



# Predictions of machine learning with mixed-effects in analyzing longitudinal data under model misspecification

Shuwen Hu<sup>1,2</sup> · You-Gan Wang<sup>3</sup> · Christopher Drovandi<sup>1</sup> · Taoyun Cao<sup>4</sup>

Accepted: 3 September 2022  
© Crown 2022

## Abstract

We consider predictions in longitudinal studies, and investigate the well known statistical mixed-effects model, piecewise linear mixed-effects model and six different popular machine learning approaches: decision trees, bagging, random forest, boosting, support-vector machine and neural network. In order to consider the correlated data in machine learning, the random effects is combined into the traditional tree methods and random forest. Our focus is the performance of statistical modelling and machine learning especially in the cases of the misspecification of the fixed effects and the random effects. Extensive simulation studies have been carried out to evaluate the performance using a number of criteria. Two real datasets from longitudinal studies are analysed to demonstrate our findings. The R code and dataset are freely available at <https://github.com/shuwen92/MEML>.

**Keywords** Longitudinal data · Misspecification · Machine learning · Mixed-effects model · Regression tree · Support vector machine · Comparison study

## 1 Introduction

Longitudinal data, which occur frequently in economics, finance, medical science and other fields, are measured repeatedly for each subject. The circumstances under which the measurements are taken cannot be exactly the same. For example,

---

✉ You-Gan Wang  
you-gan.wang@acu.edu.au

<sup>1</sup> School of Mathematical Sciences, Queensland University of Technology, Brisbane, Australia

<sup>2</sup> CSIRO Agriculture & Food, 306 Carmody Road, St Lucia, QLD, Australia

<sup>3</sup> Institute for Learning Sciences and Teacher Education, Australian Catholic University, Brisbane, Australia

<sup>4</sup> School of Statistics and Mathematics, Guangdong University of Finance and Economics, Guangzhou, China

students could be sampled in different classrooms or patients by different doctors. Therefore, the assumption of longitudinal data is that measurements are correlated for the same subjects but independent among different subjects. If the number of measurements from each subject is the same, the datasets are said to contain balanced data; otherwise, the datasets contain unbalanced data. Laird and Ware (1982) introduced the random effects models for longitudinal data because they claimed that a general multivariate model with unrestricted covariance structure is not suited for the analysis of unbalanced data. Mixed-effects models that include both fixed and random effects can handle the correlation in longitudinal data. The fixed effects are parameters related to the levels of the entire population or certain repeatable experimental factors, while the random effects are related to individual experimental units randomly chosen from a population (Pinheiro and Bates 2000). An expectation-maximisation (EM) algorithm can be used to determine the maximum likelihood and restricted maximum likelihood estimation in the longitudinal data setting (Laird et al. 1987). Lindstrom and Bates (1988) developed an efficient and computationally stable implementation of the Newton-Raphson (NR) algorithm for obtaining the parameters in mixed-effects models for longitudinal data.

The misspecification of mixed-effects models can include the misspecification of fixed effects or random effects. Grilli and Rampichini (2015) first review the literature about the consequences of misspecifying the distribution of the random effects. McCulloch and Neuhaus (2011a) investigated the impact of misspecification of the distribution of the random effects and claimed that the prediction accuracy is little affected for mild-to-moderate violations of the assumptions. Their mild-to-moderate violations of random effects implies assumption of normal distribution of random effects has been misspecified to three different distributions: a skewed and truncated distribution, a heavy-tailed distribution, and a mixture distribution. Hui et al. (2021) focused on variance components when they studied the effects of random effects misspecification in linear mixed models. There are also other references (McCulloch and Neuhaus 2011b; Albert 2012; Drikvandi et al. 2017) investigated the misspecification of shape/distribution of random effects and they confirmed that the mean square error for random effects estimation is robust to the random effects misspecification. Misspecification of random components will lead to misspecified variance and correlation structures. Therefore, our work with a slightly different focus has been that of assessing random effects misspecification from the misspecification of correlation structure with simulated data generated from marginal model. Wang and Carey (2003) provided both asymptotic and numerical results in the GEE framework.

There have been very few comparison studies of statistical models and machine learning methods in the analysis of longitudinal data. One thing we can notice is that statistical models usually have more assumptions than machine learning methods. However, this is a double-edged sword. Machine learning methods are usually recognised as having a ‘black box’ aspect, which means there is less attention paid to the processes between their inputs and outputs. Real data sets are usually complex, and it is worthwhile to investigate more about the data before definitive decisions are made. Some papers have compared the predictive performance of statistical methods and machine learning methods in the area of health (Song et al. 2004;

Venkatesh et al. 2020; Shin et al. 2021) and air quality (Wei et al. 2019; Berrocal et al. 2020). They confirmed that the nature of data is of primary importance rather than the learning technique.

Among the six machine learning methods (trees, bagging, random forest, boosting, support-vector machine and neural network) addressed in this work, the trees method is the most broadly applied for longitudinal data (Segal 1992; Hajjem et al. 2011, 2014; Berger and Tutz 2018; Kundu and Harezlak 2019). Sela and Simonoff (2012) presented the random effects expectation-maximisation (RE-EM) tree, which combined the structure of mixed-effects models with tree-based methods. They showed that the RE-EM tree had improved predictive power over traditional linear models with random effects and regression trees without random effects. However, Fu and Simonoff (2015) proposed what they claimed are unbiased RE-EM trees by using conditional inference trees instead of classification and regression trees (CARTs). In addition, Loh and Zheng (2013) had proposed an unbiased regression tree for longitudinal data based on a generalised, unbiased interaction detection and estimation (GUIDE) approach rather than the traditional CARTs. Later, Eo and Cho (2014) combined the decision tree and mixed-effects methods for longitudinal data based on GUIDE. Hajjem et al. (2014) have extended their methodology with the use of random forest instead of regression trees, which called mixed effects random forest (MERF). A framework for predicting longitudinal change in glycemic control measured by hemoglobin A1c (HbA1c) using mixed effect machine learning is presented by Ngufo et al. (2019). The machine learning methods can be applied to regression as well as classification. There are some progress in the development of mixed-effects machine learning methods with application of classification, such as generalized mixed-effects regression trees (Hajjem et al. 2017), generalized mixed-effects random forest (Pellagatti et al. 2021) and neural networks for longitudinal data (Crane-Droesch. 2017; Xiong et al. 2019). Mangino and Finch (2021) utilised a Monte Carlo simulation to compare the prediction performance of several classification algorithms and they claimed the panel neural network and Bayesian generalized mixed effects models have the highest prediction accuracy. We focus on the regression in this work in order to compare the prediction performance of linear mixed models and machine learning methods with or without mixed effects when the model is specified correctly or misspecified.

Li and Wu (2015) claimed that the traditional linear mixed model is inferior to the machine learning methods for both long- and short-term prediction in milk protein data, which is apparently because the linear mixed model is not sufficient to fit this data. This milk protein data was also illustrated by Diggle et al. (2002) using a piecewise model at breakpoint three with an exponential correlation structure. However, we noticed that the quadratic term is not necessary, and a piecewise mixed-effects model would have better performance. Yang et al. (2016) illustrated the mathematical programming for a piecewise linear regression analysis. They showed that the piecewise regression method achieved better prediction performance than a number of state-of-the-art regression methods, such as random forest (RF), support-vector regression (SVR), K-nearest neighbour (KNN) and so on. Kohli et al. (2018) investigated the estimation of a piecewise mixed-effects model with unknown breakpoints using maximum likelihood. They found that the maximum likelihood estimates are reliable and accurate under the conditions that the observed variables had a small

residual variance. The mixed-effects tree-based method is emphasized because it has shown strong prediction performance and it is explainable.

The estimation of parameters in the mixed-effects machine learning usually relied on two steps: estimation of mean function and random effect component, respectively. As far as we know, the literature lacks a comparison of the performance of statistical models and machine learning methods for longitudinal data when the fixed effects or random effect are misspecified. However, correctly specification of mean function/fixed effects and random effect components are very important in the longitudinal data analysis (Wang and Lin 2005). A new metric, true root mean square error (TRMSE) is defined to measure how close the predictions would be to the true values without noise error in the simulation. The differences between the TRMSE and RMSE are also presented according to the simulation parts. Two different ways are utilised to generate correlated data. One way is to generate data from mixed-effects models with fixed effects and random effects, the other is to generate data from a marginal model.

In this paper, we review and compare the performances of a mixed-effects model and six machine learning methods (tree, bagging, random forest, boosting, support-vector machine and neural network) and two mixed effects machine learning methods (RE-EM trees and MERF) in the prediction of longitudinal data. The remainder of this work is organized as follows. Section 2 describes the various methods that we compared in this work. In Sect. 3, a description is made of the extensive simulations that are carried out to evaluate the performance of the different methods. Two different kinds of real data (milk protein and wages) are considered as case studies in Sect. 4. Section 5 presents some conclusions and further discussion.

## 2 Methods

In this section, the details of the linear mixed-effects model, tree-based method (including the RE-EM tree), support-vector machine and neural network are introduced.

### 2.1 Linear mixed-effects models

Linear mixed-effects models are an extension of simple linear models by the inclusion of random effects that are used to account for the correlation among measurements within the same subject.

Let response vector  $\mathbf{Y}_i$  be the  $n \times 1$  vector  $(y_{i1}, \dots, y_{in})^T$ , in which  $y_{ij}$  is the  $j$ th measurement for the  $i$ th subject ( $i = 1, \dots, K$ ,  $j = 1, \dots, n$ ). The total number of subjects is  $K$ .  $\mathbf{X}_i$  (of dimension  $n \times p$ ) and  $\mathbf{Z}_i$  (of dimension  $n \times q$ ) are the separate fixed-effect and random-effect covariates.  $\boldsymbol{\beta}$  is a  $p$ -dimensional vector of the fixed effect, and  $\mathbf{b}_i$  is a  $q$ -dimensional vector of the random effect, which are assumed to be Gaussian distributed with mean zero and variance  $\boldsymbol{\Psi}$ . The formulation of the linear mixed-effects model is as follows:

$$\begin{aligned}
 Y_i &= X_i\beta + Z_i b_i + \epsilon_i, \\
 b_i &\sim N(0, \Psi), \epsilon_i \sim N(0, \sigma^2 \Lambda_i).
 \end{aligned}
 \tag{1}$$

The within-groups errors  $\epsilon_i$  and the random effects  $b_i$  are assumed to be independent. It is a special case if  $\Lambda_i = I$ . Then, it follows that  $Y_i \sim N(X_i\beta, \Sigma_i)$ , where  $\Sigma_i = \sigma^2(\Lambda_i + Z_i\Psi Z_i^T)$ . The matrix form for the model is as follows:

$$Y = X\beta + Zb + \epsilon,$$

where  $Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_K \end{bmatrix}, \quad X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_K \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix}, \quad Z = \text{diag}(Z_1, Z_2, \dots, Z_K),$

$\Lambda = \text{diag}(\Lambda_1, \Lambda_2, \dots, \Lambda_K)$ ,  $\Sigma = \text{diag}(\Sigma_1, \Sigma_2, \dots, \Sigma_K)$  and  $\tilde{\Psi} = \text{diag}(\Psi, \Psi, \dots, \Psi)$ . It follows that  $Y$  are independent multivariate normal vectors with mean  $X\beta$  and the covariance matrix is  $\Sigma = \sigma^2(\Lambda + Z\tilde{\Psi}Z^T)$ . Then, the likelihood function is

$$L(\beta, \theta, \sigma^2 | Y) = p(Y | \beta, \theta, \sigma^2) = \prod_{i=1}^K p(Y_i | \beta, \theta, \sigma^2),$$

where

$$p(Y_i | \beta, \theta, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(\frac{(Y_i - X_i\beta^T)\Sigma_i^{-1}(Y_i - X_i\beta)}{-2\sigma^2}\right) |\Sigma_i|^{-\frac{1}{2}}$$

and  $\theta$  represents the parameters in  $\tilde{\Psi}$  and  $\Lambda$ . An EM algorithm can be used to obtain both the maximum likelihood and restricted maximum likelihood estimation according to Laird et al. (1987). The *lme* function of the R-package *nlme* is implemented to fit the linear mixed model (Pinheiro et al. 2020).

