ERASMUS UNIVERSITY ROTTERDAM

ERASMUS SCHOOL OF ECONOMICS

# Simulating Haemoglobin Concentrations Using Black-box Machine Learning as a Step Towards Personalised Colorectal Cancer Screening[1]

*Author*

Yoëlle Kilsdonk (513530)

*Supervisors*

E. P. O'Neill (Erasmus University Rotterdam)

dr. I. Lansdorp-Vogelaar (Erasmus Medical Centre)

R. van den Puttelaar (Erasmus Medical Centre)

D. van den Berg (Erasmus Medical Centre)

*Second assessor*

dr. O. Vicil (Erasmus University Rotterdam)

January 21, 2023

[1]The views stated in this thesis are those of the author and not necessarily those of the supervisors, second assessor, Erasmus School of Economics, Erasmus University Rotterdam or Erasmus Medical Centre.

**Abstract**

This research aims to identify accurate predictive models for simulating haemoglobin concentrations in the MISCAN-Colon (MIcrosimulation SCreening ANalysis) model developed by the Erasmus Medical Centre, to eventually facilitate the evaluation of personalised screening strategies.

We evaluate five models black-box machine learning models. Specifically, we compare the predictive performance between the fixed-effects and mixed-effects versions of artificial neural network and eXtreme Gradient Boosting (XGBoost) models, using the framework by Hajjem et al. (2014) and we investigate whether the performance of the fixed-effects XGBoost model increases using the Tweedie loss function, based on the zero-inflation in the haemoglobin concentrations. The proposed mixed-effects models in this paper are novel, which adds to the contribution of our research.

We identify the mixed-effects XGBoost (MeXGBoost) model and the XGBoost model with Tweedie loss (TweedieXGBoost) as the best performing methods applied to the Dutch screening data provided by the Erasmus Medical Centre. These models correctly predict 81% of observations within an interval of $\pm 3\mu$g/g of the true haemoglobin concentration, if this concentration lies below 47 $\mu$g/g. Moreover, if we cast the continuous predictions to binary FIT results, we find that both models attain a true positive rate of approximately 99% based on $\pm 1.8$ million test samples, and a true negative rate of 59% for TweedieXGBoost and 67% for MeXGBoost based on $\pm 70$ thousand test samples. However, since none of our models perform particularly well in terms of exact predictions for higher haemoglobin concentrations, we do not recommend implementing any of our current model models in the MISCAN-Colon model. Instead, we suggest alternative methods for further research based on our findings.

***Keywords***— Mixed-effects machine learning, Tweedie loss, haemoglobin concentration, colorectal cancer, MISCAN-Colon

# Table of contents

# List of Figures



# List of Tables

# 1 Introduction

Colorectal cancer (CRC) is one of the leading causes of cancer-related deaths in Western countries (Loeve et al., 1999; Sung et al., 2021; Torre et al., 2015), and it is expected that the absolute number of cases will increase as a result of aging and growth of populations. Currently, the Dutch Rijksinstituut voor Volkgezondheid en Milieu estimates that one in twenty people will develop CRC in the Netherlands. Fortunately, considerable research finds that CRC, or early stages thereof, can be detected and treated through population screening, which in turn could prevent a large proportion of CRC (death) cases. The question then remains, which screening policies are most optimal?

Clinical trials often only last a couple of years, while policy makers are most interested in the (cost-)effectiveness of screening strategies over a lifetime. For example, to research whether CRC mortality can be reduced through changes in a current policy, one would have to follow individuals throughout their whole lives. Additionally, in order to compare amongst screening policies, one would have to simultaneously implement and evaluate multiple policies using a real-life population. Since both of these scenarios are infeasible in practice, the Erasmus Medical Center (EMC) developed the MISCAN-Colon (MIcrosimulation SCreening ANalysis) model – a microsimulation model for the evaluation of CRC screening.

The implementation of the MISCAN-Colon model at EMC follows the guidelines of the current screening procedure in the Netherlands. That is, each individual between 55 and 75 years of age receive a faecal immunochemical test (FIT) once every two years, which can be performed at home on voluntary basis. This stool-based test measures the level of haemoglobin (blood) present in a patient's faeces, where higher levels of blood may be related to polyps and CRC. The MISCAN-Colon model uses the sensitivity (true positive rate) and specificity (true negative rate) of these FIT results to simulate a positive or negative FIT result for simulated individuals.

Recently, however, the Public Health department of EMC explored an extension of MISCAN-Colon to evaluate the benefits of personalised screening strategies, where, instead of simulating FIT results, the model would simulate the haemoglobin concentrations in a patient's stool using a linear mixed model algorithm (van Duuren et al., 2022). In this thesis, we use black-box machine learning methods instead to predict such haemoglobin concentrations for a real-life data set.

The data for this research is provided by the EMC from the national CRC screening program from 2014-2020, originally collected by the Dutch National Institute for Public Health and the Environment. For each of the 3.2 million individuals in the data set, a maximum of four screening rounds are available. We only include individuals who participate in two or more *consecutive* rounds, and those who participate in one round in total.

Most machine learning methods rely on the assumption of iid observations – either for inference or accuracy of estimation – which may be inappropriate in longitudinal healthcare data due to correlations within individuals.[2] To overcome this issue, Hajjem et al. (2014) propose an approach which incorporates random-effects in machine learning algorithms for efficient analysis of longitudinal data. Based on this

[2]In this case, patients with negative FITs participate in multiple rounds, which allows for such correlation.

approach, van den Berg (2021) finds that mixed-effect machine learning models significantly outperform her benchmark mixed-effect zero-inflated negative binomial model in simulating haemoglobin concentrations for MISCAN-Colon. It is unclear, however, whether the increase in predictive accuracy of these mixed-effects machine learning models is specifically due to the inclusion of random-effects, or due to the use of machine learning methods in general. Therefore, our research investigates the contribution of the inclusion of random-effects to the predictive performance of black-box machine learning methods. We implement artificial neural network (ANN) and eXtreme Gradient Boosting (XGBoost) models, both with and without mixed-effects, using the framework proposed by Hajjem et al. (2014). We also employ a third XGBoost model using Tweedie loss as objective function to more explicitly account for the zero-inflation of the data. This leads to the following research questions:

RQ1a Does the introduction of random-effects in machine learning models lead to better performance, i.e., do mixed-effects machine learning models outperform 'regular' machine learning models?

RQ1b Which model is best suited for predicting the haemoglobin concentration based on the data set provided by EMC?

The main contributions of our research are novel adaptations to the mixed-effects framework by Hajjem et al. (2011) using ANN and XGBoost models. We find that the mixed-effects XGBoost model, and the XGBoost model with Tweedie loss perform best in predicting haemoglobin concentrations over all. These models correctly predict 81% of observations within an interval of $\pm 3\mu$g/g of the true haemoglobin concentration for individuals with negative FIT results. Additionally, if we cast the continuous predictions to binary FIT results, both models attain a true positive rate of approximately 99%, and a true negative rate of 59% for TweedieXGBoost and 67% for MeXGBoost. However, none of our models perform particularly well in terms of exact predictions for higher haemoglobin concentrations.

Furthermore, we provide a review and comparison on the performance, mathematical properties, and model compatibility for six (of the most) influential papers in the field of machine learning in longitudinal data: Hajjem et al. (2011, 2014, 2017); Sela and Simonoff (2011), and Ngufor et al. (2019). To our knowledge, we are among the firsts to provide a (literary) review such as ours of these five models.

The remainder of this research is structured as follows. We provide background information on colorectal cancer and MISCAN-colon in Section 2, along with an overview of the applications of ANN and XGBoost models in healthcare literature. In Section 3 we describe the data and the data imputation method. We present our methodology in Section 4, followed by our results and a discussion of these results in Sections 5 and 6, respectively. Lastly, we present our conclusion in Section 7

# 2 Literature

## 2.1 Colorectal cancer

Colorectal cancer (CRC) is the development of cancer in the colon or rectum. Colorectal adenocarcinomas are the most common form of CRC, making up over 95% of all CRCs (Thrumurthy et al., 2016). Risk

factors for CRC include, but are not limited to, age, gender, genetics, environment, diet, physical activity, and smoking (Botteri et al., 2008; Thanikachalam and Khan, 2019).

Research indicates that 90-95% of CRCs develop from benign adenomas (i.e., a noncancerous tumor) (Bronner and Haggitt, 1993; Morson, 1974). Additionally, Strum (2016) finds that 20% to 53% of the U.S. population older than 50 years develop such adenomas of the colon, although only a small percentage of these adenomas become cancerous. Therefore, we distinguish between progressive and non-progressive adenomas, where (non-)progressive adenomas do (not) develop into CRC. Figure 1 illustrates the progression of CRC in five stages. In stage 0, the adenoma has not grown beyond the mucosa (i.e., the inner lining) of the colon or rectum. Stage I is when the cancerous adenocarcinoma has grown beyond the mucosa without spreading to the lymphatic system or distant organs. In stage II the adenocarcinoma has invaded the colonic or rectal wall, with possible infection of nearby organs. Finally, in stages III and IV, the metastatic adenocarcinoma has spread to lymph nodes and distant organs, respectively.

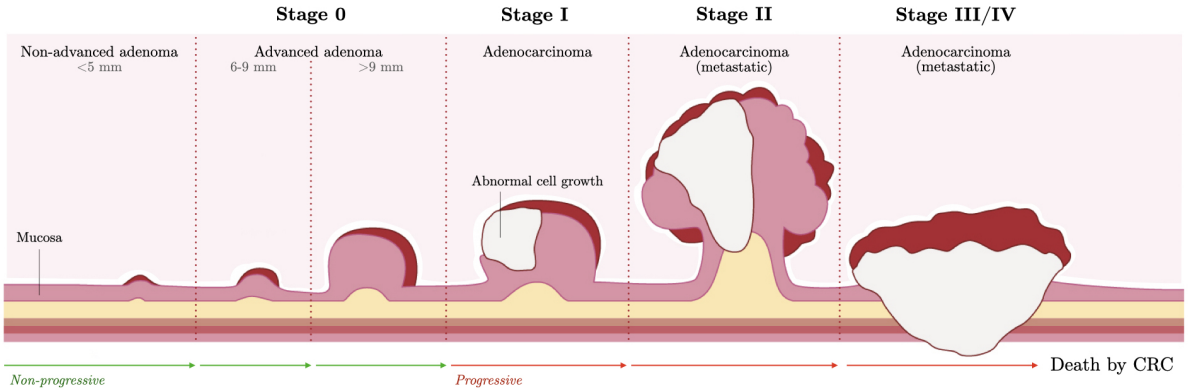



Figure 1: Progression of colorectal cancer in stages (image edited from Guts UK Charity).

### 2.1.1 Screening

CRC is one of the most commonly diagnosed and most deadly cancers worldwide (Torre et al., 2015; Sung et al., 2021). According to the Dutch Rijksinstituut voor Volkgezondheid en Milieu, five percent of the population will develop CRC in the Netherlands. Nearly nine in ten cases of that 5% occur in people over the age of 55. In fact, the worldwide burden of CRC is expected to further increase due to, *inter alia*, the rapid growth and aging of the population (Jiang et al., 2022; Winawer, 2007), which is a testament to the importance of optimising screening procedures.

The effect of screening is twofold. First, considering that the vast majority of CRCs develop from benign adenomas, early detection and removal of adenomas might aid in prevention of CRC (Loeve et al., 1999). Second, early detection of both asymptomatic and symptomatic adenocarcinomas may result in an improvement in prognosis. Both of these findings are supported by, amongst others, Jiang et al. (2022); Levin et al. (2008); Toribara and Sleisenger (1995), and Whitlock et al. (2012).

Established screening tests can be subdivided into two categories: stool-based tests and visual exams. The guaiac-based fecal occult blood test (gFOBT) and fecal immunochemical test (FIT), for example, belong to the first category, in which the stool is tested for the presence of blood. If these tests report

a high haemoglobin concentration, this could be an indication for the presence of CRC.[3] The two most common visual exams are (flexible) sigmoidoscopy, and colonoscopy, which investigate the structure of the colon and rectum for abnormal tissue. According to the review by Ding et al. (2022), colonoscopies are most effective in reducing CRC-related deaths, at an approximate 68% decrease[4] (Brenner et al., 2014). As for the stool-based tests, the biennial FIT-based screening reduces CRC-related deaths by 22% on average[5], which is approximately 7% more effective than the gFOBT test (based on a 10-year follow-up period) (Hewitson et al., 2008; Zorzi et al., 2015). The FIT also has a better performance in participation and positivity rate compared to gFOBT in CRC screening programs, while reporting fewer false negatives (Mousavinezhad et al., 2016). Moreover, the FIT is relatively close in effectiveness compared to flexible sigmoidoscopies while being considerably less invasive, with reported mortality reduction of approximately 28%[6] compared to no screening (Holme et al., 2013). If screening with a stool-based test leads to abnormal test results, defined as haemoglobin concentrations above a predetermined threshold, the general advice is to proceed with a follow-up colonoscopy (Ding et al., 2022).

Since January of 2014, each individual in the Netherlands between the ages of 55-75 receives a biennial invitation to participate in the population screening for CRC.[7] Each individual who chooses to participate then collects a stool sample using the FIT, which is sent back to the laboratory. If the haemoglobin concentration in this stool sample exceeds the predetermined threshold of 47 micrograms haemoglobin per gram of faeces, an automatic referral for a colonoscopy and treatment is sent to the participant. If any abnormalities are present during the colonoscopy, small amounts of tissue can be removed for analysis (i.e., a biopsy) and abnormal growths or adenomas can be identified and removed. Figure 2 shows that, according to the Integraal Kankercentrum Nederland, patients diagnosed with CRC through the Dutch population screening have a more favorable stage distribution than patients without screening. Additionally, patients who are diagnosed through population screening are less likely to receive invasive treatments.

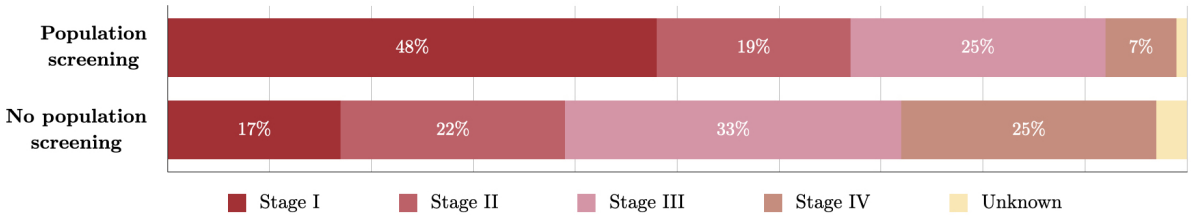



Figure 2: Distribution of diagnosed cancers in patients with, and without screening (source: Integraal Kankercentrum Nederland).

Unfortunately, however, screening policies are no silver bullet in healthcare. Screening can be costly,

[3]Intestinal abnormalities, which may progress to cancer over time, bleed more than normal tissue. Thus, if a patients blood contains high haemoglobin concentrations, this might be an indication for (early stages of) CRC.

[4]This pooled estimate is based on a meta-analysis of observational studies by Kahi et al. (2009); Manser et al. (2012), and Nishihara et al. (2013), with respective follow-up periods of 18, 6, and 24 years.

[5]Based on nine articles describing four randomised controlled trials with follow-up periods ranging between 8 to 18 years.

[6]This percentage reduction is based on the meta-analysis by Holme et al. (2013), where follow-up periods per study range from 6 to 19.5 years.

[7]For more information see: https://www.rivm.nl/darmkanker.

invasive, and it could potentially lead to, i.a., overdiagnosis and false positives. Overdiagnosis is associated with long-term psychosocial harm, lower quality of life, and unwanted or unnecessary usage of treatment, healthcare facilities, and (follow-up) tests – the latter of which could be particularly harmful as each treatment comes with its specific risk (Barton et al., 2001; Brodersen and Siersma, 2013; Jenniskens et al., 2017; van der Steeg et al., 2011).[8] Welch and Black (2010) provide a summary of current evidence that early detection of in breast, lung, and prostate cancer leads to overdiagnosis – defined as the diagnosis of a medical condition or disease that would not cause symptoms or death during a patient's lifetime. In contrast, Brasso et al. (2010) and Wardle et al. (2003) find no adverse psychological effects due to colorectal cancer screening using the FIT, although they do not specifically investigate the effects of overdiagnosis.

Given the previously stated disadvantages to screening, it is clear that policy makers must continually evaluate the trade-off between harms and benefits to identify the most efficient screening policies. A large body of literature indicates that personalised screening may aid in achieving such optimised policies for colorectal, prostate, and breast cancer (Frampton et al., 2016; Pashayan et al., 2011; Schröder et al., 2009). Moreover, Grobbee et al. (2017) suggest that FIT-based programs can be improved upon by using a screening policy with person-specific intervals and thresholds depending on previous haemoglobin concentrations in an individual's stool. Particularly, haemoglobin concentrations just below the threshold are associated with higher risks of advanced neoplasia[9] – a finding which can be exploited in personalised screening.

However, personalised screening necessitates policymakers and health care providers to make decisions on, i.a., what tools to use to identify risk levels and at which risk levels screening or prevention programs are warranted, which results in limitless possibilities of feasible personalised screening policies. Recently, van Duuren et al. (2022) adapted Habbema et al.'s (1985) microsimulation model for colorectal cancer – referred to as MISCAN-Colon (MIcrosimulation SCreening ANalysis Colorectal Cancer) – to simulate haemoglobin concentrations in a person's stool, instead of simulating positive or negative FIT results.

## 2.2 MISCAN-Colon

MISCAN-Colon allows for the evaluation of different screening policies by comparing their costs and effectiveness, as well as assessing the risk of false positives and overdiagnosis on a simulated population *before* real-life enforcement (Loeve et al., 1999). The model simulates individual life histories in which several colorectal lesions can emerge, and produces incidence and mortality rates in the simulated population using information on the epidemiology and natural history of the disease as input combined with screening- and demographic characteristics. By comparing the simulated life histories with screening to life histories without screening, MISCAN-Colon can be used to evaluate the costs and benefits of a specific screening strategy.

The MISCAN-Colon model can be decomposed into three parts: demography, natural history and

[8] For an assessment of operative risk in CRC surgery, we refer to Fazio et al. (2004) and Hanley (2005).

[9] Adopting the definition of Regula et al. (2006), advanced neoplasia is a cancer or adenoma of at least 10 mm in diameter, has high-grade dysplasia, or has villous or tubulovillous histologic characteristics, or any combination thereof.

screening. Figure 3 shows an exemplified version of these three parts. The upper line, referred to as the demography part, represents the life of a simulated individual without cancer who dies at 87 years old of other causes than CRC. The MISCAN-Colon model then adds a natural history of the disease to the demography part by simulating this individual with cancer. In this scenario, the individual dies at 72 as a result of CRC. The bottom line represents this same individual's life when screening is overlayed, with 15 gained life years as a result.

