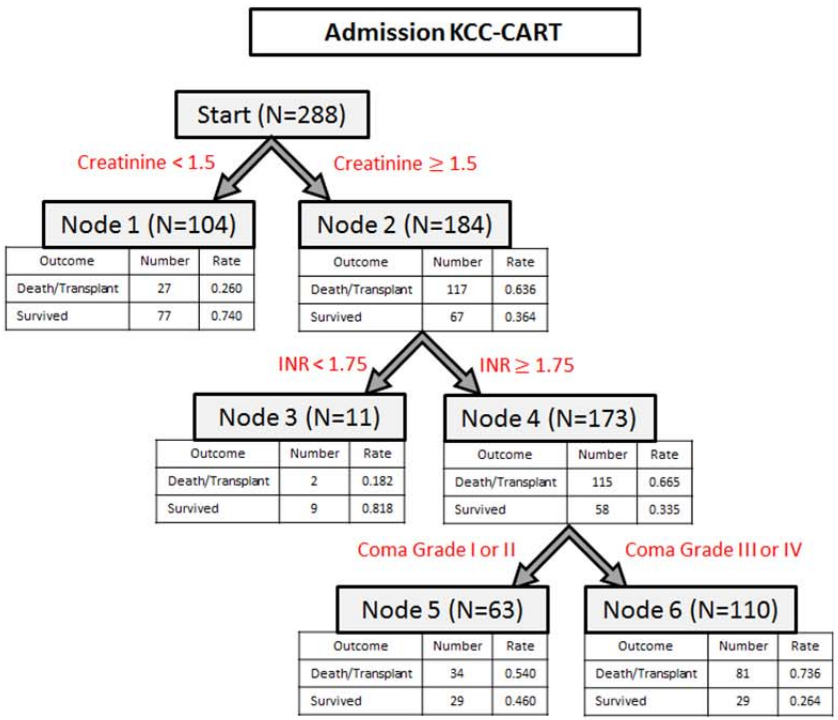


Figure 2.1: The CART algorithm

Two CARTs are presented within Figure 2.2 which predict the binary 21-day outcome (spontaneous survival versus death/liver transplant) for acetaminophen-induced acute liver failure patients at hospital admission [38]. The left panel displays a CART

developed using the same variables as a commonly used prognosis model, the King's College Criteria (denoted KCC-CART), and the right panel displays a CART developed using readily available laboratory and clinical variables (denoted NEW-CART). The admission KCC-CART (left panel of Figure 2.2) has three decision rules and consists of six total nodes. Each node provides the total number of subjects within the node, as well as the number of survivors and dead/transplant patients with the respective rates. Node 6 represents high risk of dead/transplant outcome, nodes 1 and 3 are low risk of dead/transplant outcome, and node 5 is moderate risk of dead/transplant outcome. To calculate performance measures for the model, all subjects in nodes 5 and 6 are predicted as dead/transplant outcomes, and all subjects in nodes 1 and 3 are predicted as spontaneous survivors. The admission NEW-CART (right panel of Figure 2.2) also has three decision rules and consists of six total nodes. Node 6 patients are considered high risk for dead/transplant outcome and are predicted as such, whereas nodes 1, 3 and 5 are predicted as survivors.

Tree models are visual in nature, and an example is one of the easiest ways to grasp how CARTs are developed and used in practice. Suppose a patient presents at hospital admission with the following characteristics: creatinine 3.2 mg/dL, INR 2.5, coma grade III, model for end stage liver disease (MELD) 22, lactate 5.2 mmol/L, and the patient was on a ventilator. First, the admission KCC-CART prognosis model will be used (Figure 2.2). At the start, the creatinine of 3.2 mg/dL is greater than 1.5, so we



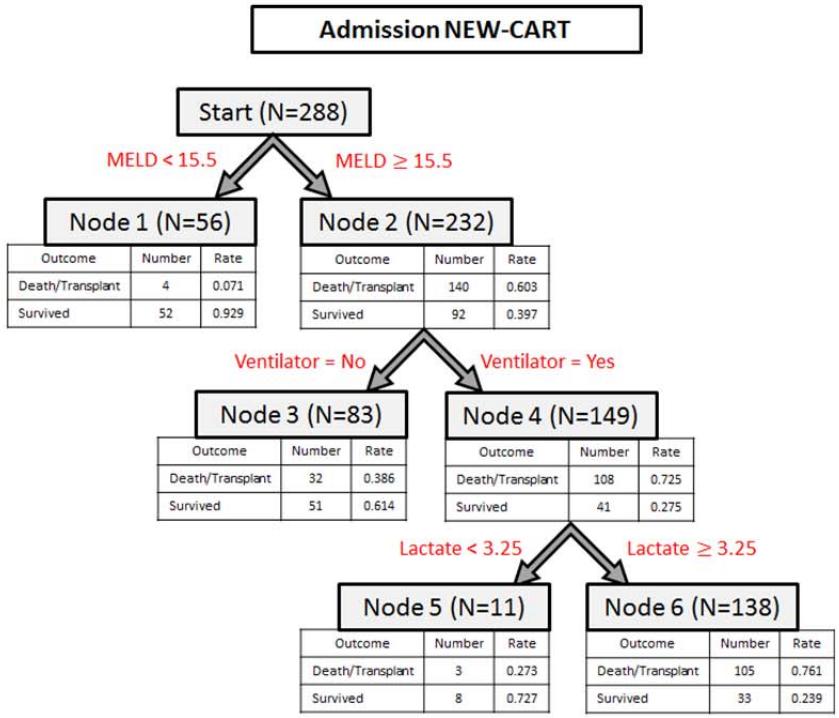


Figure 2.2: Admission CART models for predicting survival or death/transplant.

proceed to Node 2. Next, INR of 2.5 is greater than 1.75, so we proceed to Node 4. Finally, coma grade is III, so we move to terminal Node 6, which estimates that the risk of death/transplant at 21 days is high (probability of death/transplant is 0.736). Using the admission NEW-CART, we move to Node 2, then to Node 4 and reach Node 6, where the probability of death/transplant is 0.761.

Several decision tree algorithms other than CART are described within the literature. Loh discusses benefits and shortcomings of various decision tree algorithms: C4.5, CART, CHAID, CRUISE, GUIDE, and QUEST [36]. C4.5 and CART use an exhaustive search on every variable to determine variable splits and stops when criteria are met. CHAID and GUIDE differ from the other algorithms in that they group ordered variables then choose splitting points based on these groups, which aims to correct the bias to select continuous variables which other tree models suffer. CRUISE, GUIDE and QUEST use a two-step approach in which the variable most strongly associated with the outcome is selected then the cut-point is determined using an exhaustive search. While these alternatives aim to address some deficiencies within the CART framework, we focus on extending traditional decision tree (CART) and forest methodologies and leave exploration of other algorithms as future work.

2.1.2 Random Forest for Non-Clustered Data

Though CART models offer an alternative classification procedure to standard regression methods, there are some limitations which should be discussed. Firstly, CART models can create models which are complex, and users must decide how to prune models, which may introduce bias. Also, CARTs are weak learners, meaning that they may have more variability compared to more complex statistical algorithms [6]. For

example, small changes in the training dataset or predictor variables included may result in different CART models. Ensemble methods use many simple models (e.g. CARTs) which are slightly different and combine results from each model to produce predictions of outcome [39, 40]. Aggregating across many models generally results in more consistent predictions of outcome, although the price for model stability is diminishing simplicity and interpretability. While CART models may be beneficial compared to ensemble methods if interpretation of predictor variables is the main aim of a study, ensembles often offer higher prediction accuracy, particularly for complex datasets. Because CART models are sensitive to small changes within training data and may offer poor predictive ability for some complex dataset, ensemble methods may be preferable, especially if prediction accuracy is the main goal of the study.

Simple models, such as CART, may perform poorly for complex datasets, particularly when the number of predictor variables within a dataset is greater than the number of observations. There are various types of ensemble methods, which differ by how each simple model is developed (e.g. splitting training and testing data and subsets of predictor variables to be used) and how results from the simple models are combined to create overall predictions. Small differences in the simple models can lead to more accurate predictions when many models are developed and results are aggregated [39]. Rokach presents a framework for describing ensemble methods based on inducer (method for sampling observations), combiner (method for combining each of the classifiers), diversity (classifiers are as different as possible while maintaining accuracy of the training dataset), size (number of classifiers) and members' dependency (correlation structure of the classifiers) [40].

To overcome some of the limitations of CART models (e.g. poor predictive accuracy for complex datasets and high sensitivity to changes in the training dataset), an ensemble method called random forest (RF) can be employed. RF may be used to construct prediction models for continuous, categorical, ordinal, and survival outcomes. First introduced by Breiman in 2001 [4], the procedure offers an alternative modeling framework which offers many benefits compared to traditional, parametric models. RF is used in a wide variety of fields. The RF method is a standard statistical tool in the field of genetics and is often used in applications ranging from ecology to business administration [5, 41-51]. RF can be implemented on many standard computing platforms, including the commonly used R package *randomForest* [52]. RFs are an advantageous ensemble method for datasets in which the number of subjects is much smaller than the number of predictor variables, or if analyzing the relationship between predictor variables and outcome is of interest [40].

The RF procedure iteratively develops CARTs to model an outcome of interest. RF is a machine learning algorithm, which builds CART models in multiple steps (Figure 2.3). First, the dataset is randomly split into two groups using bootstrap sampling: an in-bag (training) set and an out-of-bag (validation) set. The in-bag dataset, approximately two-thirds of the entire dataset, is used to grow a CART within the forest. A CART is developed using a subset of predictor variables randomly selected at each node (splitting branches of the tree). When the tree is fully grown (i.e., when stopping criteria have been met), the out-of-bag dataset, consisting of the remaining data (approximately one-third), is run down all the trees in the forest. Each CART votes for what it predicts for the classification of all observations in the out-of-bag dataset (meaning the observations not

used to construct the tree), and the outcome group with the most votes is the prediction for the model. This process is repeated until the specified number of trees has been created. The out-of-bag error rate should be plotted against the number of trees within the forest to ensure that a sufficient number of trees have been grown, which is indicated if the error rate converges to a certain value.

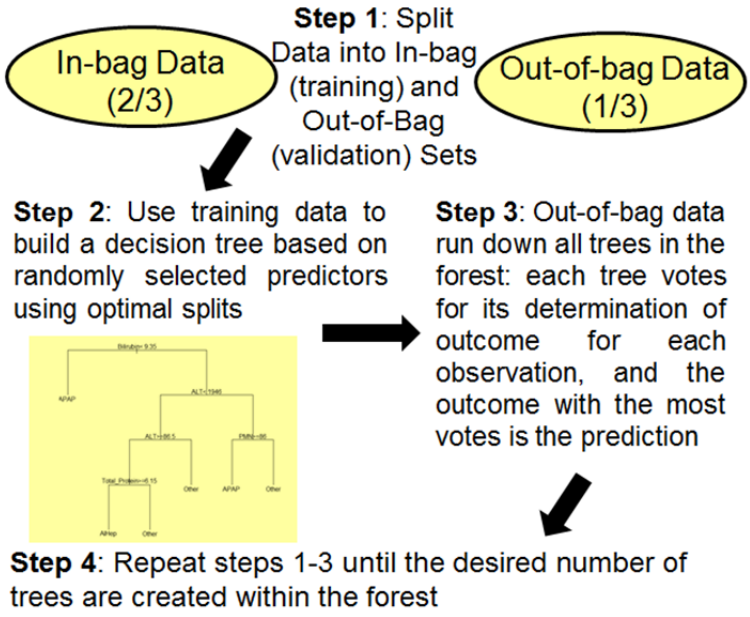


Figure 2.3: The RF algorithm

After a RF model is developed to predict a categorical outcome, there are several methods for evaluating its performance [47]. The first statistic which is typically assessed is the out-of-bag error rate, an unbiased measure of prediction error for the model. Additionally, a confusion matrix with class error rates may be assessed to determine how observations are being incorrectly classified. An example confusion matrix is provided within Table 2.1, in which a RF is used to predict etiology of ALF patients, an outcome with six groups: autoimmune hepatitis (AIHep), acetaminophen (APAP), drug induced liver injury (DILI), hepatitis B (HepB), indeterminate (Indeter), and other [47]. Columns

of the table represent the outcome class that the model predicts for the subjects, and rows represent the actual outcome class of the subject. Thus, the diagonal of the table, shaded in Table 2.1, represents the numbers of subjects correctly classified by the model. Using this table, information can be gained about which categories are most likely to be incorrectly classified. For example, there are four subjects whose etiology is autoimmune hepatitis, but the model predicts that they are in the acetaminophen group. Similar interpretations can be made for the rest of the cells in the table. The last column in Table 2 specifies the error rate of the model, broken down by outcome categories. For this model, the rate of misclassification is very low for the acetaminophen etiology group, is moderate for the autoimmune hepatitis and other groups, and is fairly high for the remaining groups.

Table 2.1: Confusion matrix for etiology of acute liver failure patients

		Predicted Class						
		AIHep	**APAP**	**DILI**	**HepB**	**Indeter**	**Other**	**Class Error**
Actual Class	**AIHep**	63	4	35	5	22	7	0.54
	APAP	1	873	3	2	3	32	0.04
	DILI	29	30	60	17	48	36	0.73
	HepB	7	30	12	48	29	16	0.66
	Indeter	15	49	34	14	87	46	0.64
	Other	10	90	28	12	17	164	0.49

A visual method for presenting the same results can be produced using a measure called the margin. The margin of a data point is the proportion of votes for the correct class minus the maximum proportion of votes for the remaining classes. Thus, positive margins correspond to correct prediction and negative margins mean incorrect prediction [52]. Moreover, the margin of an observation serves as a measure of confidence in correct classification. The values are contained in the interval [-1, 1]; margins closer to 1 indicate

higher confidence in accurate classification and margins closer to -1 indicate lower confidence in accurate classification [53]. A histogram of margins for all observations may be produced to assess model accuracy overall, and boxplots of margins by outcome groups provide a visual method for comparing model accuracy by outcome categories [47].

There are many reasons that RF is a beneficial statistical tool for classification. Primarily, it does not have the same limitations that inhibit many traditional statistical procedures (e.g. those described in Section 2.1). For example, the RF procedure can handle the high dimensional data, when the number of predictor variables is much larger than the sample size. Additionally, RF can determine which variables are important in the model and can capture nonlinear patterns between predictors and the outcome of interest without *a priori* specification [5]. Another benefit of RF is that outliers or sub-clusters of data may be identified using proximities [50]. Proximities are stored within a square matrix with dimension equal to the number of observations within the dataset. For two different observations, the proximity is increased by 1 if the observations fall within the same terminal node of a tree. This is repeated for all trees and all combinations of observations, and the proximity matrix is normalized by dividing by the number of trees within the forest [4]. RF differs greatly from its traditional statistical counterparts and offers an alternate method which often has lower prediction error rates than many traditional models. Verikas et al. illustrate the higher accuracy of RF compared to logistic regression, linear and quadratic discriminant analysis, k-nearest neighbor, support vector machines and naïve Bayes [54]. For these reasons, RF is an attractive solution to the problem of classification.

Even though there are many positive aspects to the RF procedure, it does have challenges in implementation and interpretation. The algorithm is quite different from most statistical classification procedures, and many of the statistics commonly reported for traditional regression modeling approaches are not used. RF does not calculate p-values, confidence intervals, or test statistics in the traditional sense. Moreover, it does not provide users with a closed form of a model because of the complexity of the algorithm, which is partially why RF is called a 'black box' method. Although the procedure lacks many tools that are conventionally used to evaluate models, it is possible to extract similar information from the output.

Before presentation of variable importance measures and imputation of missing data within the RF framework within the next few sections, a detour will be taken to briefly discuss conditional RF. Several researchers have developed alternative RF methodologies which aim to address some deficiencies of the method. A criticism of standard CART and RF is that the mechanism for selecting variable splits within trees is biased since it tends to use continuous variables and categorical variables with many groups more often than other types of variables [55]. This issue will be discussed in detail in the following section. The conditional RF framework offers an alternative framework in which significance tests are used to determine split variables and split points [55, 56]. Assessing variables with statistical tests reduces the bias for variables of certain forms (e.g. continuous variables or categorical variables with many groups) compared to standard RF. However, conditional RFs often have higher computation times compared to traditional RFs and have not been rigorously examined within the literature to assess its robustness for clustered datasets. For these reasons, we focus this dissertation on

traditional RF, and note that future research into conditional RF for clustered outcomes may be worthwhile.

2.1.3 Missing Data in Random Forest

Deciding how to handle missing data is an important step in the model-building process. There are many methods for imputing (filling in) missing values for parametric models, such as mean imputation, least squares estimation, iterative procedures (such as expectation-maximization) and multiple imputation [57]. While there are some available methods which may be used to impute missing predictor data when modeling clustered binary outcomes (e.g. multiple imputation using chained equations [58]), we focus on decision tree and RF methodology for imputing missing data. CART and RF frameworks have their own techniques for handling missing data.

There are many different methods for handling missing data within decision trees. The default method for handling missing data within the CART setting involves finding a surrogate split, in which a different variable is substituted within a node for observations with missing values of the predictor selected for the node. Surrogate variables are selected using the same method as variable selection for each node, which minimize the node impurity for non-missing observations. Ding and Smirnoff provide an overview of missing data techniques for classification trees applied to binary data [59]. The authors compare probabilistic split (deterministic rules for the probability that an observation follows right or left daughter node when the predictor at the node is missing), complete case analysis, grand mode or mean imputation, separate class (including "missing" as a class category for the split), surrogate split, and complete variable method for handling missing data. The results from this study indicate that the best way to handle missing data

depends mainly on two factors: the missingness mechanism (missing at random, missing completely at random, or missing not at random) and whether there are missing data in the test dataset. If there are missing predictor values in the test set, then including a separate class for missing values is optimal; otherwise, the probabilistic split method is best. However, if a test set is not available at the time of data analysis, it is unclear which method for handling missing data is optimal. Twala et al. determine that including a missing category for both continuous and categorical variables is an effective way to handle missing data for decision trees [60]. Compared to more computationally intensive proposed approaches, the technique of adding a category for missing variables performs similarly for 21 datasets in the Repository of Machine Learning Databases where missingness is induced using various mechanisms.

Methods for missing data imputation within the RF framework are slightly more complex compared to those of decision trees. However, the default procedure for RF imputation may be easily computed using the *rfImpute* function within the R package *randomForest* [52]. The algorithm to impute missing input values is as follows. First, all missing values are replaced using a rough fix, with the median of non-missing continuous variables and the most frequent class of non-missing categorical variables. Next, a RF is developed and new imputed values are estimated using the non-missing values of variables weighted by the proximities matrix. The process—create RF then impute missing values—is repeated for several iterations to obtain final imputed values. Typically four to six iterations are sufficient, and the default for *rfImpute* within the R package *randomForest* is five iterations.

Several studies demonstrate similar or better performance of RF imputation compared to other missing data imputation methods. Reiger et al. use simulations to show that using a surrogate splitting method to handle missing data is equivalent to K-nearest neighbor imputation within the conditional inference forest framework [61]. Schwarz et al. provide an imputation technique using RF for genome-wide association study data which contained missing data after combining multiple datasets [62]. Pantanowitz and Marwala demonstrate that RF imputation is more accurate and computationally efficient compared to imputation methods used in neural networks [63].

While the original imputation method for RF is beneficial compared to many other imputation techniques, some researchers have developed alternatives. For example, Hapfelmeier et al. compare multiple imputation by chained equations with surrogate splits in decision trees and forest applications, concluding that imputation may be worse than surrogate splits if there is a small percentage of missing [64]. Multiple imputation by chained equations is computationally intensive and produces ambiguous results for the twelve datasets analyzed. Surrogate splits are negligibly worse compared to this imputation method in this study. However, the only missingness mechanism assessed was missing completely at random, an assumption which may be violated for some applications. Stekhoven and Buhlmann develop a new RF imputation method which does not require the outcome to be non-missing, as in the case of the original RF imputation method [65]. *MissForest* is available as an R package, and authors conclude that the method can unbiasedly impute missing data up to 30%, with no distributional assumptions or tuning parameters.

Imputation of missing data affects all aspects of RFs, including variable importance and selection of variables. Hapfelmeier et al. develop a new variable importance measure within the conditional forest framework which is less biased than permutation importance when predictors have missing data [66]. Several missing data schemes are used in a simulation study to illustrate the unbiasedness of the new variable importance measure. Another study by Hapfelmeier and Ulm discusses a variable selection method which may be used when predictors have missing data up to 30% within conditional forest modeling [67]. While a limited number of variables are used in a simulation study, the method is more accurate than selection procedures based on performance, such as *varSelRF* by Diaz-Uriarte and De Andres [42]. The issue of missing data arises in many real-world data analyses, and careful consideration of how imputation of missing values impact modeling is essential.

2.1.4 Variable Importance in Random Forest

Aside from imputation methods, a major benefit of RF compared to some other machine learning algorithms such as neural networks and SVMs is the capability to assess variables within the model through variable importance measures. Two measures are typically considered: the mean decrease in accuracy and the mean decrease Gini [5, 54]. The latter is based on the number of splits within the decision trees for each predictor and is criticized for its bias for continuous variables. Because continuous variables have many more options for where splits can occur within each decision tree in the RF, the mean decrease Gini tends to give higher importance to these variables, as opposed to ordinal or categorical variables, which have a limited number of places for splits to occur. Formally, the Gini importance is defined in the following manner [4, 68]. Suppose there

are i = 1,2,…, n total observations, j = 1, 2,…, p predictor variables, and k = 1, 2,…, *ntree* trees included in an RF model. Let n_m represent the total observations within node m and n_{m_z} represent the number of observations within node m in outcome class z for outcomes $z=1,\ldots,Z$. The proportion of observations in node m that are contained in outcome class z is defined by

$$\hat{p}_{mz} = \frac{n_{m_z}}{n_m}.$$

The Gini index is given by:

$$GI = \sum_{z=1}^{Z} \hat{p}_{mz}(1 - \hat{p}_{mz}).$$

Gini importance is defined in terms of importance of variable j to tree k for all outcomes z:

$$GINI_{jkz} = \left(GI_{parent} - GI_{left\ daughter\ node} + GI_{right\ daughter\ node}\right) zp_{kz}\ .$$

Summing this over nodes containing variable j within tree k yields:

$$GINI_{jk} = \sum_{z_j \in Tree\ k}^{Z} GINI_{jkz}.$$

Finally, averaging over the trees within the forest gives the formula for the mean decrease in Gini importance measure for variable j is:

$$GINI_j = \frac{1}{ntree} \sum_{k=1}^{ntree} GINI_{jk}.$$

Another importance measure is the mean decrease in accuracy (also called permutation accuracy or permutation importance), which is the difference between the out-of-bag error rate from a randomly permuted dataset and the out-of-bag error rate of the original dataset, expressed as an average percent over all trees in the forest. A proper definition for permutation importance is described in the following manner [4, 68]. Let

s_{ijk} be the number of trees which split on variable *j* and misclassify observation *i*, r_{ijk} be the number of trees which do not split on variable *j* and misclassify observation *i*, $\tilde{s}_{ijk}$ be the number of trees which split on variable *j* and misclassify observation *i* when variable *j* is permuted, and $\tilde{r}_{ijk}$ be the number of trees which do not split on variable *j* and misclassify observation *i* when variable *j* is permuted. The permutation index is defined in terms of variable *j* to tree *k* for all observations *i*:

$$PERM_{jki} = \left(\tilde{s}_{ijk} + \tilde{r}_{ijk}\right) - (s_{ijk} + r_{ijk}).$$

Averaging over all observations and trees, the permutation importance is defined by:

$$PERM\ IMP_j = \frac{1}{n*ntree} \sum_{i=1}^{n} \sum_{k=1}^{ntree} PERM_{jki}.$$

For both mean decrease in Gini and permutation accuracy, high values represent important variables, and low values represent unimportant variables in the RF framework. A general discussion of theoretical considerations for variable importance measures for machine learning methods is presented by Van der Laan, in which statistical tests and confidence intervals for the importance measures are derived [69].

Several alternative variable importance measures are suggested within the literature. Sandri and Zuccolotto discuss four methods for evaluating variable importance: permutation importance, importance based on the maximal margin function, importance based on the difference of the number of lowered and raised margins, and the Gini importance [70]. A variable selection procedure incorporating these importance measures within a principal components analysis framework is presented, though no simulation study is provided to examine the effectiveness of the method [70]. Wang et al. propose a measure based on the maximal conditional chi-square p-value [71]. Traditional

importance measures average over effects of other predictors, so this new importance measure aims to accurately capture importance of variables, even ones involved in complex interactions. The method is successful for high dimensional data, and is demonstrated with a real microarray study. Zhou et al. define a new variable importance measure using the proximity matrix, which provides similar performance to the permutation importance for eight microarray datasets [51]. The new importance measure offers no distinguishable benefit compared to permutation importance.

A limitation of several variable importance measures, including Gini accuracy and permutation accuracy, is the bias towards higher values for continuous variables and variables with many categories. Strobl et al. suggest use of conditional RFs or traditional RF without replacement in the bootstrap samples to reduce bias of variable importance [55]. Simulated datasets in this study are low dimensional (do not contain many variables), so results may not generalize to other datasets. Nicodemus et al. extends results of Strobl, again using a low dimensional simulated dataset with few predictors [56]. Variable importance measures are shown to depend on forest size and the amount of correlation between predictor variables. Variable importance measures are presented in unscaled and scaled (divided by standard errors) forms. In another study, Nicodemus et al. further demonstrate the bias of Gini importance relative to permutation importance using a simulation study, also noting that permutation importance is less sensitive to small changes within the data [72]. Authors additionally claim that there is no benefit to scaling importance measures (dividing by the empirical standard error); however, the simulations in this study conducted used a balanced outcome variable, so results may differ if the proportion of observations in each outcome class is different.

It should be noted that variable importance is not synonymous with statistical significance. Variables may be important within the RF but may not be statistically or clinically significant. However, the measures provide a means of identifying variables which are most important in predicting the outcome, as well as a statistic for comparing the relative importance of many input variables. Altmann et al. incorporate a statistical test for a new variable importance measure, which is based on the p-value of repeated permutations of the outcome vector [73]. While traditional importance measures used in the RF setting will divide importance between groups of correlated predictors, the method proposed by authors provides p-values for each variable which are significant for important variables regardless of correlations with other variables. However, simulations in this study only included categorical predictors, which may limit the generalizability of results. Additionally, computing time to calculate the novel importance measure of Altmann et al. are much larger compared to other variable importance measures.

2.1.5 Variable Selection Procedures in Random Forest

Though a main benefit of RF is that it may be used for high dimensional data or datasets with many predictors, inclusion of a large number of predictors is not ideal for clinical applications where time-efficient predictions are necessary. Variable selection is critical when developing classification models since it can potentially reduce noise and improve computational efficiency. There is an essential link between variable importance measures and variable selection. The various importance measures reviewed in the previous section could each be used to develop variable selection procedures.

A standard RF variable selection procedure is developed by Diaz-Uriarte and De Andres, which uses the ranks of the permutation importance for each variable to eliminate the

least important variables [42]. Compared to linear discriminant analysis and support vector machines, the accuracy of the RF with the variable selection procedure is similar. A limitation of the method is that variables selected are not unique, meaning that the procedure may select different groups of variables each time it is employed which all lead to similar overall prediction error rates. However, this method remains a useful tool to ease the burden of data collection, which is particularly important in medical prediction modeling. A major strength of this procedure is that the number of variables to be included in the final model does not need to be determined *a priori*; the method will eliminate unimportant variables successively until a solution based on user-inputted parameters is obtained. Authors produce an R package called *varSelRF* which implements the variable selection tool [42].

Many studies have investigated the use of RF to select important variables in microarray datasets. Tang et al. compare RF and logistic regression for identifying genes and haplotypes which are predictive of rheumatoid arthritis, and claim that more research is needed to investigate the usefulness of RF [49]. Yang and Gu analyze variable importance using RF and Bayesian networks for genome-wide association study data, and conclude that RF predicts the categorical outcome with higher accuracy compared to Bayesian networks [74]. However, the authors highlight a major limitation of RF, in contrast to the Bayesian networks: the possible inability of RF to identify underlying causal relationships. In other words, RFs offer higher prediction accuracy but may fail to identify known risk factors for the outcome. This is a limitation of several methods and is based on the study design. Rodenburg et al. use RF permutation importance to identify variables which are then inputted into self-organizing maps for assessment of genes [75].

While this method correctly identifies biological processes of interest according to the authors, it is unclear if this combination of RF and other statistical models would result in better prediction or identification of variable relationships in settings other than microarrays.

There are many different methods for variable selection within the RF framework, and thus far, there is no consensus on the optimal method for identifying the most important variables. Current literature suggests that a commonly used method is that of Diaz-Uriarte and De Andres, but further exploration of how the method performs for various datasets is warranted.

2.1.6 Summary of Classification Methods for Non-Clustered Data

In this section, parametric, nonparametric and machine learning methods for classification are presented, with emphasis on tree and forest modeling. We focus on CART and RF since they have high generalizability and accuracy compared to other classification techniques for a variety of datasets comprised of both continuous and categorical features. Characteristics of RF, including variable importance measures, variable selection procedures, and imputation mechanisms, are discussed in detail for data which contain predictors collected at a single time point.

2.2 Classification Methods for Clustered and Longitudinal Data

Often in research settings, variables (both outcomes and predictors) are collected serially over time, creating a repeated measures or longitudinal dataset. The additional data collected for observations of the same patient over time results in a correlation structure since outcomes are dependent. A more general term for this is called a clustering variable, in which outcomes are correlated within a group structure. For

example, patients treated from the same doctor may be considered a clustering variable since there may be dependencies between patients treated by the same person. Clustered outcomes should be modeled properly to accurately model the variance structure. In this section, methods for developing classification models for longitudinal and clustered datasets are discussed, with emphasis on tree and forest frameworks.