### 2.2 Piecewise linear mixed-effects models

Piecewise regression is a special type of linear regression that arises when a single line is not sufficient to model a data set. Piecewise regression breaks the domain into potentially many ‘segments’ and fits a separate line through each one. Breakpoints are the values where the slope of the linear function changes. The value of the breakpoints are unknown and must be estimated. In some cases, the breakpoints can be specified by us according to plots. In other words, it is obvious to the naked eyes when one linear trends give way to other. However, this is not fit for all the cases. For some data set, it is not easy to detect the breakpoints just from eyes. In statistics, the popular way is to compare the errors with different breakpoints, which means minimize the errors between each segment’s regression and the observed data points.

A piecewise linear mixed-effects (PLME) model is an extension of linear mixed-effects model. The PLME has been used in many areas, such as in analysing longitudinal educational and psychological data sets (Kohli et al. 2018, 2015). We

introduced PLME in this work because of its flexibility for accommodating a different mean function in each phase. The mathematical forms of PLME are presented in Sect. 4.1 to analyse the milk protein data.

## 2.3 Tree-based methods

### 2.3.1 Decision trees

Tree-based methods, support-vector machine and neural network can be applied to regression as well as classification, and we focus on regression problems in this work. The decision tree, bagging, random forest, and boosting methods can be grouped together as they are all tree-based methods. CART (classification and regression tree) is a popular algorithm which was proposed by Breiman et al. (1984). In the tree method, the training data is used to construct a data tree starting at the root node. The predicted space is divided into non-overlapping  $M$  regions ( $R_1, R_2, \dots, R_M$ ) determined by recursive splitting, which is a top-down and greedy approach (James et al. 2013). In each region, a constant  $c_m$  would be the response. The model is as follows:

$$f(X) = \sum_{m=1}^M c_m I(X \in R_m).$$

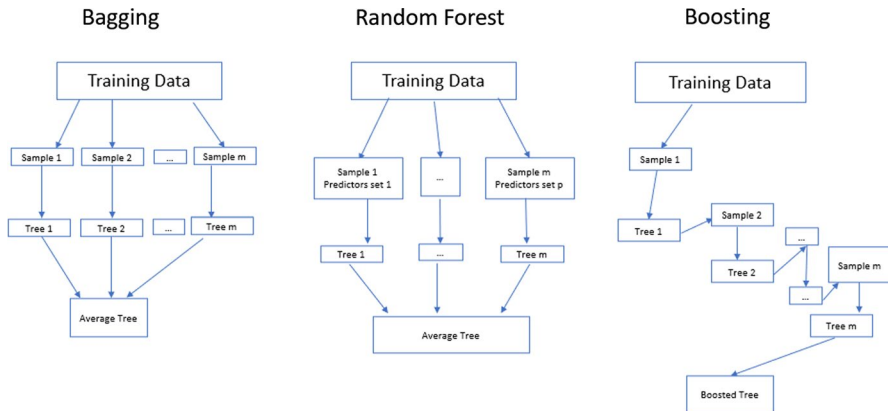
The splitting we choose will cause the largest reduction in the mean square error. We can split recursively until the mean square error reaches a defined threshold. Then it is easy to see that the best value is the average of  $Y$  in region  $R_m$ :

$$\hat{c}_m = \text{ave}(Y|X \in R_m).$$

The predicted response for a test data point is the mean of the training observations in the region to which that test point belongs. For each test data point that falls in the same region on a path starting from the root node until reaching a terminal (leaf) node, the response prediction would be the same. A usual strategy to fit a single tree is to grow a large tree and then trim it by weakest link pruning. The R-package *tree* is used to implement the above process in this work (Ripley 2019). Trees can be displayed graphically and are easy to explain but can be subject to overfitting. Also, trees are not robust, which means small changes in the training data can cause very different series of splits. Ensemble decision tree methods, including bagging, random forests and boosting, combine many decision trees to produce better predictive performances than a single decision tree.

### 2.3.2 Ensemble decision tree methods

Bagging trees, random forest and boosting trees are called ensemble decision trees. The simple flowchart of these three different ensemble decision trees is presented in Fig. 1.



**Fig. 1** Three different ensemble trees

Bagging is the application of the bootstrap procedure to decision trees in order to lower the variance. There are three main steps: firstly, generate random subsamples of the training data set with replacement; secondly, train the decision tree method on each sample; and thirdly, calculate the average prediction from each model using the test data. The average prediction would be the final prediction for each test data point. Bagging will improve the prediction accuracy compared to the tree method at the cost of interpretability.

Random forest is a popular tree-based ensemble method that builds a large collection of de-correlated trees and then averages them based on the bagging (Breiman 2001). When building this algorithm, a random sample of features is chosen as split candidates from the full set of predictors rather than using all the features in bagging. This forces each split to consider only a subset of the predictors, which is reasonable, especially when there is a very strong predictor in the training data set. After a certain number of trees are grown, the predictor is obtained by the average (for regression) or the majority vote (for classification) (James et al. 2013). This algorithm contains four main parameters: total number of observations, total number of predictor variables, randomly chosen features for determining the decision tree and the total number of decision trees. The R-package *randomForest* is used to implement the algorithm of bagging and random forest (Liaw and Wiener 2002).

The different trees based on the bootstrapped data are independent in bagging. Boosting works in a similar way to bagging, but the difference is the trees are constructed sequentially, which means that the growth of each tree depends on the trees that have already been constructed. It is a forward stagewise approach. Boosting regression trees (BRT) have three parameters: the number of trees, the shrinkage parameter that controls the learning rate and the number of splits in each tree that determines the complexity of the boosted ensemble. The BRT algorithm has three main steps: firstly, a regression tree is fitted; secondly, another tree is fitted to the residuals of the first tree; and thirdly, the model is updated to have two trees with a shrinkage parameter (this last step is repeated hundreds or thousands of times).

The final model is a linear combination of these trees. The R-package *gbm* is implemented for this algorithm (Greenwell et al. 2019).

## 2.4 Mixed-effects regression trees and random forest

Segal (1992) was the first to apply regression trees to longitudinal data. The mixed-effects tree method we have used in this work, the RE-EM tree, was proposed by Sela and Simonoff (2012). The notation in an RE-EM tree follows the linear mixed-effects model:

$$Y_i = f(X_i) + Z_i b_i + \epsilon_i, \quad (2)$$

in which the  $Y_i, X_i, Z_i, b_i$  and  $\epsilon_i$  analogous to their use in equation (1). If  $f$  is a linear function,  $f(X_i) = X_i \beta$ , then the model is a linear mixed model. Generally, this  $f$  function can be estimated by a tree method when the random effects  $b_i$  are known. However, when neither the fixed effects nor the random effects are known, an iterative two-step process is utilised. Firstly, the random effects  $\hat{b}_i$  are set to zero initially, and a regression tree is used to estimate function  $f$  based on  $Y_i - Z_i \hat{b}_i$ . A linear mixed-effects model is then fitted to estimate the random effects based on the tree regression results:  $y_{ij} = Z_{ij} b_i + I(X_{ij} \in g_p) \mu_p + \epsilon_{ij}$ , in which  $I(X_{ij} \in g_p) \mu_p$  means the estimated value for  $y_{ij}$  at terminal node  $g_p$ . The algorithm will not stop until the estimates of random effects  $\hat{b}_i$  converge. We used R package *REEMtree* (Sela and Simonoff 2012) in this work.

Hajjem et al. (2014) proposed mixed-effects random forest (MERF) for clustered data which implemented using a standard random forest algorithm within the framework of the expectation-maximization (EM) algorithm. The notations of MERF are the same with Equation (2) and the random forest is used to estimate the fixed part of the model, i.e., the estimation of function  $f$ . The MERF algorithm is similar to the EM algorithm for the linear mixed-effects model and the detailed steps of the MERF algorithm can be found in Hajjem et al. (2014). Louis (2020) implemented this MERF algorithm in R package *LongituRF*.

## 2.5 Support-vector machine

The initial idea of a support-vector machine (SVM) is to construct a linear partition of the high-dimensional space into two sub-spaces for classification or regression (Scholkopf and Smola 2002). We will focus on the regression application in this work. Given the training data  $(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N)$ , the prediction is shown as a linear function  $f(X) = \omega^T X + b_0$ , and the error function is

$$\frac{1}{2} \|\omega\|^2 + C \sum_{k=1}^N \ell_\epsilon(f(X_k) - Y_k), \quad (3)$$

where  $\ell_\epsilon(z) = \max\{0, |z| - \epsilon\}$  is the  $\epsilon$ -insensitive loss. After minimising the error function, the solution is



$$f(\mathbf{X}) = \sum_{k=1}^N (\hat{\alpha}_k - \alpha_k) \mathbf{X}_k^T \mathbf{X} + b_0, \quad (4)$$

where  $\hat{\alpha}_k$  and  $\alpha_k$  are Lagrange multipliers. The nonzero Lagrange multipliers that indicate the training vector makes  $(\hat{\alpha}_k - \alpha_k) \neq 0$  in Equation (4) are called support-vectors. Obviously, the non-support-vectors do not contribute directly to the solution. Besides this linear case, the data are often not linearly separable. A kernel function is then used to transform the nonlinear system in the input space to a linear system in the feature space. Popular kernel functions are polynomial kernel, radial kernel, among others (James et al. 2013). We will also include the support-vector regression with a polynomial kernel in our simulations. The *svm* function of the R-package *e1071* is used (Meyer et al. 2019).

## 2.6 Neural network

In this work, we consider the neural network as a multilayer perceptron (MLP), which is a class of feedforward artificial neural network. The multilayer perceptron is a popular network for classification and regression. The formula is as follows:

$$\hat{Y} = \sum_{k=0}^{NH} v_k g_k(\mathbf{X}),$$

where  $v_k$  are the weights,  $g_k(\mathbf{X})$  are the hidden functions (or hidden units), NH is the number of hidden nodes,  $\mathbf{X}$  is the input vector and  $\hat{Y}$  is the output. Here, the hidden function  $g_0$  takes a fixed value of one to allow a constant term in the equation. The sigmoid function is commonly used:  $g_k(u) = 1/(1 + \exp(-u))$ . Also, the tangent hyperbolic function  $\tanh x = (e^x - e^{-x})/(e^x + e^{-x})$  is used in the simulation. The general approach to fit this method is minimizing the sum-of-squared errors by gradient descent, which is called back-propagation. The R-package *neuralnet* is used to implement the neural network algorithm (Fritsch et al. 2019).

## 3 Simulation studies

In this section, we describe the investigation of the performances of linear mixed-effects models and machine learning methods through extensive simulations.

### 3.1 Design of simulations

There are two types of misspecification in the linear mixed model: misspecification of fixed effects and misspecification of random effects. Therefore, in our design of simulations, we generate the longitudinal data in two different ways to deal with these two different kinds of misspecification. To analyse misspecification of the fixed effects, we consider two different true mean functions: linear and quadratic in our setting. The performance of various models only with linear mean function would be

investigated, which means the fixed effects are misspecified if the true mean function is quadratic. The longitudinal data can also be generated from the marginal model with different correlation structures, such as AR(1) or exchangeable correlation. This data generation aimed to reflect the misspecification of random effects. The data generated from the linear mixed-effects model with random intercept are equivalent to that generated from the marginal model with the same mean function and exchangeable correlation. If the data are generated with exchangeable correlation structure, the linear mixed effects model with random intercept is the true model. Otherwise, it could be considered as the misspecification of random effects. The details of data generation are provided in the following paragraph and Table 1.

