Fixed Effects or Mixed Effects?: Predictive Classification in the Presence of Multilevel Data

by

Anthony A. Mangino, M.S.

Adviser: Dr. Jocelyn Bolin

Dissertation

Submitted to the Graduate School

In Partial Fulfillment of the Requirements

for the Degree Doctor of Philosophy

in Educational Psychology

Ball State University

Muncie, IN

July 2021

Table of Contents

This work utilized Ball State University's Beowulf computing cluster, which is supported by the National Science Foundation (MRI-1726017) and Ball State University, Muncie, Indiana.

Acknowledgments

This dissertation is the culmination of three years of doctoral level study and ten total years of post-secondary education. It is without reserve I can state that this doctoral program—including this work—is the most difficult thing I have ever completed. Throughout this incredible journey, there are a few folks that deserve the thanks I am about to give in permanent [digitized] form… and so much more. It is for this reason I am unapologetically breaking the one-page convention in this section.

I've often said that I consider myself a student of the Department of Educational Psychology at Ball State, for I have learned such an incredible amount from all the faculty in the department. The community that has supported my education throughout this program is, thus, quite vast and requires great thanks. I thank my adviser, Dr. Jocelyn Bolin, for her support throughout my time in the program, her substantially-more-than-a-little-helpful recommended edits on this document (among others), and her continual reminders that weekends are incredibly important. I thank Dr. Holmes Finch for his insights and intrigue to help expand my repertoire of both statistical methods and musical taste. I thank Dr. Lisa Rubenstein for her sense of humor and ability to ask of my research the questions I might not intuitively think to ask. I thank Dr. Linda Martin for her conversations and the validation she has provided throughout this process.

I must also thank Dr. Jerrell Cassady for his professional support and insights, that witty sense of humor that allows each lab meeting to be *the* event of the week, and the professional insights and mentorship he has been so generous to provide. I thank Dr. Serena Shim for being my initial contact at Ball State and helping me kick this journey off with the right support. I thank Drs. Matt Stuve, Kathryn Fletcher, and Gerardo Ramirez for their pedagogical insights that have made my experience in the program—and my time teaching for the department—so incredibly valuable and enjoyable.

As I reflect upon my past education, I must also thank my former mentors. I thank Dr. Robert Johnson for introducing me to the world of Educational Psychology and for continuing to be a positive force for my personal and professional growth throughout my education. I thank Dr. Amy Eppolito, the first professor whose class pushed me to actually read a textbook, who convinced me that maybe I *could* write well, who drove my decision (in part) to choose research methods as my focus (because I 'shouldn't be afraid of statistics, research methods is more difficult', so of course I must do *both* difficult things), and who unknowingly became a mentor long after I completed her courses.

My family has been so extraordinarily supportive throughout my life and education that I *almost* cannot begin to thank them. My dad has shown me an extraordinary amount of resilience, humor, and determination throughout my life and—even when I was wrong—has continued to be a true mentor and friend. My mom has always been there, ready to listen to my nonsensical ramblings over the phone whenever I had news (good, bad, or otherwise), and who recognized that no matter how old this man may be, sometimes he still needs his mom. My extended family have all helped me in some way, shape, or form with moral support, care, and insight, but it is my parents who have been the consistency in my life and exemplify that home truly is wherever *my* family is.

To everyone who has helped, supported, and inspired me throughout my journey (and my life) thus far, I owe more thanks than could be encompassed in a page or two. Know that my gratitude knows no bounds.

List of Abbreviations

ANOVA	Analysis of variance
AUC	Area under the [receiver operating characteristic] curve
CART	Classification and regression trees
CE	Binary cross-entropy
DFA	Discriminant function analysis
FN	False Negative
FP	False Positive
GLMM	Generalized linear mixed model
ICC	Intraclass/intracluster correlation
LGR	Large group recovery
LR	Logistic Regression
MANOVA	Multivariate analysis of variance
MERF	Mixed effects random forest
OLS	Ordinary least squares estimation
PISA	Program for International Student Assessment
RF	Random Forests
RMSE	Root mean-square error
SGR	Small group recovery
TN	True Negative
TP	True Positive

List of Figures

List of Tables

List of Algorithms

Abstract

Some of the data in the social sciences features a nesting structure in which cases (level-1 units) are nested within higher-level clusters (level-2 units). This structure violates a fundamental assumption of many statistical models, namely the independence of cases, and thus necessitates the use of multilevel modeling techniques. Little research has yet been done assessing the efficacy of fixed and mixed effects models for supervised classification, where the outcome groups are known. The present study sought to compare fixed and mixed effects models for the purposes of predictive classification in the presence of multilevel data with small sample sizes. The first part of the study utilizes a Monte Carlo simulation methodology to systematically manipulate conditions within multilevel data across several different classifiers, including fixed and mixed effects logistic regression and random forests. Following the simulation study, an applied examination of the prediction of student retention in the public use Program in International Student Achievement (PISA) dataset will be considered to further bolster findings from the simulation study. Collectively, the results of both the simulation study and PISA data examinations will be used to provide recommendations to researchers for use when implementing classifiers for the purpose of prediction. Results of this study indicate that despite the use of fixed effects models, their predictions were nearly equivalent to mixed effects models across both the simulation and PISA examinations regardless of sample size. Taken holistically, these results suggest that researchers should be more cognizant of the type of predictors and the data structure being used, as these factors carried more weight than did the model type in accuracy metrics.

It can scarcely be denied that the supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience.

Albert Einstein

CHAPTER 1: INTRODUCTION

The social sciences are rife with contexts in which parsimony is championed and pursued with incredible vigor. Never is it more apparent than in the use and interpretation of statistical analyses. Nearly uniform among statisticians and practitioners within the various areas of statistics, machine learning, and other methodological research is the desire to pursue ever more informative and concomitantly complex methodologies to permit increasingly more complex inquiry. Simultaneously, researchers in the social sciences—and the methodologists in these various domains—often (though not always) seek to utilize only those methods as complex as necessary to facilitate maximally effective investigations into social, psychological, and educational phenomena (Little, 2013; Murtaugh, 2008; Zellner et al., 2001). This push for maintaining as much simplicity as possible is a quintessential exemplification of Einstein's oft-cited quote as the complexity in social science inquiry should be relegated primarily to its theoretical implications, not necessarily its methodological implementations. As a result, the present study seeks to establish guidelines for facilitating parsimony from notoriously complex data structures to provide evidence in favor of simplicity and recommendations for effective practice on the part of social science researchers in the area of predictive classification. That is, the intersection of classification methodology and multilevel data structures is as yet sparsely studied and, thus, recommendations for practice are not well established particularly when small sample sizes are considered. The present study then serves as an examination of this intersection to identify such optimal recommendations.

Supervised statistical classification analysis (not to be confused with unsupervised, which is not part of the present study) is a family of methods designed to predict into which of two or more groups a case should be assigned when the outcome group is known (Hastie et al., 2009). Classification analyses can serve two primary purposes: Explaining the variables most salient in identifying membership in groups, and predicting the group membership of new cases. Several

methods (also referred to as ‘classifiers’ or ‘algorithms’ and refer only to supervised methods) have been identified in the classification literature including, for example, logistic regression (LR), classification and regression trees (CART), and random forests (RF), among others. Of these methods, LR has been the most commonly employed due to its resemblance to linear regression and ease of interpretation (and its lengthy history compared to many other methods).

Expanding beyond this explanatory purpose—and the purpose germane to the current studies—is the use of classifiers to predict group membership of cases not previously seen in the dataset following an initial training period in which group membership is known (thus placing these methods still within the supervised classification paradigm; Steyerberg, 2019). This prediction capacity follows a method as such: A classifier is trained on data consisting of a set of cases for which group membership is known (identified as the training set); this trained classifier is then applied to a new dataset with cases previously unseen and for which group membership is not known (the test set). These predictions provide researchers and practitioners with the information necessary to engage in maximally effective practices such as implementing classroom (VanDerHeyden, 2013) or psychiatric interventions (Zigler & Phillips, 1961), among others. The estimation of classifiers is plagued by a number of methodological issues due to the number of unknown conditions and results of cases within the test set and, as such, requires additional consideration in order to identify the optimal methods when used to make optimally accurate predictions such that early intervention may be employed.

The purposes of classification must first be considered prior to examining the conditions relevant to effective practices. While classifiers can be estimated to explain the relationships between variables when outcome groups are known, these models can be extended to settings in which researchers wish to predict groups in a new sample using an already-trained model (Steyerberg, 2019). A well-trained model can be used to predict outcome groups in new data that

could allow for early identification of individuals in need of educational or psychiatric intervention. For example, VanDerHeyden (2013) describes the importance of effective identification of classrooms in need of various academic and/or behavioral intervention programs. In these situations, a set of factors is often known (e.g., number of referrals, student- and classroom-level grades), but considered in a multivariate context, it is much more difficult to determine whether a single classroom is in need of academic or behavioral intervention. An effective classifier trained on the various features of several different classrooms that did or did not require intervention would then be, ideally, adept at predicting whether a newly presented classroom would be in need of intervention. Similarly, Mann et al. (2008), among others, describes a need to accurately predict whether individuals are at risk of attempting suicide. In this situation, it is much more likely for individuals to not be at risk than the inverse, therefore it is paramount that those at risk in particular be identified with a high degree of accuracy. In these settings, sufficiently accurate prediction models are necessary else the outcomes be deleterious to the individuals and groups involved.

A secondary factor germane to the present investigation considers a feature of the above example: Within each classroom, student-level factors are also present. Presently considered is the multilevel data structure in which cases are nested within higher-level units or clusters (Luke, 2019). While not all data in the social sciences follow a multilevel structure, this condition does present somewhat frequently—in both cross-sectional and longitudinal paradigms—and carries with it numerous considerations beyond those in single-level data that can be problematic for many simpler analytical frameworks. A classic exemplar of this structure is Raudenbush and Bryk's (2002) motivating example of students being situated within classrooms and those classrooms being nested within schools (thus resulting in a three-level structure). This nesting structure aligns with the classical developmental theoretical model of Bronfenbrenner's

bioecological model/ecological systems theory (Bronfenbrenner & Ceci, 1994; Bronfenbrenner & Morris, 2006) in which individuals are nested within and affected by multiple systems ranging from the microsystem (immediate context; e.g., classroom) to the macrosystem (larger overriding and implicit context; e.g., sociocultural values and customs). Bronfenbrenner and Morris (2006) argue that all individuals are inexorably entangled within multiple contexts and simultaneously affect and are affected by these contexts. These researchers also note that research—and by extension data—is inherently multilevel in nature due to these bidirectional contextual entanglements. Despite this not being the case in all social science data, situations in which data are obtained within a nested structure, an alternative analytical framework must account simultaneously for these factors.

An additional area that has been examined with respect to mixed effects regression models, but not mixed effects classifiers, is sample size; that is, both the number of clusters as well as the number of units within each cluster. This is a particularly prominent component when considering the comparison of fixed and mixed effects classifiers as fixed effects classifiers do not account for the cluster within which a case is situated, and thus interpret a given dataset as having a sample size of the total number of clusters * the number of cases per cluster (Allison & Waterman, 2002). Despite the sample size recommendations previously proposed for mixed effects regression models, the effects of sample size on classification accuracy (particularly when only considering raw accuracy metrics) are yet unknown.

The present investigation examines the intersection between these two analytical frameworks—classification and multilevel modeling—with respect to a small sample size. While traditional classifiers in the fixed and mixed effects frameworks—logistic regression (LR) in the single-level and the generalized linear mixed model (GLMM) in the multilevel—have been frequently used in classification settings explanatory and predictive in nature within the social

sciences, various areas of machine learning research have yielded substantive advances in model construction in both regression and classification. However, while myriad studies have considered many of these novel frameworks for both classification and regression, few have fully compared and examined the relative efficacy of fixed and mixed effects models under various conditions unique to classification with nested data structures. Further, fixed effects models may perform comparably to or greater than mixed effects classifiers in predictive contexts under certain conditions (Kilham et al., 2019; Speiser et al., 2019). Specifically, Speiser et al. (2019) found that when comparing a standard random forest (RF) classifier to their novel binary mixed model random forest (BiMM forest) model, the RF model performed at least as well as the BiMM forest and was only marginally outperformed under some conditions. Similarly, Kilham et al. (2019) found that when including both level-1 and level-2 predictors, a standard RF model could outperform GLMM despite functionally ignoring the nesting structure of the data in prediction contexts.

The current study, then, seeks to examine the relative efficacy of fixed effects LR and RF models to that of GLMM and Hajjem et al.'s (2014) mixed effects random forest (MERF) algorithm in both simulated and archival data contexts. The initial simulation study will systematically vary the conditions under which the data were generated across several iterations to compare the predictive classification accuracy of all four models. The simulation study will allow for an examination of the conditions under which prediction accuracy is optimal or non-optimal for each of the models with a particular focus on conditions with smaller sample sizes (among others). These findings will then be used to inform the hypotheses for the archival data analysis, which utilizes the publicly available version of the Program in International Student Assessment (PISA) dataset to predict whether students will be held back in school based on a number of salient individual-level (level-1) predictors (described in greater detail in Chapter 2).

The multilevel structure of this dataset lends itself to examination in the present context and will provide an applied example of the findings from the simulation study. This internal validation will allow for more robust conclusions to inform researchers' and methodologists' regarding the type of model to employ in predictive classification contexts with multilevel data.

The structure of the present dissertation employs a principal simulation study to determine the most salient factors affecting model predictive efficacy followed by an examination of an existing large-scale dataset (the PISA dataset) for the purposes of extending the findings of the simulation study to an applied forum. This structure replicates previous approaches to studying multilevel analytical frameworks and methodologies. Chapter 2 details the literature base synthesized to explain the rationale for the present investigation, the factors considered within each component of the two studies, and the models compared in both studies. Chapter 3 then describes the methodology utilized within the studies including the manner in which the simulation is constructed, the method of investigation into the PISA data in the applied examination, and the hypotheses of both studies. Chapter 4 will describe the key results found from both the simulation and archival studies. Chapter 5 will synthesize the results described in Chapter 4 for the purposes of providing methodological recommendations and conceptual meaning for researchers; the limitations and future directions of this line of research will also be discussed. Chapter 5 will also encapsulate the principle findings and their resulting implications for researchers within and without the social sciences.

CHAPTER 2: LITERATURE REVIEW

The introduction to this dissertation discussed the principle purposes of classification—those being explanation or prediction—with an emphasis on the use of classifiers for the prediction of new cases' membership into one of two outcome groups. The example offered in the present case, particularly given the use of the PISA dataset, is that of student retention (whether students were or were not ever held back a grade in school) and the conditions inherent in the complexity of such prediction tasks. Consequently, several key components of the present study must first be discussed prior. The current chapter begins with an exposition of the relevant data conditions informing model selection and affecting model accuracy rates before an expansion upon the algorithms employed within the study and the mathematical foundations underlying each LR, GLMM, RF, and MERF. The outcome metrics of relevance in the present studies are also discussed. An explanation of the PISA dataset and the relevant variables follows in order to describe the predictors used within the present study.

Nested Data

Phenomena within the social sciences often carry with them a distinct context in which they occur. Theoretical models such as Bronfenbrenner's bioecological model/ecological systems theory (Bronfenbrenner & Ceci, 1994; Bronfenbrenner & Morris, 2006) or Gebbie et al.'s (2003) model of multiple levels of public health determinants illustrate that factors at various structural levels influence individuals. That is, individuals exist within multiple contexts simultaneously, each of which interacting with the others; a common example of this is students situated within classrooms, which are situated within schools, which are situated within districts, etc. (Raudenbush & Bryk, 2002). This situation of cases or individuals within higher-level units creates the nesting structure where individuals are nested within higher-level units. Alternatively, individuals may be measured at multiple time points, thus situating each time point within the individual. It is therefore important to consider the effects of both the individual and the

collective units, clusters, or groups in measurement and analysis (Luke, 2019). Given the nesting structure, data must include variable labels for group or cluster membership, a case/individual identifier (in the case of longitudinal data), and information regarding either or both the individual/time point level as well as the individual/cluster level.

As an example, consider Raudenbush and Bryk's (2002) example within a classroom context: Students' academic growth is measured over time with the goal of determining the growth pattern as well as the effects of individual trait characteristics and classroom environment on those growth patterns. In this case, the student is the level-2 unit with the time point as the level-1 unit and the classroom as the level-3 unit; student trait characteristics and classroom characteristics are unlikely to change at each time point and, thus, constitute static, time invariant, contextual variables. Measurements of student performance at each time point, then, becomes the time-varying outcome, thus making time an additional variable to consider. The resulting dataset, then, is structured in three different levels and can be analyzed as such.

When data are collected under such conditions to gain an understanding of both the individual-level and higher-order cluster effects, an alternative analytical framework—multilevel modeling—is often employed to account for the effects of these various levels of measurement. Fundamental statistical assumptions including case independence, variable normality, and homoscedasticity underlie many commonly used models (Tabachnick & Fidell, 2018). In the case of multilevel measurement, however, individual scores within the same cluster are likely to be correlated with one another (e.g., students in the same school are more likely to score similarly to one another than they are to students in a different school; one student's academic performance scores over time are more likely to correlate with with themselves more than they are with another student's performance scores), thus violating the assumption of case independence. This correlation is identified as the *intraclass correlation* (ICC) and is calculated

as the proportion of the variance in the dependent variable accounted for due to the cluster membership; thus, the ICC is represented as

$$\rho = \frac{\sigma^2_{u_0}}{\left(\sigma^2_{u_0} + \sigma^2_r\right)}$$

where

$\sigma^2_{u_0}$= Estimate of level-2 (cluster-level) variance

σ^2_r= Estimate of level-1 (case-level) variance

(Luke, 2019).

Consequently, multilevel models account for this correlation and estimate effects of the higher-level clusters on the outcome variable. Gully and Phillips (2019) note that these effects are likely to manifest in a number of different ways as contexts are often not well-separated, many individuals may be members of multiple clusters at multiple different levels (e.g., Milliren et al. [2018] consider students situated within schools and neighborhoods, yet not all students in the same neighborhoods attended the same schools nor did all students who attended the same schools live in the same neighborhoods). Consequently, it is imperative to consider how phenomena are measured and the research questions sought to be answered as these factors dictate the nature and depth of the nesting structure of a dataset.

Relevant Data Conditions

The present study considers a number of key conditions relevant to and exhibited by multilevel data structures. Based on previous literature within the domains of both multilevel analysis and classification, the conditions of cluster size (number of cases per cluster), number of clusters, intraclass correlation, random effect specification (whether random coefficients are specified or only random intercepts), group size ratio (the ratio of cases in outcome group 1 to cases in outcome group 0), and number of predictors. Studies considering each of these

conditions are discussed below to illustrate the impetus behind the manipulation of these factors in the simulation component of the present study. Additionally, many of these factors will be examined prior to the commencement of the PISA archival data analysis component of this study to compare results from the simulation to the efficacy of these models with real data.

Cluster size. Cluster size—or the number of cases per level-2 cluster—is a key component within multilevel data due to its status as one of two segments underlying the total sample size in model parameter estimation. Many previous studies in both the classification and multilevel modeling literature bases have concentrated on the sample size recommendations and optimal sample sizes for various model architectures. Studies in the multilevel modeling literature base demonstrate a somewhat lesser concentration on cluster size compared to the number of clusters, yet as the total sample size corollary below illustrates, both are salient factors in the consideration of predictive efficacy of both classification and regression models. It should be noted that in the longitudinal setting, individuals often form the level-2 unit and time points act as the level-1 analytical unit. However, when considering this parameter, the term "cases" is utilized as a general term encompassing both cross-sectional and longitudinal multilevel analytical paradigms.

Previous literature has broadly illustrated that as the number of cases per cluster increases, so too does model accuracy. Speiser et al. (2019, 2020) illustrated that across all fixed and mixed effects models, increasing the number of time points from two to three, and ultimately to six yielded incremental increases in model predictive accuracy. A similar result was found by Ngufor et al. (2019) when considering the number of clinical visits per patient with accuracy measures increasing marginally when the number of visits increased from one to four. Conversely, Hajjem et al. (2014) found that in the regression context as the cluster size increased

from ten to 50, only mixed effects model increased in accuracy; fixed effects models simultaneously decreased appreciably in accuracy.

Maas and Hox (2005) also note that a cluster size of approximately 30 may be commonly found in educational research while a cluster size of five is commonly found in longitudinal and family-based research. Further, Hox (1998; 2010) asserted that clusters of size 20 (with 50 clusters) may be appropriate for yielding cross-level interaction effects (interactions between predictors at different levels). Consequently, the present simulation study utilizes cluster sizes of 10, 20, and 50 in order to empirically consider the accuracy discrepancies between these conditions. As yet, only one study (Mangino & Finch, 2021) has considered the effects of cluster size on classification accuracy in the multilevel context, though this study utilized many larger cluster sizes than those discussed above. Further, as emphasized in Milliren et al.'s (2018) study, cluster sizes may vary dramatically in nested data contexts.

Number of clusters. The second, and arguably more eminent, component of the nested data sample size decomposition is the number of level-2 units. Consider Raudenbush and Bryk's (2002) example scenario of students situated within schools; the number of clusters in this case is the number of schools each with a cluster size equivalent to the number of students from whom data were collected within each school. In the longitudinal context, then, the level-2 cluster would be the individual participant and the cluster size would correspond to the number of time points at which measurements were obtained from each individual. A commonly-cited benchmark for multilevel analysis sample size recommendations pertaining to the number of clusters is attributed to Kreft (1996) with a recommendation of an approximate minimum of 30 clusters. However, several studies have since disputed this recommendation through their examination of the standard GLMM framework as well as Bayesian estimation techniques and fixed effects models (e.g., McNeish & Kelley, 2019; McNeish & Stapleton, 2016). In particular,

McNeish and Stapleton (2016) illustrated that with as few as ten clusters, properly-specified models could yield sufficiently useful results, despite being underpowered; they further illustrated that with a small number of clusters, the use of fixed effects models (estimated using ordinary least squares [OLS] rather than maximum likelihood [ML]) could increase power at the expense of maximally accurate model specification of random effects. Alternatively, in a simulation study exploring in binary response models, Paccagnella (2011) illustrated that accurate parameter estimates for multilevel models could only be achieved when the number of clusters is 50 or greater, thus minimizing coefficient estimate standard errors.

Broadly, the pattern found in studies manipulating the number of level-2 clusters mirrors that of those examining the effects of increasing cluster size, as described above. Crane-Droesch (2017) illustrated that when increasing the number of level-2 units from 900 to 1800 and 2700, the panel neural network model (a multilevel neural network framework) used decreased appreciably in error. Further, in an applied example using data across each Australian territory, Downes and Carlin (2020) found that while biases for the Northern and Australian Capital Territories were similar to those of New South Wales and Victoria, the former territories' bias estimates yielded substantially larger standard errors coinciding with their respective population differentials.

The literature reviewed presently illustrates that increases in the number of clusters—like the increase in cluster size—results in improved model accuracy. However, only the study by Paccagnella (2011) considered this in the context of a binary classification task and, as such, requires consideration presently. Given the literature presently reviewed, conditions of 10, 30, and 50 clusters are considered in the simulation. Therefore, the data conditions described in Chapter 3 follow recommendations and findings by the literature presently reviewed.

Corollary: Total sample size. When considering the nature of the present study, the factors of both cluster size and number of clusters becomes of lesser importance due to the nature of the fixed effects classifiers being used. Fixed effects classifiers do not allow for the specification of random effects, thus meaning that one of two circumstances occur: Either the model is specified using the cluster identifier as a fixed effect (thus making all classifiers fixed effects models *a la* McNeish & Kelley, 2019), or the cluster identifier could simply be ignored. In the latter case, each case, irrespective of cluster, would be treated as its own unique data point (in the former, the variance due to cluster membership would be accounted for, but not made interpretable). Consequently, the fixed effects classifiers being fit would not feature a disaggregated sample size of cluster size and number of clusters, but instead would simply feature a total sample size calculated by cluster size * number of clusters (Allison & Waterman, 2002). For example, Cameron and Trivedi's (1998) illustration in the context of fitting a Poisson fixed effects model featured a dataset of the number of patents across five years (level-1 units) for each of 346 firms (level-2 units). The sample size of this dataset when used in a fixed effects model was calculated as 346 firms * 5 years for a total of 1730 unique records; the resulting model featured 345 dummy coded variables in addition to specified predictors. Therefore, while the data were nested in nature, the disaggregation of cases per cluster and number of clusters was eschewed in exchange for a holistic sample size consisting of abstracted *cases* or *instances.*

It should be noted that in the the present study, both fixed effects models will incorporate cluster identifiers and thus will require dummy coding of these cluster identifiers for sufficiently accurate fitting of the LR model. Conversely, while RF is still considered a fixed effect model, the parameter estimation method does not rely on dummy coding of cluster identifiers, but rather can include raw categorical variables in estimation. This process will occur for both the simulation study and archival data examination. Consequently, while the conceptually accurate

representation of the data will be cases within clusters, the fixed effects models will simply incorporate the cluster identifier as another predictor (in the case of RF) or set of dummy coded predictors (in the case of LR) to yield parameter estimates. The resulting sample sizes for both models, then, will be represented as the product of the cluster size and number of clusters, as in Cameron and Trivedi's (1998) and Allison and Waterman's (2002) examples.

One key item of note in the present study is the consideration of small sample sizes. Given the literature base, and as will be discussed in Chapter 3, the smallest cluster size (10 cases) and smallest number of clusters (10 clusters) results in a total sample size of 100 cases (10 clusters * 10 cases per cluster). As yet, no research has been conducted focusing on sample size for *mixed effects* classifiers, but a robust literature base exists for fixed effects classifiers and mixed effects models, independently, used with small samples. Beleites et al. (2012) noted that when using a discriminant function analysis (DFA) classifier, a test set of 75 cases of greater consistently yielded sensitivity values greater than 90% when models were trained on an initial sample of 25 cases per class while Raudys and Jain (1991) indicated 50 – 100 cases per outcome group. Further, Figueroa et al. (2012), across 568 experiments, noted that a sharp decline in root-mean-square error (RMSE) was prominent when increasing sample size from 80 to 200 cases before leveling off at sample sizes greater than 200. However, Huang and Li (2011) found that little difference existed between naive Bayes classifier accuracy in sample sizes of 100 and 1200. To compensate for small sample sizes, alternatives such as model internal cross validation processes and bootstrapping are recommended by some practitioners (Beleites et al., 2012; Martin & Hirschberg, 1995).

The literature on classification with small samples brings into question two areas of inquiry: The size of the training set and the computed error of the model (e.g., RMSE). Presently, only the former is of interest as it is in raw prediction accuracy that this study focuses. It should

be noted, however, that while computed model error rates may increase, classification accuracy may not substantially diminish. Given that the smallest total sample size in the present study is smaller than the recommendations of Figueroa et al. (2012), Beleites et al. (2012), and Raudys and Jain (1991), it is likely that model error metrics would increase, but without a concomitant decrease in outcome group prediction accuracy. Compared to contexts in which a larger sample size could be obtained (referring to the number of clusters * number of units per cluster), it is likely that while raw classification metrics may not decrease appreciably for smaller sample sizes, computational error metrics (e.g., RMSE) may; it is also likely that standard errors of parameter estimates may increase, thus limiting the explanatory power of any such model. Maas and Hox (2005) discuss that total sample sizes of 900-1500 (30-50 clusters with 30 level-1 units) are common in educational settings while sample sizes of 120-250 (30-50 clusters with five level-1 units) are typical in longitudinal designs and family-based research. The smallest total sample size in the present investigation is 100 (10 clusters with 10 units each), thus representing a sample smaller than that typically considered appropriate or recommended for mixed effects models. Therefore, the sample size proves to be a salient consideration and delimiter within the present study as little is yet known about the effects of small sample size on mixed effects classification models, particularly when compared to their fixed effects analogues.

Intraclass correlation. The ICC is a fundamental hallmark of multilevel models as it illustrates, mathematically, precisely the reason such models are utilized in many analytical contexts, namely the impact of cluster membership on the variance of the outcome variable distribution. A frequently cited *de facto* benchmark warranting use of multilevel models over fixed effects models is LeBreton and Senter's (2008) ICC $\geq$ 0.05 criterion; in the event that the ICC exceeds this 0.05 benchmark, LeBreton and Senter state this as sufficient evidence of a

cluster membership effect and warrants additional consideration for model selection (employing a fixed- or mixed-effects model).

While not directly considered as the ICC, Speiser et al. (2019, 2020) considered simulated data with "small" and "large" random effects. In their studies—which focused on the instantiation and illustration of their binary mixed model tree and forest frameworks—the cluster means were fixed at 0 with standard deviations of either 0.1 (small random effect) or 0.5 (large random effect). This could be interpreted as either more or less homogeneity between clusters. Consequently, in cases of large random effects (high cluster heterogeneity), standard RF tended to perform identically to the simplest BiMM forest model, generally outperforming other multilevel models like frequentist and Bayesian GLMMs. This result was consistent across sample sizes (number of clusters) of both N = 100 and N = 500 (with cluster sizes of 2, 4, and 7), thus demonstrating that when the between-cluster variability is higher, the ICC decreases (see discussion of this reciprocal relationship below, and in Chapter 3 within a preliminary simulation for parameter determination), thus lessening the need for multilevel models. The findings of Speiser et al.'s (2019, 2020) studies are key due to their use in a classification setting rather than a more typical regression setting.

A similar effect was found in Hajjem et al.'s (2017) examination of their generalized mixed effects random tree (GMERT) algorithm whereby all tree-based models decreased appreciably in error when random effects were more pronounced. However, this result was only found in situations of random intercepts, but was reversed when random coefficients were specified as well. An earlier study by Hajjem et al. (2014) illustrated a clear diminution of model accuracy in a regression context as the ICC increased; this effect was less pronounced in mixed effects models (including MERF) compared to fixed effects models, including RF.

It should be further noted—as alluded to in the discussion of Speiser et al.'s (2019, 2020) studies using the magnitude of the random effects rather than an explicit ICC—that the consideration of the ICC allows for a derivation of the *interclass correlation*, which could be interpreted as $1 - \rho$, or the proportion of variance in the outcome variable *not* due to cluster membership. This value thus holds an inverse relationship with the ICC whereby an increase in ICC results in a concomitant decrease in the interclass correlation and, thus, the converse. Therefore, an alternative interpretation of the interclass correlation is the proportion of variance in the outcome variable due to individual cases rather than clusters. As clusters become more heterogeneous, the ICC decreases and models exhibit less error in parameter estimates and predictions.