The most commonly used method for predicting categorical clustered and longitudinal outcomes is generalized linear mixed modeling (GLMM). A mixed model is developed, often using fixed effects for covariates collected at a single time point and random effects to account for the longitudinal measures within a subject or within a clustering variable. Users must select a mean model and covariance model based on measures such as the Akaike information criterion or Bayesian information criterion, along with an appropriate link function, depending on the categorical form of the outcome (e.g. binary, multinomial, ordinal, etc.) [76, 77]. Though GLMMs are frequently used for a variety of studies, a limitation of the framework is the user must specify interactions and if there is a nonlinear relationship between predictors and outcome through the link function, which is not always straightforward. Another drawback of GLMMs is the lack of an automated procedure for selecting the covariance structure, which often leads to subjective decisions about the optimal form. Additionally, GLMMs may have convergence issues or may be difficult to implement for high dimensional datasets (e.g. with more than 30 predictor variables).

Other methods for modeling clustered outcomes include linear and quadratic discriminant analysis, support vector machines, and neural networks; however, we refrain from further discussion of these methods due to limitations presented in Section 2.1. The

remainder of this section focuses on advances in decision tree and RF procedures for classification of clustered and longitudinal data.

2.2.1 Decision Trees for Clustered and Longitudinal Outcomes

There are several methods for developing decision trees for continuous and ordinal outcomes with clustered outcome variables. One of the first researchers to explore this type of methodology is Segal (1992) [14]. Two splitting functions are described: one which focuses on the mean structure with the covariance being treated as a nuisance and one in which the covariance structure is the primary interest. The latter uses a likelihood ratio test for the equality of covariance matrices to determine a split, assuming multivariate normality of predictor variables. The classic missing data mechanism of using surrogate splits is extended for the framework of repeated measurements. Time-varying covariates are used in the method: a regression model containing time as a covariate is compiled for each variable, and slopes and intercepts are used by the regression tree algorithm. However, the methodology may only be used for continuous or ordinal outcomes. Zhang and Ye provide a method for producing decision trees for longitudinal data where the outcome is ordinal [78]. Each response is converted into binary indicators and trees are developed using these; thus, interpretability is challenging and computing time is large due to the difficulty of the splitting function used at each node.

Various alternative methods for choosing variable splits for decision trees with clustered outcomes are suggested within the literature. Abdolell et al. provide another mechanism for developing trees for continuous clustered outcomes [7]. Likelihood ratio statistics from mixed models are used to choose variable splits. This work augments that

of Segal [14] since variable importance is calculated using a permutation test, along with bootstrap confidence intervals for variable cut points. A SAS macro is provided by the authors which implements the method. Another method for multivariate decision trees is developed by De'Ath which may be implemented for continuous outcome data using the R package *mvpart* [8]. This method may also be used for clustering since the splitting mechanism separates groups which behave similarly. Yu and Lambert propose two alternative methods for multivariate decision trees [16]. The first method involves fitting spline curves for individuals with longitudinal data and uses the estimated coefficients as the continuous outcome in regression tree models. The second reduces the dimensionality of the data using principal components and develops a regression tree using loadings from the first few components. The main issue with these approaches is the lack of direct interpretability. Additionally, the methods are not proved to be accurate compared to existing methods, and variable selection bias was not discussed.

Some researchers are investigating the use of generalized estimating equations (GEEs) to find optimal variable splits within decision trees for clustered datasets. For example, Keon Lee uses GEEs to find the best variable and optimal splitting point based on residuals for each node [11]. This method allows for any type of clustered outcome data, including binary, multinomial, and count data. However, the GEE tree technique does not allow for predictions of new observations, and it can only be implemented when outcomes are of the same type. Dine et al. overcome the limitation of Keon Lee's GEE model by developing a tree algorithm which may be used for clustered responses of differing types [9]. A likelihood-based approach is used as a variable splitting function. A main objective is to develop a single tree which could be used for predicting multiple

outcomes to be used as an exploratory tool. Like the Keon Lee model [11], the multivariate trees for mixed outcomes cannot be used for prediction.

The methods for clustered decision trees discussed in the previous paragraphs all use the CART algorithm. Several researchers investigate alternative decision tree algorithms which aim to reduce the bias in variable selection inherent in CART models. For instance, Lee and Shih discuss ways to implement multivariate decision trees in an unbiased manner, in which a conditional independence test is used to select variables at each node [79]. This variation on traditional CART is implemented to minimize the selection bias of continuous variables, while in the setting of clustered outcomes. It is an extension of decision tree algorithms called QUEST and CRUISE [37]. Hsiao and Shih implement a chi-squared test for conditional independence on the residual signs for grouped covariate values [80]. However, this splitting function may be inadequate if covariate groups effects are not all associated in the same direction as the outcome of interest (i.e. if the tails of a distribution are associated with one outcome and the middle of the distribution is associated with another). Eo and Cho [81] suggest using residuals from mixed effects models to determine variable splits, which is the approach in GUIDE trees by Loh and Zheng [13]. The method allows for interpretation of variable trends over time, can handle datasets which have imbalanced outcomes, and reduces computation time compared to methods which use exhaustive searches for all variables to determine splits.

Combining the unbiased variable selection of an alternative decision tree algorithm called GUIDE [13] and splitting on residuals as in the method proposed by Eo and Cho [81], work by Loh and Zheng [13] demonstrates that multivariate GUIDE

achieves similar prediction accuracy compared to *mvpart* [8] and GEE trees [11]. However, multivariate GUIDE does not suffer a selection bias toward continuous variables which is inherent to *mvpart*. A limitation of this method is that users must define intervals which divide continuous variables into several categories which will be tested to select the splitting variable at a node, and there is no automated way to determine this parameter. Software to implement GUIDE and multivariate GUIDE is available on the author's website (http://www.stat.wisc.edu/~loh/guide.html).

A promising novel method proposed by Sela for continuous outcome prediction is called random effects expectation-maximization (RE-EM) trees, in which an algorithm similar to expectation-maximization is used iteratively to obtain a prediction model [15]. The method can be implemented using any type of tree method (e.g. CART, GUIDE, CRUISE, etc.) in conjunction with linear mixed effects modeling to account for clustering within the data. RE-EM trees can predict future observations for subjects within the original dataset or new subjects. Diagnostic plots must be assessed to determine if assumptions about the random effects model are reasonable. Authors use surrogate splits for missing data, which may not be optimal since this method does not consider data from other time points which may be non-missing. Variable importance and variable selection methods are not presented for the method, and there is an issue with interpretability since the tree and the mixed model are compiled separately. This means that it is not possible to assess time-varying covariates.

Loh summarizes tree based methods for classification and regression outcomes, and details methods developed for longitudinal and clustered data [37]. Several problems within decision tree research are discussed as avenues for future work, namely: how to

handle missing data, how to incorporate time-varying covariates for longitudinal data, allowing for splits on linear or nonlinear combinations of variables for nodes within trees, and maximizing computational efficiency in the age of big data. Ciampi [82] provides a discussion based on the literature review from Loh [37], suggesting various other future research topics within the decision tree framework, such as developing Bayesian tree methods and allowing for "soft" nodes in which a hierarchy of variables is determined by experts which is implemented in the decision tree.

2.2.2 Random Forests for Clustered and Longitudinal Data

Though there are myriad methodologies for decision tree development for clustered datasets for continuous and ordinal outcomes, few studies have explored implementation of ensembles of trees in this setting. In this section, methods for developing RFs for longitudinal and clustered datasets for continuous outcomes are presented.

One approach to building an ensemble framework for clustered datasets is simply to aggregate predictions from multiple decision trees using methodology described in the previous section. Segal and Xiao [83] construct an ensemble of trees described in Segal's work [14] using a continuous outcome. A tutorial is provided for analysis of a continuous outcome, including analysis of accuracy, important variables identified by the RF, and proximities of variables. However, the sampling scheme for each tree is not updated to reflect the clustered nature of observations, nor is the computation of other RF features (e.g. variable importance measures, proximities, margins, etc.). While this study provides a simple way to aggregate results from many decision trees, consideration of methods which may more sufficiently use the clustered data structure is recommendable.

Karpievitch et al. develop a method called RF++, which performs subject-level bootstrapping then aggregates predictions from the same subject [84]. This is motivated by a dataset which had a small sample size relative to the number of predictors, where the majority of the variables are not associated with the continuous outcome of interest. The authors demonstrate through simulation that for data of this form, typical RF performed as well as the novel RF method which accounted for the clustering among observations. However, the results suggest that the typical RF underestimates error rates, so authors recommend that a test dataset be used to obtain an unbiased estimate for error.

A promising RF methodology for longitudinal data within the current literature is presented by Larocque [12] and Hajjem et al. [10, 17]. Larocque introduces the following methodologies: mixed effects regression tree (MERT), generalized mixed effects regression tree (GMERT), mixed effects random forest (MERF), and generalized mixed effects random forest (GMERF) [12]. The main idea of these methods is to incorporate mixed effects within the tree framework to account for the correlation structure within the data, using an expectation-maximization algorithm described by Wu and Zhang [18] to develop the prediction model. The algorithm functions as follows. First a model is fit for the fixed portion of the response variable. Next bootstrap samples from the training dataset are used to build the forest of trees and outcomes are updated based on random effects, the variance for the error term, and the variance for the random effects. These steps are iterated until the model converges. Twelve simulated scenarios are investigated using continuous outcomes, and the new methodology performs as well or better than traditional trees and forests. Though GMERT and GMERF methodology are described,

no simulations or data applications are presented, and there is no available software to implement these methods.

Hajjem et al. [10] extends the work of Laroque [12] by allowing for specification of the covariance structure between observations within a cluster for continuous outcome prediction. The method is also beneficial due to its ability to handle different numbers of observations within clusters (unbalanced clusters) and its flexibility for various types of covariates (observation- or cluster-level). Simulation results indicate that MERT performs well even if the random effects within the model are incorrectly specified. MERT offers a reduction in predicted mean squared error and tree size, making it both more accurate and easy to interpret compared to traditional tree models. Another study by Hajjem et al. further demonstrates the benefits of using MERT and MERF over standard RF, regression trees, linear mixed effects models, and linear models through simulations, demonstrating that traditional RF is not appropriate for clustered data [17]. MERF performs best when datasets had lower amounts of random noise, and the amount of correlation between predictor variables did not impact the relative improvement of MERF over other alternative models.

Thus far, the MERF methodology largely focuses on evaluation of accuracy for modeling continuous outcomes with various types of datasets (e.g. containing outliers, unbalanced categorical variables, and different amounts of correlation within observations). Authors claim that results would be similar for categorical outcomes [10, 17], but no formal simulation or data applications have been published investigating the use of GMERT or GMERF for outcomes which are not continuous.

2.2.3 Summary of Classification Methods for Clustered and Longitudinal Data

Several methods for modeling continuous outcomes with clustered and longitudinal data are presented, with emphasis on decision tree and RF frameworks. A multitude of longitudinal decision trees using various algorithms (CART and others) are presented, but none are developed for modeling categorical outcomes. Akin to the setting of data collected at a single time point, ensembles of trees are sometimes preferable since they may offer better performance for complex datasets. The main method of employing RF for clustered data, called MERF and GMERF, is introduced by Laroque and evaluated thoroughly for continuous outcomes by Hajjem et al. However, further work is warranted to develop and assess the decision tree and RF methodology for clustered categorical outcomes.

3 METHODS FOR SPECIFIC AIMS

3.1 Specific Aim 1: **To develop a CART method for clustered and longitudinal binary outcomes using an iterative procedure to combine CART and mixed effect models**

3.1.1 Introduction

Clustered binary outcomes are frequently encountered in clinical research. Correlation within datasets may result from variables representing subject clusters, such as medical centers or family groups. Another type of clustered outcome results from longitudinal or repeated measures studies, where each patient represents a cluster. For example, a longitudinal study may collect repeated measurements of outcomes to evaluate disease prognosis (e.g. poor versus good outcome), diagnosis or disease relapse (e.g. disease versus disease-free), or other endpoints (e.g. re-admitted or not re-admitted to the hospital). Outcomes collected on the same patient at multiple time points are almost always dependent on one another. This within-subject correlation should be considered because failing to account for correlation results in a loss of estimation efficiency.

Generalized linear mixed models (GLMMs) are typically employed for modeling clustered and longitudinal outcomes, but suffer limitations for some datasets. For example, GLMMs cannot be implemented in the setting of high dimensional data, when there are more predictor variables than observations. Modern datasets, particularly in the setting of genetic medical research, often have thousands of predictors and only several hundred observations, thus posing a significant challenge for traditional GLMMs for

modeling clustered outcomes. Additionally, interactions between predictor variables should be selected *a priori* to be included in GLMM modeling. However, knowledge about interactions between predictors is often lacking in practice, especially in complex clinical settings considering many personal, familial, and environmental factors. The GLMM framework also requires users to specify if there is a nonlinear relationship between predictors and outcome through the link function. Though specification of nonlinear relationships and interaction terms is not impossible, it often presents a challenge in the GLMM framework since there is not a universal method for making these decisions about modeling.

In this paper, we propose an alternative method that provides greater flexibility for complex datasets called Binary Mixed Model (BiMM) tree, which combines decision tree methodology with GLMM. Decision tree methodology can be implemented to develop prediction models which can be used in the setting of high dimensional data. Also, decision trees do not assume a linear relationship between predictor variables and outcome. Interactions between predictor variables are also naturally modeled within the decision tree framework without prior knowledge. Thus, decision trees offer a flexible framework for developing prediction models. In the BiMM tree method, we incorporate results from decision trees within mixed models to adjust for clustered and longitudinal outcomes. A Bayesian implementation of GLMM is used to avoid issues with convergence and quasi- or completely separated datasets with binary outcomes.

A specific motivating example dataset for the novel methodology in this paper is a longitudinal registry dataset of acute liver failure (ALF) patients (clinicaltrials.gov ID: NCT00518440). ALF is a rare and devastating condition characterized by rapid onset of

severe liver damage, encephalopathy (altered mental status) and coagulopathy (impaired blood clotting), with approximately 25% of patients requiring a liver transplant and approximately 30% of patients dying during the acute phase [85]. Complexities of the ALF registry data, including skewed distributions of predictors with many extreme values, nonlinear predictors of outcome, and a relatively high dimensional dataset, make it difficult to employ GLMMs for predicting outcomes.

The chapter is structured as follows. In Section 3.1.2, we present background information about decision tree modeling in general and tree models for longitudinal and clustered continuous outcomes. In Section 3.1.3, we introduce the BiMM tree method for predicting longitudinal and clustered binary outcomes and in Section 3.1.4 we describe the motivating ALF registry in detail. We compare the BiMM tree method performance to several other methods with a simulation study in Sections 3.1.5 and 3.1.6. Finally, in Section 3.1.7 we discuss implications of our study, limitations, and avenues for further research.

3.1.2 Background

A decision tree framework is utilized for the novel BiMM tree method because it offers several advantages compared to traditional models such as GLMMs. There are many different decision tree methods available, and we implement our BiMM tree method with the classification and regression tree (CART) framework, a commonly used methodology developed by Breiman [1]. CART does not require specification of non-linear relationships or interaction terms, and offers simple and intuitive interpretation of predictor variables. Moreover, CART provides an alternative method for developing prediction models when traditional models are not feasible (e.g. if the number of

predictor variables is greater than the number of observations). For these reasons, CART can sometimes better predict outcomes compared to other procedures such as discriminant analysis and logistic regression for data captured at a single time point [6].

In spite of this flexibility, few decision tree methods exist for modeling clustered categorical endpoints. The R package *party* can be used to implement CART models if two predictor variables are correlated, but it does not adjust for longitudinal and clustered measurements of the same outcome variable [19]. There are some techniques which circumvent the issue of adjusting for longitudinal and clustered outcomes, such as summarizing variables (e.g. using averages or most frequent categorical values) or using data from only a single time point (e.g. admission values); however, these methods have a marked loss of information since available data is summarized or partially used.

Several methods have been proposed to modify CART models for longitudinal and clustered continuous outcomes [7-16]. Hajjem [8] and Sela [15] develop similar methods for implementing CART models for longitudinal and clustered data with continuous outcomes. These methods incorporate mixed effects within the tree framework to account for the clustered structure within the data, using an algorithm analogous to expectation-maximization described by Wu and Zhang [18]. The main idea in the Sela RE-EM tree [15] and Hajjem mixed effects regression tree [8] algorithms is to dissociate the fixed and cluster-level components within the modeling framework. First, a CART with all predictors as fixed effects is fitted with the assumption that the random effects for the clusters are known. Next, a linear mixed model is fitted using the estimated fixed effects from the CART and the random cluster effects are estimated which account for correlation induced by clustered variables with the assumption that the fixed effects

are known. Finally, the continuous outcome is updated based on the linear mixed model using an additive effect in which the estimated random cluster effect is added to the original continuous outcome. The algorithm continues to iterate between CART (estimating CART assuming that mixed effects are known), linear mixed models (estimating mixed model assuming that fixed CART effects are known), and updating the outcome in a framework similar to the expectation-maximization algorithm [18]. The algorithms continue iteratively until convergence is satisfied, which is based on the change in the likelihood from the mixed model being less than a specified value.

While the framework for clustered CART modeling has been developed for continuous outcomes, adjusting the algorithm for clustered categorical outcomes is non-trivial. For continuous endpoints, the outcomes are updated based on random effects from the linear mixed model using an additive effect. For categorical outcomes, the optimal method for adjusting outcomes is unclear because a random effect cannot simply be added.

3.1.3 BiMM Tree Method

The BiMM tree method iterates between developing CART models using all predictors and then using information from the CART model within a Bayesian GLMM to adjust for the clustered structure of the outcome. Consistent with the continuous methods for clustered decision trees, we implement an algorithm similar to the expectation-maximization algorithm, in which the fixed (decision tree) effects are dissociated from the random (cluster-level) effects. While developing the CART model, it is assumed that the random effects are known, and while developing the Bayesian GLMM, it is assumed that the fixed components are known. The BiMM tree method may be considered as an extension of GLMMs where the fixed covariates are not assumed to

be linearly associated with the link function of the outcome and interactions do not need to be pre-specified. The traditional GLMM for binary outcomes has the form

$$\text{logit}(y_{it}) = \boldsymbol{X_{it}\beta} + Z_{it}b_{it},$$

where y_{it} is the binary outcome for cluster i = 1,…,M for longitudinal measurements t=1,…,T_i, logit() is the logistic link function, $\boldsymbol{X_{it}}$ is a matrix of fixed covariates for cluster i for longitudinal measurement t, $\boldsymbol{\beta}$ is a vector of fitted coefficients for the fixed covariates, Z_{it} is the clustered covariate for cluster i for longitudinal measurement t, and b_{it} is the fitted random effect for cluster i for longitudinal measurement t. Note that GLMMs may be fitted when the cluster sizes differ (e.g. if there are different numbers of longitudinal measurements for each cluster).

Within the BiMM tree method, the linear constraint is relaxed and interaction coefficients do not need to be specified. The GLMM portion of the BiMM method has the form

$$\text{logit}(y_{it}) = \text{CART}(\boldsymbol{X_{it}})\boldsymbol{\beta} + Z_{it}b_{it}.$$

$\text{CART}(\boldsymbol{X_{it}})$ is represented within the GLMM as indicator variables reflecting membership of each longitudinal observation t for cluster i in terminal nodes within the CART model. Terminal nodes are at the bottom of CART models and provide an outcome prediction for each subject's observation. Figure 1 provides an example CART model with terminal Nodes 1, 3, 5 and 6. Thus, the terminal nodes of CART provide a method for determining similar groups of observations [1] which may be included within the GLMM portion of the BiMM method. In this example, $\text{CART}(\boldsymbol{X_{it}})$ would contain indicator variables for

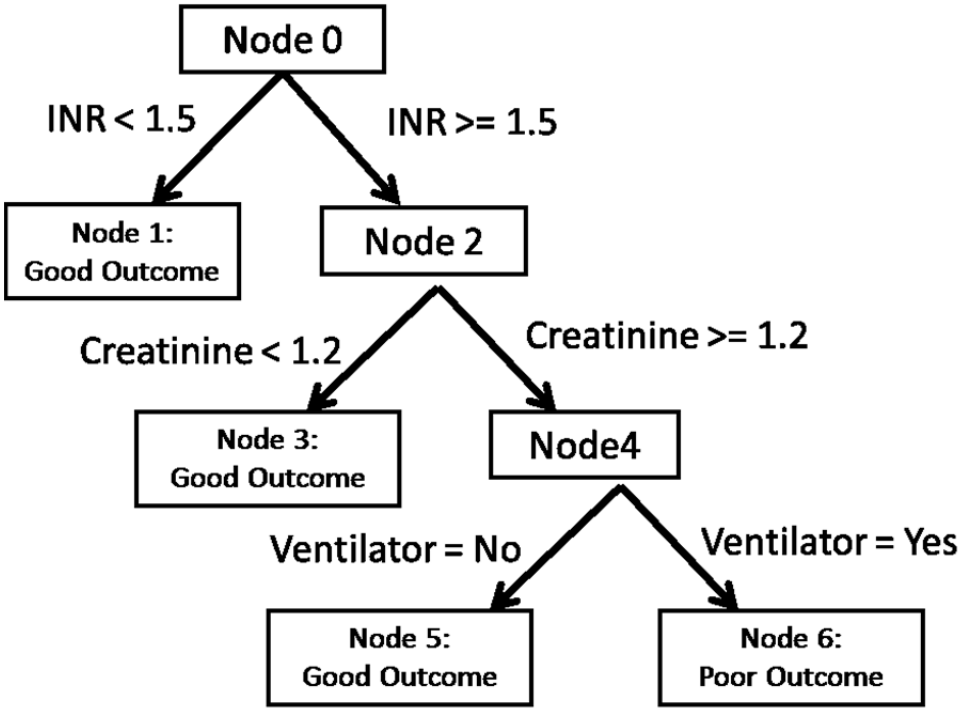


Figure 3.1: An example decision tree and the process used to generate data for the simulation study

being in Node 1, Node 3, and Node 5. It is not necessary to include the indicator variable for the last terminal node, Node 6, because this would be redundant information within the GLMM. This is consistent with traditional models, where one includes one less indicator variables than the number of categories in the regression framework.

Implementation of GLMMs is more challenging compared to standard linear mixed models employed for continuous outcomes. A consideration within the generalized model setting for categorical outcomes is that an iterative procedure (e.g. iterative reweighted least squares or Newton Raphson) must be used to compute random effects of clustered variables for GLMMs. GLMMs can have computational issues with model convergence or with inversion of large matrices, particularly when data are high dimensional, which makes GLMM fitting challenging [20]. Also, if data are quasi-

separated or completely separated, meaning that one or a combination of variables perfectly predicts the outcome, traditional implementations of GLMMs cannot be used [21, 22].

To address these challenges, we propose an algorithm that integrates CART and a Bayesian implementation of GLMM. There are several benefits to employing a Bayesian implementation of the GLMM instead of the traditional GLMM in our algorithm. First, Bayesian computation of GLMMs produce similar parameter estimates to that of frequentist GLMMs when uninformative prior distributions are used; however, weakly informative prior distributions can be used as a solution to separated or quasi-separated datasets [21]. Therefore, Bayesian implementation of the GLMM in the BiMM tree method offers more flexibility compared to frequentist GLMMs. Second, there are efficient methods for applying Bayesian GLMMs (e.g. integrated nested Laplace approximation implemented in the R package *INLA* [86] and maximum a posteriori estimation implemented in the R package *blme* [87, 88]) easily applied on open source software which offer similar computation time to frequentist GLMMs. Finally, employing the Bayesian GLMM avoids convergence issues with traditional GLMMs using the R package *lme4* [20, 89].

The Bayesian GLMM within the BiMM tree method considers uninformative priors for the fixed effects and random effect covariance parameters using Normal and Wishart distributions respectively. An unstructured covariance matrix is employed within the Bayesian GLMM. After the random effects for subjects are fitted with the Bayesian GLMM, the original outcome variable is updated using results from the CART and GLMM, which we define as the target outcome variable. A split function which divides

the observations into two groups is used to create a binary target outcome variable for each iteration since a simple additive effect does not result in a binary measure.

Specifically, the BiMM tree algorithm is as follows:

1. Initialize the CART and GLMM:
   a. Fit a CART using y_{it} as the outcome for fixed predictors ($\boldsymbol{X_{it}}$) and develop *J*-1 indicator variables for the *j* = 1, …, *J* terminal nodes of clusters *i* = 1,…,*M* for longitudinal measurements *t=1,…,*T_i:
   $$I(y_{it} \in node_j) = 1 \text{ if } y_{it} \text{ is in terminal node } j$$
   $$I(y_{it} \in node_j) = 0 \text{ if } y_{it} \text{ is not in terminal node } j$$
   Define CART($\boldsymbol{X_{it}}$) as the matrix of the *J*-1 indicator variables for cluster *i* at longitudinal measure *t*.
   b. Fit a Bayesian GLMM using y_{it} as the outcome, including $CART(\boldsymbol{X_{it}})$ and clustered variable (Z_{it}) to obtain fitted values for the random effect (b_{it}):
   $$\text{logit}(y_{it}) = \text{CART}(\boldsymbol{X_{it}})\boldsymbol{\beta} + Z_{it}b_{it}.$$
   c. Average predicted probabilities from the CART (denoted $\text{pr}_{CART}(\boldsymbol{X_{it}})$) and GLMM (denoted $\text{pr}_{GLMM}(\boldsymbol{X_{it}}, Z_{it})$) for each measurement *t* within cluster *i*:
   $$q_{it} = (\text{pr}_{CART}(\boldsymbol{X_{it}}) + \text{pr}_{GLMM}(\boldsymbol{X_{it}}, Z_{it}))/2$$
2. Iterate through the following steps until convergence is satisfied:
   a. Determine the target outcome (y_{it}^*) by adding the average predicted probability (q_{it}) from the original outcome (y_{it}) and applying a split function h() to make y_{it}^* a binary value:
   $$y_{it}^* = \text{h}(y_{it} + q_{it})$$
   b. Repeat steps 1a-c using y_{it}^* as the outcome until the change in the posterior log likelihood from the Bayesian GLMM is less than a specified tolerance value.

To summarize, the BiMM tree method begins by initializing the CART and GLMM models to obtain a predicted probability for each observation within the clusters. First, a CART model is developed using the binary outcome and all predictors assuming that the random effects are known, and indicator variables for the terminal nodes from the

CART are developed. These indicators for the CART terminal nodes (assumed to be known), as well as random effect variables, are then used in the Bayesian GLMM to account for longitudinal or clustered outcomes. The predicted probabilities from the CART model and the GLMM model are averaged because the goal of the algorithm is to combine population-level effects arising from the CART with cluster-level effects arising from the GLMM. We do not simply use the probabilities from the GLMM, which inherently consider the population-level effects, because the CART portion of the method is used for making new predictions and thus was the focus of model updating. The use of predicted probabilities from only the GLMM is investigated within our simulation study, and results in slightly worse model performance in terms of prediction accuracy (data not shown). The target outcome is updated using a split function which creates a binary outcome based on the sum of the original binary outcome and the average of the predicted probabilities from the CART and GLMM. The algorithm then continues iteratively fitting the fixed effects (from the CART) and random effects (from the Bayesian GLMM), updating the target outcome at each iteration, until the change in the posterior log likelihood is smaller than a specified value. Predictions for observations included within the model development dataset are made using the CART (population-level) and random (observation-level) components. For observations not included within the model development dataset, predictions are made using the CART (population-level) component only.

There are several different split functions (denoted $\mathrm{h}(y_{it} + q_{it})$) which may be used to create the new iteration of the binary target outcome (y_{it}^{*}). We use a function

of $y_{it} + q_{it}$ to update the target outcome to account for both the original outcome and the average predicted probability from the CART and GLMM models for the specific observation *t* within the cluster *i*. Before introducing the split functions, it is necessary to understand the distribution of $y_{it} + q_{it}$. Since y_{it} is a binary value, it iseither 0 or 1, and q_{it} is a probability which is between 0 and 1. Therefore, value of $y_{it} + q_{it}$ is between 0 and 2. We present three options for the split function which may be employed based on the overall goal of the prediction model. The first split function maximizes model sensitivity, the second split function maximizes model specificity, and the third split function equally weights sensitivity and specificity for updating the target outcome vector. Now, the split function which maximizes sensitivity uses a threshold ($0 < k_1 < 1$) to update the target outcome:

$$\mathrm{h}_1(y_{it} + q_{it}) = \begin{cases} 1 \text{ if } y_{it} + q_{it} > k_1 \\ 0 \text{ otherwise} \end{cases}$$

Thus, using $\mathrm{h}_1(y_{it} + q_{it})$, binary outcomes of 0 can be updated to be 1, but outcomes of 1 cannot be updated to be 0. This provides a mechanism for maximizing the sensitivity. Similarly, a split function which maximizes specificity may be employed using a threshold ($1 < k_2 < 2$) to update the target outcome:

$$\mathrm{h}_2(y_{it} + q_{it}) = \begin{cases} 0 \text{ if } y_{it} + q_{it} < k_2 \\ 1 \text{ otherwise} \end{cases}$$

Using $\mathrm{h}_2(y_{it} + q_{it})$, binary outcomes of 1 can be updated to be 0, but outcomes of 0 cannot be updated to be 1. This provides a mechanism for maximizing the specificity. A final, more general, split function which does not favor sensitivity or specificity updates the target outcome using the following method:

$$\mathrm{h}_3(y_{it}+q_{it}) = \begin{cases} 0 \text{ if } y_{it}+q_{it} < 0.5 \\ 1 \text{ if } y_{it}+q_{it} > 1.5 \\ 1 \text{ with probability } q_{it} \\ 0 \text{ otherwise} \end{cases}$$

Using $\mathrm{h}_3(y_{it}+q_{it})$, if the prediction from the current iteration of the BiMM method agrees with the original binary outcome (i.e. if $y_{it}+q_{it} < 0.5$ or if $y_{it}+q_{it} > 1.5$) then the target outcome is the same as the original binary outcome. Otherwise, the target outcome is updated to be 1 with probability q_{it}, and 0 with probability $1-q_{it}$. Therefore, original values of 0 can be updated to 1 and original values of 1 can be updated to 0.