Two different ways are used to generate the longitudinal data. One is from the mixed model:

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 g_i + \beta_3 x_{ij}^2 + \mathbf{b}_i + \epsilon_{ij}, i = 1, \dots, K, j = 1, \dots, n, \tag{5}$$

where  $\mu_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 g_i + \beta_3 x_{ij}^2$  is the overall mean response,  $\mathbf{b}_i$  is the random effects from the normal distribution  $N(0, \sigma_b^2)$  and  $\epsilon_{ij}$  come from an iid normal distribution  $N(0, \sigma^2)$ . In addition,  $x_{ij}$  and  $g_i$  are sampled from the uniform distribution  $(0, 1)$ . In our simulation, there are two different true mean function:  $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T = (0.5, 1, 1.2, 0)^T$  and  $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T = (0.5, 1, 1.2, -5)^T$ , which demonstrate the true mean function is linear and quadratic respectively. In this simulation, we set  $\sigma = 1$  and  $\sigma_b = 2$ .

Another way to generate the longitudinal data from the following marginal model:

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 s_i + \beta_3 x_{ij}^2 + \epsilon_{ij}, i = 1, \dots, K, j = 1, \dots, n, \tag{6}$$

where  $\mu_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 s_i + \beta_3 x_{ij}^2$ ,  $x_{ij}$  is sampled from the uniform distribution  $(0, 1)$ , and  $s_i$  is sampled from the binary distribution, which can represent the sex variable in the real dataset. In order to compare the performances between different methods under a scenario in which the mean function is correctly specified and misspecified, we have set different values for  $\beta$ . In the model in which the mean function is correctly specified,  $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T = (0, 0.5, 1, 0)^T$ . This indicates the true mean function of the simulated data is linear (without a quadratic term). In contrast,  $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T = (0, 0.5, 1, -5)^T$  is used when the mean function is misspecified, which indicates the data are generated from a quadratic model. We used

**Table 1** The design of simulations

Longitudinal data generation			
Mixed model		Marginal model	
Linear mean function	Quadratic mean function	Exchangeable	AR(1)
↓	↓	↓	↓
Correctly specified	Misspecified	Correctly specified	Misspecified
Fixed effects		Random effects	

the linear mean function in various prediction models. We had two different scenarios for  $\epsilon_{ij}$ : the first is  $\epsilon_{ij}$  are correlated with an exchangeable structure, in other words,  $\text{cor}(\epsilon_{ij}, \epsilon_{i'j'}) = 0.5$  if  $j \neq j'$ ; and the second is for each  $i$ ,  $(\epsilon_{i1}, \dots, \epsilon_{in})$  are correlated with an autoregressive AR(1) structure that also had a correlation coefficient of 0.5. It is worth noting that when the data are generated from the linear mean function with first scenario (i.e., the correlation structure is exchangeable), the linear mixed-effects model with a random intercept is the true model. Otherwise, when the data are correlated with the AR(1) structure, the linear mixed-effects model is not the true model even if the mean function of simulated data is linear.

### 3.2 Evaluation metrics

There are few references about how to measure the predictive power of methods for longitudinal data. The stratified cross-validation method cannot be used directly because the observations from longitudinal data contain sequences. Based on Sela and Simonoff (2012), three different ways are utilised: (1) predicting the future 30% of observations based on the previous 70% of observations for  $K$  different subjects, denoted as *future observation*; (2) predicting another new  $K/2$  objects based on the previous  $K$  different subjects, denoted as *new object*; and (3) predicting a future 30% of observations for new  $K/2$  objects based on the previous  $K$  different subjects and the previous 70% of observations in the new  $K/2$  subjects, denoted as *future new observation*. In this case, there are 100 subjects ( $K = 100$ ), and each subject is observed 10 times ( $n = 10$ ).

We also proposed another one-step prediction and two-step prediction in order to see the performances of different methods in real-time prediction. In this case,  $K = 100$  and  $n = 5$ . This is a bit similar to the future observation method described above but did not just consist of the overall prediction. We can obtain the prediction performance at the time of each observation. In the one-step prediction, we used the first observation to predict the second observation; the first two observations are then used to predict the third, and so on. In contrast, the first observation is used to predict the third observation in the two-step prediction. The results based on 1000 simulations are presented in Tables 2, 3, 4 and 5. The numerosity of the generated data is 1000 and 500 for the two different prediction performance evaluations, respectively.

To measure the prediction performance of the different methods, the root mean square error (RMSE) is used:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^n (y_{ij} - \hat{y}_{ij})^2}{K * n}},$$

where  $y_i$  is the measured value, and  $\hat{y}_i$  is the predicted value. Because the true values of  $\mu_i$  are known in the simulations and the curious is about how close the predictions would be to the true values without noise, we defined another metric, the true root mean square error (TRMSE), to measure the prediction performance as follows:

**Table 2** The one-step prediction for different methods in simulated data generated from a mixed-effects model

	Second observation		Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
<i>(a) Correct specification of the mean</i>								
lme	<b>1.379</b>	1.960	<b>1.206</b>	1.881	<b>1.148</b>	1.914	<b>1.113</b>	1.931
tree	2.276	0.745	2.209	0.775	2.147	0.926	2.056	1.120
re-em	1.432	2.004	1.268	1.904	1.204	1.939	1.163	1.952
bag	2.176	1.022	1.990	1.086	1.851	1.173	1.723	1.255
rf	2.146	0.919	2.029	0.974	1.961	1.024	1.899	1.055
merf	1.550	1.393	1.282	1.821	1.219	1.891	1.181	1.925
boost	2.148	1.037	1.873	1.498	1.667	1.663	1.530	1.767
svm	2.238	<b>0.380</b>	2.223	<b>0.344</b>	2.222	<b>0.325</b>	2.219	<b>0.315</b>
svmk	2.282	0.602	2.245	0.487	2.238	0.444	2.234	0.423
nn	2.270	0.499	2.270	0.526	2.279	0.554	2.276	0.554
nntanh	2.311	0.650	2.265	0.513	2.267	0.505	2.268	0.522
<i>(b) Misspecification of the mean</i>								
lme	<b>1.379</b>	1.967	<b>1.206</b>	1.893	<b>1.144</b>	1.927	1.107	1.946
tree	2.249	0.874	2.183	0.948	2.101	1.102	1.988	1.259
re-em	1.435	2.031	1.270	1.919	1.207	1.953	1.166	1.966
bag	2.118	1.064	1.913	1.170	1.747	1.273	1.604	1.382
rf	2.144	0.946	2.023	1.004	1.942	1.049	1.869	1.092
merf	1.548	1.416	1.284	1.829	1.215	1.900	1.178	1.937
boost	2.145	1.059	1.871	1.490	1.664	1.668	1.521	1.770
svm	2.286	<b>0.613</b>	2.267	<b>0.541</b>	2.264	<b>0.518</b>	2.253	0.507
svmk	2.349	0.832	2.320	0.748	2.312	0.716	2.297	0.702
nn	2.548	1.273	2.557	1.284	2.574	1.304	2.570	1.307
nntanh	2.549	1.284	2.528	1.222	2.513	1.178	2.572	1.262

Best/smallest RMSE/TRMSE values are in bold

$$TRMSE = \sqrt{\frac{\sum_{i=1}^K \sum_{j=1}^n (\mu_{ij} - \hat{y}_{ij})^2}{K * n}}$$

According to the formula of mean square error,

$$\begin{aligned} RMSE &= \sqrt{\text{ave}_{x \in \text{test}} (\hat{y} - y)^2} \\ &= \sqrt{\text{ave}_{x \in \text{test}} (\hat{y} - (\mu + \mathbf{b} + \epsilon))^2} \\ &= \sqrt{\text{ave}_{x \in \text{test}} [(\hat{y} - \mu)^2 - 2(\hat{y} - \mu)(\mathbf{b} + \epsilon) + (\mathbf{b} + \epsilon)^2]} \\ &= \sqrt{\text{ave}_{x \in \text{test}} [(\hat{y} - \mu)^2] + \text{ave}_{x \in \text{test}} [-2(\hat{y} - \mu)(\mathbf{b} + \epsilon) + (\mathbf{b} + \epsilon)^2]} \end{aligned}$$

**Table 3** The two-step prediction for different methods in simulated data generated from a mixed-effects model

	Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
<i>(a) Correct specification of the mean</i>						
lme	<b>1.373</b>	1.957	<b>1.205</b>	1.875	<b>1.142</b>	1.911
tree	2.270	0.747	2.201	0.782	2.149	0.940
re-em	1.424	2.002	1.267	1.898	1.200	1.935
bag	2.175	1.028	1.987	1.090	1.841	1.172
rf	2.143	0.922	2.028	0.976	1.955	1.019
merf	1.545	1.388	1.280	1.818	1.213	1.891
boost	2.144	1.045	1.875	1.493	1.657	1.660
svm	2.232	<b>0.380</b>	2.220	<b>0.339</b>	2.222	<b>0.321</b>
svmk	2.278	0.604	2.240	0.482	2.237	0.437
nn	2.265	0.501	2.269	0.529	2.283	0.557
nntanh	2.306	0.652	2.264	0.513	2.271	0.508
<i>(b) Misspecification of the mean</i>						
lme	<b>1.381</b>	1.965	1.204	1.886	<b>1.141</b>	1.919
tree	2.251	0.880	2.185	0.941	2.103	1.077
re-em	1.440	2.028	1.269	1.913	1.205	1.945
bag	2.121	1.066	1.916	1.156	1.750	1.262
rf	2.145	0.949	2.023	0.996	1.935	1.042
merf	1.552	1.416	1.283	1.822	1.216	1.893
boost	2.145	1.060	<b>1.868</b>	1.480	1.648	1.661
svm	2.284	<b>0.615</b>	2.259	<b>0.539</b>	2.248	<b>0.515</b>
svmk	2.348	0.843	2.313	0.750	2.296	0.716
nn	2.549	1.275	2.548	1.284	2.560	1.308
nntanh	2.550	1.286	2.520	1.222	2.490	1.163

Best/smallest RMSE/TRMSE values are in bold

where  $\hat{y}$  is the prediction value,  $y$  is the observed value,  $\mu$  is the true value and  $\epsilon$  is the error. Also,  $\text{ave}_{x \in \text{test}} [(\hat{y} - \mu)^2]$  is the square of TRMSE. If the data are generated from marginal model (i.e.  $\mathbf{b}$  is zero) and error is independent of the observations (i.e.  $\text{ave}_{x \in \text{test}} [-2(\hat{y} - \mu)\epsilon] = 0$ ), the RMSE values would be larger than the TRMSE values, which is consistent with the simulation results from Tables 4 and 5.

