

Secondary Publication



Messer, Paul; Schmid, Timo

Gradient boosting for hierarchical data in small area estimation

Date of secondary publication: 05.10.2026

Version of Record (Published Version), Article

Persistent identifier: urn:nbn:de:bvb:473-irb-117558x

Primary publication

Messer, Paul; Schmid, Timo (2026): Gradient boosting for hierarchical data in small area estimation, in: Statistics and computing, Dordrecht [u.a.]: Springer Science + Business Media B.V., Vol. 36, No. 5, 218, pp. 1–29, doi: 10.1007/s11222-026-10970-1.

Legal Notice

This work is protected by copyright and/or the indication of a licence. You are free to use this work in any way permitted by the copyright and/or the licence that applies to your usage. For other uses, you must obtain permission from the rights-holders.

This document is made available under a Creative Commons license.



The license information is available online:

<https://creativecommons.org/licenses/by/4.0/legalcode>



Gradient boosting for hierarchical data in small area estimation

Paul Messer¹ · Timo Schmid¹

Received: 2 June 2025 / Accepted: 24 August 2026
© The Author(s) 2026

Abstract

Small Area Estimation (SAE) combines survey data with auxiliary sources such as administrative records, census data, or alternative datasets that typically offer broader coverage. By integrating these sources, SAE enhances the accuracy of (direct) survey estimates. To account for the hierarchical structure of survey data, model-based SAE methods often rely on Linear Mixed Models (LMMs). However, the distributional (e.g., normality) and structural (e.g., linearity) assumptions of LMMs may not always hold in practice, and the accuracy of model-based SAE depends on the validity of these assumptions. To address these limitations, we propose a Mixed Effect Gradient Boosting (MEGB) approach, which combines the flexibility of gradient boosting machines with the ability of mixed models to account for hierarchical data structures. MEGB extends standard gradient boosting by incorporating random effects, allowing it to capture unobserved heterogeneity across domains while retaining a nonparametric framework that models nonlinearities and interactions in the data. MEGB supports the derivation of area-level means from unit-level data and uses a nonparametric bootstrap to estimate the Mean Squared Error (MSE). Its performance is assessed through a model-based simulation study, comparing MEGB to established estimators, and further demonstrated using real-world data. The results suggest that MEGB offers promising area mean estimates and may outperform existing SAE methods in various scenarios.

Keywords Expectation–maximization algorithm · Ensemble learning · Mixed-effects modeling · Nonparametric bootstrap · Uncertainty quantification · Unit-level models

1 Introduction

Statistics play an essential role in evidence-based policy-making, especially in social, economic and environmental development issues. The collection and analysis of more disaggregated and granular statistics can improve the efficiency of policy targeting and interventions. Refined statistics are crucial for the design, monitoring and evaluation of poverty reduction programs and various other development decisions. Key statistics include information such as household income, the number of complete meals per day, access to basic services and the number of people in employment. Through such data, it is possible to gain a detailed view of the socio-economic conditions of the population, allowing for targeted and well-founded decisions. In many areas,

there is a deficit of information on income distribution and well-being, which makes it difficult to identify and address poverty. This information gap is often attributed to limited funding for population surveys, especially in smaller, spatially disaggregated areas like states, districts, or municipalities (Tzavidis et al. 2018). In order to provide these reliable and efficient estimates at disaggregated levels, Small Area Estimation (SAE) methods have been developed, allowing policymakers to make informed decisions despite the lack of available information on a granular level (Rao and Molina 2015). SAE combines survey data with auxiliary data sources such as administrative records, census data, or alternative data sources, which typically have a broader coverage. By integrating these different datasets, SAE improves the accuracy and comprehensiveness of survey estimates, allowing for more accurate identification of needs and more effective allocation of resources. To account for the hierarchical structure of survey data, model-based estimators in SAE often rely on Linear Mixed Models (LMMs) (Jiang and Lahiri 2006). Examples include the Battese-Harter-Fuller (BHF) model (Battese et al. 1988) and the Empirical Best Predic-

✉ Timo Schmid
timo.schmid@uni-bamberg.de

Paul Messer
paul.messer@uni-bamberg.de

¹ Institute of Statistics, Otto-Friedrich-Universität Bamberg,
Feldkirchenstr. 21, 96052 Bamberg, Bavaria, Germany

tor (EBP) (Molina and Rao 2010), which are both based on the nested error regression model. Both models assume that random effects and error terms are normally distributed. The distributional and structural assumptions of these models may not always hold in SAE applications (Rao and Molina 2015) and are often not supported by empirical evidence, especially when dealing with social and economic inequality data. The accuracy of model-based SAE heavily depends on the validity of these assumptions (Jiang and Rao 2020). If the assumptions are not properly met, the resulting estimates may be biased and Mean Squared Error (MSE) estimates may be unreliable. In the case of violations of model assumptions, high-dimensional data, nonlinear relationships or higher-order interactions, traditional small area methods, such as the BHF and EBP, may not be suitable for prediction. To facilitate the above-mentioned possible violations of the normality assumption in LMMs, some approaches focus on the transformation of the dependent variable which is explored in Molina and Martín (2018) as well as in Rojas-Perilla et al. (2020). Molina and Martín (2018) applied a fixed log transformation of the dependent variable, whereas Sugawara and Kubokawa (2019) and Rojas-Perilla et al. (2020) used data-driven transformations for a nested error regression model. An alternative is to develop the EBP under more flexible parametric assumptions (Diallo and Rao 2018; Graf et al. 2019). Another way to avoid violating model assumptions is to use models with less restrictive assumptions. Chambers and Tzavidis (2006) suggest using M-quantile models, which are robust and do not rely on the distributional assumptions of LMMs. However, all previously mentioned approaches against model misspecification rely on a linear specification of the model. In contrast, Opsomer et al. (2008) and Ugarte et al. (2009) discuss the use of non- and semi-parametric methods for estimating small area means through penalized spline regression. Another strategy to address the issue of nonlinear relationships or higher-order interactions is the utilisation of machine learning methods. These methods offer a balance between predictive power and flexibility, without relying on parametric model assumptions. However, the interpretability is limited in comparison to linear regression approaches. In linear approaches, the influence of covariates on the target variable can be precisely quantified. In contrast, methods such as Random Forests (RF) or Gradient Boosting (GB) provide variable importance and partial dependencies. As a result, they can only demonstrate the contribution of each variable to the prediction, indicating which variables are most influential in determining the outcome. The insights into the nature of these relationships are therefore limited, which also limits the interpretability of the model.

Applications of SAE are often characterised by the use of hierarchical data but traditional machine learning models assume that the data is independent and identically distributed (i.i.d) (e.g. Schapire and Freund (2013)). This

assumption is not applicable to hierarchical data, such as data on poverty, as this depends on small geographical areas, for example municipalities. In these cases, the data is often hierarchically structured, as the observations within the same region can be correlated. The predictive performance of machine learning techniques like Classification and Regression Trees (CARTs), RFs or GB can suffer if the hierarchical structure is ignored. Including the hierarchical structure in the predictive model has the potential to improve the predictive performance (Salditt et al. 2023). While LMMs capture the dependencies between observations through random effects, traditional machine learning methods do not address this aspect. This limitation has led to the development of machine learning methods that account for hierarchical structures in the data. Various approaches have been explored with the objective of integrating random effects into machine learning methods. For example, regression tree models have been extended to include random effects, such as in the Mixed Effects Regression Tree (MERT) (Hajjem et al. 2011) and its extensions: the Mixed Effects Random Forest (MERF) (Hajjem et al. 2014) and the Generalised MERT (GMERT) (Hajjem et al. 2017). Bayesian Additive Regression Trees (BARTs) are another tree-based approach; however, they differ from the above-mentioned tree-based methods in that BARTs are a nonparametric Bayesian approach that can be extended to account for hierarchical structures by including random effects (Chipman et al. 2010). According to their research, incorporating linear terms or random effects into a BART model can be achieved by adding a straightforward Markov-Chain-Monte-Carlo step to draw the associated parameters. In particular, in settings concerning non-hierarchical data, BARTs outperform machine learning methods such as neural networks, RFs and GB (Chipman et al. 2010). Tan et al. (2019) developed an extension of the BART approach to Random Intercept Bayesian Additive Regression Trees, designed to address clustered data. In contrast to other tree-based methods, BARTs have not yet been widely applied in SAE contexts. Machine learning methods have already been applied in SAE, often by aggregating unit-level predictions from models like RFs, Cubist regression models or GB (Merfeld and Newhouse 2023; Corral et al. 2022; Viljanen et al. 2022). These methods frequently outperform traditional estimators like the EBP, although they ignore random effects. As mentioned above, Kriegler and Berk (2010), Corral et al. (2021) and Merfeld and Newhouse (2023) demonstrated that machine learning methods led to better results in their applications compared to more traditional small area methods, which take the random effects into account. In contrast, Salditt et al. (2023) emphasise the crucial role of incorporating random effects. They propose two algorithms that are used to compare an extension of GB, which includes random effects in the context of unit-level data with other existing models for hierarchi-

cal data. The findings indicate the necessity of considering random effects, as the GB algorithm with consideration of random effects outperforms the standard GB and standard RF as well as other existing models used for hierarchical data. Krennmair and Schmid (2022) extend the MERF (Hajjem et al. 2014) to a unit-level estimator in the context of SAE. This approach integrates the power and flexibility of machine learning methods with random effects, which has the advantage of not relying on many model assumptions and combines the advantages of RFs with the ability to control for cluster structures of survey data. In line with this approach, Frink et al. (2026) introduced an extension of the GMERT (Hajjem et al. 2017) to a GMERF and incorporated it into the SAE framework for the estimation of area-level proportions, for instance, poverty rates. Krennmair and Schmid (2022) also encourage the use of further machine learning methods such as Support Vector Machines (SVM) or GB instead of an RF. Both RF and GB aim to improve prediction accuracy by combining multiple trees. RFs reduce variance through bagging and decorrelation of trees, while GB reduces both bias and variance by sequentially correcting errors (Hastie et al. 2009). GB generally offers superior predictive performance, as shown in various applications (Hastie et al. 2009; Yoon 2021). Additionally, the GB-algorithm allows for the adaptation of the loss function (Friedman 2001). This can result in more robust statistics, which are particularly advantageous in the context of SAE, where the modeling of skewed income data with extreme outliers is a common challenge (Chambers et al. 2013). While GB has shown strong performance at the unit-level, it has so far only been applied to SAE without incorporating random effects (Merfeld and Newhouse 2023). This raises the key question of whether its predictive power can be further improved by accounting for hierarchical structure. Therefore, this paper proposes the estimation model Mixed Effects Gradient Boosting (MEGB), a small area model that is capable of handling different levels of complexity and flexibility through nonlinearity and a nonparametric approach. As outlined in many aforementioned algorithms, this approach is also based on separating the estimation of the fixed-effects component from that of the random-effects structure using an EM algorithm (Dempster et al. 1977). While the general framework follows the line of Krennmair and Schmid (2022), the key innovation of the present study lies in the replacement of the RF component in MERF by a GB algorithm for estimating the fixed-effects term. This modification affects only the estimation of the fixed effects, whereas the remaining elements - unit-level estimation, aggregation to area-level means, and the Random Effect Block Bootstrap (REBB) procedure, based on the approach outlined in Chambers and Chandra (2013), for MSE estimation - are adopted from the MERF framework. The novelty of MEGB is the integration of GB into the Mixed Effects estimation structure, thereby

extending the flexibility and predictive capability of existing Mixed Effects machine learning methods in the SAE context, while maintaining the ability to produce area-level estimates and quantify uncertainty through the REBB procedure. The proposed approach is also related to recent boosting-based methods for hierarchical data, in particular the MERTBoosting and REEMBoosting approaches of Salditt et al. (2023). A detailed methodological comparison to these and other related approaches is provided in Sect. 2.

The statistical properties related to uncertainty quantification in tree-based machine learning methods are not yet well developed, as the focus is typically on prediction performance (Athey et al. 2019; Hüllermeier and Waegeman 2021). In general, machine learning methods have tended to lag behind in terms of the development of uncertainty statistics (Hüllermeier and Waegeman 2021). While some approaches to uncertainty quantification for boosting exist in the literature, these mainly concern prediction uncertainty at the individual level and do not directly extend to the present SAE setting. The framework proposed in this paper provides a nonparametric bootstrap scheme for estimating the area-level MSE. Section 2 introduces the methods and concepts used where Mixed Effects and GB are briefly explained. This is followed by an introduction of the newly proposed method MEGB, which combines the previously mentioned estimation methods through an EM algorithm. Next, the same section shows how to derive area-level mean estimates from unit-level estimations. Furthermore, Sect. 3 describes the calculation of MSE estimates for the estimated area means by using a nonparametric bootstrap. For unit-level models, such as the MERF, the EBP with Box-Cox transformation (EBP-BC) (Rojas-Perilla et al. 2020), or the newly introduced MEGB, certain data requirements have to be met. Unit-level data from a survey (for instance, at the household level, which is the case in the design-based simulation study in Sect. 5) requires covariates as well as observations of the variable of interest. In contrast, for the unit-level population data, only the covariates X are needed. In Western countries, micro-level population data often cannot be used in design-based simulations. Therefore, additional data sources are needed. Furthermore, the performance of the proposed method is compared to established estimators and evaluated via a model-based and a design-based simulation study in Sect. 4 and Sect. 5 respectively. Two model-based simulation studies are distinguished: model-based simulation A and model-based simulation B. model-based simulation A (Subsection 4.2) follows the setup of Krennmair and Schmid (2022) and benchmarks MEGB against state-of-the-art SAE estimators. In this study, we also assess the uncertainty quantification of the MEGB via MSE estimates obtained through a nonparametric bootstrap. The MERF serves as a key benchmark to assess the performance of MEGB, and the design of model-based simulation A mirrors Krennmair and Schmid

(2022) to ensure comparability between MEGB and MERF. model-based simulation B (Subsection 4.3) adapts the unit-level simulation study of Salditt et al. (2023) to an SAE setting. Here, we compare MEGB to both SAE estimators and the boosting-based estimators considered by Salditt et al. (2023). Section 5 presents the design-based study including an evaluation of the MSE estimates. We report a runtime comparison for all methods only for model-based simulation B, where all algorithms were run under comparable computational settings. For model-based simulation A and the design-based study, MEGB uncertainty was estimated via a bootstrap procedure, which substantially increases runtime and is not directly comparable to the other methods. Section 6 provides suggestions for further research and concluding remarks.

2 Concepts and methods

This section introduces the different concepts and methods used to define and understand the MEGB, which itself is defined in Subsection 2.3.

2.1 Review of linear mixed effects models

LMMs include random effects in addition to the fixed effects $f(\mathbf{X})$. $\mathbf{X}$ is the $(n \times p)$ -dimensional design matrix to p known covariates. Mixed models enable the modeling and estimation of area-specific effects, especially in the case of small sizes in subsamples from areas $d = 1, \dots, D$. In this regard, the linear effect $f(\mathbf{X})$ is extended to $f(\mathbf{X}) + \mathbf{Z}\boldsymbol{\vartheta}$. Here $\mathbf{Z}$ is a subvector of the covariates with dimensions $(n \times q)$ and $\boldsymbol{\vartheta}$ is a vector of the area-specific random effects with dimensions $(D \times q)$. An advantage of mixed models is that correlations arising as a result of repeated observations per area can be considered (Fahrmeir et al. 2013).

In general terms, a mixed model is given by:

$$\mathbf{y} = f(\mathbf{X}) + \mathbf{Z}\boldsymbol{\vartheta} + \boldsymbol{\epsilon} \quad (1)$$

In the context of SAE, the model takes the following form for each domain d :

$$\mathbf{y}_d = f(\mathbf{X}_d) + \mathbf{Z}_d\boldsymbol{\vartheta}_d + \boldsymbol{\epsilon}_d, \quad d = 1, \dots, D, \quad (2)$$

$\mathbf{y}_d$ is an n_d -dimensional vector of response variables of area d with within-domain sample sizes n_d . The terms area and domain are used synonymously in this paper. D defines the number of areas, so $n = \sum_{d=1}^D n_d$. $\mathbf{X}_d$ and $\mathbf{Z}_d$ are $(n_d \times p)$ and $(n_d \times q)$ -dimensional design matrices to p known covariates, where q is limited to $q = 1$ in this setting, which means only one random effect exists per area. $\boldsymbol{\vartheta}_d$ is a scalar of area-specific effects and $\boldsymbol{\epsilon}_d$ is a n_d -dimensional error vector

(Fahrmeir et al. 2013). The design and covariance matrices are assumed to have the following block-diagonal structure:

$$\mathbf{Z} = \text{diag}(\mathbf{Z}_1, \dots, \mathbf{Z}_D),$$

$$\mathbf{R} = \mathbb{1}_n \sigma_\epsilon^2,$$

$$\mathbf{G} = \mathbb{1}_D \sigma_\vartheta^2,$$

where $\mathbb{1}_n$ and $\mathbb{1}_D$ denote the $n \times n$ and the $D \times D$ identity matrix, respectively. From $\mathbf{G}$, it can be deduced that only between-area variations are considered in this paper.