Consideration of the ICC is important in classification contexts (specifics of classification analysis discussed below) with the likelihood of biased parameter estimates resulting from ignoring this component (Moineddin et al., 2007). However, studies by Speiser et al. (2019, 2020) and Kilham et al. (2019) have illustrated that conditions may exist under which RF classifiers may perform comparably to multilevel classifiers, despite the absence of fixed effects classifiers' abilities to account for this factor. Therefore, while the magnitude of the hallmark characteristic of nested data—the ICC—substantially affects the accuracy of mixed effects models, this effect may not be as pronounced or as deleterious in the case of classification analysis. It is precisely this paradoxical occurrence that provides the impetus for the present examination of both simulated and archival data.

Group size ratio and misclassification. One key factor unique to the classification context is the group size ratio, the number of positively identified (Group 1) to negatively identified (Group 0) cases of the outcome variable. The "Group 0/1" terminology is used in the context of the simulation study, but in the PISA examination, the language may be changed to

reflect the outcome. In this case, positively identified cases are those who were retained (identified with a ‘1’ in the dataset) and negatively identified cases are those who were not retained (identified with a ‘0’ in the dataset). The ratio of cases in each group to one another has long been a major consideration in the classification literature with nearly all classifiers having an appreciable bias toward the larger of the groups (in the present example, the non-repeater group). That is, in situations of unequal group sizes (identified in some of the literature as ‘unbalanced datasets’), cases within the larger group are often classified correctly whereas cases in the smaller group are often misclassified into the larger group as a *de facto* default of many classifiers (Bolin & Finch, 2014; James et al., 2013; Lei & Koehly, 2003). These class imbalances often manifest in negative-to-positive case ratios of 8.5:1 (as in the case of retention in the PISA dataset) to examples as extreme as 29:1 or even 100:1 (Muchlinski et al., 2016; Yan, 2019). As the ratio increases, overall model accuracy may increase toward >99%, but so too does the likelihood of misclassifying positive cases (smaller group) as negative cases (larger group; Eisenstein, 2019).

Many studies to date have considered this condition and noted that under these conditions, all models tend to diminish in SGR while simultaneously increasing appreciably in overall accuracy (Chawla, 2009; Hoens & Chawla, 2013). Numerous methods of synthetic oversampling of the smaller group or bootstrapped oversampling techniques (e.g., random over-sampling examples [ROSE] or synthetic minority over-sampling examples [SMOTE]) have been proposed and demonstrated effective in situations of unbalanced data (Chawla et al., 2002; Menardi & Torelli, 2014). However, these options are oftentimes not standard practice for social science researchers due to these methods being somewhat more obscure in the social sciences; rather, researchers may consider alternatives to the more typical classifiers as LR or DFA. Consequently, the class imbalance problem stands as a substantive limitation in social science

research, particularly in settings of predictive classification. The choice of model is key when outcome group sizes are unequal due to the possible stakes of misclassifying individuals in one group or another. Consider, for example, VanDerHeyden's (2013) discussion of the corollaries of misclassifying classrooms as being high-achieving when they are, in reality, low-achieving and in need of intervention. In this case, a false-negative result would be more detrimental than would the converse with students in great need of intervention services not receiving them. Another setting in which this group size imbalance is likely to result in deleterious outcomes is in the prediction of suicide attempts. The number of individuals at risk of attempting suicide is much lower than those not at risk (a 10:1 ratio of those not at risk to those at risk; Centers for Disease Control and Prevention, 2018) and, consequently, many predictive classifiers tend to perform dismally in accurate identification of those at risk (Mann et al., 2008). Therefore, the group size ratio acts as a key consideration in a comparison of predictive classifiers and provides the impetus behind using SGR and LGR rates rather than overall accuracy rates. This factor is key in all classification settings regardless of data structure.

Number and type of predictors. A final data feature pertains more to the structure of the models specified, namely the number and type of predictors used within the model. A well-known phenomenon in the statistical learning field pertains to the bias-variance trade-off, or the degree to which models overfit or underfit the data and limit generalizability of the model (Hastie et al., 2009; Kohavi & Wolpert, 1996). Many simulation studies tend to use low-dimensional data generation processes—aligning most prominently with the data obtained in social science research contexts—simulating datasets with two (Maas & Hox, 2005), five (Crane-Droesch, 2017; Finch et al., 2014; McNeish & Stapleton, 2016), eight (Sela & Simonoff, 2012), or nine (Hajjem et al., 2014) predictors, and only Capitaine et al.'s (2019) study actively manipulated the number of predictors. Oftentimes, exploratory simulation studies tend to use

fewer predictors to maintain parsimony. For example, Lavery et al. (2019) considered the effects of multicollinearity—overlap in the variance accounted for by each individual predictor—on Type I and Type II error rates; using conditions of two, four, and six predictors, they found that increased collinearity appreciably increased the likelihood of a Type II error. In the multilevel literature, multiple predictors are used in model construction (often mirroring or approximating the structure and features of an accompanying archival dataset), yet this is seldom a manipulated condition.

The bias-variance tradeoff is eminent in contexts of high-dimensional data (where the number of predictors is greater than the number of cases; that is, where $p >> n$) whereby the number of predictors used allows for highly accurate models in training, but are unable to generalize to new data without a substantial loss in accuracy. Capitaine et al. (2019) demonstrated that MERF outperformed singular multilevel tree-based algorithms in regression contexts for predicting gene expression (a context notorious in and synonymous with the bias-variance problem) where $n = 17$ and $p = 6$ (low-dimensional) and $n = 17$ and $p = 8000$ (high-dimensional). In Capitaine et al.'s (2019) study, the singular tree models increased immensely in error when moving from the low- to high-dimensional prediction context whereas the RF models' error increased only marginally.

In addition to the number of predictors, the significance, magnitude, and correlations among predictors should be considered. Studies by Ngufor et al. (2019) and Capitaine et al. (2019) considered complex nonlinear data generation processes whereby data were simulated and drawn from a multivariate normal distribution, but were subject to nonlinear transformations to mimic their respective source datasets (for example, Capitaine et al. [2019] simulated data mimicking those in a vaccine trial dataset and featured quadratic and cosinor transformations for level-1 predictors). Regarding the predictor intercorrelations, Hajjem et al. (2014) considered

two conditions of predictor intercorrelations among the nine predictors simulated: A decorrelated condition and a condition where predictors held a 0.4 correlation with one another. Little difference was found in model error rates between these conditions. Alternatively, Finch et al. (2014) set all predictor intercorrelations among the five variables to constant values aligning with those found among subscales within the Wechsler Adult Intelligence-III, ranging from 0.36 – 0.76. Further, Downes and Carlin (2020) systematically varied the level of the predictors used and found varied results depending on the subset of the full sample used. While many previous studies discussed above have considered the number of predictors, only two have noted the level of predictors, and no studies have actively manipulated the level of predictors. Consequently, the present study seeks to examine whether a difference exists in whether level-1 or level-2 predictors are used, as this is an as-yet unexplored area.

It is further noted that in the case of the fixed effects models accounting for cluster identification, the number of predictors will be $p + 1$ for RF (due to its ability to include multicategorical predictors without dummy coding) and $p + (k - 1)$ for LR (due to its requirement of dummy coding of each cluster identifier; where k = number of level-2 clusters). Given the literature reviewed presently and the paucity of clear guidance on predictor magnitude and level, particularly when considering multilevel classification, the present study will hold constant the number of predictors and their magnitudes (*a la* Finch et al., 2014) while manipulating their level. To maintain parsimony, only linear effects will be simulated with variables drawn from a multivariate normal distribution; all will be significant and specified at a moderate magnitude effect (see Chapter 3 for specification).

Supervised Classification Analysis and the Use of Multilevel Models

In the present case, the problem is one of classification rather than regression (as is the case in much of the literature on mixed effects models) due to both the paucity of literature in

this area and the practical decisions being made (e.g., intervention support) based on binary classifications. This presents a unique case of multilevel modeling and, thus, follows that the form and function of the classification problem must be discussed. Broadly speaking, classifiers generally seek to obtain the probability that each case *i* belongs to outcome group *k* given *p* predictors (Hastie et al., 2009). In the specific case of supervised classification problems, the outcome group membership for each case in the training data (the data used to train classifiers and yield parameter estimates) is known and are constructed for one of two fundamental purposes: Explanation or prediction (Steyerberg, 2019). Explanation models utilize only the training data to yield parameter estimates for the purposes specifically of conceptual interpretation. It is for this purpose that the most typical fixed and mixed effects classifiers LR and GLMM are used as they yield interpretable coefficient estimates to determine the most salient predictors of group membership. Prediction models, conversely, focus less on the interpretation of these parameter estimates and substantially more on the utility of classifiers in predicting group membership for new cases, those within the test data (new data with previously unseen cases for use in cross-validation procedures). The present study focuses on this latter purpose with the goal of assessing model efficacy for making predictions under varied multilevel data conditions.

One important characteristic of note is the change in the variance decomposition of multilevel data with dichotomous outcomes. Raudenbush and Bryk (2002) illustrate this using a special case identified as the Bernoulli distribution in which outcomes are labeled as either 0 or 1. As a function of this shift in outcome variable distribution, the ICC must be estimated using an alternative method due to the distributional differences between dichotomous and continuous outcomes. Rather than the standard ICC calculation, as illustrated in the equation below, the ICC then takes the form

$$\rho = \frac{\sigma^2_{u_0}}{\left(\sigma^2_{u_0} + \sigma^2_r\right)}$$

where

$\sigma^2_{u_0}$= Estimate of level-2 (cluster-level) variance

σ^2_r= Estimate of constant level-1 (case-level) variance with the form$\sigma^2_r = \frac{\pi^2}{3}$

thus resulting in the full form

$$\rho = \frac{\sigma^2_{u_0}}{\left(\sigma^2_{u_0} + \left(\frac{\pi^2}{3}\right)\right)}$$

(Moineddin et al., 2007).

Despite this additional component for model estimation within the multilevel context, classifiers using both fixed and mixed effects frameworks still seek to yield parameter estimates best predicting outcome group membership (Guo & Zhao, 2000).

As the case may be, many "big data" and classical machine learning settings—such as phishing email detection, image classification, or credit risk modeling (Abu-Nimeh et al., 2007; Palvanov & Cho, 2018; Wu & Zhang, 2010; Zhang & Haerdle, 2010)—multilevel classifiers are seldom used in exchange for complex or "black box" classifiers such as neural networks or Bayesian additive regression trees, for example. These classifiers ignore the nesting structure of the data, instead basing classification decisions [often] on a large number of predictors (features); for example, Abu-Nimeh et al. (2007) use 43 predictor variables to estimate whether emails are legitimate or phishing. With these and other fixed effects classifiers being among the most prominent in the data science and machine learning fields, the question of under what conditions a multilevel classifier should be used is called into question.

A similar argument in favor of parsimony was leveraged by McNeish et al. (2017) via their study's demonstration that population-averaged models are both more interpretable and less complex while still answering the research questions within multilevel data collection and analysis settings. McNeish et al. (2017) argued that the ubiquitous implementation of multilevel modeling (referring specifically to the classical hierarchical linear modeling approach, an alternative conventional name for GLMM in the present examination) led to unnecessary complexity and increased the probability of error in model fitting and interpretation. Similarly, McNeish and Stapleton (2016) illustrated that in regression contexts, using fixed effects models yielded significantly more power in analyses with small sample sizes, outperforming both classical GLMM and Bayesian approaches while simultaneously preserving simplicity (though, at the expense of being able to interpret the level-2 random effect variance estimates). Despite not estimating level-2 effects directly—but rather simply controlling for cluster membership—fixed effects models demonstrate substantially more simplicity and require only careful coefficient interpretation on the part of the researcher/analyst to properly articulate the difference between level-1 and level-2 predictors (McNeish & Kelley, 2019).

Previous Comparisons Between Algorithms

Regarding the present argument leveraged from Speiser et al.'s (2019) and Kilham et al.'s (2019) findings favoring RF over multilevel classifiers, the question of how these models compare becomes relevant. Coupled with Ngufor et al.'s (2019) findings that mixed effects models (including Hajjem et al.'s [2014] mixed effects random forest [MERF] and GLMM) generally outperform fixed effects classifiers (including RF), the evidence is as yet inconclusive. However, when considering classification contexts in the machine learning domain, seldom are mixed effects models actively used, often eschewed in favor of the "black box" methods noted above (Abu-Nimeh et al., 2007; Palvanov & Cho, 2018; Wu & Zhang, 2010; Zhang & Haerdle,

2010). Further, researchers such as McNeish and colleagues (McNeish & Kelley, 2019; McNeish & Stapleton, 2016; McNeish et al., 2017) argue in favor of more parsimonious models such as population averaged or fixed effects models. Fixed effects models are often used when the number of higher-level clusters is small (typical guidelines of 30 clusters were derived from Kreft's unpublished, but often-cited, 1996 study) and simply include the cluster identifier variable as dummy coded predictors and serve to account for the variance due to cluster without explicitly estimating the variance decomposition (as in the case of GLMM). Fixed effects models, then, act as a method for controlling for the variability due to cluster membership while retaining the simplicity of interpretation social science researchers and methodologists often seek. However, no studies to date have explicitly compared fixed effects RF models to fixed effects LR or any mixed effects models; while Speiser et al. (2019) and Kilham et al. (2019) did make comparisons between fixed and mixed effects RF classifiers, they did not specify that RF was constructed as a fixed effects model.

In many studies comparing multilevel models, it has been common to utilize a fixed or mixed effects model as a comparison against which the multilevel model can be assessed. The present case, however, employs a novel method in the form of MERF and, consequently, has fewer studies considering its efficacy relative to fixed effects models and simpler multilevel classifiers. Hajjem et al. (2014) instantiated MERF for the purposes of regression and have not since assessed it for the purposes of classification. Mangino and Finch (2021) employed Hajjem et al.'s MERF framework for the purposes of predictive classification and found it outperformed GLMM (and several other more complex mixed effects models) under many different conditions including those of differing sample sizes and ICCs. Speiser et al. (2019) postulated an alternative multilevel RF framework in the form of the BiMM forest model. The BiMM forest framework acted as the binary classification extension of a multilevel RF framework with accuracy

generally comparable to that of the fixed effects RF algorithm. However, the BiMM forest algorithm is not yet available in commonly used statistical software package, whereas MERF is via Ngufor's (2019) `Vira` package. Hajjem et al. (2014) demonstrated that in the regression context, MERF appreciably outperformed RF and GLMM. Further, Kilham et al. (2019) in the context of tree-level harvest predictions, found that when simultaneously accounting for tree-level (level-1) and plot-level (level-2) predictors, RF outperformed GLMM with substantially less requirement for proper model effect specification and while preserving model interpretability. That is, Kilham et al. (2019) concluded that while RF did not provide a detailed consideration of plot-level effects, predictions were equal to or more accurate than those obtained from GLMM. Additionally, RF models are trained algorithmically rather than the *a priori* specifications of more traditional models such as GLMM.

Considering fixed effects RF compared to LR, nearly all comparisons of classifiers—particularly in cases of unbalanced data and in predictive contexts—RF broadly outperforms LR. For example, when predicting civil war onset in an unbalanced dataset (the ratio of peaceful to bellicose years was approximately 100:1, even more extreme than the retention grouping variable in the PISA dataset at approximately 8.5:1), Muchlinski et al. (2016) found that RF yielded substantially more accurate predictions with less model specification required. It was hypothesized that because of RF's (and tree-based models, broadly) flexibility and ability to model complex interactions and non-linear relationships, it was able to account for the variety of variables and relationships that would otherwise need to be specified *a priori* in an LR model. Similarly, a thesis by Yan (2019) illustrated that RF outperformed other tree-based classifiers when data were unbalanced at an approximately 29:1 ratio, thus making it a more prominent candidate for consideration than other tree-based methods. Additionally, in the context of calculating propensity scores, Westreich et al. (2010) noted key limitations of LR, namely the

requirement of proper model specification and assumptions of linear relationships among variables. Alternatively, Westreich et al. (2010) discuss the benefits of tree-based methods while specifically favoring RF as a viable and preferred alternative. Further, when predicting dementia diagnoses using a number of classical and novel statistical and machine learning models—including RF, LR, neural networks, support vector machines, and CART, among others—Maroco et al. (2011) found that RF performed the most consistently across a number of key accuracy metrics (including LGR and SGR, as in the present case) compared to more complex machine learning classifiers as well as the classical LR.

Thus far, few studies have compared Hajjem et al.'s (2014) MERF algorithm to other fixed or mixed effects models in either regression or classification contexts. Hajjem et al.'s (2014) original study found that MERF outperformed LR, RF, and GLMM across all conditions in a regression context. However, in many datasets, the difference between RF and MERF only gave marginal favor to MERF over RF and similar performance of RF to GLMM. Additionally, Ngufor et al. (2019) featured perhaps the most comprehensive applied classification comparison to date, comparing a number of fixed and mixed effects classifiers on several datasets predicting longitudinal hemoglobin A1c change. In this study, MERF consistently performed as well as or better than GLMM and outperformed RF in three of the four datasets considered, often only in situations with larger cluster sizes; in many cases, this performance differential was marginal (Ngufor et al., 2019). Further, MERF and GLMM tended to perform similarly to one another in nearly all of the datasets and cluster sizes Ngufor et al. (2019) considered; GLMM also tended to outperform RF in many of the datasets, but again, marginally so. Alternatively, Capitaine et al. (2019) found that MERF consistently yielded substantially lower error than GLMM in simulations using both deterministic and stochastic model construction in the context of high-dimensional data (more predictors than cases). However, similarly to findings by Speiser et al.

(2019) and Kilham et al. (2019), when applied to HIV vaccine trial data, MERF and RF performed comparably to one another with RF holding only marginally larger standard errors (Capitaine et al., 2019).

The established comparisons of these methods illustrate the need for a comprehensive comparison of the contexts and data features under which fixed or mixed effects classifiers may be appropriate or warranted for predictions. Recommendations for classifier use have yet to be established in the social sciences when using nested data. The studies currently reviewed demonstrate that in classification contexts mixed effects models may only have a marginal edge compared to fixed effects models; this is compounded by the fact that no studies thus far have included cluster membership as a fixed effect in the manner of McNeish and Kelley (2019) in order to control for the variability due to cluster membership. In regression contexts, the advantage of mixed effects models is only somewhat greater than that in classification contexts. Between these comparisons and the consistent use of fixed effects models in many classical machine learning classification contexts, recommendations should be made for social science researchers such that they can ensure the most optimally useful and minimally convoluted methods be utilized under all data conditions, including those with nested data structures.

Current Statistical Models

Among the models available in the classical statistical and machine learning literature bases, several fixed and mixed effects models have been employed with varying degrees of success in contexts with non-nested and nested data. The present investigation seeks to explore and identify the relative predictive efficacy of two such fixed effects classifiers—logistic regression and random forests—and two multilevel classifiers—generalized linear mixed effects models and mixed effects random forests. Recent studies by Kilham et al. (2019) and Speiser et al. (2019) were among the first to illustrate the potential superiority of a fixed effects classifier

when compared to its mixed effects counterpart, namely RF. No other classifiers have as yet been compared to their fixed effects analogues in classification contexts and, despite Kilham et al.'s and Speiser et al.'s findings, the conditions under which fixed effects RF may be comparable to a multilevel RF framework are still unclear. The comparison also requires that the more typical LR and GLMM methods be used, as these act as the commonly used fixed and mixed effects frameworks in current use in the social sciences. The following section explains the mathematical and algorithmic underpinnings of each of these models to illustrate their commonalities and divergences.

Fixed Effects Classifiers

Classifiers within a fixed effects (also identified as single-level) framework do not take into account the nesting structure of data, but rather simply perform naive parameter estimation under the assumption of case independence (Tabachnick & Fidell, 2018). The assumption of case independence is foundational to many statistical hypothesis testing and model estimation, including in methods such as LR and DFA classification methods (Tabachnick & Fidell, 2018). However, as previously discussed, nested data structures—whether cross-sectional, longitudinal, or dyadic,—tend to violate that assumption with ICC values greater than LeBreton and Senter's (2008) asserted threshold of ICC ≥ 0.05 and, as such, require multilevel analytic frameworks to account for this correlation. Despite this recommendation, little research has been conducted to determine the efficacy of classifiers when fit with nested data when compared to models fit to continuous outcomes. Consequently, the present investigation makes use of two such fixed effects classifiers as comparisons to their multilevel analogues.

Logistic regression. Logistic regression (LR) has long been the *de facto* default classifier —alongside DFA—in a number of settings due to its commonality across a variety of literature bases, its similarity in interpretation to linear regression, and the simplicity of its implementation

(LR is integrated into statistical software packages such as SPSS by default). Many applied and empirical settings utilize LR as a baseline classifier useful for explanation or prediction and yields marginally more stable estimates than does DFA (Lei & Koehly, 2003; Mesbane & Morris, 1996; Steyerberg, 2019). Functionally, LR operates by calculating the probability of group membership {0,1} based on estimation of a linear function for each input. Following the form of a linear regression model, LR instead estimates these probabilities via the logistic function

$$p(X) = \frac{e^{\beta_0 + \sum \beta_1 X_1 \ldots \beta_p X_i}}{1 + e^{\beta_0 + \sum \beta_1 X_1 \ldots \beta_p X_i}}$$

where

β_0 = Intercept coefficient

β_p = Coefficient estimate for predictor X_i

e = Euler's constant; base natural logarithm ≈ 2.718

(Hastie et al., 2009; James et al., 2013).

The resulting equation could then be interpreted with respect to the estimation of the log-odds (logit) of group membership whereby the above equation can be simplified to

$$\log\left(\frac{p(X)}{1 - p(x)}\right) = \beta_0 + \sum \beta_1 X_1 \ldots \beta_p X_i$$

in order to approximate more directly $\Pr(Y = 1|X)$ (James et al., 2013).

Logistic regression parameters are estimated via maximum likelihood such that the predicted probabilities of group membership for each individual is as closely aligned with their actual group membership, thereby maximizing the likelihood function

$$l(\beta_0, \beta_1, \ldots, \beta_p) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} \left(1 - p(x_{i'})\right)$$

(James et al., 2013).

Despite its commonality, LR is prone to a number of errors, particularly with regards to prediction. Eisenstein (2019) cites the probabilistic estimation of group membership as a benefit of LR, yet also indicates that the method may be "overtrained" and, consequently, may generalize poorly. Similarly, several treatises discuss LR's likelihood of overfitting (that is, 'overtraining' in Eisenstein's terminology) when sample sizes are small and the ratio of cases to predictors is reduced (Hosmer & Lemeshow, 2000; Tabachnick & Fidell, 2018). Further, Tabachnick and Fidell (2018) discuss the assumptions underlying LR including linearity of the predictor—log-likelihood relationship, absence of multicollinearity (overlap in variance accounted for between predictors) and outliers, and independence of errors (residuals). Further, LR requires that nominal predictors be dummy coded in order to be entered into the model; this allows for the accounting of variance due to cluster membership, but not as efficiently or in as interpretable a manner as multilevel models. The present study examines a data condition in which independence of cases—and consequently errors—is not upheld and, as such, LR models are unlikely to perform well. It should be noted that under conditions violating the assumption of independence of errors, the expected frequencies of group classification are likely to be highly variable and, thus, increase the likelihood of a Type I error (false positive; Tabachnick & Fidell, 2018). The result is LR's frequently lower performance when compared to other methods (e.g., Finch et al., 2014; Mangino et al., 2021); the GLMM (discussed below) then serves as a multilevel analogue to fixed effects LR (Raudenbush & Bryk, 2002). The commonality of this method across various settings, however, makes it candidate for the baseline algorithm against which all other methods may be compared.

Random forests. Despite the plethora of fixed effects classifiers available to researchers, each with their unique benefits and drawbacks, only RF and its mixed effects analogue MERF have been compared directly to LR and its mixed effect analogue GLMM (e.g., Kilham et al.,

2019; Speiser et al., 2019). Therefore, to remain consistent with the sparse existing literature comparing fixed and mixed effects classifiers—as well as its consistent noteworthy performance in classification tasks—RF was selected as a comparison method in the present study. Whereas LR employs the estimation of a linear function to maximize the probability of case membership in within each case's corresponding group, RF operates on a substantially different paradigm based on two principle foundations: A non-parametric recursive partitioning algorithm and an ensemble classifier framework. Friedman (1977) proposed an alternative to parametric classifiers via his recursive partitioning algorithm in which the total feature space (*p*-dimensional vectorial representation of all predictors) was progressively segmented into increasingly more homogeneous regions, *m*. That is, each predictor was split in such a way that the cases in either of the resulting nodes (non-terminal subgroups) most closely resembled one another; this process was repeated for each of the predictors until terminal nodes were reached in which cases were maximally homogeneous such that the Gini index

$$G = \sum_{k=1}^{K} \widehat{p_{mk}}\left(1 - \widehat{p_{mk}}\right)$$

or entropy

$$D = -\sum_{k=1}^{K} \widehat{p_{mk}} \log \widehat{p_{mk}}$$

where

$\widehat{p_{mk}}$ = Proportion of cases from group *k* in region *m*

error functions are minimized (James et al., 2013). The resulting classifier, then, is defined as

$$f(x) \equiv \sum_{m=1}^{M} c_M I\left(x \in R_m\right)$$

where

c_m = Constant for region *m*

I = Indicator function mapping input x to region m

(Ngufor et al., 2019).

The resulting model was identified and improved by Breiman et al. (1984) as classification and regression trees (CART). This framework then became the basis for the later developed RF model. The CART algorithm proved to be conducive under several data conditions in which standard models (e.g., LR and DFA) were unable to operate such as the inclusion of categorical predictors without the use of dummy coding (James et al., 2013). Further, the non-parametric estimation method of CART relaxes the parametric assumptions of methods like LR (such as linearity of the predictor-outcome relationship) and allows for inclusion of interactions and non-linear terms without *a priori* specification (Breiman et al., 1984; James et al., 2013).

Noting the tendency of CART to overfit training data, Breiman then proposed an alternative method in the form of RF (Breiman, 2001) based on the notion of ensemble models. Ensemble classifiers are models consisting of multiple singular classifiers (e.g., CART), each making their own decisions before being aggregated through various means such as boosting or bootstrapped aggregation (bagging) algorithms to improve model accuracy and diversity (Bauer & Kohavi, 1999; Dietterich, 2000; Strobl et al., 2009). Among these ensemble classifiers, RF follows a process as follows:

1. A random bootstrapped sample (sampling with replacement) of size N is drawn from the training set.
2. A singular decision tree T_b is fit to the bootstrapped sample.
3. At each split in trees $T_1 \ldots T_b$, a predictor is selected at random from a random subset of all predictors m, such that $m \approx \sqrt{p}$; where p = total number of predictors.
4. The ensemble of trees $\left\{\bar{T}_b\right\}_1^B$ is output.

5. Group membership predictions $\widehat{C}_b(x)$ are yielded from each of the trees, then aggregated such that $\widehat{C}_{rf}^{B} = majorityvote\left[\widehat{C}_b(x)\right]_1^B$

(Hastie et al., 2009).

Contrary to other commonly used ensemble methods—such as bootstrapped aggregation (bagging) or boosting—RF classifiers create an ensemble of decorrelated trees that mitigate the disproportionately impact of singular predictors and ensure the heterogeneity of the overall ensemble (Breiman, 2001; Dietterich, 2000; Hastie et al., 2009; James et al., 2013; Strobl et al., 2009). The purpose and result of such heterogeneity is twofold: First, the RF ensemble avoids the overfitting problem noted in singular decision trees (Breiman, 1996, 2001; Min et al., 2012); second, this mitigation of overfitting allows for more efficacious predictions and reduced loss in accuracy from training to test data (James et al., 2013; Strobl et al., 2009).

Several multilevel extensions of the CART and RF frameworks have been proposed, such as Sela & Simonoff's (2012) REEMTree, Speiser et al.'s (2020) BiMM tree, and Hajjem et al.'s (2017) GMERT as extensions of CART, and Hajjem et al.'s (2017) MERF and Speiser et al.'s (2019) BiMM forest as extensions of RF. These extensions each sought to account for the serially correlated errors (i.e., non-independence of cases) unaccounted for by fixed effects RF models. However, Speiser et al. (2019) and Kilham et al. (2019) found that under certain conditions, RF models performed comparably to or better than their multilevel analogues, thus providing the impetus for the present study.

Mixed Effects Classifiers

A fundamental multilevel regression model largely resembles a classic ordinary least squares (OLS) regression model, but with a built-in capability to estimate the effects of contextual or higher-level variables on a lower-level outcome. Consider, for example, the effects

of the number of hours each student in a classroom spent studying on a test as well as the self-efficacy of the students' teacher on students' test scores. The test score acts as a student-level outcome, the number of hours spent studying acts as a student-level predictor, and the teacher's self-efficacy acts as an element common to all students in the classroom, a classroom-level predictor. In this motivating example, the students in the classroom are all affected (albeit differently) by common factors such as the classroom atmosphere or the teacher's self-efficacy and, consequently, are likely to yield test scores more closely resembling one another's than those of students in other classrooms with different teachers and test structure. This example is an illustration of both the multilevel data structure as well as a violation of a common assumption in statistical analyses: assumed independence of cases and errors (Tabachnick & Fidell, 2018). While random sampling and assignment (independence in sampling and assignment procedures) is a well-known research design component (Martella et al., 2013; Shadish et al., 2002), statistical analyses also require that cases be uncorrelated with one another else model construction may lead to biased parameter estimates that do not truly represent the sample or extend to the population of interest due to the correlation between cases. Multilevel models, however, do not have this same assumption, but instead account for the commonality between cases within higher-level clusters (such as students within classrooms) through the estimation of *random effects*, the variance in the outcome variable due to cluster membership. This variance decomposition—the portioning of the total variability in the outcome into partitions of case- and cluster-level variability—is precisely what affords multilevel models their prominent position in both cross-sectional and longitudinal research (where the individual acts as the nesting structure and the time point acts as the level-1 'case') in the social sciences (Gully & Phillips, 2019).

The findings of Speiser et al. (2019), Kilham et al. (2019), and Mangino and Finch (2021) shed light on the issue of parsimony versus complexity in the context of multilevel classifiers. That is, under some conditions in Speiser et al.'s (2019) and Kilham et al.'s (2019) studies, the more simplistic RF model outperformed more complex and computationally intensive multilevel models. However, there has yet to be research specifically examining the conditions under which these parsimonious models would be more efficacious than their multilevel counterparts. Multilevel models often require larger sample sizes (discussed in Chapter 2 as the number of clusters * the cluster size) than do fixed effects models and require greater computational power to attain stable parameter estimates. Understanding when simpler models could be employed despite the presence of a multilevel data structure, would enable firm recommendations to practitioners, researchers, and interventionists regarding model selection such as RF instead of a multilevel classifier.