### 3.3 Simulation results

The objective of our extensive simulations is to compare the prediction performances among the different methods in the longitudinal data. The parameters in the linear mixed-effects model are estimated with maximum likelihood and they are varied according to the different sizes of training data. Ten-fold cross-validation was used to tune the parameters in tree-based methods. For the tree method, the common and default tree growth is limited to a depth of 31 by the use of integers to

**Table 4** The one-step prediction for different methods in simulated data which correlated with the exchangeable and AR(1) structure

	Second observation		Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
<i>(a) Correct specification of the mean</i>								
<i>(i) EXC</i>								
lme	<b>0.949</b>	0.876	<b>0.821</b>	0.577	<b>0.790</b>	0.611	<b>0.775</b>	0.631
tree	1.024	0.248	1.015	0.217	1.009	0.204	1.006	0.195
re-em	0.965	0.896	0.834	0.596	0.804	0.627	0.790	0.646
bag	1.145	0.572	1.146	0.582	1.145	0.586	1.147	0.586
rf	1.020	0.242	1.008	0.194	1.002	0.173	1.000	0.161
merf	0.884	0.537	0.828	0.588	0.797	0.618	0.781	0.637
boost	1.070	0.403	1.040	0.316	1.024	0.275	1.018	0.250
svm	1.011	<b>0.196</b>	1.002	<b>0.149</b>	0.997	<b>0.134</b>	0.995	<b>0.123</b>
svmk	1.023	0.250	1.008	0.189	1.002	0.166	0.999	0.151
nn	1.123	0.526	1.126	0.532	1.126	0.537	1.124	0.531
nntanh	1.142	0.565	1.105	0.485	1.067	0.399	1.036	0.309
<i>(ii) AR(1)</i>								
lme	0.950	0.881	<b>0.920</b>	0.579	<b>0.939</b>	0.533	<b>0.961</b>	0.493
tree	1.024	0.260	1.021	0.222	1.010	0.206	1.014	0.191
re-em	0.965	0.900	0.933	0.597	0.950	0.552	0.972	0.514
bag	1.144	0.575	1.152	0.583	1.148	0.587	1.157	0.586
rf	1.018	<b>0.246</b>	1.013	0.195	1.003	0.173	1.009	0.156
merf	<b>0.882</b>	0.540	0.927	0.590	0.946	0.551	0.968	0.512
boost	1.071	0.407	1.043	0.317	1.026	0.276	1.027	0.248
svm	1.009	0.193	1.006	<b>0.151</b>	0.999	<b>0.131</b>	1.004	<b>0.118</b>
svmk	1.022	0.251	1.013	0.192	1.003	0.165	1.008	0.148
nn	1.122	0.528	1.129	0.532	1.127	0.537	1.132	0.531
nntanh	1.140	0.565	1.110	0.487	1.070	0.402	1.046	0.302
<i>(b) Misspecification of the mean</i>								
<i>(i) EXC</i>								
lme	2.203	2.169	1.775	1.558	1.755	1.564	1.747	1.568
tree	1.238	0.744	1.156	0.602	1.139	0.570	1.130	0.552
re-em	1.306	1.196	<b>1.028</b>	0.773	<b>0.939</b>	0.759	<b>0.886</b>	0.753
bag	1.156	0.601	1.142	0.581	1.145	0.578	1.145	0.578
rf	1.252	0.765	1.233	0.740	1.229	0.732	1.230	0.726
merf	1.393	1.200	1.203	0.967	1.159	0.961	1.131	0.960
boost	1.158	0.603	1.061	<b>0.386</b>	1.039	0.319	1.028	<b>0.282</b>
svm	1.853	1.562	1.829	1.541	1.821	1.533	1.825	1.528
svmk	<b>1.147</b>	<b>0.551</b>	1.104	0.465	1.089	<b>0.424</b>	1.078	0.402
nn	1.880	1.592	1.876	1.594	1.876	1.594	1.873	1.583
nntanh	1.925	1.645	1.857	1.572	1.829	1.543	1.827	1.531
<i>(ii) AR(1)</i>								
lme	2.196	2.169	1.792	1.559	1.793	1.550	1.789	1.541

**Table 4** (continued)

	Second observation		Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
tree	1.239	0.745	1.155	0.605	1.147	0.566	1.137	0.550
re-em	1.300	1.192	1.090	0.782	1.074	0.714	1.069	0.670
bag	1.158	0.602	1.143	0.582	1.152	0.576	1.148	0.579
rf	1.250	0.764	1.232	0.738	1.233	0.729	1.228	0.726
merf	1.386	1.198	1.250	0.969	1.240	0.920	1.233	0.886
boost	1.160	0.598	<b>1.058</b>	<b>0.386</b>	<b>1.046</b>	<b>0.317</b>	<b>1.033</b>	<b>0.280</b>
svm	1.849	1.564	1.830	1.542	1.824	1.531	1.817	1.528
svmk	<b>1.148</b>	<b>0.555</b>	1.106	0.477	1.096	0.432	1.081	0.410
nn	1.878	1.596	1.876	1.594	1.876	1.593	1.865	1.584
nntanh	1.921	1.648	1.855	1.570	1.828	1.537	1.822	1.533

Best/smallest RMSE/TRMSE values are in bold

label nodes. Therefore, the range of tree maximum depth is from 20 to 40 for tuning. The random effect in RE-EM trees is the grouping variable (subject). We used 500 trees in total in bagging and random forest method. The number of variables randomly sampled as candidates at each split is 2 and 1 in bagging and random forest, respectively because we have two covariates  $x_{ij}$  and  $t_i$ . Otherwise, the number of trees ranged from 200 to 5000 for tuning in the boosting method. In the SVM, the cost of constraints violation is 1 and the epsilon in the insensitive-loss function is 0.1 ( $C = 1$  and  $\epsilon = 0.1$  in Eq. 3). The degree of polynomial kernel in SVM is 3. For the parameters of neural network, there are one layer and the hidden neurons in each layer is 2. The threshold for the partial derivatives of the error function as stopping criteria is 0.1.

Figures 2, 3 and 4 presents the prediction results of future observation, new object and future new observation respectively when the fixed effects is specified correctly. Meanwhile, the boxplots of prediction results of future observation, new object and future new observation respectively when the fixed effects is misspecified can be found in Figs. 5, 6 and 7. We can see that the linear mixed model performed the best when the fixed effects/mean function is specified correctly in terms of predicting future observations and future new observations. However, the support vector machine and neural network methods have better performance when we need to predict the observations from new objects. It is expected that the support vector regression with polynomial kernel ('svmk') and neural network with hyperbolic tangent activation function ('nntanh') would also have better performance if the mean function is misspecified. However, it seems that we should be careful to choose the nonlinear function according to the data structure which is the quadratic in this case. The RE-EM trees and mixed effects random forest (MERF) performed better when the mean function is misspecified in terms of predicting future observations and future new observations. It is interesting to find that RE-EM trees and MERF performed worse than trees and RF in terms of predicting new objects (see Fig. 6), which means that mixed effects machine learning needs to be used in caution when

**Table 5** The two-step prediction for different methods in simulated data which correlated with the exchangeable and AR(1) structure

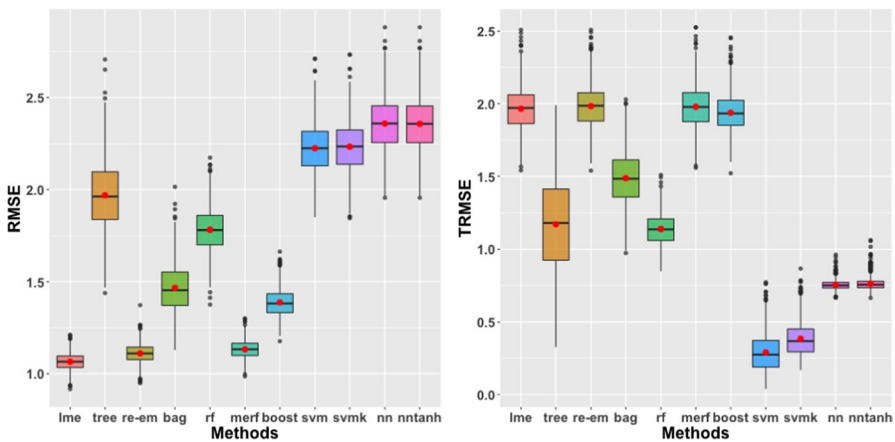
	Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
<i>(a) Correct specification of the mean</i>						
(i) EXC						
lme	0.949	0.880	<b>0.820</b>	0.576	<b>0.790</b>	0.611
tree	1.023	0.254	1.014	0.220	1.008	0.208
re-em	0.965	0.901	0.833	0.595	0.803	0.627
bag	1.145	0.577	1.149	0.583	1.145	0.587
rf	1.019	<b>0.244</b>	1.008	0.198	1.001	0.178
merf	<b>0.882</b>	0.539	0.827	0.587	0.796	0.618
boost	1.068	0.403	1.040	0.321	1.023	0.278
svm	1.010	0.195	1.002	<b>0.156</b>	0.997	<b>0.139</b>
svmk	1.023	0.253	1.009	0.194	1.002	0.171
nn	1.121	0.526	1.124	0.532	1.125	0.536
nntanh	1.140	0.566	1.103	0.484	1.065	0.400
(ii) AR(1)						
lme	1.158	0.877	1.046	0.577	1.041	0.532
tree	1.030	0.255	1.023	0.222	1.015	0.206
re-em	1.173	0.898	1.056	0.596	1.052	0.552
bag	1.147	0.572	1.159	0.585	1.156	0.585
rf	1.024	0.243	1.017	0.198	1.009	0.172
merf	<b>1.018</b>	0.537	1.052	0.589	1.050	0.550
boost	1.074	0.404	1.049	0.320	1.032	0.276
svm	1.015	<b>0.194</b>	<b>1.009</b>	<b>0.151</b>	<b>1.003</b>	<b>0.133</b>
svmk	1.027	0.249	1.018	0.195	1.008	0.166
nn	1.127	0.527	1.130	0.533	1.129	0.536
nntanh	1.147	0.565	1.109	0.486	1.072	0.396
<i>(b) Misspecification of the mean</i>						
(i) EXC						
lme	2.196	2.170	1.777	1.565	1.752	1.564
tree	1.239	0.749	1.154	0.601	1.139	0.569
re-em	1.309	1.204	<b>1.024</b>	0.777	<b>0.938</b>	0.756
bag	1.153	0.598	1.143	0.578	1.144	0.579
rf	1.246	0.763	1.232	0.741	1.225	0.731
merf	1.385	1.198	1.203	0.973	1.155	0.960
boost	1.157	0.602	1.058	<b>0.384</b>	1.038	<b>0.317</b>
svm	1.854	1.573	1.830	1.544	1.814	1.528
svmk	<b>1.147</b>	<b>0.554</b>	1.104	0.472	1.084	0.431
nn	1.877	1.597	1.877	1.599	1.867	1.589
nntanh	1.921	1.649	1.856	1.575	1.821	1.536
(ii) AR(1)						
lme	2.306	2.174	1.824	1.560	1.813	1.540
tree	1.239	0.745	1.163	0.603	1.145	0.566



**Table 5** (continued)

	Third observation		Fourth observation		Fifth observation	
	RMSE	TRMSE	RMSE	TRMSE	RMSE	TRMSE
re-em	1.440	1.200	1.185	0.776	1.152	0.713
bag	1.164	0.610	1.153	0.583	1.152	0.577
rf	1.259	0.771	1.241	0.742	1.233	0.728
merf	1.486	1.206	1.325	0.968	1.294	0.911
boost	1.163	0.608	<b>1.069</b>	<b>0.389</b>	<b>1.045</b>	<b>0.318</b>
svm	1.861	1.572	1.834	1.542	1.824	1.527
svmk	<b>1.153</b>	<b>0.562</b>	1.108	0.473	1.097	0.436
nn	1.889	1.603	1.881	1.599	1.876	1.588
nntanh	1.930	1.652	1.861	1.575	1.828	1.531