The distributional assumptions for $\boldsymbol{\vartheta}_d$ and $\boldsymbol{\epsilon}_d$ are:

$$\boldsymbol{\vartheta}_d \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\vartheta^2),$$

$$\boldsymbol{\epsilon}_d \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_\epsilon^2),$$

$$\boldsymbol{\vartheta} \perp\!\!\!\perp \boldsymbol{\epsilon}.$$

Based on these assumptions, the covariance matrix of the observations $\mathbf{y}_d$ is defined as:

$$\mathbf{V}_d = \sigma_\epsilon^2 \cdot \mathbf{I}_{n_d} + \sigma_\vartheta^2 \cdot \mathbf{Z}_d \mathbf{Z}_d',$$

where σ_ϵ^2 and σ_ϑ^2 are the residual and random intercept variances, respectively. The overall covariance structure is given by:

$$\text{Cov}(\mathbf{y}) = \mathbf{V} = \text{diag}(\mathbf{V}_1, \dots, \mathbf{V}_D).$$

Setting $f(\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$ in Eq. 1 and assuming a single random intercept per area (i.e., a nested error structure) yields the BHF proposed by Battese et al. (1988). Defining $f(\cdot)$ as an RF results in the MERF approach proposed by Hajjem et al. (2014), which Krennmair and Schmid (2022) extended to a small area context. Replacing the RF used in the approach of Hajjem et al. with an XGBoost algorithm (Chen and Guestrin 2016) results in the MEGB introduced in this paper.

2.2 Extreme gradient boosting

GB refers to a class of ensemble machine learning algorithms that are constructed from decision tree models. The trees are added to the ensemble one at a time and adjusted to correct the prediction errors of the previous models. This is a type of ensemble model for machine learning called boosting, first presented by Freund and Schapire (1997). The idea of ARCing (Adaptive Reweighting and Combining) algorithms (Breiman 1999) was extended by Friedman (2001). The models are fitted with an arbitrary differentiable loss function and a gradient descent optimization algorithm. This is where the name *Gradient Boosting* is derived from, as the loss gradient is minimized when fitting the model. A requirement for a loss function is that it must be differentiable (Friedman

2001); in regression cases, this can be the squared error for example. In this context, GB specifically employs decision trees as foundational weak learners. Henceforth, any reference to GB within this paper pertains exclusively to GB with decision trees. A GB model is a series of weak decision trees that are created one after the other to form a strong learner. In addition, each weak decision tree is trained to correct the mistakes made by the previous ones. The outcomes produced by the newly formed tree, when scaled by the learning rate η , are subsequently integrated with the outcomes from the sequence of existing trees. This process aims to refine or enhance the overall results generated by the model. A fixed number of trees is added, or training is terminated when the loss reaches an acceptable level or no longer improves. XGBoost is an efficient and effective implementation of the GB algorithm provided in an open-source library. It is an optimized implementation of GB, which enhances speed, performance, and scalability through algorithmic improvements and system optimizations, including parallel processing and regularization features. One of the main advantages of GB models is the absence of assumptions such as linearity or the distribution specification of model errors, but the observations are still assumed to be independent.

As the performance of an XGBoost model largely depends on hyperparameters, they need to be tuned and adjusted to find a good trade-off between sufficient model complexity and the likelihood of encountering overfitting.

2.3 Mixed effects gradient boosting

Combining Mixed Effects and GB results in the MEGB. As mentioned in Subsection 2.1, this approach is based on the MERF of Hajjem et al. (2014) and the SAE extension of Krennmair and Schmid (2022). The objective is to predict values for Out-Of-Sample observations, for which additional covariates from census or register information across areas are available, by identifying the relation $f(\cdot)$ as well as the random effects between covariate X and objective y . The model is formed by a fixed effect part $f(X)$ and a random part $Z\theta$, which covers the dependencies introduced by the random effects (Krennmair and Schmid 2022). The fixed part $f(X)$ of Eq. 1 is replaced by a GB Decision Tree model.

The assumption is made that the distribution of the unit-level errors follows a normal distribution, given by $\epsilon_i \sim N(0, R)$ for unit i . In this context, the assumption of normality does not impact the efficacy of GB, and there is no requirement for the residuals to follow a normal distribution for the method to be applicable. As shown below, the MEGB algorithm is estimated by an EM algorithm. However, regarding the part involving random components, an appropriate likelihood function is necessary. This is to guarantee that the EM algorithm, defined further below, successfully converges to a local maximum within the parameter space

for estimating parameters (Krennmair and Schmid 2022). Krennmair and Schmid's (2022) normality-based likelihood function is used because of two key benefits. Initially, it provides a closed-form solution of the integral of the Gaussian likelihood function, which facilitates the estimation of random effects. Additionally, by the mean of the unit-level residual sum of squares, the maximum likelihood estimate for the variance of the unit-level errors $\hat{\sigma}_\epsilon$ is given. The EM algorithm to fit Eq. 1 is structured as follows: The fixed component of the response variable y^* is computed by the algorithm at the initial stage. The variable y^* describes the adjusted response variable, where the currently estimated random effect ϑ is subtracted from y and the GB algorithm is fitted to the adjusted response variable. The number of trees of the GB algorithm, used at this stage in Line 7 of Algorithm 1, follows an early stop criterion. Early stopping is applied when the model's performance, measured by the Root Mean Squared Error (RMSE) on a cross-validation test subsample, does not improve for a specified number of rounds, halting further tree construction. In the second phase, the algorithm deducts the estimated fixed component from the raw data y to get the adjusted response y^{**} reduced by the fixed effect and an LMM is estimated. Subsequently, the random component is newly estimated. The process continues iteratively through these steps until convergence is reached. The algorithm enhances the accuracy of predictions by adjusting the fixed component and variance components based on the residuals after the fixed component has been accounted for, thus refining the model with each iteration to prevent overfitting. In summary, the MEGB algorithm first estimates the GB function, assuming that the random-effects term is correct. As a second step, it estimates the random-effects part, assuming that the predictions from the GB model are correct. The proposed algorithm for MEGB is shown in Algorithm 1.

Algorithm 1 Mixed Effects Gradient Boosting

```

1: Initialization:
2: set  $\vartheta_0 = 0$ 
3: Set hyperparameters for GB.
4: EM Algorithm:
5: for  $iter = 1$  to  $Iter_{max}$  do
6:   Subtract random effects  $\vartheta_{iter}$  from the raw data  $y$ :  $y^* = y - Z\hat{\vartheta}_{iter-1}$ .
7:   Fit a GB on the new corrected  $y^*$  to obtain  $\hat{f}_{iter}(y^* | X)$ .
8:   Determine  $y^{**} = y - \hat{f}_{iter}(y^* | X)$ , the raw data  $y$  subtracted by the GB estimate.
9:   Estimate LMM with intercept:  $LMM(y^{**} | Z)$ .
10:  Estimate random effects  $\hat{\vartheta}_{iter}$ .
11:   $\left| \frac{\log(\text{likelihood}_{iter}) - \log(\text{likelihood}_{iter-1})}{\log(\text{likelihood}_{iter-1})} \right| < \text{error tolerance} \rightarrow \text{Stop}$ 
12: end for

```

The convergence of Algorithm 1 is evaluated by monitoring the marginal change in the modified Generalized

Log-Likelihood (GLL) criterion at each iteration. The GLL is given by (Hajjem et al. 2014)

$$\text{GLL}(f, \vartheta_d \mid y) = \sum_{d=1}^D \left[\frac{1}{\sigma_\epsilon^2} \|y_d - f(X_d) - Z_d \vartheta_d\|^2 + \frac{\vartheta_d^2}{\sigma_\vartheta^2} + \log(\sigma_\vartheta^2) + n_d \log(\sigma_\epsilon^2) \right] \quad (3)$$

Since each area contains only a single random effect ($q = 1$), the vector ϑ_d is scalar and the design matrix Z_d reduces to a column of ones. As a result, the matrix expressions in the likelihood simplify to scalar terms. Krennmair and Schmid (2022) derived the estimator for ϑ for known parameters G and R as $\hat{\vartheta} = GZ^T V^{-1}(y - \hat{f}(X))$. After convergence of the EM algorithm, the maximum likelihood estimators, given the data, deliver variance components $\hat{\sigma}_\epsilon^2$ and $\hat{\sigma}_\vartheta^2$ which are required for the uncertainty estimations described in Sect. 3. Having defined how MEGB produces unit-level predictions, we now describe how to derive small area average predictions.

2.4 Small area average predictions

In the context of SAE, predictions on the individual level are of less interest than area-level predictions (Morales et al. 2021). Therefore, indicators like area-level means or area-level totals are needed. The MEGB model from Eq. 2 predicts conditional means on the individual level of a metric dependent variable $y_{d,i}$ given unit-level auxiliary information $x_{d,i}$. To align this approach with the outlined objectives, a method is needed, which generates area-level means or totals from individual predictions. To facilitate this area-level estimation, we adopt structural simplifications similar to those used in the LMM suggested by Battese et al. (1988). This implies the existence of a single random effect, consequently resulting in the design matrix Z having the dimension $n \times D$. The design matrix Z is filled with area-intercept indicators, employing a $D \times 1$ vector of random effects denoted as $\vartheta = [\vartheta_1, \dots, \vartheta_D]'$. This reduces the variance-covariance matrix for random effects to $\mathbb{1}_D \sigma_\vartheta^2$ (Morales et al. 2021). $\mathbb{1}_D$ is the $D \times D$ identity matrix, which implies the independence of the ϑ_d 's. To derive an estimator for area averages or totals, the unit-level y_i 's of the population data are estimated through the GB algorithm $\hat{f}(X)$, employing supplementary data sources typically obtained from a census. The estimated $\hat{y}_{d,i}$ are subsequently aggregated based on their respective area, with the addition of the estimated area-specific random effect $\hat{\vartheta}_d$, leading to the estimated area mean $\hat{\mu}_d^{MEGB}$ which can be seen in Eq. 4. The calculation of the area total $\hat{Y}_d^{MEGB}$ estimation is shown in Eq. 5 and follows the same procedure.

$$\hat{\mu}_d^{MEGB} = \hat{f}(x_d) + \hat{\vartheta}_d = N_d^{-1} \sum_{i=1}^{N_d} (\hat{f}(x_{d,i})) + \hat{\vartheta}_d \quad (4)$$

$$\hat{Y}_d^{MEGB} = \sum_{i=1}^{N_d} (\hat{f}(x_{d,i})) + N_d \cdot \hat{\vartheta}_d \quad (5)$$

$\hat{f}(x_d)$ denotes the aggregated GB estimations and builds the estimator for $\hat{\mu}_d^{MEGB}$ alongside $\hat{\vartheta}_d$. For areas that are not sampled, the parameter $\hat{\vartheta}_d$ is omitted. Note that computing the estimators in Eqs. 4 and 5 requires access to auxiliary information $x_{d,i}$ for all N_d population units, typically available from census micro-data. While this requirement can be demanding in certain applications, aggregated auxiliary information cannot be used as a substitute for $x_{d,i}$ in this context - unlike in linear models - due to the nonlinearity of $\hat{f}$.

2.5 Relation to existing approaches

As already mentioned in the introduction, several attempts have been made to extend GB to hierarchical data structures. In the following, we provide a more detailed discussion of how our proposed MEGB approach relates to existing contributions in this field. Our approach differs from existing methods that combine boosting and Mixed Effects modeling. Griesbach et al. (2021) directly embed LMMs into the boosting algorithm, restricting base learners to linear components and updating random effects in each iteration. While this provides a principled integration of Mixed Effects, it limits the ability to capture nonlinear structures that tree-based GB models are designed to detect. In contrast, Yang et al. (2025) apply a sequential two-step approach, first estimating an LMM and then fitting an XGBoost model to the residuals. Although this setup allows for nonlinear effects, it lacks an iterative joint estimation of linear and nonlinear components. A third closely related contribution is the MERTBoosting framework proposed by Salditt et al. (2023). Salditt et al. (2023) actually introduce two algorithms, the MERTBoosting algorithm and the REEMBoosting algorithm, both of which extend the mixed-effects regression tree framework by replacing the regression tree as the fixed-effects learner with a boosting ensemble within an EM scheme: MERTBoosting builds on the MERT algorithm of Hajjem et al. (2011), whereas REEMBoosting builds on the RE-EM tree of Sela and Simonoff (2012). Among these, our approach is most closely related to the MERTBoosting variant. However, three main differences exist between our approach and MERTBoosting. First, Salditt et al. (2023) implement a custom boosting routine, while our method builds upon XGBoost, allowing for a richer set of hyperparameter choices and regularization options. Second, whereas our implementation always includes an intercept estimated jointly with the boost-

ing component, Salditt et al. (2023)'s framework offers more flexibility in how the fixed effects are handled, including the option to exclude the intercept or to assign certain covariates (including the intercept) to the LMM part instead of the boosting component. We view this difference primarily as a design choice rather than a limitation. In our SAE setting, assigning all covariates to the boosting component preserves the full flexibility of the GB learner to detect nonlinearities and interactions, whereas allocating selected covariates exclusively to the mixed-effects part would restrict this flexibility. More broadly, the present specification can be viewed as a more constrained modeling choice within the wider framework proposed by Salditt et al. (2023), which explicitly allows for a more flexible allocation of the intercept and individual covariates between the boosting and mixed-effects components. Finally, variance components are updated differently: Salditt et al. (2023) apply closed-form, moment-based EM updates for the variance parameters, whereas our implementation estimates them via REML, as in standard LMM software. While these choices may lead to minor differences in variance estimates, the resulting BLUPs for the random effects are nearly identical. Recent work has further extended GB to hierarchical and longitudinal data settings, including the Mixed Effects GB approach by Olaniran et al. (2025) and the GMEXGBoost method proposed by Asadi et al. (2026). These approaches demonstrate that incorporating random effects into boosting frameworks can improve predictive performance for clustered or longitudinal data. The approach of Olaniran et al. (2025) is the closest of these in both name¹ and estimation scheme: it likewise refits a boosting model at each iteration to a pseudo-response net of the current random-effects estimate within an EM-type procedure. It differs, however, in being formulated for longitudinal data, with subject-specific random intercepts and slopes in its implementation and simulations, in building on `gbm` rather than `XGBoost`, and in updating the variance components, as specified in their algorithm, from the empirical moments of the predicted random effects rather than via REML. Most importantly, it targets unit-level prediction and variable selection with $p \gg n$ and does not address area-level estimation: it provides neither the aggregation of unit-level predictions to domain means nor uncertainty quantification at the domain level. We attempted to include their publicly available implementation, using the settings described in their paper. Two features of our setting lie outside the design for which those settings were selected: model-based

simulation A has two covariates, whereas their simulation study considers up to $p = 2,000$ predictors, and their design assumes repeated measurements of the same units over time, which cross-sectional SAE data do not provide. Under these conditions we did not arrive at a configuration that we would regard as a fair basis for comparison, and the implementation did not run reliably across all replications in the more complex scenarios. We therefore do not report results for this method, and we regard this as a consequence of applying it outside the regime for which it was designed rather than as an indication of its performance in the settings it targets. GMEXGBoost shares with MEGB the idea of coupling `XGBoost` with mixed-effects estimation in an iterative scheme, but the two differ in what the boosting step is asked to fit. In GMEXGBoost, a generalized linear model is fitted once before the iterations begin, and the boosting target is formed from its linear predictor net of the current random-effects estimate. The boosting step and the mixed-model step that updates the random effects are recomputed at every iteration, but that linear predictor is not, so the observed response never serves as the boosting target. Because that predictor is a linear function of the covariates, the boosting ensemble, though piecewise constant and hence formally nonlinear, cannot recover nonlinear structure in the true regression function that the initial linear fit does not already contain. This follows from the algorithm as published, in which the linear predictor is defined in the initialization step, outside the iteration loop. MEGB instead refits the boosting component to the response itself, net of the current random-effects estimate, at every iteration. A second difference concerns scope: GMEXGBoost is formulated for responses in the exponential family but is evaluated only for binary outcomes, with a Bernoulli simulation design and a binary clinical application; the authors describe the framework as applicable to continuous and count responses, but do not evaluate it for them. Like the approaches discussed above, it does not address area-level estimation. We report an empirical comparison in the Appendix (Subsection "Additional Results for Simulation A", Table 10). A further closely related contribution is the LMM-`XGBoost` framework proposed by Cho and Mueller (2026), which integrates `XGBoost` into a linear mixed model via an iterative EM scheme. Their approach is designed for multilevel cross-classified data, in which observations are simultaneously nested within two non-nested higher-level clustering dimensions, and introduces an additional projection step to enforce orthogonality between the `XGBoost` component and the random effects. Furthermore, Cho and Mueller (2026) develop a group-aware permutation importance measure and a combined-group cross-validation strategy tailored to cross-classified designs. However, none of these approaches address the specific requirements of SAE. In particular, they do not provide a framework for deriving area-level estimates from unit-level predictions or

¹ The abbreviation MEGB, and the name of the accompanying R package, coincide with those used by Olaniran et al. (2025). The two methods were developed independently: the present approach was first introduced under this name in an earlier version of this paper, released as a preprint in June 2024 (arXiv:2406.04256), predating the submission of Olaniran et al. (2025) (received 3 February 2025). Throughout this paper, MEGB refers to the method proposed here.

uncertainty quantification at the domain level, as required in the SAE context. In contrast, the proposed MEGB approach is explicitly tailored to the SAE context. It enables the estimation of area-level means by aggregating unit-level predictions and incorporates a nonparametric bootstrap procedure for estimating the MSE, thereby providing a complete framework for SAE. Overall, our MEGB approach integrates Mixed Effects directly into a GB-based estimation framework. This enables the simultaneous and iterative estimation of linear, nonlinear, and random effects within a unified algorithmic structure. Furthermore, it is customised to the context of SAE, thereby providing both domain-level predictions and domain-level uncertainty quantification.