An example of the four possible scenarios of an iteration within the BiMM method is depicted within Table 3.1, with $k_1 = 0.5$ and $k_2 = 1.5$ for observation *t* within cluster *i*. Using the split function $\mathrm{h}_1(y_{it}+q_{it})$, the original binary outcome (y_{it}) changes from a 0 to a 1 in Scenario B, which will increase the sensitivity since the next iteration of the BiMM method will contain more values of 1 within the target outcome. Likewise, using the split function $\mathrm{h}_2(y_{it}+q_{it})$, the original binary outcome (y_{it}) changes from a 1 to a 0 in Scenario C, which will increase the specificity since the next iteration of the BiMM method will contain more values of 0 within the target outcome. Using $\mathrm{h}_3(y_{it}+q_{it})$, the target outcomes are updated in Scenarios B and C based on the strength of the predicted probability from the BiMM iteration. In all split functions, if the original binary outcome agrees with the predicted probability from the BiMM iteration (i.e. in Scenarios A and D), then the target outcome is the original outcome.

Table 3.1: Example scenarios for split functions within the BiMM Tree method

Scenario	y_{it}	q_{it}	$y_{it} + q_{it}$	$y_{it}^{*} = \mathrm{h}_1(y_{it} + q_{it})$	$y_{it}^{*} = \mathrm{h}_2(y_{it} + q_{it})$	$y_{it}^{*} = \mathrm{h}_3(y_{it} + q_{it})$
A	0	$0 < q_{it} < 0.5$	$0 < y_{it} + q_{it} < 0.5$	0	0	0
B	0	$0.5 < q_{it} < 1$	$0.5 < y_{it} + q_{it} < 1$	1	0	1 with probability q_{it}, 0 otherwise
C	1	$0 < q_{it} < 0.5$	$1 < y_{it} + q_{it} < 1.5$	1	0	1 with probability q_{it}, 0 otherwise
D	1	$0.5 < q_{it} < 1$	$1.5 < y_{it} + q_{it} < 1$	1	1	1

BiMM trees for this study are computed using R software version 3.1.2 [90]. CART models are implemented using the R package *rpart* [91]. Default settings are used within the CART models, but we require that the minimum terminal node size be at least 10% of the development dataset so that node indicators within the Bayesian GLMM contain adequate data for fitting fixed effects. Bayesian GLMMs within the BiMM method are implemented using the R package *blme* [87, 88], again with all default settings. Thus, uninformative prior distributions are used for both fixed (Normal prior distribution) and random effects (Wishart prior distribution for the unstructured covariance matrix of clustered variables). However, alternative prior distributions may be applied if separation or convergence issues arise.

3.1.4 Data Description

ALF occurs in approximately 2,000 patients in the United States each year, with about half of the cases attributed to acetaminophen overdose [85]. A critical goal of the ALF Study Group is to predict the likelihood of poor outcomes of acetaminophen-induced ALF patients which may be used both on hospital admission and post-hospital admission [92]. The ALF Study Group registry consists of over 2,700 patients with a multitude of clinical data (e.g. laboratory values, treatments, complications, etc.) collected daily for up to seven days following enrollment unless a patient is transplanted, discharged from the hospital or dies. To date, most prognosis prediction models for ALF patients use variables collected at a single baseline time point (e.g. King's College Criteria and Clichy Criteria [26, 27]). Many patients may remain alive for longer periods beyond the initial insult because of advances in intensive care unit management [93, 94]. Thus, there is a need for a prediction model which may be used to determine prognosis of

acetaminophen-induced ALF patients (poor or favorable outcome) each day which can aid clinicians in management of patients during the first week of hospitalization. We define a poor outcome as having coma grade of III or IV and favorable outcome as having a coma grade of 0, I or II.

The ALF registry dataset contains many clinical predictor variables which may be used in modeling outcome. A few fixed predictor variables included within the registry are gender, ethnicity, and age. Some examples of continuous predictor variables collected daily for the first week in the hospital include aspartate aminotransferase (AST), alanine aminotransferase (ALT), creatinine, bilirubin and international normalized ratio (INR). Categorical variables collected daily include treatments and clinical measurements such as mechanical ventilation, pressor use, and renal replacement therapy.

3.1.5 Simulation Study Design

To assess the predictive performance of the proposed BiMM tree method, we conduct a simulation study based on the real motivating dataset, the ALF Study Group registry. We simulate data from the ALF registry for several reasons. First, the complexity of the ALF dataset allows comparing of novel and traditional methodologies in realistic settings. Additionally, the ALF dataset contains multiple continuous predictors which are not normally distributed and several categorical variables, so our simulated ALF data provides various types of predictors which are consistent with data that arises from real-world scenarios. A final reason we simulate data based on the real ALF dataset is that a correlation structure between repeated measures on the same person is not imposed, so we can evaluate the performance of proposed methodology with a real observed correlation structure within the ALF data.

We construct a dataset from which we sample simulation data by selecting all data from acetaminophen-induced ALF patients within the registry (N=1064) and imputing all missing predictor data using an imputation method [95] for multilevel data to preserve the original correlation structure between predictor variables within the dataset. Thus, the simulated datasets contain 1064 patients with complete data for seven days (three fixed predictors and eight longitudinal predictors). We use two data generating processes for the fixed portion of the outcome: a tree structure and a linear structure. For both processes, variables related to the outcome include INR, creatinine, and ventilator use, which is consistent with clinical literature [92, 96]. The other five longitudinal variables and the three fixed predictors are included within the simulation datasets as noise variables. The tree data generating process is depicted within Figure 3.1, which is read like a CART (i.e. begin at Node 0 and follow the arrow corresponding to the predictor variable values until a terminal node is reached). Nodes 1, 3 and 5 represent favorable outcome for the subject on the specific day, whereas Node 6 represents poor outcome for the subject on the specific day. The equation for the linear data generating process is:

$$\text{logit}(poor\ outcome_{it}) = -2.3 + 1.4 * ln(INR_{it}) + 0.6 * Creatinine_{it} + 2.1 * I(Ventilator_{it})$$

where $I(Ventilator_{it})$ is 1 if patient i is on a ventilator on the specific day t, and is 0 otherwise. Thus, high INR and creatinine and being on a ventilator are associated with higher likelihood of poor outcomes, consistent with clinical literature [96].

Small and large random effects are added to the fixed portion of the outcome to create a within-subject correlation structure. The small random effect is generated for each subject from a normal distribution centered at zero with standard deviation of 0.1,

whereas the large random effect is generated for each subject from a normal distribution centered at zero with standard deviation of 0.5. To derive the outcome of observations at every time point, the fixed portion (from the tree or linear data generating process) is added to the random effect, and a cut point is used to create the binary outcome. We used a threshold to create an unbalanced outcome, with approximately one-third of the observations having poor outcome and two-thirds having favorable outcome for each simulated dataset.

Using the simulated datasets described in the previous paragraphs based on the ALF registry, we compare the performance of several models: CART (which ignores clustering within the data), Bayesian GLMM, BiMM tree with one iteration (i.e. only Step 1 in the algorithm is performed), and BiMM tree algorithm with more than one iteration. We use $\mathrm{h}_1(y_{it} + q_{it})$ as our split function for updating the target outcome with a threshold (k_1) of 0.5 because clinicians often prefer to develop prediction models maximized for sensitivity to identify patients at highest risk of poor outcomes. For comparison, we also compile BiMM trees using $\mathrm{h}_3(y_{it} + q_{it})$ with $k_1 = 0.5$ and $k_2 = 1.5$. All models are fit using all predictors in the data (i.e. both those associated with outcome and those that were noise variables). We produce models for BiMM trees with one iteration (denoted BiMM Tree 1) and with multiple iterations (denoted BiMM Tree H1 and BiMM Tree H3 for the respective split functions $\mathrm{h}_1(y_{it} + q_{it})$ and $\mathrm{h}_3(y_{it} + q_{it})$) to assess if iterating between fixed and random effects results in increased prediction accuracy. Models are compiled for 1000 simulation runs. Sample sizes (number of subjects) for training datasets used in model development are 100, 250 and

500. All test datasets consist of 500 new subjects not included within the training dataset. The numbers of repeated measurements of outcomes in our simulation study are 2, 4 and 7.

All simulations are conducted using R software version 3.1.2 [90]. To implement the CART models, the R package *rpart* is used with default settings [91]. To implement the Bayesian GLMMs, the R package *blme* is used with default settings so that prior distributions for parameters are uninformative [87, 88].

3.1.6 Simulation Study Results

Since the main objective in this study is to develop methodology for predicting new observations, we assess the prediction (test set) accuracy of the models, defined as the number of correct predictions divided by the total number of predictions made. Prediction accuracy is presented within Figure 3.2 for the sample size of 100. Overall, the BiMM trees with one iteration or more than one iteration have higher accuracy compared to CART and Bayesian GLMM when the random effect is large, regardless of whether the data are generated using a tree or linear structure. When the random effect is small, the accuracy distributions overlap, with the CART models generally having slightly higher accuracy compared to the BiMM trees. With a linear data generating process and small random effect, the CART and Bayesian GLMM have similar predictive accuracy, whereas with a tree data generating process and small random effect, the Bayesian GLMM has the lowest prediction accuracy. The Bayesian GLMM also has the lowest prediction accuracy for the tree data generating process with a large random effect. The BiMM tree models with one iteration and with multiple iterations generally have similar

predictive accuracies for each of the scenarios. Similar results were obtained for the sample sizes of 250 and 500 (Appendix 1 Figures 1 and 2).

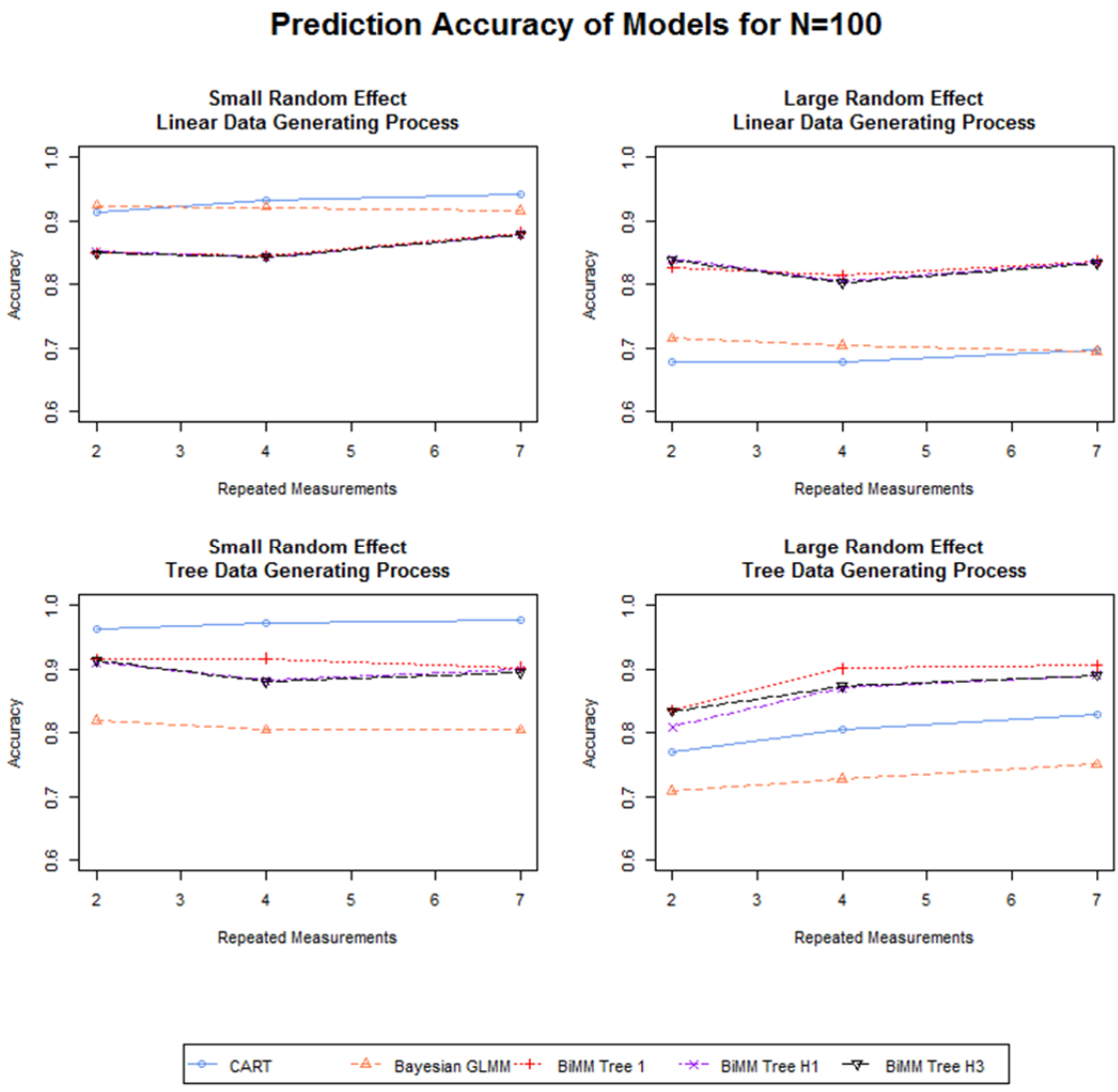


Figure 3.2: Simulated prediction (test set) accuracy of models for N=100 patients

Most BiMM iterative trees converge in two iterations regardless of the split function, and in rare cases convergence is reached in three or four iterations. Table 3.2 contains the median (interquartile range) estimates of prediction accuracy of the test dataset for each simulation scenario for CART, Bayesian GLMM, BiMM tree with one

Table 3.2: Median Prediction (Test set) Accuracy (Interquartile Range) for Simulated Datasets

Model	Repeated Outcomes	N=100				N=250			
		Linear DGP		Tree DGP		Linear DGP		Tree DGP	
		Small RE	Large RE	Small RE	Large RE	Small RE	Large RE	Small RE	Large RE
CART	2	0.913 (0.890,0.925)	0.678 (0.651.0.699)	0.963 (0.950,0.971)	0.769 (0.745,0.791)	0.929 (0.920,0.937)	0.706 (0.690,0.722)	0.970 (0.965,0.975)	0.796 (0.780,0.812)
	4	0.933 (0.922,0.940)	0.679 (0.659,0.699)	0.973 (0.966,0.978)	0.805 (0.788,0.821)	0.942 (0.937,0.947)	0.721 (0.707,0.733)	0.975 (0.970,0.979)	0.830 (0.818,0.843)
	7	0.941 (0.935,0.947)	0.697 (0.680,0.714)	0.977 (0.972,0.981)	0.828 (0.814,0.842)	0.946 (0.942,0.950)	0.732 (0.720,0.743)	0.979 (0.975,0.982)	0.853 (0.840,0.863)
Bayesian GLMM	2	0.924 (0.914,0.933)	0.715 (0.700,0.730)	0.820 (0.805,0.835)	0.709 (0.691,0.723)	0.939 (0.933,0.944)	0.731 (0.718,0.745)	0.836 (0.826,0.847)	0.722 (0.708,0.733)
	4	0.922 (0.915,0.928)	0.704 (0.689,0.716)	0.805 (0.795,0.814)	0.728 (0.717,0.738)	0.930, (0.926,0.934)	0.717 (0.704,0.729)	0.812 (0.804,0.820)	0.735 (0.724,0.744)
	7	0.917 (0.912,0.921)	0.695 (0.682,0.707)	0.805 (0.796,0.812)	0.750 (0.741,0.760)	0.921 (0.917,0.925)	0.707 (0.696,0.717)	0.808 (0.801,0.814)	0.755 (0.746,0.763)
BiMM Tree 1 Iteration	2	0.849 (0.833,0.868)	0.827 (0.776,0.852)	0.916 (0.897,0.927)	0.836 (0.782,0.902)	0.850 (0.838,0.866)	0.850 (0.830,0.873)	0.921 (0.914,0.929)	0.911 (0.880,0.923)
	4	0.845 (0.822,0.860)	0.815 (0.780,0.849)	0.917 (0.881,0.952)	0.901 (0.854,0.956)	0.850 (0.835,0.862)	0.837 (0.807,0.862)	0.942 (0.888,0.959)	0.947 (0.894,0.964)
	7	0.881 (0.869,0.891)	0.837 (0.806,0.864)	0.902 (0.892,0.913)	0.905 (0.862,0.920)	0.887 (0.878,0.894)	0.860 (0.837,0.889)	0.905 (0.895,0.914)	0.907 (0.864,0.914)
BiMM Tree H1 Algorithm	2	0.852 (0.832,0.876)	0.840 (0.813,0.854)	0.910 (0.842,0.924)	0.809 (0.757,0.870)	0.856 (0.842,0.874)	0.847 (0.836,0.861)	0.918 (0.904,0.925)	0.858 (0.819,0.916)
	4	0.844 (0.823,0.857)	0.806 (0.790,0.840)	0.882 (0.868,0.925)	0.871 (0.817,0.934)	0.847 (0.830,0.860)	0.820 (0.800,0.850)	0.887 (0.873,0.947)	0.882 (0.855,0.952)
	7	0.879 (0.862,0.891)	0.835 (0.814,0.847)	0.899 (0.890,0.911)	0.891 (0.801,0.910)	0.887 (0.876,0.894)	0.842 (0.832,0.852)	0.905 (0.894,0.913)	0.897 (0.853,0.909)
BiMM Tree H3 Algorithm	2	0.849 (0.834,0.867)	0.838 (0.804,0.852)	0.913 (0.850,0.924)	0.834 (0.775,0.909)	0.850 (0.839,0.865)	0.846 (0.835,0.857)	0.920 (0.912,0.928)	0.908 (0.830,0.923)
	4	0.843 (0.822,0.857)	0.803 (0.788,0.833)	0.881 (0.868,0.938)	0.874 (0.848,0.943)	0.848 (0.832,0.861)	0.807 (0.793,0.842)	0.881 (0.871,0.949)	0.879 (0.860,0.953)
	7	0.879 (0.852,0.891)	0.833 (0.805,0.845)	0.895 (0.887,0.905)	0.891 (0.805,0.909)	0.886 (0.877,0.894)	0.840 (0.831,0.849)	0.899 (0.890,0.909)	0.895 (0.785,0.907)

Model	Repeated Outcomes	N=500			
		Linear DGP		Tree DGP	
		Small RE	Large RE	Small RE	Large RE
CART	2	0.942 (0.936,0.948)	0.732 (0.718,0.745)	0.971 (0.965,0.975)	0.821 (0.806,0.832)
	4	0.945 (0.941,0.950)	0.735 (0.724,0.745)	0.975 (0.971,0.980)	0.844 (0.833,0.853)
	7	0.947 (0.944,0.951)	0.740 (0.729,0.750)	0.979 (0.975,0.982)	0.859 (0.849,0.867)
Bayesian GLMM	2	0.943 (0.938,0.948)	0.737 (0.723,0.748)	0.842 (0.833,0.851)	0.725 (0.714,0.737)
	4	0.932 (0.928,0.936)	0.721 (0.710,0.732)	0.814 (0.808,0.821)	0.736 (0.727,0.746)
	7	0.923 (0.919,0.926)	0.710 (0.699,0.721)	0.809 (0.803,0.814)	0.756 (0.748,0.765)
BiMM Tree 1 Iteration	2	0.850 (0.840,0.863)	0.856 (0.840,0.886)	0.920 (0.915,0.926)	0.917 (0.907,0.925)
	4	0.849 (0.838,0.862)	0.841 (0.813,0.865)	0.952 (0.891,0.960)	0.953 (0.900,0.963)
	7	0.890 (0.884,0.895)	0.867 (0.843,0.890)	0.905 (0.897,0.913)	0.906 (0.858,0.913)
BiMM Tree H1 Algorithm	2	0.859 (0.845,0.874)	0.851 (0.841,0.866)	0.918 (0.909,0.924)	0.852 (0.822,0.917)
	4	0.847 (0.832,0.860)	0.816 (0.796,0.852)	0.886 (0.874,0.901)	0.878 (0.851,0.904)
	7	0.890 (0.884,0.895)	0.843 (0.836,0.851)	0.905 (0.896,0.912)	0.896 (0.725,0.907)
BiMM Tree H3 Algorithm	2	0.850 (0.840,0.863)	0.848 (0.838,0.859)	0.920 (0.914,0.926)	0.915 (0.852,0.923)
	4	0.848 (0.836,0.861)	0.805 (0.793,0.839)	0.879 (0.869,0.891)	0.877 (0.862,0.894)
	7	0.890 (0.884,0.895)	0.842 (0.835,0.849)	0.890 (0.891,0.910)	0.895 (0.776,0.904)

iteration, and BiMM tree algorithm with more than one iteration. Interquartile ranges of prediction accuracy for the models in the different scenarios are relatively tight around the median estimates, indicating that the distribution of prediction accuracy for models does not vary greatly over the simulation runs. Across 2, 4 and 7 repeated measurements for the models and scenarios, prediction accuracy is similar, except for the tree data generating process with a large random effect, where slight gains in accuracy are achieved with increasing number of repeated measurements. In general, the prediction accuracy estimates are similar for sample sizes of 100, 250 and 500, with slight improvements in accuracy for BiMM models with larger sample sizes.

In addition to assessing the predictive accuracy of models, we present the difference between training and test accuracy for models in the simulated scenarios to measure the amount of overfitting in models for sample size of 100 (Figure 3.3). Within this plot, large values of the difference between the training and test datasets indicate that the accuracy of the training dataset is larger than the accuracy of the test dataset. For small random effects, CARTs, Bayesian GLMMs, BiMM trees with one iteration, and BiMM trees updated with $\mathrm{h}_3(y_{it} + q_{it})$ have minimal overfitting, since the difference between training and test set accuracy is small. However, for small random effects, BiMM trees with multiple iterations updated with $\mathrm{h}_1(y_{it} + q_{it})$ have larger differences in accuracy, suggesting the models may have overfit the training data. When random effects are large, the CART models tend to overfit the training data the most for both data generating processes. For the tree data generating process with a large random effect, the Bayesian GLMM overfits the data more than the BiMM trees, but this is only a slight

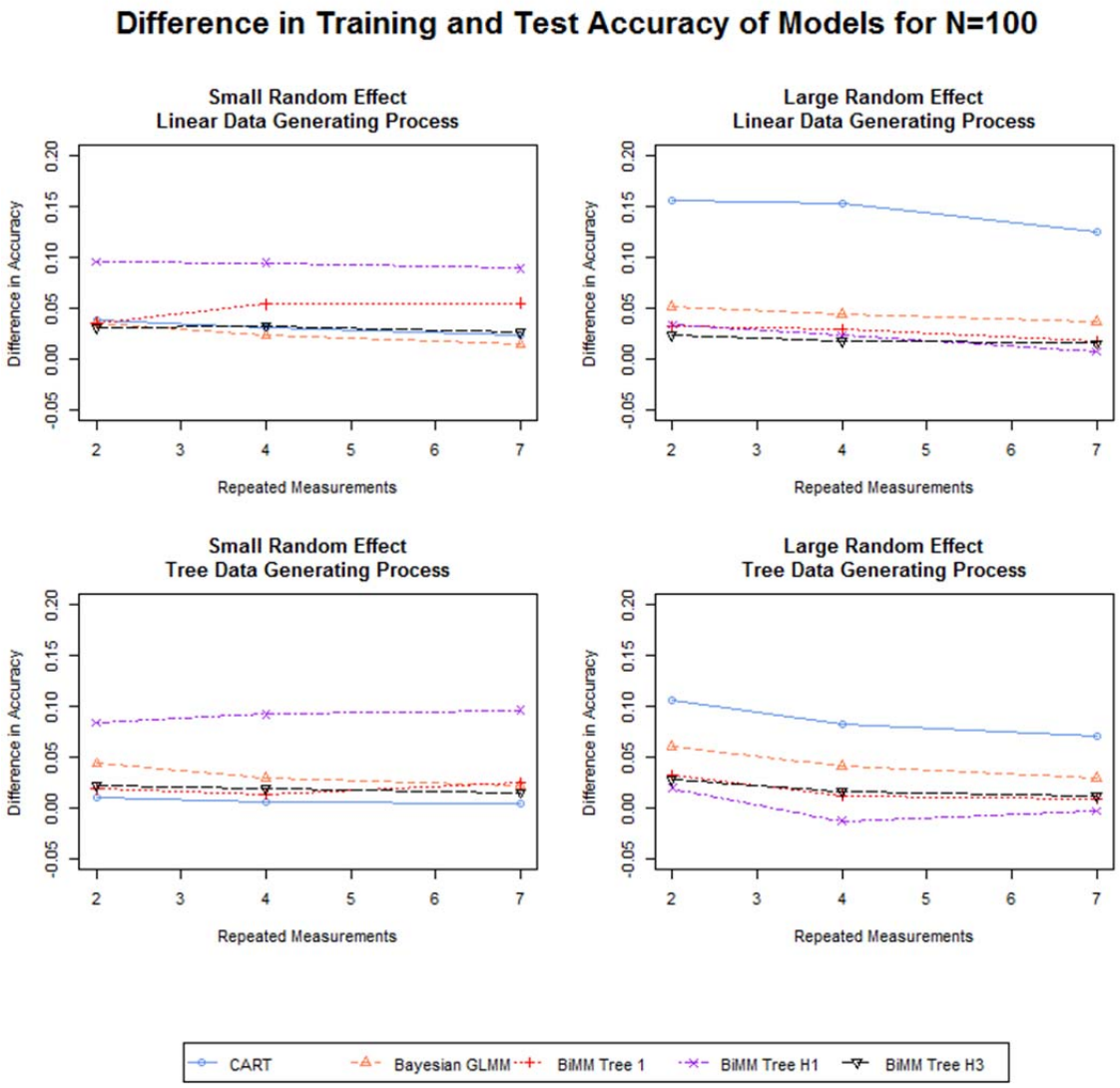


Figure 3.3: Simulated difference in training and test set accuracy of models for N=100 patients

difference. Regardless of the data generating process, the BiMM trees overfit the training data the least. As the number of repeated measurements increase, model overfitting slightly decreases for large random effect datasets, whereas model overfitting remains

similar for datasets with small random effects. The performance of each model in terms of overfitting is similar for the sample sizes of 250 and 500; however, the amount of overfitting is slightly less with the larger sample sizes for datasets with large random effects (Appendix 1 Figures 3 and 4).

In general, BiMM trees have high predictive accuracy for scenarios when there is a large clustering effect or when the data generating process follows a tree structure. For the simulated scenarios generated for a small random effect, BiMM trees with one iteration and BiMM trees with multiple iterations with the split function $\mathrm{h}_3(y_{it} + q_{it})$ tend to overfit the data less than BiMM trees using multiple iterations with the split function $\mathrm{h}_1(y_{it} + q_{it})$.

3.1.7 Discussion

Overall, the BiMM tree framework may offer advantages compared to CARTs and Bayesian GLMMs. The main benefit of BiMM tree compared to CART is that it can account for clustered outcomes in modeling so that the assumption of independent observations is not violated. BiMM trees do not require specification of nonlinear relationships or interaction terms and can be implemented for high dimensional datasets. A strength of BiMM tree is that nonlinear forms of predictors and interactions between predictors are developed by the method based on the data. The computation time of GLMMs and BiMM trees are similar, yet BiMM trees may offer higher prediction accuracy for certain situations (when the underlying structure of the data is a tree form and when there is a large clustered effect for the outcomes). A final strength of BiMM tree methodology is that missing values in predictor data is naturally handled using

surrogate splits (using other non-missing variables) within the CART portion of the algorithm; thus, observations with missing predictor data can still be included within BiMM models. GLMMs only use complete cases within datasets, so missing values would need to be imputed (filled in) within the GLMM setting in order to use the entire dataset. The BiMM tree method does not require missing data to be imputed prior to model development.

A major distinction of the BiMM tree framework compared to other decision tree methods for longitudinal and clustered outcomes within the literature [7-16] is the Bayesian implementation of GLMMs. For continuous outcomes, there are fewer issues with GLMM convergence because estimates may be computed directly; however, with categorical outcomes complete or quasi-separation may pose a challenge to GLMM fitting. The default priors specified in the BiMM tree method are uninformative, but if convergence issues arise, weakly informative priors may be used for estimating the random effects [21].

BiMM tree provides a flexible, data-driven predictive modeling framework for longitudinal and clustered binary outcomes. Our simulation study demonstrates that BiMM tree may be advantageous compared to CART which ignore clustered outcomes and Bayesian GLMM when predictors are not linearly related to the outcome through the link function and when the random effect of the clustered variable is large. Though standard CART models can have high predictive accuracy if random effects are small, failing to account for large clustering effects causes a sizeable decrease in prediction accuracy in our simulations. While Bayesian GLMM can be used to adjust for clustering within the data, model misspecification may reduce prediction accuracy (e.g. not

including a significant interaction term or specifying an incorrect nonlinear relationship between predictor and outcome). This is evident in our simulation study, where BiMM tree models have higher prediction accuracy compared to Bayesian GLMMs if the data has a tree structure or if there is a large clustering effect between outcomes. One possible reason that the Bayesian GLMMs did not perform well for simulated data in this study is that some of the continuous predictor variables have skewed distributions, and extreme values may have adversely affected the GLMM parameter estimates.

The BiMM trees with one iteration generally have similar prediction accuracy compared to BiMM trees with more than one iteration within our simulation study. While the training dataset accuracies for the BiMM trees with more than one iteration are higher than the BiMM trees with only one iteration, the multiple iteration method with the split function which maximizes sensitivity produces overfitted models which do not predict well for test datasets if the effect of clustering within subjects is small. BiMM trees which iterate between fixed and random effects have slightly higher computation time and offer minimal increases in prediction accuracy, suggesting that BiMM trees with one iteration may be sufficient. It is possible that real-world datasets are more complex than our simulated datasets, though, so multiple iterations may be necessary in some situations. However, one may easily assess this by compiling both BiMM tree models with one iteration and with multiple iterations and comparing the posterior log likelihoods.