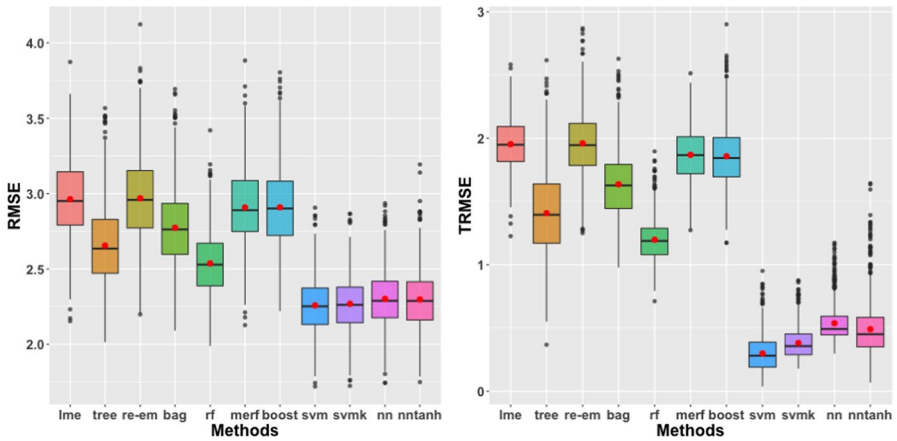
Best/smallest RMSE/TRMSE values are in bold



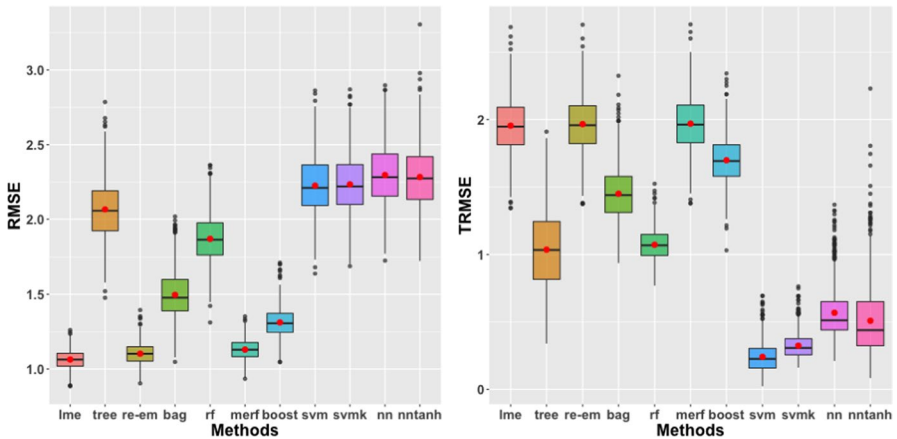
**Fig. 2** The prediction results of future observation. *RMSE* root mean square error, *TRMSE* true root mean square error, *lme* linear mixed-effects model, *tree* decision tree method, *re-em* RE-EM trees with random intercept, *bag* bagging method, *rf* random forest method, *merf* mixed-effects random forest, *boost* boosting method, *svm* support-vector regression with linear kernel, *svmk* support-vector regression with polynomial kernel, *nn* neural network method with logistic activation function, *nntanh* neural network method with hyperbolic tangent activation function

predicting unseen data. The TRMSE values that measured the differences between the predictions and mean values without random effects and errors. According to the TRMSE values, the support-vector machine with a linear kernel had the best performance whether the mean function is correctly specified or misspecified.

The performance of different methods in simulated data generated from the marginal model with exchangeable and AR(1) correlation structure is presented in Figs. 8, 9 and 10. In Figs. 8 and 10, because the linear mixed model is the true model when the correlation structure is exchangeable, it is not a surprise to see that the linear mixed model performed the best when the mean model is specified correctly and



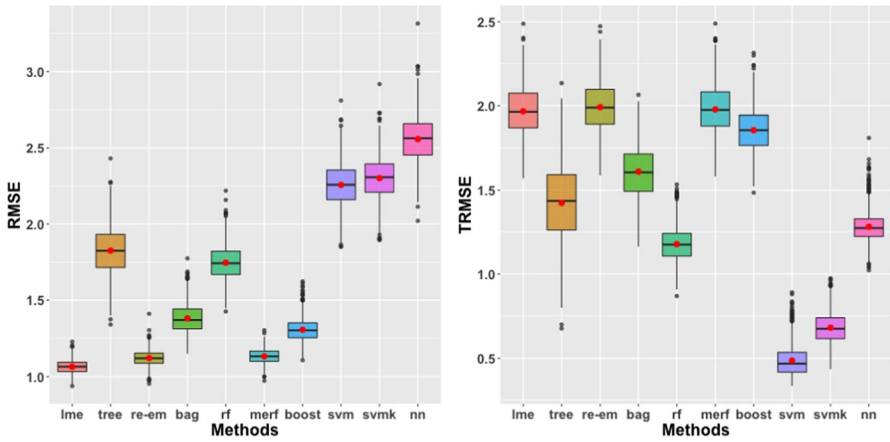
**Fig. 3** The prediction results of new objects. The notations are the same as Fig. 2



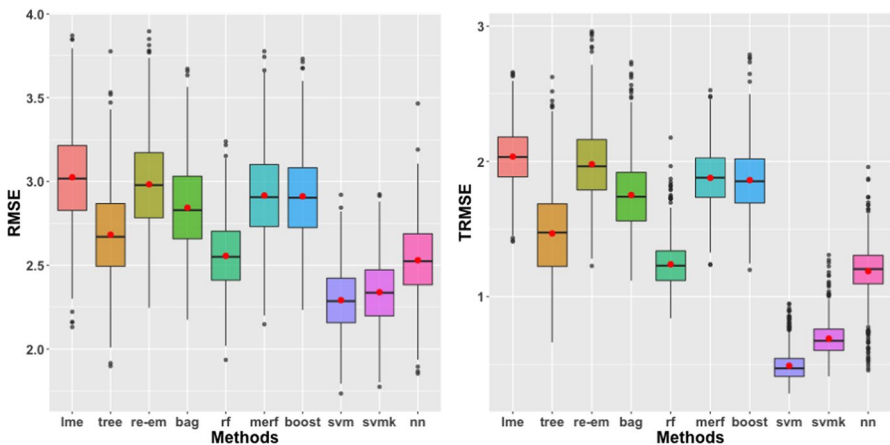
**Fig. 4** The prediction results of future new observations. The notations are the same as Fig. 2

in terms of predicting future observations and future new observations. The support-vector machine with a linear kernel and the neural had good performances when predicting the observations from new object (see Fig. 9). However, when the correlation structure is AR(1), which means that the random effect component is misspecified, the random forest had better performance. RE-EM trees and MERF do not show an advantage because these two methods were not designed for this case of correlation structure misspecification.

If the mean function is misspecified, the RE-EM trees and support-vector machine with a polynomial kernel had the advantages in terms of predicting future observations and future new observations regardless of whether the random effect component is misspecified or not (see Figs. 11, 12 and 13). It is not a surprise to see that the support-vector machine with a polynomial kernel had smaller RMSE values



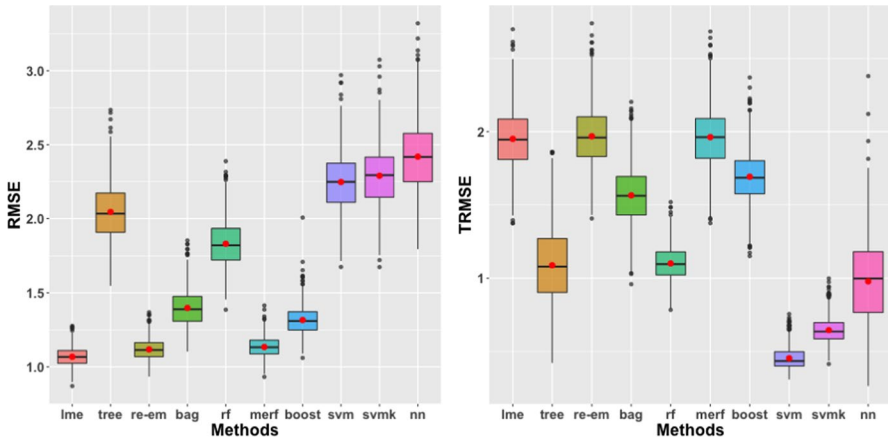
**Fig. 5** The prediction results of future observation under fixed effects misspecification. The notations are the same as Fig. 2. The results of ‘nntanh’ is omitted in the plots because the range of RMSE and TRMSE is too large (the maximum of RMSE and TRMSE is 38.79 and 38.68, respectively)



**Fig. 6** The prediction results of new objects under fixed effects misspecification. The notations are the same as Fig. 2. The results of ‘nntanh’ is omitted in the plots because the range of RMSE and TRMSE is too large (the maximum of RMSE and TRMSE is 327.38 and 327.14, respectively)

than when a linear kernel is used if the mean function is misspecified. The results according to TRMSE values are a slightly different from the conclusions according to RMSE values. The boost method had the best performance according to the TRMSE values.

The results from the one-step and two-step predictions are presented in Tables 2, 3, 4 and 5, respectively. Regardless of how the correlated data was generated, the linear mixed model had the best performance both in the one-step and two-step predictions when the mean function is correct. It is noted that in the simulated



**Fig. 7** The prediction results of future new observations under fixed effects misspecification. The notations are the same as Fig. 2. The results of ‘nntanh’ is omitted in the plots because the range of RMSE and TRMSE is too large (the maximum of RMSE and TRMSE is 4909.26 and 4908.88, respectively)

data generated from the mixed-effects model, support vector machine had better performance when the mean function is misspecified according to TRMSE values. We can also conclude that the RE-EM trees and support-vector machine with a polynomial kernel performed well when the mean function is misspecified. The performances between the one-step and two-step predictions are different when the mean function is specified correctly while the correlation structure is different, see Table 4(a)(ii) and Table 5(a)(ii). In the one-step prediction, the linear mixed model is still comparable but not for the two-step prediction. The support vector machine method had the best performance when the random effect component is misspecified in the two-step prediction.

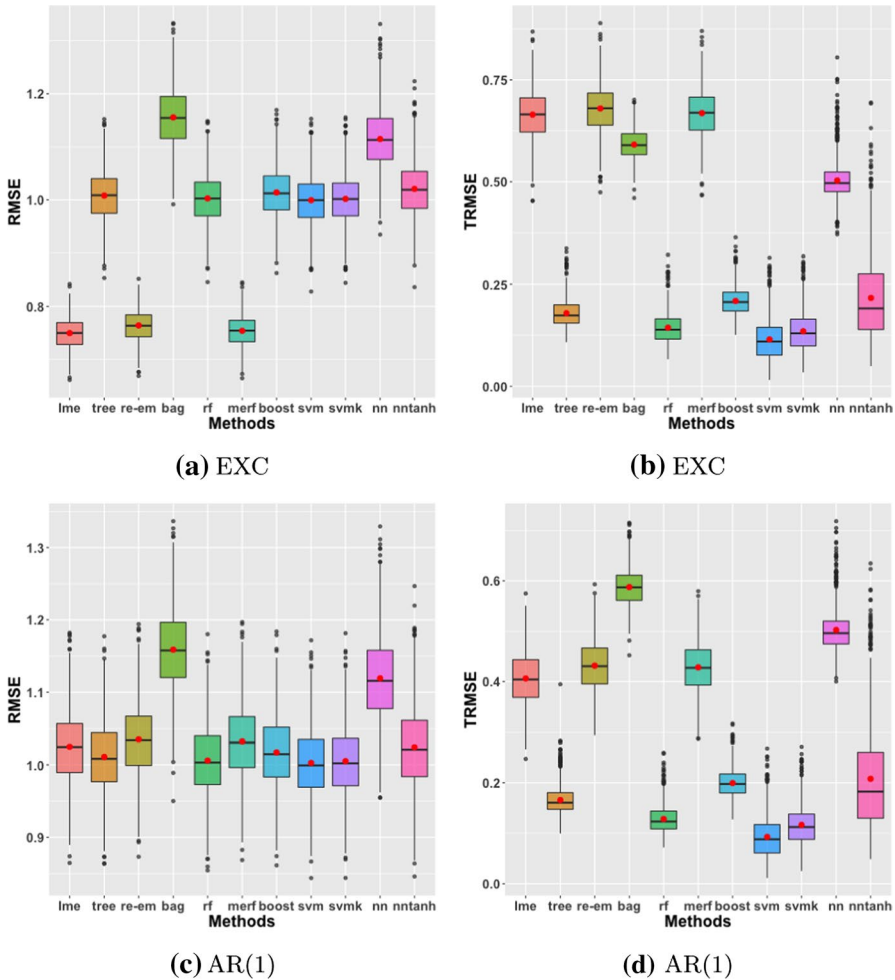
## 4 Application to real data

Two real data sets are analysed using these different methods in this section.