3 Uncertainty estimation

MSE estimates are an important component of SAE. For classical model-based estimators, analytic or approximation-based expressions for the MSE are often available under specific assumptions. For machine-learning-based estimators, however, uncertainty quantification is generally more challenging, especially when nonlinear learners are combined with hierarchical dependence structures. Although some uncertainty quantification results and approaches exist for particular boosting frameworks, these are mainly concerned with prediction uncertainty at the individual level and do not directly apply to the present SAE setting. In particular, existing results are mainly available under restrictive assumptions, such as linear or additive base learners, independent observations, or prediction-oriented frameworks that focus on unit-level uncertainty rather than area-level MSE estimation. For tree-based GB combined with random effects in a hierarchical SAE setting, no closed-form expressions or established theoretical results for the MSE of area-level predictors are currently available. Consequently, we adopt a resampling-based approach for uncertainty quantification in the present setting. In the case of the MERF, Krennmair and Schmid (2022) employed a nonparametric bootstrap procedure based on the approach of Chambers and Chandra (2013). Following the same principle, we adapt this REBB procedure to quantify the uncertainty of the MEGB estimator. Block bootstrapping allows us to expand bootstrapping to clustered data. The aim is to preserve the cluster structure of the observations. Keeping the cluster structure enables the quantification of uncertainty within the clusters (Carlstein et al. 1998). Chambers and Chandra (2013) proposed an enhancement of the block bootstrap that explicitly accounts for random effects and within-block (level 1) variance, forming the REBB. It is required that the level 1 variance is similar across the blocks. Algorithm 2 shows the REBB used for the MEGB mean estimates. The REBB simulates the random effects for every bootstrap sample in order to replicate the

variance within clusters (blocks). This is done to model and quantify the uncertainty from within the clusters. First, using the marginal residuals, the average residuals for every cluster (level 2) and the residuals within the cluster (level 1) are calculated. Next, these residuals are scaled to the estimated variance $\hat{\sigma}_\epsilon^2$ and then centered. Respectively, the level 2 residuals are scaled to $\hat{\sigma}_\eta^2$ and centered. In the next step, the scaled and centered residuals are used to generate independent bootstrap samples for the errors on level 2 and level 1. These errors are used to simulate bootstrap data, and lastly the MEGB model is estimated on the bootstrap data. The proposed REBB has, in the case described here, the advantage of not relying on distributional assumptions. The clusters in this context refer to the domains in SAE. We denote the outcome of selecting a simple random sample of size D with replacement from set A as $srsur(A, D)$. Algorithm 2 is used to calculate the area mean residuals but can easily be adapted to calculate the total area residuals by following the same procedure as for the area mean residuals, with the exception that the residuals are not divided by n_d .

The estimation of the MSE for domain d is given by:

$$\widehat{MSE}_d = B^{-1} \sum_{b=1}^B (\mu_d^{(b)} - \hat{\mu}_d^{(b)})^2 \quad (6)$$

4 Model-based simulations

In this section, the performance of the MEGB is evaluated through two simulation studies, simulation A (see Subsection 4.2) and simulation B (see Subsection 4.3), each designed to assess the model under different data-generating mechanisms and scenarios. While both simulations share the same evaluation metrics and rely on an overlapping set of comparative estimators, they differ in the underlying data-generating mechanisms, error distributions, and overall complexity of the scenarios.

All scripts required to reproduce the model-based simulation results, including the data-generating mechanisms and the MEGB implementation, are available in the following public GitHub repository: <https://github.com/pellmess/MEGB-Simulations>.

4.1 Comparative estimators, evaluation metrics

To ensure a comprehensive assessment, the MEGB is compared to both well-established model-based SAE methods and modern machine learning-inspired approaches, including related boosting-based unit-level estimators as discussed in Salditt et al. (2023). Specifically, the compared SAE methods include the BHF as implemented in the **saeTrafo** package (Würz 2022), the EBP with a data-driven Box-Cox trans-

Algorithm 2 Random Effect Block Bootstrap

1: Calculate residuals:

Calculate the marginal residuals within each area d and unit i

$$\hat{e}_{d,i} = y_{d,i} - \hat{f}_{MEGB}(X_{d,i})$$

for a given $\hat{f}(\cdot)$.

Calculate the level 2 average residuals for each of the D areas using the marginal residuals:

$$\bar{r}_d^{(2)} = n_d^{-1} \sum_{i=1}^{n_d} \hat{e}_{d,i}$$

Calculate the level 1 residuals for each area d : $\hat{r}_{d,i}^{(1)} = \hat{e}_{d,i} - \bar{r}_d^{(2)}$, $i = 1, \dots, n_d$, $d = 1, \dots, D$

$\hat{r}_{d,i}^{(1)}$ are scaled to the estimated variance $\hat{\sigma}_\epsilon^2$ and subsequently centered. This results in the scaled and centered level 1 residuals $\hat{r}_{d,i}^{1,c}$. Analogously, level 2 residuals $\bar{r}_d$ are scaled to $\hat{\sigma}_\theta^2$ and centered to yield $\bar{r}_d^{2,c}$, mirroring the process for level 1.

$\bar{r}_d^{2,c}$ is, in this case, the $D \times 1$ scaled and centered vector of the D area-level (level 2) average residuals. $\bar{r}_d^{2,c}$ and $\hat{r}_d^{1,c}$ is the $n_d \times 1$ scaled and centered vector of unit-level (level 1) residuals $\hat{r}_{d,i}^{1,c}$ for group d .

2: Definition of bootstrap sample errors:

Sample independently with replacement from the two sets of centred and scaled residuals. The level 2 bootstrap errors are given by $\bar{\mathbf{r}}^{2,b} = srswr(\bar{\mathbf{r}}^{2,c}, D)$, whereas the level 1 bootstrap errors are given by $\hat{\mathbf{r}}_d^{1,b} = srswr(\hat{\mathbf{r}}_d^{1,c}, n_d)$.

3: Simulate bootstrap data:

Use the model

$$y_{d,i}^{(b)} = \hat{f}(X_{d,i}) + Z\bar{r}_d^{2,b} + \hat{r}_{d,i}^{1,b}$$

to simulate the bootstrap population data.

Calculate the bootstrap population area means $\mu_d^{(b)} = N_d^{-1} \sum_{i=1}^{N_d} y_{d,i}^{(b)}$.

Draw bootstrap sample data from the bootstrap population maintaining sample size n_d from the original sample.

4: Fit the MEGB algorithm:

To get the bootstrap area-level estimates $\hat{\mu}_d^{(b)}$, fit the MEGB model (Subsection 2.3) with the hyperparameters of the point estimation to the bootstrap sample data generated in the step before and estimate the area mean (Subsection 2.4).

5: Repeat B times for B sets of bootstrap parameter estimates:

Repeat the steps 2 – 4 B times to generate B area mean estimates.

formation (Rojas-Perilla et al. 2020) implemented in the **emdi** package (Kreutzmann et al. 2019), and the MERF from the **SAEforest** package (Krennmair and Frink 2025). In addition, a set of boosting-based and machine learning estimators, similar to those used in the simulation study by Salditt et al. (2023), are included. Namely **MERTBoosting** (Salditt et al. 2023), **REEMBoosting** (Salditt et al. 2023), **MBOOST** (Hothorn et al. 2010), and **MBOOST_T** the tree-based variant of **MBOOST** (Hothorn et al. 2010), as well as the more general machine learning benchmarks **Random Forest** (Liaw and Wiener 2002) and **XGBoost** (Chen et al. 2023). **XGBoost** is employed as a general benchmark that

does not account for hierarchical structure or random effects. Including XGBoost allows us to evaluate the importance of modeling random effects in the small area context and to assess the value added by explicitly incorporating hierarchical information.

The point estimates for the area means are evaluated using the relative root mean squared error (RRMSE) and the relative bias (RB) out of the M Monte-Carlo simulation runs. They are defined as follows:

$$RB_d = \frac{1}{M} \sum_{j=1}^M \frac{\hat{\mu}_{d,j} - \mu_{d,j}}{\mu_{d,j}},$$

$$RRMSE_d = \frac{RMSE_d}{\bar{\mu}_d},$$

$$RMSE_d = \sqrt{\frac{1}{M} \sum_{j=1}^M (\hat{\mu}_{d,j} - \mu_{d,j})^2}$$

where $\hat{\mu}_{d,j}$ is the estimated area average for area d in Monte-Carlo run j , while $\mu_{d,j}$ denotes the corresponding true area mean.. $j = 1, \dots, M$ denotes one of M Monte-Carlo simulation runs. Further, $\bar{\mu}_d = M^{-1} \sum_{j=1}^M \mu_{d,j}$ denotes the average true area mean across all Monte-Carlo runs.

To evaluate the MSE estimates, the RB and the RRMSE of the MSE estimates are used. They are defined as follows:

$$RB-RMSE_d =$$

$$\frac{\sqrt{M^{-1} \sum_{j=1}^M MSE_{d,j}^{(est)}} - RMSE_d^{(emp)}}{RMSE_d^{(emp)}} \cdot 100,$$

$$RRMSE-RMSE_d =$$

$$\frac{\sqrt{M^{-1} \sum_{j=1}^M \left(\sqrt{MSE_{d,j}^{(est)}} - RMSE_d^{(emp)} \right)^2}}{RMSE_d^{(emp)}} \cdot 100$$

where $MSE_{d,j}^{(est)}$ denotes the estimated MSE for area d in Monte-Carlo simulation run j , as defined in Eq. 6. $RMSE_d^{(emp)}$ denotes the empirical RMSE of the area estimator across the Monte-Carlo runs.

4.2 Model-based simulation A

The model-based simulation A closely follows the approach of Krennmair and Schmid (2022) and compares the MEGB to the full set of comparative estimators introduced in Subsection 4.1. This simulation focuses on scenarios reflecting the data-generating mechanisms considered by Krennmair and Schmid (2022), allowing for a structured assessment of estimator performance under controlled linear settings.

4.2.1 Simulation setup

Simulation A considers four model-based simulation scenarios, defined as Linear-Normal, Complex-Normal, Linear-Pareto, and Complex-Pareto, which are constructed as follows: Across all scenarios, the random effects ϑ are normally distributed with $\mu = 0$. Also, the covariates x_1 and x_2 are normally distributed. The Linear-Normal and the Complex-Normal scenario both have normally distributed residuals. However, while the Linear-Normal scenario uses a linear relationship, the Complex-Normal model is more elaborate, as it includes an interaction $x_1 \cdot x_2$ and a quadratic term x_2^2 . The Linear-Pareto and Complex-Pareto scenarios are analogous to the Linear-Normal and Complex-Normal scenarios, with the difference that now, instead of the normal distribution, the Pareto distribution is used for the error term ϵ . In these cases, the Pareto distribution is defined with parameters $k = 3$ and $a = 800$, yielding a strictly positive and right-skewed distribution - a characteristic commonly observed in income data. Using the Pareto distribution makes it possible to examine the MEGB in a more complex scenario with error terms skewed to the right. Details on the scenarios can also be seen in Table 1. The differences between the Complex-Normal and Complex-Pareto scenarios are inherited from the benchmark design of Krennmair and Schmid (2022). In particular, the higher intercept in the Complex-Pareto scenario is used to keep the generated outcome values on a realistic positive scale under the more strongly right-skewed error distribution. The adjusted covariate variances and the larger random-effects variance in this scenario further increase between-area heterogeneity and were retained unchanged in order to preserve full comparability with the original MERF benchmark setting. Hence, these parameterization differences are not introduced to favor a particular method, but to replicate the established simulation design against which MEGB is compared. Each scenario is repeated for $M = 500$ Monte-Carlo simulation runs. The setting for the simulations is as follows: It consists of a population U with size $N = 50,000$ and $D = 50$ disjunct areas $U_1, \dots, U_D$ of size $N_d = 1000$ for all areas d . The sample size created through stratified random sampling is given as $n = \sum_{d=1}^D n_d = 1421$, where the observations vary between 6 and 50 per area, with a mean of $\bar{n}_d = 28.4$ and a median of 28 per area. The simulation setting follows the paper of Krennmair and Schmid (2022) and is used to compare the MEGB and the MERF. In addition to evaluating point estimates, this simulation setting also allows for assessing the accuracy of the uncertainty estimation. Specifically, the estimated MSE values obtained via the bootstrap procedure proposed in Sect. 3 with $B = 100$ are compared to the corresponding empirical (true) MSE values to examine the performance of the proposed MSE estimation approach. It was also determined that increasing the number of Bootstrap

replications (for example, $B = 200$ or $B = 500$) yielded similar results, indicating stable estimates. The MSE is then calculated using the already available empirical true values. In Subsection 4.2.2 the empirical true and the estimated MSE values are compared to assess the performance of the MSE estimation bootstrap procedure used. The hyperparameters of the XGBoost model, except for the number of trees (n_{rounds}) are already chosen before running the EM algorithm and are therefore fixed. To determine these hyperparameters, we conducted a grid search within the model-based simulation setup, using the RMSE of the unit-level predictions of the GB model on a cross-validated test subsample as the optimisation criterion. The search grid comprised the following combinations:

$$\begin{aligned}\eta &\in \{0.01, 0.015, 0.025, 0.05, 0.1\} \\ \gamma &\in \{0, 0.05, 0.1, 0.5, 0.7, 0.9, 1.0\} \\ max_depth &\in \{2, 3, 4, 5, 6, 10\} \\ min_child_weight &\in \{1, 2, 3\} \\ subsample &\in \{0.5, 0.75, 1\},\end{aligned}$$

where η denotes the learning rate and γ the minimum loss reduction required to make a further partition. For each Monte-Carlo simulation run and for each of the four model-based simulation A scenarios, we determined the optimal hyperparameter configuration within this grid. The grid-search results indicated that predictive performance remained stable across neighbouring hyperparameter combinations and across all four scenarios. Based on this observation, we used the modal (i.e., most frequently optimal) hyperparameter configuration for MEGB and applied it consistently across the simulation studies. This decision ensured comparability with MERF: since we used the fixed hyperparameter configuration from the original MERF implementation of Krennmair and Schmid (2022), adopting a fixed configuration for MEGB avoids introducing differences in tuning as a source of unfair performance variation between the two approaches.

The configuration specifies a learning rate η of 0.1, controlling the step size for each iteration in the optimization process. The parameter max_depth is limited to 3 and defines the maximum depth of each tree. The min_child_weight is also set to 3, affecting decisions on further splitting at the nodes. The subsample-parameter is adjusted to 0.5, which specifies that only half of the data samples are used to grow the trees. Additionally, γ is set to 0.9, specifying the minimum loss reduction required to make a further partition.

For the MERTBoosting and REEMBoosting approaches, we follow the tuning recommendations and parameter settings proposed by Salditt et al. (2023). This includes the specification of the boosting procedure and tree-related parameters as described in their simulation study. By adher-

ing to the original configurations, we ensure that these methods are evaluated under conditions consistent with their intended use. For MERF, which relies on RF as the fixed-effects learner, we use the default hyperparameter configuration and implementation as described in Krennmair and Schmid (2022), without additional tuning, to ensure comparability with the original benchmark setting. Further details on the tuning of MERF and the corresponding results are provided in the appendix. The comparison confirms that additional tuning does not materially affect the relative performance of MERF in our simulation setup. For the non-hierarchical benchmark models (RF and XGBoost), we use the same hyperparameter configurations as in their corresponding hierarchical counterparts (MERF and MEGB, respectively). This ensures that differences in performance are attributable to the inclusion of random effects rather than to differences in tuning. Given the shared simulation design and data-generating processes, these configurations provide a suitable and consistent basis for comparison across both hierarchical and non-hierarchical models.

4.2.2 Simulation results

In this subsection, the results of the simulation described in Subsection 4.2.1 will be discussed and evaluated using the RB and the RRMSE. Figure 1 shows boxplots of the RB for the four scenarios described in Table 1 and the corresponding mean and median RB values are available in Table 2.

The Linear-Normal scenario, which assumes a linear relationship with normally distributed random effects and error terms, represents the correctly specified case for the BHF estimator. The BHF is derived from an LMM with Gaussian random effects and errors and therefore serves as the gold standard in this setting. In the remaining scenarios, the BHF is included primarily to assess its robustness under model misspecification, rather than as the reference model. As demonstrated in Table 2, the bias magnitude is consistently below 0.6% for all models, except for RF and XGBoost, indicating an absence of substantial systematic error in the remaining approaches. Due to the very low bias values across all methods, the primary focus of the comparison is on evaluating their performance in terms of RRMSE. As demonstrated in Fig. 1 and Fig. 2, for a linear relationship with normally distributed residuals, the BHF and the EBP-BC seem to perform slightly better in terms of a lower RRMSE than the MERF, MEGB, MERTBoosting and REEMBoosting. This is also underlined by Table 3, where BHF and EBP-BC have a median RRMSE of 3.55 and 3.61% as opposed to the higher median RRMSE of 3.65% for the MEGB and 3.84% for the MERF in the Linear-Normal scenario. RF and XGBoost have the highest median RRMSE with 10.19 and 10.25%.