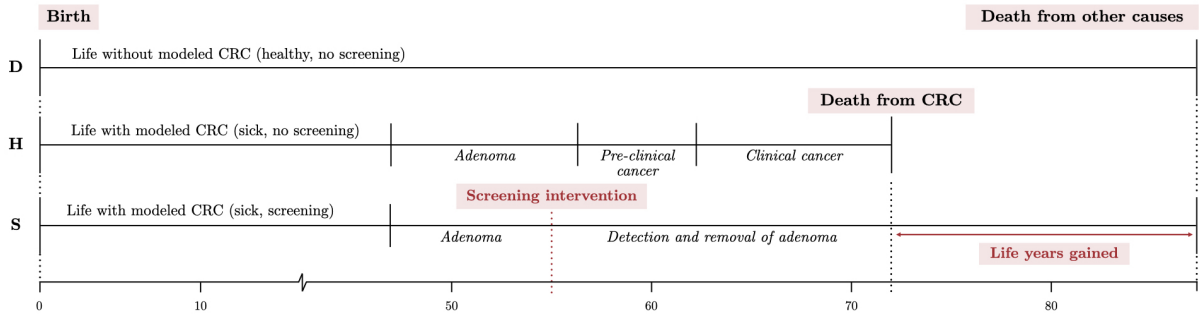



Figure 3: Simulations from the MISCAN-Colon model, where the upper bar shows the demography part (D), the middle bar adds the natural history of CRC (H) to D, and the lower bar adds both screening (S) and H to D.

We make three remarks on Figure 3. First, the chances of surviving a lesion after diagnosis depends on the stage of the cancer (and other risk factors). Thus, screening does not ensure that an individual survives CRC. The possible prognoses after a positive test result for CRC screening are: delay in moment of death, no change in moment of death, or premature death by complications of treatment. Second, the figure only shows an example of an individual with one lesion for simplicity, but the MISCAN-Colon model also allows for the modelling of zero or multiple lesions. New lesions that appear after clinical diagnosis of CRC are accounted for in the simulated survival. Third, Figure 3 only shows lethal progressive adenomas, but it is also possible that an individual develops non-lethal adenomas.

## 2.3 Machine learning methods for longitudinal health data

This paper using black-box machine learning methods on zero-inflated and longitudinal healthcare data, with repeated measurements per individual over two-year intervals of time (see Section 3). This data set with repeated measurements contains observations that are (likely to be) correlated across time for each individual, which in turn might violate the assumption that the data are independently identically distributed (iid). In turn, violation of this assumption might lead to misleading inference (Ngufor et al., 2019) and/or decreased performance (Sela and Simonoff, 2011; Hajjem et al., 2017), the latter of which is particularly relevant to our research.

One possible solution to this problem could be to employ 'regular' machine learning models while explicitly modeling the intracluster correlation through inclusion of time-specific variables (e.g., how often a patient has participated including the current round, previous haemoglobin concentrations, and maximum haemoglobin concentration over previous rounds). However, the nature of this data suggests that better estimation may be possible if the information of the repeated measurements would be included

at the level of the algorithm itself, e.g., through mixed-effects machine learning methods.

This research employs artificial neural network (ANN) and eXtreme Gradient Boosting (XGBoost) models in a mixed-effects machine learning framework. In the remainder of this section, we provide a literature overview of ANN and XGBoost models in longitudinal health data to assess how researchers currently account for possible intra-individual correlations. We also motivate our chosen mixed-effects framework.

### 2.3.1 Artificial neural networks

The trajectory of cancer is clearly nonlinear, highly variable, and dependent on a large variety of factors, most of which are not understood to this day. The flexibility of ANNs can be used to effectively address (part of) these problems.

However, ANNs make the implicit assumption of iid data, which is often violated in longitudinal data. Although certain ANNs have been successfully adjusted to account for temporal trends (e.g., Choi et al.'s (2016) recurrent neural networks), longitudinal data often also contain unequal time intervals between measurements, and an unequal number of observations per individual. To account for these specific data characteristics, Xiong et al. (2019) propose a new type of ANN, called the mixed effects neural network model, which adapts mixed effects within a deep neural network architecture for gaze estimation based on eye images. This model is person-specific, and uses few calibration samples to eliminate the person-specific bias in longitudinal data. In the field of Alzheimers, Tandon et al. (2006) introduce another mixed effects neural network to accurately model the nonlinear course of the disease. Their model generalizes a linear mixed effects model by incorporating a general nonlinear function of the input variables. This model is shown to be much more accurate and effective compared to standard ANNs and linear mixed effects models. Lastly, Mandel et al. (2021) propose a generalised neural network mixed effects model – also structured as a generalised linear mixed model (GLMM) – where the linear fixed effect is replaced by a feed-forward ANN and a random effect component is added to predict depression and anxiety levels of schizophrenic patients using longitudinal data.

Another important property of ANN models with respect to our application, is their suitability for prediction of non-negative variables (Haghani et al., 2017; Sakthivel and Rajitha, 2017). Moreover, Haghani et al. (2017) show that ANNs outperform (zero-inflated) Poisson regression and (zero-inflated) negative binomial regression in their research of predicting the number of return to blood donations using zero-inflated data. Thus, the results of this study indicate that ANNs also adapt well to zero-inflation of the dependent variable.

### 2.3.2 XGBoost

**Tree-based algorithms** The first to extend regression trees to longitudinal data was Segal (1992), who based his methodology on modifying the split function to accommodate repeated measures over time. This method, however, cannot handle time-varying covariates, and the resulting trees cannot be used to predict future periods for the same objects. Consequently, Sela and Simonoff (2011) propose a

novel random effects expectation maximization (RE-EM) algorithm, which accounts for the structure of longitudinal data and allows for prediction of future time periods and unbalanced panels. Hajjem et al. (2011) propose a comparable method to RE-EM – referred to as mixed effects regression tree (MERT) – which also first fits a tree without random effects and then updates the estimates with random effects until convergence. Three years later, Hajjem et al. (2014) proposed their mixed effects regression forest (MERF) model, which uses the same framework as MERT, with a random forest model as base learner instead of a regression tree to enhance predictive performance. The MERF model is widely used in various fields, including health care. For example, Sheen (2019) uses MERF to model infant weight gains trajectories using longitudinal clinical trial data, and Rekabdar et al. (2022) use MERF to identify individuals with alcohol and drug misuse in a screening program. For both studies MERF ranks amongst the best performing models. Moreover, Cochrane et al.'s (2021) study on the relation between sleep and performance using longitudinal data demonstrates that MERF also performs well in their ensemble algorithm when combined with a linear mixed-effects model.

Although RE-EM, MERT, and MERF can appropriately deal with the possible random effects of observation-level covariates – in contrast to Segal (1992) – none of these methods allow for noncontinuous data. To this end, Hajjem et al. (2017) propose a generalised mixed effects regression tree (GMERT), which is a tree-based approach that is suitable for noncontinuous data and can incorporate observation-level covariates and their potential random effects. This extension uses the penalised quasi-likelihood method and expectation maximization for the estimation and computation, respectively. When the random effects are non-negligible, RE-EM, MERT, MERF, and GMERT each outperform regression trees without random effects based on both real-world and simulated data (Hajjem et al., 2011, 2014, 2017; Sela and Simonoff, 2011). Lastly, Ngufor et al. (2019) also propose a model which integrates the random-effects structure of GLMM in nonlinear machine learning models. Specifically, they combine Sela and Simonoff's (2011) RE-EM estimation method with the structure of the GMERT model by Hajjem et al. (2017) to predict longitudinal change in haemoglobin A1c. Their proposed mixed-effects machine learning (MEml) method allows for implementation of random forests, model-based recursive partitioning, conditional inference trees, or gradient boosting machines to estimate the fixed-effects part of the models. For an extensive review on the (mathematical) similarities and difference between RE-EM, (G)MERT, MERF and MEml, see Appendix B.

**Boosted tree algorithms** One way to improve predictions in machine learning is through ensemble methods, such as Chen and Guestrin's (2016) XGBoost algorithm. The premise of boosting is to sequentially add weak base learners and to iteratively adjust the weight of each these base learners according to the prediction errors in the previous iteration, to eventually create a single strong classifier. The superior performance of ensemble methods has inspired many boosted alternatives to existing algorithms, such as boosted (non)-linear mixed-models (Griesbach et al., 2021; Sigrist, 2020; Tutz and Groll, 2010), boosted additive mixed-models (Groll and Tutz, 2012), and boosted poisson regression (Lee, 2021).

Boosted *tree-based* algorithms have only recently gained popularity in longitudinal health care data. Like most machine learning algorithms, XGBoost does not inherently account for longitudinal structures

of data sets, and its performance is highly dependent on the chosen training data in case of violation of the iid data assumption. However, some researchers continue to employ machine learning approaches for data inference without taking (possible) violations into account. For example, Ryu et al. (2020) employ XGBoost using a combination of cross-sectional and longitudinal data to predict dementia risk,[10] and choose their final model based on shapley values. However, they disregard the possibility of confounding effects of between-subject variability entirely, which could lead to misleading inference, as discussed in Ngufor et al. (2019). Therefore, the motivation behind their final chosen model could be invalid. Additionally, they could be wasting an opportunity to achieve increased performance through capturing temporal relations in the data (Sela and Simonoff, 2011; Hajjem et al., 2017). In similar fashion, Moore and Bell (2022) compare myocardial infarction predictions of XGBoost to logistic regression using panel data, and also use shapley values without any notion of the (possible) violation of the iid assumption.[11]

That said, according to Dundar et al. (2007), violation of the iid assumption should not matter much if the temporal dependency between samples is very weak and each cluster occurs with highly similar frequency. For example, it might not be necessary (or even beneficial) to explicitly account for temporal dependencies at the level of the algorithm itself in a setting similar to Panchavati et al. (2022), who compare the infection predictions of hospitalized patients using machine learning methods (including XGBoost). I.e., it might be acceptable to assume no temporal dependencies are present between observations even though they use longitudinal data, because the data is collected over such a short period of time (six hours).[12]

### 2.3.3 General mixed-effects machine learning models

Recently, Wu et al. (2022) proposed an algorithm to incorporate mixed-effects in longitudinal data for the prediction of disability trajectories. They use a growth mixture model to identify latent categories (disability trajectories), considering individual and population heterogeneity. Once each trajectory is defined, any machine learning model can be used to predict within these trajectories. In their specific application, XGBoost outperforms support vector machines, logistic regression, and ANN. Chowdhury and Tomal (2022) present a comparable framework, which divides a complex multivariate problem into several univariate problems using observed time points, after which they employ multiple statistical and machine learning models to obtain marginal and conditional models as base learners. They also propose to include a lagged dependent variable as covariate to incorporate temporal dependencies. In their appplication, an ensemble of six machine learning models including ANN performs best. Although Chowdhury and Tomal (2022) do not employ an XGBoost classifier, this could be easily implemented.

Thus, both methods extend algorithms developed for cross-sectional data to predict risk trajectories for repeated responses. However, both are exclusively made for binary dependent variables, which is unsuitable for our application. Fortunately, the (code to the) MERF framework discussed in Section

[10] They combine the open source OASIS-1 and OASIS-2 data, which is advised against by OASIS, thus their analysis might contain more flaws in data processing than discussed here.

[11] It should be noted that this is a working paper, which has not been peer reviewed.

[12] They do include the summary statistics of all values measured in the data set as covariates for continuously measured features in XGBoost, which might capture (some of) the correlation if present after all.

2.3.2 – which does allow for continuous dependent variables – has recently been extended such that any nonlinear estimator can be used for estimation of the fixed effects. Even though the GMERT and MEml algorithms (also discussed in Section 2.3.2) also allow for continuous dependent variables, our research employs the MERF framework based on relative performance, mathematical properties, compatibility (all of which is discussed in detail in Appendix B.6), and availability.[13]

# 3 Data

## 3.1 Dutch screening data

The data for this research is collected by the National Institute for Public Health and the Environment during the Dutch CRC screening program from 2014-2020, and provided by the Erasmus Medical Centre. For each individual who participated in the biennial screening, a maximum of four rounds of data are available. This analysis exclusively focuses on those who participated in one round only, or multiple *consecutive* rounds.

Given that this research explores, i.a., 'regular' machine learning models even though intra-individual correlation might be present, we introduce additional variables as input to allow for as much individual variation as possible. Inspired by Chowdhury and Tomal (2022), we include a lagged dependent variable as covariate (previous haemoglobin concentrations), both to incorporate temporal dependency between the haemoglobin values of an individual, and because Grobbee et al. (2017) find that an undetectable haemoglobin concentration two years ago decreases the current risk of having CRC. We also include the minimum and maximum haemoglobin value per individual over all rounds prior to the current time of screening.

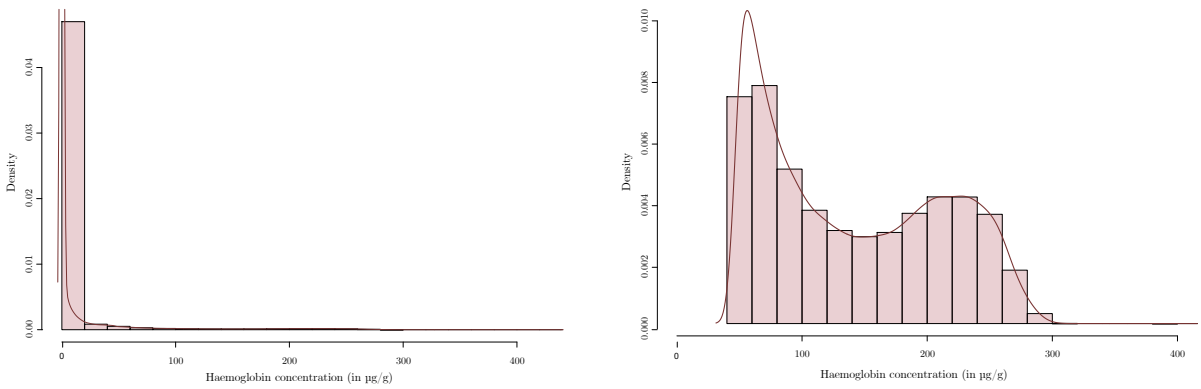
After data pre-processing (described in Appendix A.1), the data set contains 6,795,742 observations for 3,169,796 individuals, of which 52.4% are female. In total, 849,081 individuals participated in one round only, 1,145,920 individuals participated in two consecutive rounds, 1,044,359 individuals participated in three consecutive rounds, and 130,436 individuals participated in all four rounds. Table 1 shows the name, description, and range of all variables included in the data set. Figure 4a shows the distribution of haemoglobin concentration in the complete data set, which clearly shows the zero-inflation of our dependent variable. Figure 4b shows the distribution of haemoglobin concentrations amongst observations with positive FITs. We can distinguish a bimodal distribution in these positive FITs, with the largest peak between [47 $\mu$g/g; 80 $\mu$g/g] and a second peak around [180 $\mu$g/g; 260 $\mu$g/g].

[13] Only few of the methods discussed in this section, and Section 2.3.1 are widely available in commonly used statistical software. Click here for the code/documentation of RE-EM, MERF, and MEml. An adapted version of MERT (namely stochastic MERT) is available here.

Table 1: Original variables in the data set provided by the Erasmus Medical Centre.

| Variable name | Description | Range |
|---|---|---|
| `Age` | Age of respondent at time of screening | 55 – 77 |
| `Birth year` | Year of birth | 1938 – 1963 |
| `FIT number`[1] | Discrete sequence number of the FIT in current round | 1 – 4 |
| `Haemoglobin current` | Reported haemoglobin concentration in current round | 0 $\mu$g/g – 437.1 $\mu$g/g |
| `Haemoglobin max`[2] | Maximum reported haemoglobin concentration over all previous rounds | 0 $\mu$g/g – 47.0 $\mu$g/g |
| `Haemoglobin min`[2] | Minimum reported haemoglobin concentration over all previous rounds | 0 $\mu$g/g – 47.0 $\mu$g/g |
| `Haemoglobin previous`[2] | Reported haemoglobin concentration in previous round | 0 $\mu$g/g – 47.0 $\mu$g/g |
| `Haemoglobin threshold` | Threshold used to classify the FIT result | 47 $\mu$g/g |
| `ID` | Personal identification number | 1 – 3,710,672 |
| `Result` | Binary indicator for FIT result in current round | 0 (Favourable, 96.1%),<br>1 (Unfavourable, 3.9%) |
| `Round` | Discrete indicator for current round | 1 – 4 |
| `Sex` | Gender of participant | 0 (Male, 46.6%),<br>1 (Female, 52.4%) |
| `Stage`[3] | Stage of lesions in current round | 1 (Healthy, 0.8%),<br>2 (Non-advanced adenoma, 1.1%),<br>3 (Advanced adenoma, 1.7%),<br>4 (Colorectal cancer, 0.3%),<br>NA (Unknown, 96.1%) |

*Notes:* [1]`Fit number` is one-hot encoded, such that the resulting dummy variables are equal to one for the current FIT, and zero otherwise. [2]These variables are set equal to zero for individuals in round one. [3]`Stage` is only available for individuals who have had a colonoscopy, and is unknown otherwise.




(a) Complete data set

(b) Exclusively positive FITs

Figure 4: Densities and histograms of the haemoglobin concentration in the Dutch CRC screening program data set.

## 3.2 Missing values

Table 1 shows that `stage` is only known for 3.9% of the observations in our dataset. This low percentage is due to the fact that the stage of lesions at time of screening can only be determined by means of a colonoscopy, and only individuals with positive (unfavorable) FIT results are referred to such a follow-up procedure. As a result, 96.1% of our `stage` observations are missing.

Most statistical procedures are designed for complete data, and ANNs are no exception to this rule. There are adaptations to ANNs to account for missing values such as, e.g., the combination of deep networks with probabilistic mixture modes by Śmieja et al. (2018). This method is based on the premise that instead of calculating the activation function on a single data point, the first hidden layer in the network computes the *expected* activation of neurons.

XGBoost is apt to handle sparse data through its *sparsity-aware split finding* algorithm, which, in short, assigns a default direction to each branch for which a sample's feature is missing if a decision node splits on that feature, such that the path can continue (Chen and Guestrin, 2016). However, this built-in algorithm is unlikely to perform well with the large number of missing values in our data. Moreover, if would employ this adapted ANN and XGBoost model, we would not be able to validate whether differences in performance between both models are a result of the machine learning models themselves or a result of the inherent method of data processing. Thus, we are left with two options: either deleting or imputing `stage`.

If we delete all observations without reported stages, the resulting data set exclusively contains individuals above the cut-off value of 47 micrograms of haemoglobin per gram of faeces. As a result, the remaining data set would contain a fraction of the total available data, and it would be unrepresentative for the Dutch population. Moreover, deletion will likely result in poor predictive performance, as previous internal research by EMC shows that `stage` is a strong predictor of haemoglobin concentrations. Based on these arguments and the fact that identifying the current stage of lesions in an individual is the main purpose of screening, we opt to impute `stage`.