Another interesting result from the simulation study is that the prediction accuracy of models remained similar whether models were developed using 2, 4 or 7 repeated measurements. We expected to see increases in prediction accuracy with increases in the number of repeated measurements. However, the simulated dataset for

our study is created based on the real ALF Study Group registry, so this result may be because the clustering effect for repeated outcomes does not change whether 2, 4 or 7 measurements are included. Though a simulated dataset could have been constructed to induce a specific correlation structure for repeated observations (e.g. autoregressive structure), we wanted the data simulation to resemble our motivating dataset as closely as possible. Our simulated dataset based on the real ALF registry also allows us to assess how the models performed when certain aspects of the data make modeling challenging (e.g. collinear predictors, predictors with skewed distributions with extreme values, and complex interactions between predictors). A future study could assess the performance of BiMM tree methodology for more complex simulated scenarios, such as a high dimensional dataset or a dataset containing nonlinear predictors and high-order interactions.

The main objective of this study is to develop a flexible framework for constructing prediction models for binary outcomes. BiMM tree methodology offers comparable or higher prediction accuracy to other models and may be considered an alternative to using GLMM for complex datasets. Future work could investigate the use of alternative implementations of decision tree algorithms within the BiMM tree framework for modeling longitudinal and clustered binary outcomes (e.g. C4.5, GUIDE, QUEST, CRUISE, BART and bartMachine [37, 97, 98]).

An R package for implementing BiMM tree methodology is being developed and will be available on the Comprehensive R Archive Network. An R program implementing BiMM tree methodology is available in Appendix 2.

3.2 Specific Aim 2: **To develop a RF method for clustered and longitudinal binary outcomes using an iterative procedure to combine RF and mixed effect models**

3.2.1 Introduction

Often in research settings, measurements of binary outcomes are clustered within a group which results in a correlation structure. For example, repeated measurements of outcomes may be collected for patients over time within a clinical study to evaluate disease prognosis (e.g. poor versus good outcome), diagnosis or disease relapse (e.g. disease versus disease-free), or other endpoints (e.g. re-admitted versus not re-admitted to the hospital). In longitudinal or repeated measurements studies, each patient represents a cluster. Another example of a cluster is a hospital or study center because outcomes for patients at the same location may be correlated. Within the setting of clustered data, a common goal is to develop prediction models that determine the probability of an event of interest given a set of prognostic factors and cluster groups. Statistical models should account for within-cluster correlation when it is present in a dataset.

Common statistical methods for developing prediction models for clustered and longitudinal binary outcomes have limitations. Generalized linear mixed models (GLMMs) typically employed for datasets with clustered outcomes cannot be implemented for high dimensional datasets, when the number of predictors is larger than the number of observations. Another limitation of GLMMs is that users must specify interactions among predictors and nonlinear relationships between predictors and outcome, which is not always straightforward. In Aim 1 of this dissertation, we propose a more flexible, data-driven framework called Binary Mixed Model (BiMM) tree, which

combines decision tree and GLMM within a unified framework. BiMM tree addresses many of the limitations of GLMMs: it can be employed for high dimensional datasets and naturally handles interactions among predictors and nonlinear relationships of predictors with outcomes. For these reasons, BiMM tree demonstrates higher prediction accuracy than standard GLMMs for datasets with a large clustering effect; however, decision tree methodologies such as BiMM tree, can be unstable, meaning that changes in predictor variables considered in modeling and changes in the observations (e.g. deleting or adding an observation) can result in very different models. Stability, as well as prediction accuracy, can often be improved by developing many decision trees and aggregating results within an ensemble method (e.g. random forest (RF)[4]).

In this paper, we propose an extension of BiMM tree called BiMM forest, which combines RF methodology with GLMM. RF can be implemented to develop prediction models which can be used in the setting of high dimensional data. Also, RF naturally handles nonlinear relationships between predictors and outcome, as well as interactions among predictor variables, without user specification of these relationships. Thus, RF provides a flexible framework for developing prediction models which offers superior prediction accuracy compared to standard parametric models and machine learning models for datasets without clustering effects [99]. In the BiMM forest method, we incorporate results from RF within mixed models to adjust for clustered and longitudinal outcomes.

A specific motivating example dataset for the novel BiMM forest methodology is a longitudinal registry dataset of acute liver failure (ALF) patients (clinicaltrials.gov ID: NCT00518440). ALF is a rare and devastating condition characterized by rapid onset of

severe liver damage, encephalopathy (altered mental status) and coagulopathy (impaired blood clotting). Approximately 25% of patients require a liver transplant and 30% of patients die during the acute phase [85]. The ALF Study Group registry dataset is complex, including skewed distributions of predictors with many extreme values, nonlinear predictors of outcome, and multi-way interactions among predictor variables. Also, it is a relatively high dimensional dataset, which poses a challenge in developing prediction models.

The chapter is structured as follows. In Section 3.2.2, we present background information about random forest modeling in general and forest models for longitudinal and clustered continuous outcomes. In Section 3.2.3, we introduce the BiMM forest method for predicting longitudinal and clustered binary outcomes. We compare the BiMM forest method performance to several other methods with a simulation study in Sections 3.2.4 and 3.2.5. Finally, in Section 3.2.6 we discuss implications of our study, limitations, and avenues for further research.

3.2.2 Background

A RF framework is implemented for the novel BiMM forest method because it offers several advantages compared to traditional GLMMs. There are many different ensemble methods available, and we implement our BiMM forest method with the RF framework, a commonly used methodology developed by Breiman [4]. RF does not require specification of nonlinear relationships or interaction terms, and offers information about the relative importance of predictor variables. Moreover, RF provides an alternative method for developing prediction models when traditional models are not feasible (e.g. if the number of predictor variables is greater than the number of

observations). While decision tree methods are sometimes unstable, aggregation of tree models within the random forest setting generally offers improved stability. This means that small changes within the data or variables included will not result in substantially different predictions from the random forest model. For these reasons, RF can often better predict outcomes compared to other procedures such as logistic regression, classification and regression trees, and support vector machines for data captured at a single time point [99].

In spite of this flexibility, few RF methods exist for modeling clustered categorical endpoints. The R package *party* can be used to implement random forest models if two predictor variables are correlated, but it does not adjust for longitudinal and clustered measurements of the same outcome variable [19]. There are some techniques which circumvent the issue of adjusting for longitudinal and clustered outcomes, such as summarizing variables (e.g. using averages or most frequent categorical values) or using data from only a single time point (e.g. admission values); however, these methods have a marked loss of information since available data is summarized or partially used.

Several methods have been proposed to modify decision tree and RF models for longitudinal and clustered continuous outcomes [7-16]. Hajjem [10] and Sela [15] develop similar methods for implementing models for longitudinal and clustered data with continuous outcomes. These methods incorporate mixed effects within the tree framework to account for clustering effects, using an algorithm analogous to expectation-maximization described by Wu and Zhang [18]. Hajjem [17] extends the decision tree method to the RF setting for clustered and longitudinal continuous outcomes.

Aside from these decision tree and RF methods for clustered continuous outcomes, a decision tree method for clustered and longitudinal binary outcomes called BiMM tree in Aim 1 of this dissertation. Using a similar framework to continuous outcome methods by Hajjem [10, 17] and Sela [15], BiMM tree combines classification and regression tree methodology [1] with GLMMs. While the framework for clustered decision tree modeling has been developed for binary outcomes, adjusting the algorithm for the RF setting is non-trivial. In the BiMM tree method, indicator variables for the terminal nodes of the decision tree are used within the GLMM. For a RF model, several hundred decision trees are developed. Thus, simply using indicator variables for terminal nodes within the GLMM would not be possible because there would likely be more indicator variables than observations (i.e. high dimensional data), and it would not be possible to develop the GLMM. In order to address these challenges, in this paper, we propose the BiMM Forest, which will be discussed in detail in the next section.

3.2.3 BiMM Forest Method

The BiMM forest method iterates between developing RF models using all predictors and then using information from the RF model within a Bayesian GLMM to account for the clustered structure of the outcome. Consistent with the continuous methods for clustered decision trees and the BiMM tree framework, we implement an algorithm similar to the expectation-maximization algorithm, in which the fixed (forest) effects are dissociated from the random (cluster-level) effects. While developing the RF model, it is assumed that the random effects are known, and while developing the Bayesian GLMM, it is assumed that the fixed components are known. The BiMM forest method may be considered as an extension of GLMMs where the fixed covariates are not

assumed to be linearly associated with the link function of the outcome and interactions do not need to be pre-specified. The traditional GLMM for binary outcomes has the form

$$\text{logit}(y_{it}) = \boldsymbol{X_{it}\beta} + Z_{it}b_{it},$$

where y_{it} is the binary outcome for cluster i = 1,…,M for longitudinal measurements t=1,…,T_i, logit() is the logistic link function, $\boldsymbol{X_{it}}$ is a matrix of fixed covariates for cluster i for longitudinal measurement t, $\boldsymbol{\beta}$ is a vector of coefficients for the fixed covariates, Z_{it} is the clustered covariate for cluster i for longitudinal measurement t, and b_{it} is the random effect for cluster i for longitudinal measurement t. Note that GLMMs may be fitted when the cluster sizes differ (e.g. if there are different numbers of longitudinal measurements for each cluster).

Within the BiMM forest method, the linear constraint is relaxed and interaction coefficients do not need to be specified. The GLMM portion of the BiMM method has the form

$$\text{logit}(y_{it}) = \text{RF}(\boldsymbol{X_{it}})\beta + Z_{it}b_{it}.$$

$\text{RF}(\boldsymbol{X_{it}}) = (\text{RF}(\boldsymbol{X_{11}}) \ldots \text{RF}(\boldsymbol{X_{1T_1}})\ \text{RF}(\boldsymbol{X_{21}}) \ldots \text{RF}(\boldsymbol{X_{2T_2}}) \ldots\ \text{RF}(\boldsymbol{X_{M1}}) \ldots \text{RF}(\boldsymbol{X_{MT_I}}))'$ is represented within the GLMM as the predicted probability of each longitudinal observation t=1,…,T_i for cluster i=1,…,M. β is the coefficient for the vector $\text{RF}(\boldsymbol{X_{it}})$. We used the predicted probability from the RF model to incorporate covariate effects within the GLMM model which adjusts for dependent observations within clusters.

Implementation of GLMMs is more challenging compared to standard linear mixed models employed for continuous outcomes. A consideration within the generalized model setting for categorical outcomes is that an iterative procedure (e.g. iterative

reweighted least squares or Newton Raphson) must be used to compute random effects of clustered variables for GLMMs. GLMMs can have computational issues with model convergence or with inversion of large matrices, particularly when data are high dimensional, which makes GLMM fitting challenging [20]. Also, if data are quasi-separated or completely separated, meaning that one or a combination of variables perfectly predicts the outcome, traditional implementations of GLMMs cannot be used [21, 22].

To address these challenges, we propose an algorithm that integrates RF and a Bayesian implementation of GLMM [88]. There are benefits to employing a Bayesian implementation of the GLMM instead of the traditional GLMM in our algorithm. First, Bayesian computation of GLMMs produce similar parameter estimates to that of frequentist GLMMs when uninformative prior distributions are used. Second, employing the Bayesian GLMM avoids convergence issues with traditional GLMMs, e.g. using the R package *lme4* [20, 89]. Finally, there are efficient methods for applying Bayesian GLMMs (e.g. integrated nested Laplace approximation implemented in the R package *INLA* [86] and maximum a posteriori estimation implemented in the R package *blme* [87, 88]) easily applied on open source software which offer similar computation time to frequentist GLMMs.

The Bayesian GLMM within the BiMM forest method considers uninformative priors for the fixed effect and random effect covariance parameters using Normal and Wishart distributions respectively. An unstructured covariance matrix is employed within the Bayesian GLMM. After the random effects for subjects are fitted with the Bayesian GLMM, the original outcome variable is updated using results from the RF and GLMM,

which we define as the target outcome variable. A split function which divides the observations into two groups is used to create a binary target outcome variable for each iteration since a simple additive effect does not result in a binary measure.

Specifically, the BiMM forest algorithm is as follows:

3. Initialize the RF and GLMM:
    a. Fit a RF using y_{it} as the outcome for fixed predictors ($\boldsymbol{X_{it}}$) and calculate predicted probabilities for clusters i = 1,…,M for longitudinal measurements t=1,…,T_i. Define RF($\boldsymbol{X_{it}}$) as the predicted probability from the RF for cluster i at longitudinal measure t.
    b. Fit a Bayesian GLMM using y_{it} as the outcome, including RF($\boldsymbol{X_{it}}$) and clustered variable (Z_{it}) to obtain fitted values for the random effect (b_{it}):
    $$\text{logit}(y_{it}) = \text{RF}(\boldsymbol{X_{it}})\boldsymbol{\beta} + Z_{it}b_{it}.$$
    c. Average predicted probabilities from the RF and GLMM (denoted $\text{pr}_{GLMM}(\boldsymbol{X_{it}}, Z_{it})$) for each measurement t within cluster i:
    $$q_{it} = (\text{RF}(\boldsymbol{X_{it}}) + \text{pr}_{GLMM}(\boldsymbol{X_{it}}, Z_{it}))/2$$
4. Iterate through the following steps until convergence is satisfied:
    a. Determine the target outcome (y_{it}^*) by adding the average predicted probability (q_{it}) from the original outcome (y_{it}) and applying a split function h() to make y_{it}^* a binary value:
    $$y_{it}^* = \text{h}(y_{it} + q_{it})$$
    b. Repeat steps 1a-c using y_{it}^* as the outcome until the change in the posterior log likelihood from the Bayesian GLMM is less than a specified tolerance value.

In this algorithm, in the step 1c, the predicted probabilities from the RF model and the GLMM model are averaged because the goal of the algorithm is to combine population-level effects arising from the RF with cluster-level effects arising from the GLMM. Predictions for observations included within the model development dataset are made using the forest (population-level) and GLMM (observation-level) components. For

observations not included within the model development dataset, predictions are made using the forest (population-level) component only.

We propose different split functions (denoted $\mathrm{h}(y_{it} + q_{it})$) which may be used to create the new iteration of the binary target outcome (y_{it}^{*}) in the BiMM forest method. We use a function of $y_{it} + q_{it}$ to update the target outcome to account for both the original outcome and the average predicted probability from the RF and GLMM models for the specific observation *t* within the cluster *i*. Since y_{it} is a binary value, it is either 0 or 1, and q_{it} is a probability which is between 0 and 1. Therefore, the value of $y_{it} + q_{it}$ is between 0 and 2. We present three options for the split function which may be employed based on the overall goal of the prediction model and note that users may define alternative split functions. The first split function maximizes sensitivity uses a threshold ($0 < k_1 < 1$) to update the target outcome:

$$\mathrm{h}_1(y_{it} + q_{it}) = \begin{cases} 1 \text{ if } y_{it} + q_{it} > k_1 \\ 0 \text{ otherwise} \end{cases}$$

Thus, using $\mathrm{h}_1(y_{it} + q_{it})$, binary outcomes of 0 can be updated to be 1, but outcomes of 1 cannot be updated to be 0, providing a mechanism for maximizing the sensitivity. Similarly, the second split function maximizes specificity by employing a threshold ($1 < k_2 < 2$) to update the target outcome:

$$\mathrm{h}_2(y_{it} + q_{it}) = \begin{cases} 0 \text{ if } y_{it} + q_{it} < k_2 \\ 1 \text{ otherwise} \end{cases}$$

Using $\mathrm{h}_2(y_{it} + q_{it})$, binary outcomes of 1 can be updated to be 0, but outcomes of 0 cannot be updated to be 1, providing a mechanism for maximizing the specificity. A

final, more general, split function which does not favor sensitivity or specificity updates the target outcome using the following method:

$$\mathrm{h}_3(y_{it} + q_{it}) = \begin{cases} 0 \text{ if } y_{it} + q_{it} < 0.5 \\ 1 \text{ if } y_{it} + q_{it} > 1.5 \\ 1 \text{ with probability } q_{it} \\ 0 \text{ with probability } 1 - q_{it} \end{cases}$$

Using $\mathrm{h}_3(y_{it} + q_{it})$, if the prediction from the current iteration of the BiMM method agrees with the original binary outcome (i.e. if $y_{it} + q_{it} < 0.5$ or if $y_{it} + q_{it} > 1.5$) then the target outcome is the same as the original binary outcome. Otherwise, the target outcome is updated to be 1 with probability q_{it}, and 0 with probability $1 - q_{it}$. Therefore, original values of 0 can be updated to 1 and original values of 1 can be updated to 0.

BiMM forests for this study are computed using R software version 3.1.2 [90]. RF models are implemented using the R package *randomForest* [52] with default settings. Bayesian GLMMs within the BiMM method are implemented using the R package *blme* [87, 88], again with all default settings. Thus, uninformative prior distributions are used for both fixed (Normal prior distribution) and random effects (Wishart prior distribution for the unstructured covariance matrix of clustered variables). However, alternative prior distributions may be applied if separation or convergence issues arise.

3.2.4 Simulation Study Design

To assess the predictive performance of the proposed BiMM forest method, we conduct a simulation study based on the real motivating dataset, the ALF Study Group registry. Similar to the simulation within Aim 1, we construct a dataset from which we sample simulation data by selecting all data from acetaminophen-induced ALF patients within the registry (N=1064) and imputing all missing predictor data using an imputation

method [95] for multilevel data to preserve the original correlation structure between predictor variables within the dataset. Thus, the simulated datasets contain 1064 patients with complete data for seven days (three fixed predictors and eight longitudinal predictors). We use three data generating processes for the fixed portion of the outcome: a tree structure, a linear structure, and a complex structure. Variables related to the outcome include INR, creatinine, and ventilator use, which is consistent with clinical literature [92, 96]. The other five longitudinal variables and the three fixed predictors are included within the simulation datasets as noise variables. The tree data generating process is depicted within Figure 3.1, which is read like a decision tree (i.e. begin at Node 0 and follow the arrow corresponding to the predictor variable values until a terminal node is reached). Nodes 1, 3 and 5 represent favorable outcome for the subject on the specific day, whereas Node 6 represents poor outcome for the subject on the specific day. The equation for the linear data generating process is:

$$\text{logit}(poor\ outcome_{it}) = -2.3 + 1.4 * ln(INR_{it}) + 0.6 * Creatinine_{it} + 2.1 * I(Ventilator_{it})$$

where $I(Ventilator_{it})$ is 1 if patient *i* is on a ventilator on the specific day *t*, and is 0 otherwise. Thus, high INR and creatinine and being on a ventilator are associated with higher likelihood of poor outcomes, consistent with clinical literature [96]. For the complex structure data generating processes, we develop outcomes based on five unique decision trees. Tree 1 generates outcome based on INR, creatinine and ventilator use; Tree 2 uses pressor use, creatinine and bilirubin; Tree 3 uses ventilator use, pressor use, and age; Tree 4 uses INR, creatinine and ventilator use; and Tree 5 uses bilirubin, ALT and ventilator use. Also, we include nuisance variables: sex, ethnicity and AST. The variable for age is included to derive the binary outcome, but is intentionally omitted

from model development within the simulation to represent an unmeasured predictor which is significantly related to outcome.

Small and large random effects are added to the fixed portion of the outcome to create a within-subject correlation structure. The small random effect is generated for each subject from a normal distribution centered at zero with standard deviation of 0.1, whereas the large random effect is generated for each subject from a normal distribution centered at zero with standard deviation of 0.5. To derive the observed outcome at every time point, the fixed portion (from the tree, linear, or complex data generating process) is added to the random effect, and a cut point is used to create the binary outcome. We used a threshold to create an unbalanced outcome, with approximately one-third of the observations having poor outcome and two-thirds having favorable outcome for each simulated dataset.

Using the simulated datasets described in the previous paragraphs based on the ALF registry, we compare the performance of several models: standard RF (ignoring clustering within the data), Bayesian GLMM, BiMM tree with one iteration, BiMM forest with one iteration (i.e. only Step 1 in the algorithm is performed), and BiMM forest with updating functions. We produce models for standard RF, BiMM forest with one iteration (denoted BiMM RF 1) and with multiple iterations (denoted BiMM RF H1 and BiMM RF H3 for the respective split functions $\mathrm{h}_1(y_{it} + q_{it})$ and $\mathrm{h}_3(y_{it} + q_{it})$) to assess if iterating between fixed and random effects results in increased prediction accuracy. We use $\mathrm{h}_1(y_{it} + q_{it})$ as our split function for updating the target outcome with a threshold (k_1) of 0.5 because clinicians often prefer to develop prediction models maximized for

sensitivity to identify patients at highest risk of poor outcomes. For comparison, we also compile BiMM forest using $\mathrm{h}_3(y_{it} + q_{it})$ with $k_1 = 0.5$ and $k_2 = 1.5$. We specify BiMM forest model convergence when the change in the posterior log likelihood between iterations is less than 0.1. All models are fit using all predictors in the data (i.e. both those associated with outcome and those that were noise variables). Models are compiled for 1000 simulation runs. Sample sizes (number of subjects) for training datasets used in model development are 100 and 500. All test datasets consist of 500 new subjects not included within the training dataset. The number of repeated outcome measurements in our simulation study are 2, 4 and 7. All simulations are conducted using R software version 3.1.2 [90].

3.2.5 Simulation Study Results

Since the main objective in this study is to develop methodology for predicting new observations, we assess the prediction (test set) accuracy of the models, defined as the number of correct predictions divided by the total number of predictions made. Median prediction accuracy is presented in Figure 3.4 for the sample size of 100 for the different scenarios. Overall, the BiMM forest with one iteration and standard RF have among the highest accuracy compared to BiMM tree, Bayesian GLMM and BiMM forest with updating functions employed. Note that the prediction accuracy statistics for the standard RF and BiMM forest with one iteration are identical because they use the exact same random forest model to make predictions for test datasets. For the small random effect and the linear data generating process, the BiMM tree performs slightly worse than competing models; however, BiMM forest with one iteration and BiMM tree have

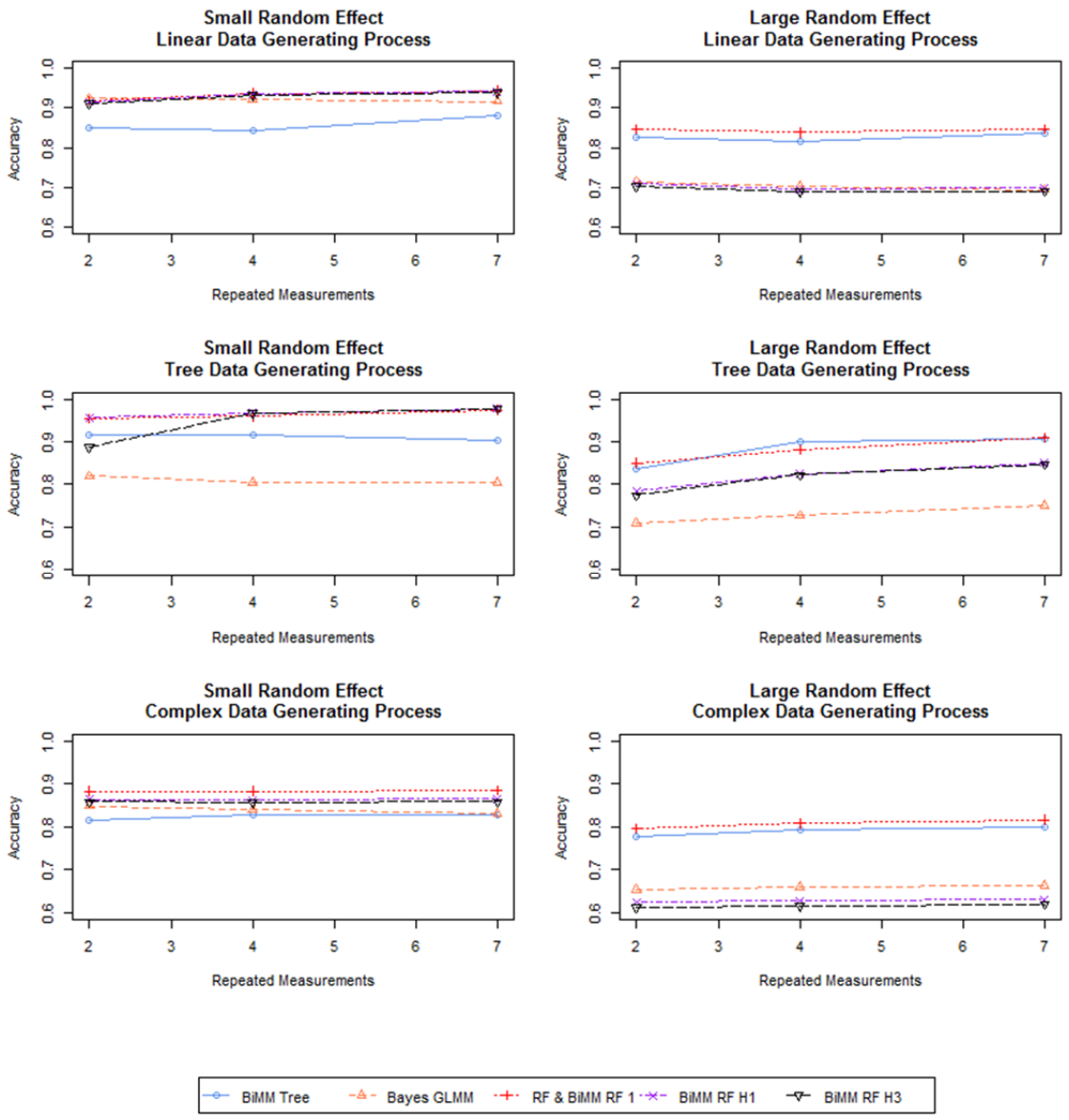


Figure 3.4: Simulated prediction (test set) accuracy of models for N=100 patients

higher prediction accuracy compared to the Bayesian GLMM and BiMM forest with multiple iterations when there is a large random effect (i.e. when there is a large amount of correlation within clusters). The Bayesian GLMM has the lowest prediction accuracy for the tree data generating process, regardless of the random effect size. In the tree data

generating scenario with a small random effect, the BiMM forest methods generally have similar prediction accuracy, which is slightly higher compared to the BiMM tree for 4 and 7 repeated measurements. The BiMM forest with one iteration and BiMM tree methods have similar prediction accuracy for the tree data generating process with a large random effect, which is slightly higher compared to the BiMM forest with multiple iterations.

For the complex data generating process with a small random effect, the BiMM forest with one iteration has the highest prediction accuracy, although all methods have fairly similar performance across the 2, 4 and 7 repeated measurements. With a large random effect and complex data generating process, the BiMM forest one iteration method has the highest prediction accuracy, followed closely behind by the BiMM tree with one iteration. The Bayesian GLMM and BiMM forest with multiple iterations perform similarly in this scenario, all with lower prediction accuracy compared to the BiMM tree and BiMM forest with one iteration. In general, the BiMM forest with one iteration and BiMM tree methods have the highest prediction accuracy, with the BiMM forest performing slightly better than the BiMM tree when there is a small amount of within-cluster correlation and when there is a complex data generating process. Similar results were obtained for the sample size of 500 (Appendix Figure 5), with slight increases in prediction accuracy for the BiMM forest method compared to the BiMM tree for the linear and complex data generating process with a large clustering effect.