The Linear-Pareto scenario, which maintains a linear relationship but differs from the Linear-Normal case through non-normally distributed residuals, reveals slight changes in estimator performance. The estimation quality of the MEGB, MERTBoosting, and REEMBoosting is comparable to that of the BHF, whereas the EBP-BC achieves the best results in this setting. These findings are illustrated in Fig. 2 and summarized in Table 3. Because the EBP-BC employs a data-driven Box–Cox transformation, it can better accommodate the skewness of the error distribution and therefore serves as the gold standard in this scenario.

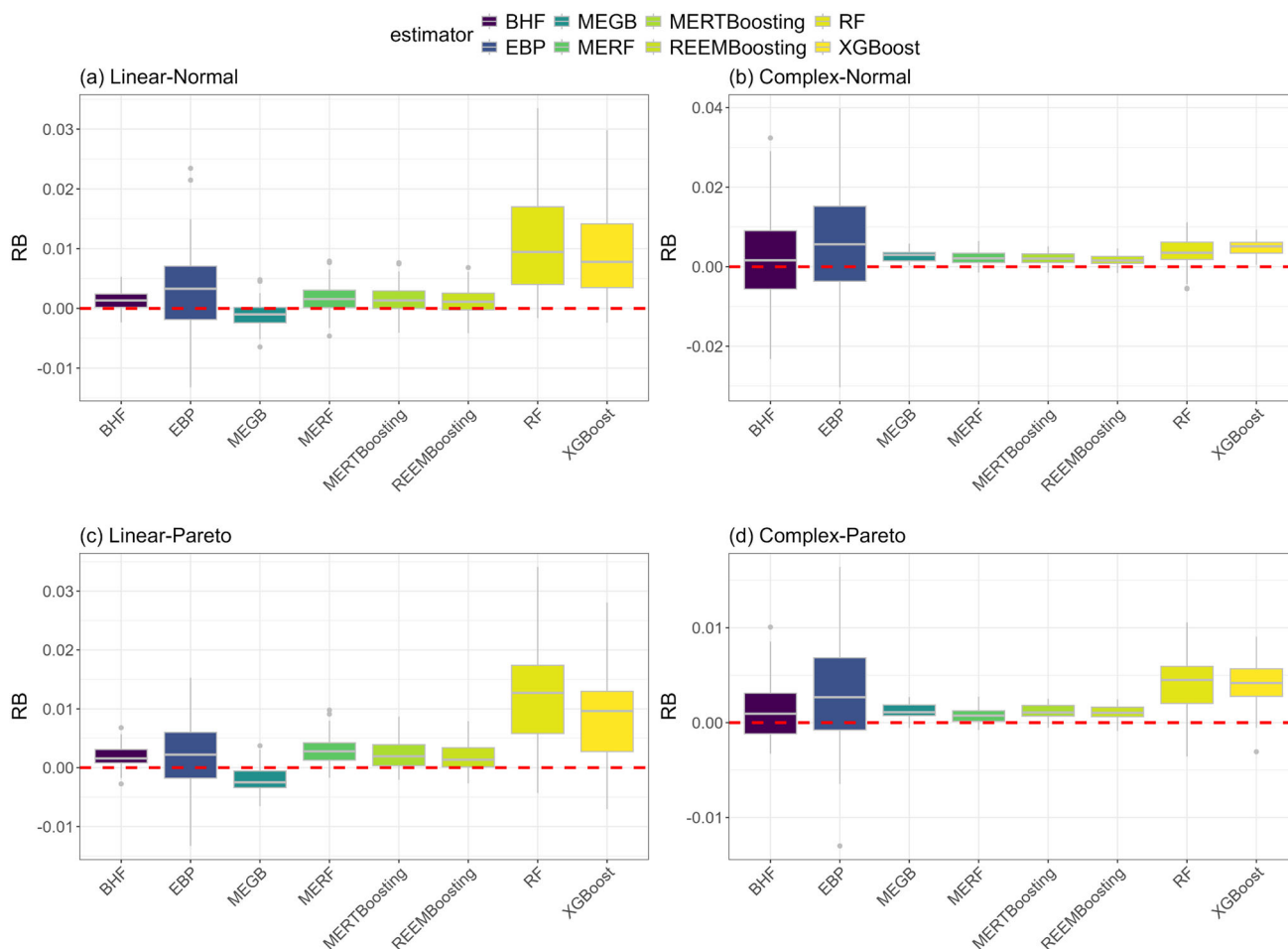
The EBP-BC achieves in this scenario a mean RRMSE of 3.51%, while MEGB and BHF have a very similar mean RRMSE with 3.78% and 3.77% respectively, as is shown in Table 3. Furthermore, it is shown in Fig. 2 that the MERF performs slightly worse than the other hierarchical estimators in this scenario. Again, XGBoost shows the highest mean RRMSE with 10.44%.

The RRMSE seems to follow a similar pattern across the four estimators in the case of the Complex-Normal and Complex-Pareto scenarios. In both scenarios, machine learning methods with consideration of a random effect (MERF, MEGB, MERTBoosting and REEMBoosting) have been shown to outperform the BHF and EBP-BC. The MERF, MEGB, MERTBoosting and REEMBoosting seem to have a higher estimation quality than BHF and EBP-BC; in addition, the MEGB, MERTBoosting and REEMBoosting have a lower RRMSE than the MERF. In the case of the Complex-Pareto scenario, the MEGB has a mean RRMSE of 1.11% as opposed to the MERF's mean RRMSE of 1.37%. The MERTBoosting and REEMBoosting estimators achieve comparable results with mean RRMSE values of 1.21% and 1.21%, respectively, indicating that all hierarchical boosting variants perform at a similarly high level of accuracy. Again, this result matches our prior expectations, as one strength of the proposed MEGB is that complex relationships are detected automatically and therefore have no negative impact on the performance of the MEGB. Using residuals, which are not normally distributed, seems to have very little effect on the estimation quality of the MERF, MEGB, MERTBoosting or REEMBoosting, as can be seen in the scenarios using Pareto distributed error terms. Furthermore, the results indicate that RF and XGBoost consistently underperform across all scenarios when random effects are not considered. This finding is consistent with the conclusions drawn by (Salditt et al. 2023) and underscores the importance of adequately modeling random effects in the context of SAE.

Table 4 and Fig. 3 summarize the accuracy of the proposed bootstrap MSE estimation method. Figure 3 shows the empirical RMSE and the corresponding RMSE estimated by the bootstrap procedure across all four scenarios. The results indicate that the proposed MSE estimation procedure is able to handle different levels of complexity effectively, as

Table 1 The four scenarios used in the model-based simulation study

Scenario	Model	x_1	x_2	μ	ϑ	ϵ
Linear-Normal	$y = 5000 - 500x_1 - 500x_2 + \vartheta + \epsilon$	$N(\mu, 3^2)$	$N(\mu, 3^2)$	$unif(-1, 1)$	$N(0, 500^2)$	$N(0, 1000^2)$
Complex-Normal	$y = 15000 - 500x_1 \cdot x_2 - 250x_2^2 + \vartheta + \epsilon$	$N(\mu, 4^2)$	$N(\mu, 2^2)$	$unif(-1, 1)$	$N(0, 500^2)$	$N(0, 1000^2)$
Linear-Pareto	$y = 5000 - 500x_1 - 500x_2 + \vartheta + \epsilon$	$N(\mu, 3^2)$	$N(\mu, 3^2)$	$unif(-1, 1)$	$N(0, 500^2)$	$Par(3, 800)$
Complex-Pareto	$y = 20000 - 500x_1 \cdot x_2 - 250x_2^2 + \vartheta + \epsilon$	$N(\mu, 2^2)$	$N(\mu, 2^2)$	$unif(-1, 1)$	$N(0, 1000^2)$	$Par(3, 800)$

**Fig. 1** RB averaged over the Monte-Carlo simulation runs for the estimators BHF, EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting, RF and XGBoost for all four scenarios**Table 2** Mean and median RB in % for the estimators BHF, EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting, RF and XGBoost across the four scenarios

RB (%)	Linear-Normal		Complex-Normal		Linear-Pareto		Complex-Pareto	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
BHF	0.133	0.133	0.239	0.161	0.190	0.157	0.127	0.094
EBP-BC	0.279	0.329	0.576	0.565	0.190	0.221	0.308	0.268
MEGB	-0.095	-0.100	0.275	0.298	-0.205	-0.249	0.116	0.111
MERF	0.166	0.156	0.237	0.210	0.310	0.277	0.072	0.074
MERTBoosting	0.154	0.134	0.208	0.204	0.225	0.192	0.110	0.107
REEMBoosting	0.120	0.111	0.162	0.157	0.191	0.135	0.102	0.104
RF	1.061	0.943	0.362	0.350	1.238	1.271	0.424	0.450
XGBoost	0.916	0.778	0.485	0.512	0.857	0.962	0.412	0.419

Table 3 Mean and median RRMSE in % for the estimators BHF, EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting, RF and XGBoost across the four scenarios

RRMSE (%)	Linear-Normal		Complex-Normal		Linear-Pareto		Complex-Pareto	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
BHF	3.907	3.548	3.652	3.602	3.770	3.328	2.755	2.645
EBP-BC	3.986	3.614	3.660	3.593	3.512	3.103	2.630	2.462
MEGB	4.008	3.654	1.590	1.505	3.781	3.347	1.107	0.997
MERF	4.187	3.842	2.171	2.066	4.021	3.586	1.372	1.278
MERTBoosting	4.016	3.653	1.828	1.772	3.767	3.337	1.211	1.111
REEMBoosting	4.061	3.709	1.870	1.767	3.811	3.381	1.213	1.109
RF	10.445	10.192	3.662	3.659	10.404	9.977	5.291	5.289
XGBoost	10.527	10.250	3.684	3.687	10.438	10.057	5.339	5.329

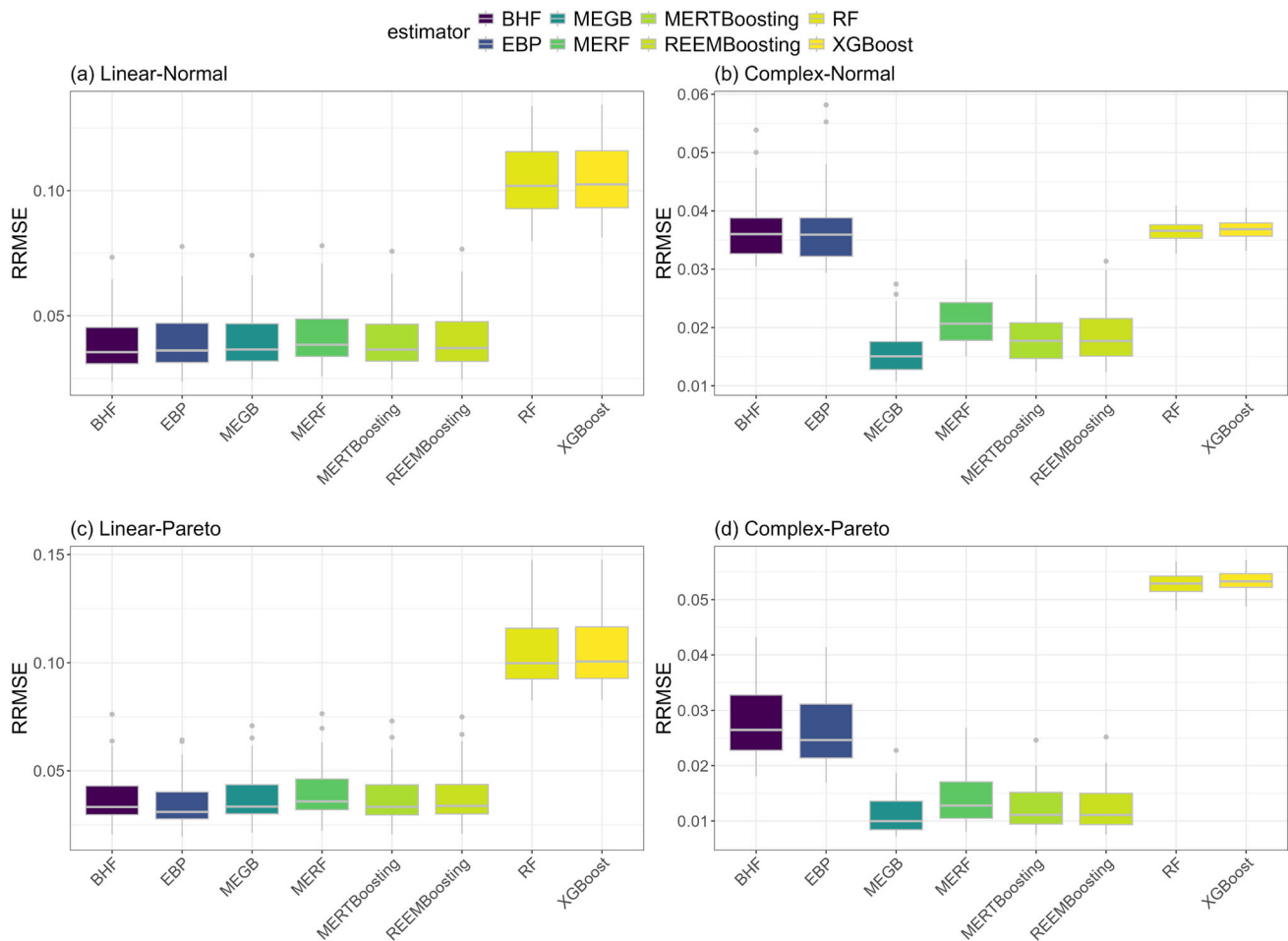
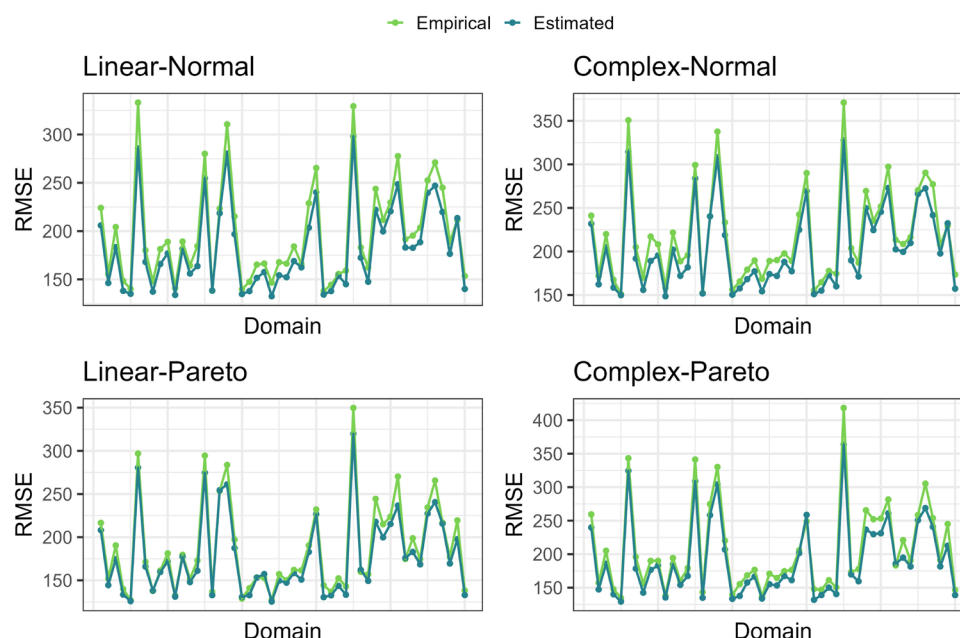


Fig. 2 RRMSE averaged over the Monte-Carlo simulation runs for the estimators BHF, EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting, RF and XGBoost for all four scenarios

Table 4 Mean and median results of the proposed MSE estimation via RB-RMSE and RRMSE-RMSE for the four scenarios

	Linear-Normal		Complex-Normal		Linear-Pareto		Complex-Pareto	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
RB-RMSE (%)	-6.8	-7.2	-6.1	-6.2	-4.2	-3.9	-6.5	-6.3
RRMSE-RMSE (%)	15.0	14.6	13.7	13.4	42.7	41.7	38.9	37.6

Fig. 3 Comparison of (estimated) REBB-based RMSE and empirical RMSE from Monte-Carlo simulations across all scenarios



the estimated RMSE closely matches the empirical RMSE, despite a minimal systematic underestimation. Additionally, Table 4 provides the RB-RMSE and RRMSE-RMSE values for each scenario, further confirming that the bootstrap method delivers reliable uncertainty estimates across all considered scenarios.

As a further robustness check, we examined the sensitivity of the performance gap between MEGB and MERF to the signal-to-noise ratio (SNR). Starting from the Complex-Normal scenario of model-based simulation A and leaving the data-generating mechanism otherwise unchanged, we varied the absolute levels of the residual and random-effects variance while holding the ICC fixed at 0.2, yielding a high-SNR and a low-SNR variant of this reference scenario. Each variant is based on the same $M = 500$ Monte Carlo replications as the remainder of model-based simulation A. We additionally included a tuned version of MERF for comparison, retuning its hyperparameters separately for the high- and low-SNR scenarios via the same five-fold cross-validation procedure used elsewhere in the paper (MEGB itself was not retuned for this analysis). Detailed results are provided in Appendix Fig. 9 and Table 11: MEGB retains an advantage over both the default and the tuned MERF across all signal-to-noise levels considered, though this advantage narrows under the low-SNR condition. Under the low-SNR condition the distributions of the area-level RRMSE overlap substantially across the three estimators, so that the advantage of MEGB, while still present on average, is of limited practical relevance in this regime.