There are two major iterative approaches for multiple imputation in general missing data patterns: joint modeling and the fully conditional specification. Joint modeling assumes joint multivariate normality of all variables, which is inapt for imputing categorical variables, and therefore unsuitable for this analysis. In contrast, the fully conditional specification does not rely on multivariate normality, and applies a multivariate imputation model variable by variable, using a collection of conditional densities per incomplete variable (Van Buuren, 2018).

A popular data imputation method amongst the fully conditional specification is the Multiple Imputation via Chained Equations (MICE) by Van Buuren (2018), which is an often used and recommended method in healthcare literature (Ambler et al., 2007; Baneshi and Talei, 2011; Chowdhury et al., 2017; Faris et al., 2002; Jolani et al., 2015). We employ MICE to impute `stage`, using `haemoglobin current`, `result`, `age`, and `sex`. A required assumption for MICE is that the missing observations are missing at random, which means that there might be systematic differences between the missing and observed stages, but these can be entirely explained by other observed variables (Bhaskaran and Smeeth, 2014). This assumption is appropriate in our case because the missingness of `stage` is directly caused by the FIT result.

### 3.2.1 Additional data sets for the MICE algorithm

In an attempt to improve the accuracy of our imputation using MICE, we include two additional data sets where `stage` is always known – the '15 threshold' and 'MISCAN simulation' data set.

**'15 threshold' data set** In the screening data set described in Section 3, the current stage of lesions is known when the haemoglobin concentration exceeds 47 micrograms of blood per gram of faeces. However, the first round in the Dutch screening program in 2014 originally contained two groups of individuals: those who are admitted to the follow-up program if the haemoglobin concentrations exceed 47 $\mu$g/g, and (2) those who are admitted to the follow-up program based on a threshold of 15 $\mu$g/g. It is reasonable to assume that the data from these two groups originate from the same data generating process, since the data are simultaneously and identically obtained by the Dutch National Institute for Public Health and the Environment, and both data sets overlap in haemoglobin concentrations. Thus, the only difference between the data sets is the threshold that determines which individuals are admitted to the follow-up program. Consequently, the added benefit of including all observations with known `stage` from the '15 threshold' data set in the MICE algorithm is the gain of information on the current stage of lesions in individuals with `current haemoglobin` between $15 - 47$ micrograms of blood per gram of faeces,[14] based real-life data. This additional data set does not contain any missing values.

**'MISCAN simulation' data set** The second data set is obtained from a population simulation run in MISCAN-Colon. Specifically, we simulate two million individuals from 2014-2020, with the same sex ratio as in our original data set (see Table 1). The data inputs for the MISCAN-Colon model are calibrated on either the Dutch CRC screening program between 2014 and 2017, the CRC incidence between 2009 and 2014 provided by the Netherlands Cancer Registry, or a combination of these with international literature.[15,16] This 'MISCAN simulation' data set consists of 3,076,778 observations, where the current `stage` and `result` are always known, while `haemoglobin current` is always unknown.

Table 4 in Appendix A.2 reports descriptive statistics for both additional data sets. The combination of all three data sets results in 9,890,100 observations in total, of which 6,533,768 `stage` observations and 3,076,778 `haemoglobin current` observations are missing. Thus, `stage` is known for 33.9% of the observations in the combined data set, compared to 3.9% in the original data set.

### 3.2.2 The MICE algorithm applied to our data

In each iteration of MICE we first impute `haemoglobin current`, and then impute `stage`. In step one, we replace all missing values in the data set with a random draw from the data as temporary place holder.

[14] The '15 threshold' also includes a few observations for which the haemoglobin concentration is zero with known `stage`, due to medical interventions, but these are an exception.

[15] In specific, the MISCAN-Colon model uses the following parameters to simulate our individuals: the maximum number and localization of lesions, individual hazard rates, age of onset, the probability that an adenoma is non-progressive, adenoma size, dwell times of non-progressive adenomas, dwell times between the onset of a progressive adenoma and the transition to pre-clinical cancer, dwell times in preclinical, screen-detected and interval cancer stages, transition probabilities based on location and age, survival groups, age-dependent probabilities that a lesion is cured, and lastly the time to death if a lesion is not cured. For more details, please see the 'THESIS_crc_data.py' file in the attachments to this paper.

[16] It should be noted that by incorporating this data set into the MICE algorithm, we assume that the imposed data generating process in this 'MISCAN simulation' data set is at least similar to the true data generating process in the '15 threshold' and original data set. Since the CRC process in MISCAN-Colon is calibrated on Dutch observations, this assumption is likely to hold to some extent, but we cannot rule out the possibility that the data generating processes are different.

In step two, we set the place holder back to missing only for the variable we wish to impute. In step three, we replace these missing values using an appropriate imputation method – sampling and predictive mean matching in our case – using (part of) the remaining variables in the data set. Steps two and three are then repeated until all missing variables are imputed, at which point we completed one full cycle. We perform ten cycles in total, as per recommendation of Raghunathan et al. (2002). The observed data combined with the imputed values at the end of the tenth cycle constitute one imputed data set. This process is repeated to create 5 imputed data sets, such that a total of $5 \times 10$ cycles are performed.

In a best case scenario we would run all five of our models separately on each final imputed dataset, such that we obtain five sets of estimates per model. The difference in these estimates would then reflect the uncertainty in the missing data (to an extent). Unfortunately, doing so was not possible due to time constraints.[17] Instead, we compare the final distribution of all five imputed versions of `stage` to the stage distribution in the 'MISCAN simulation' data set. The imputed variable which most closely compares to the distribution of the MISCAN stage variable is then used as replacement for `stage` in the original data set.[18]

As a final step, all observations from the '15 threshold' and 'MISCAN simulation' are removed. The resulting data set is exactly the same as the data set described in Section 3.1, with the exception of the `stage` variable, which now no longer contains missing values. Appendix A.2 provides a more detailed (visual) explanation of the MICE algorithm specific to this paper.

# 4 Methodology

In the following section, we first present the mathematical background of ANN and XGBoost, followed by an explanation of the mixed-effects framework used in our mixed-effects ANN (MeANN) and mixed-effects XGBoost (MeXGBoost) model. We then expand on the Tweedie loss function, which is used in our fifth and final TweedieXGBoost model. Once all five of our models are introduced, we discuss how to tune each machine learning model, and how we estimate the dependent variable. Finally, this section concludes with mathematical formulations of the tests we use to obtain our results.

## 4.1 Artificial neural networks

ANNs, developed by Lippmann (1987), are inspired by the human brain, mimicking the way that biological neurons signal to one another. ANNs are comprised of an input layer, possibly one or more hidden layers, and an output layer. The input variables are related to the output variable(s) through a network of interconnected nodes, with associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network.

[17] Both of our mixed-effects models take approximately 2 weeks to run. Hence, running all five of our models four additional times would take us at least sixteen weeks (given that we only have one computer available with sufficient RAM to run the mixed-effects models), which is infeasible within our time frame.

[18] This choice was made after internal EMC discussions, and even though it is admittedly arbitrary, we deemed it as better than the equally arbitrary alternatives of either randomly selecting one of the imputed data sets or using a transformation of all five data sets. We discuss alternative methods to assessing the quality of our data imputation in Section 6.

The optimal weights are estimated when the ANN is fitted, such that a predetermined loss function is minimised – the root mean squared error (RMSE) in our case. The input layer of the ANN consists of $p$ nodes, where $p$ is equal to the number of explanatory variables. In our setting, the output node $\hat{f}(x)$ represents the predicted haemoglobin concentration.

To advance from one layer to another, the ANN uses activation function $h(\cdot)$, with the sum of the weights and the intercept (referred to as the bias) as input. Since each activation function has unique properties, we cannot determine which one is best in advance. Therefore, we consider three different activation functions in tuning (for more details, see Section 4.5): the identity activation function $h(x) = kx$ – with $k$ a fixed constant – the rectified linear unit (ReLu) activation function $h(x) = \max(0, x)$, and the sigmoid activation function $h(x) = \frac{e^x}{1+e^x}$. We do not consider the linear identity activation function for each hidden layer because this reduces any ANN model to a single layer ANN that can only capture linear relations in the data, which is unsuitable for our analysis. The derivative of the identity activation function is constant, such that the gradient is not influenced by the input. In contrast, small changes of input in the zero-centered bell-shaped derivative of the nonlinear sigmoid activation function creates large changes of output when the input is near 0. However, this activation function might lead to gradient saturation for very small or very large values. The ReLu activation function does not suffer from this *vanishing gradients* problem as long as the input is positive. Calculating this ReLu function is also less computationally expensive compared to the sigmoid activation function.

Hornik et al. (1989) show in their universal approximation theorem that an ANN with at least one hidden layer, and a large enough number of neurons, can approximate any finite-dimensional Borel measurable function up to any arbitrary accuracy. In other words, an ANN with zero hidden layers can only represent linear functions, whereas an ANN with one hidden layer can approximate *any* function with a continuous mapping with finite spaces using an ANN with one hidden layer. In practice, however, a network with multiple hidden layers can be more efficient. Therefore, we also tune the number of layers (for more details, see Section 4.5). In case of an ANN with $H$ nodes in the first hidden layer and $L$ nodes in the second, the values at each node are calculated as follows:

$$
\begin{aligned}
z_h^1 &= g\left(\sum_{j=1}^{p} w_{hj}^1 x_j\right) && \forall h \in \{1, \ldots, H\}, \\
z_l^2 &= g\left(\sum_{h=1}^{H} w_{lh}^2 z_h^1\right) && \forall l \in \{1, \ldots, L\}, \\
\hat{f}(x) &= g\left(\sum_{l=1}^{L} w_l^3 z_l^2\right),
\end{aligned}
$$

where $x_j$ represents each of the input regressors, $z_i^j$ represents the $i^{\text{th}}$ node of the $j^{\text{th}}$ hidden layer, and $w_{ik}^j$ is the weight of node $k$ on node $i$ in hidden layer $j$. Figure 5 shows an example of such an ANN.

Lastly, one of the risks of ANNs is that they tend to overfit on the training data. To mitigate overfitting, we use the efficient early stopping regularisation (Prechelt, 1998), and dropout in the hidden layer(s) (Srivastava et al., 2014).

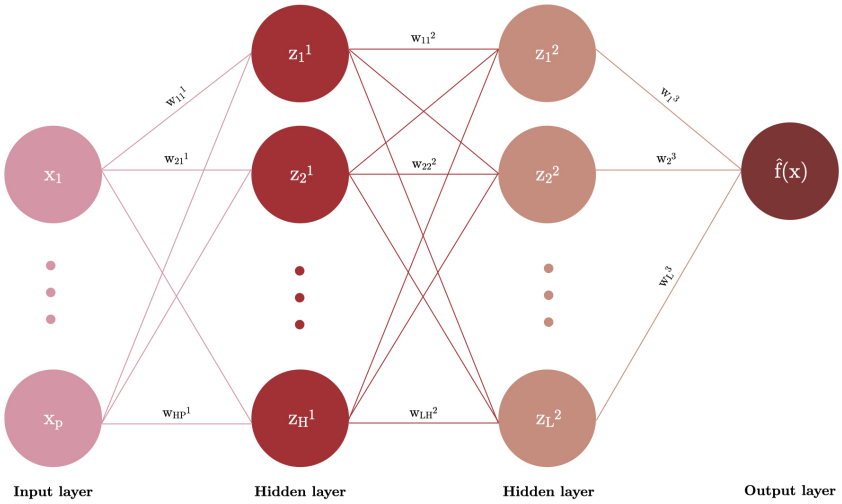



Figure 5: Example of an artificial neural network with two hidden layers and one output node.

## 4.2 XGBoost

For our second model, we consider the scalable XGBoost algorithm. The predicted values are obtained by sequentially building shallow classification and regression trees, such that each subsequent tree corrects for the prediction errors of the previous trees, using gradient descent. Adopting the notation of Chen and Guestrin (2016), we can write these predicted values as

$$\hat{y}_i = E(y_i|X_i) = \sum_{k=1}^{K} f_k\left(\mathbf{x}_i\right),$$

where each $f_k \in \left\{f(\mathbf{x}) = \omega_{q(\mathbf{x})}\right\}$ corresponds to an independent tree structure $q$ with leaf weights $\omega$.

To train the model, XGBoost minimises a negative log-likelihood loss function that measures the difference between the prediction $\hat{y}_i$ and the true outcome $y_i$ for each observation $i$, using a regularisation term $\Omega(f_k)$. Specifically, the algorithm minimises

$$\mathcal{L}(\hat{y}_i) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{1}$$

where $l(\hat{y}_i, y_i)$ is a differentiable convex training loss function. The regularisation term equals

$$\Omega(f_k) = \gamma T_{f_k} + \frac{1}{2}\lambda\|\omega_{f_k}\|^2 \qquad\qquad \forall f_k \in (f_1, ..., f_K),$$

where $\lambda$ is a L2 regularisation term on the leaf weights, $\gamma$ denotes the minimum loss reduction required to make a further partition on a leaf node of the tree, and $T$ denotes the number of leaves in the tree. This regularisation term penalises the complexity of tree $f_k$ and, as a result, aids in the reduction of over-fitting. To further mitigate over-fitting, we also implement bagging through subsampling of the training data and/or the features once in every boosting iteration. Just as for the ANN model, we use the RMSE as loss function, for our second model.

Instead of using traditional optimization methods to minimise the objective function in Equation 1,

Chen and Guestrin (2016) propose to train the model in an additive manner. Let $\hat{y}_i^{(r)}$ be the $i^{\text{th}}$ instance at the $r^{\text{th}}$ iteration, we can rewrite the objective function as

$$\mathcal{L}^{(r)} = \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{(r-1)} + f_r\left(\mathbf{x}_i\right)\right) + \Omega\left(f_r\right), \tag{2}$$

where the algorithm greedily adds a tree $f_r$ that most improves the model according to Equation 1. The rewritten objective function in Equation 2 is then optimised using second-order Taylor approximation, to guide the construction of the decision tree models. This approximation uses first and second order gradient statistics on the loss function with respect to the output of the previous tree, which clearly shows why XGBoost is a gradient boosting algorithm.[19] Thereafter, we can calculate the optimal weight $\omega$ of each leaf and the corresponding optimal value for fixed tree structures.[20]

### 4.2.1 Tweedie loss

The RMSE is a symmetric loss function, which may not be the optimal loss function to use for training in our application due to the zero-inflation of our dependent variable. Therefore, we employ XGBoost using the Tweedie loss function described in Yang et al. (2018) as our fifth model. Accordingly, the loss function $l(y_i, \hat{y}_i)$ in Equation 1 can be written as

$$\begin{aligned} l\left(y_i, \hat{y}_i, \rho\right) &= \sum_{i=1}^{N} -y_i \frac{\exp\left[\log(\hat{y}_i)(1-\rho)\right]}{1-\rho} + \frac{\exp\left[\log(\hat{y}_i)(2-\rho)\right]}{2-\rho} \\ &= \sum_{i=1}^{N} -y_i \frac{\hat{y}_i^{(1-\rho)}}{1-\rho} + \frac{\hat{y}_i^{(2-\rho)}}{2-\rho}, \end{aligned} \tag{3}$$

where $\rho$ denotes the Tweedie power parameter. Tweedie distributions are a family of distributions that include gamma, normal, Poisson and their combinations. The power parameter $\rho$ allows the user to specify which mean-variance relation to use. To attain the compound Poisson-gamma Tweedie distribution – which is non-negative with mass at zero – the power parameter should range between $[1, 2]$. In our research $\rho$ is set to 1.6.[21]

To illustrate how Tweedie loss might be more appropriate in a zero-inflated setting such as ours, recall that our dependent variable is always nonnegative. Thus from Equation 3, we can see that if $y = 0 \wedge \hat{y} > 0$, the returned loss for that observation is always strictly positive.[22] Therefore, to attain the lowest possible loss, $\hat{y}$ should be as close to zero as possible when $y$ equals zero. Since we require

[19] Specifically, we can rewrite Equation 2 as $\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}^{(t-1)}\right) + g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \Omega\left(f_t\right)$, where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ denotes the first order gradient and $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ the hessian.

[20] For a full mathematical formulation of these values, we refer to Chen and Guestrin (2016).

[21] This number is based on an exploratory analysis on the training data using one Hyperopt iteration (see Section 4.5), which resulted in two candidates: $\rho = 1.3 \wedge \rho = 1.6$, based on optimal values for the Tweedie loss and RMSE. We then chose $\rho = 1.6$ based on five Hyperopt iterations, as it provided more similar descriptive statistics of the predictions versus true values, and it resulted in a much lower Tweedie loss. The RMSE was similar for both values of the power parameter.

[22] The Tweedie variance power parameter $\rho$ controls how much to penalise the model for deviations from the ideal scenario in which the predicted value equals zero if the true value is equal to zero.

$\rho \in [1, 2]$, our models cannot predict exact zeroes, as this would result in division by zero.

### 4.3 Mixed-effects machine learning

To explore whether we can exploit the dependencies within individuals, we also model the mixed-effects counterparts of ANN and XGBoost using an adaptation to Hajjem et al.'s (2011) proposed mixed-effects framework. To explain this framework, we first introduce some notation adopted from Hajjem et al. (2017). Define $y_i = [y_{i1}, \ldots, y_{in_i}]^\top$ as the $n_i \times 1$ vector of responses for the $n_i$ observations in cluster $i = 1, \ldots, n$. Let $X_i = [x_{i1}, \ldots, x_{in_i}]^\top$ denote the $n_i \times p$ matrix of fixed-effects covariates, and let $Z_i = [z_{i1}, \ldots, z_{in_i}]^\top$ denote the $n_i \times c$ matrix of random-effects covariates.[23] The $c \times 1$ (unknown) vector of random effects for cluster $i$ are denoted by $b_i$. The proposed mixed-effects framework then follows the functional form:

$$\begin{aligned} y_i &= f(X_i) + Z_i b_i + \varepsilon_i \\ b_i &\sim N(0, D), \varepsilon_i \sim N(0, R_i) \\ i &= 1, \ldots, n, \end{aligned} \tag{4}$$

where $D$ denotes the covariance matrix of the random effects $b_i$, and $R_i$ denotes the (assumed diagonal) covariance matrix of the error terms $\varepsilon_i$. Inspired by Hajjem et al.'s (2014) extension of Hajjem et al. (2011), we estimate the fixed-effects component $f(X_i)$ in Equation 4 using machine learning models. Our research contributes to the existing literature by using both ANN and XGBoost in this mixed-effects machine learning framework, instead of the proposed random forests in Hajjem et al. (2014).