### 4.1 Case study 1: milk protein data

In this data set, milk was collected weekly from 79 Australian cows and analyses for its protein content. There are three diets: 25 cows received a barley diet, 27 cows a mixture of barley and lupins, and 27 cows a diet of lupins only. The observation period of each cow is not necessarily the same and each cow is observed for between 12 weeks and 19 weeks (Fig. 14). There are 1337 observations of protein in total.

It appears from the Fig. 14 that barley gives higher values than the mixture, which in turn have higher values than lupins alone. The mean response profiles are approximately parallel, showing an initial sharp decline associated with a settling-in



**Fig. 8** The results of future observations for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure. The notations are the same as Fig. 2

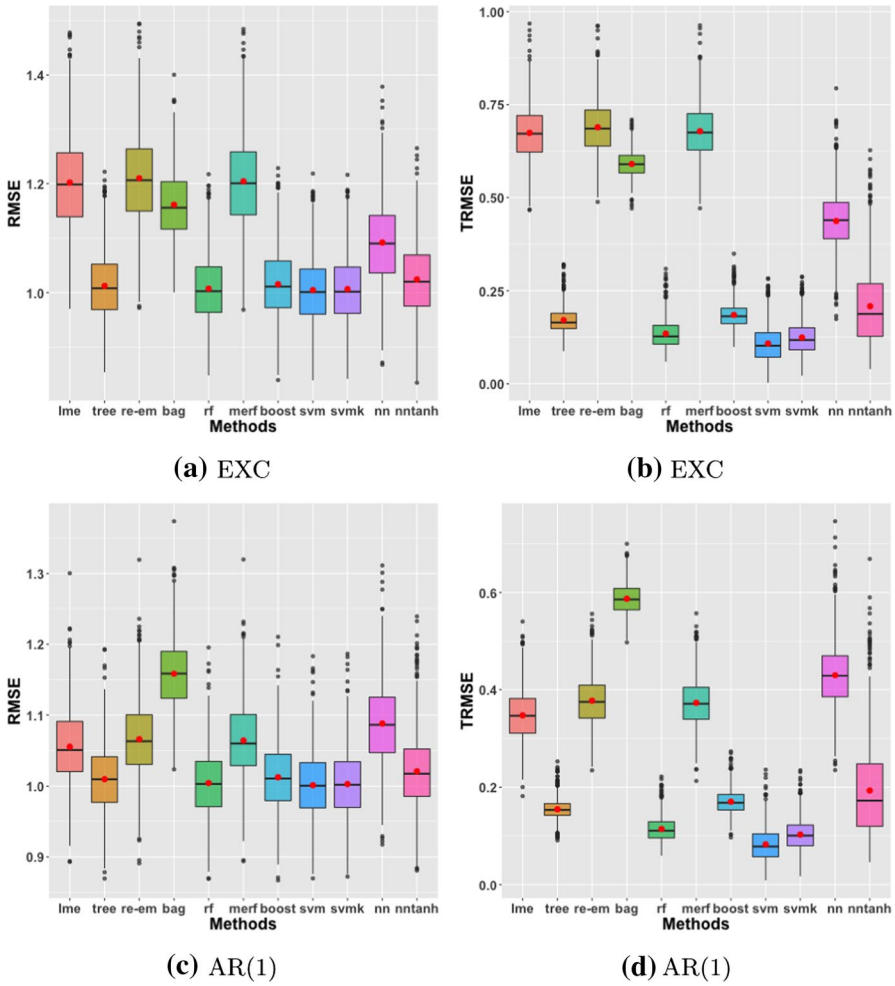
period, followed by an approximately constant mean response through the following period and a slow rise towards the end.

Diggle et al. (2002) used the following mean response profiles model:

$$\mu_i = \begin{cases} \beta_{0i} + \beta_1 t & \text{if } t \leq 3, \\ \beta_{0i} + 3\beta_1 + \beta_2(t - 3) + \beta_3(t - 3)^2 & \text{if } t > 3, \end{cases}$$

where  $i = 1, 2, 3$  denotes treatment group with an exponential correlation function  $\text{Cov}(\epsilon_j, \epsilon_k) = \sigma^2 \exp(-\phi|t_j - t_k|)$ . The covariates include time and quadratic of time.

However, the quadratic term is not significant and the breakpoint is not necessarily to be a integer. According to the mean square error, the breakpoint we chose for



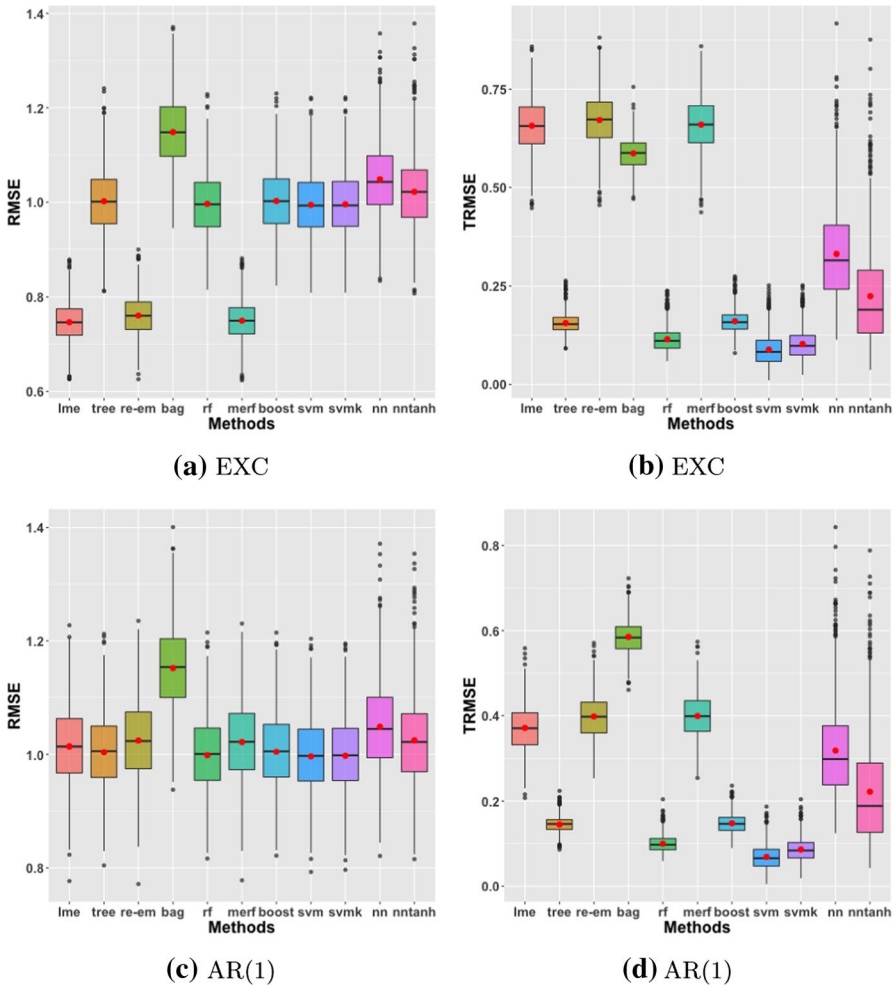
**Fig. 9** The results of new objects for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure. The notations are the same as Fig. 2

this milk protein data is 2.6. So we use the piecewise mixed model with the mean response profiles model as follows:

$$y_i = \begin{cases} \beta_{0i} + \beta_1 t + b_{i1} + tb_{i2} + \epsilon_i & \text{if } t \leq 2.6, \\ \beta_{0i} + 2.6\beta_1 + \beta_2(t - 2.6) + b_{i1} + tb_{i2} + \epsilon_i & \text{if } t > 2.6, \end{cases}$$

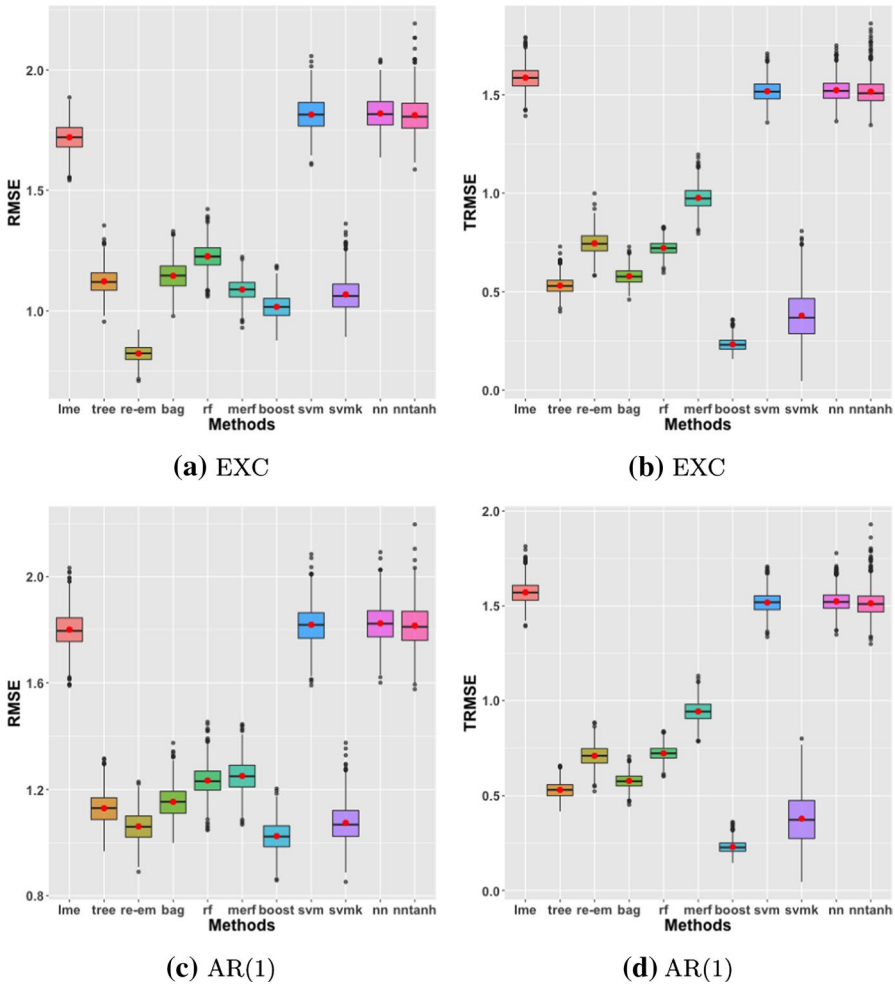
where  $i = 1, 2, 3$  denotes treatment group and with the different mean function

$$\mu_i = \begin{cases} \beta_{0i} + \beta_1 t & \text{if } t \leq 2.6, \\ \beta_{0i} + 2.6\beta_1 + \beta_2(t - 2.6) & \text{if } t > 2.6. \end{cases}$$



**Fig. 10** The results of future new observations for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure. The notations are the same as Fig. 2

The  $b_{i1}$  and  $b_{i2}$  are the corresponding random effects for different groups. The estimated parameters of  $\beta_{0i}, \beta_1, \beta_2, b_{i1}$  and  $b_{i2}$  ( $i = 1, 2, 3$ ) varied a bit according to the different size of training data in piecewise linear mixed-effects model. We focus on the predictive performance of the different models and the estimation of the parameters is not reported here. The one-step prediction and two-step prediction results are presented in Table 6(a). We can see that the piecewise linear mixed model has the best performance in one-step prediction. RE-EM trees also has advantages. Tree-based methods have smaller RMSE values than support-vector machine and neural network methods.