Aside from the median prediction accuracy, we were also interested in describing the accuracy distributions for the simulated scenarios. Table 3.3 contains the median (interquartile range) estimates of prediction accuracy of the test dataset for each

Table 3.3: Median Prediction Accuracy (Interquartile Range) for Test set of Simulated Datasets

Model	Repeated Outcomes	N=100					
		Linear DGP		Tree DGP		Complex DGP	
		Small RE	Large RE	Small RE	Large RE	Small RE	Large RE
BiMM Tree	2	0.849 (0.833,0.868)	0.827 (0.776,0.852)	0.916 (0.897,0.927)	0.836 (0.782,0.902)	0.815 (0.794,0.854)	0.778 (0.734,0.808)
	4	0.845 (0.822,0.860)	0.815 (0.780,0.849)	0.917 (0.881,0.952)	0.901 (0.854,0.956)	0.830 (0.798,0.849)	0.792 (0.770,0.814)
	7	0.881 (0.869,0.891)	0.837 (0.806,0.864)	0.902 (0.892,0.913)	0.905 (0.862,0.920)	0.829 (0.808,0.839)	0.764 (0.749,0.774)
Bayesian GLMM	2	0.924 (0.914,0.933)	0.715 (0.700,0.730)	0.820 (0.805,0.835)	0.709 (0.691,0.723)	0.849 (0.835,0.860)	0.653 (0.639,0.668)
	4	0.922 (0.915,0.928)	0.704 (0.689,0.716)	0.805 (0.795,0.814)	0.728 (0.717,0.738)	0.840 (0.830,0.849)	0.660 (0.649,0.672)
	7	0.917 (0.912,0.921)	0.695 (0.682,0.707)	0.805 (0.796,0.812)	0.750 (0.741,0.760)	0.831 (0.823,0.838)	0.664 (0.655,0.672)
BiMM RF 1 & Standard RF	2	0.919 (0.907,0.928)	0.847 (0.827,0.864)	0.954 (0.939,0.965)	0.850 (0.819,0.882)	0.884 (0.875,0.892)	0.797 (0.772,0.817)
	4	0.937 (0.930,0.943)	0.839 (0.818,0.861)	0.960 (0.956,0.973)	0.881 (0.854,0.901)	0.884 (0.878,0.890)	0.808 (0.793,0.823)
	7	0.944 (0.940,0.948)	0.847 (0.827,0.867)	0.974 (0.967,0.977)	0.910 (0.883,0.928)	0.885 (0.880,0.891)	0.817 (0.805,0.828)
BiMM RF H1	2	0.915 (0.900,0.926)	0.710 (0.693,0.725)	0.957 (0.925,0.971)	0.785 (0.769,0.814)	0.864 (0.850,0.875)	0.625 (0.596,0.650)
	4	0.935 (0.928,0.942)	0.695 (0.678,0.711)	0.968 (0.965,0.972)	0.825 (0.811,0.836)	0.864 (0.854,0.872)	0.628 (0.608,0.647)
	7	0.942 (0.936,0.946)	0.699 (0.679,0.713)	0.977 (0.975,0.980)	0.850 (0.839,0.854)	0.866 (0.859,0.873)	0.631 (0.615,0.646)
BiMM RF H3	2	0.911 (0.891,0.924)	0.703 (0.675,0.721)	0.888 (0.879,0.896)	0.775 (0.752,0.804)	0.859 (0.832,0.872)	0.612 (0.571,0.642)
	4	0.933 (0.919,0.940)	0.689 (0.663,0.706)	0.968 (0.965,0.968)	0.823 (0.807,0.834)	0.856 (0.830,0.868)	0.615 (0.576,0.639)
	7	0.939 (0.931,0.945)	0.691 (0.663,0.710)	0.978 (0.974,0.981)	0.847 (0.831,0.853)	0.859 (0.825,0.869)	0.620 (0.586,0.639)

Model	Repeated Outcomes	N=500					
		Linear DGP		Tree DGP		Complex DGP	
		Small RE	Large RE	Small RE	Large RE	Small RE	Large RE
BiMM Tree	2	0.850 (0.840,0.863)	0.856 (0.840,0.886)	0.920 (0.915,0.926)	0.917 (0.907,0.925)	0.810 (0.793,0.860)	0.808 (0.790,0.853)
	4	0.849 (0.838,0.862)	0.841 (0.813,0.865)	0.952 (0.891,0.960)	0.953 (0.900,0.963)	0.839 (0.801,0.850)	0.817 (0.793,0.848)
	7	0.890 (0.884,0.895)	0.867 (0.843,0.890)	0.905 (0.897,0.913)	0.906 (0.858,0.913)	0.839 (0.834,0.844)	0.822 (0.803,0.838)
Bayesian GLMM	2	0.943 (0.938,0.948)	0.737 (0.723,0.748)	0.842 (0.833,0.851)	0.725 (0.714,0.737)	0.864 (0.857,0.871)	0.681 (0.669,0.693)
	4	0.932 (0.928,0.936)	0.721 (0.710,0.732)	0.814 (0.808,0.821)	0.736 (0.727,0.746)	0.850 (0.844,0.856)	0.677 (0.669,0.686)
	7	0.923 (0.919,0.926)	0.710 (0.699,0.721)	0.809 (0.803,0.814)	0.756 (0.748,0.765)	0.838 (0.833,0.844)	0.673 (0.667,0.680)
BiMM RF 1 & Standard RF	2	0.947 (0.942,0.952)	0.890 (0.878,0.900)	0.969 (0.964,0.974)	0.920 (0.906,0.933)	0.904 (0.898,0.910)	0.841 (0.829,0.854)
	4	0.947 (0.943,0.951)	0.888 (0.877,0.898)	0.974 (0.970,0.978)	0.937 (0.926,0.947)	0.903 (0.898,0.908)	0.845 (0.837,0.854)
	7	0.951 (0.948,0.954)	0.893 (0.883,0.902)	0.978 (0.975,0.981)	0.949 (0.940,0.956)	0.907 (0.902,0.911)	0.846 (0.838,0.853)
BiMM RF H1	2	0.944 (0.939,0.951)	0.731 (0.719,0.744)	0.970 (0.965,0.974)	0.826 (0.815,0.835)	0.880 (0.872,0.889)	0.646 (0.630,0.663)
	4	0.944 (0.939,0.948)	0.725 (0.714,0.736)	0.975 (0.971,0.979)	0.844 (0.834,0.853)	0.879 (0.872,0.885)	0.645 (0.632,0.656)
	7	0.945 (0.941,0.949)	0.727 (0.717,0.737)	0.979 (0.976,0.982)	0.858 (0.849,0.866)	0.883 (0.877,0.890)	0.642 (0.632,0.651)
BiMM RF H3	2	0.942 (0.932,0.949)	0.727 (0.710,0.740)	0.970 (0.964,0.974)	0.824 (0.812,0.834)	0.873 (0.855,0.884)	0.636 (0.607,0.655)
	4	0.941 (0.933,0.946)	0.721 (0.705,0.732)	0.975 (0.971,0.979)	0.843 (0.833,0.852)	0.871 (0.845,0.881)	0.635 (0.601,0.651)
	7	0.942 (0.934,0.947)	0.723 (0.708,0.733)	0.979 (0.976,0.982)	0.858 (0.846,0.866)	0.876 (0.837,0.885)	0.633 (0.604,0.646)

simulation scenario. Interquartile ranges of prediction accuracy for the models in the different scenarios are relatively tight around the median estimates, indicating that the distribution of prediction accuracy for models does not vary greatly over the simulation runs. Across 2, 4 and 7 repeated measurements for the models and scenarios, prediction accuracy is similar, except for the tree data generating process with a large random effect, where slight gains in accuracy are achieved with increasing number of repeated measurements. In general, the prediction accuracy estimates are similar for sample sizes of 100 and 500, with slight improvements in accuracy for BiMM tree and BiMM forest models with the larger sample size.

In addition to assessing the predictive accuracy of models, we present the difference between training and test accuracy for models in the simulated scenarios to measure the amount of overfitting in models for sample size of 100 (Figure 3.5). Within this plot, large values of the difference between the training and test datasets indicate that the accuracy of the training dataset is larger than the accuracy of the test dataset. Patterns are similar across the linear, tree and complex data generating processes in terms of the difference between training and test dataset accuracies. In general, for small random effects, all models have small differences between the training and test datasets, indicating that models do not overfit the data. When random effects are large, BiMM forest with multiple iterations have the largest difference in accuracy, indicating these models overfit the training data the most. In all three data generating processes with a large random effect, the BiMM forest with one iteration has the least amount of overfitting. These plots highlight the differences between the standard random forest and BiMM forest methods because while these two methods have identical test set

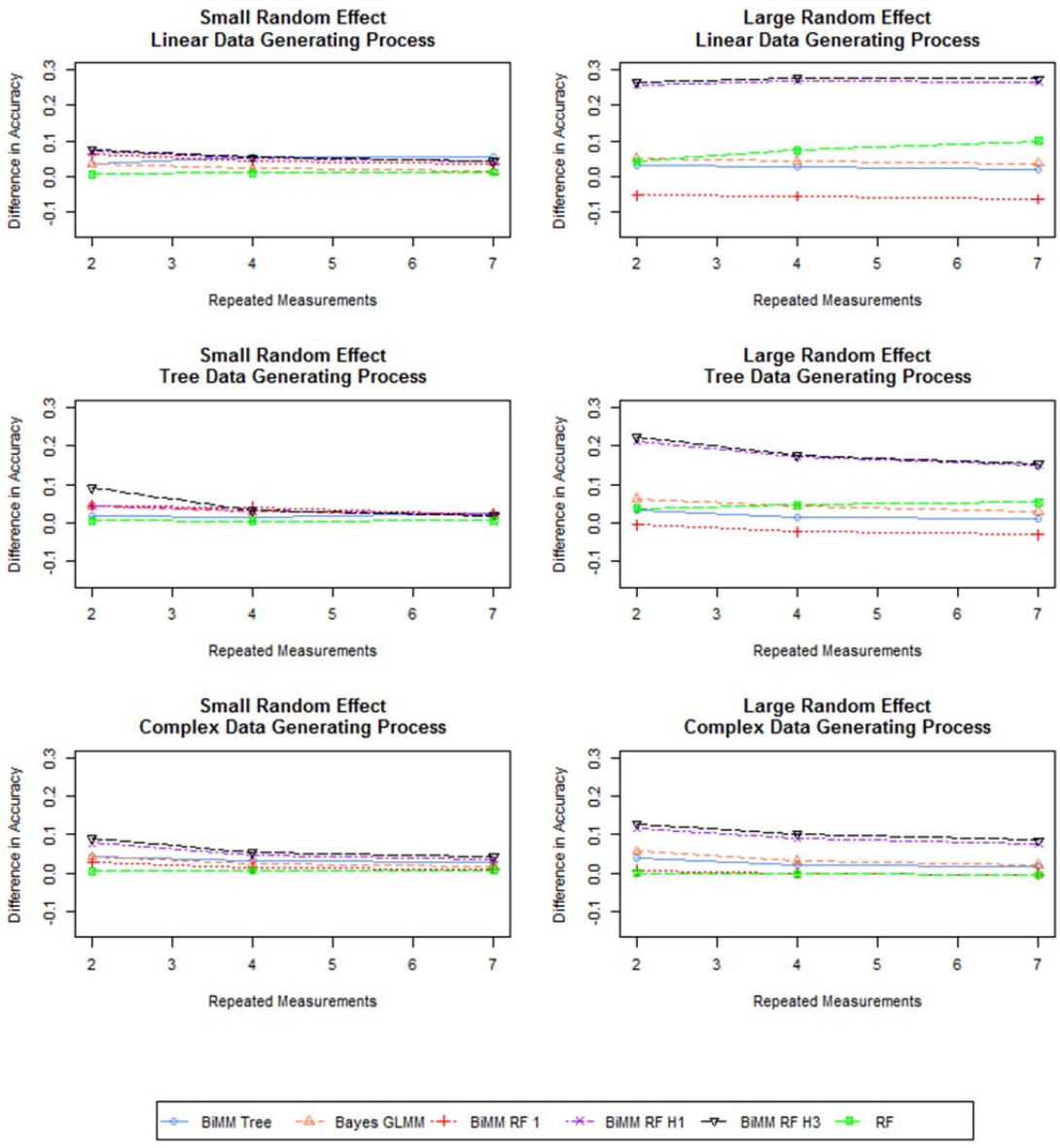


Figure 3.5: Simulated difference in training and test set accuracy of models for N=100 patients

predictions, they have different training dataset predictions. The standard random forest method tended to overfit the training data more than the BiMM forest method. Similar results are observed for the sample size of 500 (Appendix 1 Figure 6).

We also assess the rate of convergence of the BiMM forest models which employ updating functions (Table 3.4). Regardless of the data generating process and sample size, the number of iterations within the BiMM forest algorithm increases as the number of repeated outcomes increases. The number of iterations remained similar for different data generating processes with the same number of repeated measurements.

In general, BiMM forest with one iteration consistently has among the highest predictive accuracy across all simulated scenarios. Specifically, BiMM forest with one iteration and BiMM tree tend to perform best when there is a large amount of within-cluster correlation. BiMM forest with one iteration offers improvement in accuracy compared to BiMM tree in the scenarios where there is a small amount of within-cluster correlation or if there is a large amount of within-cluster correlation and the data is generated by a complex process. Additionally, BiMM forest with one iteration has the lowest amount of overfitting within our simulated scenarios with a large random effect. BiMM forest using updating functions tends to converge in 6 iterations for two repeated outcomes, 8 to 10 iterations for four repeated outcomes, and 12 iterations for seven repeated outcomes within the simulation study.

3.2.6 Discussion

Overall, the BiMM tree and BiMM forest framework may offer advantages compared to RF and Bayesian GLMMs. The main benefit of BiMM methods compared to standard RF is that it can account for clustered outcomes in modeling so that the assumption of independent observations is not violated. BiMM tree and BiMM forest do not require specification of nonlinear relationships or interaction terms and can be implemented for high dimensional datasets. A strength of BiMM methodology is that

Table 3.4: BiMM forest Median Number of Iterations (Interquartile Range) for Updating Function H1 and H3 models

N	Model	Repeated Outcomes	Linear DGP		Tree DGP		Complex DGP	
			Small RE	Large RE	Small RE	Large RE	Small RE	Large RE
100	BiMM RF H1	2	6 (4,9)	6 (4,7)	6 (4,9)	6 (4,7)	6 (4,9)	6 (4,7)
		4	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)
		7	12 (9,18)	11 (8,15)	12 (9,18)	11 (8,15)	12 (9,18)	11 (8,15)
	BiMM RF H3	2	6 (5,9)	6 (4,8)	6 (5,9)	6 (4,8)	6 (5,9)	6 (4,8)
		4	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)
		7	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)
500	BiMM RF H1	2	6 (4,9)	6 (4,7)	6 (4,9)	6 (4,7)	6 (4,9)	6 (4,7)
		4	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)
		7	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)
	BiMM RF H3	2	6 (5,9)	6 (4,8)	6 (5,9)	6 (4,8)	6 (5,9)	6 (4,8)
		4	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)	10 (7,15)	8 (6,11)
		7	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)	12 (9,18)	11 (8,16)

nonlinear forms of predictors and interactions between predictors are developed by the method based on the data. The computation time of GLMMs, BiMM tree and BiMM forest with one iteration are similar, yet the BiMM methods may offer higher prediction accuracy. BiMM forest may have higher prediction accuracy compared to BiMM tree for complex datasets. BiMM forest provides a flexible, data-driven predictive modeling framework for longitudinal and clustered binary outcomes.

Our simulation study demonstrates that BiMM forest may be advantageous compared to Bayesian GLMM when predictors are not linearly related to the outcome through the link function and when the random effect of the clustered variable is large. While Bayesian GLMM can be used to adjust for clustering within the data, model misspecification may reduce prediction accuracy (e.g. not including a significant interaction term or specifying an incorrect nonlinear relationship between predictor and outcome). One possible reason that the Bayesian GLMMs did not perform well for simulated data in this study is that some of the continuous predictor variables have skewed distributions, and extreme values may have adversely affected the GLMM parameter estimates.

The BiMM forest with one iteration generally has higher prediction accuracy compared to BiMM forest with more than one iteration for simulated data scenarios. The multiple iteration method BiMM forest using a split function produces overfitted models which do not predict well for test datasets if there is a large amount of within-cluster correlation. BiMM forest models which iterate between fixed and random effects have significantly higher computation time and offer minimal increases in prediction accuracy, suggesting that BiMM forest with one iteration may be sufficient. It is possible that real-

world datasets are more complex than our simulated datasets, though, so multiple iterations may be necessary in some situations. However, one may easily assess this by compiling BiMM forest models with one iteration and with multiple iterations and comparing the posterior log likelihoods.

BiMM forest with one iteration has the same predictive capabilities of standard RF for test datasets because predictions are made from the same RF model. However, BiMM forest with one iteration has an additional step of using the random forest results within a Bayesian GLMM to account for clustered outcomes. Standard RF does not adjust for correlated outcomes within clusters, so although predictions are identical for the two models, standard RF is not accurately representing the underlying correlation structure within the data. This is evident within the plots displaying the difference between the training and test dataset accuracy for simulated scenarios with large within-cluster correlations, in which the standard RF tends to overfit the training data whereas the BiMM forest did not.

It is somewhat surprising that the predictive performance of BiMM tree and BiMM forest with one iteration are similar for the linear and tree data generating processes when there is a large clustering effect. However, the BiMM forest with one iteration has higher prediction accuracy than the BiMM tree for simulated scenarios with a small amount of within-cluster correlation. Additionally, the BiMM forest with one iteration had slightly higher prediction accuracy compared to the BiMM tree for the complex data generating process, which makes sense because this data was generated using a forest structure. In almost all of the simulated scenarios, BiMM forest has

substantially higher prediction accuracy compared to GLMM, the standard method for clustered binary outcomes.

The main objective of this study is to provide a RF framework for developing prediction models for clustered and longitudinal binary outcomes. BiMM forest methodology offers comparable or higher prediction accuracy to other models and may be considered an alternative to using GLMM for complex datasets. Future work could investigate methods for assessing variable importance within clusters, as well as methods for performing variable selection to develop parsimonious prediction models.

An R package for implementing BiMM forest methodology is being developed and will be available on the Comprehensive R Archive Network. R code to implement BiMM forest methodology is available in Appendix 2.

3.3 Specific Aim 3: **To develop a prediction model for daily outcomes of acetaminophen-induced acute liver failure patients**

3.3.1 Introduction

Acetaminophen (APAP) is the most common cause of acute liver failure (ALF) in Europe and North America [100, 101]. Injury and recovery follow a hyper-acute pattern, in which maximum hepatocyte destruction is complete by 72 hours following a one-time ingestion, with potential recovery equally swift. Despite reasonable post-transplant outcomes, liver transplantation (LT) for acetaminophen-induced acute liver failure (APAP-ALF) often presents significant challenges in management due to the rapidity and severity of illness, the potential for recovery without LT and the presence of complex psychosocial issues in most patients [102, 103]. Data from the NIH-funded Acute Liver Failure Study Group (ALFSG) shows that approximately 25% of APAP patients are listed for LT and less than 10% receive LT [104]. Current data suggest that APAP recovery for many patients is determined by 3-4 days following onset of illness [105]. With advances in intensive care unit (ICU) management such as continuous renal replacement therapy (RRT) and neuroprotective strategies, many patients who would otherwise have succumbed may remain alive for longer periods well beyond the initial insult [93, 94].

Several prognosis models are available for predicting survival in ALF, but few are developed using daily measures of outcome with post-admission data. While the King's College Criteria (KCC) [27] has been validated on admission, prediction of outcome at later time points appears less accurate [106] when hepatic dysfunction would be

characterized primarily by immunosuppression rather than multi-organ failure [93]. Numerous studies have shown relatively poor sensitivity of the KCC APAP criteria, ranging between 25% and 76%, meaning that many patients who did not meet criteria had poor outcomes during the incident hospitalization [101, 107-109]. Conversely, low specificity implies that some patients may have a good outcome despite meeting KCC and potentially could undergo unnecessary LT [108, 110]. Aside from KCC, the Acute Liver Failure Study Group Prognostic Index (ALFSG-PI) [96] has been evaluated for predicting 21-day transplant-free survival at admission and post-admission time points. However, a limitation of the ALFSG-PI is that only admission data are used to develop the model rather than longitudinal data. Speiser et al. [111] provide post-admission prognosis models using decision tree methodology, but these are developed using summary statistics from post-admission data rather than including data from each day. The use of summary statistics across multiple days of data for patients may have resulted in a loss of information, so models may not achieve optimal accuracy.

The primary aim of this study is to explore the use of binary mixed model (BiMM) tree and BiMM forest methodologies to determine prognosis for use at admission (early) and post-admission (days 2-7) in APAP-ALF patients. BiMM tree (Aim 1) provides a decision tree framework for developing prediction models for longitudinal outcomes using binary splits on variables which can be read like a flow chart. Decision trees are popular in diverse medical fields [2, 3], and BiMM tree models offer an intuitive method for predicting longitudinal measures of outcome, using processes familiar to clinicians (e.g. “high” versus “low” values of a predictor). Though decision tree methods such as BiMM tree provide a simple, intuitive method for

obtaining predictions, accuracy of models can often be improved using an ensemble, or collection, of decision trees (e.g. random forest (RF) [4]). Therefore, we also employ BiMM forest (Aim 2), a RF method for developing prediction models for longitudinal binary outcomes. We hypothesize that BiMM models will have similar or modestly higher predictive accuracy, sensitivity, and specificity compared to traditional generalized linear mixed models (GLMMs).

3.3.2 Materials and Methods

Study Design

Data from 1042 APAP-ALF patients enrolled within the ALFSG database from January 1998 to February 2016 (25 sites overall, 14 currently active; see acknowledgements) are used in this retrospective cohort study. The authors' Institutional Review Board (IRB)/Health research ethics boards of all enrolling US ALFSG sites have approved all research and all clinical investigation has been conducted according to the principles expressed in the Declaration of Helsinki. Consent/assent is obtained from all patients/their next of kin for collection of data in the ALFSG registry. Patient records are anonymized and de-identified prior to use in this analysis. Participants who are medically competent provide written informed consent to participate in this study. In cases when patients are unable to provide written consent (critical illness, hepatic encephalopathy) written assent is obtained by the next of kin. Upon regaining capacity, patients are given the option to withdraw written consent. In those cases, data are not included in the registry. Health research ethics boards/ Institutional review boards at all sites of the ALFSG have approved this consent procedure.

Participants

ALFSG registry eligibility criteria include: a) hepatic encephalopathy of any degree; b) evidence of moderately severe coagulopathy (international normalized ratio (INR) greater than or equal to 1.5); c) presumed acute illness onset of less than 26 weeks; and d) no cirrhosis [112]. For this study, only patients within the ALFSG registry with primary diagnoses of APAP determined by the site investigator are eligible.

Operational Definitions

Hepatic encephalopathy (HE) grade is defined using the West Haven Criteria (summarized); grade 1: any alteration in mentation, grade 2: being somnolent or obtunded but easily rousable or presence of asterixis, grade 3: being rousable with difficulty and, grade 4: unresponsive to deep pain [113]. In this study we defined 'low coma grade' as grade 1 or 2 and 'high coma grade' as grade 3 or 4. For evaluating the predictive performance of the models, specificity is the proportion of correctly predicted poor outcomes and sensitivity is the proportion of correctly predicted good outcomes.

Variables

The primary outcome of interest is binary: low coma grade versus high coma grade, which is collected daily for the first seven days following study admission until patients die, receive a LT, or are discharged/transferred from the hospital. We define 'good outcome' as low coma grade, and 'poor outcome' as high coma grade. We consider several variables collected at one time point, as well as daily variables, for developing prediction models. Variables collected only on admission include gender, ethnicity and age. Daily variables considered for prediction modeling include AST, ALT, phosphate, lactate, platelets, bilirubin, ammonia, creatinine, INR, pressor use, and RRT.

Statistical Methods

All models are constructed using a training dataset (525 patients and 2253 observations) and are assessed using a test dataset (517 patients and 2208 observations). Training and test data are randomly split such that daily measurements for each patient appear only in one of the datasets. Analyses are completed using SAS Version 9.3 (SAS Institute, Cary, NC) and R software [114]. Patient characteristics are presented as mean (standard deviation (SD)) or N percent and compared using t-tests and binomial tests using the R package *tableone* [115]. P-values adjusted for longitudinal measures within the daily dataset are computed using standard GLMM methodology. We develop the following models: classification and regression tree (CART), RF, frequentist GLMM, Bayesian GLMM, BiMM tree, and BiMM forest. We note that the first two methods (CART and RF) do not adjust for longitudinal outcomes, whereas the other methods account for longitudinal outcomes. R packages employed to develop models include: *rpart* [91], *randomForest* [52], *lme4* [89] and *blme* [87]. Because some of the methods should be employed with a complete dataset (i.e. RF, GLMMs and BiMM forest), we also develop models using an imputed dataset. For simplicity, we use the rfImpute function within the *randomForest* R package to impute missing predictor values [52]. Models are assessed in terms of overall accuracy, sensitivity and specificity for training and test datasets using binomial estimates and confidence intervals. Area under the receiver operating curve (AUROC) is determined using the R package *ROCR* [116].

Our primary focus of this study is to compare novel BiMM tree and BiMM forest to traditional methods (GLMMs and standard tree/forest models). BiMM tree (Aim 1) and BiMM forest (Aim 2) are machine learning algorithms which may be applied to

develop accurate prediction models for complex datasets (e.g. containing many predictors, interactions among predictors, and predictors with extreme values) which have clustered and longitudinal endpoints. Statistical models should account for data of this structure because values of a variable collected for a patient at many time points are correlated, creating groups called clusters. In addition to having clustered and longitudinal outcomes, some datasets (e.g. ALFSG registry data) contain complexities which make developing prediction models challenging using traditional methodology. For example, GLMMs may be suboptimal if datasets contain nonlinear predictors of outcome or complex interactions among predictors which are not specified correctly. BiMM tree and BiMM forest provide data-driven methods for developing prediction models which do not require the user to specify nonlinear associations or interaction terms. Compared to standard CART and RF, BiMM methods are more appropriate for longitudinal data since they incorporate clustering effects. Based on data simulations (Aim 2), BiMM forest may provide higher accuracy compared to BiMM tree; however, BiMM tree is simpler to use in practice than BiMM forest, which requires an application to obtain predictions. In this study, we compare accuracy and other performance statistics for traditional mixed models, novel BiMM models and standard CART and RF methodology.

3.3.3 Results

Patient Characteristics

Demographic and clinical characteristics of patients are displayed in Table 3.5 by outcome status for admission and all daily data. Of the 1042 patients, the mean age is

Table 3.5: Patient Characteristics: Mean (SD) or N (%)

Type of Variable	Variable	Admission Data (N=1042 observations)				All Daily Data (T=4461 observations)			
		N	Poor Outcome Mean (SD) or Number (%)	Good Outcome Mean (SD) or Number (%)	P-value	T	Poor Outcome Mean (SD) or Number (%)	Good Outcome Mean (SD) or Number (%)	P-value
Collected at one time	Female	1042	443 (81.7)	348 (69.6)	<0.001				
	Non-Hispanic	1040	504 (93.3)	472 (94.4)	0.558				
	Age	1042	39.11 (12.68)	36.19 (12.74)	<0.001				
Collected at days 1-7	ALT	1029	4049.11 (3149.08)	5173.12 (3921.35)	<0.001	4292	2359.43 (2538.94)	2687.80 (3078.94)	0.087
	AST	1029	5013.51 (5015.01)	5703.38 (5634.56)	0.038	4319	2230.88 (3621.64)	2126.72 (3869.71)	0.427
	Bilirubin	1026	5.64 (4.94)	5.02 (4.88)	0.043	4295	8.27 (6.48)	6.38 (6.00)	<0.001
	Creatinine	1036	3.02 (7.86)	2.12 (1.91)	0.013	4360	2.70 (4.24)	2.31 (2.29)	<0.001
	Phosphate	921	3.15 (2.13)	2.72 (1.71)	0.001	2499	3.32 (4.43)	3.40 (7.41)	0.587
	Lactate	191	1.28 (3.07)	2.00 (3.70)	0.149	816	2.31 (3.89)	4.43 (4.48)	<0.001
	Platelets	1029	136.66 (89.19)	208.34 (96.37)	0.235	4316	104.05 (69.76)	137.15 (68.46)	<0.001
	Ammonia	393	149.43 (117.36)	122.19 (134.28)	0.033	1088	115.95 (90.95)	93.40 (99.85)	0.002
	INR	1019	3.67 (2.77)	3.76 (2.59)	0.621	4236	3.15 (15.19)	2.57 (2.90)	0.012
	MV	1042	471 (86.9)	86 (17.2)	<0.001	4456	2057 (88.4)	450 (21.1)	<0.001
	Pressors	1042	188 (34.7)	46 (9.2)	<0.001	4456	734 (31.5)	147 (6.9)	<0.001
	RRT	1042	134 (24.7)	44 (8.8)	<0.001	4456	663 (28.5)	243 (11.4)	<0.001

significantly higher for patients with poor outcome compared to patients with good outcome on admission (39 versus 36). There are significantly more females with poor outcome compared to good outcome (82% versus 70%). There are no significant differences in ethnicity between the outcome groups. Upon study admission day, patients with poor outcome have significantly lower ALT, lower AST, higher bilirubin, higher creatinine, higher phosphate, and higher ammonia compared to those with good outcome. The poor outcome group also has a higher percentage of patients being treated with MV, pressors, and RRT. In total, there are 4461 observations of data collected for the 1042 patients. On days 1-7 there are respectively 1042, 875, 704, 571, 488, 423, and 358 patients with data available. Patients have an average of approximately four days of data. The right panel of Table 3.5 displays clinical characteristics of patients, with p-values adjusted for repeated measurements. Aside from ALT, AST and phosphate, all predictors differ significantly by outcome group.

Patients are randomly assigned to be in either the training dataset or test dataset for model development, regardless of the number of daily measurements of data. Table 3.6 displays demographic and clinical characteristics of patients for each dataset. There are no significant differences of predictor variables between the test and training datasets, aside from RRT use, which is slightly higher in the test dataset compared to the training dataset.

Original Dataset Models

We develop CART, Frequentist GLMM, Bayesian GLMM, and BiMM tree models using the original training dataset. RF and BiMM forest require all missing data to be imputed prior to modeling, so these models could not be developed using the

Table 3.6: Comparing Training and Test Datasets

Type of Variable	Variable	All Daily Data (T=4461 observations from N=1042 patients)			
		T	Training Data Mean (SD) or Number (%)	Test Data Mean (SD) or Number (%)	P-value
Collected at one time	Female	1042	399 (76.0)	392 (75.8)	1.000
	Non-Hispanic	1040	490 (93.5)	486 (94.2)	0.746
	Age	1042	38.17 (12.78)	37.24 (12.79)	0.242
Collected at days 1-7	ALT	4292	2589.47 (2846.02)	2443.30 (2782.92)	0.089
	AST	4319	2298.40 (3989.40)	2061.22 (3470.36)	0.037
	Bilirubin	4295	7.20 (6.02)	7.52 (6.61)	0.101
	Creatinine	4360	2.54 (3.76)	2.49 (3.13)	0.653
	Phosphate	2499	3.51 (7.79)	3.21 (3.48)	0.221
	Lactate	816	3.53 (4.36)	3.44 (4.36)	0.755
	Platelets	4316	113.17 (69.70)	126.30 (67.82)	0.365
	Ammonia	1088	107.52 (90.83)	104.23 (99.92)	0.572
	INR	4236	3.03 (15.54)	2.72 (2.99)	0.367
	MV	4456	1281 (56.9)	1226 (55.6)	0.415
	Pressors	4456	446 (19.8)	435 (19.7)	0.985
	RRT	4456	414 (18.4)	492 (22.3)	0.001
	Poor Outcome	4461	1086 (48.2)	1043 (47.2)	0.538

original (unimputed) dataset which contains missing predictor values. Diagrams for the CART and BiMM tree are displayed within Figure 3.6. If the logic statement is true, then one follows the left branch, and if the logic statement is false, one follows the right branch. For binary predictors (e.g. pressors), 1 indicates that the patient is on the treatment and 0 indicates that the patient is not on the treatment. The CART uses seven variables and eight nodes in order to obtain predictions of outcome, whereas the BiMM tree uses three variables and three nodes. The models are identical up until the fourth node, in which the BiMM tree has a terminal node, but the CART continues to use AST and additional variables.