The mixed-effects machine learning models are estimated using an expectation-maximization approach, in which the random effects $Z_i b_i$ and the population-level fixed-effects $f(X_i)$ in Equation 4 are alternatively estimated. We first initialize the random effects $\hat{b}_i = 0$, and use this $\hat{b}_i$ to compute $y_i^* = y_i - Z_i \hat{b}_i$. We then train our machine learning model to estimate $\hat{f}(X_i)$, which is an estimate of $f(X_i)$ obtained from either an ANN or XGBoost model with $y_i^*$ as responses and $X_i$ as covariates.[24] This process repeats until convergence of the generalised log-likelihood.[25]

### 4.4 Forecasting

Recall that our study aims to identify which model is best suited to simulate haemoglobin concentrations, as a step towards evaluating personalised CRC screening strategies in MISCAN-Colon. The input for each of the models is the age, sex, birth year, and FIT sequence number at time $t$, and the stage, and maximum, minimum, and previous haemoglobin value $y_{t-1}^{Hb}$ at time $t-1$ to predict the haemoglobin

[23] Clearly, this notation is easily extended to longitudinal data, if we define each individual as it's own group, such that $j = 1, \ldots, n_i$ represent the observations for each individual $i = 1, \ldots, n$.

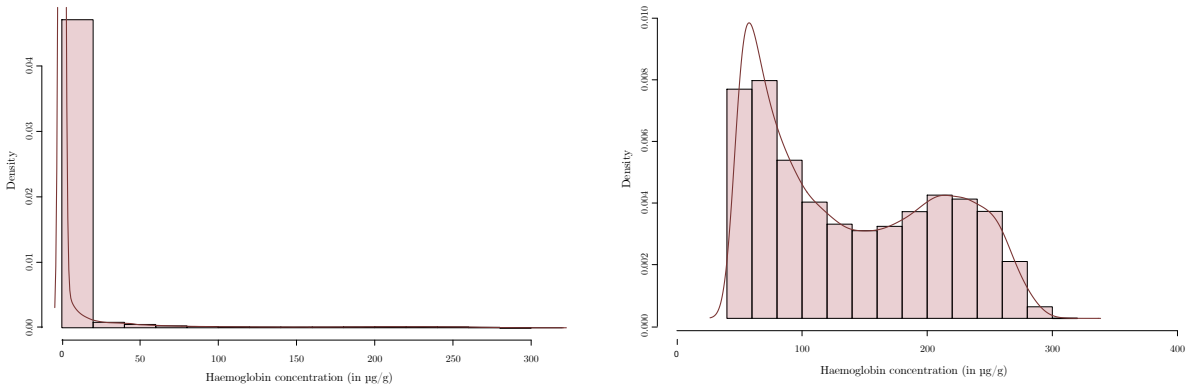
[24] Both mixed-effects models are trained using RMSE loss. We do not include an additional model using Tweedie loss because, within this mixed-effects framework, the machine learning models are used to estimate the fixed-effects based on the adjusted dependent variable $y_i^*$ for which we do not know the distribution. Therefore, we could not motivate the appropriateness of including Tweedie loss prior to implementation.

[25] For more details on the estimation procedure, we refer to Appendix B and Hajjem et al. (2011, 2014).

concentration $\hat{y}_t^{Hb}$ at time $t$.

To construct the training and test sets, we take into account that simulated individuals from the MISCAN-Colon model can be thought of as new individuals for which we do not have any prior information. Therefore, we create a 70/30 train/test split, ensuring that an individual only occurs once between these two groups.[26,27] Thus, the test set exclusively consists of individuals for whom no observations in any time period were included in the training data, and vice versa.

Figure 6 shows the distribution of the dependent variable in the test set, which is near identical to that of the complete dependent variable in Figure 4b. For a numeric comparison of the distribution of the dependent variable between the full data, training, and test set, and a zoomed in rendition of Figure 6a, see Table 6 and Figure 12 in Appendix A, respectively.




(a) Complete test set

(b) Exclusively positive FITs

Figure 6: Densities and histograms of the haemoglobin concentration in the test set.

We then create four folds of the training data using stratified $k$-fold cross validation on the dependent variable, i.e., we create four folds with a 75/25 train/validation split based on the distribution of haemoglobin concentrations. Similar to the train/test split, we create the train/validation split such that all observations per individual exclusively occur in either the training or validation fold, never in both folds. These train and validation folds are then used to perform hyperparameter tuning, which we will elaborate on in the following Section 4.5.

Moreover, given the non-negative nature of the haemoglobin concentrations, we cast all negative predictions of the validation and test data to zero. This makes no difference for the ANN models, since all ANN predictions lie within the training range per definition. However, recall that for the XGBoost models, each subsequent tree after the first iteration in XGBoost are based on predicting the error of

[26]Ideally, we would also use cross-validation or repeated sampling for the train/test split to evaluate the uncertainty in the test error to some extent, as the results might be sensitive this split. Using such nested cross-validation, however, was computationally infeasible within time constraints.

[27]One of the strengths of the mixed-effects models is the ability to distinguish between predictions for new individuals versus existing individuals. Therefore it may be interesting to also investigate the performance of all models on a test set with individuals whose previous observations are included in the training set as complementary analysis. The enclosed code already facilitates implementation for an additional test set of any desired number of observations (currently coded as 10,0000), with equal part individuals who have participated for two, three, and four consecutive rounds.

the previous tree(s). Consequently, only the initial tree is restricted to the training domain of of the dependent variable, but the sum across gradient boosted trees need not be. Thus, it may be possible to have predictions outside of the training range.

### 4.5 Tuning

Every machine learning model has parameters that are fixed before the learning process begins, referred to as hyperparameters. Since these parameters are predetermined, and not estimated, we tune these parameters to asses which set of hyperparameters results in the best model performance. In our research, we cross-validate the hyperparameters of the different models using a Bayesian search called Hyperopt by Bergstra et al. (2013) using four (stratified) folds. Just as with random grid search, the Hyperopt algorithm minimises the (RMSE or Tweedie loss) objective function in training through iterating over a search space – which defines the range of values a given hyperparameter can take – for a predetermined number of times. However, the difference between these two tuning methods is that random grid search randomly iterates over the search space, whereas the Hyperopt method can be seen as an exploration/exploitation strategy. That is, the algorithm starts by exploring the performance across the candidate hyperparameter space, and subsequently randomly exploits the most promising subspace of hyperparameters.

Figure 7 shows a (simplified) illustration of Bayesian Hyperopt versus random grid search. The top two figures show that both methods start by randomly exploring the candidate hyperparameter space. The bottom two figures show that, after a predetermined set of iterations, Hyperopt switches to a guided search in seemingly promising subspaces in terms of cross-validated performance measures (the red region in the figure), whereas random grid search randomly continues through the search space. Bergstra et al. (2013) show that for the same number of iterations, their Hyperopt method can lead to better hyperparameter settings than random search.[28]

[28] For computational efficiency one might also consider Putatunda and Rama's (2018) randomised Hyperopt. This method first randomly samples a predetermined fraction $\phi \in [0, 1]$ from the validation train fold without replacement, and then performs a Hyperopt iteration on this sampled fold. In their application, they show that the loss in performance is limited, while drastically decreasing computation time, allowing for more Hyperopt iterations. However, due to the zero-inflated nature of our data, we opted against using randomised Hyperopt to ensure the dependent variable in each train fold follows a similar distribution.

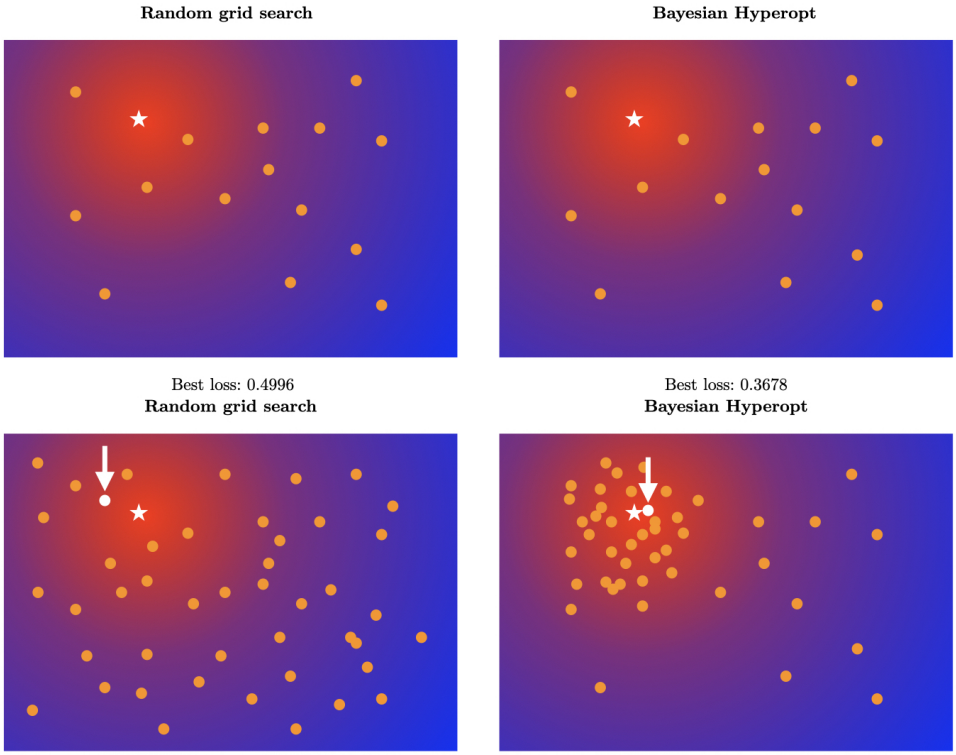



Figure 7: Simplified illustration of Random grid search (left) and Bayesian Hyperopt (right) after 15 and 45 iterations, respectively.

*Notes:* The white star denotes the hypothetical (unknown) optimal loss. The white dots with arrows denote the best loss with each respective method, the exact values of which are shown in the middle of the figure.

The combination of hyperparameters determines the allowed complexity of the model, and determines the extent to which we control for overfitting. For the (Me)ANNs, we tune the number of hidden layers, dropout rate per layer, activation function, number of neurons, batch size, and the learning rate. Since we already account for overfitting with early stopping and dropout, we do not consider weight decay. For the (Me)XGBoost models we tune the number of trees, maximum tree depth, learning rate, L2 regularisation term on weights, and the fraction of randomly selected training samples and features. In addition, data normalisation might be necessary, as Jayalakshmi and Santhakumaran (2011) show that the performance of ANNs are contingent on normalisation of the explanatory variables. The XGBoost models do not require normalisation, as the base learners are trees and no monotonic function of feature variables will change how the trees are formed. However, tree-based models can be sensitive to outliers, therefore it might be beneficial to perform a (robust) normalisation on the data regardless. We consider four distinct normalisation schemes: no normalisation, min-max normalisation, standardisation, and robust standardisation using the median and $25\% - 75\%$ interquantile range. For an overview of hyperparameter names, search spaces, and optimal values per model, see Appendix C. These reported optimal hyperparameter settings are based on a search using 125 Hyperopt iterations for each model.[29]

[29]With the exception of the optimal hyperparameter settings for the MeANN model. Training this model required too much memory, and repeatedly terminated after $\pm 77$ iterations.

### 4.6 Tests

To assess the individual predictions of each model, we calculate the RMSE, mean absolute error (MAE), and median absolute error (MedAE). However, one must take into account the large zero-inflation of the dependent variable when evaluating these measures. Namely, if our models only predict zeroes, the predictions would still be correct for more than 87% of the observations (see Table 6 in Appendix A). To allow for a more fair evaluation, we include two additional measures, which are separately calculated for observations above, and below the threshold of 47 $\mu g/\text{g}$.

First, we calculate the percentage correctly specified (PCC), which is defined as the percentage of predictions that are below (above) this threshold when the true value is also below (above) this threshold. Thus, the PCC below and above the threshold represent the percentage of true negatives and true positives, respectively.

We also include the percentage deviation (PDev). The PDev is based on an interval $w$ of either 3 or 10 micrograms of haemoglobin per gram of faeces, and indicates the percentage of predictions that lie within a range of $\pm w$ $\mu$g/g of the true haemoglobin concentration. We use this measure to gauge the precision of our model estimates.

## 5 Results

Table 2 shows the evaluation metrics discussed in the previous section for each of our five models. All models attain a relatively low RMSE, MAE, and MedAE. The MeXGBoost performs best based on RMSE and TweedieXGBoost performs best based on MedAE and MAE. The PCCs in Table 2 directly translate to the ability of our models to correctly predicting individual FIT results. All five models show great performance in correctly predicting negative FITs (represented as PCC_lb), with TweedieXGBoost and MeXGBoost as best performing models with true negative rate of 99.11% and 98.51%, and XGBoost the worst with a PCC_lb of 93.46%. All models perform worse in terms of the true positive rate, with PCCs ranging from 59.20% for TweedieXGBoost to 75.64% for XGBoost.

The PDevs indicate a similar relative performance amongst the models as the PCCs for both $w = 3$ and $w = 10$. That is, the MeXGBoost and TweedieXGBoost clearly outperform the other models below the threshold, whereas the XGBoost model provides the most accurate predictions when the true observations lie above the threshold, although differences between the five models are small in the latter case.

Thus, based on Table 2, it appears that our models perform well below the threshold for both classification and exact predictions. The models still attain relatively high PCCs ($> 59.19\%$) when the true haemoglobin concentrations exceed the threshold of 47 $\mu$g/g, but they perform rather poorly in terms of exact predictions (as measured by PDev $\in [1.80\%; 7.50\%]$ for both intervals $w$).

Table 2: Evaluation metrics on test data per model.

| | ANN | MeANN | XGBoost | MeXGBoost | TweedieXGBoost |
|---|---|---|---|---|---|
| *RMSE* | 28.26 | 28.36 | 29.73 | **22.41** | 22.91 |
| *MAE* | 10.21 | 10.17 | 13.89 | 6.94 | **6.72** |
| *MedAE* | 1.55 | 1.78 | 5.29 | 1.61 | **1.48** |
| *PCC_lb* | 95.79% | 95.85% | 93.46% | 98.51% | **99.11%** |
| *PCC_ub* | 71.60% | 71.65% | **75.64%** | 67.25% | 59.20% |
| *PDev_lb (3 μg/g)* | 62.67% | 65.80% | 39.41% | **81.04%** | 80.48% |
| *PDev_ub (3 μg/g)* | 2.06% | 2.13% | **2.40%** | 1.99% | 1.80% |
| *PDev_lb (10 μg/g)* | 82.92% | 84.86% | 76.03% | 91.52% | **92.23%** |
| *PDev_ub (10 μg/g)* | 6.76% | 6.83% | **7.50%** | 6.76% | 5.92% |

*Notes:* This table shows the root mean squared error (RMSE), mean absolute error (MAE), median absolute error (MedAE), and the percentage correctly specified (PCC) and percentage deviation (PDev) for observations with true haemoglobin concentrations below 47 $\mu$g/g (_lb) and above this threshold (_ub). The bold numbers in red colored cells denote the best performing model per model evaluation metric.

Table 3 shows descriptive statistics of the true haemoglobin concentrations and predicted haemoglobin concentrations prediction model. Based on this table, we see that the distribution of our MeXGBoost and TweedieXGBoost predictions seem to be the most comparable to the true distribution. However, hypothetically, while our predictions might follow the appropriate distribution, the individual predictions could still be incorrect. Therefore, we also present the median prediction errors per percentile in Figure 8, to show how the predicted values relate to the true haemoglobin concentrations.

To create this figure, we first sort the true dependent variable into percentiles before calculating the median prediction errors per model for each of these ten percentiles.[30] The true median haemoglobin concentrations – denoted by the crosses in Figure 8 – are equal to zero for the first 90% of data and equals 63.5 $\mu$g/g for the last 10% of data. The (Me)ANN models and the MeXGBoost models attain the lowest median prediction errors for the first 10% of data. The median (Me)ANN predictions are also lowest of all models for the second to fifth percentiles of data. The median prediction errors in the following 40% of data are lowest for the MeXGBoost and TweedieXGBoost models. Interestingly, although the XGBoost model showcases the largest median differences between predicted and true observations for the first 90% of the data, the median XGBoost prediction errors are lowest of all five models for the last 10%, though the differences are small.

Table 3: Descriptive statistics of predicted and true haemoglobin concentrations.

| | Mean | .05 | .10 | .25 | Median | .75 | .90 | .95 | Max |
|---|---|---|---|---|---|---|---|---|---|
| *True values* | 6.40 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.6 | 29.4 | 318.0 |
| *ANN* | 10.25 | **0.00** | **0.00** | **0.00** | **1.39** | 6.23 | 15.07 | 78.01 | 178.45 |
| *MeANN* | 10.16 | **0.00** | **0.00** | **0.00** | 1.61 | 5.46 | 14.96 | 77.26 | 183.28 |
| *XGBoost* | 14.62 | 0.08 | 0.97 | 2.02 | 5.13 | 10.14 | 36.45 | 81.50 | 246.36 |
| *MeXGBoost* | **6.59** | **0.00** | **0.00** | 0.75 | 1.56 | 2.39 | 8.07 | **20.40** | 266.33 |
| *TweedieXGBoost* | 5.91 | 0.17 | 0.19 | 0.66 | 1.43 | **2.16** | **7.53** | 16.88 | **358.74** |

*Notes:* This table shows the mean, median, maximum, and the 5%, 10%, 25%, 75%, 90%, and 95% percentiles of the true dependent variable and its predicted counterpart per machine learning model. The bold numbers in red colored cells denote the best performing model(s) per statistic.

[30] The mean and median prediction error indicate the same relative performances, hence why we only present the median prediction errors here. The mean prediction errors are shown in Figure 14 in Appendix D for completeness.