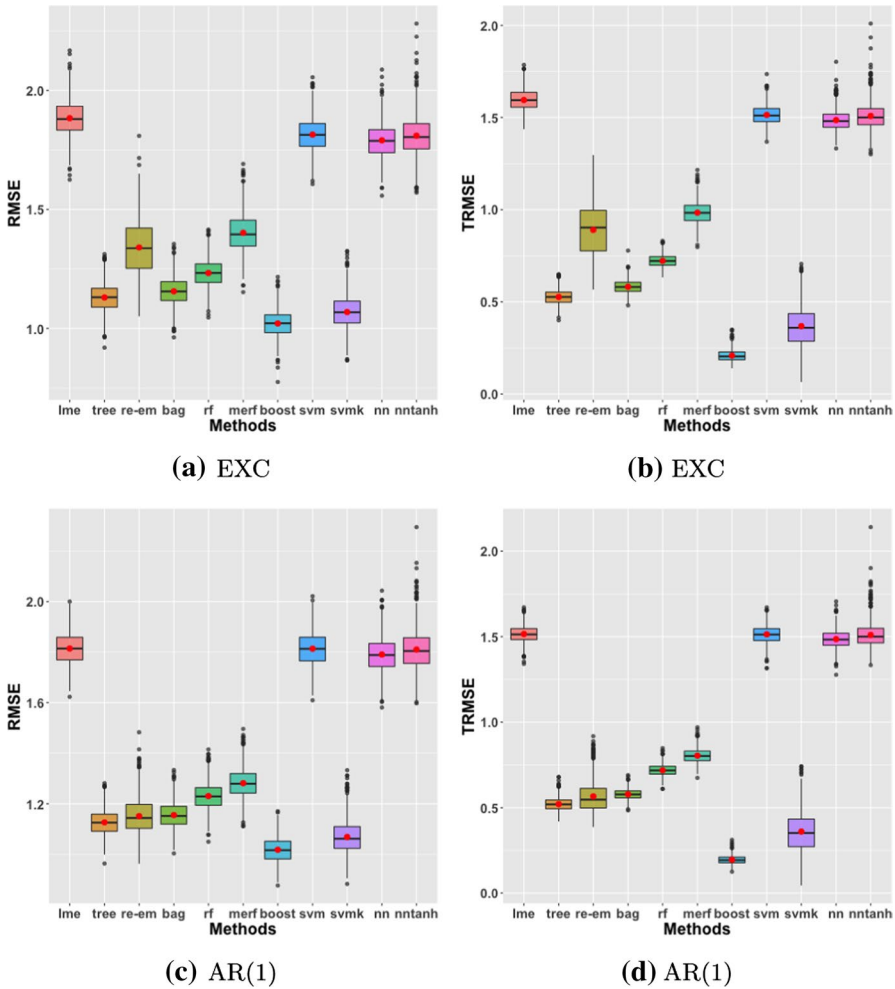


**Fig. 11** The results of future observations for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure under fixed effects misspecification. The notations are the same as Fig. 2

### 4.2 Case study 2: wages data

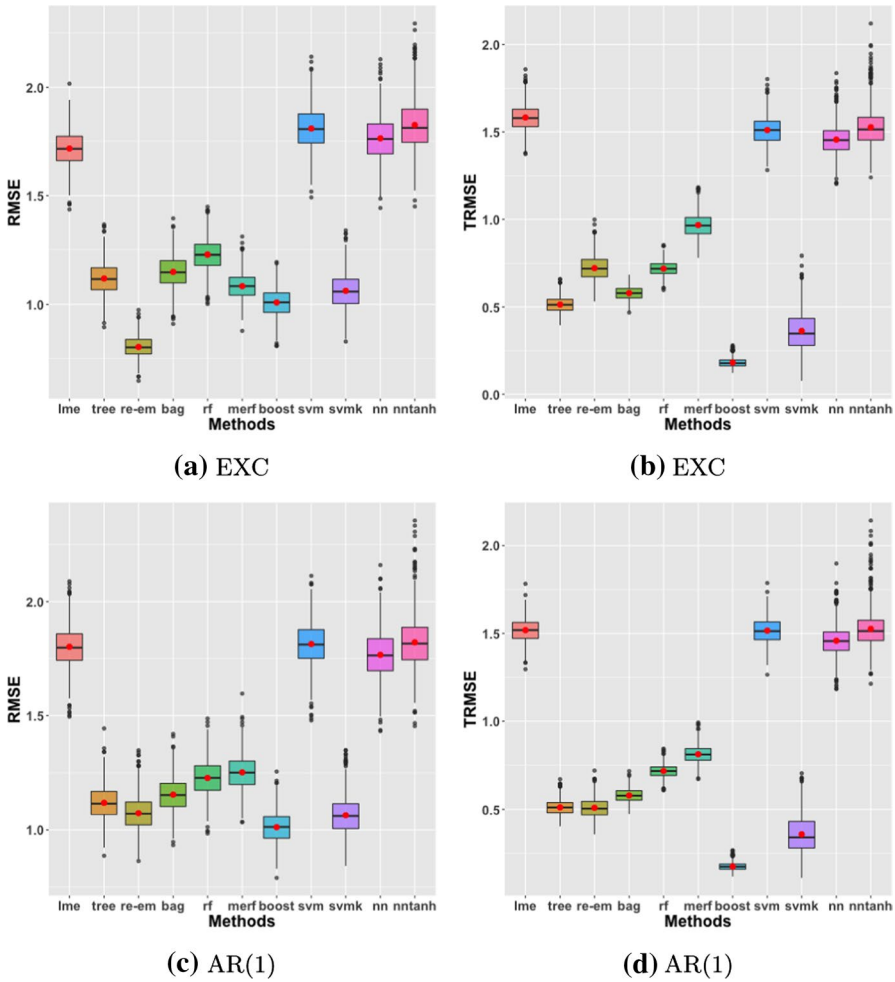
Wages data came from the National Longitudinal Survey of Youth (NLSY), which was previously studied by Singer and Willett (2003), Eo and Cho (2014) and Fu and Simonoff (2015). The data has the information of 888 individuals' hourly wage. Each individual has the different observation times, ranged from 1 to 13. There are 6402 observations in total. In the linear mixed-effects model, the log of individual's hourly wage (*logwage*) is the response variable, the covariates include *exper*, *hgc* and *race*. The individual's races are White, Black and Hispanic. The variable *hgc* means the highest grade completed by





**Fig. 12** The results of new objects for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure under fixed effects misspecification. The notations are the same as Fig. 2

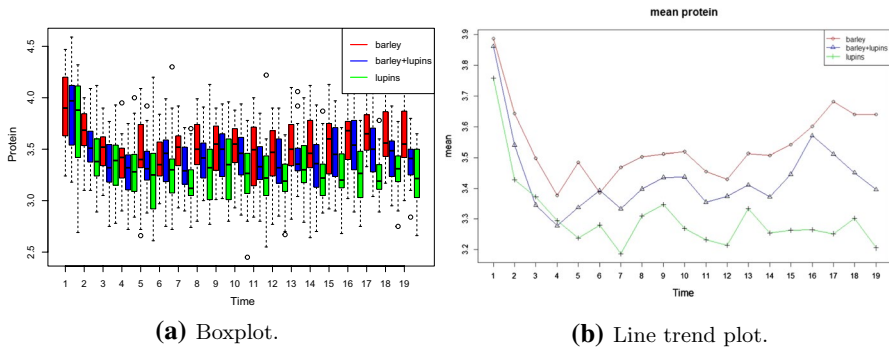
the individual. Figure 15 present the plots of the time variable (*exper*, which is the duration of the working experience) and the log of wages at different *race* and *hgc*. The random intercept is included to indicate the differences between individuals. We used the eight cross-validation method to compare the prediction performances between statistical models and machine learning methods. According to Table 6(b), RE-EM methods has the smallest RMSE. Tree-based methods and support-vector machine have similar results while the average RMSE values of LME and neural network are close in this case.



**Fig. 13** The results of future new observations for the simulated data generated from marginal model with exchangeable correlation structure and AR(1) correlation structure under fixed effects misspecification. The notations are the same as Fig. 2

### 5 Conclusions and discussion

We have presented the performances of the statistical models and six machine learning methods and two mixed effects machine learning methods for the longitudinal data analysis. The parameters in the machine learning methods we used in the work are indicated and justified. Overall, the simulation results showed that the linear mixed-effects model is comparable with the various machine learning methods when the models are correctly specified, included the fixed effects and random effects because we knew the truth model in the simulations. The performances under the scenarios of the different mean function and the different



**Fig. 14** The mean of protein for three different kinds of diet in milk protein data

correlation structures (exchangeable and AR(1)) are compared. Otherwise, even with the milk dataset (a real world dataset), the statistical model (especially, the piecewise linear mixed model) still performed better than the machine learning methods. This means that the piecewise linear mixed model provided an adequate fit to the original data. It can also be concluded that the model diagnostics are very important before making decisions regarding performance.

There are few references about how to measure the predictive power of methods in longitudinal data. The prediction accuracy according to a cross-validation method are not reasonable because longitudinal data are always sequential. In this work, we used one-step and two-step prediction along with future observation, new object and future new observation prediction. The performances of all kinds of methods are demonstrated comprehensively. In addition, we also presented the differences between RMSE and TRMSE values in the predictions. It is not surprising to see that the TRMSE values are smaller than the RMSE values in data generated from marginal model because we measured that differences between the predictions and true values without noise. However, this is not always true, which can be found from the predictions in the data generated from a mixed-effects model.

There are still some limitations in this study. The predictions between the different methods are discussed rather than the parameter estimates and inferences in the longitudinal data. Misspecified models, including the mean function are considered in this work. Wang and Lin (2005) also investigated the effects of variance function and correlation structure misspecification in the analysis of longitudinal data. In this work, we only investigated the popular exchangeable and AR(1) correlation structures that are appropriate for equally spaced (in time) longitudinal data. However, unequally spaced observations and time-dependent correlated errors deserves more attention by researchers (Nunez-Anton and Woodworth 1994). It would be of great interest to evaluate machine learning performance in these settings. There are also other modified methods that combine mixed-effects models and tree methods (Fu and Simonoff 2015; Loh and Zheng 2013; Eo and Cho 2014) that deserve further