The concept of mixed effects models encompasses the relatively simple notion that the context of data collection may statistically be taken into account (Luke, 2019; Zhou et al., 2019). That is, in order to account for the correlation between individuals within shared contexts (e.g., students within schools) or the correlation between individuals' measurements at multiple sequential time points (in the case of longitudinal analyses), multilevel models estimate two levels of statistical effects: fixed and random effects. Fixed effects (level 1) are estimated with respect to the level-2 structure while random effects are allowed to vary in their estimation via the selective inclusion and estimation of error terms for model coefficients, including both the constant and predictor coefficients (Luke, 2019). A generalized form of the mixed effects model was identified by Bagiella et al. (2000) as

$$Y_i = X_i\boldsymbol{\beta} + Z_i\boldsymbol{\gamma_i} + \boldsymbol{\varepsilon_i}$$

where

Y_i = Predicted response value for case i

X_i = Fixed effects design matrix of dimensions k x p for case i

$\boldsymbol{\beta}$ = Fixed effects coefficient vector

Z_i = Random effects design matrix for case i

$\boldsymbol{\gamma_i}$ = Random effects coefficients vector

$\boldsymbol{\varepsilon_i}$ = Vector of random error for case i.

The models presently discussed then replace Y_i with the predicted probability of group membership $p(X)$, as above, while incorporating a logistic link function to account for the binary response variable. The methods of estimation differ in the two principle methods discussed, however, and as such are discussed in greater detail below.

This generalized model can then be deconstructed to represent a classical regression equation with the level-1 (micro-level; in the students-within-schools example, this is the student level) form

$$y_i' = \beta_0 + \sum \beta_p x_{ip} + \varepsilon$$

where

y_i' = Predicted outcome for case i

β_0 = Intercept coefficient estimate

β_p = Slope coefficient for predictor p

x_{ip} = Input p for case i

ε = Estimation error

(Luke, 2019)

with the additional level-2 (macro-level; in the students-within-schools example, this is the school level) estimation of

$$\beta_0 = \gamma_{00} + \gamma_{01} w_i + r_{ij}$$

where

γ_{00} = Intercept of the intercept

γ_{01} = Level-2 slope coefficient for input w for case i

r_{ij} = Random effect for case i in cluster j

to account for the macro-level structure (Luke, 2019).

The present study focuses on two such multilevel models—the generalized linear mixed model (GLMM) and the mixed effects random forest (MERF)—each of which is discussed in the following sections.

Generalized linear mixed models. The most commonly employed mixed effects model is the GLMM, which acts as the multilevel analogue to the fixed effects LR model discussed above. As an extension of the standard hierarchical linear mixed effects model, GLMM makes use of a similar parameter estimation method to estimate the probability of group membership, as was the case with LR. The basic structure of GLMM makes use of a binomial sampling model with a logit link function at level-1. The overall model estimates the number of cases belonging to a particular group with the equation

$$Y_{ij}|\varphi_{ij} \sim B(m_{ij}, \varphi_{ij})$$

where

Y_{ij} = Number of cases identified as 1 (assuming group labels of {0, 1}) in m_{ij} trials; distributed as binomial with φ_{ij} probability of success per trial across m_{ij} trials

φ_{ij} = Probability of identification in group 1 on each trial with a structural model resembling a traditional regression model

$$\eta_{ij} = \beta_{0j} + \beta_{1j} X_{1ij} + ... + \beta_{pj} X_{pij}$$

where

β_{0j} = Overall intercept; grand mean

β_{pj} = Coefficient estimate for cluster j for predictor p

X_{pij} = Predictor p value for case i in cluster j

and a link function represented as

$$\eta_{ij} = \log\left(\frac{\phi_{ij}}{1-\phi_{ij}}\right)$$

where

η_{ij} = Probability of membership in group 1

(Raudenbush & Bryk, 2002).

The level-2 model then seeks to estimate the cluster-level effects on level-1 parameter estimates as represented by

$$\beta_{qj} = \gamma_{q0} + \sum_{s=1}^{S_q} \gamma_{qs} W_{sj} + u_{qj}$$

where

β_{qj} = Level-1 coefficient estimate q for cluster j

γ_{q0} = Intercept for random effect q

γ_{qs} = Level-2 coefficient estimate s for random effect q

W_{sj} = Level-2 predictor effect s for cluster j

u_{qj} = Random error for random effect q in cluster j

(Raudenbush & Bryk, 2002).

In many studies utilizing multilevel regression models or classifiers, this framework is utilized as the *de facto* default selection (akin to the case with LR) due to the ease of implementation and interpretation as well as the efficacy of method compared to LR (Clarke, 2008). Further, for studies employing both fixed and mixed effects methodology, GLMM serves as a baseline multilevel model against which other types of models may be compared. For

example, Kilham et al. (2019) employed GLMM as a comparison against CART and RF models considering the use of data at both level-1 and level-2 (that is, aggregated); Hajjem et al. (2017) and Ngufor et al. (2019) both used GLMM in the classification context as a baseline multilevel classifier against which various other methods were compared (discussed in greater detail below). In the present study, GLMM acts as the most commonly employed mixed effects classifier against which both the more complex MERF and both fixed effects models will be compared.

Mixed effects random forests. Expanding upon the baseline GLMM, Hajjem et al. (2014) proposed a framework for developing the MERF algorithm as one of several multilevel extensions of RF including, among others, Speiser et al.'s (2019) binary mixed model random forest (BiMM forest) and Capitaine et al.'s (2020) stochastic mixed effects random forest (SMERF). Among these numerous frameworks, the only one presently available to implementation in the R Statistical Software package (R Core Team, 2020) via Ngufor's (2019) `Vira` package is Hajjem et al.'s (2014) MERF algorithm. Consequently, while previous comparisons have employed alternative frameworks, it is Hajjem et al.'s (2014) MERF framework that will be presently utilized due to its availability in the R statistical software package.

The MERF algorithm follows a function estimating response variable y^* with an RF model estimating fixed effects parameters and random effects assumed to be linear. Fixed and random effects within the model are estimated iteratively with each successive iteration updating the residuals of the previous until the algorithm converges and population-averaged predictions are yielded for the training set. This model is represented as

$$y_i = f(X_i) + Z_i b_i + \varepsilon_i$$

where

$f(X_i)$ = Random forest function for fixed effects X_i

Z_i = Random effects matrix of covariates with dimensions $n_i * q$

b_i = Unknown random effects vector for cluster i with distribution $b_i \sim N(0, D)$

ε_i = Vector of errors of dimensions $n_i * 1$ with distribution $\varepsilon_i \sim N(0, R_i)$

(Hajjem et al., 2014).

Each iteration begins by estimating $f(X_i)$ before updating variance components

$$\widehat{\sigma^2_{(r)}} = N^{-1} \sum_{i=1}^{n} \left\{ \hat{\epsilon}^T_{i(r)} \widehat{\epsilon_{i(r)}} + \hat{\sigma}^2_{(r-1)} \left[n_i - \hat{\sigma}^2_{(r-1)} trace\left(\hat{V}_{i(r-1)} \right) \right] \right\}$$

for the model error ε_i and

$$\widehat{D_{(r)}} = n^{-1} \sum_{i=1}^{n} \left\{ \hat{b}_{i(r)} \hat{b}^T_{i(r)} + \left[\hat{D}_{(r-1)} - \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{(r-1)} Z_i \hat{D}_{(r-1)} \right] \right\}$$

for the random effects b_i (Hajjem et al., 2014). A generalized log-likelihood criterion is calculated at each iteration until the change is sufficiently minimal as to characterize convergence. Predictions from this model are estimated using population-averaged parameters $f(x_{ij})$ and $Z_i b_i$ in cases where new data belong to existing clusters; predictions for new cases in new clusters use $f(x_{ij})$ only.

While Hajjem et al.'s (2014) model was instantiated for continuous outcomes rather than binary (Hajjem et al. [2017] created an as-yet inaccessible generalized mixed effects random tree [GMERT] algorithm to begin remedying this), studies by Mangino and Finch (2021) and Ngufor et al. (2019), as well as preliminary informal investigations, demonstrated that the current MERF algorithm has sufficient efficacy in binary classification settings to be considered in the present examination, despite not having been designed for the classification purpose. Further, the present study will serve a tertiary purpose of further illustrating this algorithm's efficacy in classification contexts despite its initial validation in regression contexts.

Outcome Metrics

Many studies comparing classification models examine several possible outcome metrics: a) overall and group-specific accuracy rates (i.e., sensitivity and specificity; Altman & Bland, 1994; Feldesman, 2002; Mann et al., 2008; Powers, 2011), b) various error metrics such as overall misclassification or error rates (Bolin & Finch, 2014; Breiman, 2001; Finch et al., 2014; Lei & Koehly, 2003), c) false positive and false negative classification rates (Abu-Nimeh et al., 2007), and d) singular representations such as receiver operating characteristic (ROC) curve or area-under-the-curve (AUC) values (Mann et al., 2008; Powers, 2011). However, as noted by Chawla (2009), many common metrics such as overall accuracy may be inappropriate when considered in the context of unbalanced data (that is, when outcome group sizes are unequal). In such situations, accuracy metrics that account for group-specific accuracy rates—such as sensitivity and specificity, AUC, or precision and recall—provide more accurate representations of classifier accuracy (Chawla, 2009; Powers, 2011). A notable and widely demonstrated phenomenon in cases of unbalanced datasets is an increase in model overall accuracy and large-group recovery (LGR; true negative identification rate or often called specificity) with a concomitant decrease in small-group recovery (SGR; true positive identification rate or often sensitivity; Bolin & Finch, 2014; Finch et al., 2014; Hoens & Chawla, 2013; Lei & Koehly, 2003; Mangino et al., forthcoming). This is caused by a high LGR rate increasing the overall accuracy rate while the SGR rate remains low; as Hoens and Chawla (2013) illustrate, model trained on a dataset where 99% of cases are in the larger group will likely yield an accuracy rate of 99%, but will misclassify nearly all small-group cases.

Therefore, the present investigation utilizes two such accuracy metrics, sensitivity and specificity, and considers them each with respect to the other. For ease of interpretation and generalizability, these terms will be referred to rather as LGR and SGR, as the context of the data

will dictate whether the positively identified or negatively identified group will be in the minority of cases. Rather, using SGR and LGR provides an indication of model efficacy in identifying cases in whichever group is smaller or larger rather than relying on a judgment of whether each case is positively or negatively identified.

Large-Group Recovery

Large-group recovery (LGR) for the purposes of this investigation will be interpreted as the "true negative classification rate" and resembles the proportion of cases correctly classified into the larger group. This calculation proceeds as follows: TN / (TN + FN); this metric holds a value between 0.0 and 1.0 and can be interpreted as a percentage of cases correctly classified into the larger group. In the case of unbalanced group sizes, LGR tends to largely mirror the overall accuracy rate and, thus, should be interpreted cautiously and with reference to SGR. Given that the present series of studies consider both equal and unequal group sizes, LGR may be interpreted as Group 0 (outcome group identified as 0 rather than 1) accuracy within the context of the simulation study. In the examination of PISA data, LGR will resemble the correct prediction of students who were never held back in school.

Small-Group Recovery

Small-group recovery (SGR) will be interpreted as the "true positive classification rate" and resembles the proportion of cases correctly classified into the smaller group. The calculation proceeds as follows: TP / (TP + FP); this metric also holds a value between 0.0 and 1.0 that can be interpreted as the percentage of cases correctly classified into the smaller group. In the present series of studies, SGR may also be interpreted as Group 1 accuracy when group sizes are equal. When considering the PISA data examination, SGR may be interpreted as the correct prediction of cases that were never held back in school.

Ultimately, the results from both the simulation study and the PISA examination will be compared to one another to collectively triangulate the efficacy of the methods proposed under various conditions within nested data structures for the purpose of classification.

CHAPTER 3: METHODOLOGY

The present investigation encompasses both a simulation study and archival data analysis in order to determine the comparative efficacy of fixed and mixed effects classifiers under varied data conditions. The principle purpose of the present study is to ascertain the predictive capability of various fixed and mixed effects classifiers in binary classification settings. An initial simulation study serves to provide a preliminary consideration of fixed effects classifiers compared to mixed effects models such that the conditions inherent in data featuring a nested structure are thoroughly considered. Results from the simulation study then serve to inform the methods to be employed in the archival analyses, given the conditions within the dataset presently utilized. Throughout both investigations, the R Statistical Software Package (R Core Team, 2020) was employed with several additional packages utilized to implement the various analytical techniques currently considered (to be detailed below).

Research Questions and Hypotheses

Given the literature reviewed and the methodology utilized, several research questions and hypotheses may be leveraged in the present study. The guiding research question could be identified as follows: *Under what conditions could fixed-effects classifiers be utilized for predictive classification with multilevel data structures?* Six hypotheses could then be instantiated across the two investigations. First, within the context of the simulation study, the hypotheses are as follows:

1. Across all conditions, little to no appreciable difference will be found between each fixed effects model and its mixed effects analogue. It is hypothesized that the fixed effects RF algorithm will perform as well as MERF, and LR and GLMM perform comparably to one another on both LGR and SGR metrics.

2. Under conditions of unequal group size ratios, all classifiers will yield higher LGR rates (percentages) with simultaneously lower SGR values compared to conditions of equal group sizes ratios. (H2)
3. Under conditions of increasing cluster size and number of clusters, all classifiers will increase in both LGR and SGR rates. Corollary, as the total sample size increases, so too will both LGR and SGR rates. (H3)
4. Under conditions of increasing ICC, all classifiers will yield lower LGR and SGR rates. However, it is hypothesized that both RF and MERF demonstrate this diminution of accuracy to a lesser extent than LR and GLMM. (H4)
5. Across all conditions, RF and MERF will demonstrate the greatest predictive efficacy with the highest SGR and LGR rates, outperforming both LR and GLMM. (H5)

In the PISA data examination, one eminent outcome is hypothesized:

6. In the prediction of student retention, the LGR and SGR outcomes of all algorithms will largely mirror their simulation study results for the conditions most closely resembling those of the PISA dataset. (H6)

Results from the present studies will inform the model selection process for use in examinations of nested data for the purposes of classification to illustrate the necessity of more robust models, if not necessarily those accounting for an existing nested data structure.

Simulation Study

To determine the comparative efficacy of fixed- and mixed-effects models in the presence of multilevel data, a Monte Carlo simulation provides preliminary evidence to inform the conditions under which fixed or mixed effects classifiers should be employed. Within the Monte Carlo simulation framework, six data characteristics are sequentially permuted—thus yielding 243 total simulated conditions—and each set of conditions iterated five hundred times (due to the

computational capacity required to estimate the current models) to approximate the likely outcomes to arise while accounting for the stochasticity of any single data generation process (Harrison, 2010). Permissions were obtained to utilize the Ball State University computing cluster to run all simulation conditions. The conditions are as follows: Number of cases per level-2 cluster, number of level-2 clusters, intraclass correlation, group size ratio (ratio of outcome Group 0 to outcome Group 1), and the type of predictors used (all three at level-1; one at level-2 with two at level-1, and two at level-2 with one at level-1). The conditions manipulated and classifiers employed are illustrated in Table 1 (below).

Table 1: Data Simulation Conditions

Simulation variable	Conditions
Number of level-1 cases per level-2 cluster	10; 20; 50
Number of level-2 clusters	10; 30; 50
Correlation within level-2 clusters (intraclass correlation)	0.1; 0.3; 0.8
Number of Predictors	3 at level-1; 2 at level-1 and 1 at level-2; 1 at level-1 and 2 at level-2
Outcome Group Size Ratio	50:50; 75:25; 90:10
Method	LR; GLMML RF; MERF
Outcome Metrics	Large-Group Recovery, Small-Group Recovery

The literature reviewed in Chapter 2 and preliminary descriptive analyses of the PISA dataset provided a basis for the various levels to be used in each of the conditions, as illustrated in Table 1 (above). For the number of level-1 cases, Hajjem et al.'s (2014) use of cluster sizes of 10 and 50, Maas and Hox's (2005) finding that a cluster size of 30 is common in educational settings, and Hox's (1998) statement that cluster sizes of 10 and 20 could yield accurate cross-level interaction and variance estimates led to the decision to use cluster sizes of 10, 20, and 50.

The number of clusters has been previously examined with larger size conditions (e.g., Mangino & Finch [2021] used a number of clusters up to 100; Crane-Droesch [2017] used a number of clusters from 900 to 2700). However, Kreft's (1996) recommendation of 30 clusters and Paccagnella's (2011) finding that 50 or more clusters provides robust standard errors led to the use of these conditions. Further, McNeish and Stapleton's (2016) results indicated that with as few as 10 clusters, multilevel models can provide reliable parameter estimates even if the models are underpowered. However, note that the results using the above-cited studies on the number of clusters are also conditioned on the number of units per cluster. Despite the robust literature base on large samples in both multilevel modeling and classification, little work has yet been done incorporating smaller sample sizes in the area of classification with nested data. Consequently, conditions of 10, 30, and 50 clusters are presently considered. The total number of cases, thus, ranges from N = 100 (10 cases to each of 10 clusters) to N = 2500 (50 cases to each of 50 clusters).

The ICC conditions were determined based on the only study in the presently reviewed literature base to specify exact ICC values rather than the oblique "large" and "small" random effects language. Mangino and Finch (2021) specified their ICC conditions to be 0.1, 0.3, and 0.8. While LeBreton and Senter (2008) specified that any dataset for which the ICC ≥ 0.05 should be analyzed utilizing a multilevel framework. Further, while Speiser et al. (2019, 2020) specified standard deviations of 0.1 and 0.5 to mean small and large random effects, respectively, the actual calculated ICC is unknown for these studies. Therefore, Mangino and Finch's (2021) conditions were selected as they resemble a wide variety of cluster heterogeneity. To determine the parameters to be entered into the `sim.multi` function to attain these ICCs, a preliminary simulation study was conducted and described in the "Preliminary Simulation" section, below. The outcome group size ratios (ratio of case membership in Group 0 to Group 1) were

determined by previous studies in the classification literature base that utilized equal (50:50) and various unequal (75:25 and 90:10, among others) group size ratios (Bolin & Finch, 2014; Lei & Koehly, 2003). Further, given the conditions of the PISA dataset—with a 5108/604 split of individuals who were not retained to those who were (an approximately 8.5 : 1 ratio)—it is evident that a comparison between equal and various levels of unequal group size ratios should be considered. Therefore, between the PISA data and the literature base reviewed in Chapter 2, the 50:50, 75:25, and 90:10 group size ratios were selected.

Many studies reviewed in Chapter 2 featured a number of predictors ranging from two (Lavery et al. [2019]; Maas & Hox [2005]) to 8000 (Capitaine et al. [2019]). However, the number and type of predictors is a set of conditions as yet minimally examined, particularly in the domain of multilevel classifiers. Few studies have explicitly examined the effects of level-1 v. level-2 predictors on model accuracy in either the classification or regression contexts (Kilham et al. [2019] and Downes and Carlin [2020] have done so in applied contexts with real data), but recommendations for practice have yet to be established based on the the intersection of tightly controlled simulation and real data settings. Consequently, the consideration of the number and type of predictors is a highly exploratory area. Therefore, the present study seeks to examine a relatively simple constellation of predictors with some resemblance to the behavior of the predictor and predictor-outcome relationships in the PISA data.

To best approximate the conditions present in the PISA data, a preliminary examination of the data was conducted and involved correlations between each of the predictors (as identified in Chapter 2) and the outcome, as well as between each pair of predictors. These preliminary correlations would determine the magnitude of correlation between each predictor and the outcome as well as among each pair of predictors in the simulation. This analysis of the PISA data revealed a median absolute value predictor-outcome correlation coefficient of 0.1 (i.e.,

among all pairwise correlations, the median magnitude), a 90th-percentile coefficient of 0.146, and a 10th-percentile coefficient of 0.014; additionally, predictor intercorrelations yielded absolute correlation coefficients of 0.006 at the 10th-percentile, 0.087 at the median, and 0612 at the 90th-percentile. These preliminary analyses and existing literature were considered with respect to the intent of preserving parsimony while making the simulation as realistic as possible without sacrificing experimental control. Therefore, while the median correlations among PISA predictors and with the retention outcome were low (according to Cohen's [1988] commonly-cited guidelines), the 75th-percentile correlation coefficients were selected with the predictor-outcome relationship being held constant at 0.13 and predictor intercorrelations being held constant at 0.31. These values represent a commonly-identified low effect size, but act as reasonable values given the archival investigation component of the present study.

Additionally, given the paucity of literature on the type of predictors utilized in situations with nested data, it was determined that the level of predictor be considered while holding constant the number of predictors. Therefore, a total of three predictors were simulated across all conditions with one condition featuring all predictors at level-1, one condition featuring two at level-1 and one at level-2, and a third condition featuring one at level-1 and two at level-2.

Each of the four classifiers are fit to the data—parameters to be specified below in the *Full Simulation Design* section—generated in each iteration with average outcome measures obtained for each set of simulation conditions. The process of the simulation method is detailed in Algorithm 1 (below).

Algorithm 1: Monte Carlo Simulation Method

initialization: Let group size ratio = {0.5, 0.9}; Clusters = {10, 30, 50};

Cases = {10, 20, 50}; ICC = {0.1, 0.3, 0.8};

Predictors = {3 at level-1, 2 at level-1, 1 at level-1};

Methods = {LR, GLMML, RF, MERF}; Metrics = {LGR, SGR};

repeat

1. Simulate data to the set of parameters for all:

Group size ratio, Clusters, Cases, ICC;

2. Fit classifiers to the simulated data for each:

Methods;

3. Output accuracy metrics for each in Methods;

until iterations per condition = 500;

output: Average Metrics for each Method in parameters;

continue until average Metrics for all permutations of parameters obtained;

end

Utilizing a simulation design allows for controlled manipulation of specific characteristics of the data such that the effects of each condition on the classifiers, as well as the various interactions between conditions, can be ascertained (Sanchez, 2005). Within the Monte Carlo simulation framework, the procedure follows that detailed in Algorithm 1 (above). All sets of conditions were simulated using the `sim.multi` function in the R library `psych` (Revelle, 2017). Parameters manipulated within the `sim.multi` function itself were inputs identified as `n.obs, nvar, ntrials, days, sigma` and `sigma.i` to alter the cluster size, the number of variables, the number of clusters (`ntrials` and `days` used the same parameter specification;

these parameters were required to match in the `sim.multi` function call), between-cluster correlation, and the intraclass correlation.

Preliminary Simulation and Parameter Specification for `sim.multi` Function

To ensure the ICCs to be simulated in the full study were done so accurately, a preliminary simulation study was required. Given the high degree of parameter specification in the `sim.multi` function, the parameters in the function call that directly affected the ICC of the data simulated—namely the standard deviations within and between clusters—needed to be determined to ensure proper simulation of the ICCs determined in Chapter 2.

Within this preliminary simulation, the most extreme values of all other irrelevant parameters from the full simulation (i.e., 50 clusters with 50 cases each) were held constant and only the group size ratio, `sigma,` and `sigma.i` function inputs were manipulated (the between-person and within-person standard deviations, respectively). This was performed in order to determine the various manners of relationships between these function parameters such that the desired ICCs could be obtained from the function call in the full simulation. Additionally, to account for potential variability in results due to group size ratio, all three group size ratio conditions used in the full simulation were utilized in accordance with previous simulation studies (e.g., Bolin & Finch, 2014; Lei & Koehly, 2003). Parameters for this study are shown in Table 2 (below).

Table 2: Preliminary Simulation Conditions

Simulation variable	Conditions
`Sigma` Parameter (Interclass Standard Deviation)	1; 2; 4; 6; 10
`Sigma.i` Parameter (Intraclass Standard Deviation)	1; 2; 4; 6; 10
Outcome Group Size Ratio	50:50; 75:25; 90:10
Outcome Metrics	Intraclass Correlation; Interclass Correlation

This preliminary simulation was run for 100 iterations across the 75 total sets of conditions with optimal parameter ratios selected for use in the full simulation study; mean ICCs were obtained across the 100 iterations per set of conditions. The optimal function parameters (i.e., those closest to the ICCs identified for the full simulation) were selected based on the values closest to the full simulation data conditions. Mean intra- and interclass correlations for each group size ratio condition are shown below in Table 3 (below).

Table 3: Preliminary Simulation Results for ICC and Interclass Correlation Parameters

Intraclass Correlation	**Sigma.i Parameter**	**Sigma Parameter**	**Interclass Correlation**
	50:50 Group Size Ratio		
0.1 (0.1)	10	4	0.9 (0.9)
0.3 (0.32)	4	4	0.7 (0.68)
0.8 (0.78)	4	10	0.2 (0.22)
	75:25 Group Size Ratio		
0.1 (0.1)	1	1	0.9 (0.9)
0.3 (0.3)	10	6	0.7 (0.7)
0.8 (0.79)	6	10	0.2 (0.21)
	90:10 Group Size Ratio		
0.1 (0.12)	10	1	0.9 (0.88)
0.3 (0.3)	4	2	0.7 (0.7)
0.8 (0.81)	1	2	0.2 (0.19)

Note: Actual calculated values shown parenthetically.
Note 2: Cluster size set to 50 and number of clusters set to 50.

The intra- and interclass correlations were computed manually (rather than using established functions) using the binary outcome ICC from Chapter 2

$$\rho = \frac{\sigma^2_{u_0}}{\left(\sigma^2_{u_0} + \sigma^2_r\right)}$$

where

σ_r^2= Estimate of level-2 (cluster-level) variance

σ_r^2= Estimate of constant level-1 (case-level) variance with the form $\sigma_r^2 = \frac{\pi^2}{3}$

(Moineddin et al., 2007)

for the ICC and the derivation $1 - \rho$ for the interclass correlation. The latter equation thus represents the proportion of variance in the outcome not accounted for by cluster membership; thus, as the ICC increases, the interclass correlation decreases. Results obtained from the preliminary study indicate the values at which the `Sigma` and `Sigma.i` parameters should be set in the full simulation.

In the full simulation, the `ntrials` and `days` parameters will be set equal to one another; the function creates data presumed to be longitudinal and, in order to generate a dataset such that the "time" variable could be interpreted as "measurement occasion" (i.e., a discrete measurement; more appropriately, a sub-level-1 predictor, given the cross-sectional nature of the PISA examination and its analogous simulated conditions), the vector created for "time" was divided by 24 to remove the "hours" metric used natively in the `sim.multi` function; the resulting variable could, more appropriately, be considered as the number of individuals within each cluster.

Full Simulation Design

The present investigation focuses on a comparison of four classifiers: LR and RF and their respective mixed effects extensions, GLMM and MERF (as discussed in Chapter 2), in a classification context. Among the methods presently employed, LR was implemented using the base R function `glm` with a specified binomial link function; GLMM was implemented using the `glmer` function from the `lme4` package (Bates et al., 2014) and a binomial link function; RF was implemented using the `ranger` function and package in R (Wright & Ziegler, 2015), which itself

is reliant on the `randomForest` package (Liaw & Wiener, 2002) and uses the same algorithm, though does not have the same internally coded inefficiencies as the `randomForest` function, and is programmatically optimized for model estimation with larger sample sizes and fixed effect predictors with a high number of categories (thus allowing it to utilize school identifier as a predictor; the `ranger` function also includes a specified 200 trees and an `mtry` parameter of randomly selecting from two predictors at each split in each tree ($m = \sqrt{p}$); and MERF utilized the `MEml` function in the `Vira` package (Ngufor, 2019) with a specified 'RF' classifier to employ a call to an internal `MERF` function (in contrast to calls to the generalized linear mixed model 'GLM' or gradient boosting algorithms 'GBM' contained within the function), 200 trees, and an `mtry` parameter of two predictors.

Across the conditions noted above and discussed in detail in Chapter 2, the data are generated, then subsequently split randomly in equal proportions such that half constitutes the training set and half constitutes the test set. The four classifiers are first fit to the training set with SGR and LGR rates obtained; the fit models are then applied to the test set with SGR and LGR rates obtained for predictions. The SGR and LGR rates obtained from predictions on the test set act as the present principle outcomes of interest. To compare the effects of the data conditions and methods on the outcome accuracy metrics, 3 * 3 * 3 * 3 * 3 * 4 factorial ANOVA designs are employed with LGR and SGR acting as outcome variables. Further, in order to ensure the interpretability of the interactions identified, five- and six-way interactions are not included in the analysis, thus limiting the interaction depth to four-way interactions. The implementation of factorial ANOVAs provides statistical significance tests as well as effect size measures to determine the practical significance of each main effect and interaction within the models. Considering practical significance alongside statistical significance is necessary due to the high likelihood that interactions will be statistically significant due simply to the large sample size. In

accordance with previous simulation studies, η_p^2 serves as the effect size with values of $\eta_p^2 > 0.2$ indicating practical significance; only the strongest interactions not subsumed under a higher-level interaction are interpreted (Bolin & Finch, 2014; Lei & Koehly, 2003). Therefore, only interactions that are both statistically significant and practically significant where $\eta_p^2 > 0.2$ are interpreted.

Application: Using the Program for International Student Assessment (PISA) Data to Predict Student Retention

In addition to the comparison of methods on simulated data acting as a preliminary examination and verification of previous studies' results, the present investigation then applies the methods chosen to the Program for International Student Assessment (PISA) United States dataset in order to predict whether students have been held back a grade in school. The coupling of these two examinations—both the simulation and the PISA investigation—allow for internal verification of results and an extension of simulation study findings to real data. This section describes the dataset, predictors, and outcome chosen for the PISA investigation so as to contextualize this component of the full investigation.

The publicly available PISA dataset includes 5712 unique students in 177 schools with 947 variables, many of which are multicategorical coded representations of variables from the restricted dataset (e.g., GPA is represented in categorical ranges rather than specific continuous values for each individual). The outcome of retention was chosen due to both its status as a binary outcome (either students have or have not been retained) and the fact that it is a powerful predictor of later school dropout and poor academic performance (Glick & Sahn, 2010). The present examination—following the simulation study thus described—seeks to predict student retention based on several variables measured at the student-level. In the total dataset, 4941 students (86.5%) were not ever held back and 592 (10.4%) were; a remaining 178 individuals

(3.1%) had missing values for this variable. Consequently, the present outcome variable results in an unbalanced dataset (as described in the group size ratio section, above) and an approximately 8.5:1 ratio of students who were never held back to those who were.

To predict this outcome, several variables are considered in accordance with literature within the school retention literature base. As previously noted, many variables that would traditionally be continuous in nature (e.g., grade point average) are categorical in the public use PISA dataset. However, several variables remained in their continuous representations in the public dataset and, thus, only these variables were considered in order to maintain continuity with the simulation study's data generation processes. More positive attitudes toward school (Ikeda & Garcia, 2014), greater sense of belonging, (McMahon et al., 2008), and a higher expected educational attainment (Lee & Stankov, 2018) were associated with a reduced likelihood of being retained. Additionally, parent factors including improved parent education (Corman, 2003), and greater home resources (e.g., books), wealth, and income were identified as resulting in a reduced likelihood of students being retained (Choi et al., 2018; Corman, 2003; Eisemon & United Nations Educational, Scientific, and Cultural Organization, Paris, France, 1997; Wößmann, 2003).