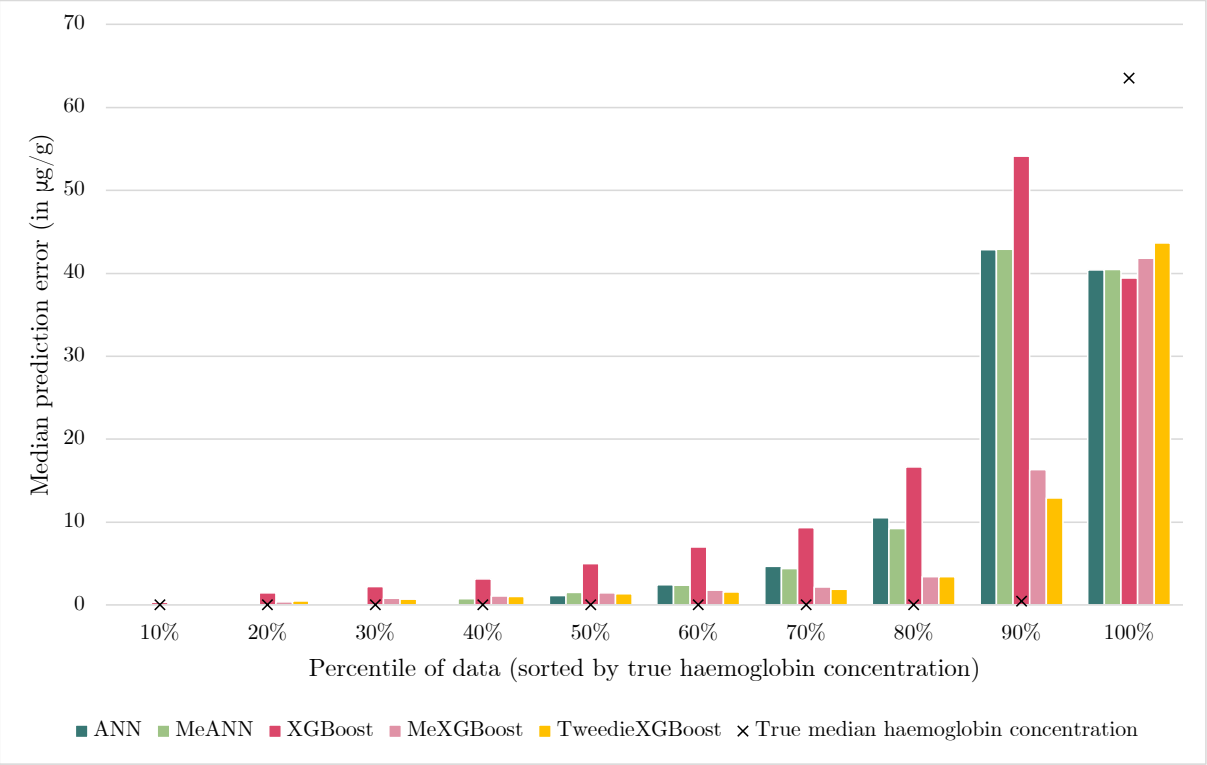


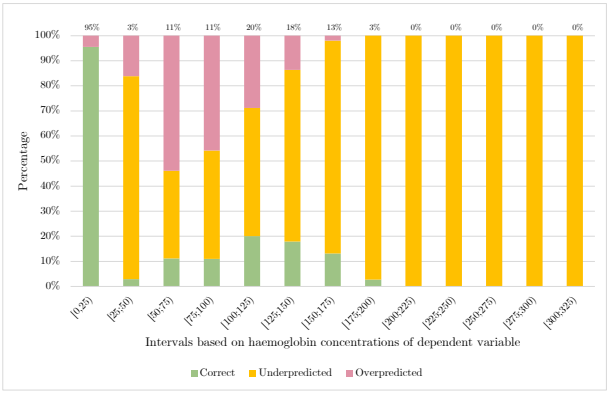

Figure 8: Median prediction errors per model (represented by the bars) and true median haemoglobin concentrations (represented by the crosses) per tenth percentile.

*Notes:* To create this figure, we first sort the true dependent variable in the test set into ten buckets of equal size. We then calculate the median prediction error per bucket, per model, which represents the height of each bar. The black crosses denote the median of the true dependent variable per tenth percentile.

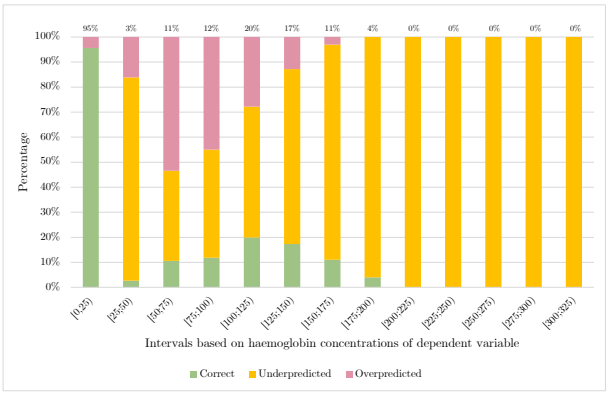
To further investigate the predictive performance of our models also for the higher haemoglobin concentrations, we split the data into bins of width 25 $\mu$g/g based on true haemoglobin concentrations. Specifically, Figure 9 shows the percentage of correctly classified, underpredicted and overpredicted oberservations in percentages per interval.[31] The first of these intervals contains 94.58% of the observations of true dependent variable (see Table 6 in Appendix A). Figure 9 shows that, for these observations, TweedieXGBoost and MeXGBoost correctly predict 99% and 98% of the observations to be in this interval, respectively. The (Me)ANN models follow with 95%, and the worst performing model is XGBoost with 91% of observations predicted correctly predicted in this interval.

Then, for the second interval, the share of correctly predicted observations drops below 7%. This percentage increases again for observations with true haemoglobin concentrations between 50 $\mu$g/g and 150 $\mu$g/g, before it gradually decreases until none of our models provide any correct predictions anymore for haemoglobin concentrations over 225 $\mu$g/g, which can be seen from the entirely yellow bars for the last four intervals.

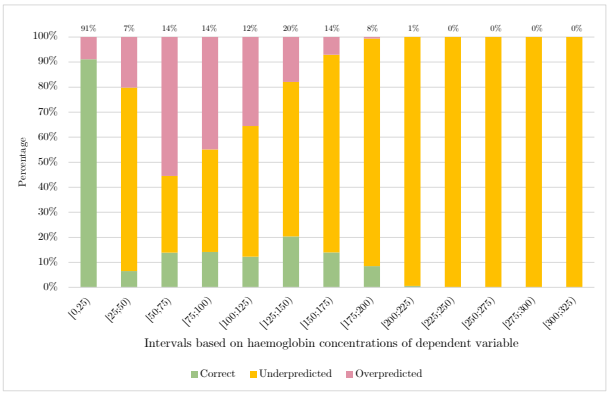
[31] Due to the number of observations in the test set and the relatively small scale of our predictions, we cannot show our results in a 2D plot. However, for the interested reader, Figure 13 in Appendix D displays a more in depth version of Figure 9, which shows exactly in which bin our predicted values lie with respect to the true haemoglobin concentration.



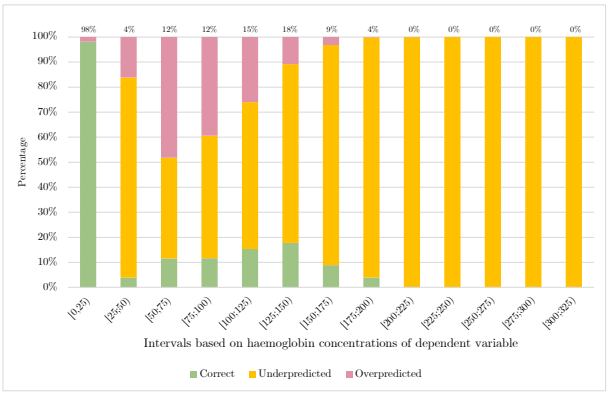

(a) ANN



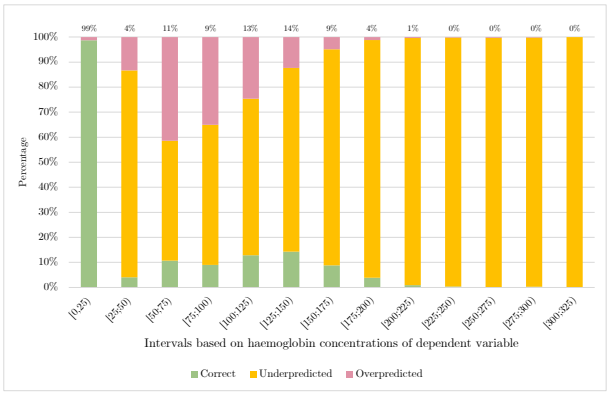

(b) MeANN




(c) XGBoost




(d) MeXGBoost




(e) TweedieXGBoost

Figure 9: Percentage of observations that are correctly predicted, underestimated and overestimated per interval, for all five models.

*Notes:* The intervals in this figure are created based on the true haemoglobin concentrations in the test set, such that each bar in each subfigure shows how the predicted values per model correspond to the true values in that interval.

Figure 9 also shows the increase in the percentage of observations that are underpredicted by our models from the third interval onward, i.e., for true haemoglobin concentrations over 50 $\mu$g/g. This progression of underprediction is also visible in Figure 10, which shows the median predicted values plotted against the median true values, calculated based on 16 intervals with equal width of 20 $\mu$g/g.[32]

[32]The mean and median prediction error indicate the same relative performances, hence why we only present the median prediction errors here. The mean plot is shown in Figure 15 in Appendix D.

In the best case scenario our predictions would lie on the gray dotted 45-degree line, in which case the predictions are exactly equal to the true values. However, all five of our models drop beneath this line more and more as the true dependent variable increases. Figure 16 in Appendix D – which is the same as Figure 10 but presented with interquartile ranges – shows that once the true haemoglobin exceeds 160 $\mu$g/g, even the interquartile ranges of our median predicted values are fully below the 45-degree line.

Figure 10 also shows that the median predicted values never exceed 200 $\mu$g/g. In fact, Figures 17 and 18 in Appendix D show that, even if we decrease the interval width used to calculate the mean and median from 20 to 2 and 4, respectively, the median (and mean) predictions of our models still never exceed 200 $\mu$g/g.

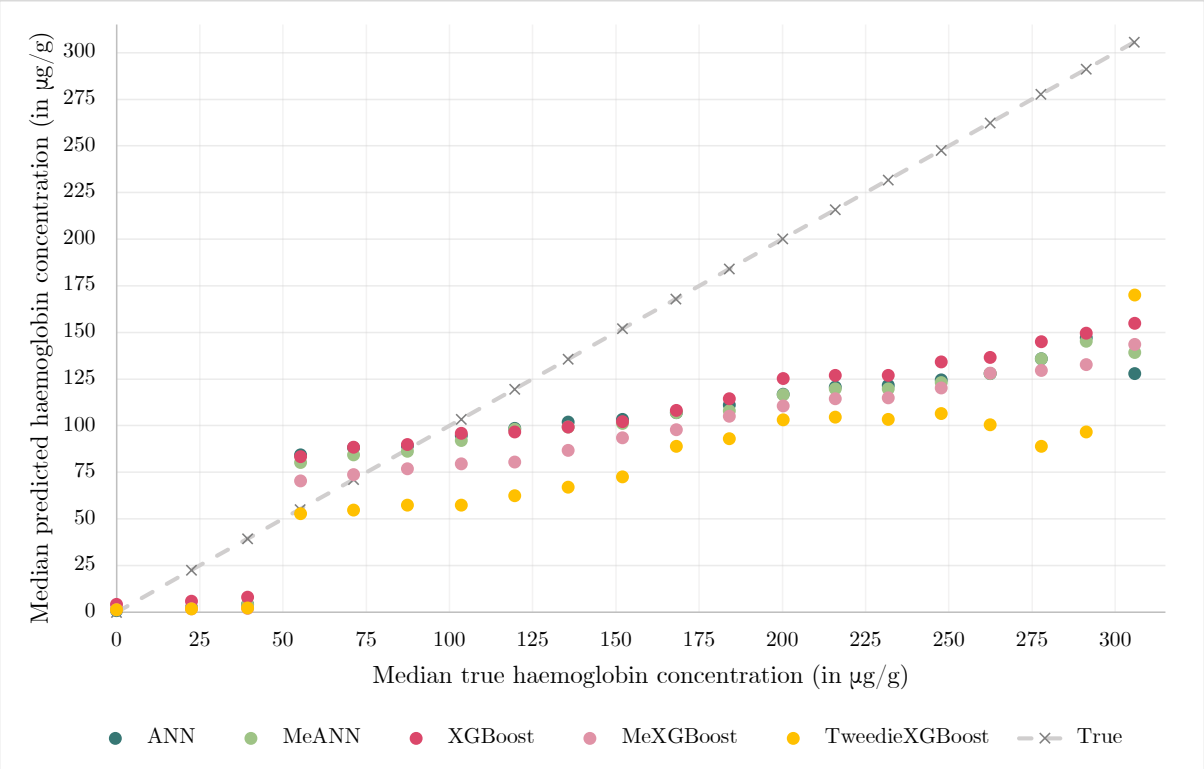



Figure 10: Median predicted versus median true values per model, calculated based on 16 intervals.

*Notes:* To create the figure above, we first sort the data set into 16 intervals, each with a width of 20 micrograms haemoglobin per gram of faeces, to reduce the number of data points. We then plot the median of the true dependent variable against its five predicted counterparts for each of these intervals. The gray 45-degree line is a line of reference, and shows all points where median predicted values are exactly equal to median true values. The crosses on this dashed line denote the medians of the true dependent variable per interval.

# 6 Discussion

Based on the results in the previous section, (1) it seems that our models have difficulty making accurate predictions, especially if the true haemoglobin concentration lies above the threshold, and (2) there appears to be a jump in median predictions around the threshold. In this section, we discuss the possible reasons behind these findings, along with solutions for future research.

With respect to our first point, recall the histogram in Figure 4b; the distribution of haemoglobin concentration in these positive FITs is bimodal, with a first peak between [47 $\mu$g/g; 80 $\mu$g/g] and a second peak (approximately half its size) between [180 $\mu$g/g; 260 $\mu$g/g]. However, Figure 10 shows that the median predicted values of our models never exceed 200 $\mu$g/g. We identify three possible causes.

It could be possible that our models interpret the second peak as noise rather than a true relation in the data, since the share of observations with haemoglobin concentrations over 200 $\mu$g/g in the training set is relatively low (1.09% of total training data). However, Figure 9 shows that our models are able to provide correct predictions for at least $1/10^{\text{th}}$ of observations between [100 $\mu$g/g; 125 $\mu$g/g) and [125 $\mu$g/g; 150 $\mu$g/g), even though only 0.35% and 0.28% of total observations in the training data lie within these intervals, compared to the (relatively) much larger share of observations that lie between [200 $\mu$g/g; 225 $\mu$g/g) and [225 $\mu$g/g; 250 $\mu$g/g) in the training data (0.41% and 0.40%, respectively).

Thus, it may be more likely that the decrease in model performance above the threshold is because our explanatory variables are not informative enough. Following the current Dutch screening set-up, all individuals who receive a positive FIT are referred to a follow-up program, and are therefore excluded from the data set from the next screening round onward. Consequently, all observations for which the true haemoglobin concentration lies above the threshold are final observations of individuals, such that the lagged haemoglobin concentration, and the maximum and minimum haemoglobin concentrations over previous rounds always lie below 47 $\mu$g/g. It could be possible that predicting high concentrations when these explanatory variables are below the threshold is simply infeasible. Additionally, if the the positive FIT occurs in the first round of an individual, the explanatory variables only contain real-life information on age, sex, birth year, FIT sequence number, and stage, which may not be informative enough to predict these haemoglobin concentrations.[33]

Alternatively, it may be possible that the second peak in haemoglobin concentrations is due to faulty data collection, which is not accounted for in our data set. For example, the FIT can only measure concentration levels until the binding material in the test tube of the FIT is exhausted. Thus, it is possible that haemoglobin concentrations are truncated above a certain limit, depending on the amount of binding material present. This phenomenon is referred to as the prozone effect. Conti et al. (2014) shows that the FIT used in the Dutch screening program (FOB-Gold) suffers from this prozone, or hook, effect for haemoglobin concentrations above approximately 400 $\mu$g/g. They show that the reported values for concentrations above 400 $\mu$g/g are approximately equal to half of the true haemoglobin concentration. I.e., the prozone effect should be visible from approximately 200 $\mu$g/g in terms of reported values. In fact, Piggott et al. (2014) find that the prozone effect may already be visible from 150 $\mu$g/g. Both of these studies would be logical explanations for the peak in haemoglobin density between [180 $\mu$g/g; 260 $\mu$g/g], with mass point around 220 $\mu$g/g. Our predictions for haemoglobin concentrations above 250 $\mu$g/g may improve if we have information on, e.g., the amount of binding material and faeces in the test tube.

[33] This may explain the large decrease in correct predictions between between 25 $\mu$g/g $-$ 50 $\mu$g/g in Figure 9, even though the second largest share of observations of the total training data lie between this interval.

As for our second point, the jump in median predictions around the threshold may be a result of the imputation technique – even though our models perform best for observations between 0 and 25 $\mu$g/g, which is where the vast majority of imputed observations are. Due to the black-box nature of our models, there is no straightforward way to interpret our results. Therefore, we suggest that the quality of our imputation technique should be assessed for further research. We propose three ways to evaluate our model outcomes based on the imputed data set.

First, as mentioned in Section 3.2, it would be desired to run our analyses on all five imputed data sets to assess to assess the extent to which our estimates are sensitive to randomness in the imputed values. Second, one should use interpretation techniques to evaluate our model predictions (e.g., using *post hoc* interpretability methods, see Molnar (2019)), to see what variables are most influential. Third, one could evaluate the effect of imputation method on our estimates through, for example, training the ANN adaptation by Śmieja et al. (2018) described in Section 3.2 on the original data set before imputation, and subsequently comparing its predictions to our ANN model on the imputed data set. One could also embed the ANN adaptation within the mixed-effects framework to eventually perform a similar comparison for our our MeANN model. Likewise, we could train any of our XGBoost models on the original data set before imputation – which would then use their *sparsity-aware split finding algorithm* to account for the missing values (also see Section 3.2) – and compare the results to our version of that XGBoost model. Alternatively, instead of using additional models to evaluate our data imputation method, one could also add real-life data sets collected from other studies for which the current `stage` is always known (regardless of the threshold) within our MICE algorithm. One could use the Dutch data described in De Wijkerslooth et al. (2012), for example, which includes 1,256 observations, or the data of any of the 28 international papers described in Table 1 in Grobbee et al. (2022).

# 7 Conclusion

In our research, we use black-box machine learning models to predict haemoglobin concentrations in faecal immunochemical tests, as preliminary model development to eventually analyse personalised screening strategies using MISCAN-Colon (MIcrosimulation SCreening ANalysis) model.

MISCAN-Colon is a microsimulation model for the evaluation of colorectal cancer screening, which allows for the evaluation of different screening policies by comparing their costs and effectiveness, as well as assessing the risk of false positives and overdiagnosis on a simulated population before real-life enforcement (Loeve et al., 1999). This microsimulation model currently follows the guidelines of the national screening procedure in the Netherlands, and uses the sensitivity and specificity of the faecal immunochemical test results to simulate a positive or negative test result for simulated individuals, instead of haemoglobin concentrations.

Previous research by van den Berg (2021) indicates that mixed-effect machine learning models significantly outperform the benchmark mixed-effect zero-inflated negative binomial model in simulating haemoglobin concentrations in MISCAN-Colon. However, it is unclear whether this improvement in predictive performance is due to the inclusion of mixed-effects or due to the use of machine learning methods

in general.

In turn, our research analyses the contribution of the inclusion of random-effects to the predictive performance of black-box machine learning methods in predicting haemoglobin concentrations. We base our study on two main models. We use the non-parametric regression-based artificial neural network (ANN), originally developed by Lippmann (1987), and Chen and Guestrin's (2016) eXtreme Gradient Boosting (XGBoost) algorithm as machine learning models. We train a 'regular' ANN and XGBoost model, a mixed-effects ANN and XGBoost model using the framework proposed by Hajjem et al. (2014), and a final XGBoost model using the Tweedie loss function described in Yang et al. (2018) instead of the root mean squared error loss function used in the other four models. We focus on the relative performance of the 'regular' machine learning methods compared to their mixed-effects counterparts. Hence, this research aims to answer the following research questions:

RQ1a Does the introduction of random-effects in machine learning models lead to better performance, i.e., do mixed-effects machine learning models outperform 'regular' machine learning models?