4.3 Model-based simulation B

The simulation scenarios of model-based simulation B are based on the experimental design of Salditt et al. (2023), but are adapted to an SAE context. While the same data-generating process as in Salditt et al. (2023) is applied, the simulation design is adapted to an SAE framework that distinguishes between sample and population data. This extension allows the evaluation of estimator performance on the area-level rather than on the unit-level. The study examines the robustness of the estimators with respect to varying levels of intra-class correlation (ICC), the number of domains (D) and the within-domain sample sizes (n_d). In this simulation, the boosting-based estimators considered by Salditt et al. (2023) are compared with our MEGB approach, the EBP with data-driven Box-Cox transformation and the Small Area MERF by Krennmair and Schmid (2022). For MERTBoosting, REEMBoosting, and GB (the basic GB implementation of Salditt et al. (2023)), we use the hyperparameters reported in Salditt et al. (2023), adopting the values tuned in their study. The same applies to MBOOST and MBOOST_T, as well as RF, for which we likewise follow the configurations specified in Salditt et al. (2023). In contrast, MEGB is used without additional tuning specific to model-based simulation B; we retain the hyperparameter settings from model-based simulation A.

4.3.1 Simulation setup

This simulation setup follows the general data-generating process proposed by Salditt et al. (2023), which is extended

here to an SAE context. Specifically, a finite population is generated according to the same underlying model structure and a stratified sample is subsequently drawn from this population to estimate the models. The boosting-related unit-level models described by Salditt et al. (2023) are then employed to generate predictions for all population units. These unit-level predictions are subsequently aggregated to obtain area-level estimates, allowing for a direct evaluation of small area estimators based on the true population means. In contrast to Salditt et al. (2023), who implemented a MERF version tailored for general hierarchical data, this study employs the MERF approach developed by Krennmair and Schmid (2022), which is explicitly designed for SAE. The simulation setup for model-based simulation B is designed to systematically evaluate the performance of the MEGB under varying data-generating conditions. It follows a fully factorial design, where several key parameters of the hierarchical data structure are varied, including the number of domains, the average number of observations per domain and the ICC. Specifically, the number of predictors is fixed at $p = 9$, which are jointly generated from a multivariate normal distribution with mean vector $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$. The diagonal elements of $\boldsymbol{\Sigma}$ are set to 1, while the off-diagonal elements are 0.4, inducing moderate correlation among predictors. For each scenario, the ICC takes values of 0.05, 0.25, and 0.5, representing low, medium, and high levels of area dependence. The residual variance is fixed at $\sigma_\epsilon^2 = 1$. Thus, the residual noise level is held fixed across scenarios, and only the relative contribution of the random-effects variance to this fixed residual level is systematically varied through the ICC. As is inherent in this parameterization, adopted from Salditt et al. (2023), the total error variance therefore also increases with the ICC, so that the design varies the relative contribution of the random effects and the overall noise level jointly rather than separately. Two different domain configurations are considered, with $D \in \{100, 250\}$ domains, and mean domain sample sizes of $\bar{n}_d \in \{10, 30\}$. Each domain contains $N_d = 300$ population units, corresponding to total population sizes of $N = 30,000$ and $N = 75,000$ for $D = 100$ and $D = 250$, respectively. Combining the levels of D , $\bar{n}_d$, and the ICC results in $2 \times 2 \times 3 = 12$ distinct simulation scenarios. For each scenario, $M = 100$ Monte-Carlo repetitions are conducted. The outcome variable Y_{id} is generated according to a nonlinear Mixed Effects model of the form

$$Y_{id} = g(\mathbf{X}_{id}) + \vartheta_d + \varepsilon_{id},$$

where $g(\mathbf{X}_{id}) = 2X_1 + X_2^2 + 4I(X_3 > 0) + 2\log(|X_1|)X_3$ represents a nonlinear mean function of the predictors, $\vartheta_d \sim N(0, \sigma_\vartheta^2)$ denotes the random effect for domain d , and $\varepsilon_{id} \sim N(0, \sigma_\epsilon^2)$ are independent residuals. The random effect variance σ_ϑ^2 is determined by the chosen ICC value according to

$$\sigma_\vartheta^2 = \frac{\text{ICC} \cdot \sigma_\epsilon^2}{1 - \text{ICC}}.$$

All outcome values are truncated at a minimum of 0.01 to avoid negative values. For each domain, a stratified simple random sample of size n_d is drawn from the population, where n_d varies uniformly around the target mean $\bar{n}_d$ to introduce heterogeneity in domain-specific sample sizes. In total, the simulation design allows assessing the performance of the MEGB across varying hierarchical structures, predictor correlations, and levels of between-domain variability, while maintaining full control over the data-generating process.

4.3.2 Simulation results

In this subsection, the results of the simulation described in Subsection 4.3.1 will be discussed and evaluated using the RB and the RRMSE. Table 5 and Table 6 summarize the simulation results for $D = 100$ and $n_d = 30$, reporting the RRMSE and RB across different levels of the ICC. The results for the other scenarios, including different combinations of the number of domains ($D \in \{100, 250\}$), domain sizes ($n_d \in \{10, 30\}$), and ICC levels (0.05, 0.25, 0.5), are provided in the Appendix (see Tables 12, 13, 14, 15, 16 and 17). Across the experimental scenarios the three hierarchical boosting estimators that combine GB with random effects, namely *MEGB*, *MERTBoosting*, and *REEMBoosting*, achieve the overall best performance. This finding is consistent with the expected behavior given their conceptual similarities and aligns well with the results reported by Salditt et al. (2023), who also emphasize the benefits of integrating GB with mixed-effects structures. For scenarios with smaller within-domain sample sizes ($n_d = 10$), all three estimators yield nearly identical results at each ICC level, with median RRMSE rising from about 5% at ICC = 0.05 to about 10–11% at ICC = 0.5, and absolute bias below 1% throughout; as noted in Subsection 4.3.1, part of this increase reflects the simultaneous rise in total error variance with the ICC, rather than the strength of the hierarchical structure alone. This pattern is observed for both $D = 100$ and $D = 250$. When the sample size increases ($n_d = 30$) and the number of domains grows to $D = 250$, the proposed *MEGB* shows a slight advantage, reaching a median RRMSE of 3.80% at ICC = 0.05 compared with 3.96% for *MERTBoosting* and 3.95% for *REEMBoosting*, and retaining this margin at the higher ICC levels. While the differences are small, they indicate a marginal improvement in estimation efficiency for MEGB when the data exhibits stronger hierarchical patterns and larger within-domain sample sizes. Nevertheless, *MEGB*, *MERTBoosting*, and *REEMBoosting* achieve slightly lower RRMSE and RB values in most settings compared to the non-hierarchical benchmark models, particularly when the number of observations per domain is

Table 5 RB results (in %) for 100 domains with $n_d = 30$ observations per domain

RB (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	2.16	2.10	2.49	2.60	2.71	2.63
MBOOST	0.82	0.82	1.73	1.78	4.91	4.83
MBOOST_T	0.89	0.86	1.81	1.85	5.11	5.10
MEGB	−0.11	−0.04	0.04	0.07	0.04	0.03
MERF	0.39	0.38	0.51	0.51	0.48	0.43
MERTBoosting	−0.05	−0.05	0.12	0.12	0.10	0.11
REEMBoosting	−0.10	−0.08	0.15	0.12	0.05	0.11
RF	0.61	0.55	1.95	1.99	5.30	5.41
GB	0.10	0.09	1.56	1.61	5.04	5.16

Table 6 RRMSE results (in %) for 100 domains with $n_d = 30$ observations per domain

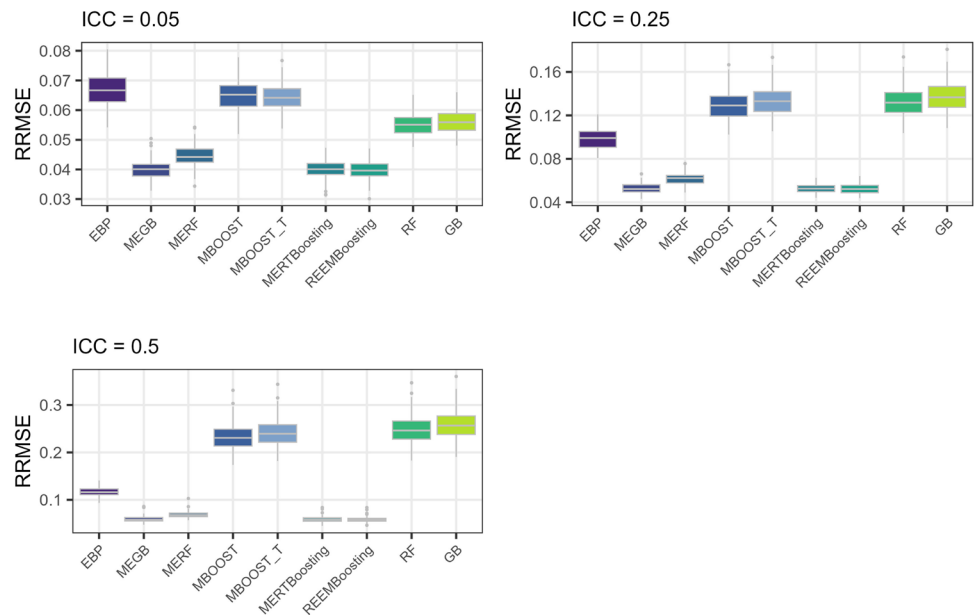
RRMSE (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	6.67	6.70	9.91	9.90	11.61	11.72
MBOOST	6.53	6.53	12.91	12.93	23.07	23.36
MBOOST_T	6.41	6.46	13.30	13.33	23.93	24.21
MEGB	4.00	4.00	5.21	5.24	5.83	5.91
MERF	4.42	4.46	6.22	6.18	6.79	6.89
MERTBoosting	3.98	4.00	5.18	5.26	5.86	5.88
REEMBoosting	4.00	4.00	5.23	5.25	5.83	5.88
RF	5.50	5.50	13.19	13.26	24.66	24.87
GB	5.56	5.59	13.70	13.75	25.63	25.85

small or when the correlation between units within domains is high. These findings suggest that the advantages of integrating GB with random effects, as demonstrated by Salditt et al. (2023) in a unit-level context, also hold in the SAE setting. Figure 4 provides a graphical summary of the simulation outcomes, illustrating the RRMSE across ICC levels for $D = 100$ and $n_d = 30$. The figure visually confirms that hierarchical boosting estimators consistently outperform both classical and non-hierarchical approaches, with performance gaps widening as the ICC increases.

Across all experimental settings and ICC levels, the three hierarchical boosting methods, MEGB, MERTBoosting, and REEMBoosting, consistently achieve the lowest RRMSE among all estimators considered, while remaining essentially unbiased. For low ICC (0.05), the gap between hierarchical and non-hierarchical models is comparatively small: MEGB, MERTBoosting, and REEMBoosting achieve the lowest errors (RRMSE $\approx 4\%$), but non-hierarchical models such as RF and plain GB-boosting yield slightly higher values (RRMSE $\approx 5\text{--}6\%$), whereas the EBP shows moderately larger errors in this scenario (RRMSE $\approx 6.7\%$; see Table 6). In terms of bias, MEGB, MERTBoosting, and REEMBoosting remain essentially unbiased (RB around 0%), while non-hierarchical learners show slightly positive RB (e.g.,

RF $\approx 0.6\%$). In the context of ICC levels that are particularly low, non-hierarchical learners show competitive performance. This is due to the weak hierarchical structure in the data, which reduces the benefit of explicitly modeling random effects. The performance advantage of MEGB, MERTBoosting, and REEMBoosting is particularly evident in scenarios with stronger hierarchical structure (ICC = 0.25 and 0.5), where they maintain RRMSE values around 5–6% and remain virtually unbiased. This stability across varying data complexities highlights the ability of these estimators to flexibly capture both fixed and random effects within the boosting framework. The tables in the Appendix (Tables 12, 13, 14, 15, 16 and 17) confirm that this consistent advantage is observed not only for $D = 100$ but also for larger domain settings ($D = 250$), underlining the scalability of these hierarchical boosting approaches. The RRMSE of non-hierarchical estimators increases substantially, reflecting their inability to account for the hierarchical correlation structure. Conversely, methods accounting for the hierarchical structure maintain low error levels and nearly unbiased estimates. Accordingly, the MERF model also yields competitive results and successfully captures a substantial portion of the hierarchical variance through its Mixed Effects structure. However, its error levels are consistently higher, typically

Fig. 4 RRMSE for 100 domains with $n_d = 30$ observations per domain, averaged over 100 Monte-Carlo simulation runs. RRMSE is displayed as a proportion; the corresponding values in per cent are reported in Table 6



by about 0.3 to 2 percentage points in RRMSE, and it shows slightly larger bias values across most ICC configurations. This pattern indicates that the boosting-based approaches (MEGB, MERTBoosting, REEMBoosting) may capture the underlying function $f(x)$ more flexibly than the MERF. By contrast, methods such as `mboost` and its tree-based variant (`mboost_T`) perform markedly worse across all ICC levels and domain settings. As illustrated in Tables 5 and 6, both variants exhibit pronounced increases in RRMSE, reaching values of up to about 26% at $\text{ICC} = 0.5$, and show stronger bias patterns, especially as the ICC grows. These findings align with the unit-level simulation evidence reported by Salditt et al. (2023), confirming that the inability of standard boosting algorithms to account for hierarchical dependencies leads to systematic underperformance when domain-level effects are present. Overall, the results across all configurations highlight that explicitly modeling the hierarchical structure within the boosting process, as implemented in MEGB, MERTBoosting, and REEMBoosting, yields consistently more accurate and unbiased domain-level estimates. Their robustness across ICC levels and within-domain sample sizes underscores their suitability for small area applications with hierarchical structures. When the domain size or total number of observations increases (see detailed results in Appendix Tables 12, 13, 14, 15, 16 and 17), MEGB further consolidates its advantage. Although MERTBoosting and REEMBoosting remain close competitors, MEGB tends to achieve marginally smaller RRMSE and RB values. In summary, the simulation results confirm that integrating random effects into GB substantially improves predictive accuracy and reduces estimation bias in SAE. This underscores the importance of incorporating the hierarchical data structure directly into the modeling process.

The MEGB estimator performs comparably to, and in several scenarios slightly better than, the related MERTBoosting and REEMBoosting approaches, while clearly outperforming both classical model-based estimators and pure machine learning methods, particularly when hierarchical dependencies are strong.

The mean computation times across Monte-Carlo simulation runs for each setting of model-based simulation B are reported in the Appendix (Table 20).

The results show the advantages of combining GB with Mixed Effects in an SAE context. To gain further insights into the performance of MEGB, a design-based simulation study was conducted in addition to the model-based studies, as described in Sect. 5.

5 Design-based simulation

5.1 Simulation setup

In this section, the performance of MEGB is tested and evaluated using real data. The proposed method is used to estimate the area-level average labour income of Mexican municipalities in Chihuahua, one of the 31 states of Mexico. Chihuahua is the largest state in Mexico in terms of area and is located in the north of the country. The state's unemployment rate is 2.62%, which is slightly lower than the Mexican average of 3%. Furthermore, Chihuahua is one of the country's leading exporters. In our analysis, a complete census is utilized as a pseudo-population, serving as a realistic proxy for the unknown true target population. From this census dataset, random samples are repeatedly drawn to mirror the income and expenditure household survey, whereas the com-

plete census is used for the evaluation of the design-based study. The census provides information on earned per capita income, and the main objective is to model per capita income derived from employment. The survey motivates the structure of the design-based simulation study. The objective of this study is to compare the main model-based estimators EBP-BC, MERF, MEGB, and MERTBoosting on real data with respect to their bias and empirical RMSE performance. These estimators were selected as they demonstrated the most favorable results in the model-based simulation study. The REEMBoosting approach was also tested; however, its results are reported in the Appendix because a robust configuration for its practical application in this setting could not be identified with confidence. The census dataset includes $N = 58,578$ households across all $D = 67$ municipalities, whereas the survey dataset contains $n = 1,280$ households from only 22 municipalities. On average, each surveyed municipality has approximately $\bar{n}_d = 58$ households, with 45 municipalities entirely Out-Of-Sample. Observations per municipality range from $\min(n_d) = 7$ to $\max(n_d) = 609$, with a $\text{median}(n_d) = 19.5$. The estimation involves a design-based Monte-Carlo simulation repeated $M = 500$ times. Each iteration involves drawing a new sample from the census dataset, precisely matching the number of observations in the actual survey ($n = 1,280$), thus replicating the survey's sampling structure in size and municipal distribution.

We use the same hyperparameter configuration as described in Subsection 4.2.1, with the exception of n_{rounds} , which is determined dynamically in Algorithm 1 via early stopping. In addition to the previously utilized hyperparameters, the hyperparameter *colsample_bytree* was introduced because the design-based simulation involves a substantially larger set of covariates compared to the model-based study. This hyperparameter determines the subsample ratio of columns used when building each tree and is set to 0.4.

5.2 Simulation results

The bias and the empirical RMSE are used to assess the performance of the MEGB in relation to the other estimators (EBP-BC, MERF and MERTBoosting). The Horvitz-Thompson estimator (Horvitz and Thompson 1952), denoted as HT, is also included in the analysis as a direct estimator. The HT mean estimator is denoted as $\hat{\mu}_d^{HT} = N_d^{-1} \sum_{i=1}^{n_d} \frac{y_{d,i}}{\pi_{d,i}}$, where $\pi_{d,i}$ is the probability that unit d, i is included in the sample. It should be noted that the HT-estimator is only included for the In-Sample data, which consists of 22 municipalities. This limitation arises because the HT-estimator is applicable only when the data points, represented by $y_{d,i}$, are directly accessible for the estimation process. The empirical RMSE is calculated using the empir-

ical true values, where the empirical true values are defined as the area-level averages of household incomes from labour in the census data used. The design-based simulation also enables the analysis of Out-Of-Sample areas, in addition to the In-Sample areas.