The models being constructed in the present investigation consist only of two levels with the student as the level-1 unit and school as the level-2 unit; while these variables are not comprehensive predictors of student retention, they serve as salient continuous predictors within both the literature and in the PISA restricted dataset. Twelve total predictors are used aligning with the variables and conditions described above. The twelve total predictors will be accompanied by the school identifier (CNTSCHID) as the variable specifying the nesting structure in GLMM and MERF, acting as a categorical predictor in RF, and being dummy coded into $k - 1$ dummy coded predictors for LR.

The publicly available PISA dataset consists of 5712 unique individuals in 177 schools and 947 variables relevant to students' school performance and other related variables. The unrestricted dataset largely features numerous continuous variables and, thus, are considered within the present investigation. The use of the PISA dataset serves to illustrate an applied exemplar of the differences in model capabilities and allow for triangulated results of the various methods' efficacy under various data conditions. Descriptive statistics for the predictors and outcome within the PISA dataset are shown in Table 4 (below).

Table 4: Descriptive Statistics for Raw and Imputed PISA Datasets

	Mean (standard deviation)	
Variable	PISA Original	PISA Imputed
MOTIVAT Student attitudes, preferences, and self-related beliefs: Achieving motivation	0.661 (0.943)	0.675 (0.946)
HEDRES Home educational resources	-0.115 (1.142)	-0.1053 (1.145)
WEALTH Family wealth	0.473 (1.086)	0.476 (1.085)
ESCS Index of economic, social and cultural status	0.079 (1.003)	0.083 (1.002)
AGE Age	15.806 (0.287)	15.806 (0.287)
BMMJ1 International socio-economic index of occupational status of mother	49.873 (21.777)	47.976 (22.046)
BFMJ2 International socio-economic index of occupational status of father	43.317 (22.309)	41.334 (22.063)
PARED Index highest parental education in years of schooling	13.621 (2.802)	13.632 (2.791)
BELONG Subjective well-being: Sense of belonging to school	-0.085 (1.013)	-0.074 (1.024)
Hisei Index highest parental occupational status	53.643 (21.713)	53.667 (21.593)
HOMEPOS Home possessions	0.022 (1.112)	0.227 (1.110)
BSMJ Students' expected occupational status	62.419 (17.045)	63.247 (17.309)
	Repeated / Not (%)	
REPEAT Grade repetition	4941 (89.30) / 592 (10.70)	5108 (89.43) / 604 (10.57)

Several preliminary descriptive analyses were conducted to assess the ICC of the retention variable as well as its group size ratio; this is done to determine the simulation conditions most similar in nature to those in the PISA dataset. As discussed above, the REPEAT variable in the imputed dataset used for the present investigation indicated 5108 students (89.43%) did not repeat a grade and 604 students (10.57%) did, thus making this dataset unbalanced. Further, a multilevel model specifying only the nesting structure of the dataset allowed for the variance decomposition to be assessed yielding an ICC of approximately 0.1338 for the imputed dataset (from which a random subset of 1428 cases in each the training and test sets will be selected for the model comparison analysis). Therefore, when considering the present analysis, the imputed version of the public-use dataset was employed with the school identifier variable specified as the nesting structure and all variables measured at the student level. To clean the dataset for the full investigation, the following procedure was used:

1. Select a subset the full PISA dataset of only the relevant variables: REPEAT, MOTIVAT, HEDRES, WEALTH, ESCS, AGE, BMMJ1, BFMJ2, PARED, BELONG, hisei, HOMEPOS, BSMJ, CNTSTUID, and CNTSCHID.
2. All cells with negative values indicate an uninterpretable or absent response and, thus, were removed from the dataset, replaced with “NA” values.
3. All missing values were imputed using multivariate equations by chained equations (MICE) with a random forest estimator as per recommendations by Shah et al. (2014) and Waljee et al. (2013).
4. Create a person-period format of this dataset such that all level-1 (student-level) variables are stacked and reduced into a single column.

The full comparative analysis proceeded in a manner akin to that found in many studies in the areas of multilevel modeling and classification whereby the full modified dataset was

randomly split in two mutually exclusive and equally sized partitions to serve as the training (89 schools) and test (88 schools) sets (mean cluster size = 8.4 cases; Capitaine et al., 2019; Hajjem et al., 2014; Speiser et al., 2019). All four models were then fit to the training set with accuracy measures obtained; these models were then be applied to the test set with accuracy measures obtained. The test set SGR and LGR values as assessed as a measure by which models were compared to one another in predictive capability. The results of this examination were compared to the results of the simulation study in order to attain a consistent understanding of the present models' predictive capability. The use of both simulated and archival data will allow for the controlled examination of the various models presently under investigation to ascertain their efficacy under a variety of data conditions. The controlled setting of the simulation allow for a broad range of settings under which these models may be used; the use of the PISA dataset, then, serves as a more readily applicable examination and verification of the findings of the simulation.

The average SGR and LGR values for each model across the simulation conditions and the results of the PISA analysis were compared to one another to triangulate the most efficacious models in the set of those presently considered under the presently relevant conditions.

CHAPTER 4: RESULTS

Results from the two studies conducted are detailed in this chapter beginning with a discussion of the most important interactions ($\eta_p^2 > 0.2$) from the simulation study and followed by the results of the PISA examination. As detailed in Chapter 3, only those four-way interactions (to maintain interpretability) that were both statistically and practically significant ($\eta_p^2 > 0.2$) are interpreted to only identify those factors most salient in the present investigation.

Simulation Study Results

Within the simulation study, the two outcomes of LGR and SGR yielded three and four practically significant interactions, respectively, for a total of seven interactions to be interpreted.

Outcome: Large Group Recovery

Within the LGR outcome, three interactions were identified as statistically and practically significant: An interaction of method by predictor type by ICC by group size ratio ($F_{24, 368} = 59.287$, $p < 0.001$, $\eta_p^2 = 0.795$); an interaction of method by predictor type by group size ratio by the number of level-2 clusters ($F_{24, 368} = 11.271$, $p < 0.001$, $\eta_p^2 = 0.424$); and an interaction of predictor type by ICC by group size ratio by number of level-1 cases within clusters ($F_{16, 368} = 16.917$, $p < 0.001$, $\eta_p^2 = 0.424$). The results of these interactions are shown in the ANOVA table below (Table 5).

Table 5: ANOVA Table for LGR Interactions

	Sum of Squares	df	Mean Square	F	p	η_p^2
method	0.11771	3	0.03924	4485.019	< .001	0.973
predictors	1.34221	2	0.67110	76714.744	< .001	0.998
icc	0.39604	2	0.19802	22635.632	< .001	0.992
gsr	17.97781	2	8.98891	1.03e+6	< .001	1.000
ngroups	0.00944	2	0.00472	539.639	< .001	0.746
ncases	3.09e-4	2	1.54e-4	17.650	< .001	0.088
method ✻ predictors	0.05611	6	0.00935	1069.091	< .001	0.946
method ✻ icc	0.01030	6	0.00172	196.146	< .001	0.762
predictors ✻ icc	0.10374	4	0.02593	2964.658	< .001	0.970
method ✻ gsr	0.03526	6	0.00588	671.826	< .001	0.916
predictors ✻ gsr	2.83055	4	0.70764	80891.036	< .001	0.999
icc ✻ gsr	0.89870	4	0.22467	25682.807	< .001	0.996

Table 5: ANOVA Table for LGR Interactions

	Sum of Squares	df	Mean Square	F	p	η_p^2
method ✻ ngroups	0.00101	6	1.69e-4	19.295	< .001	0.239
predictors ✻ ngroups	1.09e-4	4	2.72e-5	3.107	0.016	0.033
icc ✻ ngroups	6.31e-4	4	1.58e-4	18.021	< .001	0.164
gsr ✻ ngroups	0.00158	4	3.94e-4	45.050	< .001	0.329
method ✻ ncases	4.91e-4	6	8.19e-5	9.357	< .001	0.132
predictors ✻ ncases	0.00235	4	5.88e-4	67.160	< .001	0.422
icc ✻ ncases	6.19e-4	4	1.55e-4	17.691	< .001	0.161
gsr ✻ ncases	0.00322	4	8.06e-4	92.081	< .001	0.500
ngroups ✻ ncases	4.14e-4	4	1.04e-4	11.845	< .001	0.114
method ✻ predictors ✻ icc	0.01071	12	8.92e-4	101.987	< .001	0.769
method ✻ predictors ✻ gsr	0.09483	12	0.00790	903.355	< .001	0.967
method ✻ icc ✻ gsr	0.02236	12	0.00186	213.035	< .001	0.874
predictors ✻ icc ✻ gsr	0.61635	8	0.07704	8807.002	< .001	0.995
method ✻ predictors ✻ ngroups	9.37e-4	12	7.81e-5	8.929	< .001	0.225
method ✻ icc ✻ ngroups	2.26e-4	12	1.89e-5	2.156	0.013	0.066
predictors ✻ icc ✻ ngroups	1.97e-4	8	2.47e-5	2.819	0.005	0.058
method ✻ gsr ✻ ngroups	2.64e-4	12	2.20e-5	2.513	0.003	0.076
predictors ✻ gsr ✻ ngroups	0.00322	8	4.02e-4	45.972	< .001	0.500
icc ✻ gsr ✻ ngroups	2.46e-4	8	3.07e-5	3.514	< .001	0.071
method ✻ predictors ✻ ncases	0.00215	12	1.79e-4	20.505	< .001	0.401
method ✻ icc ✻ ncases	1.15e-4	12	9.62e-6	1.100	0.359	0.035
predictors ✻ icc ✻ ncases	3.85e-4	8	4.82e-5	5.506	< .001	0.107
method ✻ gsr ✻ ncases	3.83e-4	12	3.19e-5	3.645	< .001	0.106
predictors ✻ gsr ✻ ncases	0.00272	8	3.41e-4	38.925	< .001	0.458
icc ✻ gsr ✻ ncases	0.00156	8	1.95e-4	22.243	< .001	0.326
method ✻ ngroups ✻ ncases	2.36e-4	12	1.97e-5	2.247	0.010	0.068
predictors ✻ ngroups ✻ ncases	2.42e-4	8	3.03e-5	3.459	< .001	0.070
icc ✻ ngroups ✻ ncases	1.79e-4	8	2.24e-5	2.561	0.010	0.053
gsr ✻ ngroups ✻ ncases	1.02e-4	8	1.28e-5	1.461	0.170	0.031
method ✻ predictors ✻ icc ✻ gsr	0.01245	24	5.19e-4	59.287	< .001	0.795
method ✻ predictors ✻ icc ✻ ngroups	2.47e-4	24	1.03e-5	1.176	0.260	0.071
method ✻ predictors ✻ gsr ✻ ngroups	0.00237	24	9.86e-5	11.271	< .001	0.424
method ✻ icc ✻ gsr ✻ ngroups	1.86e-4	24	7.74e-6	0.885	0.624	0.055
predictors ✻ icc ✻ gsr ✻ ngroups	4.06e-4	16	2.54e-5	2.899	< .001	0.112
method ✻ predictors ✻ icc ✻ ncases	3.85e-4	24	1.60e-5	1.834	0.010	0.107
method ✻ predictors ✻ gsr ✻ ncases	5.72e-4	24	2.38e-5	2.724	< .001	0.151
method ✻ icc ✻ gsr ✻ ncases	2.61e-4	24	1.09e-5	1.243	0.201	0.075
predictors ✻ icc ✻ gsr ✻ ncases	0.00237	16	1.48e-4	16.917	< .001	0.424

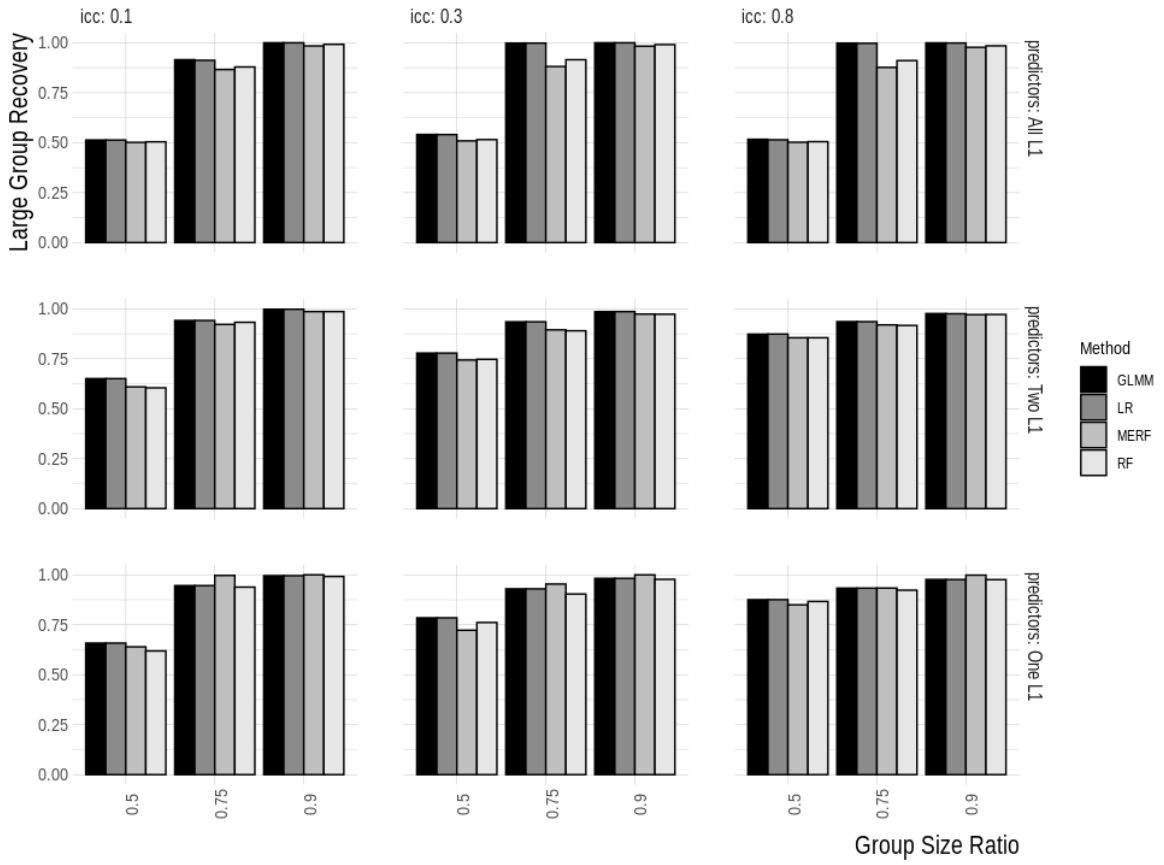
Table 5: ANOVA Table for LGR Interactions

	Sum of Squares	df	Mean Square	F	p	η_p^2
method ✻ predictors ✻ ngroups ✻ ncases	1.71e-4	24	7.13e-6	0.815	0.718	0.050
method ✻ icc ✻ ngroups ✻ ncases	1.24e-4	24	5.16e-6	0.590	0.940	0.037
predictors ✻ icc ✻ ngroups ✻ ncases	2.60e-4	16	1.62e-5	1.856	0.023	0.075
method ✻ gsr ✻ ngroups ✻ ncases	8.88e-5	24	3.70e-6	0.423	0.993	0.027
predictors ✻ gsr ✻ ngroups ✻ ncases	4.35e-4	16	2.72e-5	3.111	< .001	0.119
icc ✻ gsr ✻ ngroups ✻ ncases	4.62e-4	16	2.89e-5	3.303	< .001	0.126
Residuals	0.00322	368	8.75e-6			

Method by Predictor Type by ICC by Group Size Ratio. The interaction of method by predictor type by ICC by group size ratio reveals a general increase in LGR as group size ratios become increasingly more discrepant (as shown in Figure 1, below). When all predictors were at level-1, a relatively common pattern was observed in which LGR began at approximately 50% for all methods at a 50:50 ratio before increasing dramatically to rates between approximately 90 – 100% when the group size ratio increased to 75:25 and 90:10; this was consistent across all ICC conditions. Under all ICC conditions, RF and MERF yielded the lowest LGR when the group size ratio was 75:25; all methods performed comparably when the group size ratio was 90:10. When two predictors were simulated at level-1, the effect of ICC was more apparent: The LGR at a 50:50 group size ratio was higher, ranging from approximately 55 – 70% before increasing to approximately 90%+ when group size ratios were unequal; this was common to both the 75:25 and 90:10 group size ratios, though the 90:10 condition resulted in marginally higher LGR across all methods. Once again, all methods performed similarly with RF and MERF yielding slightly lower LGR than GLMM and LR. When only one predictor was simulated at level-1, a nearly identical pattern to that observed with two predictors at level-1 was identified.

However, when one predictor was simulated at level-1, MERF yielded the highest LGR under both unequal group size conditions; LR and GLMM yielded the highest (and nearly identical) LGR across all conditions. Broadly speaking, as the ICC increased, so too did LGR under all conditions except when all predictors were at level-1. Similarly, as more predictors were added at level-2, LGR tended to increase. In sum, this interaction illustrates that as the ICC and number of predictors simulated at level-2 increase, so too does LGR across all methods. Additionally, LGR increases as group sizes become increasingly more discrepant, and the ICC and number of level-2 predictors increases.

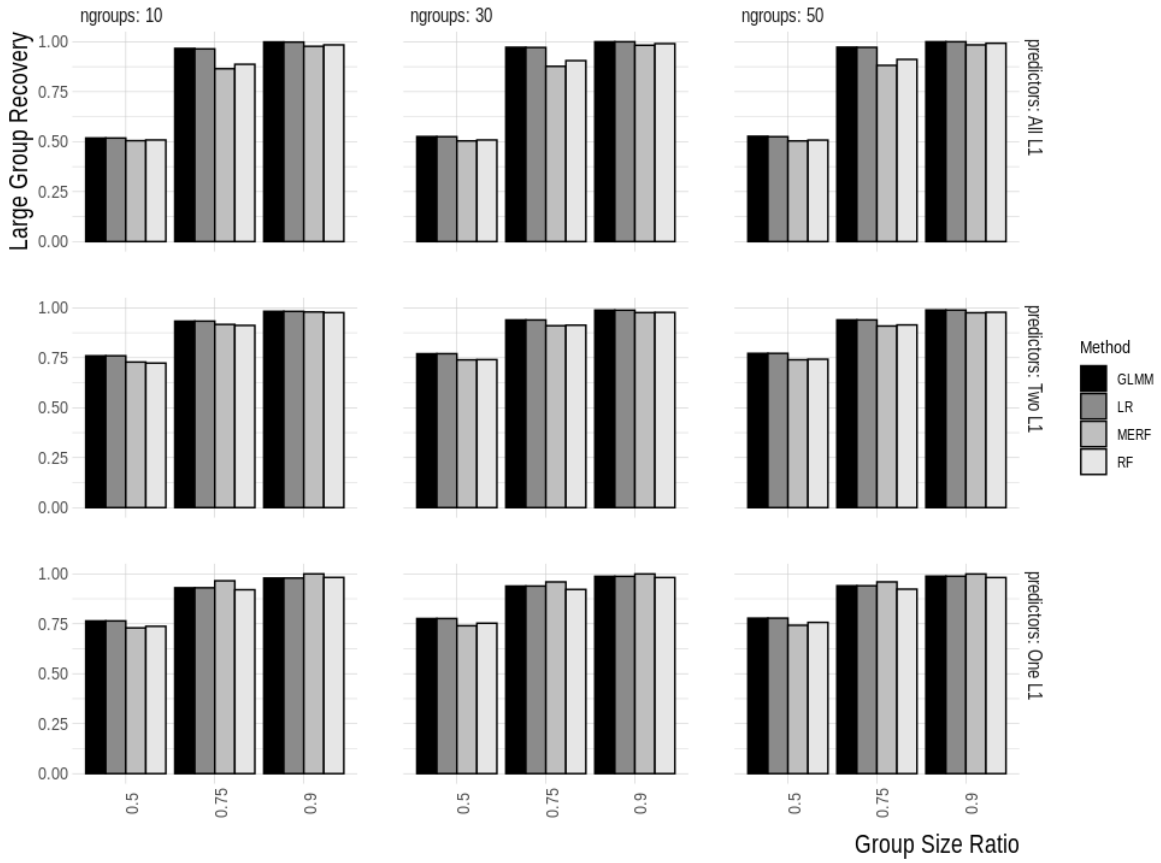
Figure 1: LGR for Method by Predictor Type by ICC by Group Size Ratio Interaction



Method by Predictor Type by Group Size Ratio by Number of Level-2 Clusters. The interaction of method by predictor type by group size ratio by number of level-2 clusters

demonstrated a pattern much akin to that observed in the method by predictor type by group size ratio by ICC interaction (see Figure 2, below). Across all sets of conditions, LGR increased as group size ratios became increasingly more discrepant. When all predictors were at level-1 and the group size ratio was equal, LGR was consistent at approximately 50% across all methods, increasing to 90 – 100% when the group size ratio was 75:25; all methods resulted in LGR near 100% when the group size ratio was 90:10. When two predictors were at level-1, LGR increased from approximately 75% to approximately 92% and to nearly 100% when the group size ratio increased from 50:50 to 75:25 and 90:10, respectively. When all predictors were at level-1 and when one was simulated at level-2, GLMM and LR performed nearly identically and yielded the highest LGR. When one predictor was simulated at level-1, LGR began just above 75% with a 50:50 group size ratio before increasing to approximately 92% and up to nearly 100% as the group size ratio became increasingly more discrepant; with one level-1 predictor, MERF yielded the highest LGR while RF, LR, and GLMM performed comparably to one another and with slightly lower LGR. As the number of level-2 clusters increased, LGR slightly increased, though only marginally. Similarly, as more level-2 predictors were added, all methods increased marginally in LGR. Summarily, the same pattern that emerged across the group size ratio, method, and predictor type conditions in the first interaction was replicated here, though with a slight variation through the manipulation of the number of clusters. However, differences that emerged across different numbers of clusters were marginal.

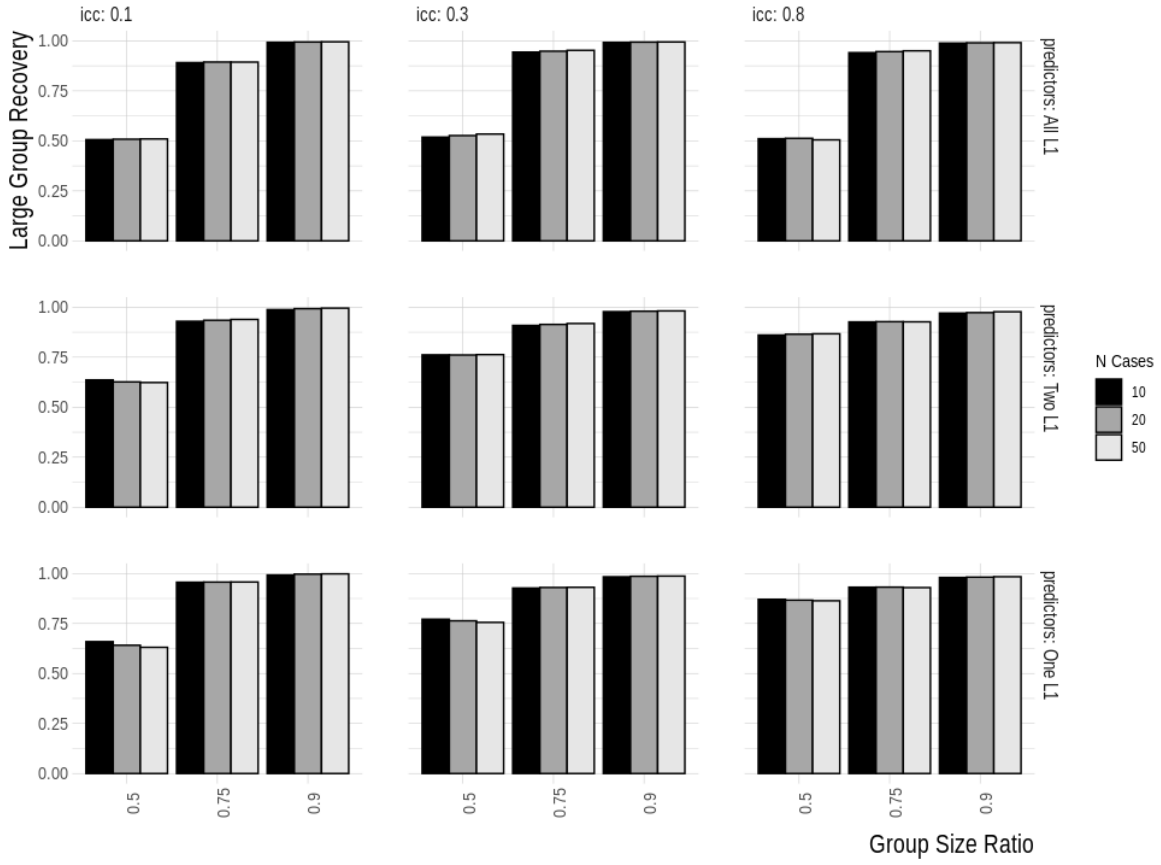
Figure 2: LGR for Method by Predictor Type by Group Size Ratio by Number of Level-2 Clusters



Predictor Type by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster. The pattern operating across all conditions regardless of method was the interaction of predictor type, ICC, group size ratio, and number of level-1 cases per cluster (see Figure 3, below). A general pattern of an increase in LGR was observed as group size ratios became increasingly more discrepant across all conditions. When all predictors were simulated at level-1, a more pronounced increase in LGR was observed as groups became more discrepant in size at ICCs of 0.3 and 0.8; in particular, when the ICC was 0.3 or 0.8, a more dramatic increase in LGR was observed as the group size ratio moved to 75:25 than was the case with an ICC of 0.1. When two predictors were simulated at level-1 and equal group sizes, LGR increased from

approximately 60% to approximately 75% with an ICC of 0.3, and to approximately 88% with an ICC of 0.8. The increase in LGR as group sizes became more discrepant was less pronounced, though still apparent when data were simulated with one or two predictors at level-1. Broadly speaking, as the number of cases per cluster increased from 10 to 50, a slight increase was observed across all conditions, sans the condition in which the ICC was 0.1, the group size ratio was 50:50, and one predictor was at level-1: Under this set of conditions, an increasing number of cases per cluster resulted in marginally progressively lower LGR. Irrespective of the model used, the type of predictors and ICC substantively affected LGR with the number of cases per cluster only producing marginal differences.

Figure 3: LGR for Predictor Type by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster



Collectively, the above-discussed results broadly indicate an increase in LGR as group size ratios became increasingly more discrepant, more predictors were added at level-2, and ICC increased. While slight increases were seen under most cases as the number of level-2 clusters and number of level-1 cases per cluster increase, the increased LGR is less pronounced than is the case for the other conditions discussed. Under most cases, GLMM and LR yielded the highest LGR except in cases where one predictor was simulated at level-1, in which MERF yielded the highest LGR.

Outcome: Small Group Recovery

The second outcome considered was SGR. Within the simulation study, four four-way interactions were identified as statistically and practically significant: An interaction of method by predictor type by ICC by group size ratio ($F_{24, 368} = 1131.410$, $p < 0.001$, $\eta_p^2 = 0.987$); an interaction of method by predictor type by group size ratio by the number of level-2 clusters ($F_{24, 368} = 4.539$, $p < 0.001$, $\eta_p^2 = 0.228$); an interaction of predictor type by ICC by group size ratio by number of level-1 cases within clusters ($F_{16, 368} = 25.333$, $p < 0.001$, $\eta_p^2 = 0.524$); and an interaction of method by ICC by group size ratio by number of level-1 cases per cluster ($F_{24, 368} = 5.335$, $p < 0.001$, $\eta_p^2 = 0.258$). These interactions are shown in Table 6 (below).