RQ1b Which model is best suited for predicting the haemoglobin concentration based on the data set provided by the Erasmus Medical Centre?

All five of our models perform well in terms of specificity, with MeXGBoost and TweedieXGBoost as top performing models, correctly predicting negative FIT results for approximately 99% of the time. These two models also provide the most precise predictions below the threshold, and they most accurately capture the zero inflation of the data compared to (Me)ANN and XGBoost.

Moreover, even though the XGBoost model shows the worst performance below the threshold, it provides the most precise estimates above the threshold and attains the lowest median prediction error for the last $10^{\text{th}}$ percentile of the data compared to the (Me)ANN models and XGBoost variations – although differences are small. The XGBoost model also attains the highest true positive rate (76%). In comparison, the MeXGBoost and TweedieXGBoost achieve the lowest true positive rates of 67% and 59%, respectively.

Based on these findings in combination with the fact that TweedieXGBoost and MeXGBoost provide correctly predicted observations within the appropriate interval for 97% of total observations in the test set – compared to 94% for (Me)ANN and 91% for XGBoost – we identify TweedieXGBoost and MeXGBoost as best performing models for predicting the haemoglobin concentration based on the data set provided by the Erasmus Medical Centre. Our research is particularly valuable for clinicians interested in the prediction of true negatives, or in the accurate prediction of lower FIT values, in which case we recommend using either MeXGBoost or TweedieXGBoost. Taking computational time into consideration, we might prefer the latter. However, one should first execute the proposed complementary analysis in Section 4.4, to determine whether the differences in performance between our mixed-effects and 'regular' XGBoost model are due to the inclusion of random-effects – in which case using TweedieXGBoost would result in faulty inference – or whether the intra-individual correlation appears to be negligible.

Furthermore, although the ANN models are never the worst performing, they also never attain the highest performance, and the differences between the ANN model and its mixed-effects component are

small and inconclusive. Additionally, even though our results clearly show that including mixed-effects to XGBoost increases the predictive performance on our data, changing the loss function from root mean squared error to Tweedie loss results in a similar positive change. Therefore, whether the inclusion of random-effects to machine learning models increases our performance, seems to depend on the machine learning model used to estimate the fixed-effects itself, and the loss function used in training.

Additionally, besides the obvious extension of repeating our research while taking (part) of our limitations from the discussion into account (e.g., through additional, more informative data), one could also investigate whether using different models per time period ahead would improve our predictive performance. Doing so could at least give new insights into which FIT (out of the maximum of four) can be most accurately predicted.

Finally, a natural extension of our research would be to implement our best performing model in the MISCAN-Colon model, to eventually individualise screening programs. However, given that none of our models provide very accurate predictions for higher haemoglobin concentrations, we recommend looking into a naive two-step method. First, one would use either TweedieXGBoost or MeXGBoost to predict whether the haemoglobin concentration should be zero, followed by a 'regular' XGBoost model trained on exclusively nonzero haemoglobin concentrations. Lastly, since changing the loss function from root mean squared error to Tweedie loss drastically increases the predictive performance of XGBoost below the threshold (for exact predictions in particular), it might also be interesting to create a MeXGBoost model using Tweedie loss in the first step of this hurdle model.

# References


Ambler, G., Omar, R. Z., and Royston, P. (2007). A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome. *Statistical Methods in Medical Research*, 16(3):277–298.

Baneshi, M. and Talei, A. (2011). Multiple Imputation in Survival Models: Applied on Breast Cancer Data. *Iranian Red Crescent Medical Journal*, 13(8):544.

Barton, M. B., Moore, S., Polk, S., Shtatland, E., Elmore, J. G., and Fletcher, S. W. (2001). Increased patient concern after false-positive mammograms. *Journal of General Internal Medicine*, 16(3):150–156.

Bergstra, J., Yamins, D., and Cox, D. D. (2013). Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms. In *Proceedings of the 12th Python in Science Conference*, volume 13, page 20. Citeseer.

van den Berg, D. (2021). Simulation of haemoglobin concentrations in MISCAN-Colon using a mixed-effect machine learning model. Master's thesis, Erasmus University Rotterdam.

Bhaskaran, K. and Smeeth, L. (2014). What is the difference between missing completely at random and missing at random? *International Journal of Epidemiology*, 43(4):1336–1339.

Botteri, E., Iodice, S., Bagnardi, V., Raimondi, S., Lowenfels, A. B., and Maisonneuve, P. (2008). Smoking and Colorectal Cancer: A Meta-analysis. *Journal of the American Medical Association*, 300(23):2765–2778.

Brasso, K., Ladelund, S., Frederiksen, B. L., and Jørgensen, T. (2010). Psychological distress following fecal occult blood test in colorectal cancer screening–a population-based study. *Scandinavian Journal of Gastroenterology*, 45(10):1211–1216.

Brenner, H., Stock, C., and Hoffmeister, M. (2014). Effect of screening sigmoidoscopy and screening colonoscopy on colorectal cancer incidence and mortality: systematic review and meta-analysis of randomised controlled trials and observational studies. *British Medical Journal*, 348.

Brodersen, J. and Siersma, V. D. (2013). Long-Term Psychosocial Consequences of False-Positive Screening Mammography. *The Annals of Family Medicine*, 11(2):106–115.

Bronner, M. P. and Haggitt, R. C. (1993). The Polyp-Cancer Sequence: Do All Colorectal Cancers Arise from Benign Adenomas? *Gastrointestinal Endoscopy Clinics of North America*, 3(4):611–622.

Capitaine, L., Genuer, R., and Thiébaut, R. (2021). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, 30(1):166–184.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. (2016). Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. In *Machine Learning for Healthcare Conference*, pages 301–318. Proceedings of Machine Learning Research.

Chowdhury, M. H., Islam, M. K., and Khan, S. I. (2017). Imputation of Missing Healthcare Data. In *20th International Conference of Computer and Information Technology*, pages 1–6. IEEE.

Chowdhury, R. I. and Tomal, J. H. (2022). Risk prediction for repeated measures health outcomes: A divide and recombine framework. *Informatics in Medicine Unlocked*, 28:100847.

Cochrane, C., Ba, D., Klerman, E. B., and Hilaire, M. A. S. (2021). An Ensemble Mixed Effects Model of Sleep and Performance. *Journal of Theoretical Biology*, 509:110497.

Conti, N., Gramegna, M., De Cunto, C., La Motta, M., Longo, G., Lucini, R., Dioli, R Cugini, A., and Anelli, M. (2014). Prozone effect check: a mandatory feature for automated Fecal Immunochemical Test (FIT) for Haemoglobin.

De Wijkerslooth, T., Stoop, E., Bossuyt, P., Meijer, G., van Ballegooijen, M., Van Roon, A., Stegeman, I., Kraaijenhagen, R., Fockens, P., Van Leerdam, M., et al. (2012). Immunochemical fecal occult blood testing is equally sensitive for proximal and distal advanced neoplasia. *Official journal of the American College of Gastroenterology| ACG*, 107(10):1570–1578.

Ding, H., Lin, J., Xu, Z., Chen, X., Wang, H. H., Huang, L., Huang, J., Zheng, Z., and Wong, M. C. (2022). A Global Evaluation of the Performance Indicators of Colorectal Cancer Screening with Fecal Immunochemical Tests and Colonoscopy: A Systematic Review and Meta-Analysis. *Cancers*, 14(4):1073.

Dundar, M., Krishnapuram, B., Bi, J., and Rao, R. B. (2007). Learning Classifiers When the Training Data Is Not IID.

Faris, P. D., Ghali, W. A., Brant, R., Norris, C. M., Galbraith, P. D., Knudtson, M. L., Investigators, A., et al. (2002). Multiple imputation versus data enhancement for dealing with missing data in observational health care outcome analyses. *Journal of Clinical Epidemiology*, 55(2):184–191.

Fazio, V. W., Tekkis, P. P., Remzi, F., and Lavery, I. C. (2004). Assessment of operative risk in colorectal cancer surgery: the Cleveland Clinic Foundation colorectal cancer model. *Diseases of the Colon & Rectum*, 47(12):2015–2024.

Frampton, M., Law, P., Litchfield, K., Morris, E., Kerr, D., Turnbull, C., Tomlinson, I., and Houlston, R. (2016). Implications of polygenic risk for personalised colorectal cancer screening. *Annals of Oncology*, 27(3):429–434.

Griesbach, C., Säfken, B., and Waldmann, E. (2021). Gradient boosting for linear mixed models. *The International Journal of Biostatistics*, 17(2):317–329.

Grobbee, E. J., Schreuders, E. H., Hansen, B. E., Bruno, M. J., Lansdorp-Vogelaar, I., Spaander, M. C., and Kuipers, E. J. (2017). Association Between Concentrations of Hemoglobin Determined by Fecal Immunochemical Tests and Long-term Development of Advanced Colorectal Neoplasia. *Gastroenterology*, 153(5):1251–1259.

Grobbee, E. J., Wisse, P. H., Schreuders, E. H., van Roon, A., van Dam, L., Zauber, A. G., Lansdorp-Vogelaar, I., Bramer, W., Berhane, S., Deeks, J. J., et al. (2022). Guaiac-based faecal occult blood tests versus faecal immunochemical tests for colorectal cancer screening in average-risk individuals. *Cochrane Database of Systematic Reviews*, (6).

Groll, A. and Tutz, G. (2012). Regularization for Generalized Additive Mixed Models by Likelihood-Based Boosting. *Methods of Information in Medicine*, 51(02):168–177.

Habbema, J., van Oortmarssen, G., Lubbe, J. T. N., and van der Maas, P. (1985). The MISCAN simulation program for the evaluation of screening for disease. *Computer Methods and Programs in Biomedicine*, 20(1):79–93.

Haghani, S., Sedehi, M., and Kheiri, S. (2017). Artificial Neural Network to Modeling Zero-inflated Count Data: Application to Predicting Number of Return to Blood Donation. *Journal of Research in Health Sciences*, 17(3):392.

Hajjem, A., Bellavance, F., and Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, 81(4):451–459.

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

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

Hanley, J. A. (2005). Analysis of Mortality Data from Cancer Screening Studies: Looking in the Right Window. *Epidemiology*, pages 786–790.

Hewitson, P., Glasziou, P., Watson, E., Towler, B., and Irwig, L. (2008). Cochrane Systematic Review of Colorectal Cancer Screening Using the Fecal Occult Blood Test (Hemoccult): An Update. *Journal of the American College of Gastroenterology*, 103(6):1541–1549.

Holme, Ø., Bretthauer, M., Fretheim, A., Odgaard-Jensen, J., and Hoff, G. (2013). Flexible sigmoidoscopy versus faecal occult blood testing for colorectal cancer screening in asymptomatic individuals (Review). *Cochrane Database of Systematic Reviews*.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural networks*, 2(5):359–366.

Jayalakshmi, T. and Santhakumaran, A. (2011). Statistical Normalization and Back Propagation for Classification. *International Journal of Computer Theory and Engineering*, 3(1):1793–8201.

Jenniskens, K., De Groot, J. A., Reitsma, J. B., Moons, K. G., Hooft, L., and Naaktgeboren, C. A. (2017). Overdiagnosis across medical disciplines: a scoping review. *BMJ Open*, 7(12):e018448.

Jiang, Y., Yuan, H., Li, Z., Ji, X., Shen, Q., Tuo, J., Bi, J., Li, H., and Xiang, Y. (2022). Global pattern and trends of colorectal cancer survival: a systematic review of population-based registration data. *Cancer Biology & Medicine*, 19(2):175.

Jolani, S., Debray, T. P., Koffijberg, H., van Buuren, S., and Moons, K. G. (2015). Imputation of systematically missing predictors in an individual participant data meta-analysis: a generalized approach using mice. *Statistics in Medicine*, 34(11):1841–1863.

Kahi, C. J., Imperiale, T. F., Juliar, B. E., and Rex, D. K. (2009). Effect of Screening Colonoscopy on Colorectal Cancer Incidence and Mortality. *Clinical Gastroenterology and Hepatology*, 7(7):770–775.

Kilham, P., Hartebrodt, C., and Kändler, G. (2018). Article generating tree-level harvest predictions from forest inventories with random forests. *Forests*, 10(1):20.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, pages 963–974.

Lee, S. C. (2021). Addressing imbalanced insurance data through zero-inflated Poisson regression with boosting. *ASTIN Bulletin: The Journal of the IAA*, 51(1):27–55.

Levin, B., Lieberman, D. A., McFarland, B., Andrews, K. S., Brooks, D., Bond, J., Dash, C., Giardiello, F. M., Glick, S., Johnson, D., et al. (2008). Screening and Surveillance for the Early Detection of Colorectal Cancer and Adenomatous Polyps, 2008: A Joint Guideline From the American Cancer Society, the US Multi-Society Task Force on Colorectal Cancer, and the American College of Radiology. *Gastroenterology*, 134(5):1570–1595.

Lippmann, R. (1987). An Introduction to Computing with Neural Nets. *IEEE ASSP magazine*, 4(2):4–22.

Loeve, F., Boer, R., van Oortmarssen, G. J., van Ballegooijen, M., and Habbema, J. D. F. (1999). The MISCAN-COLON Simulation Model for the Evaluation of Colorectal Cancer Screening. *Computers and Biomedical Research*, 32(1):13–33.

Mandel, F., Ghosh, R. P., and Barnett, I. (2021). Neural networks for clustered and longitudinal data using mixed effects models. *Biometrics: A Journal of the International Biometric Society*.

Mangino, A. A. and Finch, W. H. (2021). Prediction with mixed effects models: A monte carlo simulation study. *Educational and Psychological Measurement*, 81(6):1118–1142.

Manser, C. N., Bachmann, L. M., Brunner, J., Hunold, F., Bauerfeind, P., and Marbet, U. A. (2012). Colonoscopy screening markedly reduces the occurrence of colon carcinomas and carcinoma-related death: a closed cohort study. *Gastrointestinal Endoscopy*, 76(1):110–117.

Molnar, C. (2019). *Interpretable Machine Learning*. Leanpub.

Moore, A. and Bell, M. (2022). XGBoost, a novel explainable AI technique, in the prediction of myocardial infarction, a UK Biobank cohort study. *medRxiv preprint*.

Morson, B. (1974). The polyp-cancer sequence in the large bowel. *Journal of the Royal Society of Medicine*, 67:451–457.

Mousavinezhad, M., Majdzadeh, R., Sari, A. A., Delavari, A., and Mohtasham, F. (2016). The effectiveness of FOBT vs. FIT: A meta-analysis on colorectal cancer screening test. *Medical Journal of the Islamic Republic of Iran*, 30:366.

Ngufor, C., van Houten, H., Caffo, B. S., Shah, N. D., and McCoy, R. G. (2019). Mixed Effect Machine Learning: A framework for predicting longitudinal change in hemoglobin A1c. *Journal of Biomedical Informatics*, 89:56–67.

Nishihara, R., Wu, K., Lochhead, P., Morikawa, T., Liao, X., Qian, Z. R., Inamura, K., Kim, S. A., Kuchiba, A., Yamauchi, M., et al. (2013). Long-Term Colorectal-Cancer Incidence and Mortality after Lower Endoscopy. *New England Journal of Medicine*, 369(12):1095–1105.

Panchavati, S., Zelin, N. S., Garikipati, A., Pellegrini, E., Iqbal, Z., Barnes, G., Hoffman, J., Calvert, J., Mao, Q., and Das, R. (2022). A comparative analysis of machine learning approaches to predict C. difficile infection in hospitalized patients. *American Journal of Infection Control*, 50(3):250–257.

Pashayan, N., Duffy, S. W., Chowdhury, S., Dent, T., Burton, H., Neal, D. E., Easton, D. F., Eeles, R., and Pharoah, P. (2011). Polygenic susceptibility to prostate and breast cancer: implications for personalised screening. *British Journal of Cancer*, 104(10):1656–1663.

Piggott, M., Pearson, S., Seaman, H., and Halloran, S. (2014). Evaluation of quantitative faecal immunochemical tests for haemoglobin.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767.

Putatunda, S. and Rama, K. (2018). A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost. In *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*, pages 6–10.

Raghunathan, T. E., Solenberger, P. W., and Van Hoewyk, J. (2002). IVEware: Imputation and Variance Estimation Software. *University of Michigan*.

Regula, J., Rupinski, M., Kraszewska, E., Polkowski, M., Pachlewski, J., Orlowska, J., Nowacki, M. P., and Butruk, E. (2006). Colonoscopy in Colorectal-Cancer Screening for Detection of Advanced Neoplasia. *New England Journal of Medicine*, 355(18):1863–1872.

Rekabdar, B., Albright, D. L., McDaniel, J. T., Talafha, S., and Jeong, H. (2022). From machine learning to deep learning: A comprehensive study of alcohol and drug use disorder. *Healthcare Analytics*, 2:100104.

Ryu, S.-E., Shin, D.-H., and Chung, K. (2020). Prediction Model of Dementia Risk Based on XGBoost Using Derived Variable Extraction and Hyper Parameter Optimization. *IEEE Access*, 8:177708–177720.

Sakthivel, K. and Rajitha, C. (2017). A Comparative Study of Zero-inflated, Hurdle Models with Artificial Neural Network in Claim Count Modeling. *International Journal of Statistics and Systems*, 12(2):265–276.

Schröder, F. H., Hugosson, J., Roobol, M. J., Tammela, T. L., Ciatto, S., Nelen, V., Kwiatkowski, M., Lujan, M., Lilja, H., Zappa, M., et al. (2009). Screening and Prostate-Cancer Mortality in a Randomized European Study. *New England Journal of Medicine*, 360(13):1320–1328.

Segal, M. R. (1992). Tree-Structured Methods for Longitudinal Data. *Journal of the American Statistical Association*, 87(418):407–418.

Sela, R. J. and Simonoff, J. S. (2011). RE-EM Trees: A New Data Mining Approach for Longitudinal Data. *Machine learning*, 86(2):169–207.

Sheen, E. M. (2019). An Exploration of Mixed Effects Models for Analysis of Infant Weight Gain Trajectories.

Sigrist, F. (2020). Gaussian Process Boosting. *arXiv preprint arXiv:2004.02653*.

Śmieja, M., Struski, Ł., Tabor, J., Zieliński, B., and Spurek, P. (2018). Processing of missing data by neural networks. *Advances in Neural Information Processing Systems*, 31.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

van der Steeg, A., Keyzer-Dekker, C., De Vries, J., and Roukema, J. (2011). Effect of abnormal screening mammogram on quality of life. *Journal of British Surgery*, 98(4):537–542.