The bias of the mean area-level estimates is shown in Fig. 5. In terms of bias, the MEGB outperforms the EBP-BC and MERF for all domains as well as for the Out-Of-Sample domains. The MERTBoosting estimator achieves bias levels similar to MEGB, which is consistent with their conceptual similarities, as both approaches integrate Mixed Effects estimation within a boosting framework. This difference in estimation quality is underlined in Table 7. Figure 5 shows that the HT has the lowest median and mean bias for In-Sample areas (as expected for an unbiased estimator). In contrast, MERF shows the highest bias for In-Sample and Out-Of-Sample municipalities and has therefore also in total the highest bias. The bias is across EBP-BC, MEGB, MERF and MERTBoosting mostly positive.

The empirical RMSE of the mean area-level estimates is presented in Fig. 6. An examination of the RMSE also confirms the impression of the bias evaluation. The MEGB has a lower empirical median RMSE than the EBP-BC, MERF, MERTBoosting and HT for the In-Sample domains and the MEGB's median RMSE is also in the Out-Of-Sample domains and for all domains lower than the median RMSE of EBP-BC, MERF and MERTBoosting. This finding is consistent with the expected behavior given their conceptual similarities. In this more complex real-data setting with additional covariates, the advanced regularization and sampling features of the XGBoost framework may contribute to slightly improved generalization compared to the Salditt et al. (2023) implementation on which MERTBoosting is based. While the median RMSE of EBP-BC and MERF seems to be on a similar level for all domains, the median RMSE of the EBP-BC seems to be higher than MERF's for the In-Sample areas and lower than the MERF for the Out-Of-Sample areas. Table 8 shows that the MEGB performs better than the MERF and EBP-BC and has the lowest mean and median empirical RMSE across all areas. MEGB performs especially for the 45 Out-Of-Sample municipalities better than the MERF, where the difference between the mean and median empirical RMSE for the MEGB and the MERF is the highest. The mean empirical RMSE is in this case 255.27 while the median empirical RMSE is 205.58 for the MEGB, while the MERF has a mean empirical RMSE of 315.70 and a median empirical RMSE of 288.92. This indicates a good generalizability of the MEGB, which is an important feature for Out-Of-Sample estimations. MERTBoosting shows a nearly identical level of generalization, confirming that both methods effectively balance fixed and random effects through the boosting process. However, MEGB slightly benefits from the optimized tree construction and regularization mechanisms

Fig. 5 Bias for In-Sample-, and Out-Of-Sample- and all municipalities in total in Chihuahua for the EBP-BC, MEGB, MERF, MERTBoosting and the direct estimator HT

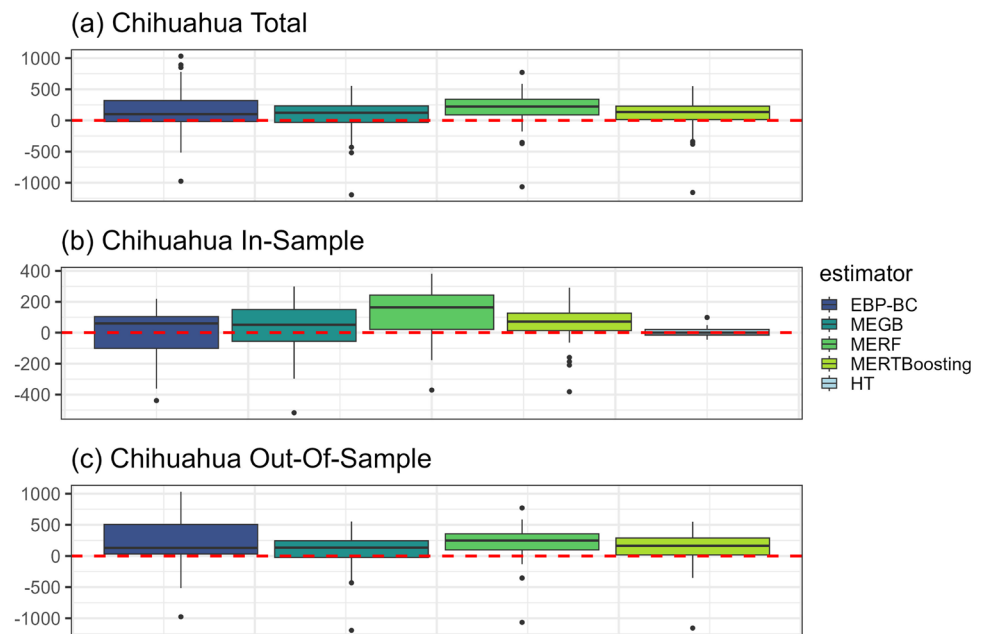


Table 7 Mean and median bias for In-Sample-, and Out-Of-Sample- and all municipalities in total in Chihuahua for the EBP-BC, MEGB, MERF, MERTBoosting and the direct estimator HT in the design-based simulation study

Bias	Total		In-sample		Out-of-sample	
	Mean	Median	Mean	Median	Mean	Median
EBP-BC	153.616	101.327	-9.346	59.876	233.286	130.207
MEGB	74.782	122.483	23.317	50.743	99.942	134.758
MERF	183.743	222.718	126.816	164.079	211.573	247.918
MERTBoosting	99.522	135.044	42.136	71.437	127.578	163.634
HT	—	—	3.677	-0.181	—	—

inherent to XGBoost, which appear to be advantageous in this higher-dimensional setting. This aligns with the expectation that GB is an effective method to make Out-Of-Sample predictions (Merfeld and Newhouse 2023). In terms of the empirical RMSE, MEGB performs consistently better than EBP-BC across the In-Sample and Out-Of-Sample areas and all areas in total.

The point estimates of the area means, which are shown in Fig. 7, track the empirical means well. This is especially the case for the In-Sample areas. In addition, it is shown that the estimations for the Out-Of-Sample areas are close to the empirical values and provide correspondingly good area mean estimates, which highlights the generalizability and usability of the MEGB.

The RB-RMSE and RRMSE-RMSE for the MEGB can be seen in Table 9 which allows the analysis of the proposed MSE estimation method. The table shows that the MSE estimations achieve lower results in In-Sample areas than in Out-Of-Sample areas. This aligns with the expectations made beforehand, as the estimation of the Out-Of-Sample areas is solely based on the GB algorithm $\hat{f}(x)_{MEGB}$ and no estimation of the random effect $\hat{\vartheta}_{OOS}$ can be performed. Overall,

the RB of the RMSE estimates seems to be closely around 0, which suggests that the RMSE is very well tracked.

Looking at the RB-RMSE estimates separately for In-Sample and Out-Of-Sample areas, it appears that the RMSE is slightly underestimated for In-Sample areas with a median of -9.2%, and slightly overestimated for Out-Of-Sample areas with a median of 5.2% and a mean of 6.5%.

6 Conclusion and future research

Building on the estimation approach of the MERF, the aim was to develop an estimation method that meets the requirements of flexible, powerful machine learning methods combined with Mixed Effects. To achieve this, the method MEGB was introduced. Combining the MEGB with REBB creates a flexible estimator, which can be used without the constraints imposed by distributional assumptions. This flexibility distinguishes the MEGB, as well as the MERF, from other estimation methods, as it automatically detects and exploits (higher-order) relations between predictive covariates. This feature significantly lowers the risk of model

Table 8 Mean and median RMSE for the estimators EBP-BC, MEGB, MERF, MERTBoosting and the direct estimator HT and In-Sample-, and Out-Of-Sample- and all municipalities in total

RMSE	Total		In-sample		Out-of-sample	
	Mean	Median	Mean	Median	Mean	Median
EBP-BC	330.791	258.624	275.018	265.135	358.057	249.446
MEGB	236.396	177.989	197.791	165.089	255.269	205.577
MERF	288.485	249.212	232.813	200.395	315.703	288.919
MERTBoosting	250.198	205.001	206.130	176.091	271.743	211.556
HT	—	—	485.958	471.165	—	—

Fig. 6 RMSE averaged over the Monte-Carlo simulation runs for the estimators EBP-BC, MEGB, MERF, MERTBoosting and the direct estimator HT

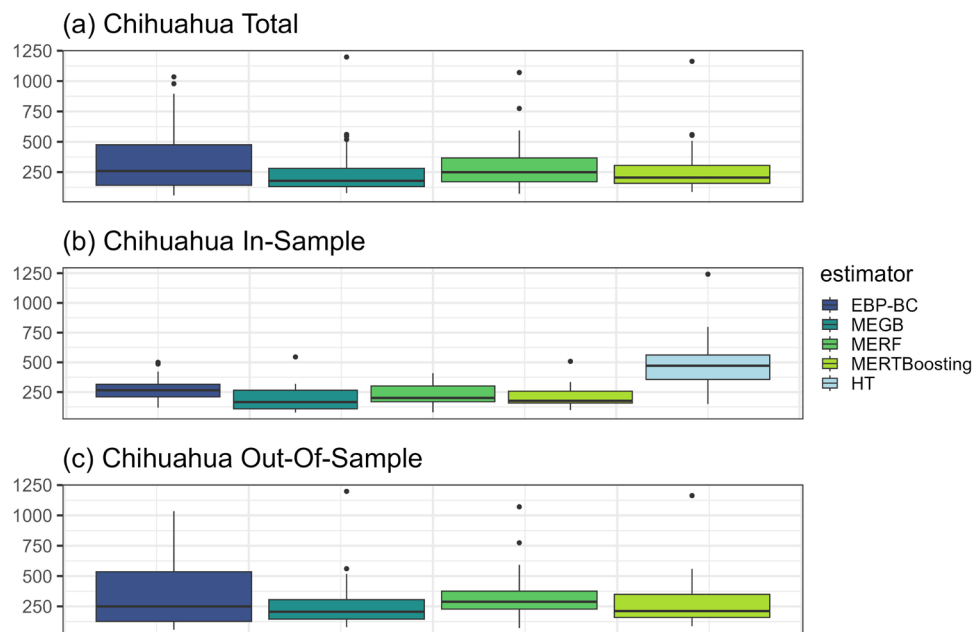


Fig. 7 Point estimates of the average household income per area compared to the empirical average income per area for In-Sample and Out-Of-Sample areas



Table 9 Mean and median results of the proposed MSE estimation with $B = 100$ through RB-RMSE and RRMSE-RMSE for the In-Sample and Out-Of-Sample areas and in total

	Total		In-sample		Out-of-sample	
	Mean	Median	Mean	Median	Mean	Median
RB-RMSE (%)	4.2	−4.2	−0.5	−9.2	6.5	5.2
RRMSE-RMSE (%)	57.1	45.9	51.3	44.4	60.0	49.6

misspecification, as is the case with a few other estimators. Therefore, the MEGB method is also suitable for scenarios with high-dimensional problems processing big data sources and non-normally distributed residuals and yields accurate estimation results. The results of this paper indicate that the combination of the mentioned GB model advantages together with the ability to control for dependency structures builds a powerful method for SAE. The evaluation of the model-based scenarios and the design-based simulation shows that the MEGB produced good point estimates for the area means across all scenarios. Across both model-based studies (A and B) and the design-based study, MEGB consistently performed at least as well as MERF in terms of RRMSE, with the largest and most practically relevant improvements arising in settings with pronounced nonlinear structures or a higher SNR, while differences were smaller and less systematic under low-SNR conditions. This suggests practitioners can apply MEGB to obtain reliable area estimates even when model assumptions fail, without sacrificing inferential capabilities (through our bootstrap MSE).

Looking ahead, further research could explore extending the proposed method to binary or count data, as well as examine the potential of the MEGB to estimate nonlinear indicators such as head-count ratios or Gini coefficients, as such indicators often hold high relevance in socioeconomic analyses. Additionally, our research results could be used for a comparative analysis of the performance of different GB hyperparameter tuning approaches. Furthermore, the integration of parameter tuning within the EM algorithm, as well as an integration in the bootstrap of the MSE estimations, should be discussed. However, it is essential to note that such an integration may introduce computational challenges due to increased processing time, which requires careful consideration.

Another possible research aspect of interest could be to include further machine learning methods in the EM algorithm, such as SVMs. Because of SVMs' capability to handle nonlinear data through the use of kernel methods, their inclusion warrants further examination in future studies. Another promising avenue for further investigation would be to attempt the implementation of CatBoost (Dorogush et al. 2018), LightGBM (Ke et al. 2017) or UnbiasedGBM (Zhang et al. 2023) within the EM algorithm. In particular, UnbiasedGBM represents a promising alternative due to the unbiased split criteria, which are important considerations

given that the bias that arises when finding the splits can lead to overfitting issues. The ability to generalise is also a crucial factor in the context of SAE. One potential approach for further research would be to apply model-boosting to a base learner that accounts for the hierarchical structure. Consequently, the EM algorithm is not required for the separation of fixed and random effect estimates. As outlined in Salditt et al. (2023), the BART yields better results than the GB with random effects and L2 loss in their particular unit-level comparison. It might therefore be interesting to compare this in a small area context as well. Transformations of the target variable could also be a promising approach to further enhance the performance of the MEGB method. Another potential direction for future research concerns the use of aggregated auxiliary information when census micro-data is not available. Recent work by Krennmair et al. (2026) has shown that calibration-based strategies can successfully integrate aggregated covariate information in RF-type models for SAE. Extending such ideas to the MEGB framework would be of considerable interest, particularly for applications where unit-level auxiliary data is inaccessible. Future research could also extend the MEGB framework towards robust SAE settings that explicitly account for atypical observations or outliers, for instance by incorporating robust loss functions such as the Huber loss (Huber 1964).

Appendix

Additional results for simulation A

As an additional robustness check, we provide a supplementary comparison between MEGB, MERF, and a tuned version of MERF for model-based simulation A. The tuned variant of MERF was obtained as follows. Its RF hyperparameters, *mtry* (the number of variables considered at each split), *min.node.size* (the minimum node size) and the number of trees, were selected via five-fold cross-validation on unit-level prediction error. The resulting configuration (*mtry* = 2, *min.node.size* = 20, 800 trees) was then applied as a single fixed setting across all four scenarios, mirroring MEGB's strategy of a single fixed configuration throughout. For the signal-to-noise sensitivity analysis reported below, MERF's hyperparameters were instead retuned separately for each of the two SNR variants via the same five-fold cross-validation

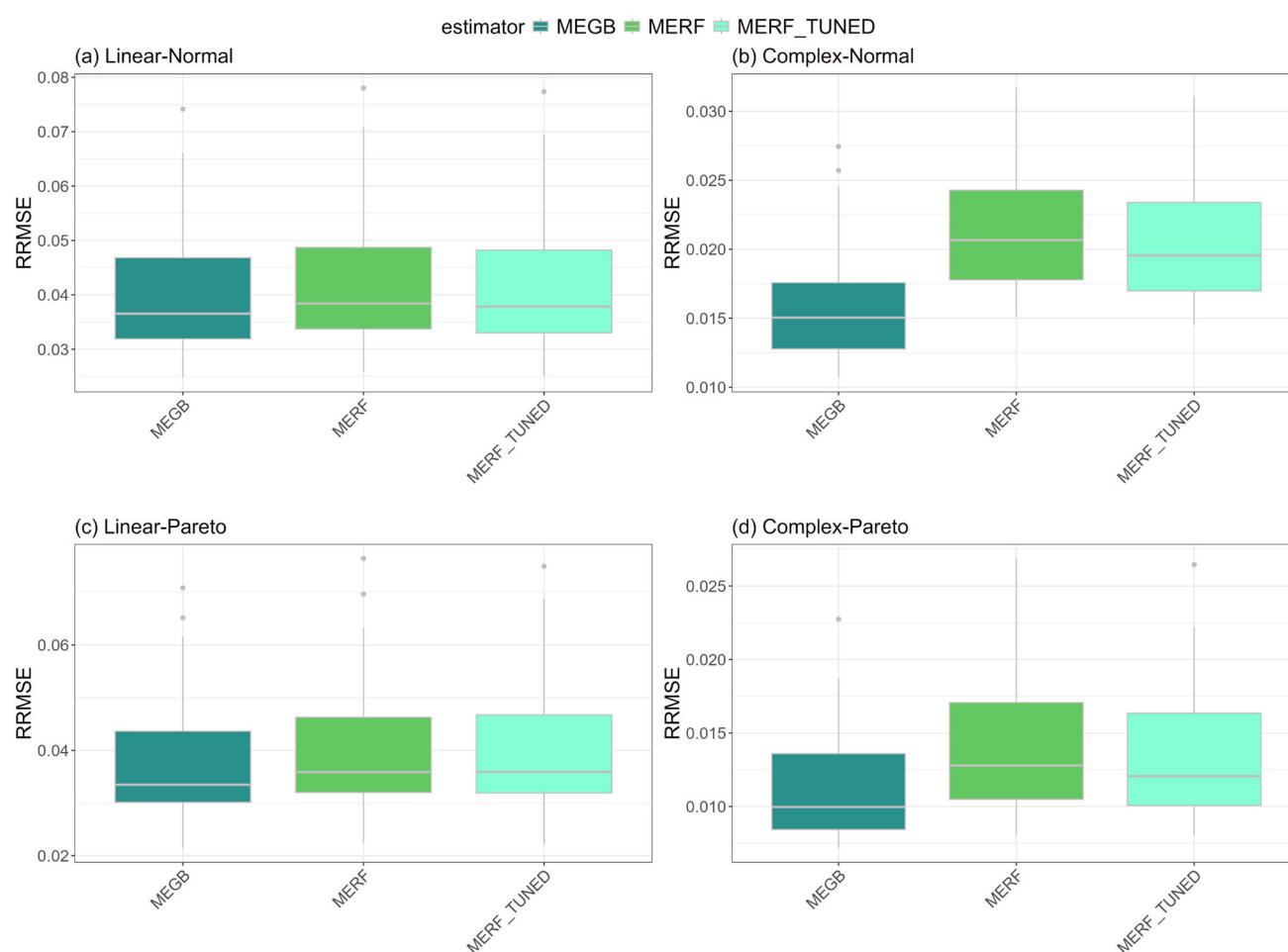


Fig. 8 RRMSE averaged over Monte Carlo simulation runs for the estimators MEGB, MERF, and a tuned version of MERF across the four scenarios of model-based simulation A. RRMSE is displayed as a proportion. Details of the MERF tuning procedure are given in the text of this subsection

procedure, with the number of trees held fixed at 500: the resulting configurations were $mtry = 2$, $min.node.size = 5$ (High-SNR) and $mtry = 2$, $min.node.size = 10$ (Low-SNR). The default (untuned) MERF configuration follows Krennmaier and Schmid (2022) ($mtry = 1$, $min.node.size = 5$, 500 trees). All replications completed under both configurations. Figure 8 shows that additional tuning of MERF does not materially change its performance relative to MEGB and therefore does not affect the substantive conclusions drawn in the main text.