Accuracy, sensitivity and specificity for the original training and test dataset models are presented within Table 3.7. The BiMM tree model has the highest training

Figure 3.6: Original Dataset Tree Diagrams (1=low coma grade/good outcome, 0=high coma grade/bad outcome)

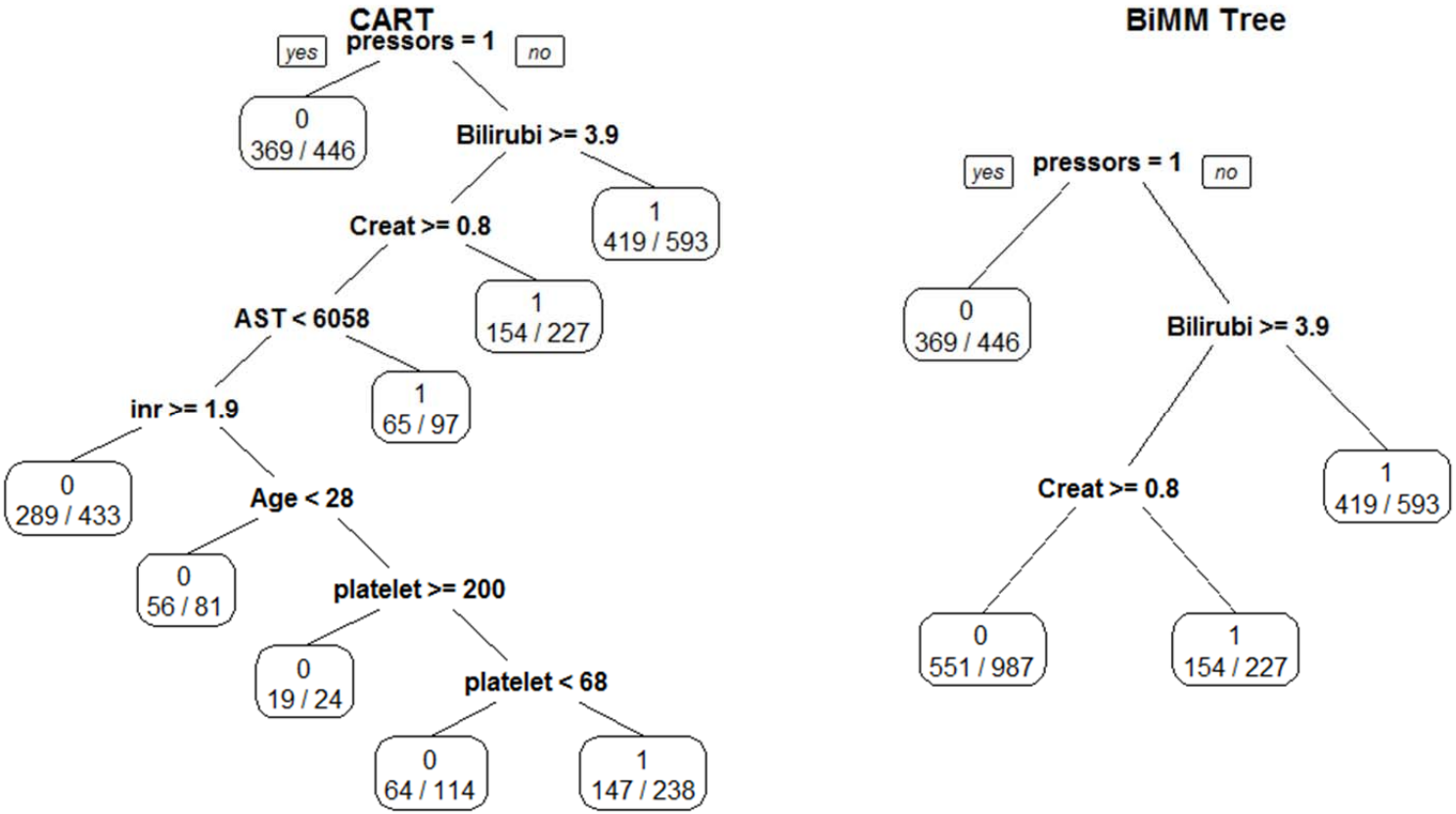


Table 3.7: Accuracy Statistics for Models

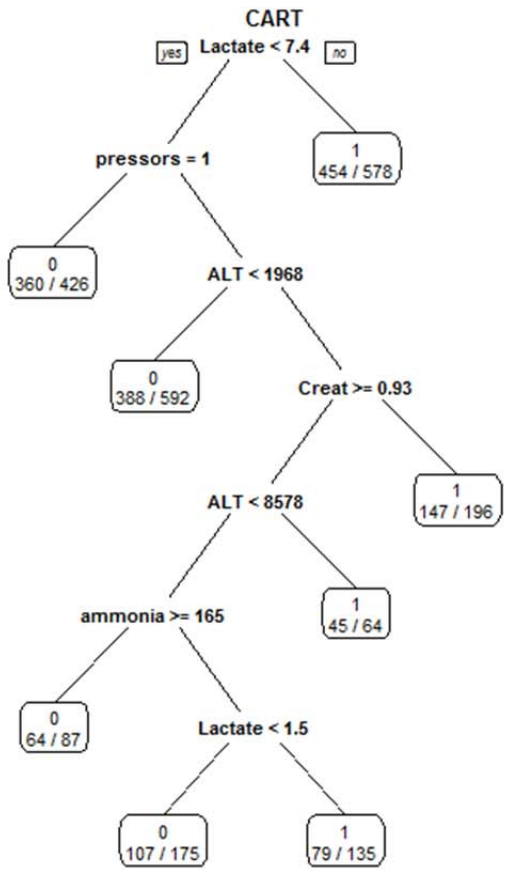
Original Dataset								
	Training Data T=2253 observations				Test Data T=2208 observations			
Method	T	Accuracy	Sensitivity	Specificity	M	Accuracy	Sensitivity	Specificity
CART	2253	0.702 (0.683,0.721)	0.683 (0.655,0.710)	0.723 (0.695,0.749)	2208	0.639 (0.619,0.660)	0.613 (0.584,0.641)	0.669 (0.640,0.698)
Frequentist GLMM	127	0.417 (0.330,0.508)	0.312 (0.211,0.427)	0.580 (0.432,0.718)	138	0.630 (0.544,0.711)	0.551 (0.426,0.671)	0.710 (0.588,0.813)
Bayesian GLMM	127	0.417 (0.330,0.508)	0.312 (0.211,0.427)	0.580 (0.432,0.718)	138	0.638 (0.552,0.718)	0.536 (0.412,0.657)	0.739 (0.619,0.837)
BiMM Tree	2253	0.907 (0.894,0.918)	1.000 (0.997,1.000)	0.820 (0.797,0.842)	2208	0.630 (0.610,0.651)	0.530 (0.499,0.561)	0.720 (0.693,0.746)
Imputed Dataset								
	Training Data T=2253 observations				Test Data T=2208 observations			
Method	T	Accuracy	Sensitivity	Specificity	T	Accuracy	Sensitivity	Specificity
CART	2253	0.730 (0.711,0.748)	0.787 (0.763,0.811)	0.668 (0.639,0.696)	2208	0.653 (0.633,0.673)	0.762 (0.737,0.786)	0.531 (0.500,0.562)
RF	2253	0.757 (0.739,0.775)	0.799 (0.775,0.822)	0.712 (0.684,0.739)	2208	0.688 (0.668,0.707)	0.743 (0.717,0.768)	0.626 (0.596,0.656)
Frequentist GLMM	2253	0.869 (0.854,0.882)	0.886 (0.866,0.904)	0.850 (0.827,0.871)	2208	0.686 (0.666,0.705)	0.724 (0.697,0.749)	0.643 (0.613,0.672)
Bayesian GLMM	2253	0.868 (0.853,0.881)	0.888 (0.868,0.905)	0.846 (0.823,0.867)	2208	0.686 (0.666,0.705)	0.724 (0.697,0.749)	0.644 (0.614,0.673)
BiMM Tree	2253	0.920 (0.908,0.931)	1.000 (0.997,1.000)	0.845 (0.823,0.865)	2208	0.653 (0.632,0.672)	0.669 (0.640,0.698)	0.638 (0.609,0.655)
BiMM forest	2253	0.872 (0.857,0.855)	0.868 (0.848,0.887)	0.876 (0.854,0.895)	2208	0.688 (0.668,0.707)	0.743 (0.717,0.768)	0.626 (0.596,0.656)

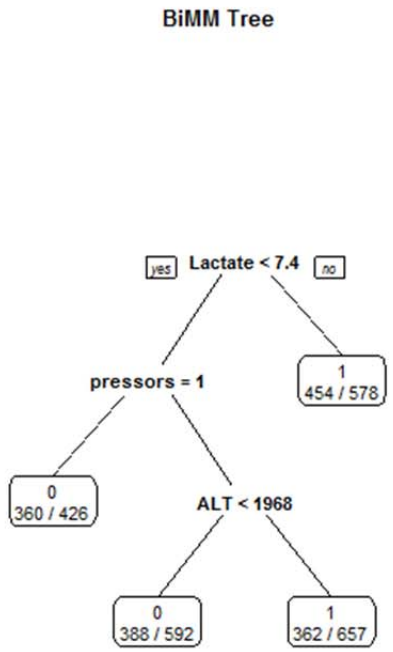
dataset accuracy compared to other models, along with the highest sensitivity and specificity. The Frequentist and Bayesian GLMM models make identical predictions for the training dataset. For the test dataset, all models have similar prediction accuracy of approximately 63%, though the breakdown of sensitivity and specificity is different comparing the models. The GLMMs and BiMM tree have slightly higher specificity compared to CART, which balance out the sensitivity and specificity more than the other models. A drawback of the GLMM models is that only observations with non-missing values for all variables could be used in modeling, so that a substantial portion of outcomes could not be obtained. The CART and BiMM tree methods can handle missing predictor data, so predictions are obtained for all observations within the original, unimputed training and test datasets.

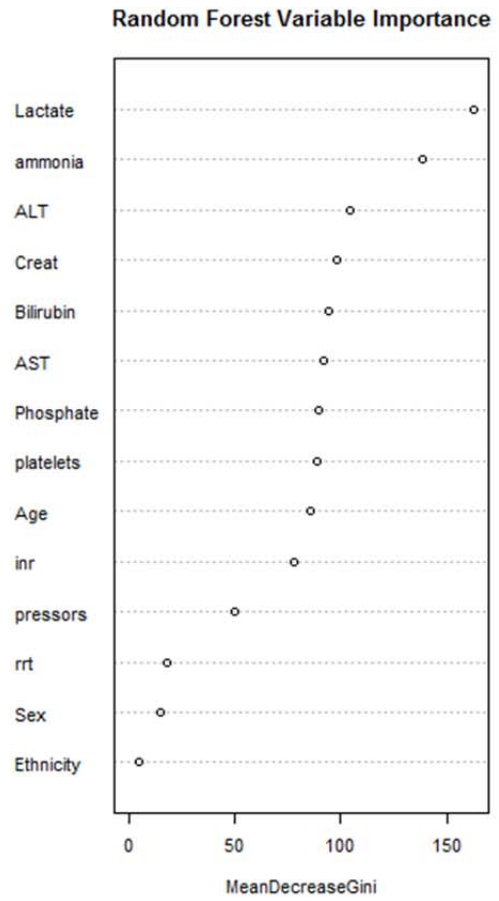
Imputed Dataset Models

In order to compare all models, we use an imputed dataset to predict daily outcomes of ALF patients. Figure 3.7 displays the CART and BiMM tree models, along with the variable importance plot from the RF. The CART and BiMM tree models look fairly similar, though the CART includes four additional nodes compared to the BiMM tree. Again, the CART model uses more predictors compared to the BiMM tree, which uses only three predictors. Within the RF variable importance plot, the most important predictors appear at the top and the least important predictors appear at the bottom. The RF identifies lactate as the most important predictor of daily outcome, followed by ammonia and ALT. The least important predictors of outcome are sex and ethnicity, consistent with clinical literature. Partial dependence plots are examined to assess the relationship between important predictors and outcome. Lactate greater than 6 mmol/L

Figure 3.7: Imputed Dataset Diagrams (1=low coma grade/good outcome, 0=high coma grade/bad outcome)







and ALT greater than 5000 IU/L are associated with higher odds of poor outcome (Figure 3.8).

Accuracy, sensitivity and specificity for the imputed dataset models are presented within Table 3.7. Similar to the original dataset results, the BiMM tree model has the highest training dataset accuracy compared to other models, along with the highest sensitivity. The Frequentist and Bayesian GLMM models make very similar predictions for the training dataset. Models which adjusted for longitudinal outcomes (i.e. GLMMs and BiMM methods) have higher performance statistics for the training dataset compared to models which did not adjust for longitudinal outcomes (i.e. CART and RF). For the test dataset, the standard RF, GLMMS, and BiMM forest have similar prediction accuracy of approximately 69%. All models have higher sensitivity than specificity for the test dataset.

Area Under the Receiver Operating Curve

In addition to comparing accuracy, sensitivity and specificity, we compare models for original data and imputed training and test datasets using ROC plots (Figure 3.9). For the original training dataset, the BiMM tree model clearly has the best AUC, which was 0.907, followed by CART with 0.735 and the GLMM methods with 0.417. Thus, the BiMM tree has the best model fit for the complete training dataset. For the imputed training dataset, the BiMM forest has the highest AUC (0.952), followed closely behind by Frequentist GLMM (0.941), Bayesian GLMM (0.940) and BiMM tree (0.921). RF and CART have lower AUC (0.829 and 0.770 respectively) compared to the other methods, which adjust for longitudinal outcomes. Overall, the BiMM forest has the highest AUC, indicating best model fit for the imputed training dataset.

Figure 3.8: Partial Dependence Plots for Lactate and ALT

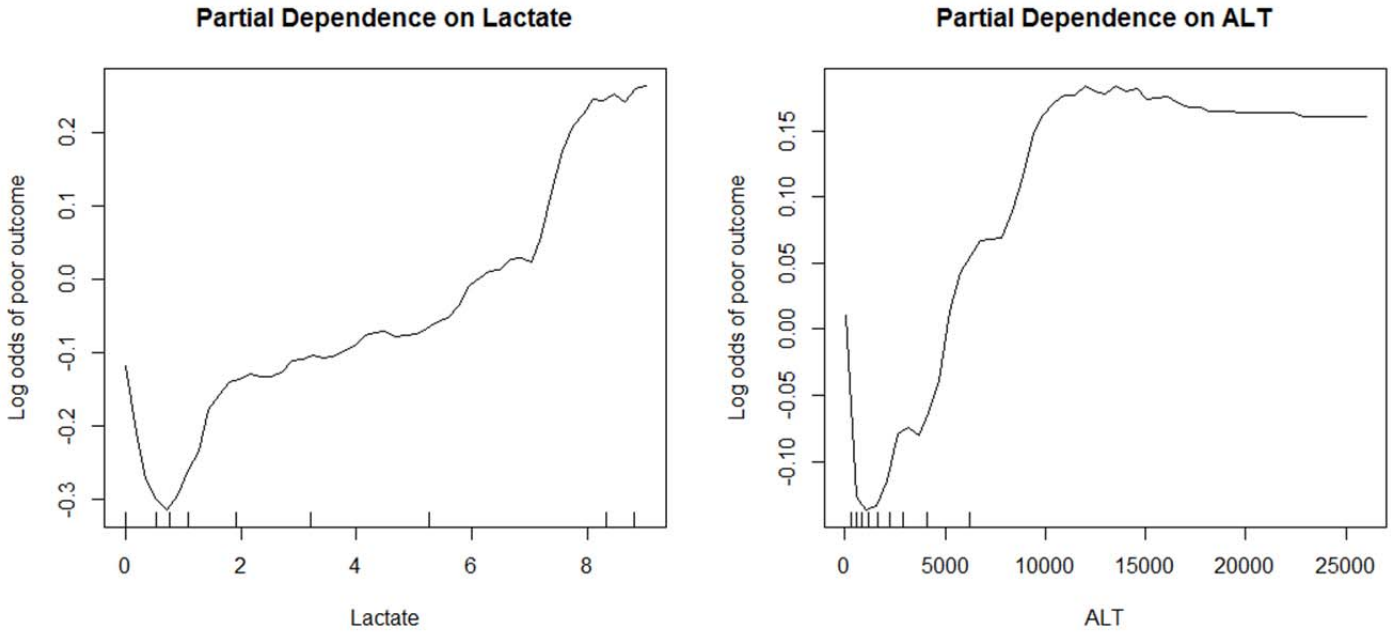
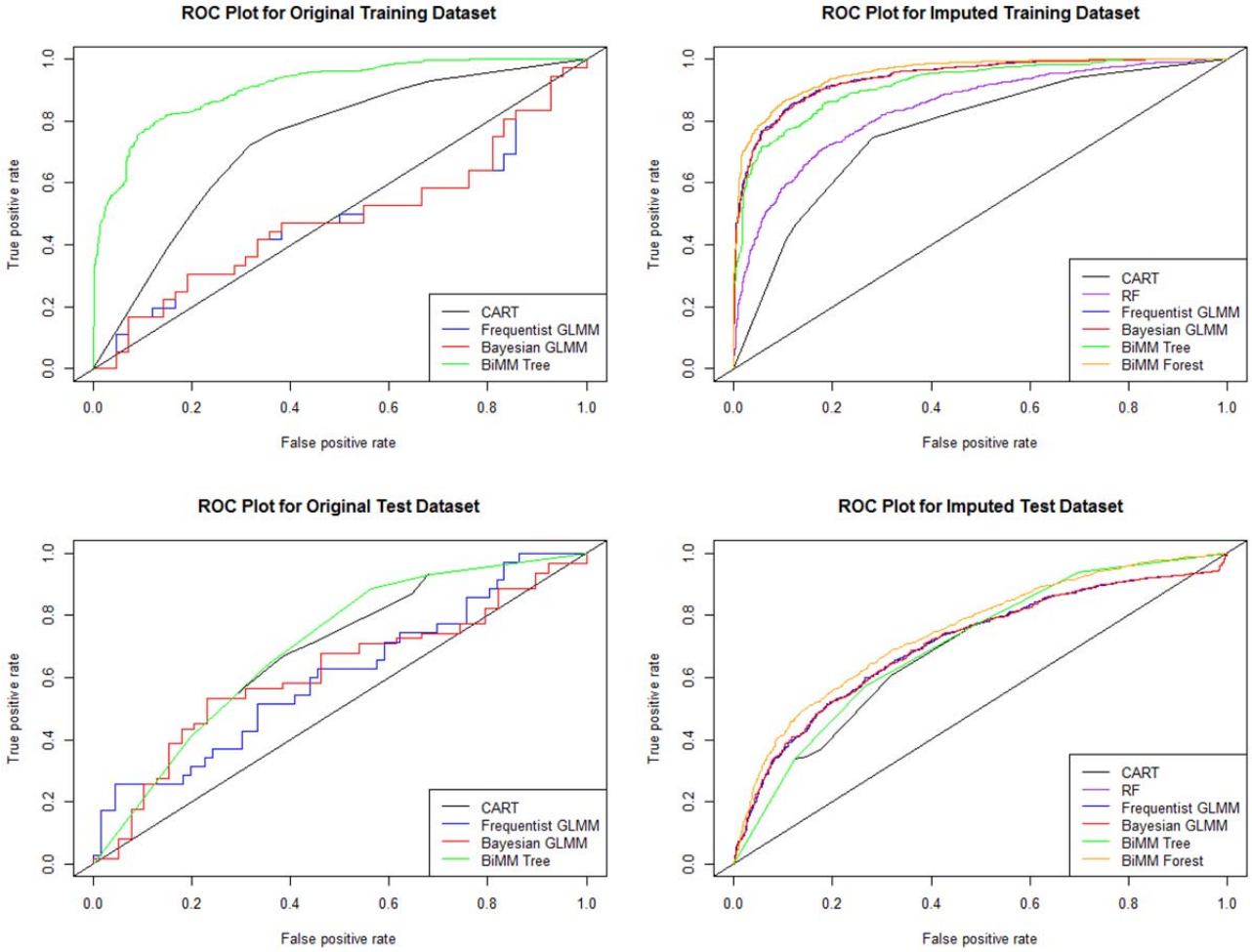


Figure 3.9: Receiver Operating Curve (ROC) Plots for Original Dataset Models and Imputed Dataset Models



For the original test dataset, the BiMM tree has the best AUC of 0.697, followed by CART with 0.682 and GLMM methods with 0.603. For the imputed test dataset, the BiMM forest and RF have the highest AUCs (0.749), followed by Frequentist GLMM (0.707), Bayesian GLMM (0.708), BiMM tree (0.707) and CART (0.698).

3.3.4 Discussion

Key Results

BiMM tree and BiMM forest models provide prediction models developed specifically for APAP-ALF patients which can be used at hospital admission and during in-patient hospitalization using daily outcomes. Models are developed using a training dataset and evaluated using a validation dataset for both original unimputed data and imputed data with missing values filled in. The prediction (test dataset) accuracy of the models with the original dataset are similar, around 63%. The BiMM tree has significantly higher training dataset accuracy compared to the standard CART, which does not account for clustered outcomes. The CART model is also more complex compared to the BiMM tree because it includes more predictor variables. Moreover, the CART splits may not be consistent with observations in clinical practice. For example, the fourth node, which splits AST < 6058 indicates that high AST is associated with high coma grade, may be counterintuitive because high AST is typically associated with poor survival. Additionally, AST is not typically a laboratory variable which is considered to be predictive of outcome based on current prediction models [27, 96]. On the other hand, the BiMM tree is clinically relevant, in which poor daily outcomes are associated with pressor use, high bilirubin, and high creatinine. A benefit of the BiMM tree method compared to the Frequentist GLMM and Bayesian GLMM is that all data, regardless of

missing values, can be evaluated, whereas only observations with non-missing values of all predictors could be included for the GLMMs.

To compare the prediction models using all observations, we additionally develop models using an imputed dataset. The standard CART and RF have the lowest training dataset accuracy, which makes sense because these models do not account for clustering in the outcomes. Consistent with the original dataset model, CART with the imputed dataset is more complex and disagrees with clinical observation because the first node identifies high lactate and high ALT as predictors of good outcome. The BiMM tree has a similar structure to the CART, which is not consistent with clinical observations of outcome. These models, which are contrary to clinical presentation of patients, highlight the danger of imputing missing values, particularly when there is a large percentage of missing data (e.g. lactate in this dataset which is missing 82% of values). Although it has a large amount of missing data, we considered lactate in prediction modeling because it has been identified as an important predictor of outcome in ALF [109]. While the CART and BiMM tree models do not make clinical sense, the BiMM forest is able to identify that high lactate and high ALT is associated with poor outcomes in APAP ALF patients. For the imputed dataset, the BiMM forest offers test set accuracy of 69%, training set accuracy of 87% and training set AUC of 0.952. The GLMM models have similar training and test dataset accuracy to the BiMM forest, with slightly lower AUCs.

Overall, the model which offers good predictive ability, is consistent with clinical practice, and is easy to use for obtaining predictions is the BiMM tree with the original dataset. While the prediction accuracy was slightly lower than the competing models, we believe it is the best model because it is simple to use in practice at the bedside for

predicting daily outcomes and it is consistent with what is seen in the clinical presentation of APAP-ALF patients. Compared to the GLMM models, the BiMM tree is easier to use because it requires only three variables within a user-friendly flow chart which does not require calculation or an application. Additionally, interpretation is simpler for the BiMM tree compared to the GLMMs because there is no need for understanding odds ratios or regression parameter estimates. We advise against using models developed with the imputed dataset because there is a substantial amount of missing data for some predictor variables, and resulting models may not be consistent with clinical practice. A benefit of the BiMM tree method is that it can handle missing data without the need for imputation. The BiMM forest is another viable option for daily predictions with clinically meaningful associations between predictors and outcome; however, an online application would need to be developed so that predictions could be obtained for new patients.

Comparison with Previous Studies

In this study, a mechanism for predicting daily outcomes during the first week of hospitalization is developed, which is novel since most prognostic models are constructed using hospital admission data and are not meant for use over time. Direct comparison of performance characteristics of models presented in this paper with current clinical prediction models is not possible because different outcome variables are used. In the current study, we use daily measures of high versus low coma grade, whereas most prediction models in the clinical literature use survival at a fixed time point. However, some similar clinical variables are used between models presented in this study and current clinical models. The BiMM tree developed with original data uses similar

predictors as other prognosis models in the clinical literature: KCC includes creatinine [27], model for end stage liver disease (MELD) includes creatinine and bilirubin [117], and ALFSG-PI includes pressor use and bilirubin [96]. CART models for the prediction of 21-day survival produced using aggregated post-admission data in a previous study use MELD, ventilator use, and lactate [111]; thus, decision tree models considering longitudinal data in the present study are quite different from those which do not use daily data. A key difference between these studies is the outcome was defined in different ways: the Speiser et. al. CART models use 21-day survival as the outcome of interest [111], and we use high versus low coma grade in this study. The BiMM tree for daily outcomes provides a method for obtaining predictions using a simple flow chart, whereas MELD and ALFSG-PI require the use of an application or calculation of scores. We use a daily measurement of outcome to develop prediction models rather than an outcome for a single time point because disease progression can change on a daily basis in the ALF setting. It is of clinical interest to obtain predictions of outcome which fluctuate over time rather than obtaining a single prediction for several weeks in advance to help clinicians develop management plans for ALF patients (e.g. whether to list a patient for a liver transplant).

Limitations

Though BiMM tree offers an alternative to current prognosis criteria, there are some limitations of this study which should be considered. First, data used to develop and assess new models are from the North-American ALFSG registry, so models may not be appropriate for populations elsewhere where transplant decisions may vary. Given the orphan status of ALF, it is difficult to find robust external datasets that have many

patients with serially collected clinical features. However, models are created using internal validation (test dataset) to address the issue of generalizability. Therefore, it is hypothesized that the BiMM tree model should perform well with other populations of APAP-ALF patients. The BiMM forest offers among the highest of prediction accuracies of the models; however, a limitation is that an online application is required for obtaining predictions in practice and interpretability is not as straight-forward as the decision tree models.

An important consideration in this study is that the models handle missing data in different ways, and it is challenging to compare all models regardless of missing data. A benefit of BiMM tree is that it can handle large amounts of missing data, whereas GLMMs and BiMM forest need complete data, which requires imputation of missing values. Because there is a large amount of missing data in some of the predictors (e.g. lactate and ammonia), models produced with the imputed data may not be appropriate. This is evident in the resulting models, which are not consistent with clinical observations in practice. This is the main reason we recommend use of the BiMM tree prediction model produced with the original unimputed dataset, even though it had slightly lower prediction accuracy than some of the other models. Given these limitations, it would be beneficial to use external datasets to validate the BiMM tree model developed in this study. Additionally, future incorporation of biomarkers of hepatic regeneration may improve upon models for prognosticating ALF.

3.3.5 Conclusions

Several models are produced for determining daily outcomes of APAP-ALF patients which can be used during the course of hospitalization. Offering a simple, accurate, and clinically consistent method for assessing high versus low coma grade, BiMM tree provides a prediction model developed for daily outcome measurements. Data from the ALFSG registry suggests that the BiMM tree prediction model offers good prediction accuracy (63%) and overall performance (AUC 0.907), but additional datasets should be used to externally validate these findings.

4 CONCLUSION

BiMM tree and BiMM forest methodology for clustered and longitudinal binary outcomes extends decision tree and RF to account for groupings within a dataset. ALF models to predict daily outcomes for acetaminophen patients are presented which will help clinicians determine the probability of poor outcomes throughout hospitalization. Innovative methods developed in this study are available using software for a commonly used statistical computing program, called R (Appendix 2). This offers researchers a freely available and easily implemented code which can be used to apply the novel methods. We are currently developing an R package which will be available on the comprehensive R archive network website. Many clinical datasets contain clustered binary endpoints, so providing rigorously assessed tree and forest mechanisms for this setting is a significant contribution to the field of biostatistics. The clustered CART and RF methods for continuous outcomes published by Hajjem [10, 12, 17] and Sela [15] are often cited in the literature, and the R software package by Sela has over 4,000 downloads in 2015. However, the RF method for continuous outcomes by Hajjem lacks freely available software for implementation, and the software for Sela's method for continuous outcomes only uses a single tree (not forest) framework. This suggests that our BiMM tree and forest methodology for modeling binary outcomes with accompanying software will be a significant contribution to machine learning literature.

Commonly in clinical settings, many variables are collected over time with the aim of developing a prediction model for binary outcomes. CART and RF are used in several clinical settings, and remain promising modeling tools which often offer high

accuracy rates and ease of interpretability for datasets where variables are not clustered [5, 47, 118]. The newly proposed BiMM tree and forest methodology provides accurate, efficient, interpretable and widely applicable prediction models for diverse clinical and other fields. Specific to the ALF setting, there is a need for bedside tools to aid in predicting daily clinical events of interest in an acute setting. The BiMM tree and forest models developed in this study have been evaluated by clinical experts to ensure plausibility of results and ease of use at the bedside. Model results can be used in practice to aid in clinical decisions (i.e. developing patient management plans and deciding to list for liver transplant) during the first week of hospitalization, which may improve outcomes for ALF patients.

There are several avenues of future work which can be investigated within the BiMM forest framework. Despite its improvement in accuracy of prediction models, the BiMM forest framework does not provide a method for determining the relative importance of predictors within clusters or a method for selecting optimal predictors to be used in a simpler model. In clinical prediction modeling, an interest is to understand the relationship between predictors and outcome, and to determine the most important predictors that should be included in a final, simpler model. Many complex datasets have hundreds of predictors (e.g. the ALFSG registry), so reducing the number of predictors within a model is an essential part of developing models which can be easily and readily used at the bedside. A future study could investigate a BiMM forest method to quantify the importance of predictors within clusters and to identify optimal predictors to be included within a model.