Strum, W. B. (2016). Colorectal Adenomas. *New England Journal of Medicine*, 374(11):1065–1075.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., and Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, 71(3):209–249.

Tandon, R., Adak, S., and Kaye, J. A. (2006). Neural networks for longitudinal studies in Alzheimer's disease. *Artificial Intelligence in Medicine*, 36(3):245–255.

Thanikachalam, K. and Khan, G. (2019). Colorectal Cancer and Nutrition. *Nutrients*, 11(1):164.

Thrumurthy, S. G., Thrumurthy, S. S., Gilbert, C. E., Ross, P., and Haji, A. (2016). Colorectal adenocarcinoma: risks, prevention and diagnosis. *British Medical Journal*, 354.

Toribara, N. W. and Sleisenger, M. H. (1995). Screening for Colorectal Cancer. *New England Journal of Medicine*, 332(13):861–867.

Torre, L. A., Bray, F., Siegel, R. L., Ferlay, J., Lortet-Tieulent, J., and Jemal, A. (2015). Global Cancer Statistics, 2012. *CA: A Cancer Journal for Clinicians*, 65(2):87–108.

Tutz, G. and Groll, A. (2010). Generalized Linear Mixed Models Based on Boosting. In *Statistical Modelling and Regression Structures*, pages 197–215. Springer.

Van Buuren, S. (2018). *Flexible Imputation of Missing Data*. CRC press.

van Duuren, L. A., Ozik, J., Spliet, R., Collier, N. T., Lansdorp-Vogelaar, I., and Meester, R. G. (2022). An Evolutionary Algorithm to Personalize Stool-Based Colorectal Cancer Screening. *Frontiers in Physiology*, page 2515.

Wardle, J., Williamson, S., Sutton, S., Biran, A., McCaffery, K., Cuzick, J., and Atkin, W. (2003). Psychological Impact of Colorectal Cancer Screening. *Health Psychology*, 22(1):54.

Welch, H. G. and Black, W. C. (2010). Overdiagnosis in cancer. *Journal of the National Cancer Institute*, 102(9):605–613.

Whitlock, E. P., Lin, J. S., Liles, E., Beil, T. L., and Fu, R. (2012). Screening for Colorectal Cancer: A Targeted, Updated Systematic Review for the U.S. Preventive Services Task Force. *Annals of Internal Medicine*, 157(2):120–134.

Winawer, S. J. (2007). Colorectal cancer screening. *Best Practice & Research Clinical Gastroenterology*, 21(6):1031–1048.

Wu, Y., Xiang, C., Jia, M., and Fang, Y. (2022). Interpretable classifiers for prediction of disability trajectories using a nationwide longitudinal database. *BioMed Central Geriatrics*, 22(1):1–17.

Xiong, Y., Kim, H. J., and Singh, V. (2019). Mixed Effects Neural Networks (MeNets) With Applications to Gaze Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, Y., Qian, W., and Zou, H. (2018). Insurance Premium Prediction via Gradient Tree-Boosted Tweedie Compound Poisson Models. *Journal of Business & Economic Statistics*, 36(3):456–470.

Zorzi, M., Fedeli, U., Schievano, E., Bovo, E., Guzzinati, S., Baracco, S., Fedato, C., Saugo, M., and Dei Tos, A. P. (2015). Impact on colorectal cancer mortality of screening programmes based on the faecal immunochemical test. *Gut*, 64(5):784–790.

# Appendices

## A Data

### A.1 Data pre-processing

The data cleaning procedure is as follows. We first delete variables which are inane to our analysis (e.g., information on the morphology and topography of a cancer), and variables which possibly contain patient sensitive information (e.g., participation date, patient pseudonym, and invitation date). We then remove individuals with invalid or missing entries, individuals who returned to the data set after a positive FIT, and individuals younger than 55 or older than 77 in the first round of 2014. The final data set only includes individuals who participated in two or more consecutive rounds and those who participated in one round at most.

With respect to data engineering, we first transform `result` to attain three categories: favourable, unfavourable, and missing. Here, 'unfavourable' contains all observations with '*unfavourable*' and '*unfavourable (unreliable)*' as result, and 'favourable' contains only observations with '*favourable*' as result. The remaining observations are cast to 'missing', and are deleted from the data set. Hereafter, given that the results of the FIT are based on two thresholds: 275 ng/ml and 47 $\mu$g/g, we multiply observations where `haemoglobin current` is based on 275 as threshold by $\frac{47}{275}$, such that all haemoglobin values are represented in the same unit. Finally, we create the following variables: `haemoglobin previous`, `haemoglobin max`, `haemoglobin min`, and we perform one-hot-encoding to `FIT number` and `stage`. More detailed descriptions of each of the variables in the final data set are shown in Table 1.

### A.2 MICE

This research employs Multiple Imputation via Chained Equations (MICE) to impute missing values in the stage variable of the original data set. To run this algorithm we create a data set consisting of two data sets from the Dutch screening program and a simulated population run in MISCAN-Colon. Table 4 shows a subset of the combined data. Note that the the '15 threshold' data set contains information on all variables at all times, while the simulated population never contains information on `haemoglobin current` and the original data set only contains information on `stage` 3.8% of the time. Table 5 shows descriptive statistics for each of the additional data sets. The MICE iterations are as follows:

1 First replace all missing values with placeholders. In this case, all missing values are replaced by a random draw of data (with replacement) within each respective variable.

2.1 Remove the placeholder of `haemoglobin current`.

2.2 Use random sampling to impute all missing values in `haemoglobin current`.

3.1 Remove the placeholder of `stage`.

3.2 Using the newly imputed `haemoglobin current` in combination with `result`, `age` and `sex`, perform predictive mean matching to impute `stage`.

Once these steps are complete, we have completed one full cycle of MICE. We perform 5 × 10 cycles, after which we are left with five distinct imputed `stage` variables. We then compare the distribution of stages in each of these imputed variables to the distribution of stages in the 'MISCAN simulation' data set, and select the imputed variable which most closely matches the distribution in the MISCAN `stage` to replace the original `stage`. Finally, we drop the '15 threshold' and 'MISCAN simulation' data sets.

Table 4: Hypothetical example of one full cycle of the Multiple Imputation via Chained Equations algorithm.

**Step 0**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | NA |
| 471 | Negative | 70 | Female | 20.0 | NA |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | NA | 1 |
| MI1 | Negative | 58 | Male | NA | 2 |

**Step 1**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | 2 |
| 471 | Negative | 70 | Female | 20.0 | 2 |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | 37.3 | 1 |
| MI1 | Negative | 58 | Male | 47.7 | 2 |

**Step 2.1**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | 2 |
| 471 | Negative | 70 | Female | 20.0 | 2 |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | ? | 1 |
| MI1 | Negative | 58 | Male | ? | 2 |

**Step 2.2**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | 2 |
| 471 | Negative | 70 | Female | 20.0 | 2 |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | 20.8 | 1 |
| MI1 | Negative | 58 | Male | 42.6 | 2 |

**Step 3.1**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | ? |
| 471 | Negative | 70 | Female | 20.0 | ? |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | 20.8 | 1 |
| MI1 | Negative | 58 | Male | 42.6 | 2 |

**Step 3.2**

| ID | Result | Age | Sex | Hb | Stage |
|---|---|---|---|---|---|
| 471 | Negative | 68 | Female | 0 | 1 |
| 471 | Negative | 70 | Female | 20.0 | 1 |
| 471 | Positive | 72 | Female | 307.1 | 4 |
| ⋮ | | | | | |
| 151 | Negative | 73 | Male | 37.3 | 1 |
| 152 | Positive | 73 | Female | 47.7 | 2 |
| ⋮ | | | | | |
| MI1 | Negative | 65 | Male | 20.8 | 1 |
| MI1 | Negative | 58 | Male | 42.6 | 2 |

*Notes:* This table represents an exemplified version of one full cycle of Multiple Imputation via Chained Equations. The data set consists of individuals from the original, the '15 threshold' and the 'MISCAN simulation' data set, denoted by 47*, 15* and MI* as `ID` preface, respectively. The red numbers in Step 1 are obtained from a random draw with replacement from the full data set. The red numbers in Step 2.2 and 3.2 are obtained through predictive mean matching using all variables except the one that will be imputed (i.e., excluding the variable with a question mark in Step 2.1 and 3.1, respectively). `Hb` represents `haemoglobin current`. For more information on each variable see Table 1. The numbers in this table are for illustrative purposes only.

Table 5: Descriptive statistics of additional data sets required for performing Multiple Imputation via Chained Equations.

| **Variable** | **Data set** | |
|---|---|---|
| | MISCAN simulation | 15 threshold |
| `Age` | $[55; 77]$ | $[56; 76]$ |
| `Haemoglobin current` | – | $[0; 292.8]^*$ |
| `Haemoglobin threshold` | – | $[15, 45, 88, 275]$ |
| `Sex` | 0 (Male, 48%),<br>1 (Female, 52%) | 0 (Male, 58.2%),<br>1 (Female, 47.8%) |
| `Stage` | 1 (Healthy, 84.3%),<br>2 (Non-advanced adenoma, 8.3%),<br>3 (Advanced adenoma, 7.0%),<br>4 (Colorectal cancer, 0.4%) | 1 (Healthy, 20.4%),<br>2 (Non-advanced adenoma, 28.9%),<br>3 (Advanced adenoma, 42.8%),<br>4 (Colorectal cancer, 7.9%) |

*Notes:* *The '15 threshold' data set contains five observations with `haemoglobin current` below the (lowest) threshold of 15 $\mu$g/g, which were not deleted since `stage` was known.

## A.3 Descriptives for full, training, and test set

Table 6: Number of true observations in absolute values and percentages per interval in the full data set, train set, and test set.

| **Interval** | **Full data set** | | **Train set** | | **Test set** | |
|---|---|---|---|---|---|---|
| | # Observations | Percentage | # Observations | Percentage | # Observations | Percentage |
| $[0; 25)$ | 6422843 | 94.50% | 4566137 | 94.45% | 1856706 | 94.63% |
| $[25; 50)$ | 120692 | 1.78% | 86363 | 1.79% | 34329 | 1.75% |
| $[50; 75)$ | 61099 | 0.90% | 43947 | 0.91% | 17152 | 0.87% |
| $[75; 100)$ | 34928 | 0.51% | 24934 | 0.52% | 9994 | 0.51% |
| $[100; 125)$ | 23593 | 0.35% | 16763 | 0.35% | 6830 | 0.35% |
| $[125; 150)$ | 19010 | 0.28% | 13653 | 0.28% | 5357 | 0.27% |
| $[150; 175)$ | 18873 | 0.28% | 13494 | 0.28% | 5379 | 0.27% |
| $[175; 200)$ | 22904 | 0.34% | 16641 | 0.34% | 6263 | 0.32% |
| $[200; 225)$ | 26983 | 0.40% | 19645 | 0.41% | 7338 | 0.37% |
| $[225; 250)$ | 26150 | 0.38% | 19176 | 0.40% | 6974 | 0.36% |
| $[250; 275)$ | 16611 | 0.24% | 11833 | 0.24% | 4778 | 0.24% |
| $[275; 300)$ | 3010 | 0.04% | 2042 | 0.04% | 968 | 0.05% |
| $[300; 325)$ | 33 | 0.00% | 19 | 0.00% | 14 | 0.00% |
| $[325; 250)$ | 0 | 0% | 0 | 0% | 0 | 0% |
| $[350; 375)$ | 0 | 0% | 0 | 0% | 0 | 0% |
| $[375; 400)$ | 1 | 0.00% | 1 | 0.00% | 0 | 0% |
| $[400; 425)$ | 0 | 0% | 0 | 0% | 0 | 0% |
| $[425; 450)$ | 1 | 0.00% | 1 | 0.00% | 0 | 0% |

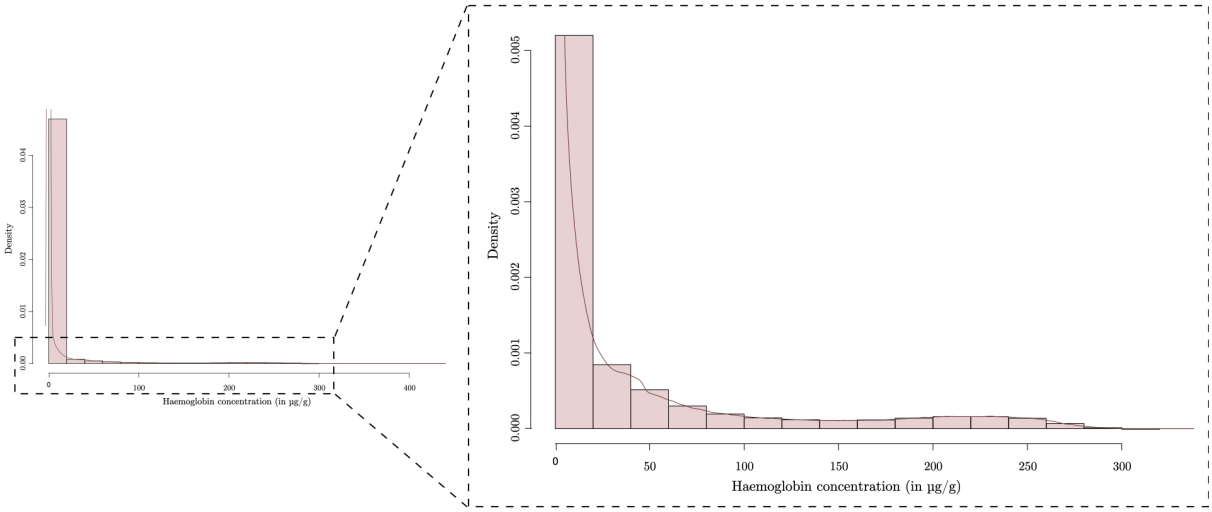



Figure 11: Zoomed in rendition of the density and histogram of haemoglobin concentrations in the CRC data set shown in Figure 4a.

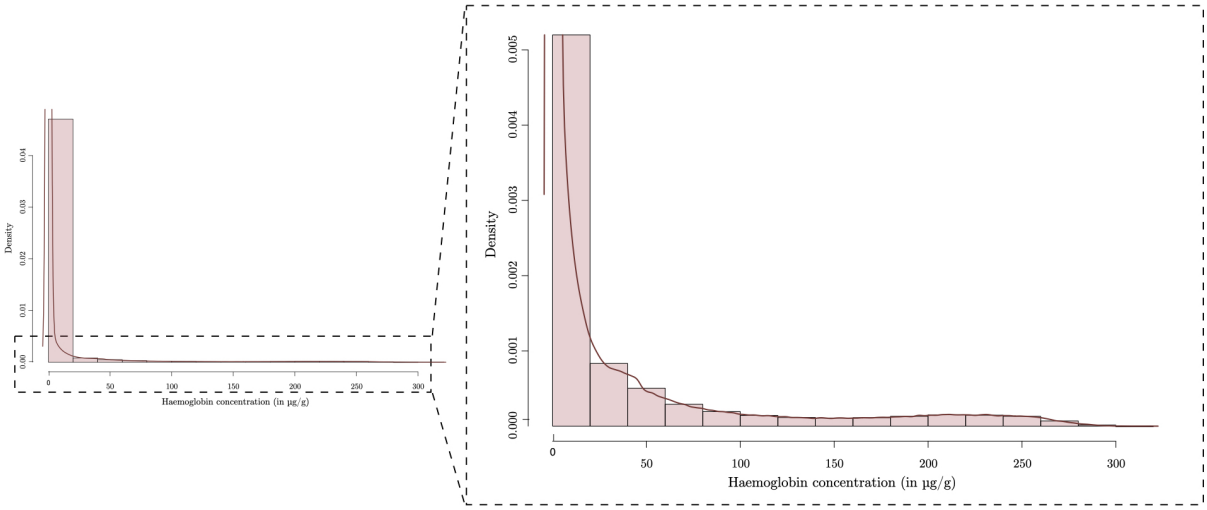



Figure 12: Zoomed in rendition of the density and histogram of haemoglobin concentrations in the test set shown in Figure 6a.

# B MERF, (G)MERT, RE-EM, and MEml

We have discussed six (of the most) influential papers in the field of machine learning in longitudinal data. The pioneer in this field is Segal (1992), whose goal is to extend tree based methods to clustered data. The approach is quite straightforward; the split function is modified for each node to accommodate multiple responses. Though this method is a step in the right direction, it has its shortcomings. First, it only allows for equal observations within individuals, which is not always representative for the data (at least not in our case). Second, it does not allow for splits on observation-level covariates, i.e., all observations within a cluster/individual end up in the same terminal node, as they remain together during the tree building process. As a result, it is not possible to exploit time-varying values of attributes observed after the first period to predict observations within clusters/individuals.[34] Third, the estimated response value at each node is calculated as a vector of means per node. Thus, this method cannot be used to predict periods ahead, as calculating the mean requires observations for this period.

Acknowledging these shortcomings, many new and improved methods have been proposed including Sela and Simonoff's (2011) random-effects expectation-maximization (RE-EM) trees, Hajjem et al.'s (2011) mixed-effects regression trees (MERT), Hajjem et al.'s (2014) mixed-effects regression forests (MERF), Hajjem et al.'s (2017) generalised mixed-effects regression trees (GMERT), and lastly Ngufor et al.'s (2019) mixed-effects machine learning (MEml) models. In what follows, we provide extensive explanations and mathematical formulations of all five of these approaches.

## B.1 RE-EM trees

Sela and Simonoff (2011) propose their tree-based RE-EM method with a similar goal in mind as Segal (1992): extending tree based algorithms to longitudinal data, but this time through incorporating object-specific random effects. They propose to estimate the fixed- and random-effects using an algorithm reminiscent to Laird and Ware's (1982) expectation-maximisation algorithm.

To explain this method, we first introduce some notation adopted from Hajjem et al. (2017). Define $y_i = [y_{i1}, \ldots, y_{in_i}]^\top$ as the $n_i \times 1$ vector of responses for the $n_i$ observations in cluster $i = 1, \ldots, n$. Let $X_i = [x_{i1}, \ldots, x_{in_i}]^\top$ denote the $n_i \times p$ matrix of fixed-effects covariates, and let $Z_i = [z_{i1}, \ldots, z_{in_i}]^\top$ denote the $n_i \times q$ matrix of random-effects covariates. The $q \times 1$ (unknown) vector of random effects for

[34] Consider for example a node splitting on whether it is the first round or a later round, the latter splits again into second round or higher, and a final split is made for the distinction of the third or fourth round. In this case, we can use the current round value in prediction, exploiting the time-varying values of the covariates. In the Segal (1992) method, however, a single set of attributes is used for all observations within an individual, such that it would not possible to split on, e.g., current round. Hence, it is near impossible to exploit the time-varying information beyond the first period. The only possible way would be to use *all* observations of a time-varying covariate in prediction. However, this most likely will not be useful in practice as including such observations can result in using future information to predict past observations.

cluster $i$ are denoted by $b_i$.[35] The proposed model follows the functional form:

$$\begin{aligned} y_i &= f(X_i) + Z_i b_i + \varepsilon_i \\ b_i &\sim N(0, D), \varepsilon_i \sim N(0, R_i) \\ i &= 1, \ldots, n, \end{aligned} \tag{5}$$

where $D$ denotes the covariance matrix of the random effects $b_i$, and $R_i$ denotes the (assumed diagonal) covariance matrix of the error terms $\varepsilon_i$.