Table 6: ANOVA Table for SGR Interactions

	Sum of Squares	df	Mean Square	F	p	η^2p
predictors	17.66362	2	8.83181	128288.244	< .001	0.999
method	1.02240	3	0.34080	4950.380	< .001	0.976
icc	5.54023	2	2.77011	40237.828	< .001	0.995
gsr	43.82195	2	21.91098	318272.288	< .001	0.999
ngroups	0.03850	2	0.01925	279.655	< .001	0.603
ncases	0.00168	2	8.40e-4	12.203	< .001	0.062
predictors ✻ method	1.88125	6	0.31354	4554.424	< .001	0.987
predictors ✻ icc	4.07499	4	1.01875	14798.031	< .001	0.994
method ✻ icc	0.18657	6	0.03109	451.669	< .001	0.880
predictors ✻ gsr	1.65379	4	0.41345	6005.617	< .001	0.985
method ✻ gsr	0.25255	6	0.04209	611.399	< .001	0.909
icc ✻ gsr	9.36077	4	2.34019	33992.939	< .001	0.997

Table 6: ANOVA Table for SGR Interactions

	Sum of Squares	df	Mean Square	F	p	η^2p
predictors ✻ ngroups	0.03729	4	0.00932	135.422	< .001	0.595
method ✻ ngroups	0.01061	6	0.00177	25.690	< .001	0.295
icc ✻ ngroups	0.01998	4	0.00499	72.549	< .001	0.441
gsr ✻ ngroups	0.00346	4	8.66e-4	12.582	< .001	0.120
predictors ✻ ncases	0.01852	4	0.00463	67.249	< .001	0.422
method ✻ ncases	0.01229	6	0.00205	29.742	< .001	0.327
icc ✻ ncases	0.03527	4	0.00882	128.082	< .001	0.582
gsr ✻ ncases	6.59e-4	4	1.65e-4	2.394	0.050	0.025
ngroups ✻ ncases	1.65e-4	4	4.13e-5	0.600	0.663	0.006
predictors ✻ method ✻ icc	0.17948	12	0.01496	217.262	< .001	0.876
predictors ✻ method ✻ gsr	0.79775	12	0.06648	965.655	< .001	0.969
predictors ✻ icc ✻ gsr	1.39454	8	0.17432	2532.090	< .001	0.982
method ✻ icc ✻ gsr	1.72752	12	0.14396	2091.125	< .001	0.986
predictors ✻ method ✻ ngroups	0.01683	12	0.00140	20.371	< .001	0.399
predictors ✻ icc ✻ ngroups	0.01588	8	0.00199	28.835	< .001	0.385
method ✻ icc ✻ ngroups	0.00408	12	3.40e-4	4.942	< .001	0.139
predictors ✻ gsr ✻ ngroups	0.01620	8	0.00203	29.423	< .001	0.390
method ✻ gsr ✻ ngroups	0.00753	12	6.28e-4	9.119	< .001	0.229
icc ✻ gsr ✻ ngroups	0.01001	8	0.00125	18.175	< .001	0.283
predictors ✻ method ✻ ncases	0.00293	12	2.44e-4	3.550	< .001	0.104
predictors ✻ icc ✻ ncases	0.01969	8	0.00246	35.758	< .001	0.437
method ✻ icc ✻ ncases	0.00232	12	1.93e-4	2.802	0.001	0.084
predictors ✻ gsr ✻ ncases	0.00553	8	6.92e-4	10.045	< .001	0.179
method ✻ gsr ✻ ncases	0.00321	12	2.68e-4	3.891	< .001	0.113
icc ✻ gsr ✻ ncases	0.05416	8	0.00677	98.332	< .001	0.681
predictors ✻ ngroups ✻ ncases	0.00167	8	2.09e-4	3.031	0.003	0.062
method ✻ ngroups ✻ ncases	0.00101	12	8.40e-5	1.220	0.267	0.038
icc ✻ ngroups ✻ ncases	4.61e-4	8	5.76e-5	0.837	0.571	0.018
gsr ✻ ngroups ✻ ncases	3.70e-4	8	4.62e-5	0.672	0.717	0.014
predictors ✻ method ✻ icc ✻ gsr	1.86937	24	0.07789	1131.410	< .001	0.987
predictors ✻ method ✻ icc ✻ ngroups	0.00382	24	1.59e-4	2.312	< .001	0.131
predictors ✻ method ✻ gsr ✻ ngroups	0.00750	24	3.12e-4	4.539	< .001	0.228
predictors ✻ icc ✻ gsr ✻ ngroups	0.00488	16	3.05e-4	4.432	< .001	0.162
method ✻ icc ✻ gsr ✻ ngroups	0.00299	24	1.25e-4	1.811	0.012	0.106
predictors ✻ method ✻ icc ✻ ncases	0.00589	24	2.45e-4	3.562	< .001	0.189
predictors ✻ method ✻ gsr ✻	0.00407	24	1.70e-4	2.462	< .001	0.138

Table 6: ANOVA Table for SGR Interactions

	Sum of Squares	df	Mean Square	F	p	η^2p
ncases						
predictors ✻ icc ✻ gsr ✻ ncases	0.02790	16	0.00174	25.333	< .001	0.524
method ✻ icc ✻ gsr ✻ ncases	0.00882	24	3.67e-4	5.335	< .001	0.258
predictors ✻ method ✻ ngroups ✻ ncases	2.06e-4	24	8.57e-6	0.125	1.000	0.008
predictors ✻ icc ✻ ngroups ✻ ncases	4.11e-4	16	2.57e-5	0.373	0.988	0.016
method ✻ icc ✻ ngroups ✻ ncases	4.15e-4	24	1.73e-5	0.251	1.000	0.016
predictors ✻ gsr ✻ ngroups ✻ ncases	7.34e-4	16	4.59e-5	0.667	0.827	0.028
method ✻ gsr ✻ ngroups ✻ ncases	9.34e-4	24	3.89e-5	0.565	0.953	0.036
icc ✻ gsr ✻ ngroups ✻ ncases	0.00110	16	6.89e-5	1.001	0.455	0.042
Residuals	0.02533	368	6.88e-5			

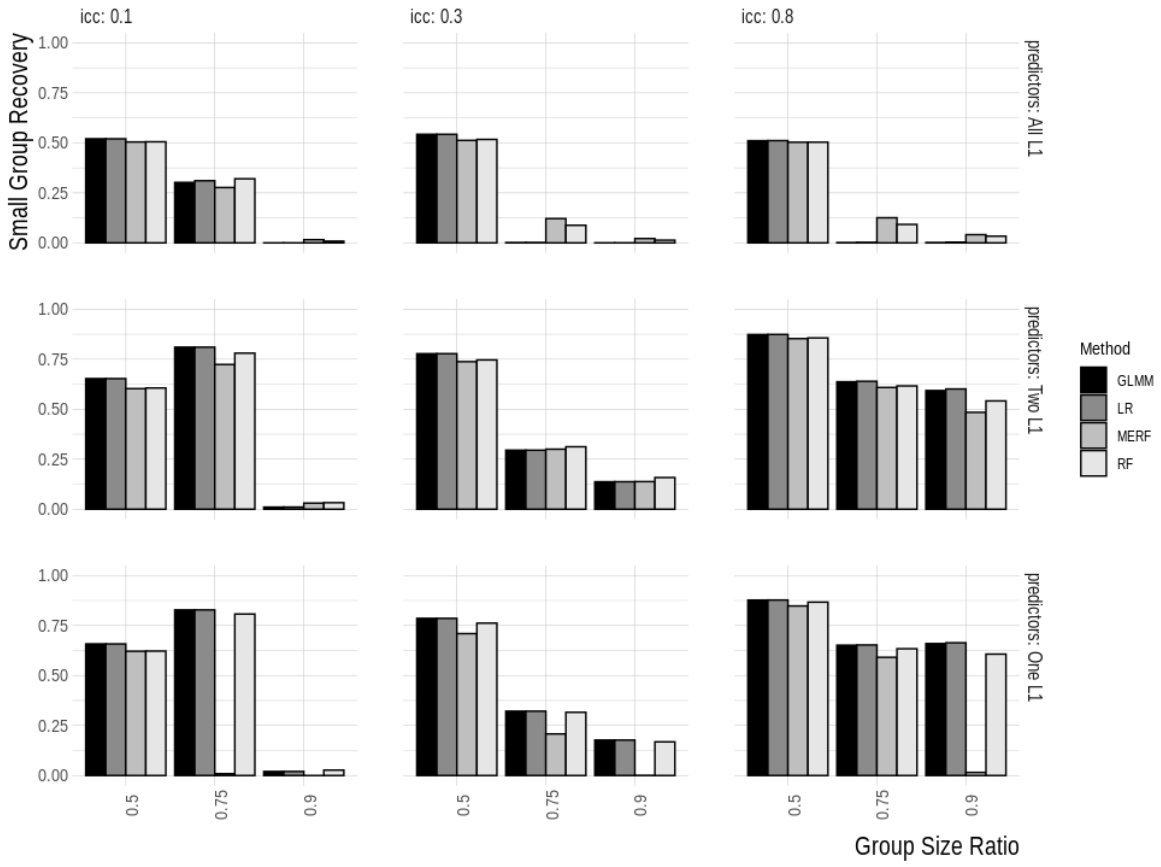
Method by Predictor Type by ICC by Group Size Ratio. The interaction of method by predictor type by ICC by group size ratio largely resulted in a pattern characteristic of that found in previous literature on classification with unbalanced data (e.g., Bolin & Finch, 2014; Lei & Koehly, 2003; Mangino & Finch, 2021) with a consistent decrease in SGR as groups became increasingly more discrepant in size (shown in Figure 4, below). However, deviations emerged when more predictors were added at level-2. When all predictors were simulated at level-1, SGR was approximately 50% with equal group sizes before decreasing appreciably as group sizes became increasingly more unequal. When the ICC was 0.1, the decrease was not as dramatic, though as the ICC increased to 0.3 and 0.8, SGR for LR and GLMM was functionally zero at both a 75:25 and 90:10 group size ratio. While RF and MERF decreased appreciably with values approaching zero in the unequal group size conditions, they still remained the highest with SGR of 5 – 10%.

With two predictors at level-1, the pattern dramatically changed: SGR increased notably from approximately 60% to approximately 75 – 78% for all methods when the group size ratio was 75:25 before plummeting to near-zero values within the 90:10 group size ratio. However, this pattern was only apparent when the ICC was 0.1. When the ICC was 0.3, a notable decrease was observed when moving from the 50:50 to the 75:25 and 90:10 conditions with RF retaining a slight advantage across all methods with unequal group sizes. When the ICC was 0.8, SGR dropped from approximately 88% to approximately 63% across all methods when the group size ratio was 75:25 and only diminished slightly when the group size ratio was 90:10. Under the ICC = 0.8 condition when the group size ratio was 90:10, LR and GLMM retained a slight advantage with RF performing at a slightly lower level and MERF falling below RF.

With one predictor at level-1, a similar pattern to that found with two level-2 predictors emerged, albeit with some deviation. When the ICC was 0.1, the same increase in SGR was observed when moving from a group size ratio of 50:50 to 75:25, though MERF dropped to functionally zero at this point; all other methods decreased appreciably to near-zero levels under the 90:10 condition with RF retaining a slight advantage. When the ICC was 0.3, LR, GLMM, and RF performed comparably one another, beginning with SGR at approximately 75% before decreasing appreciably to approximately 38% when the group size ratio shifted to 75:25 and decreasing again to approximately 20% when the group size ratio was 90:10; MERF was notably outperformed by all other methods at the 75:25 group size ratio and had SGR of functionally zero at a 90:10 group size ratio. At an ICC of 0.8, a similar pattern emerged as was the case with two level-1 predictors with a decrease from a 50:50 group size ratio to 75:25 before remaining relatively constant with a 90:10 group size ratio. Once again, MERF performed below all other methods at the 75:25 group size ratio and was functionally zero at the 90:10 group size ratio.

Broadly speaking, across all conditions, an increase in ICC also increased SGR under conditions with one or two predictors simulated at level-2, though it had little apparent effect between the ICC = 0.3 and ICC = 0.8 conditions when all predictors were at level-1. Further, as the number of predictors at level-2 increased, SGR tended to increase for all methods except MERF, which demonstrated substantially more volatility compared to LR, GLMM, and RF. The highest SGR rates were observed when the ICC was 0.8, one predictor was at level-1, and groups were equal in size.

Figure 4: SGR for Method by Predictor Type by ICC by Group Size Ratio

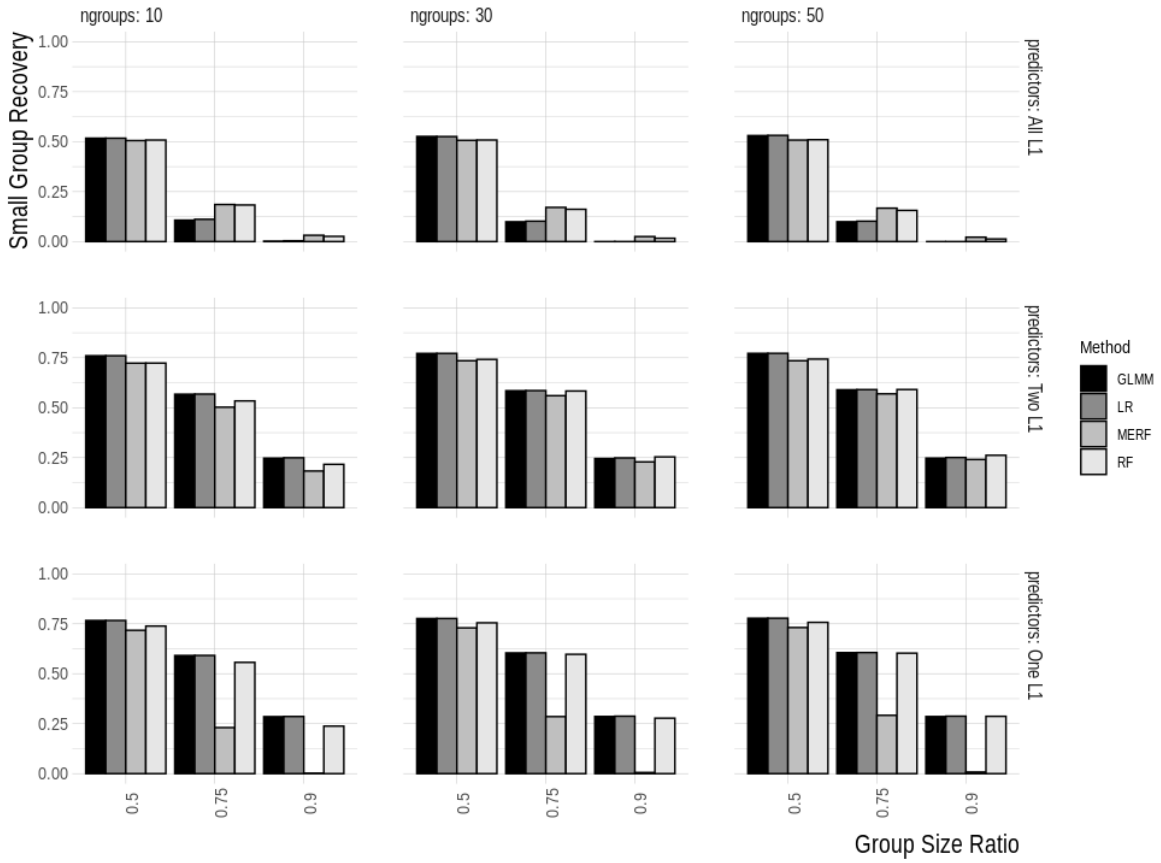


Method by Predictor Type by Group Size Ratio by Number of Level-2 Clusters. The interaction of method by predictor type by group size ratio by number of level-2 clusters resulted in a pattern similar to the above-discussed interaction (as shown in Figure 5, below). When all

predictors were simulated at level-1, the conventional pattern of decrease in SGR as group sizes became increasingly more discrepant was observed across all conditions. When groups were equal in size, LR and GLMM held a slight advantage compared to RF and MERF, but as groups became increasingly more discrepant in size, RF and MERF retained the advantage. This advantage was particularly apparent in the 90:10 group size ratio conditions in which LR and GLMM held functionally zero SGR while MERF and RF were near zero.

When two predictors were simulated at level-1, SGR was approximately 75% across methods when group sizes were equal, but decreased to approximately 60% when the group size ratio was 75:25 before decreasing to below 25% when the ratio was 90:10. This progressive decrease in SGR was apparent as the number of level-2 clusters increased, though all methods' performance improved slightly as the number of clusters increased to 50. When the number of clusters was 10, LR and GLMM held a slight advantage with RF performing at a slightly lower level and MERF performing the poorest. However, as the number of clusters increased to 30 and 50, RF held a slight advantage in all conditions of unequal group sizes. When one predictor was simulated at level-1, the same pattern as was found with two level-1 predictors again emerged: As the group size ratio became increasingly more discrepant, SGR tended to decrease. However, as this discrepancy was increased, MERF tended to decrease appreciably, performing at a level much lower than LR, GLMM, and RF under all conditions of unequal group sizes (reaching functionally zero when the group size ratio was 90:10. When the number of clusters was 10, LR and GLMM again held the advantage, but when the number of clusters was 30 and 50, LR, GLMM, and RF all performed comparably to one another. The highest SGR rates were observed with LR, GLMM, and RF when the number of clusters was 50 and one or two predictors were simulated at level-1.

Figure 5: SGR for Method by Predictor Type by Group Size Ratio by Number of Level-2 Clusters

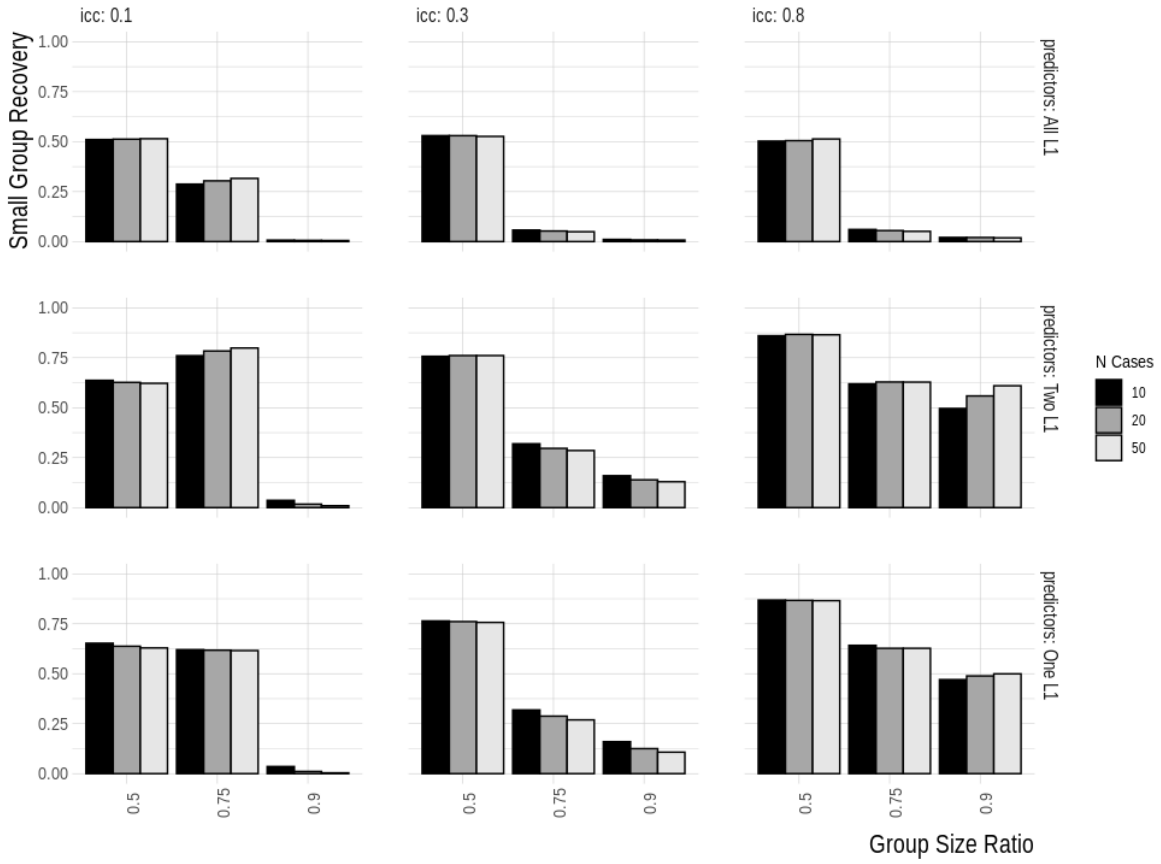


Predictor Type by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster. The interaction of predictor type by ICC by group size ratio by number of cases per cluster revealed a pattern akin to that found in the above-discussed interactions on the SGR outcome (as shown in Figure 6, below). When all predictors were situated at level-1 and the ICC was 0.1, an incremental decrease in SGR was noted as the group size ratio became more unequal. Further, in the 75:25 group size ratio condition, as the number of cases per cluster increased, so too did SGR. This decrease in SGR was more dramatic when the ICC was 0.3 and 0.8 with SGR at the 50:50 group size ratio being approximately 50% before decreasing to approximately 10% at a 75:25 group size ratio and near zero at a 90:10 group size ratio. With two predictors

simulated at level-1 and an ICC of 0.1, SGR increased when moving from the 50:50 group size ratio to the 75:25 group size ratio before plummeting to near-zero rates at a 90:10 group size ratio. However, this pattern did not replicate when the ICC was 0.3, with SGR instead diminishing appreciably—from approximately 75% to approximately 30%—when the group size ratio was 75:25 and decreasing again to approximately 15% when the group size ratio was 90:10. Additionally, SGR decreased slightly as cases per cluster increased. When the ICC was 0.8, SGR peaked when the group sizes were equal (approximately 88%) before dropping to approximately 65% with a group size ratio of 75:25. At a 90:10 group size ratio, SGR was approximately equivalent to its 75:25 rates, but began at a lower rate when the number of cases per cluster was 10 before increasing as the number of cases increased.

When one predictor was simulated at level-1 and the ICC was 0.1, SGR rates for both the 50:50 and 75:25 group size ratios were approximately equal at 65% before dropping to near-zero rates with a 90:10 group size ratio. When the ICC was 0.3 and groups were equal in size, SGR was held constant at approximately 75%, decreasing appreciably when the group size ratio was 75:25 and again when it was 90:10. As the number of cases per cluster increased was noted in the 75:25 and 90:10 group size ratios with an ICC of 0.3; this parallels the pattern observed when two predictors were simulated at level-1. When the ICC was 0.8, SGR peaked at approximately 88% when group sizes were equal before dropping to approximately 63% with a 75:25 group size ratio and again to approximately 50% at a 90:10 group size ratio. At the 90:10 group size ratio, SGR increased slightly as the number of cases per cluster increased. Broadly speaking, as the ICC increased, so too did SGR across all conditions. Similarly, as more predictors were simulated at level-2, SGR increased.

Figure 6: SGR for Predictor Type by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster

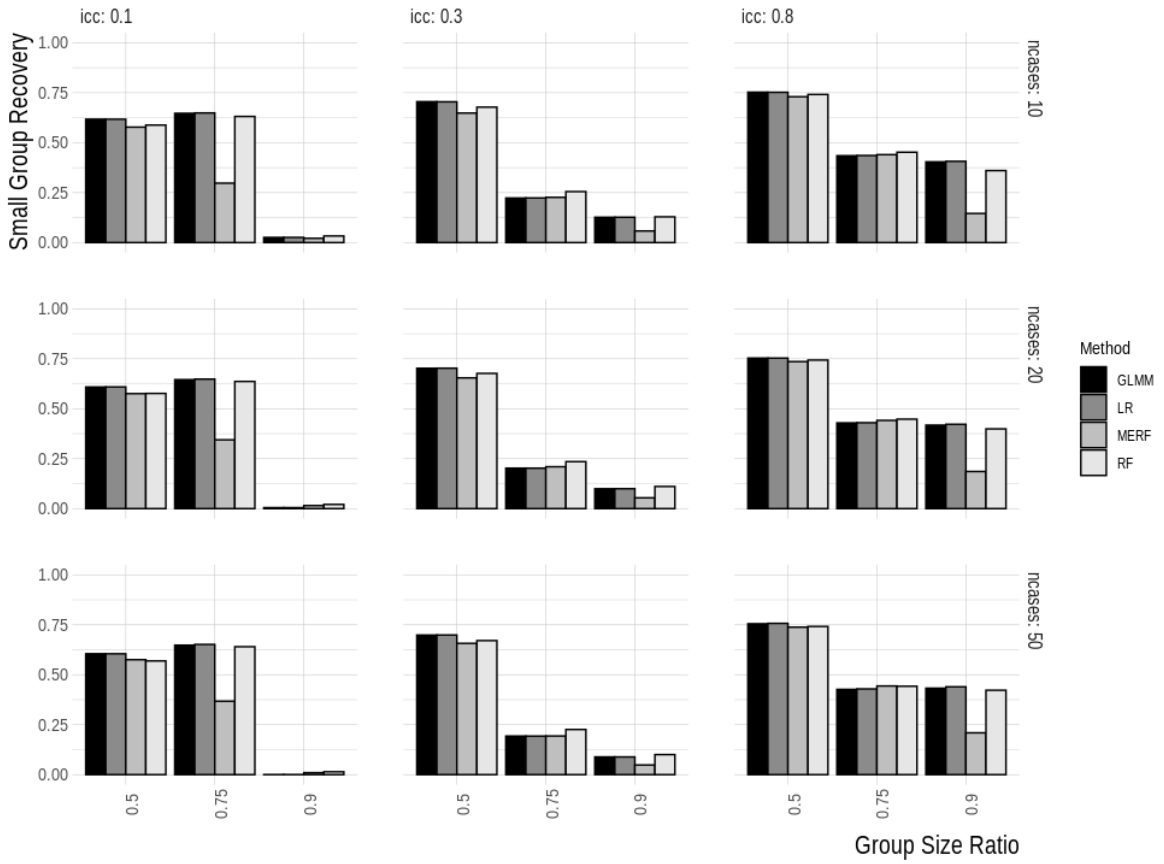


Method by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster. The interaction of method by group size ratio by ICC by number of cases per cluster revealed a pattern similar to those discussed above. Across all conditions varying the number of cases per cluster—whether 10, 20, or 50—the pattern in SGR was repeated (as shown in Figure 7, below). When the ICC was 0.1 and group sizes were equal, LR and GLMM slightly outperformed RF and MERF. When the group size ratio was 75:25, LR, GLMM, and RF all performed comparably and with a slightly higher SGR than when group sizes were equal; MERF was dramatically outperformed. When the group size ratio was 90:10, all methods performed at near-zero rates with RF performing the best of them all.

When the ICC was 0.3, SGR was higher for all methods when group sizes were equal, but decreased substantially when group sizes were unequal; however, none of the methods' SGR dipped as low as was the case for ICC = 0.1 while the group size ratio was 90:10. A similar pattern was observed when the ICC was 0.8, though the decrease when shifting from a 50:50 group size ratio to a 75:25 ratio was less dramatic. Further, when the group size ratio was 90:10, SGR for all methods (except MERF) was only marginally lower than in the 75:25 group size ratio. In the 75:25 group size ratio condition, RF slightly outperformed all other methods, but in the 90:10 group size ratio, LR and GLMM performed the best.

Broadly, as the number of cases per cluster increased, slight decreases in SGR were observed for all conditions in which ICC was 0.1 or 0.3, but did not manifest when the ICC was 0.8. Rather, when the ICC was 0.8 and group sizes were unequal, a very slight and progressive increase in SGR was noted as the number of cases increased. Conversely, when the ICC was 0.1 and the group size ratio was 90:10, a slight decrease in SGR was noted as the number of cases per cluster increased. Generally speaking, SGR increased as the ICC increased, peaking when the ICC was 0.8, group sizes were equal, and there were 50 cases per cluster. Across all sets of conditions, GLMM, LR, and RF all performed comparably to one another, with either GLMM and LR or RF slightly outperforming all others; MERF tended to yield more volatile results and tended to perform the worst of all methods.

Figure 7: SGR for Method by ICC by Group Size Ratio by Number of Level-1 Cases per Cluster



Summary of Simulation Results

Holistically, the results of the simulation study indicated a pervasive and conditional impact of the predictor structure—the levels at which the predictors were simulated—and the ICC. These two factors were consistent across nearly all interactions. Within the LGR outcome, largely across all methods, as the ICC increased and more predictors were simulated at level-2, LGR increased appreciably. This pattern was evident across all group size ratios, but was most notable when groups were equal in size as a higher ICC alone did not appreciably increase LGR, but a higher ICC coupled with more level-2 predictors did result in higher LGR rates.

Regarding the SGR outcome, the resulting patterns observed within the SGR interactions revealed that as group sizes became increasingly more unequal, SGR tended to decrease

appreciably, though the nature and amount of this decrease depended on other factors. As the number of level-2 predictors and the ICC increased, so too did SGR. Further, no single method singularly demonstrated a notable performance, though MERF was consistently outperformed by all other methods under nearly all conditions.

Archival Examination

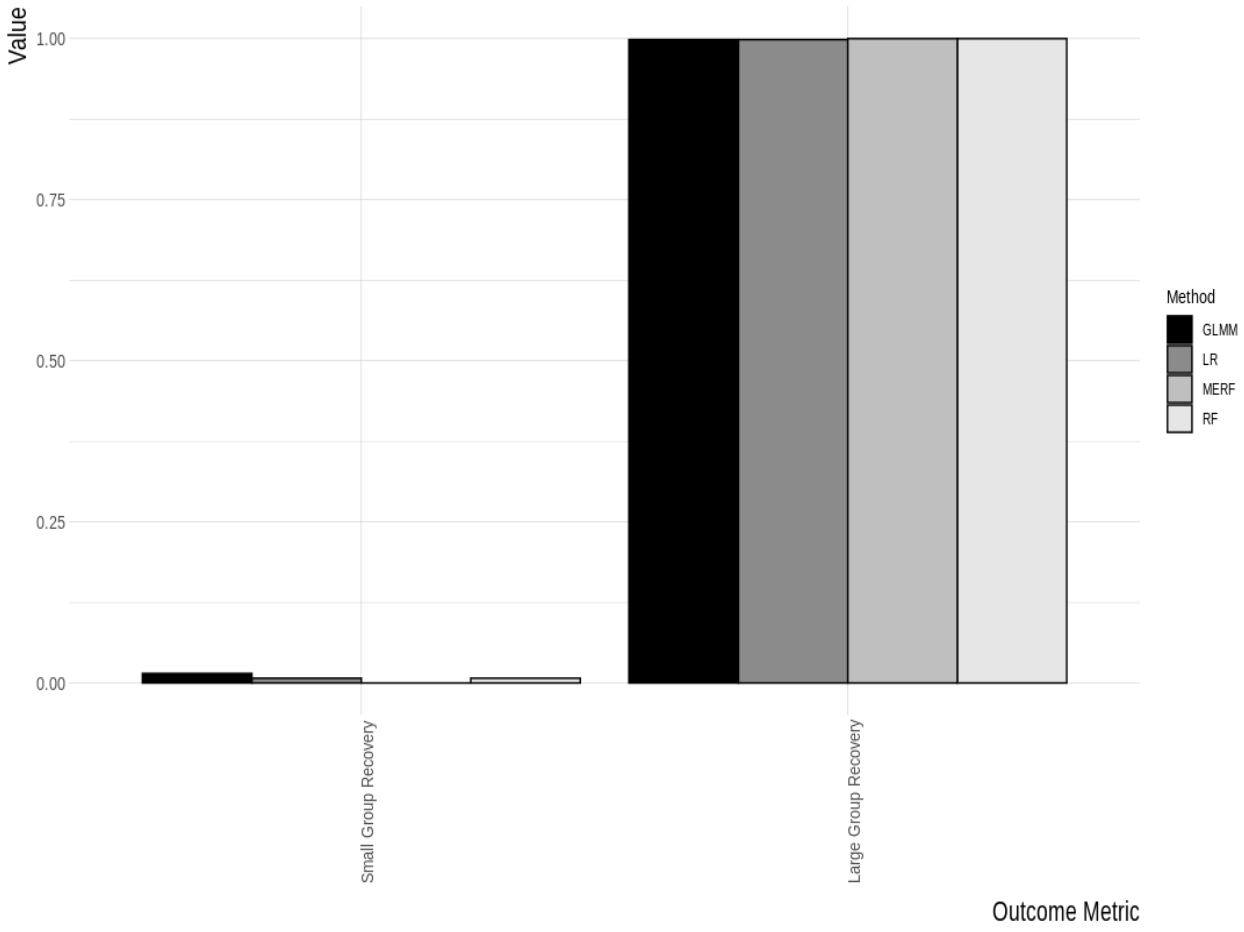
In the prediction of whether students were held back in the public version of the PISA dataset, both LGR and SGR metrics were obtained for all methods. Further, to given the methods used and the unequal group size ratio, probabilities of membership in the repeater group were obtained and used used with a default classification probability of 0.5. That is, if the probability of membership in the repeater group was greater than 0.5, then the case was classified into this group; otherwise, the case was classified into the non-repeater group. The results of this examination are shown in Figure 8 (below).

The conditions of the imputed PISA dataset most closely resembled those seen in the simulation conditions with the largest number of clusters and number of cases per cluster, an ICC of 0.1, a group size ratio of 90:10, and three level-1 predictors. Under these conditions, the classifiers all yielded high LGR rates (approaching 100% for all methods) with near-zero SGR with MERF and RF just slightly outperforming LR and GLMM.

In the PISA data, LGR was near 100% and SGR was near 0% across all methods. This pattern is common in unequal group size settings in which all (or nearly all) cases are classified into the larger of two groups due to the algorithm attempting to maximize overall accuracy (e.g., Lei & Koehly, 2003). Among the methods compared, GLMM yielded the highest SGR (1.48%) while all other methods yielded values < 1% and MERF yielding a 0% SGR. For LGR, RF and MERF both yielded the highest LGR with values of 100%; LR and GLMM both yielded values in excess of 99%. All results for the PISA investigation are shown in Figure 8 (below). The

results of the PISA investigation largely paralleled those of their most similar simulation conditions, as expected.