The source code accompanying Asadi et al. (2026) is publicly available. We adapted their GMEXGBoost implementation to our continuous SAE setting and added our own aggregation of unit-level predictions to domain means. Hyperparameters were selected by a grid search over 84 configurations, evaluated on a held-out validation split of one Monte Carlo replication and restricted to configurations for which the algorithm's iterative procedure converged; the selected configuration was then held fixed across all four scenarios, as for MEGB, though on a smaller tuning bud-

get than the grid search described in Subsection 4.2.1. We evaluated the method on all four scenarios of model-based simulation A ($M = 500$ replications each), alongside MEGB and MERF for comparison. Table 10 reports the results. In terms of RRMSE, GMEXGBoost was outperformed by both MEGB and MERF in all four scenarios, with a small gap under the linear data-generating processes and a substantially larger one once nonlinearity is present; this is the pattern that the structure of the published algorithm would lead one to expect, and it is consistent with the observation of Asadi et al. (2026) that, in their simulation study, the method does not universally outperform plain XGBoost in terms of raw predictive error, its strength lying in scenarios where the correlation between observations is substantial. Relative bias is negligible for all three estimators throughout, and GMEXGBoost attains the smallest absolute bias in three of the four scenarios; the differences between the methods are therefore driven by variance rather than by bias. In the Complex-Normal scenario, 30 of 500 replications did not complete and are excluded from the corresponding row. We

Table 10 Mean RB (in %) and mean RRMSE (in %) for GMEXGBoost, MEGB, and MERF across the four scenarios of model-based simulation A ($M = 500$ replications, except where noted)

Scenario	GMEXGBoost		MEGB		MERF	
	RB	RRMSE	RB	RRMSE	RB	RRMSE
Linear-Normal	−0.04	4.39	−0.10	4.01	0.17	4.19
Complex-Normal	0.22	3.70	0.28	1.59	0.24	2.17
Linear-Pareto	0.03	4.26	−0.21	3.78	0.31	4.02
Complex-Pareto	0.10	2.84	0.12	1.11	0.07	1.37

For GMEXGBoost in the Complex-Normal scenario, 30 of 500 replications failed to complete and are excluded ($M = 470$)

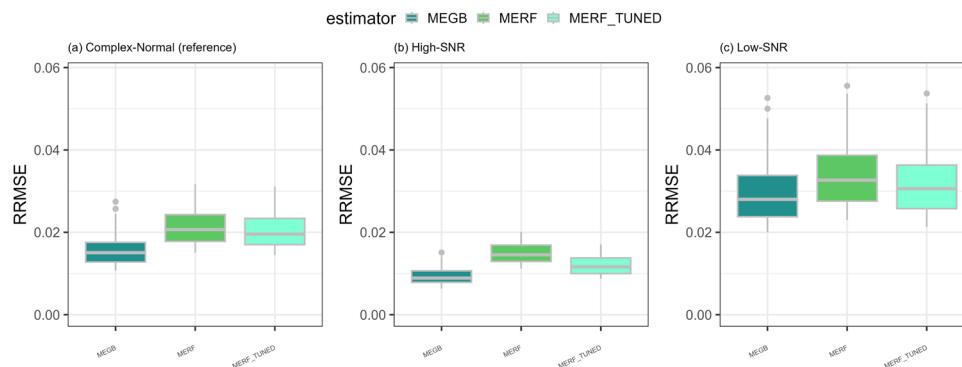


Fig. 9 Sensitivity of area-level RRMSE to the SNR for the estimators MEGB, MERF, and a tuned version of MERF, under the Complex-Normal data-generating process with fixed ICC (0.2) but varying absolute noise levels, as described in Sect. 4.2. RRMSE is displayed

as a proportion; the corresponding values in per cent are reported in Table 11. All panels share a common vertical scale and are based on $M = 500$ Monte Carlo replications per scenario

did not identify the cause and cannot rule out that it lies in our adaptation to a continuous response rather than in the published method.

Two qualifications apply. First, Asadi et al. (2026) evaluate their method only for binary outcomes; applying it to a continuous target is consistent with the scope the authors describe for the framework, but lies outside the setting in which it was validated, and should be borne in mind when interpreting the results. Second, we restrict the comparison to model-based simulation A. The four scenarios of that study vary both functional form and error distribution, and the structural property described in Subsection 2.5 is not specific to a particular data-generating process, however, we did not evaluate GMEXG Boost on model based simulation B or on the design-based study, and we therefore do not extend our conclusions to those settings.

Additional results for simulation B

Additional results for the design-based study

This subsection reports the results of the design-based study including REEMBoosting, which is omitted from the corresponding figures and tables in Sect. 5 for the reason given there. Figures 10 and 11 show the bias and the RMSE aver-

aged over the Monte-Carlo simulation runs, and Tables 18 and 19 report the corresponding mean and median values.

Convergence and stability of the iterative estimators

Several of the estimators compared in this study are iterative: MEGB, MERF, MERTBoosting and REEMBoosting, the EBP with a data-driven Box–Cox transformation, and, in model-based simulation B, MBOOST, MBOOST_T and GB. The GMEXGBoost implementation examined in the comparison reported above is iterative as well; it is discussed separately below. For MEGB, convergence of the EM-type procedure in Algorithm 1 is monitored through the marginal change in the modified GLL criterion, with a tolerance of 10^{-4} and a maximum of 25 iterations. Across all simulation studies (the 2000 replications of model-based simulation A, the further 1000 replications of the two signal-to-noise variants, the 1200 replications of model-based simulation B (12 scenarios with $M = 100$ each) and the 500 replications of the design-based study) every replication of these estimators completed and returned area-level estimates, and no runtime failures occurred. We did not record the number of iterations required in individual replications and therefore do not report how often the iteration limit was reached.

Table 11 Mean RB (in %) and mean RRMSE (in %) for MEGB, MERF, and a tuned version of MERF across the signal-to-noise sensitivity scenarios (Complex-Normal reference: $\sigma_\epsilon = 1000$, $\sigma_\vartheta = 500$; High-SNR: $\sigma_\epsilon = 500$, $\sigma_\vartheta = 250$; Low-SNR: $\sigma_\epsilon = 2000$, $\sigma_\vartheta = 1000$; all with ICC = 0.2 and $M = 500$ Monte Carlo replications per scenario)

Scenario	MEGB		MERF		MERF (tuned)	
	RB	RRMSE	RB	RRMSE	RB	RRMSE
Complex-Normal (reference)	0.28	1.59	0.24	2.17	0.24	2.08
High-SNR	0.25	0.94	0.27	1.50	0.03	1.20
Low-SNR	0.32	3.00	0.25	3.42	0.12	3.21

Table 12 RB results (in %) for 100 domains with $n_d = 10$ observations per domain in model-based simulation B

RB (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	2.63	2.50	3.47	3.41	4.25	4.54
MBOOST	0.74	0.75	1.54	1.65	4.66	4.74
MBOOST_T	1.03	1.06	1.76	1.88	4.88	5.10
MEGB	0.04	0.10	0.22	0.22	0.60	0.71
MERF	0.64	0.73	1.07	1.07	1.57	1.66
MERTBoosting	0.36	0.32	0.37	0.31	0.92	0.93
REEMBoosting	0.25	0.25	0.37	0.32	0.81	0.88
RF	0.77	0.83	1.79	1.95	5.30	5.53
GB	0.35	0.40	1.48	1.43	4.79	5.15

Table 13 RRMSE results (in %) for 100 domains with $n_d = 10$ observations per domain in model-based simulation B

RRMSE (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	7.71	7.76	13.11	13.16	17.74	18.22
MBOOST	7.28	7.33	13.45	13.46	22.74	23.25
MBOOST_T	7.01	7.07	13.63	13.61	23.60	24.08
MEGB	5.45	5.51	8.92	9.01	11.01	11.22
MERF	5.77	5.80	10.14	10.26	13.00	13.24
MERTBoosting	5.46	5.52	8.83	8.96	11.16	11.24
REEMBoosting	5.58	5.63	9.02	9.10	11.24	11.29
RF	5.90	5.96	13.87	13.79	25.13	25.70
GB	5.67	5.73	13.72	13.62	25.16	25.58

Table 14 RB results (in %) for 250 domains with $n_d = 10$ observations per domain in model-based simulation B

RB (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	2.05	2.02	3.09	3.12	4.01	4.05
MBOOST	0.51	0.56	1.93	1.94	5.09	5.05
MBOOST_T	0.60	0.60	1.88	1.93	5.28	5.17
MEGB	-0.20	-0.18	0.40	0.42	0.65	0.76
MERF	0.35	0.32	1.11	1.10	1.40	1.49
MERTBoosting	-0.16	-0.14	0.49	0.52	0.82	0.88
REEMBoosting	-0.11	-0.09	0.39	0.42	0.75	0.81
RF	0.39	0.43	2.01	2.15	5.74	5.71
GB	-0.04	-0.01	1.56	1.66	5.21	5.27

Table 15 RRMSE results (in %) for 250 domains with $n_d = 10$ observations per domain in model-based simulation B

RRMSE (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	7.19	7.23	12.59	12.65	17.56	17.68
MBOOST	6.92	6.90	13.88	13.92	25.16	25.45
MBOOST_T	6.77	6.74	13.98	14.01	25.53	25.82
MEGB	5.01	5.03	8.37	8.40	10.53	10.63
MERF	5.28	5.32	9.39	9.44	12.21	12.35
MERTBoosting	5.02	5.03	8.30	8.33	10.44	10.55
REEMBoosting	5.05	5.06	8.32	8.35	10.31	10.53
RF	5.70	5.68	13.62	13.71	25.96	26.09
GB	5.60	5.59	13.64	13.72	26.10	26.21

Table 16 RB results (in %) for 250 domains with $n_d = 30$ observations per domain in model-based simulation B

RB (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	1.95	1.96	2.36	2.46	2.27	2.39
MBOOST	0.51	0.47	1.88	1.93	5.28	5.40
MBOOST_T	0.48	0.44	1.89	1.95	5.31	5.45
MEGB	0.04	0.05	0.14	0.15	0.15	0.16
MERF	0.34	0.34	0.46	0.46	0.38	0.39
MERTBoosting	0.04	0.07	0.22	0.21	0.23	0.23
REEMBoosting	0.02	0.05	0.22	0.20	0.19	0.21
RF	0.55	0.53	1.99	1.99	5.28	5.45
GB	0.24	0.25	1.72	1.71	5.15	5.34

Table 17 RRMSE results (in %) for 250 domains with $n_d = 30$ observations per domain

RRMSE (%)	ICC = 0.05		ICC = 0.25		ICC = 0.5	
	Median	Mean	Median	Mean	Median	Mean
EBP	6.48	6.50	9.73	9.78	11.44	11.59
MBOOST	6.57	6.60	13.94	13.98	24.99	25.64
MBOOST_T	6.45	6.46	14.15	14.14	25.44	25.99
MEGB	3.80	3.79	4.87	4.89	5.66	5.65
MERF	4.14	4.15	5.61	5.61	6.34	6.32
MERTBoosting	3.96	3.96	5.21	5.22	5.97	5.97
REEMBoosting	3.95	3.98	5.20	5.21	5.92	5.94
RF	5.32	5.33	13.35	13.31	24.58	25.09
GB	5.51	5.53	13.99	13.93	25.78	26.32

Table 18 Mean and median bias for In-Sample-, and Out-Of-Sample- and all municipalities in total in Chihuahua for the EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting and the direct estimator HT in the design-based simulation study

Bias	Total		In-sample		Out-of-sample	
	Mean	Median	Mean	Median	Mean	Median
EBP-BC	153.616	101.327	−9.346	59.876	233.286	130.207
MEGB	74.782	122.483	23.317	50.743	99.942	134.758
MERF	183.743	222.718	126.816	164.079	211.573	247.918
MERTBoosting	99.522	135.044	42.136	71.437	127.578	163.634
REEMBoosting	−70.816	−31.560	−28.409	12.033	−91.548	−50.049
HT	−	−	3.677	−0.181	−	−

Fig. 10 Bias averaged over the Monte-Carlo simulation runs for the estimators EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting and the direct estimator HT

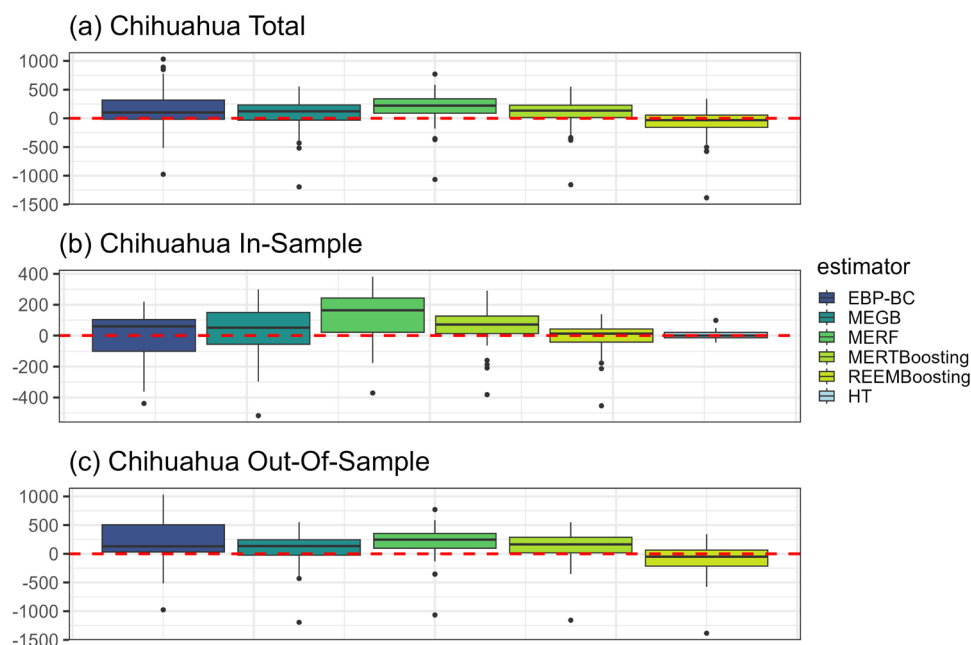


Fig. 11 RMSE averaged over the Monte-Carlo simulation runs for the estimators EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting and the direct estimator HT

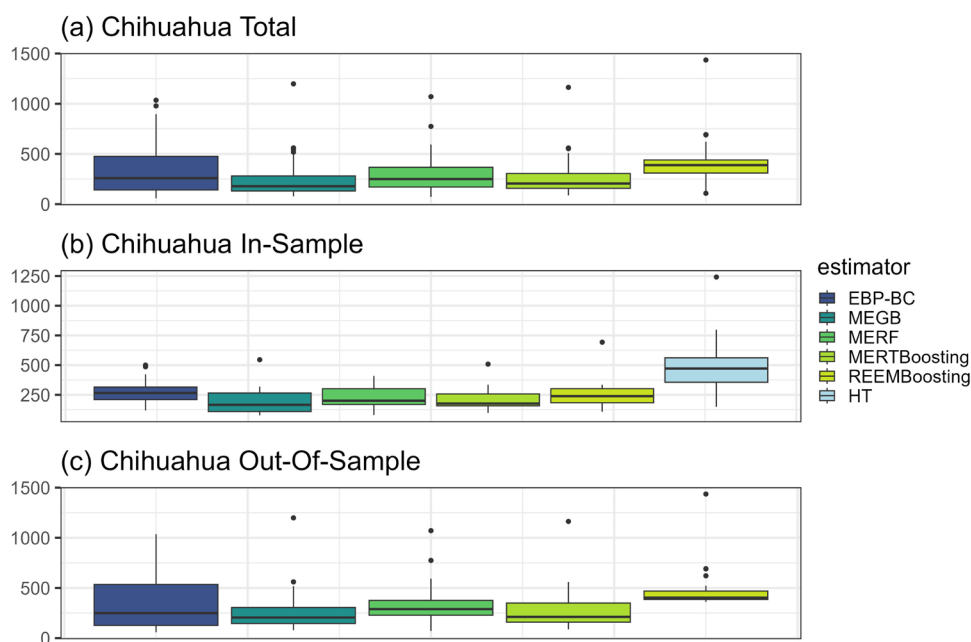


Table 19 Mean and median RMSE for the estimators EBP-BC, MEGB, MERF, MERTBoosting, REEMBoosting and the direct estimator HT and In-Sample-, and Out-Of-Sample- and all municipalities in total