Aside from variable importance and variable selection, another avenue of future research within the BiMM forest framework is to develop a method for imputing missing data. Though standard RF offers an unbiased imputation method, there is no forest method available for imputing missing data in the presence of clustered and longitudinal outcomes. Many clinical datasets contain missing values of predictor variables, especially when data are repeatedly collected over time because patients may become lost to follow up or may not attend certain visits. A future study could propose a novel imputation method for BiMM forest based on the framework for the default imputation procedure used in the *rfImpute* function of the open source RF software R package *randomForest* [52].

Many medical tools may be developed using novel BiMM tree and forest methodology, including prognosis models (e.g. predicting specific categorical outcomes), diagnostic models (e.g. determining whether or not a patient has a disease or condition), disease prediction models (e.g. examining risk factors of disease and determining if a person has a disease or not), treatment or therapy models (e.g. assessing the effectiveness of treatments on categorical patient outcomes), screening models (e.g. identifying patients at highest risk of developing a disease), and hospital models (e.g. estimating whether or not a patient will be readmitted). Thus, innovative BiMM tree and forest methods developed in this dissertation can be applied in diverse medical research settings to provide accurate and efficient prediction models for clustered and longitudinal binary outcomes.

5 REFERENCES

1. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and regression trees.* Wadsworth and Brooks: Monterrey, CA, USA, 1984.
2. Aguiar FS, Almeida LL, Ruffino-Netto A, Kritski AL, Mello FC, Werneck GL. Classification and regression tree (CART) model to predict pulmonary tuberculosis in hospitalized patients. *BMC Pulm Med* 2012; **12**: 40.
3. Garzotto M, Beer TM, Hudson RG, Peters L, Hsieh YC, Barrera E, Klein T, Mori M. Improved detection of prostate cancer using classification and regression tree analysis. *J Clin Oncol* 2005; **23**: 4322-4329.
4. Breiman L. Random forests. *Machine Learning* 2001; **45**: 5-32.
5. Boulesteix AL, Janitza S, Kruppa J, et al. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2012; **2**: 493-507.
6. Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning.* (2nd edn). Springer: New York, 2001.
7. Abdolell M, LeBlanc M, Stephens D, Harrison R. Binary partitioning for continuous longitudinal data: categorizing a prognostic variable. *Statistics in medicine* 2002; **21**: 3395-3409.
8. De'Ath G. Multivariate regression trees: a new technique for modeling species-environment relationships. *Ecology* 2002; **83**: 1105-1117.
9. Dine A, Larocque D, Bellavance F. Multivariate trees for mixed outcomes. *Computational Statistics & Data Analysis* 2009; **53**: 3795-3804.
10. Hajjem A, Bellavance F, Larocque D. Mixed effects regression trees for clustered data. *Statistics & probability letters* 2011; **81**: 451-459.
11. Keon Lee S. On generalized multivariate decision tree by using GEE. *Computational Statistics & Data Analysis* 2005; **49**: 1105-1119.
12. Larocque D. Mixed Effects Random Forest for Clustered Data A. Hajjem, F. Bellavance. 2010.
13. Loh W-Y, Zheng W. Regression trees for longitudinal and multiresponse data. *The Annals of Applied Statistics* 2013; **7**: 495-522.
14. Segal MR. Tree-structured methods for longitudinal data. *Journal of the American Statistical Association* 1992; **87**: 407-418.
15. Sela RJ, Simonoff JS. RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine Learning* 2012; **86**: 169-207.
16. Yu Y, Lambert D. Fitting trees to functional data, with an application to time-of-day patterns. *Journal of Computational and Graphical Statistics* 1999; **8**: 749-762.
17. Hajjem A, Bellavance F, Larocque D. Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation* 2014; **84**: 1313-1328.
18. Wu H, Zhang J-T. *Nonparametric regression methods for longitudinal data analysis: mixed-effects modeling approaches.* John Wiley & Sons, 2006.
19. Hothorn T, Hornik K, Zeileis A. party: A Laboratory for Recursive Part (y) itioning. R package version 0.9-9999. 2011. *URL: http://cran. r-project. org/package= party (1 December 2010, date last accessed).*

20. Bates D. Online Response to Convergence Issues in CRAN R LME4 Package. *Retrieved from* 2009.
21. Gelman A, Jakulin A, Pittau MG, Su Y-S. A weakly informative default prior distribution for logistic and other regression models. *The Annals of Applied Statistics* 2008: 1360-1383.
22. Zorn C. A solution to separation in binary response models. *Political Analysis* 2005; **13**: 157-170.
23. Hoofnagle J, Carithers R, Sapiro C, et al. Fulminant hepatic failure: summary of a workshop. *Hepatology* 1995; **21**: 240-252.
24. Bernal W, Auzinger G, Dhawan A, Wendon J. Acute liver failure. *The Lancet* 2010; **376**: 190-201.
25. Karvellas CJ, Safinia N, Auzinger G, Heaton N, Muiesan P, O'Grady J, Wendon J, Bernal W. Medical and psychiatric outcomes for patients transplanted for acetaminophen-induced acute liver failure: a case–control study. *Liver International* 2010; **30**: 826-833.
26. Bernuau J. Fulminant and subfulminant viral hepatitis. *La Revue du praticien* 1990; **40**: 1652-1655.
27. O'Grady JG, Alexander GJ, Hayllar KM, Williams R. Early indicators of prognosis in fulminant hepatic failure. *Gastroenterology* 1989; **97**: 439-445.
28. Bailey B, Amre DK, Gaudreault P. Fulminant hepatic failure secondary to acetaminophen poisoning: a systematic review and meta-analysis of prognostic criteria determining the need for liver transplantation. *Crit Care Med* 2003; **31**: 299-305.
29. Hastie T, Tibshirani R, Friedman J, Hastie T, Friedman J, Tibshirani R. *The elements of statistical learning.* Springer, 2009.
30. Fisher RA. The use of multiple measurements in taxonomic problems. *Annals of eugenics* 1936; **7**: 179-188.
31. Rao CR. *Linear statistical inference and its applications.* John Wiley & Sons, 2009.
32. Vapnik VN, Vapnik V. *Statistical learning theory.* Wiley New York, 1998.
33. Hertz J. *Introduction to the theory of neural computation.* Basic Books, 1991.
34. Bishop CM. Neural networks for pattern recognition. 1995.
35. Ripley BD. *Pattern recognition and neural networks.* Cambridge university press, 1996.
36. Loh WY. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2011; **1**: 14-23.
37. Loh WY. Fifty Years of Classification and Regression Trees. *International Statistical Review* 2014.
38. Karvellas CJ, Speiser JL, Lee WM. 476 Determining Late Predictors of Outcome for Acetaminophen-Induced Acute Liver Failure Using Classification and Regression Tree (CART) Modeling Analysis. *Gastroenterology* 2014; **146**: S-913.
39. Dietterich TG. Ensemble methods in machine learning. Springer, 2000.
40. Rokach L. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis* 2009; **53**: 4046-4072.

41. Cutler DR, Edwards TCJ, Beard KH, et al. Random forest for classification in ecology. *Ecology* 2007; **88**: 2783-2792.
42. Díaz-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 2006; **7**: 3.
43. Ishwaran H, Kogalur UB, Blackstone EH, Lauer MS. Random survival forests. *The Annals of Applied Statistics* 2008: 841-860.
44. Larivière B, Van den Poel D. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications* 2005; **29**: 472-484.
45. Schulter S, Roth PM, Bischof H. Ordinal Random Forests for Object Detection. Springer, 2013.
46. Siroky DS. Navigating random forests and related advances in algorithmic modeling. *Statistics Surveys* 2009; **3**: 147-163.
47. Speiser JL, Durkalski VL, Lee WM. Random forest classification of etiologies for an orphan disease. *Statistics in medicine* 2014.
48. Svetnik V, Liaw A, Tong C, Wang T. Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. Springer, 2004.
49. Tang R, Sinnwell JP, Li J, Rider DN, de Andrade M, Biernacka JM. Identification of genes and haplotypes that predict rheumatoid arthritis using random forests. BioMed Central Ltd: 2009.
50. Touw WG, Bayjanov JR, Overmars L, Backus L, Boekhorst J, Wels M, van Hijum SA. Data mining in the Life Sciences with Random Forest: a walk in the park or lost in the jungle? *Briefings in bioinformatics* 2012: bbs034.
51. Zhou Q, Hong W, Luo L, Yang F. Gene selection using random forest and proximity differences criterion on DNA microarray data. *Journal of Convergence Information Technology* 2010; **5**: 161-170.
52. Liaw A, Weiner M. Classification and Regression by randomForest. *R News* 2002; **2**: 18-22.
53. Schapire RE, Freund Y, Bartlett P, et al. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics* 1998; **26**: 1651-1686.
54. Verikas A, Gelzinis A, Bacauskiene M. Mining data with random forests: A survey and results of new tests. *Pattern Recognition* 2011; **44**: 330-349.
55. Strobl C, Boulesteix A-L, Zeileis A, Hothorn T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics* 2007; **8**: 25.
56. Nicodemus KK, Malley JD, Strobl C, Ziegler A. The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC bioinformatics* 2010; **11**: 110.
57. Little RJA, Rubin DB. Statistical analysis with missing data. 2002.
58. White IR, Royston P, Wood AM. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine* 2011; **30**: 377-399.
59. Ding Y, Simonoff JS. An investigation of missing data methods for classification trees applied to binary response data. *The Journal of Machine Learning Research* 2010; **11**: 131-170.

60. Twala B, Jones M, Hand DJ. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters* 2008; **29**: 950-956.
61. Rieger A, Hothorn T, Strobl C. Random forests with missing values in the covariates. University of Munich: 2010.
62. Schwarz DF, Szymczak S, Ziegler A, König IR. Evaluation of single-nucleotide polymorphism imputation using random forests. BioMed Central Ltd: 2009.
63. Pantanowitz A, Marwala T. Evaluating the impact of missing data imputation. Springer, 2009.
64. Hapfelmeier A, Hothorn T, Ulm K. Recursive partitioning on incomplete data using surrogate decisions and multiple imputation. *Computational Statistics & Data Analysis* 2012; **56**: 1552-1565.
65. Stekhoven DJ, Bühlmann P. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 2012; **28**: 112-118.
66. Hapfelmeier A, Hothorn T, Ulm K, Strobl C. A new variable importance measure for random forests with missing data. *Statistics and Computing* 2014; **24**: 21-34.
67. Hapfelmeier A, Ulm K. Variable selection with Random Forests for missing data. University of Munich: 2013.
68. Goldstein BA, Polley EC, Briggs F. Random forests for genetic association studies. *Statistical applications in genetics and molecular biology* 2011; **10**: 1-34.
69. van der Laan MJ. Statistical inference for variable importance. *The International Journal of Biostatistics* 2006; **2**.
70. Sandri M, Zuccolotto P. Variable selection using random forests. Springer, 2006.
71. Wang M, Chen X, Zhang H. Maximal conditional chi-square importance in random forests. *Bioinformatics* 2010; **26**: 831-837.
72. Nicodemus KK. Letter to the Editor: On the stability and ranking of predictors from random forest variable importance measures. *Briefings in bioinformatics* 2011; **12**: 369.
73. Altmann A, Toloşi L, Sander O, Lengauer T. Permutation importance: a corrected feature importance measure. *Bioinformatics* 2010; **26**: 1340-1347.
74. Yang WW, Gu CC. Selection of important variables by statistical learning in genome-wide association analysis. BioMed Central Ltd: 2009.
75. Rodenburg W, Heidema AG, Boer JM, Bovee-Oudenhoven IM, Feskens EJ, Mariman EC, Keijer J. A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes. *Physiological Genomics* 2008; **33**: 78-90.
76. Diggle P, Heagerty P, Liang K-Y, Zeger S. *Analysis of longitudinal data.* Oxford University Press, 2002.
77. Fitzmaurice GM, Laird NM, Ware JH. *Applied longitudinal analysis.* John Wiley & Sons, 2012.
78. Zhang H, Ye Y. A tree-based method for modeling a multivariate ordinal response. *Statistics and its interface* 2008; **1**: 169.
79. Lee T-H, Shih Y-S. Unbiased variable selection for classification trees with multivariate responses. *Computational Statistics & Data Analysis* 2006; **51**: 659-667.
80. Hsiao W-C, Shih Y-S. Splitting variable selection for multivariate regression trees. *Statistics & probability letters* 2007; **77**: 265-271.

81. Eo S-H, Cho H. Tree-structured Mixed-effects Regression Modeling for Longitudinal Data. *Journal of Computational and Graphical Statistics* 2013.
82. Ciampi A. Discussion. *International Statistical Review* 2014: n/a-n/a.
83. Segal M, Xiao Y. Multivariate random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2011; **1**: 80-87.
84. Karpievitch YV, Hill EG, Leclerc AP, Dabney AR, Almeida JS. An introspective comparison of random forest-based classifiers for the analysis of cluster-correlated data by way of RF++. *PloS one* 2009; **4**: e7087.
85. Lee WM, Squires RH, Nyberg SL, Doo E, Hoofnagle JH. Acute liver failure: summary of a workshop. *Hepatology* 2008; **47**: 1401-1415.
86. Fong Y, Rue H, Wakefield J. Bayesian inference for generalized linear mixed models. *Biostatistics* 2010; **11**: 397-412.
87. Dorie V. blme: Bayesian Linear Mixed-Effects Models. In blme: Bayesian Linear Mixed-Effects Models. R package: 2013.
88. Dorie V. Mixed Methods for Mixed Models. Columbia University: 2014.
89. Bates D, Maechler M, Bolker B, Walker S, Christensen RHB, Singmann H, Dai B, Grothendieck G, Eigen C, Rcpp L. Package 'lme4'. *convergence* 2015; **12**: 1.
90. Team RDC. *R: a language and environment for statistical computing*: Vienna, Austria, 2008.
91. Therneau TM, Atkinson EJ. An introduction to recursive partitioning using the Rpart routines. *Mayo Foundation* **1997**.
92. Speiser JL, Lee WM, Karvellas CJ. Predicting outcome on admission and post-admission for acetaminophen-induced acute liver failure using classification and regression tree models. *PloS one* 2015.
93. Antoniades CG, Berry PA, Wendon JA, Vergani D. The importance of immune dysfunction in determining outcome in acute liver failure. *Journal of hepatology* 2008; **49**: 845-861.
94. Stravitz RT, Kramer AH, Davern T, Shaikh AO, Caldwell SH, Mehta RL, Blei AT, Fontana RJ, McGuire BM, Rossaro L, Smith AD, Lee WM. Intensive care of patients with acute liver failure: recommendations of the U.S. Acute Liver Failure Study Group. *Crit Care Med* 2007; **35**: 2498-2508.
95. Mistler SA. A SAS macro for applying multiple imputation to multilevel data.
96. Koch DG, Tillman H, Durkalski V, Lee WM, Reuben A. Development of a Model to Predict Transplant-free Survival of Patients with Acute Liver Failure. *Clinical Gastroenterology and Hepatology* 2016.
97. Chipman HA, George EI, McCulloch RE. BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 2010: 266-298.
98. Kapelner A, Bleich J. bartMachine: Machine Learning with Bayesian Additive Regression Trees. *arXiv preprint arXiv:1312.2171* 2013.
99. Fernández-Delgado M, Cernadas E, Barro S, Amorim D. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res* 2014; **15**: 3133-3181.
100. Fagan E, Wannan G. Reducing paracetamol overdoses. *BMJ* 1996; **313**: 1417-1418.

101. Larson AM, Polson J, Fontana RJ, Davern TJ, Lalani E, Hynan LS, Reisch JS, Schiodt FV, Ostapowicz G, Shakil AO, Lee WM. Acetaminophen-induced acute liver failure: results of a United States multicenter, prospective study. *Hepatology* 2005; **42**: 1364-1372.
102. Karvellas CJ, Safinia N, Auzinger G, Heaton N, Muiesan P, O'Grady J, Wendon J, Bernal W. Medical and psychiatric outcomes for patients transplanted for acetaminophen-induced acute liver failure: a case-control study. *Liver Int* 2010; **30**: 826-833.
103. Bernal W, Auzinger G, Dhawan A, Wendon J. Acute liver failure. *Lancet* 2010; **376**: 190-201.
104. Reddy KR, Schilsky M, Stravitz RT, Eberle C, Durkalski V, Fontana RJ, Lee WM. Liver transplantation for Acute Liver Failure: Results from the NIH Acute Liver Failure Study Group. *Hepatology* 2012; **56**: 246A.
105. Simpson KJ, Bates CM, Henderson NC, Wigmore SJ, Garden OJ, Lee A, Pollok A, Masterton G, Hayes PC. The utilization of liver transplantation in the management of acute liver failure: comparison between acetaminophen and non-acetaminophen etiologies. *Liver Transpl* 2009; **15**: 600-609.
106. Pauwels A, Mostefa-Kara N, Florent C, Levy VG. Emergency liver transplantation for acute liver failure. Evaluation of London and Clichy criteria. *Journal of hepatology* 1993; **17**: 124-127.
107. Schmidt LE, Dalhoff K. Serum phosphate is an early predictor of outcome in severe acetaminophen-induced hepatotoxicity. *Hepatology* 2002; **36**: 659-665.
108. Schmidt LE, Larsen FS. MELD score as a predictor of liver failure and death in patients with acetaminophen-induced liver injury. *Hepatology* 2007; **45**: 789-796.
109. Bernal W, Donaldson N, Wyncoll D, Wendon J. Blood lactate as an early predictor of outcome in paracetamol-induced acute liver failure: a cohort study. *Lancet* 2002; **359**: 558-563.
110. Shakil AO, Kramer D, Mazariegos GV, Fung JJ, Rakela J. Acute liver failure: clinical features, outcome analysis, and applicability of prognostic criteria. *Liver Transpl* 2000; **6**: 163-169.
111. Speiser JL, Lee WM, Karvellas CJ. Predicting outcome on admission and post-admission for acetaminophen-induced acute liver failure using classification and regression tree models. *PloS one* 2015; **10**: e0122929.
112. O'Grady JG, Schalm SW, Williams R. Acute liver failure: redefining the syndromes. *Lancet* 1993; **342**: 273-275.
113. Atterbury CE, Maddrey WC, Conn HO. Neomycin-sorbitol and lactulose in the treatment of acute portal-systemic encephalopathy. A controlled, double-blind clinical trial. *The American journal of digestive diseases* 1978; **23**: 398-406.
114. Team. RDC. R: a language and environment for statistical computing. 2008.
115. Yoshida K, Bohn J. tableone: Create" Table 1" to Describe Baseline Characteristics. *R package version 0.7* 2015; **3**.
116. Sing T, Sander O, Beernwinkel N, Lengauer T. ROCR: visualizing classifier performance in R. *Bioinformatics* 2005; **21**.

117. Kamath PS, Wiesner RH, Malinchoc M, Kremers W, Therneau TM, Kosberg CL, D'Amico G, Dickson ER, Kim WR. A model to predict survival in patients with end-stage liver disease. *Hepatology* 2001; **33**: 464-470.
118. Díaz-Uriarte R, De Andres SA. Gene selection and classification of microarray data using random forest. *BMC bioinformatics* 2006; **7**.

APPENDIX 1: Supplementary Figures

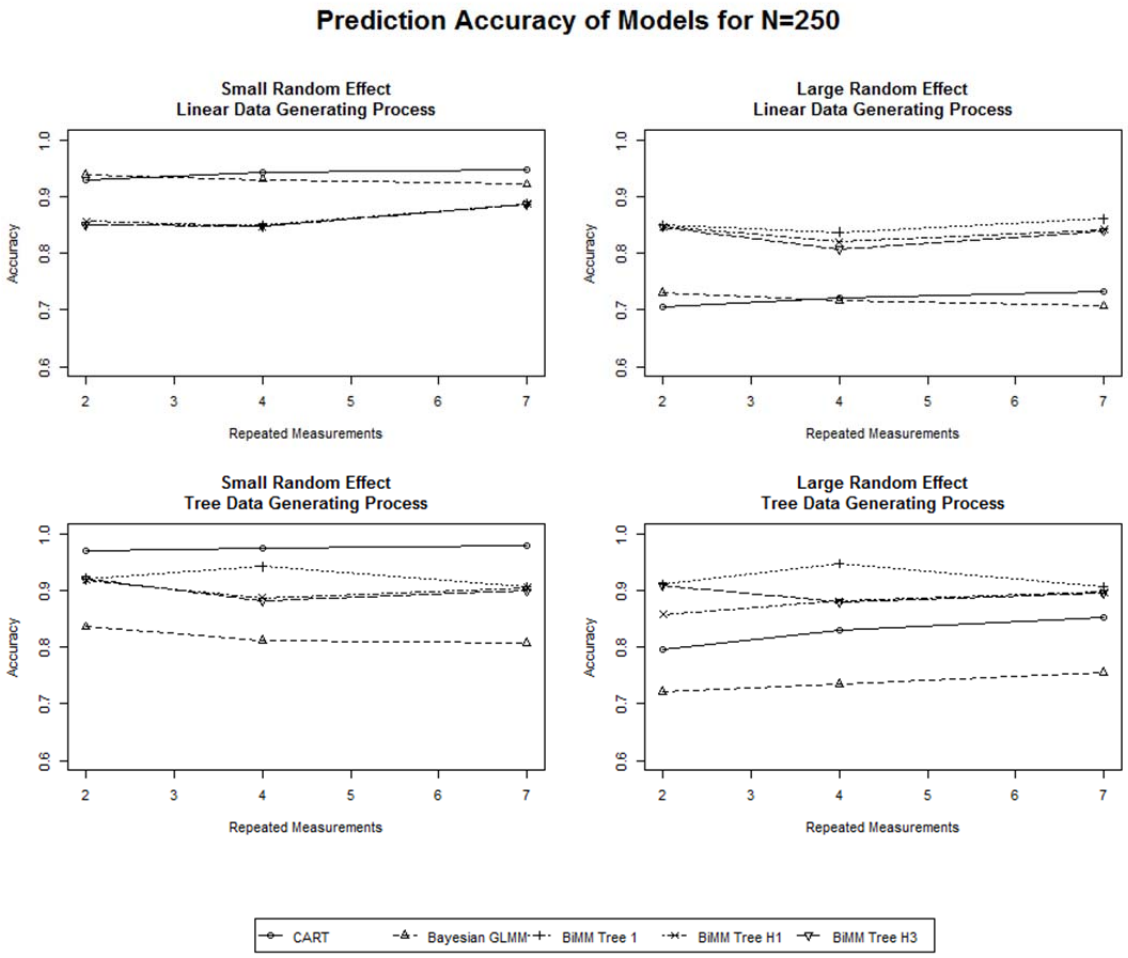


Figure 1: Simulated prediction (test set) accuracy of models for N=250 patients

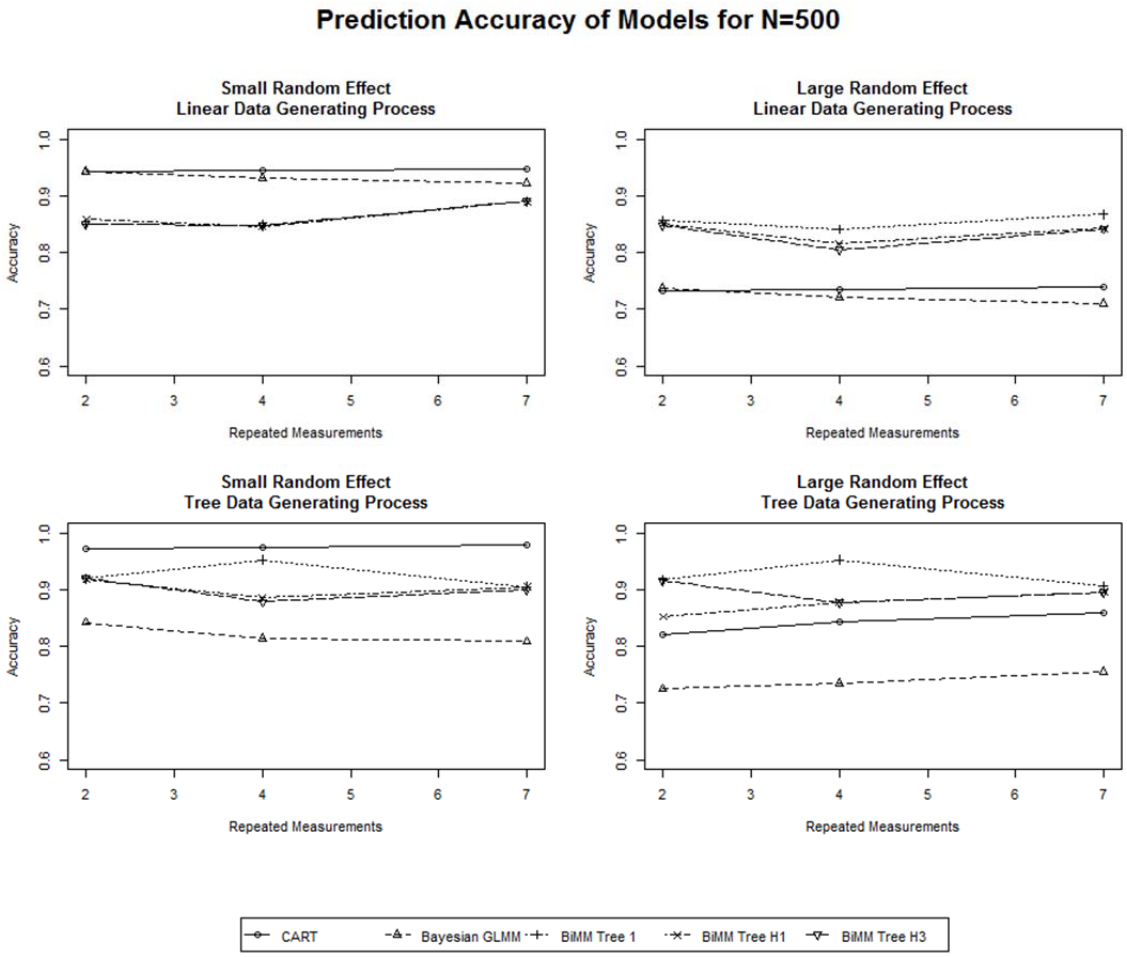


Figure 2: Simulated prediction (test set) accuracy of models for N=500 patients

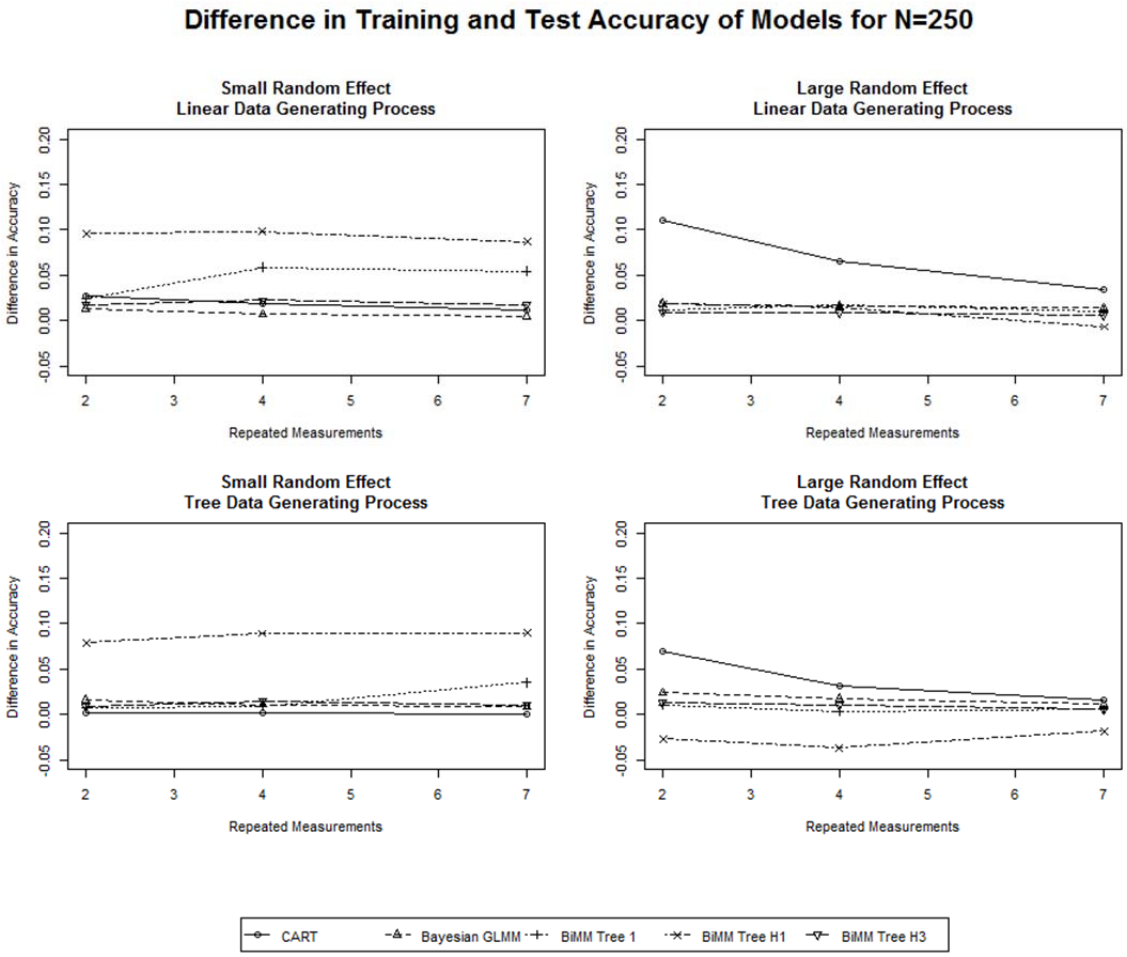


Figure 3: Simulated difference in training and test set accuracy of models for N=250 patients