Figure 8: Accuracy Metrics for all Classifiers on PISA Grade Repetition



CHAPTER 5: DISCUSSION

Given the results of both the simulation and archival data examinations discussed in Chapter 4, the hypotheses described in Chapter 3 can be effectively substantiated or refuted. Presently, this chapter details the outcomes of each hypothesis, synthesizes the findings of the two investigations, describes practical recommendations for researchers with respect to the intersection of analytical paradigms presently investigated, and limitations of the study are detailed.

Hypotheses

Across all conditions, little to no appreciable difference will be found between each fixed effects model and its mixed effects analogue. It is hypothesized that the fixed effects RF algorithm will perform as well as MERF, and LR and GLMM perform comparably to one another on both LGR and SGR metrics. (H1)

Considering H1, the evidence largely supported the contention that little to no difference existed in LGR and SGR when comparing the four methods to one another, and each fixed effects model to its mixed effects analogue. Across nearly all cells of each interaction for both outcomes, the similarity between the four algorithms was striking with LR, GLMM, and RF largely performing comparably to one another. While MERF performed comparably to the other three algorithms under most conditions, it became notably more volatile in SGR when more predictors were shifted to level-2 and the ICC increased. With the caveat of MERF's performance under conditions of higher ICCs and fewer predictors simulated at level-1, this hypothesis is supported.

Under conditions of unequal group size ratios, all classifiers will yield higher LGR rates (percentages) with simultaneously lower SGR values compared to conditions of equal group sizes ratios. (H2)

With respect to H2, the hypothesis was uniformly supported by both the simulation and PISA examinations. Within the simulation, a notable decrease in SGR was accompanied by an appreciable increase in LGR across all classifiers when the group size ratio shifted from 50:50 to 75:25 and 90:10. This finding corroborates those from previous studies (e.g., Bolin & Finch, 2014; Lei & Koehly, 2003; Mangino & Finch, 2021) and indicates that the large-group bias was a prevalent and pervasive determinant of classifier performance, regardless of whether the classifier was fixed or mixed effects. As expected, this commonly found result was demonstrated in the present study.

Under conditions of increasing cluster size and number of clusters, all classifiers will increase in both LGR and SGR rates. Corollary, as the total sample size increases, so too will both LGR and SGR rates. (H3)

In contrast, H3 was not uniformly supported, but rather was only supported under some conditions. Little change was observed in LGR as the number of level-2 clusters was increased in the simulation context, and this was accompanied by a slight increase in SGR as the number of clusters increased. This change in the number of groups did not affect the classifiers differently, as the classifiers each performed comparably to one another (with the exception of MERF on the SGR metric). In considering this hypothesis relative to the total sample size corollary discussed in Chapter 2, the absence of an appreciable change in LGR and SGR as total sample size changed (through either cluster size or the number of clusters) suggests that when considering raw predictive accuracy, sample size may be a less salient determinant of accuracy metrics. Consequently, it has been demonstrated that despite likely being underpowered (according to McNeish & Stapleton's [2016] assertions), SGR and LGR may not diminish.

Additionally, when ANOVAs were run with a computed "Total Sample Size" variable (where N = Number of clusters * number of cases per cluster) in lieu of the disaggregated

number of cases and number of clusters, statistically, but not practically, significant interactions were found between sample size, group size ratio, ICC, and predictor type ($F_{8, 780} = 6.943$; $p < 0.001$, $\eta_p^2 = 0.06$) for SGR, and interactions of sample size, group size ratio, ICC, and predictor type ($F_{8, 780} = 3.13$; $p < 0.01$, $\eta_p^2 = 0.03$) and sample size, group size ratio, predictor type, and method ($F_{12, 780} = 62.681$; $p < 0.01$, $\eta_p^2 = 0.03$) for LGR. The absence of practical significance of these interactions including total sample size further indicated that sample size (limited to values of N = 100 – 2500) is likely not a salient factor in classification accuracy. All evidence considered, H3 may be plausible if considered in conjunction with larger sample sizes, but is not fully supported when total sample sizes are limited to N = 2500.

Under conditions of increasing ICC, all classifiers will yield lower LGR and SGR rates. However, it is hypothesized that both RF and MERF demonstrate this diminution of accuracy to a lesser extent than LR and GLMM. (H4)

Similarly, H4 was broadly not supported, sans for one particular situation. While an increase in ICC did result in a diminution of SGR, it only did so when all three predictors were simulated at level-1. When one or more predictors were simulated at level-2, SGR actually increased as the ICC increased. A similar pattern was observed for LGR with values being initially high, but increasing to nearly 100% when the ICC increased and when one or more predictors were simulated at level-2. Further, with respect to the differential performance of the models, LR, GLMM, and RF all performed comparably to one another—only varying slightly across all sets of conditions—while MERF yielded consistently lower SGR (particularly when only one predictor was simulated at level-1) and LGR similar to RF; MERF only had an advantage in SGR when the ICC was 0.3 or 0.8 and all predictors were simulated at level-1, and an advantage in LGR across all conditions when one predictor was simulated at level-1.

Therefore, H3 can largely be refuted with a caveat being the level at which the predictors were simulated must be entirely generated at level-1.

Across all conditions, RF and MERF will demonstrate the greatest predictive efficacy with the highest SGR and LGR rates, outperforming both LR and GLMM. (H5)

The final hypothesis of the simulation study, H5, was only partially supported. On the LGR outcome, no method demonstrated any notable advantage with MERF only slightly outperforming the other methods on conditions in which only one predictor was simulated at level-1. Within the SGR outcome, MERF was consistently outperformed by LR, GLMM, and RF on all conditions when predictors were simulated at level-2; rather, MERF only held an advantage—one that it shared with RF—on the SGR metric when all predictors were simulated at level-1. When predictors were simulated at level-2, all methods (except MERF) performed similarly to one another. Rather, the comparable performance of LR, GLMM, and RF indicated no substantive difference between these methods on either SGR or LGR. Therefore, this hypothesis can only be partially supported.

Considering the hypotheses within the PISA examination, the results were consistent with previous research in the classification literature base as well as the results of the simulation study.

In the prediction of student retention, the LGR and SGR outcomes of all algorithms will largely mirror their simulation study results for the conditions most closely resembling those of the PISA dataset. (H6)

Regarding H6, the comparison must be made between the simulation condition most readily matching those conditions found in the PISA data. The retention outcome is highly unbalanced, approximately aligning with the 90:10 group size ratio condition. Additionally, the ICC of the PISA data was calculated at 0.13, thus aligning with the ICC = 0.1 conditions in the

simulation. Further, all predictors in the PISA data were situated at the student-level, thus aligning with the conditions in which all predictors were simulated at level-1. The average number of cases per cluster in the PISA data was eight, thus closely corresponding with the simulation condition in which ten cases per cluster were simulated. Similarly, while the number of clusters in the test split of the PISA dataset was 88 and the highest number of clusters simulated was 50, the total sample size was greater for some conditions in the simulation than was the case for the PISA dataset ($N_{PISA} = 704$; $N_{simMax} = 2500$). Further, the lack of evidence in support of the effects of the number of clusters on LGR and SGR indicates this factor will be subsumed under more salient data conditions in the PISA examination. In the LGR outcome all classifiers had values near 100%, thus indicating that no single method held a particular advantage. Simultaneously, SGR was at near-zero values for all classifiers in the PISA data. This effect was likely present due to the preeminent impact of the highly unequal group size ratio in the PISA data's REPEAT outcome, particularly due to all predictors being individual-level with a low ICC. With the simulation results indicating that lower ICCs with all predictors being simulated at level-1 and a 90:10 group size ratio, it is likely these factors were the most salient in affecting the accuracy rates in predicting grade repetition in the PISA data.

Accordingly, SGR was consistently lower for all methods than was LGR. For all methods, LGR was near 100% while SGR values were near-zero; GLMM held the highest SGR, though this was still only 1.48%. All methods demonstrated the same pattern of high LGR and near-zero SGR. Under this condition, LR and GLMM were the next most efficacious methods with SGR < 1%; MERF yielded SGR of 0%. The consistency across the simulation conditions most similar to the conditions present in the PISA data largely supports H6. None of the classifiers were particularly efficacious under the conditions of the PISA data, just as they were

all similarly poor-performing under similar conditions in the simulation. Holistically, it can be concluded that no method was presented as the uniformly superior algorithm.

Synthesis

The results of this study broadly indicated that the question of fixed or mixed effects classifiers in the presence of nested data is far from simplistic. Rather, factors affecting both SGR and LGR are found in both the classification and multilevel modeling literature bases. In particular, the effects of group size ratio, type of predictors, and ICC had considerable impact on the SGR outcome (LGR as well, though to a lesser extent). Given that the smaller group is typically of greater interest when considering binary classification problems (presently, this is the repeater group in the PISA data), the substantive differential effects of the predictor type and ICC across various group size ratios indicated that classification in a multilevel context may become a simpler endeavor when more features of the higher-level cluster are incorporated. That is, as the ICC increased and more level-2 predictors were added to the models, LR, GLMM, and RF all improved appreciably in both outcome metrics (more notably SGR). This finding aligns with Kilham et al.'s (2019) finding that including a mix of level-1 and level-2 predictors can allow for a greater degree of model accuracy in classification settings; Kilham et al.'s finding was replicated in the present simulation setting.

Of particular note is the appreciable increase in both LGR and SGR metrics as both the ICC increased and as more predictors were simulated at level-2. As was found in the study by Kilham et al. (2019)—with respect to the inclusion of tree-level and plot-level variables to predict tree-level harvests—the inclusion of level-2 predictors allowed for a more holistic representation of the *in vivo* setting of the outcome being predicted. While perhaps statistically (thought not conceptually) counterintuitive, this increase in both LGR and SGR (and consequential increase in overall accuracy) is likely due to the inclusion of context as a salient

factor within the model being estimated. When all predictors were simulated at level-1, SGR rates began lower than when at least one predictor was simulated at level-2 and plummeted when the group size ratio became unequal (this was more pronounced with ICC = 0.1). However, this phenomenon was not observed to nearly the same extent when the ICC was 0.8 and at least one predictor was simulated at level-2. Two mechanisms appear to be at work in this situation: The greater emphasis on the context (level-2 cluster membership) as a key factor in accounting for variability in the outcome (increased ICC); and the inclusion of contextual factors (level-2 variables) as predictors of outcome groups. Just as Bronfenbrenner and Morris' (2006) bioecological model/ecological systems theory proposes the inexorable entanglement of person and context, so too could it be said of the statistical representations of these factors, given by the results of the present investigation.

Of additional note is the consideration of the small sample size problem as a delimiter of the present investigations. As was noted in Chapter 2, estimating a mixed effects model can be performed and may yield relatively accurate parameter estimates, but is likely to be underpowered (McNeish & Stapleton, 2016). However, this degradation in statistical power may not result in a concomitant decrease in LGR and SGR metrics. As noted in Chapter 2, a smaller sample size in classification models with multilevel data—by way of either smaller cluster size, fewer clusters, or both—may be accompanied by a concomitant increase in model computed error metrics, even if not necessarily raw accuracy rates (e.g., LGR and SGR; Beleites et al. [2012]; Figueroa et al. [2012]; Raudys & Jain [199]) when models are estimated on smaller samples.

The absence of a decrease in LGR and SGR with smaller sample sizes was most apparent in the fixed effects models RF and LR. Despite their computational simplicity compared to mixed effects models, the predictive capabilities of all models currently employed (sans MERF

under some conditions) were comparable to one another. One possible explanation for this is in the metrics used: SGR and LGR. When considering regression models in the presence of nested data, it is often strongly recommended that mixed effects models be used as there exists an appreciable bias in parameter estimates when explaining or predicting continuous outcomes (Cunningham, 2021; Snijders & Bosker, 2011). However, when considering solely the raw accuracy metrics in a predictive classification setting—in which parameter estimates and computational accuracy are arguably less important than in explanation settings—the metrics themselves may be less sensitive to this bias. For example, if a regression model predicted the outcome of two cases as being 0.85 while the actual values were 0.8 and 0.9, respectively, then the model would be in error 0.05 for both cases. However, in a classification setting, predicted probabilities of 0.85 for both cases would still result in classification decisions of group 1 for both cases (assuming equal classification probabilities of 0.5 for both groups as was presently used). As discussed in the limitations section below, the presently used metrics alone may not convey a fully representative or accurate portrayal of the models' efficacy.

Of additional note regarding sample size was the presence of convergence errors in the estimation of MERF. On 17 occasions (12 while all predictors were simulated at level-1, one with two predictors at level-1, and two while one predictor was at level-1) the MERF algorithm was unable to compute its desired internal accuracy metrics and, thus, yielded an error. The conditions under which this error was encountered in condition 72 of 243 when the group size ratio was 90:10 and ICC = 0.3 with 50 clusters and 50 cases per cluster. It is suspected that the 90:10 group size ratio was the principal problematic condition, but was also seen with a 75:25 group size ratio when one predictor was at level-1. The commonality among these conditions was the largest sample size of 50 clusters with 50 cases each. Therefore, the difference between larger and smaller sample sizes should be further examined within this model's function call to

determine the source of this non-convergence. Incidences of the above-discussed errors are shown in Table 7 (below).

Table 7: Simulation Error Incidences

Error Code	**Conditions & Model**	**Number of Occurrences**
`Something is wrong; all the ROC metric values are missing: ...` `Error in the Expression: : original error message = Stopping`	Model: MERF; Predictors = ALL L1; ngroups = 50; ncases = 50; ICC = 0.3; Group size ratio = 90:10	10
	Model: MERF; Predictors = ALL L1; ngroups = 10; ncases = 10; ICC = 0.8; Group size ratio = 90:10	2
	Model: MERF; Predictors = Two L1; ngroups = 50; ncases = 50; ICC = 0.3; Group size ratio = 90:10	1
	Model: MERF; Predictors = One L1; ngroups = 50; ncases = 50; ICC = 0.8; Group size ratio = 75:25	1
	Model: MERF; Predictors = One L1; ngroups = 50; ncases = 50; ICC = 0.3; Group size ratio = 90:10	1

Taken holistically, the results of this investigation largely indicate that regardless of the model selected, the SGR and LGR results are approximately equivalent within smaller samples: Across the conditions assessed presently, LR, GLMM, and RF all performed comparably to one another, usually within a 1-3% differential. This result indicates that it is rather the conditions present within the data, not necessarily the classifier itself, that has the most prominent effect on prediction accuracy. Conversely, the MERF algorithm stood out as having the least reliable results, particularly with respect to SGR. While MERF's LGR was consistently high—largely aligning with that of the other algorithms—its SGR was either commensurate with that of RF when all predictors were simulated at level-1 or when one was simulated at level-2, but fell dramatically in nearly all conditions when two level-2 predictors were included. That this

reduction was observed most often when group sizes were unequal, it is likely that MERF is considerably biased toward the larger group compared to the other methods considered. Further, while no level-2 predictors were used in the PISA examination, the pattern of SGR and LGR across classifiers largely paralleled their values within the simulation for conditions in which all predictors were simulated at level-1. Therefore, while there was little difference between classifiers under most conditions, it is clear that the implementation of MERF requires additional investigation and careful tuning in order to make the algorithm viable for predictive classification purposes. Nonetheless, it is apparent that the findings of Kilham et al. (2019) and Speiser et al. (2019) demonstrating similar performance of RF, its mixed effects analogue MERF, and GLMM were largely supported within the present study.

Limitations

This study sought to assess the question of predictive capability of fixed- and mixed-effects models in multilevel data with small samples through both simulation and archival data examinations. Despite the robust nature of the study, several limitations are noted with potential corrections and extensions proposed.

One key point noted in both the Kilham et al. (2019) and Speiser et al. (2019) articles is the statement that while RF may perform similarly to mixed effects classification frameworks (including GLMM, MERF, and Speiser et al.'s BiMM Forest), fixed effects models such as RF do not effectively account for the nesting structure of multilevel data and, thus, do not provide the most accurate conceptual or statistical representation of the phenomenon being investigated. While the inclusion of both level-1 and level-2 predictors better accounts for this representation, it does not account for the effect of the nesting structure itself. As discussed above, the metrics presently used do not accurately convey the inability of fixed effects models to effectively account for the nesting structure of data. Therefore, the limitation of model specification could

be better understood by the utilization of alternative model accuracy metrics. Similarly, it should be noted that all models were fit as two-level models with random intercepts only in order to account for the nesting structure of the data. The LGR and SGR results presently obtained may differ (particularly in archival data examinations, such as the PISA data presently used) if models were specified with random coefficients. The consideration of nesting structure and the degree to which random effects should be included would become notably more difficult in the context of three-level models with a tertiary nesting structure. Consequently, additional metrics would be required in the form of the AUC, CE, or RMSE to more fully explore model *computational* accuracy compared to raw prediction accuracy in the form of SGR and LGR.

A secondary limitation is in the verification of results across the simulation and PISA examinations due, in large part, to the conditions simulated and variables selected in the PISA dataset. In order to maintain continuity across both investigations, the constellation of predictors used were all continuous in nature. However, many of the variables that would likely be more eminently related to the outcome of student retention were either coded into categorical representations of continuous variables, or were Likert-type items with a limited range of potential values. Given the exploratory nature of the present study, the use of continuous predictors was reasonable, though future investigations would likely make use of categorical or restricted-range numeric (i.e., Likert-type) variables, particularly with various non-normal distributions. Were these item types and distributions to be used as predictors, it is likely that level-2 predictors could be used in the PISA examination so as to provide a more holistic consideration of the effects of contextual variables (level-2 predictors) on prediction accuracy. While the actual LGR and SGR when predicting student retention in the PISA data are unlikely to deviate appreciably from their values obtained in the present study, it is likely that the expansion of the level of predictors used would result in different results than those presently

obtained (matching the LGR and SGR found in the simulation with one or more level-2 predictors).

A tertiary limitation could be found in the form of the four-way ANOVA interactions considered. In many cases, five-way interactions were found to be statistically and practically significant, but were functionally uninterpretable and, thus, severely limited in utility. The five-way interactions tend toward a case similar to common critiques of maximalist theoretical models (e.g., Bandura's social cognitive theory; Bronfenbrenner's bioecological model) in which all factors are entangled and relevant, but cannot sufficiently be used to explain observed outcomes. Therefore, while the factors identified within the present analyses were shown to have an appreciable impact, the question of true magnitude and explanatory power stands. For example, the interaction of method by group size ratio by predictor type by number of level-2 clusters revealed a striking pattern for all conditions except the change in the number of clusters: While SGR and LGR did increase as the number of clusters increased, the extent to which this increase was observed was functionally negligible (note the substantial difference in effect size between the Method by Predictor Type by ICC by Group Size Ratio interaction compared to one including number of groups or number of cases in lieu of ICC). Therefore, a more targeted simulation with fewer conditions may be warranted to more fully isolate those factors most germane to changes in accuracy metrics.

A final limitation pertains to a potential ceiling effect observed through the use of a limited range of sample size conditions. Little difference was found in LGR and SGR between a sample size of N = 100 (10 clusters of size 10) and N = 2500 (50 clusters of size 50) across the presently studied methods (as corroborated by non-practically significant interactions including total sample size). However, such a difference may arise when comparing conditions with sample sizes from 100 to 10000. Additionally, larger sample size conditions—coupled with the

presently used smaller sample size conditions—should be incorporated in subsequent investigations to better determine whether an appreciable difference exists in accuracy metrics across more extreme sample sizes, both large and small. The use of more extreme sample sizes would allow for a more holistic consideration of the effects of this factor.

Practitioner Recommendations

Given the results of the present investigation, several recommendations for practice may be proposed. When considering model selection—fixed or mixed effects—the research questions, type of data collected, and purpose of model estimation should be carefully and intentionally defined. The findings of this study largely matched those found by Kilham et al. (2019) and Speiser et al. (2019) with respect to fixed effects models, specifically RF, being viable prediction models despite the multilevel structure of data. Therefore, it is likely unnecessary for researchers to use multilevel classifiers for the purpose of *predictive* classification (in which raw accuracy metrics are favored over accurate parameter estimates), as the results largely do not differ substantially. Conversely, this result does not indicate any recommendation for models estimated for the purpose of *explanation* (in which parameter estimates are more eminently salient) and, instead, would recommend against the use of MERF for this purpose as not only were its predictions less consistent than the other models, but its R implementation does not feature any explanatory information (e.g., variable importance, coefficient estimates). Therefore, the findings of this study indicate that LR, GLMM, and RF are all approximately equal in predictive capability, regardless of the multilevel data structure.

A second recommendation is drawn from the pattern of increase in both LGR and SGR as more predictors were simulated at level-2 and the ICC increases; as the salience of context increases, it then becomes increasingly more important to consider these factors within the model chosen. That is, it is strongly recommended that variables measured at both the case and cluster

levels be included in classification models when data are nested in nature. Considering both the simulation and PISA examination results, only including case-level predictors resulted in notably lower SGR, particularly when outcome group sizes were unequal; this was particularly apparent in the PISA data, which yielded near-zero SGR across all models. It should be noted that in the simulation context, even when groups were highly unequal in size (90:10 ratio), both LGR and SGR rates were improved (in the case of SGR, substantially so) when even one level-2 predictor was included, particularly when the ICC was 0.3 or larger. Therefore, a holistically representative collection of predictors should be employed in predictive classification settings in order to ensure classifiers are provided with sufficient information to make reasonable predictions.

Collectively, the results suggest that when accurate predictions are desired and the explanatory power of the predictors is not of concern, LR, GLMM, or RF would all be eminently appropriate in this task provided that a constellation of both level-1 and level-2 predictors is employed, particularly if the ICC is 0.3 or higher.

Future Directions

Given the limitations of the present study and niche focus on *predictive* classification, additional research is needed in order to determine the degree to which these results would be found for explanatory classification models. It could be hypothesized that, as is the case with regression models, coefficient estimates may be biased while raw classification metrics may be insensitive to this bias. Further, it could be hypothesized that SGR and LGR for the training set (for which explanatory power is afforded through the estimation of model parameters) would be higher than the values observed presently for prediction. This is common in prediction settings in which models are initially trained, but may not generalize well to new data (Steyerberg, 2019). Therefore, an investigation into the effects of multilevel data on the computation of explanatory classifiers—considering both parameter estimates and classification accuracy—may be

warranted and contrasted with the results of the present study. Similarly, within this study, the AUC and/or CE should also be considered in order to determine whether an appreciable difference exists in the computational accuracy of the models employed.

A secondary preeminent future direction would be an expansion of the type of predictors to those categorical or range-restricted numeric (Likert-type items) in nature. Given that much of the data in the social sciences are measured on Likert-type scales (or, minimally, seldom truly continuous), it is evident that this type of variable must be considered with respect to their effects on both fixed- and mixed-effects classifiers. Similarly, the consideration of various distributions of predictors (heavily skewed, bimodal, etc.) may become eminently reasonable. Several predictors used in the PISA dataset were heavily left-skewed in nature, in contrast to the normally distributed predictors of the simulation. While the results of both the simulation and the PISA examination were largely consistent, the effects of the different predictor distributions are not well-documented, particularly when observed in Likert-type items. Therefore, the consideration of both alternative predictor distributions and types should be a notable area of further inquiry.

Conclusion

The task of prediction is neither a simplistic nor lightly-approached endeavor, particularly when the complexities of reality enter into the realm of statistical representations (e.g., nested data structures). As was noted initially with respect to Bronfenbrenner's model (Bronfenbrenner & Morris, 2006), the entanglement of person and context is inevitable, thus dictating the importance of considering multilevel data. The present study sought to examine the eminently complex phenomenon of predictive classification in the presence of multilevel data in order to uncover the simplicity within the phenomenon. Through both simulation and archival data examinations, it was found that three such classifiers used—LR, GLMM, and RF—all performed

similarly to one another under most data conditions simulated with results verified in the PISA examination. Similarly, it was found that when incorporating factors at both the case- and cluster-level in multilevel contexts, most models performed substantially better than when only level-1 predictors were used. Therefore, the theoretical postulate that both the micro-level unit (e.g., person, time point) and context have an impact on outcomes was shown to be supported. Consequently, it is important for researchers to consider and represent all reasonably available levels of data when engaging in predictive modeling. Ultimately, with respect to estimating and using prediction models, it is in the model itself that simplicity may be maintained, as little difference was found between fixed and mixed effects classifiers. However, it is in the data themselves that complexity is found. In order to make irreducible the basic elements of this contention: A simpler model, when used properly despite complex data, may provide a more parsimonious and equally efficacious tool in the complex task of prediction.

References

Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. In *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit (eCrime ' 07)*. Association for Computing Machinery, New York, NY, 60-69. doi:https://doi-org.proxy.bsu.edu/10.1145/1299015.1299021

Allison, P. D., & Waterman, R. P. (2002). Fixed-effects negative binomial regression models. *Sociological Methodology, 32*(1), 247-265. doi:10.1111/1467-9531.00117

Altman, D. G., & Bland, J. M. (1994). Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, *308*(6943), 1552.

Bagiella, E., Sloan, R. P., & Heitjan, D. F. (2000). Mixed-effects models in psychophysiology. *Psychophysiology, 37*(1), 13-20.

Bates, D., Maechler, M., Bolker, B., Walker, S., (2015). Fitting Linear Mixed Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, *36*(1-2), 105-139.

Beleites, C., Neugebauer, U., Bocklitz, T., Krafft, C., & Popp, J. (2012). Sample size planning for classification models. https://doi.org/10.1016/j.aca.2012.11.007

Bolin, J., & Finch, W. (2014). Supervised classification in the presence of misclassified training data: A monte carlo simulation study in the three group case. *Frontiers in Psychology, 5(118)*. https://www.doi.org/10.3389/fpsyg.2014.00118

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123-140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Bronfenbrenner, U., & Ceci, S. J. (1994). Nature-nurture reconceptualized in developmental perspective: A bioecological model. *Psychological Review, 101*(4), 568-586. doi:10.1037/0033-295X.101.4.568

Bronfenbrenner, U., & Morris, P. A. (2006). The bioecological model of human development. In the *Handbook of child psychology*, *1*.

Cameron, A. C., & Trivedi, P. K. (1998). *Regression analysis of count data*. Cambridge, UK; New York, NY, USA;: Cambridge University Press.

Capitaine, L., Genuer, R., & Thiebaut, R. (2020). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research,* doi:10.1177/0962280220946080

Centers for Disease Control and Prevention. (2018b). Youth Risk Behavior Surveillance System (YRBS). http://www.cdc.gov/HealthyYouth/yrbs/index.htm.

Chawla, N. V. (2009). Data mining for imbalanced datasets: An overview. In *Data mining and knowledge discovery handbook* (pp. 875-886). Springer, Boston, MA.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

Choi, A., Gil, M., Mediavilla, M., & Valbuena, J. (2018). predictors and effects of grade repetition. Revista De Economía Mundial, (48), 21-42.

Clarke, P. (2008). When can group level clustering be ignored? Multilevel models versus single-level models with sparse data. *Journal of Epidemiology & Community Health*, *62*(8), 752-758.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). L. Erlbaum Associates.

Corman, H. (2003). The effects of state policies, individual characteristics, family characteristics, and neighbourhood characteristics on grade repetition in the united states. Economics of Education Review, 22(4), 409-420. https://doi.org/10.1016/S0272-7757(02)00070-5

Crane-Droesch, A. (2017). Semiparametric panel data models using neural networks. *arXiv preprint arXiv:1702.06512*.

Cunningham, S. (2021). *Causal inference: The mixtape*. Yale University Press.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, *40*(2), 139-157.

Downes, M., & Carlin, J. B. (2020). Multilevel regression and poststratification as a modeling approach for estimating population quantities in large population health studies: A simulation study. *Biometrical Journal*, *62*(2), 479-491.

Eisemon, T. O., & United Nations Educational, Scientific, and Cultural Organization, Paris (France). International Inst. for Educational Planning. (1997). Reducing repetition: Issues and strategies. fundamentals of educational planning series, number 55

Eisenstein, J. (2019). *Introduction to natural language processing*. MIT press.

Feldesman, M. R. (2002). Classification trees as an alternative to linear discriminant analysis. *American Journal of Physical Anthropology, 119*(3), 257-275. doi:10.1002/ajpa.10102

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making, 12*(1), 8-8. https://doi.org/10.1186/1472-6947-12-8

Finch, W. H., Bolin, J. H., & Kelley, K. (2014). Group membership prediction when known groups consist of unknown subgroups: a Monte Carlo comparison of methods. *Frontiers in Psychology*, *5*, 337.

Friedman, J. H. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Trans. Computers*, *26*(4), 404-408.

Gebbie, K., Rosenstock, L., & Hernandez, L. M., (Eds.). (2003). *Who will keep the public healthy?: educating public health professionals for the 21st century*. National Academies Press.

Glick, P., & Sahn, D. E. (2010). Early academic performance, grade repetition, and school attainment in senegal: A panel data analysis. The World Bank Economic Review, 24(1), 93-120. https://doi.org/10.1093/wber/lhp023

Gooty, J., & Yammarino, F. J. (2011). Dyads in organizational research: Conceptual issues and multilevel analyses. *Organizational Research Methods, 14*(3), 456-483. doi:10.1177/1094428109358271

Gully, S. M., & Phillips, J. M. (2019). On finding your level. In *The handbook of multilevel theory, measurement, and analysis.* (pp. 11 – 38). American Psychological Association.

Guo, G., & Zhao, H. (2000). Multilevel modeling for binary data. *Annual Review of Sociology, 26*(1), 441-462. doi:10.1146/annurev.soc.26.1.441

Hajjem, A., Bellavance, F., & Larocque, D. (2014). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, *84*(6), 1313-1328.

Hajjem, A., Larocque, D., & Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics & Probability Letters*, *126*, 114-118.

Harrison, R. L. (2010, January). Introduction to monte carlo simulation. In *AIP conference proceedings* (Vol. 1204, No. 1, pp. 17-21). American Institute of Physics.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hoens, T. R., & Chawla, N. V. (2013). Imbalanced datasets: from sampling to classifiers. *Imbalanced learning: Foundations, algorithms, and applications*, 43-59.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression*. New York: Wiley.

Huang, Y., and Li, L., (2011). Naive Bayes classification algorithm based on small sample set. *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, Beijing, China, 2011, pp. 34-39, doi:10.1109/CCIS.2011.6045027.

Ikeda, M., & García, E. (2014). Grade repetition: A comparative study of academic and non-academic consequences. OECD Journal: Economic Studies, 2013(1), 269-315. https://doi.org/10.1787/eco_studies-2013-5k3w65mx3hnx

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: Springer.