RMSE	Total		In-sample		Out-of-sample	
	Mean	Median	Mean	Median	Mean	Median
EBP-BC	330.791	258.624	275.018	265.135	358.057	249.446
MEGB	236.396	177.989	197.791	165.089	255.269	205.577
MERF	288.485	249.212	232.813	200.395	315.703	288.919
MERTBoosting	250.198	205.001	206.130	176.091	271.743	211.556
REEMBoosting	390.033	388.111	251.381	238.327	457.818	402.534
HT	—	—	485.958	471.165	—	—

Table 20 Mean computation times (s) across all scenarios of model-based simulation B with varying ICC, number of domains (D) and observations per domain (n_d)

ICC	D	n_d	EBP	MBOOST	MBOOST_T	MEGB	MERF	MERT Boosting	REEM Boosting	RF	GB
0.05	100	10	2.93	8.10	22.31	12.92	10.48	57.03	241.17	1.49	0.56
0.05	100	30	3.14	14.71	31.60	55.35	47.52	43.25	163.64	8.66	0.93
0.05	250	10	5.77	12.31	26.32	39.53	38.36	61.94	600.98	6.03	0.75
0.05	250	30	6.46	20.16	39.52	165.76	130.58	70.69	400.13	38.71	1.66
0.25	100	10	3.04	8.82	22.49	14.26	14.76	62.21	272.30	1.57	0.58
0.25	100	30	2.73	12.58	26.07	51.01	42.16	31.33	100.85	8.04	0.78
0.25	250	10	5.60	12.36	25.89	39.18	38.61	47.25	469.79	6.07	0.72
0.25	250	30	6.29	22.15	41.77	163.34	153.02	56.50	407.49	44.02	1.59
0.50	100	10	2.80	8.62	22.22	12.46	12.72	60.87	239.88	1.51	0.58
0.50	100	30	3.17	15.81	32.44	53.62	50.14	36.86	131.48	8.71	0.96
0.50	250	10	5.58	13.11	26.18	37.21	38.28	41.70	347.91	6.08	0.71
0.50	250	30	8.53	24.89	46.73	184.36	168.44	54.27	403.66	46.36	1.59

Two qualifications should be added. First, in the design-based study REEMBoosting did not fail to complete, but its results were unstable across replications, so that a configuration we would regard as robust for this application could not be identified; its results are therefore reported in this Appendix rather than in Sect. 5.2. Second, in the comparison with GMEXGBoost reported above, 30 of 500 replications did not complete in the Complex-Normal scenario; as discussed there, we were not able to establish whether this originates in the published method or in our adaptation of it to a continuous response.

Beyond completion rates, we examined the internal stability of the EM-type procedure of MEGB directly. Using an illustrative dataset, we compared the resulting area-level estimates when the procedure is initialized from three very different starting values for the random effects: zero, a large constant of 50,000, and a data-driven warm start based on BHF residuals. The resulting estimates were essentially identical across initializations, with a correlation of approximately 1 between the zero start and the large-constant start, and of approximately 0.999 between the zero start and the BHF warm start. This indicates that the procedure reaches the same solution irrespective of its starting point rather than being sensitive to initialization.

Computation times for model-based simulation B

For comparability, we report runtimes only for model-based simulation B (no bootstrap-based uncertainty estimation for MEGB). The computation times were reported across $M = 100$ Monte-Carlo replications in Table 20.

Acknowledgements The authors gratefully acknowledge CONEVAL for providing access to the data used in the design-based simulation study. The authors also thank the Editor-in-Chief, the Associate Editor,

and two anonymous referees for their valuable comments and suggestions, which significantly improved the manuscript.

Author Contributions Paul Messer: Conceptualization, Methodology, Formal Analysis, Software, Writing-Original Draft, Writing-Review & Editing Timo Schmid: Conceptualization, Methodology, Formal Analysis, Writing-Review & Editing, Supervision

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability All scripts required to reproduce the model-based simulation results (Section 4), including the data-generating mechanisms and the MEGB implementation, are available in the following public GitHub repository: <https://github.com/pellmess/MEGB-Simulations>. The real-world microdata used in the design-based study (in Section 5) is protected by confidentiality agreements with the data provider and cannot be shared publicly. Access is restricted and requires formal authorization from the provider.

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

- Asadi, F., Homayounfar, R., Mehrali, Y., Masci, C., Zayeri, F.: Innovative statistical method for longitudinal and hierarchical data modeling: the gmexgboost method. *BMC Med. Res. Methodol.* **26**(24), (2026). <https://doi.org/10.1186/s12874-025-02751-7>

- Athey, S., Tibshirani, J., Wager, S.: Generalized random forests. *Ann. Stat.* **47**(2), 1148–1178 (2019). <https://doi.org/10.1214/18-AOS1709>
- Battese, G., Harter, R., Fuller, W.: An error-components model for prediction of county crop areas using survey and satellite data. *J. Am. Stat. Assoc.* **83**(401), 28–36 (1988). <https://doi.org/10.1080/01621459.1988.10478561>
- Breiman, L.: Prediction games and arcing algorithms. *Neural Comput.* **11**(7), 1493–1517 (1999). <https://doi.org/10.1162/089976699300016106>
- Carlstein, E., Do, K.A., Hall, P., Hesterberg, T., Künsch, H.R.: Matched-block bootstrap for dependent data. *Bernoulli* **4**(3), 305–328 (1998)
- Chambers, R., Chandra, H.: A random effect block bootstrap for clustered data. *J. Comput. Graph. Stat.* **22**(2), 452–470 (2013). <https://doi.org/10.1080/10618600.2012.681216>
- Chambers, R., Chandra, H., Salvati, N., Tzavidis, N.: Outlier robust small area estimation. *J. Royal Stat. Soc. Series B (Stat. Methodol.)* **76**(1), 47–69 (2013). <https://doi.org/10.1111/rssb.12019>
- Chambers, R., Tzavidis, N.: M-quantile models for small area estimation. *Biometrika* **93**(2), 255–268 (2006). <https://doi.org/10.1093/biomet/93.2.255>
- Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. Association for Computing Machinery, New York (2016)
- Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K.: xgboost: extreme Gradient Boosting. R package version 1.7.7.1 (2023)
- Chipman, H.A., George, E.I., McCulloch, R.E.: Bart: Bayesian additive regression trees. *The Annals of Appl. Stat.* **4**(1), 266–298 (2010). <https://doi.org/10.1214/09-AOAS285>
- Cho, S.J., Mueller, S.: Mixed-effects XGBoost with group-aware permutation importance and cross-validation for multilevel cross-classified continuous outcomes. *Psychometrika* (2026). <https://doi.org/10.1017/psy.2026.10108>
- Corral, P., Himelein, K., McGee, K., Molina, I.: A map of the poor or a poor map? *Mathematics* **9**(21), 2780 (2021). <https://doi.org/10.3390/math9212780>
- Corral, P., Molina, I., Cojocar, A., Segovia, S.: Guidelines to Small Area Estimation for Poverty Mapping. The World Bank, Washington DC (2022)
- Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc.: Ser. B (Methodol.)* **39**(1), 1–22 (1977). <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>
- Diallo, M.S., Rao, J.: Small area estimation of complex parameters under unit-level models with skew-normal errors. *Scand. J. Stat.* **45**(4), 1092–1116 (2018). <https://doi.org/10.1111/sjos.12336>
- Dorogush, A.V., Ershov, V., Gulin, A.: Catboost: gradient boosting with categorical features support. (2018)
- Fahrmeir, L., Kneib, T., Lang, S., Marx, B.: *Regression: Models, Methods and Applications*. Springer, Berlin (2013)
- Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997). <https://doi.org/10.1006/jcss.1997.1504>
- Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**(5), 1189–1232 (2001). <https://doi.org/10.1214/aos/1013203451>
- Frink, N., Schmid, T., Würz, N.: Small area estimation with generalized random forests: estimating poverty rates in Mexico. *J. Royal Stat. Soc. Series A: Stat. Soc.* (2026) <https://doi.org/10.1093/jrsssa/qnag092>
- Graf, M., Marín, J.M., Molina, I.: A generalized mixed model for skewed distributions applied to small area estimation. *TEST* **28**(2), 565–597 (2019). <https://doi.org/10.1007/s11749-018-0594-2>
- Griesbach, C., Säfken, B., Waldmann, E.: Gradient boosting for linear mixed models. *Int. J. Biostat.* **17**(2), 317–329 (2021). <https://doi.org/10.1515/ijb-2020-0136>
- Hajjem, A., Bellavance, F., Larocque, D.: Mixed effects regression trees for clustered data. *Stat. Probab. Letters* **81**(4), 451–459 (2011). <https://doi.org/10.1016/j.spl.2010.12.003>
- Hajjem, A., Bellavance, F., Larocque, D.: Mixed-effects random forest for clustered data. *J. Stat. Comput. Simul.* **84**(6), 1313–1328 (2014). <https://doi.org/10.1080/00949655.2012.741599>
- Hajjem, A., Larocque, D., Bellavance, F.: Generalized mixed effects regression trees. *Stat. Prob. Letters* **100**(126), 114–118 (2017). <https://doi.org/10.1016/j.spl.2017.02.033>
- Hastie, T., Tibshirani, R., Friedman, J.H.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York (2009)
- Horvitz, D.G., Thompson, D.J.: A generalization of sampling without replacement from a finite universe. *J. Am. Stat. Assoc.* **47**(260), 663–685 (1952). <https://doi.org/10.1080/01621459.1952.10483446>
- Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., Hofner, B.: Model-based boosting 2.0. *J. Mach. Learn. Res.* **11**(71), 2109–2113 (2010)
- Huber, P.J.: Robust estimation of a location parameter. *Ann. Math. Stat.* **35**(1), 73–101 (1964)
- Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.* **110**(3), 457–506 (2021). <https://doi.org/10.1007/s10994-021-05946-3>
- Jiang, J., Lahiri, P.: Mixed model prediction and small area estimation. *TEST Official J. Span. Soc. Stat. Oper. Res.* **15**(1), 1–96 (2006). <https://doi.org/10.1007/BF02595419>
- Jiang, J., Rao, J.S.: Robust small area estimation: an overview. *Annual Rev. Stat. Appl.* **7**(1), 337–360 (2020). <https://doi.org/10.1146/annurev-statistics-031219-041212>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: a highly efficient gradient boosting decision tree. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Volume 30 of NIPS’17, Red Hook, pp. 3149–3157. Curran Associates Inc (2017)
- Krennmair, P., Frink, N.: SAEforest: mixed effect random forests for small area estimation. R package version 1.0.1, (2025) <https://github.com/krennpa/SAEforest>
- Krennmair, P., Schmid, T.: Flexible domain prediction using mixed effects random forests. *J. Royal Stat. Soc.: Ser. C (Appl. Stat.)* **71**(5), 1865–1894 (2022). <https://doi.org/10.1111/rssc.12600>
- Krennmair, P., Würz, N., Schmid, T.: Analysing opportunity cost of care work using mixed effects random forests under aggregated auxiliary data. *J. Royal Stat. Soc. Series C (Appl. Stat.)* **75**(1), 1–20 (2026). <https://doi.org/10.1093/jrsssc/qlaf031>
- Kreutzmann, A.K., Pannier, S., Rojas-Perilla, N., Schmid, T., Tempel, M., Tzavidis, N.: The R package emdi for estimating and mapping regionally disaggregated indicators. *J. Stat. Softw.* **91**(7), 1–33 (2019). <https://doi.org/10.18637/jss.v091.i07>
- Kriegler, B., Berk, R.: Small area estimation of the homeless in los angeles: an application of cost-sensitive stochastic gradient boosting. *The Annals of Appl. Stat.* **4**(3), 1234–1255 (2010). <https://doi.org/10.1214/10-AOAS328>
- Liaw, A., Wiener, M.: Classification and regression by randomforest. *R News* **2**(3), 18–22 (2002)
- Merfeld, J.D., Newhouse, D.: Improving estimates of mean welfare and uncertainty in developing countries. *World Bank Policy Research Working Paper* (10348) (2023) <https://doi.org/10.1596/1813-9450-10348>
- Molina, I., Martín, N.: Empirical best prediction under a nested error model with log transformation. *Ann. Stat.* **46**(5), 1961–1993 (2018). <https://doi.org/10.1214/17-AOS1608>

- Molina, I., Rao, J.N.: Small area estimation of poverty indicators. *Can. J. Stat.* **38**(3), 369–385 (2010). <https://doi.org/10.1002/cjs.10051>
- Morales, D., Esteban, M.D., Pérez, A., Hobza, T.: A course on small area estimation and mixed models: methods, theory and applications in R. In: *Statistics for Social and Behavioral Sciences*. Springer, Cham (2021)
- Olaniran, O.R., Olaniran, S.F., Allohobi, J., Alharbi, A.A., Alharbi, N.M.: Mixed effect gradient boosting for high-dimensional longitudinal data. *Sci. Rep.* **15**(1), 30927 (2025). <https://doi.org/10.1038/s41598-025-16526-z>
- Opsomer, J.D., Claeskens, G., Ranalli, M.G., Kauermann, G., Breidt, F.J.: Non-parametric small area estimation using penalized spline regression. *J. Royal Stat. Soc. Series B (Stat. Methodol.)* **70**(1), 265–286 (2008). <https://doi.org/10.1111/j.1467-9868.2007.00635.x>
- Rao, J.N., Molina, I.: *Small Area Estimation*. John Wiley & Sons, Hoboken (2015)
- Rojas-Perilla, N., Pannier, S., Schmid, T., Tzavidis, N.: Data-driven transformations in small area estimation. *J. Royal Stat. Soc. A. Stat. Soc.* **183**(1), 121–148 (2020). <https://doi.org/10.1111/rssa.12488>
- Salditt, M., Humberg, S., Nestler, S.: Gradient tree boosting for hierarchical data. *Multivar. Behav. Res.* **58**(5), 911–937 (2023). <https://doi.org/10.1080/00273171.2022.2146638>
- Schapire, R.E., Freund, Y.: Boosting: foundations and algorithms. *Kybernetes* **42**(1), 164–166 (2013). <https://doi.org/10.1108/03684921311295547>
- Sela, R.J., Simonoff, J.S.: Re-em trees: a data mining approach for longitudinal and clustered data. *Mach. Learn.* **86**(2), 169–207 (2012). <https://doi.org/10.1007/s10994-011-5258-3>
- Sugasawa, S., Kubokawa, T.: Adaptively transformed mixed-model prediction of general finite-population parameters. *Scand. J. Stat.* **46**(4), 1025–1046 (2019). <https://doi.org/10.1111/sjos.12380>
- Tan, Y.V., Flannagan, C.A., Elliott, M.R.: “Robust-squared” imputation models using bart. *J. Survey Stat. Methodol.* **7**(4), 465–497 (2019). <https://doi.org/10.1093/jssam/smz002>
- Tzavidis, N., Zhang, L.C., Luna, A., Schmid, T., Rojas-Perilla, N.: From start to finish: a framework for the production of small area official statistics. *J. Royal Stat. Soc. A. Stat. Soc.* **181**(4), 927–979 (2018). <https://doi.org/10.1111/rssa.12364>
- Ugarte, M., Goicoa, T., Militino, A., Durbán, M.: Spline smoothing in small area trend estimation and forecasting. *Comput. Stat. Data Anal.* **53**(10), 3616–3629 (2009). <https://doi.org/10.1016/j.csda.2009.02.027>
- Viljanen, M., Meijerink, L., Zwakhals, L., Van de Kasstele, J.: A machine learning approach to small area estimation: predicting the health, housing and well-being of the population of netherlands. *Int. J. Health Geogr.* **21**(1), 4 (2022). <https://doi.org/10.1186/s12942-022-00304-5>
- Würz, N.: saeTrafo: transformations for unit-level small area models. R package version 1.0.0 (2022)
- Yang, S.I., Brandeis, T.J., Helmer, E.H., Marcano-Vega, H.: Predicting species-specific diameter growth rate for Caribbean trees using mixed-effects extreme gradient boosting. *For. Ecol. Manage.* **580**(1), 122520 (2025). <https://doi.org/10.1016/j.foreco.2025.122520>
- Yoon, J.: Forecasting of real gdp growth using machine learning models: gradient boosting and random forest approach. *Comput. Econ.* **57**(1), 247–265 (2021). <https://doi.org/10.1007/s10614-020-10054-w>
- Zhang, Z., Zhang, T., Li, J.: 8. Unbiased gradient boosting decision tree with unbiased feature importance. In: Elkind, E. (ed.) *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4629–4637. International Joint Conferences on Artificial Intelligence Organization. (2023)

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