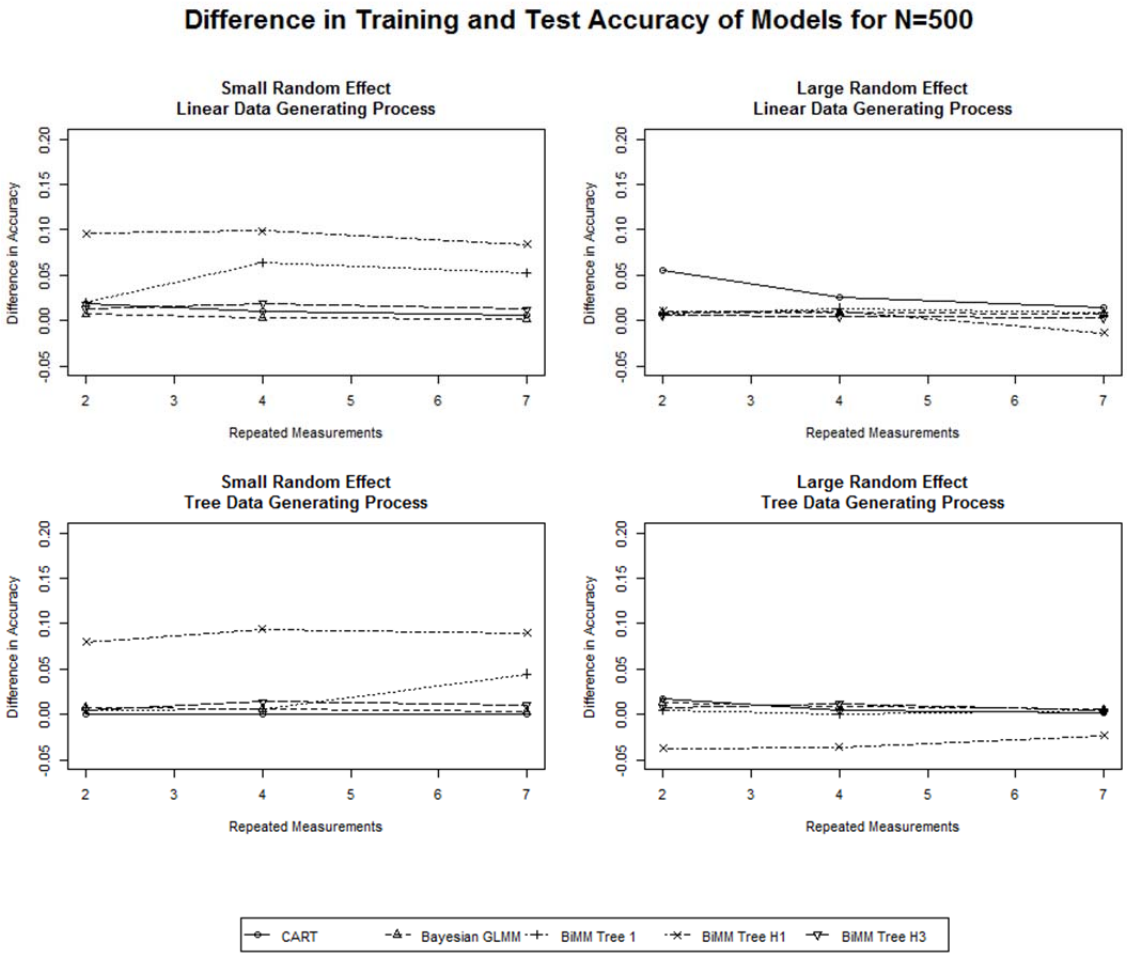


Figure 4: Simulated difference in training and test set accuracy of models for N=500 patients

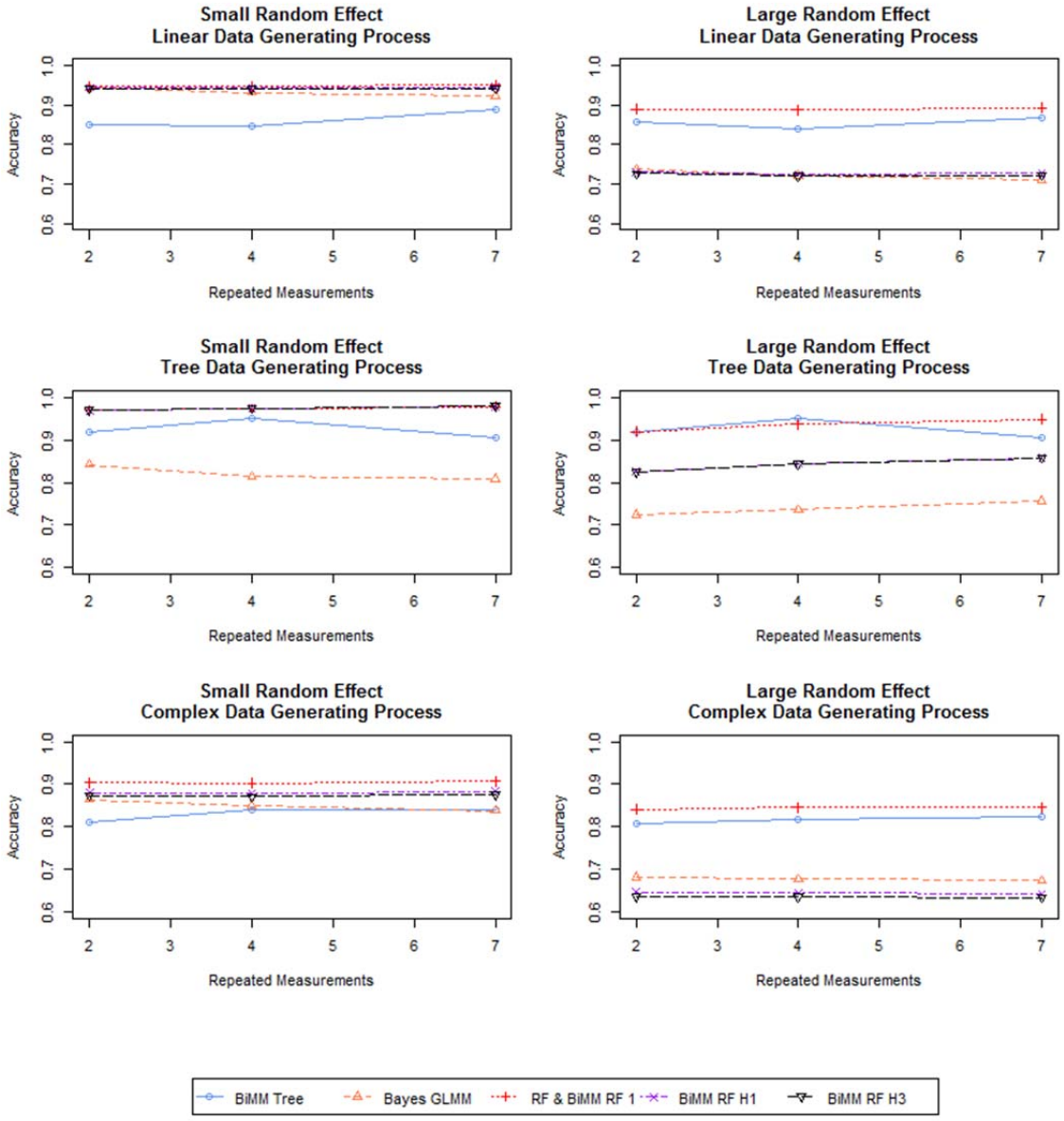


Figure 5: Simulated prediction (test set) accuracy of models for N=500 patients

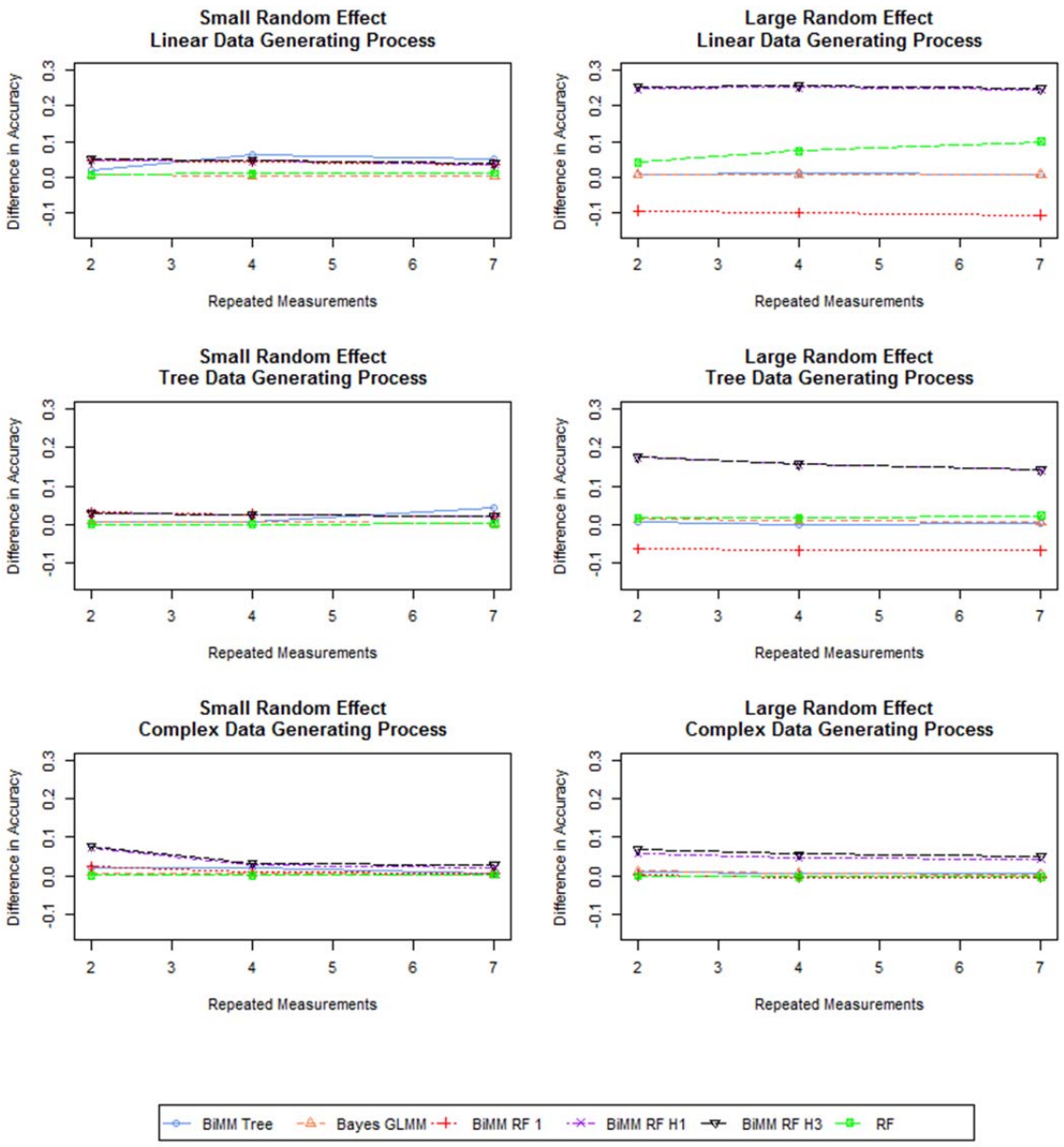


Figure 6: Simulated difference in training and test set accuracy of models for N=500 patients

APPENDIX 2: R code for BiMM tree and BiMM forest functions

```
#load libraries
library(rpart)
library(blme)
library(randomForest)

#################################################################################
#variable names
#traindata: name of the training dataset
#testdata: name of the test dataset
#formula: formula for fixed variables with binary outcome
       #example:
comagradelow1~Sex+Ethnicity+Age+ALT+AST+Bilirubin+Creat+Phosphate+Lactate+plate
lets+ammonia
              +inr+pressors+rrt
#random: name of the random clustering variable

#################################################################################
#BiMM tree with one iteration
BiMMtree1<-function(traindata,testdata,formula,random){
       #initialize parameters
       minsize=round(length(traindata[,1])/10,0)
       data=traindata
       initialRandomEffects=rep(0,length(data[,1]))
       ErrorTolerance=0.001
       MaxIterations=1000
       tree.control=rpart.control(minbucket=minsize)
       #parse formula
       Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
       TargetName<-formula[[2]]
       Target<-data[,toString(TargetName)]
       #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

       #run 1 iteration of algorithm
              newdata[,"AdjustedTarget"] <- AdjustedTarget
              iterations <- iterations+1
              #build tree
              tree <- rpart(formula(paste(c("AdjustedTarget",
Predictors),collapse = "~")),
                     data = data, method = "class", control = tree.control)

              ## Estimate New Random Effects and Errors using BLMER
              # Get variables that identify the node for each observation
              data[,"nodeInd"] <- 0
              data["nodeInd"] <- tree$where
              # Fit linear model with nodes as predictors (we use the original
target so likelihoods are comparable)
              # Check that the fitted tree has at least two nodes.
       if(min(tree$where)==max(tree$where)){
                     lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),1), collapse="~"),
"+(1|random)",sep=""))),
```

```
	data=data,family=binomial,control=glmerControl(optCtrl=list(maxfun=20000)
)),error=function(cond)"skip")
		} else {
			 lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),"as.factor(nodeInd
)"), collapse="~"), "+(1|random)",sep=""))),
				data=data,
family=binomial,control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=200
00000000))),error=function(cond)"skip")
		 }

	#if GLMM did not converge, return NA's for accuracy statistics
	if(class(lmefit)[1]=="character"){
		#return train and test confusion matrices
		return(list(
			c(NA,NA,NA,NA),
			c(NA,NA,NA,NA),
			NA
		))
	}
	else if(!(class(lmefit)[1]=="character")){
		#train dataset predictions
		train.preds.ave<- AdjustedTarget
		train.preds<-predict(tree,traindata,type="class")
		#test dataset predictions
		test.preds<-predict(tree,testdata,type="class")
		#format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
		t1<-table(data$comagradelow,train.preds.ave)
		t4<-table(testdata$comagradelow,test.preds)
		#code if table for train or test data if all predictions are for
same group
		if(ncol(t1)==1 & train.preds.ave[1]==1){
			t1<-c(0,0,t1[1,1],t1[2,1])
		}
		else if(ncol(t1)==1 & train.preds.ave[1]==0){
			t1<-c(t1[1,1],t1[2,1],0,0)
		}
		if(ncol(t4)==1 & test.preds[1]==1){
			t4<-c(0,0,t4[1,1],t4[2,1])
		}
		else if(ncol(t4)==1 & test.preds[1]==0){
			t4<-c(t4[1,1],t4[2,1],0,0)
		}
		#return train and test confusion matrices, # iterations
		return(list(
			c(t1),
			c(t4),
			iterations
		))
	}
}
```

```
#############################################################################
#BiMM forest with one iteration
#note: requires training and test data with no missing values

BiMMforest1<-function(traindata,testdata,formula,random,seed){
	#set up variables for Bimm method
	data=traindata1
	initialRandomEffects=rep(0,length(data[,1]))
	ErrorTolerance=0.006
	MaxIterations=1000
	#parse formula
	Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
	TargetName<-formula[[2]]
	Target<-data[,toString(TargetName)]
	#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

	#compile one iteration of the BiMM forest algorithm
		newdata[,"AdjustedTarget"] <- AdjustedTarget
		iterations <- iterations+1
		#build tree
		set.seed(seed)
		forest <- randomForest(formula(paste(c("factor(AdjustedTarget)",
Predictors),collapse = "~")),
			data = data, method = "class")
		forestprob<-predict(forest,type="prob")[,2]
		## Estimate New Random Effects and Errors using GLMER
		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 GLMM did not converge, produce NAs for accuracy statistics
	if(class(lmefit)[1]=="character"){
		#return train and test confusion matrices
		return(list(
			c(NA,NA,NA,NA),
			c(NA,NA,NA,NA),
			NA
		))
	}
	else if(!(class(lmefit)[1]=="character")){
		test.preds<-predict(forest,testdata1)
		traindata1<-cbind(traindata1,random)
		train.preds<-
ifelse(predict(lmefit,traindata1,type="response")<.5,0,1)
		#format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
		t1<-table(traindata1$comagradelow1,train.preds)
		t4<-table(testdata1$comagradelow1,test.preds)
		if(ncol(t1)==1 & train.preds[1]==1){
```

```
                t1<-c(0,0,t1[1,1],t1[2,1])
            }
            else if(ncol(t1)==1 & train.preds[1]==0){
                t1<-c(t1[1,1],t1[2,1],0,0)
            }
            if(ncol(t4)==1 & test.preds[1]==1){
                t4<-c(0,0,t4[1,1],t4[2,1])
            }
            else if(ncol(t4)==1 & test.preds[1]==0){
                t4<-c(t4[1,1],t4[2,1],0,0)
            }

            #return train and test confusion matrices, # iterations
            return(list(
                c(t1),
                c(t4),
                iterations
            ))
        }
}
```

```
################################################################################
#BiMM tree with H1 updates

BiMMtreeH1<-function(traindata,testdata,formula,random,seed){
	#set up variables for Bimm method
	data=traindata1
	initialRandomEffects=rep(0,length(data[,1]))
	ErrorTolerance=0.006
	MaxIterations=1000
	#parse formula
	Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
	TargetName<-formula[[2]]
	Target<-data[,toString(TargetName)]
	#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 tree
		tree <- rpart(formula(paste(c("AdjustedTarget",
Predictors),collapse = "~")),
			data = data, method = "class", control = tree.control)

		## Estimate New Random Effects and Errors using BLMER
		# Get variables that identify the node for each observation
		data[,"nodeInd"] <- 0
		data["nodeInd"] <- tree$where
		# Fit linear model with nodes as predictors (we use the original
target so likelihoods are comparable)
		# Check that the fitted tree has at least two nodes.
	if(min(tree$where)==max(tree$where)){
			lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),1), collapse="~"),
"+(1|random)",sep=""))),

	data=data,family=binomial,control=glmerControl(optCtrl=list(maxfun=20000)
)),error=function(cond)"skip")
		} else {
			 lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),"as.factor(nodeInd
)"), collapse="~"), "+(1|random)",sep=""))),
				data=data,
family=binomial,control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=200
00000000))),error=function(cond)"skip")
		 }
		# Get the likelihood to check on convergence
		if(!(class(lmefit)[1]=="character")){
			newlik <- logLik(lmefit)
			ContinueCondition <- (newlik-oldlik>ErrorTolerance &
iterations < MaxIterations)
			oldlik <- newlik
			# Extract random effects to make the new adjusted target
```

```
                    logit<-predict(tree,type="prob")[,2]
                    logit2<-
exp(predict(lmefit,re.form=NA))/(1+exp(predict(lmefit,re.form=NA))) #population
level effects
                    AllEffects <- (logit+logit2)/2 #average them
                    #split function h1
                    AdjustedTarget <- ifelse(as.numeric(AdjustedTarget) +
AllEffects>.5,1,0)
             }
             else{ ContinueCondition<-FALSE }
      }

      if(class(lmefit)[1]=="character"){
             #return train and test confusion matrices
             return(list(
                    c(NA,NA,NA,NA),
                    c(NA,NA,NA,NA),
                    NA
             ))
      }
      else if(!(class(lmefit)[1]=="character")){
             #average effects
             train.preds.ave<- AdjustedTarget
             #test dataset predictions-same for all 3 updating methods for the
1 iteration model
             test.preds<-predict(tree,testdata,type="class")
             #format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
             t1<-table(data$ys,train.preds.ave)
             t4<-table(testdata$ys,test.preds)
             if(ncol(t1)==1 & train.preds.ave[1]==1){
                    t1<-c(0,0,t1[1,1],t1[2,1])
             }
             else if(ncol(t1)==1 & train.preds.ave[1]==0){
                    t1<-c(t1[1,1],t1[2,1],0,0)
             }
             if(ncol(t4)==1 & test.preds[1]==1){
                    t4<-c(0,0,t4[1,1],t4[2,1])
             }
             else if(ncol(t4)==1 & test.preds[1]==0){
                    t4<-c(t4[1,1],t4[2,1],0,0)
             }
             #return train and test confusion matrices
             return(list(
                    c(t1),
                    c(t4),
                    iterations
             ))
      }
}
```

```
################################################################################
#BiMM tree with H3 updates

BiMMtreeH3<-function(traindata,testdata,formula,random,seed){
       #set up variables for Bimm method
       data=traindata1
       initialRandomEffects=rep(0,length(data[,1]))
       ErrorTolerance=0.006
       MaxIterations=1000
       #parse formula
       Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
       TargetName<-formula[[2]]
       Target<-data[,toString(TargetName)]
       #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 tree
              tree <- rpart(formula(paste(c("AdjustedTarget",
Predictors),collapse = "~")),
                     data = data, method = "class", control = tree.control)
              ## Estimate New Random Effects and Errors using BLMER
              # Get variables that identify the node for each observation
              data[,"nodeInd"] <- 0
              data["nodeInd"] <- tree$where
              # Fit linear model with nodes as predictors (we use the original
target so likelihoods are comparable)
              # Check that the fitted tree has at least two nodes.
       if(min(tree$where)==max(tree$where)){
                     lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),1), collapse="~"),
"+(1|random)",sep=""))),

       data=data,family=binomial,control=glmerControl(optCtrl=list(maxfun=20000)
)),error=function(cond)"skip")
              } else {
                      lmefit <-
tryCatch(bglmer(formula(c(paste(paste(c(toString(TargetName),"as.factor(nodeInd
)"), collapse="~"), "+(1|random)",sep=""))),
                                 data=data,
family=binomial,control=glmerControl(optimizer="bobyqa",optCtrl=list(maxfun=200
00000000))),error=function(cond)"skip")
               }
              # Get the likelihood to check on convergence
              if(!(class(lmefit)[1]=="character")){
                     newlik <- logLik(lmefit)
                     ContinueCondition <- (newlik-oldlik>ErrorTolerance &
iterations < MaxIterations)
                     oldlik <- newlik
                     # Extract random effects to make the new adjusted target
                     logit<-predict(tree,type="prob")[,2]
```

```
                    logit2<-
exp(predict(lmefit,re.form=NA))/(1+exp(predict(lmefit,re.form=NA))) #population
level effects
                    AllEffects <- (logit+logit2)/2 #average them
                    #AdjustedTarget <- ifelse(as.numeric(AdjustedTarget) +
AllEffects>.5,1,0)
                    #new split function h3
                    for(k in 1:length(AllEffects)){

      if(as.numeric(AdjustedTarget[k])+AllEffects[k]<.5){AdjustedTarget[k]=0}
                           else
if(as.numeric(AdjustedTarget[k])+AllEffects[k]>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])
                           }
                    }
             }
             else{ ContinueCondition<-FALSE }
      }

      if(class(lmefit)[1]=="character"){
             #return train and test confusion matrices
             return(list(
                    c(NA,NA,NA,NA),
                    c(NA,NA,NA,NA),
                    NA
             ))
      }
      else if(!(class(lmefit)[1]=="character")){
             #average effects
             train.preds.ave<- AdjustedTarget
             #test dataset predictions-same for all 3 updating methods for the
1 iteration model
             test.preds<-predict(tree,testdata,type="class")
             #format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
             t1<-table(data$ys,train.preds.ave)
             t4<-table(testdata$ys,test.preds)
             if(ncol(t1)==1 & train.preds.ave[1]==1){
                    t1<-c(0,0,t1[1,1],t1[2,1])
             }
             else if(ncol(t1)==1 & train.preds.ave[1]==0){
                    t1<-c(t1[1,1],t1[2,1],0,0)
             }
             if(ncol(t4)==1 & test.preds[1]==1){
                    t4<-c(0,0,t4[1,1],t4[2,1])
             }
             else if(ncol(t4)==1 & test.preds[1]==0){
                    t4<-c(t4[1,1],t4[2,1],0,0)
             }
             #return train and test confusion matrices, # iterations
             return(list(
                    c(t1),
                    c(t4),
                    iterations
             ))
      }
}
```

```
################################################################################
#BiMM forest with H1 updates

BiMMforestH1<-function(traindata,testdata,formula,random,seed){
	#set up variables for Bimm method
	data=traindata1
	initialRandomEffects=rep(0,length(data[,1]))
	ErrorTolerance=0.006
	MaxIterations=1000
	#parse formula
	Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
	TargetName<-formula[[2]]
	Target<-data[,toString(TargetName)]
	#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
	shouldpredict=TRUE

	while(ContinueCondition){
		# Current values of variables
		newdata[,"AdjustedTarget"] <- AdjustedTarget
		iterations <- iterations+1
		#build tree
		set.seed(seed)
		forest <- randomForest(formula(paste(c("factor(AdjustedTarget)",
Predictors),collapse = "~")),
			data = data, method = "class")
		forestprob<-predict(forest,type="prob")[,2]
		## Estimate New Random Effects and Errors using BLMER
		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")
		# 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,re.form=NA))/(1+exp(predict(lmefit,re.form=NA))) #population
level effects
			AllEffects <- (logit+logit2)/2 #average them
			#h1 update
			AdjustedTarget <- ifelse(as.numeric(Target) + AllEffects-
1>.5,1,0)

		}
		else{ ContinueCondition<-FALSE }
		#if all of the binary outcomes are the same then get out of loop
		if(min(AdjustedTarget)==max(AdjustedTarget)){
```

```
                        ContinueCondition<-FALSE
                        shouldpredict=FALSE
                }
        }

        if(class(lmefit)[1]=="character" | shouldpredict==FALSE){
                #return train and test confusion matrices
                return(list(
                        c(NA,NA,NA,NA),
                        c(NA,NA,NA,NA),
                        NA,
                        NA
                ))
        }
        else if(!(class(lmefit)[1]=="character")){
                #predictions
                test.preds<-predict(forest,testdata)
                traindata1<-cbind(traindata,random)
                train.preds<-
ifelse(predict(lmefit,traindata1,type="response")<.5,0,1)
                #format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
                t1<-table(traindata$ys,train.preds)
                t4<-table(testdata$ys,test.preds)
                if(ncol(t1)==1 & train.preds[1]==1){
                        t1<-c(0,0,t1[1,1],t1[2,1])
                }
                else if(ncol(t1)==1 & train.preds[1]==0){
                        t1<-c(t1[1,1],t1[2,1],0,0)
                }
                if(ncol(t4)==1 & test.preds[1]==1){
                        t4<-c(0,0,t4[1,1],t4[2,1])
                }
                else if(ncol(t4)==1 & test.preds[1]==0){
                        t4<-c(t4[1,1],t4[2,1],0,0)
                }

                #return train and test confusion matrices, # iterations, and RF
OOBER
                return(list(
                        c(t1),
                        c(t4),
                        iterations,
                        mean(forest$err.rate[,1])
                ))
        }
}
```

```
################################################################################
#BiMM forest with H3 updates

BiMMforestH3<-function(traindata,testdata,formula,random,seed){
	#set up variables for Bimm method
	data=traindata1
	initialRandomEffects=rep(0,length(data[,1]))
	ErrorTolerance=0.006
	MaxIterations=1000
	#parse formula
	Predictors<-paste(attr(terms(formula),"term.labels"),collapse="+")
	TargetName<-formula[[2]]
	Target<-data[,toString(TargetName)]
	#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
	shouldpredict=TRUE

	while(ContinueCondition){
		# Current values of variables
		newdata[,"AdjustedTarget"] <- AdjustedTarget
		iterations <- iterations+1
		#build tree
		set.seed(seed)
		forest <- randomForest(formula(paste(c("factor(AdjustedTarget)",
Predictors),collapse = "~")),
			data = data, method = "class")
		forestprob<-predict(forest,type="prob")[,2]
		## Estimate New Random Effects and Errors using BLMER
		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")
		# 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,re.form=NA))/(1+exp(predict(lmefit,re.form=NA))) #population
level effects
			AllEffects <- (logit+logit2)/2 #average them
			#split function 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)
                                    set.seed(seed)
                                    AdjustedTarget[k]<-rbinom(1,1,AllEffects[k])
                              }
                        }


                }
                else{ ContinueCondition<-FALSE }
                #if all of the binary outcomes are the same then get out of loop
                if(min(AdjustedTarget)==max(AdjustedTarget)){
                        ContinueCondition<-FALSE
                        shouldpredict=FALSE
                }
        }

        if(class(lmefit)[1]=="character" | shouldpredict==FALSE){
                #return train and test confusion matrices
                return(list(
                        c(NA,NA,NA,NA),
                        c(NA,NA,NA,NA),
                        NA,
                        NA
                ))
        }
        else if(!(class(lmefit)[1]=="character")){
                #predictions
                test.preds<-predict(forest,testdata)
                traindata1<-cbind(traindata,random)
                train.preds<-
ifelse(predict(lmefit,traindata1,type="response")<.5,0,1)
                #format table to make sure it always has 4 entries, even if it is
only 2 by 1 (0's in other spots)
                t1<-table(traindata1$ys,train.preds)
                t4<-table(testdata$ys,test.preds)
                if(ncol(t1)==1 & train.preds[1]==1){
                        t1<-c(0,0,t1[1,1],t1[2,1])
                }
                else if(ncol(t1)==1 & train.preds[1]==0){
                        t1<-c(t1[1,1],t1[2,1],0,0)
                }
                if(ncol(t4)==1 & test.preds[1]==1){
                        t4<-c(0,0,t4[1,1],t4[2,1])
                }
                else if(ncol(t4)==1 & test.preds[1]==0){
                        t4<-c(t4[1,1],t4[2,1],0,0)
                }
                #return train and test confusion matrices, # iterations, and RF
OOBER
                return(list(
                        c(t1),
                        c(t4),
                        iterations,
                        mean(forest$err.rate[,1])
                ))
        }
}
```