Kenny, D. A., Kashy, D. A., & Cook, W. L. (2006). *Dyadic data analysis* (Ser. Methodology in the social sciences). Guilford Press.

Kilham, P., Hartebrodt, C., & Kändler, G. (2019). Generating Tree-Level Harvest Predictions from Forest Inventories with Random Forests. *Forests*, *10*(1), 20.

Kohavi, R., & Wolpert, D. H. (1996, July). Bias plus variance decomposition for zero-one loss functions. In *ICML* (Vol. 96, pp. 275-83).

Kreft, I. G. (1996). Are multilevel techniques necessary? An overview, including simulation studies. *Unpublished manuscript, California State University, Los Angeles*.

Lavery, M. R., Acharya, P., Sivo, S. A., & Xu, L. (2019). Number of predictors and multicollinearity: What are their effects on error and bias in regression?. *Communications in Statistics-Simulation and Computation*, *48*(1), 27-38.

LeBreton, J. M., & Senter, J. L. (2008). Answers to 20 questions about interrater reliability and interrater agreement. *Organizational research methods*, *11*(4), 815-852.

Lee, J., & Stankov, L. (2018). Non-cognitive predictors of academic achievement: Evidence from TIMSS and PISA. Learning and Individual Differences, 65, 50-64. https://doi.org/10.1016/j.lindif.2018.05.009

Lei, P. W., & Koehly, L. M. (2003). Linear discriminant analysis versus logistic regression: A comparison of classification errors in the two-group case. *The Journal of Experimental Education*, *72*(1), 25-49.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, *2*(3), 18-22.

Little, R. J. (2013). In praise of simplicity not mathematistry! ten simple powerful ideas for the statistical scientist. *Journal of the American Statistical Association, 108*(502), 359-369. https://doi.org/10.1080/01621459.2013.787932

Luke, D. A. (2019). *Multilevel modeling* (Vol. 143). SAGE Publications, Incorporated.

Maas, C. J. M., & Hox, J. J. (2005). Sufficient sample sizes for multilevel modeling. *Methodology, 1*(3), 86-92. doi:10.1027/1614-2241.1.3.86

Mangino, A. A., & Finch, W. H. (2021). Prediction With Mixed Effects Models: A Monte Carlo Simulation Study. *Educational and Psychological Measurement*. https://doi.org/10.1177/0013164421992818

Mangino, A.A., Smith, K.A., Finch, W.H., & Hernández-Finch, M. E., (2021). Improving Predictive Classification Models Using Generative Adversarial Networks in the Prediction of Suicide Attempts. *Measurement and Evaluation in Counseling and Development*, doi: 10.1080/07481756.2021.1906156

Mann, J. J., Ellis, S. P., Waternaux, C. M., Liu, X., Oquendo, M. A., Malone, K. M., Brodsky, B. S., Haas, G. L., & Currier, D. (2008). Classification trees distinguish suicide attempters in major psychiatric disorders: A model of clinical decision making. *The Journal of Clinical Psychiatry, 69*(1), 23-31. doi:10.4088/JCP.v69n0104

Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I., & de Mendonça, A. (2011). Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC Research Notes, 4*(1), 299-313. doi:10.1186/1756-0500-4-299

Martella, R. C., Nelson, J. R., Morgan, R. L., & Marchand-Martella, N. E. (2013). *Understanding and interpreting educational research*. Guilford Press.

Martin, J. K, & Hirschberg, D. S. (1995). Small sample statistics for classification error rates I: error rate measurements. *UC Irvine: Donald Bren School of Information and Computer Sciences*. Retrieved from https://escholarship.org/uc/item/76g4v06v

McMahon, S. D., Parnes, A. L., Keys, C. B., & Viola, J. J. (2008). School belonging among low□income urban youth with disabilities: Testing a theoretical model. Psychology in the Schools, 45(5), 387-401. https://doi.org/10.1002/pits.20304

McNeal, R. B. (1995). Extracurricular activities and high school dropouts. *Sociology of Education, 68*(1), 62-80. doi:10.2307/2112764

McNeish, D., & Kelley, K. (2019). Fixed effects models versus mixed effects models for clustered data: Reviewing the approaches, disentangling the differences, and making recommendations. *Psychological Methods, 24*(1), 20-35. doi:10.1037/met0000182

McNeish, D., & Stapleton, L. M. (2016). Modeling clustered data with very few clusters. *Multivariate Behavioral Research, 51*(4), 495-518. doi:10.1080/00273171.2016.1167008

McNeish, D., Stapleton, L. M., & Silverman, R. D. (2017). On the unnecessary ubiquity of hierarchical linear modeling. *Psychological Methods, 22*(1), 114-140. doi:10.1037/met0000078

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, *28*(1), 92-122.

Meshbane, A., & Morris, J. D. (1996). Predictive Discriminant Analysis Versus Logistic Regression in Two-Group Classification Problems. Paper presented at the Annual

Meeting of the American Educational Research Association (New York, NY, April 8 – 12).

Milliren, C. E., Evans, C. R., Richmond, T. K., & Dunn, E. C. (2018). Does an uneven sample size distribution across settings matter in cross-classified multilevel modeling? results of a simulation study. *Health & Place, 52*, 121-126. doi:10.1016/j.healthplace.2018.05.009

Min, F., Wenke, N., & Xiaosong, Z. (2012). Multiple attractor cellular automata classification method and over-fitting problem with CART. *Journal of Computer Research and Development*, *49*(8), 1747-1752.

Moineddin, R., Matheson, F. I., & Glazier, R. H. (2007). A simulation study of sample size for multilevel logistic regression models. *BMC Medical Research Methodology, 7*(1), 34-34. doi:10.1186/1471-2288-7-34

Muchlinski, D., Siroky, D., He, J., & Kocher, M. (2016). Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis, 24*(1), 87-103. doi:10.1093/pan/mpv024

Murtaugh, P. A. (2007). Simplicity and complexity in ecological data analysis. *Ecology (Durham), 88*(1), 56-62. https://doi.org/10.1890/0012-9658(2007)88[56:SACIED]2.0.CO;2

Ngufor, C. (2019). Vira: Virtual Inteligent Robot Assistant. R package version 0.1.

Paccagnella, O. (2011). Sample size and accuracy of estimates in multilevel models: New simulation results. *Methodology, 7*(3), 111-120. doi:10.1027/1614-2241/a000029

Palvanov, A., & Cho, Y. I., (2018). Comparisons of deep learning algorithms for MNIST in real-time environment. *International Journal of Fuzzy Logic and Intelligent Systems*, *18*(2), 126-134.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *International Journal of Machine Learning Technology, 2*(1), 37-63.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (Vol. 1). Sage.

Raudys, S. J., & Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(3), 252-264. https://doi.org/10.1109/34.75512

Revelle, W. R. (2017). psych: Procedures for personality and psychological research.

Sanchez, S. M. (2005, December). Work smarter, not harder: guidelines for designing simulation experiments. In *Proceedings of the Winter Simulation Conference, 2005.* (pp. 69-82). IEEE.

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: a data mining approach for longitudinal and clustered data. *Machine learning*, *86*(2), 169-207.

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O., & Hemingway, H. (2014). Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. *American journal of epidemiology*, *179*(6), 764-774.

Snijders, T. A., & Bosker, R. J. (2011). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. Sage.

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2019). BiMM forest: A random forest method for modeling clustered and longitudinal binary outcomes. *Chemometrics and Intelligent Laboratory Systems*, *185*, 122-134.

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2020). BiMM tree: a decision tree method for modeling clustered and longitudinal binary outcomes. *Communications in Statistics-Simulation and Computation*, *49*(4), 1004-1023.

Steyerberg, E. W. (2019). *Clinical prediction models*. Springer International Publishing.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological methods*, *14*(4), 323-348.

Tabachnick, B., & Fidell, L. (2018). *Using multivariate statistics*. Upper Saddle River: Pearson Education.

VanDerHeyden, A. M. (2013). Universal screening may not be for everyone: Using a threshold model as a smarter way to determine risk. School Psychology Review, 42(4), 402-414. https://www.doi.org/10.1080/02796015.2013.12087462

Waljee, A. K., Mukherjee, A., Singal, A. G., Zhang, Y., Warren, J., Balis, U., Marrero, J., Zhu, J., & Higgins, P. D. (2013). Comparison of imputation methods for missing laboratory data in medicine. *BMJ Open, 3*(8), e002847. doi:10.1136/bmjopen-2013-002847

Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: machine learning and classification methods as alternatives to logistic regression. *Journal of clinical epidemiology*, *63*(8), 826-833.

Wößmann, L. (2003). Schooling resources, educational institutions and student performance: The international evidence. Oxford Bulletin of Economics and Statistics, 65(2), 117-170. https://doi.org/10.1111/1468-0084.00045

Wu, M., & Zhang, Z. (2010). Handwritten digit classification using the mnist data set. *Course project CSE802: Pattern Classification & Analysis*.

Yan, P. (2019). Anomaly Detection in Categorical Data with Interpretable Machine Learning: A random forest approach to classify imbalanced data. Thesis.

Zellner, A., Keuzenkamp, H. A., & McAleer, M. (2001). *Simplicity, inference and modelling: Keeping it sophisticatedly simple*. Cambridge University Press.

Zhang, J. L., & Haerdle, W. K. (2010). The bayesian additive classification tree applied to credit risk modelling. *Computational Statistics & Data Analysis*, *54*(5), 1197-1205. doi:10.1016/j.csda.2009.11.022

Zhou, L., Song, Y., Alterman, V., Liu, Y., & Wang, M. (2019). Introduction to data collection in multilevel research. In *The handbook of multilevel theory, measurement, and analysis.* (pp. 225-252). American Psychological Association.

Zigler, E., & Phillips, L. (1961). Psychiatric diagnosis: A critique. *Journal of Abnormal and Social Psychology, 63*(3), 607-618. doi:10.1037/h0040556