**Table 6** The RMSE values for different methods in Milk data and Wages data

	plme	diggle	re-em	tree	bag	rf	boost	svm	nn
<i>(a) Milk data</i>									
One step prediction									
Week 8	<b>0.211</b>	0.303	0.227	0.306	0.311	0.303	0.314	0.368	0.324
Week 9	<b>0.226</b>	0.292	0.237	0.301	0.291	0.293	0.292	0.360	0.310
Week 10	<b>0.247</b>	0.291	0.264	0.282	0.275	0.275	0.276	0.300	0.294
Week 11	<b>0.291</b>	0.322	0.300	0.308	0.307	0.307	0.306	0.304	0.317
Week 12	<b>0.273</b>	0.281	0.284	0.289	0.280	0.283	0.280	0.281	0.295
Week 13	<b>0.223</b>	0.293	0.248	0.286	0.296	0.287	0.296	0.306	0.297
Week 14	<b>0.230</b>	0.320	0.296	0.315	0.318	0.316	0.320	0.320	0.315
Week 15	<b>0.282</b>	0.317	0.304	0.316	0.315	0.315	0.315	0.330	0.315
Week 16	0.342	0.363	<b>0.339</b>	0.353	0.340	0.347	0.341	0.372	0.387
Week 17	<b>0.264</b>	0.309	0.272	0.307	0.284	0.291	0.284	0.319	0.294
Week 18	<b>0.234</b>	0.302	0.260	0.301	0.293	0.290	0.294	0.307	0.292
Week 19	<b>0.263</b>	0.315	<b>0.263</b>	0.314	0.308	0.309	0.305	0.314	0.310
Two step prediction									
Week 9	0.259	0.294	<b>0.247</b>	0.303	0.311	0.299	0.309	0.419	0.341
Week 10	0.276	0.312	<b>0.270</b>	0.284	0.273	0.275	0.275	0.359	0.272
Week 11	0.333	0.383	0.312	<b>0.307</b>	0.314	0.310	0.315	0.315	0.315
Week 12	0.329	0.321	0.298	0.290	0.288	0.288	0.285	<b>0.284</b>	0.295
Week 13	<b>0.266</b>	0.289	0.261	0.287	0.292	0.285	0.291	0.309	0.284
Week 14	<b>0.265</b>	0.319	0.311	0.316	0.318	0.315	0.318	0.328	0.331
Week 15	<b>0.311</b>	0.317	0.314	0.316	0.316	0.316	0.316	0.333	0.313
Week 16	0.377	0.355	0.351	0.354	0.355	0.353	<b>0.348</b>	0.384	0.381
Week 17	0.314	0.324	<b>0.285</b>	0.309	0.293	0.302	0.296	0.334	0.296
Week 18	<b>0.268</b>	0.304	0.270	0.303	0.299	0.295	0.294	0.315	0.332
Week 19	0.292	0.315	<b>0.272</b>	0.315	0.311	0.311	0.310	0.315	0.319
	lme	re-em	tree	bag	rf	boost	svm	svmk	nn

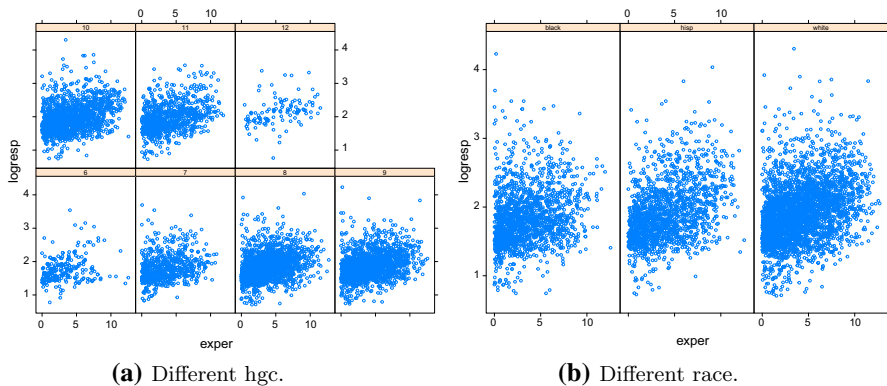
*(b) Wages data*

RMSE value	0.399	<b>0.314</b>	0.407	0.439	0.402	0.432	0.401	0.401	0.398
------------	-------	--------------	-------	-------	-------	-------	-------	-------	-------

Best/smallest RMSE/TRMSE values are in bold

*plme* Piecewise Linear Mixed-Effects Model, *diggle* Diggle's Model, *re-em* RE-EM trees with random intercept, *tree* Decision Tree Method, *bag* Bagging Method, *rf* Random Forest Method, *boost* Boosting Method, *svm* Support-Vector Regression Method, *nn* Neural Network Method

examination. An extended comparison with more recently developed machine learning methods, such as deep learning, would be of interest.



**Fig. 15** The wages data

**Acknowledgements** This work is in part supported by Australian Research Council (ARC) Discovery Project (DP160104292), the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS), under grant number CE140100049, Guangdong Basic and Applied Basic Research Foundation (2020A1515011580), and Guangdong Provincial key platforms and major scientific research projects of Guang-dong universities (2018GKTSCX010).

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Albert PS (2012) A linear mixed model for predicting a binary event from longitudinal data under random effects misspecification. *Stat Med* 31(2):143–154
- Berger M, Tutz G (2018) Tree-structured clustering in fixed effects models. *J Comput Graph Stat* 27(2):380–392
- Berrocal VJ, Guan Y, Muyskens A, Wang H, Reich BJ, Mulholland JA, Chang HH (2020) A comparison of statistical and machine learning methods for creating national daily maps of ambient PM2.5 concentration. *Atmosph Environ* 222:117130
- Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and regression trees*. Wadsworth, Monterey
- Breiman L (2001) Random forests. *Mach Learn* 45:5–32
- Crane-Droesch A (2017) Semiparametric panel data models using neural networks. [arXiv:1702.06512](https://arxiv.org/abs/1702.06512)

- Diggle PJ, Heagerty PJ, Liang K-Y, Zeger SL (2002) Analysis of longitudinal data. Oxford University Press, New York
- Drikvandi R, Verbeke G, Molenberghs G (2017) Diagnosing misspecification of the random-effects distribution in mixed models. *Biometrics* 73(1):63–71
- Eo S-H, Cho H (2014) Tree-structured mixed-effects regression modeling for longitudinal data. *J Comput Graph Stat* 23:740–760
- Fu W, Simonoff JS (2015) Unbiased regression trees for longitudinal and clustered data. *Comput Stat Data Anal* 88:53–74
- Fritsch S, Guenther F, Wright MN (2019) neuralnet: training of Neural Networks. R package version 1.44.2. <https://CRAN.R-project.org/package=neuralnet>
- Greenwell B, Boehmke B, Cunningham J, GBM Developers (2019) gbm: Generalized Boosted Regression Models. R package version 2.1.5. <https://CRAN.R-project.org/package=gbm>
- Grilli L, Rampichini C (2015) Specification of random effects in multilevel models: a review. *Qual Quant* 49(3):967–976
- Hajjem A, Bellavance F, Larocque D (2011) Mixed effects regression trees for clustered data. *Stat Prob Lett* 81(4):451–459
- Hajjem A, Bellavance F, Larocque D (2014) Mixed-effects random forest for clustered data. *J Stat Comput Simul* 84:1313–1328
- Hajjem A, Bellavance F, Larocque D (2017) Generalized mixed effects regression trees. *Stat Prob Lett* 126:114–118
- Hui FK, Müller S, Welsh AH (2021) Random effects misspecification can have severe consequences for random effects inference in linear mixed models. *Int Stat Rev* 89(1):186–206
- James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning. Springer, Heidelberg
- Kohli N, Sullivan AL, Sadeh S, Zopluoglu C (2015) Longitudinal mathematics development of students with learning disabilities and students without disabilities: a comparison of linear, quadratic, and piecewise linear mixed effects models. *J Sch Psychol* 53(2):105–120
- Kohli N, Peralta Y, Zopluoglu C, Davison ML (2018) A note on estimating single-class piecewise mixed-effects models with unknown change points. *Int J Behav Dev Method Meas Sect* 42:518–524
- Kundu MG, Harezlak J (2019) Regression trees for longitudinal data with baseline covariates. *Biostat Epidemiol* 3(1):1–22
- Laird NM, Ware JH (1982) Random-effects models for longitudinal data. *Biometrics* 38:963–974
- Laird N, Lange N, Stram D (1987) Maximum likelihood computations with repeated measures: application of the EM algorithm. *J Am Stat Assoc* 82:97–105
- Li H, Wu X (2015) Compare machine learning methods and linear mixed models with random effects of longitudinal data prediction. *Hans J Data Min* 5:39–45
- Liaw A, Wiener M (2002) Classification and regression by random forest. *R News* 2(3):18–22
- Lindstrom MJ, Bates DM (1988) Newton–Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. *J Am Stat Assoc* 83:1014–1022
- Loh W-Y, Zheng W (2013) Regression trees for longitudinal and multiresponse data. *Ann Appl Stat* 7:495–522
- Louis C (2020) LongituRF: random forests for longitudinal data. R package version 0.9. <https://CRAN.R-project.org/package=LongituRF>
- Mangino, Anthony A, Finch, WH (2021) Prediction with mixed effects models: a Monte Carlo simulation study. *TEducational and Psychological Measurement* 0013164421992818
- McCulloch CE, Neuhaus JM (2011a) Prediction of random effects in linear and generalized linear models under model misspecification. *Biometrics* 67(1):270–279
- McCulloch CE, Neuhaus JM (2011b) Misspecifying the shape of a random effects distribution: why getting it wrong may not matter. *Stat Sci* 26(3):388–402
- Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2019) e1071: Misc functions of the department of statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-3. <https://CRAN.R-project.org/package=e1071>
- Ngufer C, Houten HV, Caffo BS, Shah ND, McCoy RG (2019) Mixed effect machine learning: a framework for predicting longitudinal change in hemoglobin A1c. *J Biomed Inform* 89:56–67
- Nunez-Anton V, Woodworth GG (1994) Analysis of longitudinal data with unequally spaced observations and time-dependent correlated errors. *Biometrics* 44:5–6

- Pellagatti M, Masci C, Ieva F, Paganoni AM (2021) Generalized mixed-effects random forest: a flexible approach to predict university student dropout. *Stat Anal Data Min ASA Data Sci J* 14(3):241–257
- Pinheiro JC, Bates DM (2000) *Mixed-effects models in S and S-PLUS*. Springer, New York
- Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2020) *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-148. <https://CRAN.R-project.org/package=nlme>
- Ripley B (2019) *tree: Classification and Regression Trees*. R package version 1.0-40. <https://CRAN.R-project.org/package=tree>
- Scholkopf B, Smola AJ (2002) *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge
- Segal MR (1992) Tree-structured models for longitudinal data. *J Am Stat Assoc* 87:407–418
- Sela RJ, Simonoff JS (2012) RE-EM trees: a data mining approach for longitudinal and clustered data. *Mach Learn* 86:169–207
- Shin S, Austin PC, Ross HJ, Abdel-Qadir H, Freitas C, Tomlinson G, Chicco D, Mahendiran M, Lawler PR, Billia F, Gramolini A (2021) Machine learning vs. conventional statistical models for predicting heart failure readmission and mortality. *ESC Heart Failure* 8(1):106–115
- Singer JD, Willett JB (2003) *Applied longitudinal data analysis: modeling change and event occurrence*. Oxford University Press, Oxford
- Song X, Mitnitski A, Cox J, Rockwood K (2004) Comparison of machine learning techniques with classical statistical models in predicting health outcomes. In *MEDINFO 2004*, pp. 736–740). IOS Press
- Venkatesh KK, Strauss RA, Grotegut C, Heine RP, Chescheir NC, Stringer JS, Stamilio DM, Menard MK, Jelovsek JE (2020) Machine learning and statistical models to predict postpartum hemorrhage. *Obstet Gynecol* 135(4):935
- Wang YG, Carey V (2003) Working correlation structure misspecification, estimation and covariate design: implications for generalised estimating equations performance. *Biometrika* 90(1):29–41
- Wang Y-G, Lin X (2005) Effects of variance-function misspecification in analysis of longitudinal data. *Biometrics* 61:413–421
- Wei W, Ramalho O, Malingre L, Sivanantham S, Little JC, Mandin C (2019) Machine learning and statistical models for predicting indoor air quality. *Indoor Air* 29(5):704–726
- Xiong Y, Kim HJ, Singh V (2019) Mixed effects neural networks (menets) with applications to gaze estimation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 7743–7752
- Yang L, Liu S, Tsoka S, Papageorgiou LG (2016) Mathematical programming for piecewise linear regression analysis. *Expert Syst Appl* 44:156–167

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.