Appendix

```
#### AAM Fixed v Mixed Simulation, All L1 Predictors ####
{
  rm(list=ls())
  library(psych)
  library(caret)
  library(lme4)
  #library(jrfit)
  library(quantreg)
  #library(lmmen)
  library(reshape2) # Can also use multilevel::make.univ to reshape wide to long
  library(tidyr)
  library(dplyr)
  #library(tidyverse)
  library(GLMMadaptive)
  library(MLmetrics) # For AUC function: AUC(predicted_probs, actual_groups)
  library(MuMIn) # For calculating pseudo R-squared; r.squaredLR/r.squaredGLMM
  #library(Vira)
  library(ranger)
  library(e1071)
```

```
library(PresenceAbsence)
library(readr)

parm_frame = read_csv("~/diss_sim/parameter_file.csv")

set.seed(8675309)
# Creating 0 - 1 Range normalizing function
# range01 = function(x){(x - min(x)) / (max(x) - min(x))}

# Create correlated variable
correlatedValue = function(x, r){
  r2 = r**2
  ve = 1-r2
  SD = sqrt(ve)
  e  = rnorm(length(x), mean=0, sd=SD)
  y  = r*x + e
  return(y)
}

# Create Train.Test function
```

```
Train.Test = function (...)
{
  res = list(GLMER = function(trn, tst, para, resp.vars, rand.vars,
                              rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                              ...) {
    result <- list()
    form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(c(paste0(rhs.vars,
                                                                       collapse = "+"), "+", "(",
    paste0(c(rand.vars), collapse = "+"),
                                                                  "|", groups, ")"), collapse = "")))
    result$model <- glmer(form, data = trn, family = binomial,
                          control = glmerControl(optimizer = "bobyqa"), nAGQ = 0)
    result$pred <- predict(result$model, newdata = tst, type = "response")
    result$perf <- Performance.measures(result$pred, tst[,
                                                         resp.vars])
    result
  }, MEgbm = function(trn, tst, para, resp.vars, rand.vars,
                      rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                      ...) {
    result <- list()
```

```
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  result$model <- MEgbmRules(form = form, dat = trn, groups = groups,
                             rand.vars = rand.vars, para = para, tol = 1e-05,
                             max.iter = para$max.iter, include.RE = para$include.RE,
                             verbose = FALSE, maxdepth = para$maxdepth, glmer.Control =
  glmerControl(optimizer = "bobyqa"),
                             nAGQ = 0, K = para$K, para$decay)
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred$pred[,
                                                       2], tst[, resp.vars], threshold =
  result$model$threshold)
  result
}, MErf = function(trn, tst, para, resp.vars, rand.vars,
                   rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                   ...) {
  result <- list()
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  result$model <- MErfRules(form = form, dat = trn, groups = groups,
```

```
                          rand.vars = rand.vars, para = para, tol = para$tol,
                          max.iter = para$max.iter, include.RE = para$include.RE,
                          verbose = FALSE, maxdepth = para$maxdepth, glmer.Control =
  glmerControl(optimizer = "bobyqa"),
                          nAGQ = 0, K = para$K, decay = para$decay)
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred$pred[,
                                                       2], tst[, resp.vars], threshold =
  result$model$threshold)
  result
}, MEmixgbm = function(trn, tst, para, resp.vars, rand.vars,
                       rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                       ...) {
  result <- list()
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  result$model <- MEmixgbm(form = form, dat = trn, groups = groups,
                           rand.vars = rand.vars, para = para, tol = 1e-05,
                           max.iter = para$max.iter, include.RE = para$include.RE,
```

```
                          verbose = FALSE, maxdepth = para$maxdepth, glmer.Control =
  glmerControl(optimizer = "bobyqa"),
                          nAGQ = 0, K = para$K, krang = para$krange, para$decay)
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred$pred[,
                                                       2], tst[, resp.vars], threshold =
  result$model$threshold)
  result
}, MEmixgbm2 = function(trn, tst, para, resp.vars, rand.vars,
                        rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                        ...) {
  result <- list()
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  result$model <- MEmixgbm2(form = form, dat = trn, groups = groups,
                            rand.vars = rand.vars, para = para, tol = 1e-05,
                            max.iter = para$max.iter, include.RE = para$include.RE,
                            verbose = FALSE, maxdepth = para$maxdepth, glmer.Control =
  glmerControl(optimizer = "bobyqa"),
                            nAGQ = 0, K = para$K, krang = para$krange, para$decay)
```

```
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred$pred[,
                                                       2], tst[, resp.vars], threshold =
  result$model$threshold)
  result
}, MEglmTree = function(trn, tst, para, resp.vars, rand.vars,
                        rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                        ...) {
  result <- list()
  X.trn <- trn
  X <- X.trn[, unique(c(rhs.vars, reg.vars, part.vars,
                        groups)), drop = FALSE]
  Y <- X.trn[, resp.vars]
  result$model <- MEglmTree(X = X, Y = Y, part.vars = part.vars,
                            reg.vars = reg.vars, rand.vars = rand.vars, groups = groups,
                            include.RE = para$include.RE, max.iter = para$max.iter,
                            alpha = para$alpha, minsize = para$minsize, maxdepth = para$maxdepth,
                            para = para)
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred[, 2],
```

```
                                    tst[, resp.vars], threshold = result$model$threshold)
  result
}, MECTree = function(trn, tst, para, resp.vars, rand.vars,
                      rhs.vars, reg.vars = NULL, part.vars = NULL, groups,
                      ...) {
  result <- list()
  X.trn <- trn
  X <- X.trn[, unique(c(rhs.vars, reg.vars, part.vars,
                        groups)), drop = FALSE]
  Y <- X.trn[, resp.vars]
  result$model <- MECTree(X = X, Y = Y, con.tree = para$con.tree,
                          rhs.vars = rhs.vars, rand.vars = rand.vars, groups = groups,
                          max.iter = para$max.iter)
  result$pred <- predict(result$model, newdata = tst)
  result$perf <- Performance.measures(result$pred[, 2],
                                    tst[, resp.vars], threshold = result$model$threshold)
  result
}, GLM = function(trn, tst, para, resp.vars, rand.vars, rhs.vars,
                  reg.vars = NULL, part.vars = NULL, groups, ...) {
  result <- list()
```

```
  xx <- trn
  xx[, resp.vars] <- factor(ifelse(xx[, resp.vars] == 1,
                                   "Yes", "No"))
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  result$model <- train(form, data = xx, method = "glm",
                        family = "binomial", control = list(maxit = 200))
  result$pred = predict(result$model, newdata = tst, type = "prob")
  result$perf <- Performance.measures(result$pred[, 2],
                                      tst[, resp.vars])
  result
}, GBM = function(trn, tst, para, resp.vars, rand.vars, rhs.vars,
                  reg.vars = NULL, part.vars = NULL, groups, ...) {
  result <- list()
  xx <- trn
  xx[, resp.vars] <- factor(ifelse(xx[, resp.vars] == 1,
                                   "Yes", "No"))
  form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                            collapse = "+")))
  if (para$opt.para) {
```

```
    fitControl <- trainControl(method = para$method,
                               number = para$number, classProbs = TRUE, summaryFunction =
  twoClassSummary)
    result$model <- train(form, data = xx, method = "gbm",
                          trControl = fitControl, verbose = FALSE, metric = "ROC",
                          tuneLength = para$tuneLength)
  } else {
    result$model <- train(form, data = xx, method = "gbm",
                          trControl = trainControl(method = "none"), verbose = FALSE,
                          tuneGrid = data.frame(n.trees = para$n.trees,
                                                n.minobsinnode = para$n.minobsinnode,
  interaction.depth = para$interaction.depth,
                                                shrinkage = para$shrinkage))
  }
  result$pred = predict(result$model, newdata = tst, type = "prob")
  result$perf <- Performance.measures(result$pred[, 2],
                                      tst[, resp.vars])
  result
}, RF = function(trn, tst, para, resp.vars, rand.vars, rhs.vars,
                 reg.vars = NULL, part.vars = NULL, groups, ...) {
```

```
result <- list()
xx <- trn
xx[, resp.vars] <- factor(ifelse(xx[, resp.vars] == 1,
                                 "Yes", "No"))
form <- as.formula(paste0(paste0(resp.vars, " ~"), paste0(rhs.vars,
                                                          collapse = "+")))
if (para$opt.para) {
  fitControl <- trainControl(method = para$method,
                             number = para$number, classProbs = TRUE, summaryFunction =
twoClassSummary)
  result$model <- train(form, data = xx, method = "rf",
                        trControl = fitControl, verbose = FALSE, metric = "ROC",
                        tuneLength = para$tuneLength)
} else {
  result$model <- train(form, data = xx, method = "rf",
                        trControl = trainControl(method = "none"), verbose = FALSE,
                        tuverbose = FALSE, tuneGrid = data.frame(mtry =
floor(length(rhs.vars)/3)),
                        ntree = para$ntree)
}
```

```
    result$pred <- predict(result$model, newdata = tst, type = "prob")
    result$perf <- Performance.measures(result$pred[, 2],
                                        tst[, resp.vars])
    result
  })
  return(res)
}

# Performance.measures function
Performance.measures = function (pred, obs, threshold = NULL, prevalence = NULL)
{
  if (length(unique(obs)) == 2) {
    obs <- as.numeric(factor(obs)) - 1
    if (is.null(threshold))
      threshold <- opt.thresh(pred, obs)
    nme = c("PCC", "PCC.sd", "AUC", "AUC.sd", "sensitivity",
            "sensitivity.sd", "specificity", "specificity.sd")
    xx = cbind.data.frame(plotID = 1:length(pred), Observed = obs,
                          Predicted = pred)
    accuracy <- presence.absence.accuracy(xx, threshold = threshold,
```

```
                                        st.dev = TRUE)[, nme]
accuracy$G.mean <- sqrt(as.numeric(accuracy$sensitivity) *
                          as.numeric(accuracy$specificity))
accuracy$BER <- 1 - 0.5 * (as.numeric(accuracy$sensitivity) +
                             as.numeric(accuracy$specificity))
pred.prev <- predicted.prevalence(DATA = xx, threshold = threshold)[,
                                                                    c("Obs.Prevalence",
"Predicted")]
nme <- c("Pos Pred Value", "Neg Pred Value", "Balanced Accuracy")
if (is.null(prevalence))
  prevalence <- as.numeric(pred.prev$Obs.Prevalence)
obs <- factor(ifelse(obs == 1, "Yes", "No"), levels = c("Yes",
                                                        "No"))
pred <- factor(ifelse(pred >= threshold, "Yes", "No"),
               levels = c("Yes", "No"))
cmx <- confusionMatrix(data = pred, reference = obs,
                       prevalence = prevalence)$byClass[nme]
accuracy$PPV <- cmx[1]
accuracy$NPV = cmx[2]
accuracy$BACC <- cmx[3]
```

```
    accuracy$F.measure = 2 * (accuracy$PPV * accuracy$sensitivity)/(accuracy$PPV +
                                                                    accuracy$sensitivity)
    accuracy$threshold = threshold
  }
  else accuracy = data.frame(t(regr.eval(obs, pred, stats = c("mae",
                                                              "mse", "rmse"))))
  return(accuracy)
}

# opt.thresh function
opt.thresh = function (pred, obs)
{
  thresh = 0.5
  if (length(unique(obs)) > 1) {
    obs <- as.numeric(as.factor(obs)) - 1
    SIMDATA = cbind.data.frame(plotID = 1:length(obs), Observed = obs,
                               Predicted = pred)
    thresh <- optimal.thresholds(SIMDATA, threshold = 101,
                                 which.model = 1, opt.methods = 9)
    thresh <- ifelse(length(thresh["Predicted"]) >= 1, as.numeric(thresh["Predicted"]),
```

```
                    0.5)
  }
  return(thresh)
}

# optimal.thresholds function
optimal.thresholds = function (DATA = NULL, threshold = 101, which.model = 1:(ncol(DATA) -
                                                                              2), model.names =
    NULL, na.rm = FALSE, opt.methods = NULL,
                               req.sens, req.spec, obs.prev = NULL, smoothing = 1, FPC,
                               FNC)
{
  POSSIBLE.meth <- c("Default", "Sens=Spec", "MaxSens+Spec",
                     "MaxKappa", "MaxPCC", "PredPrev=Obs", "ObsPrev", "MeanProb",
                     "MinROCdist", "ReqSens", "ReqSpec", "Cost")
  if (is.null(DATA) == TRUE) {
    return(POSSIBLE.meth)
  }
  else {
    if (is.null(opt.methods) == TRUE) {
```

```
  opt.methods <- POSSIBLE.meth
}
N.meth <- length(opt.methods)
if (is.numeric(opt.methods) == TRUE) {
  if (sum(opt.methods %in% (1:length(POSSIBLE.meth))) !=
      N.meth) {
    stop("invalid optimization method")
  }
  else {
    opt.methods <- POSSIBLE.meth[opt.methods]
  }
}
if (sum(opt.methods %in% POSSIBLE.meth) != N.meth) {
  stop("invalid optimization method")
}
if ("ReqSens" %in% opt.methods) {
  if (missing(req.sens)) {
    warning("req.sens defaults to 0.85")
    req.sens <- 0.85
  }
```

```
}
if ("ReqSpec" %in% opt.methods) {
  if (missing(req.spec)) {
    warning("req.spec defaults to 0.85")
    req.spec <- 0.85
  }
}
if ("Cost" %in% opt.methods) {
  if (missing(FPC) || missing(FNC)) {
    warning("costs assumed to be equal")
    FPC <- 1
    FNC <- 1
  }
  if (FPC <= 0 || FNC <= 0) {
    stop("costs must be positive")
  }
}
if (length(smoothing) != 1) {
  stop("'smoothing' must be a single number greater than or equal to 1")
}
```

```
else {
  if (is.numeric(smoothing) == FALSE) {
    stop("'smoothing' must be a single number greater than or equal to 1")
  }
  else {
    if (smoothing < 1) {
      stop("'smoothing' must be a single number greater than or equal to 1")
    }
  }
}
if (sum(is.na(DATA)) > 0) {
  if (na.rm == TRUE) {
    NA.rows <- apply(is.na(DATA), 1, sum)
    warning(length(NA.rows[NA.rows > 0]), " rows ignored due to NA values")
    DATA <- DATA[NA.rows == 0, ]
  }
  else {
    return(NA)
  }
}
```

```
DATA[DATA[, 2] > 0, 2] <- 1
N.models <- ncol(DATA) - 2
if (is.null(obs.prev) == TRUE) {
  obs.prev <- sum(DATA[, 2])/nrow(DATA)
}
if (obs.prev < 0 || obs.prev > 1) {
  stop("'obs.prev' must be a number between zero and one")
}
if (obs.prev == 0) {
  warning("because your observed prevalence was zero, results may be strange")
}
if (obs.prev == 1) {
  warning("because your observed prevalence was one, results may be strange")
}
OBS <- DATA[, 2]
if (length(OBS[OBS == 0]) == 0) {
  stop("no observed absences in dataset, therefore specificity does not",
       "exist, and modeling, much less threshold optimization, is not very",
       "meaningful")
}
```

```
if (length(OBS[OBS == 1]) == 0) {
  stop("no observed presences in dataset, therefore sensitivity does not",
       "exist, and modeling, much less threshold optimization, is not very",
       "meaningful")
}
if (min(which.model) < 1 || sum(round(which.model) !=
                                  which.model) != 0) {
  stop("values in 'which.model' must be positive integers!")
}
if (max(which.model) > N.models) {
  stop("values in 'which.model' must not be greater than number of models in 'DATA'")
}
if (is.null(model.names) == TRUE) {
  model.names <- if (is.null(names(DATA)) == FALSE) {
    names(DATA)[-c(1, 2)]
  }
  else {
    paste("Model", 1:N.models)
  }
}
```

```
if (N.models != length(model.names) && (length(which.model) !=
                                          1 || length(model.names) != 1)) {
  stop("If 'model.names' is specified it must either be a single name, or a vector",
       "of the same length as the number of model predictions in 'DATA'")
}
DATA <- DATA[, c(1, 2, which.model + 2)]
if (length(model.names) != 1) {
  model.names <- model.names[which.model]
}
N.dat <- ncol(DATA) - 2
N.thr <- length(threshold)
if (min(threshold) < 0) {
  stop("'threshold' can not be negative")
}
if (max(threshold) > 1) {
  if (N.thr == 1 && round(threshold) == threshold) {
    threshold <- seq(length = threshold, from = 0,
                     to = 1)
    N.thr <- length(threshold)
  }
```

```
  else {
    stop("non-interger 'threshold' greater than 1")
  }
}
OPT.THRESH <- data.frame(matrix(0, N.meth, N.dat))
names(OPT.THRESH) <- model.names
for (dat in 1:N.dat) {
  ACC <- presence.absence.accuracy(DATA, which.model = dat,
                                   threshold = threshold, find.auc = FALSE, st.dev = FALSE)
  for (meth in 1:N.meth) {
    if (opt.methods[meth] == "Default") {
      OPT.THRESH[meth, dat] <- 0.5
    }
    if (opt.methods[meth] == "Sens=Spec") {
      SENS.SPEC <- abs(ACC$sensitivity - ACC$specificity)
      OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(SENS.SPEC,
                                                      ties.method = "min") <= smoothing])
    }
    if (opt.methods[meth] == "MaxSens+Spec") {
      SENS.SPEC <- ACC$sensitivity + ACC$specificity
```

```
  OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(-SENS.SPEC,
                                                  ties.method = "min") <= smoothing])
}
if (opt.methods[meth] == "MaxKappa") {
  OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(-ACC$Kappa,
                                                  ties.method = "min") <= smoothing])
}
if (opt.methods[meth] == "MaxPCC") {
  OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(-ACC$PCC,
                                                  ties.method = "min") <= smoothing])
}
if (opt.methods[meth] == "PredPrev=Obs") {
  PREV <- predicted.prevalence(DATA, which.model = dat,
                               threshold = threshold)
  PREV.diff <- abs(obs.prev - PREV[, 3])
  OPT.THRESH[meth, dat] <- mean(PREV$threshold[rank(PREV.diff,
                                                   ties.method = "min") <= smoothing])
}
if (opt.methods[meth] == "ObsPrev") {
  OPT.THRESH[meth, dat] <- obs.prev
```

```
}
if (opt.methods[meth] == "MeanProb") {
  OPT.THRESH[meth, dat] <- mean(DATA[, 2 + dat])
}
if (opt.methods[meth] == "MinROCdist") {
  ROC <- (ACC$specificity - 1)^2 + (1 - ACC$sensitivity)^2
  OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(ROC,
                                                  ties.method = "min") <= smoothing])
}
if (opt.methods[meth] == "ReqSens") {
  OPT.THRESH[meth, dat] <- max(ACC$threshold[(ACC$sensitivity) >=
                                               req.sens])
}
if (opt.methods[meth] == "ReqSpec") {
  OPT.THRESH[meth, dat] <- min(ACC$threshold[(ACC$specificity) >=
                                               req.spec])
}
if (opt.methods[meth] == "Cost") {
  if (obs.prev == 0) {
    obs.prev <- 1e-06
```

```
          }
          sl <- (FPC/FNC) * (1 - obs.prev)/obs.prev
          x <- (1 - ACC$specificity)
          y <- ACC$sensitivity
          rad <- (x^2 + y^2)^0.5
          theta <- atan2(y, x)
          theta.sl <- atan(sl)
          theta.new <- theta - theta.sl
          x.new <- rad * cos(theta.new)
          y.new <- rad * sin(theta.new)
          OPT.THRESH[meth, dat] <- mean(ACC$threshold[rank(-y.new,
                                                          ties.method = "min") <= smoothing])
        }
      }
    }
    OPT.THRESH <- cbind(Method = opt.methods, OPT.THRESH)
    return(OPT.THRESH)
  }
}
```

```
# presence.absence.accuracy function
presence.absence.accuracy = function (DATA, threshold = 0.5, find.auc = TRUE, st.dev = TRUE,
                                      which.model = (1:(ncol(DATA) - 2)), na.rm = FALSE)
{
  if (is.logical(find.auc) == FALSE) {
    stop("'find.auc' must be of logical type")
  }
  if (is.logical(st.dev) == FALSE) {
    stop("'st.dev' must be of logical type")
  }
  if (is.logical(na.rm) == FALSE) {
    stop("'na.rm' must be of logical type")
  }
  if (sum(is.na(DATA)) > 0) {
    if (na.rm == TRUE) {
      NA.rows <- apply(is.na(DATA), 1, sum)
      warning(length(NA.rows[NA.rows > 0]), " rows ignored due to NA values")
      DATA <- DATA[NA.rows == 0, ]
    }
    else {
```

```
    return(NA)
  }
}
DATA[DATA[, 2] > 0, 2] <- 1
if (min(which.model) < 1 || sum(round(which.model) != which.model) !=
    0) {
  stop("values in 'which.model' must be positive integers")
}
if (max(which.model) + 2 > ncol(DATA)) {
  stop("values in 'which.model' must not be greater than number of models in 'DATA'")
}
DATA <- DATA[, c(1, 2, which.model + 2)]
N.models <- ncol(DATA) - 2
model.names <- if (is.null(names(DATA))) {
  paste("Model", 1:N.models)
}
else {
  names(DATA)[-c(1, 2)]
}
N.thr <- length(threshold)
```

```
N.dat <- ncol(DATA) - 2
REP.dat <- FALSE
if (min(threshold) < 0) {
  stop("'threshold' can not be negative")
}
if (max(threshold) > 1) {
  if (N.thr == 1 && round(threshold) == threshold && (N.dat ==
                                                      1 || N.dat == threshold)) {
    threshold <- seq(length = threshold, from = 0, to = 1)
    N.thr <- length(threshold)
  }
  else {
    stop("either length of 'threshold' doesn't equal number of",
         "models, or 'threshold is a non-integer greater than 1")
  }
}
if (N.thr == 1 && N.dat > 1) {
  threshold <- rep(threshold, N.dat)
  N.thr <- length(threshold)
}
```

```
if (N.thr != N.dat) {
  if (N.dat == 1) {
    REP.dat <- TRUE
  }
  else {
    stop("length of 'threshold' doesn't equal number of models!")
  }
}
if (REP.dat == FALSE) {
  if (st.dev == FALSE) {
    ERROR <- data.frame(matrix(0, N.dat, 7))
    names(ERROR) <- c("model", "threshold", "PCC", "sensitivity",
                      "specificity", "Kappa", "AUC")
    ERROR[, 1] <- model.names
    ERROR[, 2] <- threshold
    for (dat in 1:N.dat) {
      CMX <- cmx(DATA = DATA, threshold = threshold[dat],
                 which.model = dat)
      ERROR[dat, 3] <- pcc(CMX = CMX, st.dev = FALSE)
      ERROR[dat, 4] <- sensitivity(CMX = CMX, st.dev = FALSE)
```

```
    ERROR[dat, 5] <- specificity(CMX = CMX, st.dev = FALSE)
    ERROR[dat, 6] <- Kappa(CMX = CMX, st.dev = FALSE)
    if (find.auc == TRUE) {
      ERROR[dat, 7] <- auc(DATA = DATA, st.dev = FALSE,
                           which.model = dat)
    }
  }
}
else {
  ERROR <- data.frame(matrix(0, N.dat, 12))
  names(ERROR) <- c("model", "threshold", "PCC", "sensitivity",
                    "specificity", "Kappa", "AUC", "PCC.sd", "sensitivity.sd",
                    "specificity.sd", "Kappa.sd", "AUC.sd")
  ERROR[, 1] <- model.names
  ERROR[, 2] <- threshold
  for (dat in 1:N.dat) {
    CMX <- cmx(DATA = DATA, threshold = threshold[dat],
               which.model = dat)
    ERROR[dat, c(3, 8)] <- pcc(CMX)
    ERROR[dat, c(4, 9)] <- sensitivity(CMX)
```

```
      ERROR[dat, c(5, 10)] <- specificity(CMX)
      ERROR[dat, c(6, 11)] <- Kappa(CMX)
      if (find.auc == TRUE) {
        ERROR[dat, c(7, 12)] <- auc(DATA = DATA, which.model = dat)
      }
    }
  }
}
else {
  if (st.dev == FALSE) {
    ERROR <- data.frame(matrix(0, N.thr, 7))
    names(ERROR) <- c("model", "threshold", "PCC", "sensitivity",
                      "specificity", "Kappa", "AUC")
    ERROR[, 1] <- model.names
    ERROR[, 2] <- threshold
    for (thresh in 1:N.thr) {
      CMX <- cmx(DATA = DATA, threshold = threshold[thresh])
      ERROR[thresh, 3] <- pcc(CMX = CMX, st.dev = FALSE)
      ERROR[thresh, 4] <- sensitivity(CMX = CMX, st.dev = FALSE)
      ERROR[thresh, 5] <- specificity(CMX = CMX, st.dev = FALSE)
```

```
    ERROR[thresh, 6] <- Kappa(CMX = CMX, st.dev = FALSE)
  }
  if (find.auc == TRUE) {
    ERROR[, 7] <- auc(DATA = DATA, st.dev = FALSE)
  }
}
else {
  ERROR <- data.frame(matrix(0, N.thr, 12))
  names(ERROR) <- c("model", "threshold", "PCC", "sensitivity",
                    "specificity", "Kappa", "AUC", "PCC.sd", "sensitivity.sd",
                    "specificity.sd", "Kappa.sd", "AUC.sd")
  ERROR[, 1] <- model.names
  ERROR[, 2] <- threshold
  for (thresh in 1:N.thr) {
    CMX <- cmx(DATA = DATA, threshold = threshold[thresh])
    ERROR[thresh, c(3, 8)] <- pcc(CMX)
    ERROR[thresh, c(4, 9)] <- sensitivity(CMX)
    ERROR[thresh, c(5, 10)] <- specificity(CMX)
    ERROR[thresh, c(6, 11)] <- Kappa(CMX)
  }
```

```
      if (find.auc == TRUE) {
        area <- auc(DATA)
        ERROR[, 7] <- area$AUC
        ERROR[, 12] <- area$AUC.sd
      }
    }
  }
  if (find.auc == TRUE) {
    return(ERROR)
  }
  else {
    if (st.dev == FALSE) {
      return(ERROR[, 1:6])
    }
    else {
      return(ERROR[, c(1:6, 8:11)])
    }
  }
}
```

```
# cmx function
cmx = function (DATA, threshold = 0.5, which.model = 1, na.rm = FALSE)
{
  if (is.logical(na.rm) == FALSE) {
    stop("'na.rm' must be of logical type")
  }
  if (sum(is.na(DATA)) > 0) {
    if (na.rm == TRUE) {
      NA.rows <- apply(is.na(DATA), 1, sum)
      warning(length(NA.rows[NA.rows > 0]), " rows ignored due to NA values")
      DATA <- DATA[NA.rows == 0, ]
    }
    else {
      return(NA)
    }
  }
  if (length(which.model) != 1) {
    stop("this function will only work for a single model, 'which.model' must be of length one")
  }
  if (which.model < 1 || round(which.model) != which.model) {
```

```
  stop("'which.model' must be a positive integer")
}
if (which.model + 2 > ncol(DATA)) {
  stop("'which.model' must not be greater than number of models in 'DATA'")
}
if (length(threshold) != 1) {
  stop("'threshold' must be a single number between zero and one")
}
if (max(threshold) > 1) {
  stop("'threshold' must be a single number between zero and one")
}
if (min(threshold) < 0) {
  stop("'threshold' must be a single number between zero and one")
}
OBS.ind <- DATA[, 2] > 0
if (threshold == 0) {
  PRED.ind = DATA[, which.model + 2] >= threshold
}
else {
  PRED.ind = DATA[, which.model + 2] > threshold
```

```
  }
  C = c(sum(PRED.ind & OBS.ind), sum(!PRED.ind & OBS.ind),
        sum(PRED.ind & !OBS.ind), sum(!PRED.ind & !OBS.ind))
  C = as.table(matrix(C, nrow = 2))
  dimnames(C) = list(predicted = c(1, 0), observed = c(1, 0))
  storage.mode(C) = "double"
  return(C)
}

# pcc function
pcc = function (CMX, st.dev = TRUE)
{
  if (nrow(CMX) != ncol(CMX) || is.matrix(CMX) == FALSE) {
    stop("'CMX' must be a confusion matrix")
  }
  if (is.logical(st.dev) == FALSE) {
    stop("'st.dev' must be of logical type")
  }
  if (sum(is.na(CMX)) != 0) {
    return(NA)
```

```
  }
  PCC <- sum(diag(CMX))/sum(CMX)
  if (st.dev == FALSE) {
    return(PCC = PCC)
  }
  else {
    PCC.sd <- ((PCC * (1 - PCC))/(sum(CMX) - 1))^0.5
    return(data.frame(PCC = PCC, PCC.sd = PCC.sd))
  }
}

# Create MEml function
MEml = function (classifier, dat.trn, dat.tst, id, rhs.vars, resp.vars,
          rand.vars = NULL, reg.vars = NULL, part.vars = NULL, para,
          max.iter = 10, seed = 1, return.model = FALSE, ...)
{
  Trn.Tst <- lapply(Train.Test(), function(x) x)
  mod.res <- tryCatch({
    lapply(classifier, function(xx, ...) {
      trn <- dat.trn[, unique(c(resp.vars, rhs.vars, reg.vars,
```

```
                              part.vars, id)), drop = FALSE]
    tst <- dat.tst[, unique(c(resp.vars, rhs.vars, reg.vars,
                              part.vars, id)), drop = FALSE]
    if (return.model)
      Trn.Tst[[xx]](trn = trn, tst = tst, para = para,
                    resp.vars = resp.vars, rand.vars = rand.vars,
                    rhs.vars = rhs.vars, reg.vars = reg.vars, part.vars = part.vars,
                    groups = id, ...)
    else cbind(Classifier = xx, Trn.Tst[[xx]](trn = trn,
                                              tst = tst, para = para, resp.vars = resp.vars,
                                              rand.vars = rand.vars, rhs.vars = rhs.vars, reg.vars
  = reg.vars,
                                              part.vars = part.vars, groups = id, ...)$perf)
  })
}, error = function(e) {
  cat("Error in the Expression: ", paste(e$call, collapse = ", "),
      ": original error message = ", e$message, "\n")
  list()
})
gc()
```

```
    names(mod.res) <- classifier
    return(mod.res)
  }

    #pb = tkProgressBar("Simulation progress bar",
                       #"Percent completed",
                       #1, g, 0)
  output_frame = list()
}

fixed_mixed_sim = function(ncases, ngroups, rwg, rbg, gsr, simcount, g, icc){

  require(psych)
  require(caret)
  require(lme4)
  require(jrfit)
  require(quantreg)
  require(lmmen)
  require(MCMCglmm)
  require(reshape2) # Can also use multilevel::make.univ to reshape wide to long
```

```
require(tidyr)
require(dplyr)
require(GLMMadaptive)
require(MLmetrics) # For AUC function: AUC(predicted_probs, actual_groups)
#require(inTrees)
#require(Vira)
require(ranger)
require(e1071)

#### Introductory Block of Output Object Creation ####
{
   # Creating empty result vectors
  lr_acc_train = rep(NA, g)
  lr_sens_train = rep(NA, g)
  lr_spec_train = rep(NA, g)
  lr_ce_train = rep(NA, g)
  lr_auc_train = rep(NA, g)
  lr_acc_test = rep(NA, g)
  lr_sens_test = rep(NA, g)
  lr_spec_test = rep(NA, g)
```

```
lr_ce_test = rep(NA, g)
lr_auc_test = rep(NA, g)

rf_acc_train = rep(NA, g)
rf_sens_train = rep(NA, g)
rf_spec_train = rep(NA, g)
rf_ce_train = rep(NA, g)
rf_auc_train = rep(NA, g)
rf_acc_test = rep(NA, g)
rf_sens_test = rep(NA, g)
rf_spec_test = rep(NA, g)
rf_ce_test = rep(NA, g)
rf_auc_test = rep(NA, g)

glmm_acc_train = rep(NA, g)
glmm_sens_train = rep(NA, g)
glmm_spec_train = rep(NA, g)
glmm_ce_train = rep(NA, g)
glmm_auc_train = rep(NA, g)
glmm_acc_test = rep(NA, g)
```

```
glmm_sens_test = rep(NA, g)
glmm_spec_test = rep(NA, g)
glmm_ce_test = rep(NA, g)
glmm_auc_test = rep(NA, g)

merf_acc_train = rep(NA, g)
merf_sens_train = rep(NA, g)
merf_spec_train = rep(NA, g)
merf_ce_train = rep(NA, g)
merf_auc_train = rep(NA, g)
merf_acc_test = rep(NA, g)
merf_sens_test = rep(NA, g)
merf_spec_test = rep(NA, g)
merf_ce_test = rep(NA, g)
merf_auc_test = rep(NA, g)

icc_calc = rep(NA, g)
l2var_calc = rep(NA, g)

outcome_cor = rep(NA, g)
```

```
    pred_cor = rep(NA, g)
}
  #### Begin Loop ####
  for(i in 1:g){
    tryCatch({

      lr_acc_train[i] = lr_acc_train
      lr_sens_train[i] = lr_sens_train
      lr_spec_train[i] = lr_spec_train
      lr_ce_train[i] = lr_ce_train
      lr_auc_train[i] = lr_auc_train
      lr_acc_test[i] = lr_acc_test
      lr_sens_test[i] = lr_sens_test
      lr_spec_test[i] = lr_spec_test
      lr_ce_test[i] = lr_ce_test
      lr_auc_test[i] = lr_auc_test

      rf_acc_train[i] = rf_acc_train
      rf_sens_train[i] = rf_sens_train
      rf_spec_train[i] = rf_spec_train
```

```
rf_ce_train[i] = rf_ce_train

rf_auc_train[i] = rf_auc_train

rf_acc_test[i] = rf_acc_test

rf_sens_test[i] = rf_sens_test

rf_spec_test[i] = rf_spec_test

rf_ce_test[i] = rf_ce_test

rf_auc_test[i] = rf_auc_test


glmm_acc_train[i] = glmm_acc_train

glmm_sens_train[i] = glmm_sens_train

glmm_spec_train[i] = glmm_spec_train

glmm_ce_train[i] = glmm_ce_train

glmm_auc_train[i] = glmm_auc_train

glmm_acc_test[i] = glmm_acc_test

glmm_sens_test[i] = glmm_sens_test

glmm_spec_test[i] = glmm_spec_test

glmm_ce_test[i] = glmm_ce_test

glmm_auc_test[i] = glmm_auc_test


merf_acc_train[i] = merf_acc_train
```

```
merf_sens_train[i] = merf_sens_train
merf_spec_train[i] = merf_spec_train
merf_ce_train[i] = merf_ce_train
merf_auc_train[i] = merf_auc_train
merf_acc_test[i] = merf_acc_test
merf_sens_test[i] = merf_sens_test
merf_spec_test[i] = merf_spec_test
merf_ce_test[i] = merf_ce_test
merf_auc_test[i] = merf_auc_test

icc_calc[i] = icc_calc
l2var_calc[i] = l2var_calc

#### Generating data & creating Training & Test Data ####
raw_data = sim.multi(n.obs = ngroups, # N clusters
                     nvar = 3, # N L1 Predictors
                     nfact = 1, # L2 Predictor, aggregate factor of L1 predictors
                     ntrials = ncases, # L1/Time variable
                     days = ncases, # N time points/cluster size; match ntrials
                     mu = c(0, 0, 0), # Grand mean for each variable across people
```

```
                    sigma = rbg, # L2 SD (bt/w person); RBG
                    fact = 3,
                    loading = 0.31,
                    phi = 0.31, # L2 Bt/w factor intercorrelations
                    phi.i = c(0.13, 0.13, 0.13), # L1 factor intercorrelations
                    beta.i = 0.39, # L1 d/t (time coefficient); scalar of actual cor ~ 0.13
                    sigma.i = rwg, # W/in person standard deviation; RWG
                    AR1 = 0, # Autoregressive effect
                    plot = FALSE)
raw_data$y = as.factor(ifelse(raw_data$F1 >= quantile(raw_data$F1, gsr), 1, 0))
raw_data$time = raw_data$time/24 # Normalizing from 24 hour cycle

raw_data2 = raw_data %>%
  group_by(id) %>%
  mutate(p1 = correlatedValue(as.numeric(y), r = 0.93),
         p2 = correlatedValue(as.numeric(y), r = 0.93)) # scalar of actual x,x cor ~ 0.31
p = raw_data2 %>%
  summarise(p1 = mean(p1),
            p2 = mean(p2)) %>%
  as.data.frame()
```

```
p1 = rep(p$p1, each = ncases)
p2 = rep(p$p2, each = ncases)

raw_data = cbind(raw_data, p1, p2)

raw_long = as.data.frame(cbind(raw_data$y, raw_data$time, raw_data$p1, raw_data$p2,
                               raw_data$V1, raw_data$V2, raw_data$V3, raw_data$F1, raw_data$id))
names(raw_long) = c("y", "time", "p1", "p2", "V1", "V2", "V3", "F1", "id")
raw_long[2:8] = scale(raw_long[2:8], scale = TRUE, center = TRUE)

# To test correlations
# raw_long$y = as.numeric(raw_long$y)
# cor(raw_long[1:8])
raw_long$y = raw_long$y -1
raw_long$y = as.factor(raw_long$y)
raw_long$id = as.factor(raw_long$id)

# Creating train & test split
```

```
trainIndex = createDataPartition(raw_long$id, times = 1, p = 0.5, list = FALSE) # Randomly
sample by ID
train_long = raw_long %>%
  filter(id %in% trainIndex)
test_long  = raw_long %>%
  filter(!id %in% trainIndex)

#### Capturing actual average ICC ####
# Full Dataset ICC
m0 = glmer(y ~ 1 + (1|id), data = raw_long,
           family = binomial(link = "logit"))
l2var = unlist(VarCorr(m0))
l2var_calc[i] = sqrt(l2var)
icc_calc[i] = l2var/(l2var + ((pi ^ 2) / 3))

icc_calc_avg = mean(as.matrix(icc_calc), na.rm = TRUE)
l2var_avg = mean(as.matrix(l2var_calc), na.rm = TRUE)

#### Logistic Regression (LR) ####
mglm = glm(y ~ time + p1 + p2, data = train_long, family = "binomial")
```

```
mglm_pred = predict(mglm, train_long, type = "response")
mglm_cm_train = confusionMatrix(as.factor(
  ifelse(mglm_pred >= 0.5, 1, 0)), train_long$y, positive = "1")

# Iteration outcome metrics; training
lr_acc_train[i] = mglm_cm_train$overall
lr_sens_train[i] = mglm_cm_train$byClass[1]
lr_spec_train[i] = mglm_cm_train$byClass[2]
lr_ce_train[i] = LogLoss(mglm_pred, as.numeric(train_long$y))
lr_auc_train[i] = AUC(mglm_pred, train_long$y)

# Average outcome metrics; training
lr_acc_train_avg = mean(as.matrix(lr_acc_train), na.rm = TRUE)
lr_sens_train_avg = mean(as.matrix(lr_sens_train), na.rm = TRUE)
lr_spec_train_avg = mean(as.matrix(lr_spec_train), na.rm = TRUE)
lr_ce_train_avg = mean(as.matrix(lr_ce_train), na.rm = TRUE)
lr_auc_train_avg = mean(as.matrix(lr_auc_train), na.rm = TRUE)

# Predictions for test data
mglm_test = predict(mglm, test_long, type = "response")
```

```
mglm_cm_test = confusionMatrix(as.factor(
  ifelse(mglm_test >= 0.5, 1, 0)), test_long$y, positive = "1")

# Iteration outcome metrics; test
lr_acc_test[i] = mglm_cm_test$overall
lr_sens_test[i] = mglm_cm_test$byClass[1]
lr_spec_test[i] = mglm_cm_test$byClass[2]
lr_ce_test[i] = LogLoss(mglm_pred, as.numeric(test_long$y))
lr_auc_test[i] = AUC(mglm_pred, test_long$y)

# Average outcome metrics; test
lr_acc_test_avg = mean(as.matrix(lr_acc_test), na.rm = TRUE)
lr_sens_test_avg = mean(as.matrix(lr_sens_test), na.rm = TRUE)
lr_spec_test_avg = mean(as.matrix(lr_spec_test), na.rm = TRUE)
lr_ce_test_avg = mean(as.matrix(lr_ce_test), na.rm = TRUE)
lr_auc_test_avg = mean(as.matrix(lr_auc_test), na.rm = TRUE)

#### Random Forest (RF) ####
mrf = ranger(y ~ time + p1 + p2,
             data = train_long,
```

```
              num.trees = 200,
              mtry = 3,
              importance = "impurity",
              probability = TRUE)
mrf_pred = as.factor(ifelse(mrf$predictions[,1] <= 0.5, 1, 0))
mrf_cm_train = confusionMatrix(mrf_pred, train_long$y, positive = "1")

# Iteration outcome metrics; training
rf_acc_train[i] = mrf_cm_train$overall
rf_sens_train[i] = mrf_cm_train$byClass[1]
rf_spec_train[i] = mrf_cm_train$byClass[2]
rf_ce_train[i] = LogLoss(as.vector(1 - mrf$predictions[,1]), as.numeric(train_long$y))
rf_auc_train[i] = AUC(as.vector(1 - mrf$predictions[,1]), train_long$y)

# Average outcome metrics; training
rf_acc_train_avg = mean(as.matrix(rf_acc_train), na.rm = TRUE)
rf_sens_train_avg = mean(as.matrix(rf_sens_train), na.rm = TRUE)
rf_spec_train_avg = mean(as.matrix(rf_spec_train), na.rm = TRUE)
rf_ce_train_avg = mean(as.matrix(rf_ce_train), na.rm = TRUE)
rf_auc_train_avg = mean(as.matrix(rf_auc_train), na.rm = TRUE)
```

```
# Predictions for test data
mrf_test = predict(mrf, test_long)
mrf_cm_test = confusionMatrix(as.factor(
  ifelse(mrf_test$predictions[,1] <= 0.5, 1, 0)), test_long$y, positive = "1")

# Iteration outcome metrics; test
rf_acc_test[i] = mrf_cm_test$overall
rf_sens_test[i] = mrf_cm_test$byClass[1]
rf_spec_test[i] = mrf_cm_test$byClass[2]
rf_ce_test[i] = LogLoss(as.vector(1 - mrf$predictions[,1]), as.numeric(test_long$y))
rf_auc_test[i] = AUC(as.vector(1 - mrf$predictions[,1]), test_long$y)

# Average outcome metrics; test
rf_acc_test_avg = mean(as.matrix(rf_acc_test), na.rm = TRUE)
rf_sens_test_avg = mean(as.matrix(rf_sens_test), na.rm = TRUE)
rf_spec_test_avg = mean(as.matrix(rf_spec_test), na.rm = TRUE)
rf_ce_test_avg = mean(as.matrix(rf_ce_test), na.rm = TRUE)
rf_auc_test_avg = mean(as.matrix(rf_auc_test), na.rm = TRUE)
```

```
#### Generalize Linear Mixed Effects Model (GLMM) ####
mglmer = glmer(y ~ time + p1 + p2 + (1|id),
               data = train_long,
               family = binomial(link = "logit"),
               control = glmerControl(optimizer = "bobyqa",
                                      boundary.tol = 1e-1,
                                      check.conv.singular = .makeCC(action = "ignore",
                                                                    tol = 1e-1),
                                      tolPwrss = 1e-1))
mglmer_pred = predict(mglmer, train_long, type = "response", allow.new.levels = TRUE)
mglmer_cm_train = confusionMatrix(as.factor(
  ifelse(mglmer_pred >= 0.5, 1, 0)), train_long$y, positive = "1")

# Iteration outcome metrics; training
glmm_acc_train[i] = mglmer_cm_train$overall
glmm_sens_train[i] = mglmer_cm_train$byClass[1]
glmm_spec_train[i] = mglmer_cm_train$byClass[2]
glmm_ce_train[i] = LogLoss(as.numeric(mglmer_pred), as.numeric(train_long$y))
glmm_auc_train[i] = AUC(mglmer_pred, train_long$y)
```

```
# Average outcome metrics per simcount; training
glmm_acc_train_avg = mean(as.matrix(glmm_acc_train), na.rm = TRUE)
glmm_sens_train_avg = mean(as.matrix(glmm_sens_train), na.rm = TRUE)
glmm_spec_train_avg = mean(as.matrix(glmm_spec_train), na.rm = TRUE)
glmm_ce_train_avg = mean(as.matrix(glmm_ce_train), na.rm = TRUE)
glmm_auc_train_avg = mean(as.matrix(glmm_auc_train), na.rm = TRUE)

# Test data predictions
mglmer_test = predict(mglmer, test_long, type = "response", allow.new.levels = TRUE)
mglmer_test_cm = confusionMatrix(as.factor(
  ifelse(mglmer_test >= 0.5, 1, 0)), test_long$y, positive = "1")

# Iteration outcome metrics; test
glmm_acc_test[i] = mglmer_test_cm$overall
glmm_sens_test[i] = mglmer_test_cm$byClass[1]
glmm_spec_test[i] = mglmer_test_cm$byClass[2]
glmm_ce_test[i] = LogLoss(as.numeric(mglmer_pred), as.numeric(test_long$y))
glmm_auc_test[i] = AUC(mglmer_pred, test_long$y)

# Average outcome metrics per simcount; test
```

```
glmm_acc_test_avg = mean(as.matrix(glmm_acc_test), na.rm = TRUE)
glmm_sens_test_avg = mean(as.matrix(glmm_sens_test), na.rm = TRUE)
glmm_spec_test_avg = mean(as.matrix(glmm_spec_test), na.rm = TRUE)
glmm_ce_test_avg = mean(as.matrix(glmm_ce_test), na.rm = TRUE)
glmm_auc_test_avg = mean(as.matrix(glmm_auc_test), na.rm = TRUE)

#### Mixed Effects Random Foreste (MERF) ####
train_long$id = as.numeric(train_long$id)  ## random effect grouping variable
test_long$id = as.numeric(test_long$id)

mmerf <- MEml(classifier = "RF",
              dat.trn = train_long,
              dat.tst = test_long,
              id = "id",
              resp.vars = "y",
              rhs.vars = c("p1", "p2"),
              order.vars = "time",
              rand.vars = ~id,
              para = list(
                method = "cv",
```

```
  tuneLength = 1,
  number = 3,
  ntree = 200,
  mtry = 3,
  interaction.depth = 3,
  shrinkage = 0.01,
  m.minobsinnode = 1,
  opt.para = TRUE,
  coefReg = 0.5,
  coefImp = 1,
  include.RE = FALSE,
  con.tree = FALSE,
  max.iter = 10, alpha = 0.05, minsize = 20, maxdepth = 30,
  K = 3, decay = 0.05, tol = 1e-5, seed = 1942773,
  trControl = trainControl(method = "repeatedcv",
                           repeats = 3,
                           classProbs = TRUE,
                           summaryFunction = twoClassSummary)
),
max.iter = 100,
```

```
              return.model = TRUE)

# Iteration outcome metrics; training
merf_acc_train[i] = (mmerf$RF$perf$sensitivity + mmerf$RF$perf$specificity) / 2
merf_sens_train[i] = mmerf$RF$perf$sensitivity
merf_spec_train[i] = mmerf$RF$perf$specificity
merf_ce_train[i] = LogLoss(as.vector(mmerf$RF$pred$Yes), as.numeric(train_long$y))
merf_auc_train[i] = mmerf$RF$perf$AUC

# Average outcome metrics; training
merf_acc_train_avg = mean(as.matrix(merf_acc_train), na.rm = TRUE)
merf_sens_train_avg = mean(as.matrix(merf_sens_train), na.rm = TRUE)
merf_spec_train_avg = mean(as.matrix(merf_spec_train), na.rm = TRUE)
merf_ce_train_avg = mean(as.matrix(merf_ce_train), na.rm = TRUE)
merf_auc_train_avg = mean(as.matrix(merf_auc_train), na.rm = TRUE)

# Test data predictions
mmerf_test = mmerf$RF$model$modelInfo$prob(mmerf$RF$model, test_long)
mmerf_test_pred = as.factor(ifelse(mmerf_test$No <= 0.5, 1, 0))
mmerf_cm_test = confusionMatrix(mmerf_test_pred, test_long$y, positive = "1")
```

```
# Iteration outcome metrics; test
merf_acc_test[i] = mmerf_cm_test$overall
merf_sens_test[i] = mmerf_cm_test$byClass[1]
merf_spec_test[i] = mmerf_cm_test$byClass[2]
merf_ce_test[i] = LogLoss(as.vector(mmerf_test$Yes), as.numeric(test_long$y))
merf_auc_test[i] = AUC(mmerf_test$Yes, test_long$y)

# Average outcome metrics; test
merf_acc_test_avg = mean(as.matrix(merf_acc_test), na.rm = TRUE)
merf_sens_test_avg = mean(as.matrix(merf_sens_test), na.rm = TRUE)
merf_spec_test_avg = mean(as.matrix(merf_spec_test), na.rm = TRUE)
merf_ce_test_avg = mean(as.matrix(merf_ce_test), na.rm = TRUE)
merf_auc_test_avg = mean(as.matrix(merf_auc_test), na.rm = TRUE)

#### Creating output dataframes & Compiling Results ####
full_results = data.frame(simcount, ncases, ngroups, rwg, rbg, gsr, icc,
                          icc_calc_avg, l2var_avg,
                          lr_acc_train_avg, lr_sens_train_avg, lr_spec_train_avg,
                          lr_ce_train_avg, lr_auc_train_avg,
```

```
                        lr_acc_test_avg, lr_sens_test_avg, lr_spec_test_avg,
                        lr_ce_test_avg, lr_auc_test_avg,
                        rf_acc_train_avg, rf_sens_train_avg, rf_spec_train_avg,
                        rf_ce_train_avg, rf_auc_train_avg,
                        rf_acc_test_avg, rf_sens_test_avg, rf_spec_test_avg,
                        rf_ce_test_avg, rf_auc_test_avg,
                        glmm_acc_train_avg, glmm_sens_train_avg, glmm_spec_train_avg,
                        glmm_ce_train_avg, glmm_auc_train_avg,
                        glmm_acc_test_avg, glmm_sens_test_avg, glmm_spec_test_avg,
                        glmm_ce_test_avg, glmm_auc_test_avg,
                        merf_acc_train_avg, merf_sens_train_avg, merf_spec_train_avg,
                        merf_ce_train_avg, merf_auc_train_avg,
                        merf_acc_test_avg, merf_sens_test_avg, merf_spec_test_avg,
                        merf_ce_test_avg, merf_auc_test_avg)
print(i)
#info = sprintf("%d%% done", round(i))
#setTkProgressBar(pb, i, label = paste(round(i/parm_frame$g*100, 0),
#"% done"))

#close(pb)
```

```
    }
    , error=function(e){cat("ERROR :",conditionMessage(e), "\n")})
  }
  return(full_results)
}
j = 1

start_time = Sys.time()
while(j <= nrow(parm_frame))
{
  results_full = fixed_mixed_sim(ncases = parm_frame$ncases[j],
                                 ngroups = parm_frame$ngroups[j],
                                 rwg = parm_frame$rwg[j],
                                 rbg = parm_frame$rbg[j],
                                 gsr = parm_frame$gsr[j],
                                 simcount = parm_frame$simcount[j],
                                 g = parm_frame$g[j],
                                 icc = parm_frame$icc[j])

  output_frame[[j]] = results_full
```

```
  output_full = as.data.frame(do.call("rbind", output_frame))


  j = j + 1
  print(j - 1)
}
end_time = Sys.time()
duration = end_time - start_time

write.csv(output_full, file = "~/diss_sim/fixed_mixed_2L2.csv")
write.csv(duration, file = "~/diss_sim/fixed_mixed_2L2_duration.csv")

#save.image(file = "~/Desktop/Data Sets/DISSERTATION CODE/For Cluster/workspace.RData")
```