Any tree-based algorithm recursively partitions the feature space into disjoint regions such that observations with similar response values $y_i$ are grouped together. Adopting (part of) the notation by Ngufor et al. (2019), let $\mathbf{R}_v$ denote the collection of these disjoint regions $v = 1, \ldots, V$. The unknown functional relationship between the response and the predictors can be written as

$$f(x) \equiv \sum_{v=1}^{V} c_v \cdot \mathbf{I}(X_i \in \mathbf{R}_v), \tag{6}$$

where $c_v$ is the constant term for the $v$'th region and second term is the indicator function mapping $X_i$ to regions in $\mathbf{R}_v$.

**Algorithm 1:** RE-EM

**Data:** Longitudinal or clustered data: $\{(x_{ij}, y_{ij}), i = 1, \ldots, n, j = 1, \ldots, n_i\}$
**Result:** Estimated machine learning model $\hat{f}$ and random-effects $\hat{b}_i$
Set $r = 0$. Let $\hat{b}_i = 0$;
**while** *Change in (restricted) likelihood function* $> \epsilon$ **do**
  $r \longleftarrow r + 1$
  **E-step:**
  (i) $y^*_{i(r)} = y_i - Z_i \hat{b}_{i(r-1)}, i = 1, \ldots, n$;
  (ii) Let $\hat{f}(X_i)_{(r)}$ an estimate of $f(X_i)$ obtained from a standard tree algorithm with $y^*_{i(r)}$ as responses and $X_i, i = 1, \ldots, n$, as covariates;
  (iii) Use this regression tree to create a set of indicator variables, $\mathbf{I}(X_i \in \mathbf{R}_v)_{(r)}$, where $v$ ranges over all of the terminal nodes in the tree;

  **M-step:**
  (i) Fit the linear mixed effects model, $y_i = Z_i b_i + \mathbf{I}(X_i \in \mathbf{R}_v) c_v + \varepsilon_{it}$. Extract $\hat{b}_{i(r)}$ from the estimated model.
**end**

Using this notation, we present the RE-EM method in Algorithm 1. To estimate the population-level fixed effects, $\hat{f}$, Sela and Simonoff (2011) fit a regression tree (although any tree-based algorithm can be used) using adjusted response variables from which the estimated random effects, $Zi\hat{b}_i$, have been removed. Based on this fitted regression tree, they create a set of terminal nodes, which are then used to fit a linear mixed effects (LME) model to estimate the random effects using (restricted) maximum likelihood. The RE-EM model, in contrast Segal's (1992) model, allows for each node to be split on any

[35] Clearly, this notation is easily extended to longitudinal data, if we define each individual as it's own group, such $j = 1, \ldots, n_i$ represent the observations for each individual $i = 1, \ldots, n$.

covariate, such that different observations for the same object may be placed in different nodes. They also allow for unbalanced panels.

## B.2 MERT

Hajjem et al. (2011) propose another method for longitudinal data using tree-based methods. Their MERT model aims to dissociate the fixed effects from random effects, and follows the same functional form as presented in Equation 5. Similar to Sela and Simonoff (2011), MERT also allows for modeling unbalanced clusters and splitting on observation-level covariates. Moreover, MERF also estimates the fixed- and random effects in an expectation-maximization manner. Specifically, Algorithm 2 shows that MERT uses a standard regression tree to model the fixed-effects, and a LME with (restricted) maximum likelihood to estimate the random-effects. This process repeats itself until the generalised log-likelihood (GLL) is smaller than a predetermined tolerance value $\epsilon$.

**Algorithm 2:** MERT

**Data:** Longitudinal or clustered data: $\{(x_{ij}, y_{ij}), i = 1, \ldots, n, j = 1, \ldots, n_i\}$
**Result:** Estimated machine learning model $\hat{f}$ and random-effects $\hat{b}_i$
Set $r = 0$. Let $\hat{b}_{i(0)} = 0, \hat{\sigma}^2_{(0)} = 1$, and $\hat{D}_{(0)} = I_q$.
**while** *Change in GLL* $> \varepsilon$ **do**
$r \longleftarrow r + 1$
**E-step:**
(i) $y^*_{i(r)} = y_i - Z_i \hat{b}_{i(r-1)}, i = 1, \ldots, n$;
(ii) Let $\hat{f}(X_i)_{(r)}$ an estimate of $f(X_i)$ obtained from a standard tree algorithm with $y^*_{i(r)}$ as responses and $X_i, i = 1, \ldots, n$, as covariates;
(iii) $\hat{b}_{i(r)} = \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} \left( y_i - \hat{f}(X_i)_{(r)} \right), i = 1, \ldots, n$, where $\hat{V}_{i(r-1)} = Z_i \hat{D}_{(r-1)} Z_i^T + \hat{\sigma}^2_{(r-1)} I_{n_i}, i = 1, \ldots, n$;

**M-step:**
(i) $\hat{\sigma}^2_{(r)} = N^{-1} \sum_{i=1}^n \left\{ \hat{\varepsilon}^T_{i(r)} \hat{\varepsilon}_{i(r)} + \hat{\sigma}^2_{(r-1)} \left[ n_i - \hat{\sigma}^2_{(r-1)} \text{trace} \left( \hat{V}_{i(r-1)} \right) \right] \right\}$, where $\hat{\varepsilon}_{i(r)} = y_i - \hat{f}(X_i)_{(r)} - Z_i \hat{b}_{i(r)}$;
(ii) $\hat{D}_{(r)} = n^{-1} \sum_{i=1}^n \left\{ \hat{b}_{i(r)} \hat{b}^T_{i(r)} + \left[ \hat{D}_{(r-1)} - \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} Z_i \hat{D}_{(r-1)} \right] \right\}$.
**end**

We distinguish two (main) differences between MERT and RE-EM trees. First, MERT assumes that all correlation is induced solely via between-subject variation, such that $R_i$ is assumed to be diagonal, whereas RE-EM allows for more general correlation structures. Second, the modeling of random-effects in MERT is node-invariant, whereas RE-EM trees obviously are not.

## B.3 MERF

Further improvement of the predictive accuracy of MERT could be achieved when used as the base learner in an ensemble algorithms. Consequently, Hajjem et al. (2014) introduce MERF, which generalizes MERT through replacing the regression trees within each iteration in MERT with a forest of regression trees. MERF follows the same functional form as MERT and RE-EM (see Equation 5) and also assumes $R_i$ is diagonal.

The approach is detailed in Algorithm 3. Each step in MERF which is identical to that in MERT is presented in gray, to highlight the similarities between both methods. Clearly, they only differ in step two of the expectation-step in which the regression forest is built. Besides the premise of the algorithm and most of the algorithm itself being similar in MERT and MERF, the difference in step two comes with an additional assumption. One must resample observations in order to build random forests, which is done through bootstrapping in this case. For optimal performance of bootstrapping, it is required that the observations are i.i.d.. Consequently, we must assume that the random effects $Z_i b_i$ fully explain the correlation within clusters/individuals, such that $y_i^*$ is i.i.d. once the random effects have been removed.

**Algorithm 3:** MERF

**Data:** Longitudinal or clustered data: $\{(x_{ij}, y_{ij}), i = 1, \ldots, n, j = 1, \ldots, n_i\}$

**Result:** Estimated machine learning model $\hat{f}$ and random-effects $\hat{b}_i$

Set $r = 0$. Let $\hat{b}_{i(0)} = 0, \hat{\sigma}^2_{(0)} = 1$, and $\hat{D}_{(0)} = I_q$.;

**while** *Change in GLL* $> \varepsilon$ **do**

$r \longleftarrow r + 1$

**E-step:**

(i) $y^*_{i(r)} = y_i - Z_i \hat{b}_{i(r-1)}, i = 1, \ldots, n$;

(ii.a) Build a forest of trees using a standard RF algorithm with $y^*_{ij(r)}$ as the training set responses and $x_{ij}$ as the corresponding training set of covariates, $i = 1, ..., n, j = 1, ..., n_i$. The bootstrap training samples to build the forest are simple random samples drawn with replacement from the training set $y^*_{ij(r)}, x_{ij}$;

(ii.b) Estimate $\hat{f}(x_{ij})_{(r)}$ using only the subset of trees in the forest that are built with the bootstrap samples not containing $y^*_{ij}$, that is, the out-of-bag prediction of the RF;

(ii.c) Let $\hat{f}(X_i)_{(r)} = \left[\hat{f}(x_{i1})_{(r)}, ..., \hat{f}(x_{in_i})_{(r)}\right]$;

(iii) $\hat{b}_{i(r)} = \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} \left(y_i - \hat{f}\left(X_i\right)_{(r)}\right), i = 1, \ldots, n$, where $\hat{V}_{i(r-1)} = Z_i \hat{D}_{(r-1)} Z_i^T + \hat{\sigma}^2_{(r-1)} I_{n_i}, i = 1, \ldots, n$;

**M-step:**

(i) $\hat{\sigma}^2_{(r)} = N^{-1} \sum_{i=1}^n \left\{\hat{\varepsilon}^T_{i(r)} \hat{\varepsilon}_{i(r)} + \hat{\sigma}^2_{(r-1)} \left[n_i - \hat{\sigma}^2_{(r-1)} \text{trace}\left(\hat{V}_{i(r-1)}\right)\right]\right\}$, where $\hat{\varepsilon}_{i(r)} = y_i - \hat{f}\left(X_i\right)_{(r)} - Z_i \hat{b}_{i(r)}$;

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

**end**

## B.4 GMERT and MEml

RE-EM, MERT and MERF are all designed for Gaussian response data. In practice, it can be useful to also model non-Gaussian (e.g., binary) response data. To this end, we introduce the GMERT model by Hajjem et al. (2017) and the MEml model by Ngufor et al. (2019). Both approaches are based on GLMM, and allow for non-Gaussian dependent variables.

Adopting the notation by Hajjem et al. (2017) the GLMM assumes that the response vector $y_i$, conditional on the (assumed i.i.d. normal) random effects $b_i$, is independent and follows a distribution from the exponential family with density

$$f\left(y_i \mid b_i, \beta\right),$$

where the fixed-effects parameter $\beta_{[p\times 1]}$ is an unknown common vector over all clusters. Now, define

$$\eta_i = g(\mu_i) = g\left(E\left(y_i \mid b_i\right)\right),$$
$$\text{Cov}\left(y_i \mid b_i\right) = \sigma^2 v_i\left(\mu_i\right),$$

where $\eta_i = g\left(\mu_i\right)_{[n_i\times 1]}$ denotes the population fixed-effect parameters with known link function $g(\cdot)$, possibly known $\sigma^2$ a dispersion parameter, and known variance function $v(\cdot)$ where $v_i\left(\mu_i\right)$ a $n \times n_i$ diagonal matrix with $v(\mu_{ij})$ as elements. The GLMM assumes a parametric distribution and imposes restrictive linear relationships between the link function and the covariates.

The proposed GMERT and MEml models can be written as

$$\begin{aligned} \eta_i &= f\left(X_i\right) + Z_i b_i, \\ b_i &\sim N(0, D), \\ i &= 1, \ldots, n. \end{aligned} \tag{7}$$

Then, following the PQL approach, the data is approximated by $\tilde{y}_i = \mu_i + e_i$ and then taking the first order Taylor expansion about the current parameter estimates, which results in

$$\tilde{y}_i = g\left(\hat{\mu}_i\right) + \left(y_i - \hat{\mu}_i\right) g'\left(\hat{\mu}_i\right), \tag{8}$$

which can be simultaneously defined as

$$\tilde{y}_i = f\left(X_i\right) + Z_i b_i + e_i. \tag{9}$$

### B.4.1 GMERT

Hajjem et al. (2017)'s GMERT model extends their aforementioned MERT model through replacing the linear structure normally used to model fixed-effect component in GLMMs with a regression tree structure. The estimation of the random component is still represented using a linear structure, as in GLMMs. Algorithm 3 presents the GMERT pseudocode, again with gray parts indicating identical steps to Hajjem et al.'s (Hajjem et al. (2011)) MERT model (see Algorithm 2).

The aim of the transformation of $\tilde{y}_i$ in Equation 8 is to make the outcome behave like a normally distributed variable, for which a LME model can be fitted. Or, in the GMERT case, to fit the MERT approach, which is designed for a normally distributed outcome. Clearly, the inner while loop in Algorithm 4 almost perfectly coincides with the MERT algorithm apart from a weight factor (shown in red) and the definition of the responses (although any tree-based algorithm could be used). Once the inner loop has been completed, each of the variables in the weighted LME pseudo-model are updated until convergence of the estimated population fixed-effect parameter $\hat{\eta}_i$. Algorithm 4 also clearly shows one drawback of GMERT. That is, GMERT is computationally expensive as the algorithm is a doubly iterating process, which requires many trees to be built (Hajjem et al., 2017).

**Algorithm 4:** GMERT

**Data:** Longitudinal or clustered data: $\{(x_{ij}, y_{ij}), i = 1, \ldots, n, j = 1, \ldots, n_i\}$

**Result:** Estimated machine learning model $\hat{f}$ and random-effects $\hat{b}_i$

Set $M = 0$, $m = 0$. Given initial estimates of the mean values, $\hat{\mu}_{ij}^{(0)}, j = 1, \ldots, n_i$, fit a weighted LME pseudo-model using the linearised pseudo responses, $\tilde{y}_i^{(0)} = g\left(\hat{\mu}_i^{(0)}\right) + \left(y_i - \hat{\mu}_i^{(0)}\right) g'\left(\hat{\mu}_i^{(0)}\right)$, and the weights, $W_i^{(0)} = \text{diag}\left(w_{ij}^{(0)}\right)$ where $w_{ij}^{(0)} = \left(v_{ij} g'\left(\hat{\mu}_{ij}^{(0)}\right)^2\right)^{-1}$. Let $\hat{\sigma}^2_{(0)}$ and $\hat{D}_{(0)}$ be the estimates of this weighted LME pseudo-model. **while** *non-convergence of* $\hat{\eta}_i$ **do**

$M \longleftarrow M + 1$

**while** *Change in GLL* $> \epsilon$ **do**

Denote $\tilde{y}_{i(m)} := y_{i(m)}$; // Only to illustrate similarity between MERT

$m \longleftarrow m + 1$

**E-step:**

(i) $y^*_{i(m)} = y_i^{(M)} - Z_i \hat{b}_{i(m-1)}$;

(ii) Let $\hat{f}\left(X_i\right)_{(m)}$ be an estimate of $f\left(X_i\right)$ obtained from a standard regression tree algorithm with $y^*_{i(m)}$ as responses, $X_i$ as covariates, and $W_i$ as weights, $i = 1, \ldots, n$;

(iii) $\hat{b}_{i(m)} = \hat{D}_{(m-1)} \left(W_i^{\frac{1}{2}(M)} Z_i\right)^T \hat{V}^{-1}_{i(m-1)} \left(W_i^{\frac{1}{2}(M)} y_i^{(M)} - W_i^{\frac{1}{2}(M)} \hat{f}_{(m)}\left(X_i\right)\right)$, where

$\hat{V}_{i(m-1)} = W_i^{\frac{1}{2}(M)} Z_i \hat{D}_{(m-1)} \left(W_i^{\frac{1}{2}(M)} Z_i\right)^T + \hat{\sigma}^2_{(m-1)} I_{n_i}, i = 1, \ldots, n.$;

**M-step:**

(i) $\hat{\sigma}^2_{(m)} = N^{-1} \sum_{i=1}^n \left\{\hat{\varepsilon}^T_{i(m)} \hat{\varepsilon}_{i(m)} + \hat{\sigma}^2_{(m-1)} \left[n_i - \hat{\sigma}^2_{(m-1)} \text{trace}\left(\hat{V}_{i(m-1)}\right)\right]\right\}$, where

$\hat{\varepsilon}_{i(m)} = W_i^{\frac{1}{2}(M)} y_i^{(M)} - W_i^{\frac{1}{2}(M)} \hat{f}_{(m)}\left(X_i\right) - W_i^{\frac{1}{2}(M)} Z_i \hat{b}_{i(m)}$;

(ii) $\hat{D}_{(m)} =$

$n^{-1} \sum_{i=1}^n \left\{\hat{b}_{i(m)} \hat{b}^T_{i(m)} + \left[\hat{D}_{(m-1)} - \hat{D}_{(m-1)} \left(W_i^{\frac{1}{2}(M)} Z_i\right)^T \hat{V}^{-1}_{i(m-1)} W_i^{\frac{1}{2}(M)} Z_i \hat{D}_{(m-1)}\right]\right\}$;

**end**

(i) $\hat{\eta}_i^{(M)} = \hat{f}_{(m)}\left(X_i\right) + Z_i \hat{b}_{i(m)}$;

(ii) $\hat{\mu}_i^{(M)} = g^{-1}\left(\hat{\eta}_i^{(M)}\right)$;

(iii) $\tilde{y}_i^{(M)} = g\left(\hat{\mu}_i^{(M)}\right) + \left(\tilde{y}_i - \hat{\mu}_i^{(M)}\right) g'\left(\hat{\mu}_i^{(M)}\right)$;

(iv) $w_{ij}^{(M)} = \left(v_{ij} g'\left(\hat{\mu}_{ij}^{(M)}\right)^2\right)^{-1}$;

(v) $W_i^{(M)} = \text{diag}\left(w_{ij}^{(M)}\right)$.

**end**

### B.4.2 MEml

The MEml approach by Ngufor et al. (2019) uses a node-based expectation-maximization approach reminiscent to RE-EM, in a general GLMM framework similar to GMERT. The random effects in Equation 7 and the population-level effects in Equation 9 are alternatively estimated, as shown in Algorithm 5.

First, the random effects are initiated at zero, and are then used to compute the adjusted response variable. Subsequently, a machine learning model is trained to estimate $\hat{f}(X_i)$ from Equation 9 using the adjusted response variables. Depending on the employed machine learning algorithm, the algorithm either extracts rules or terminal nodes for all disjoint regions $v$. Finally, the random effects are estimated using $\hat{f}(X_i)$ in the functional form shown in Equation 6. This process repeats until convergence.

**Algorithm 5:** MEml

**Data:** Longitudinal or clustered data: $\{(x_{ij}, y_{ij}), i = 1, \ldots, n, j = 1, \ldots, n_i\}$

**Result:** Estimated machine learning model $\hat{f}$ and random-effects $\hat{b}_i$

Set $r = 0$. Let $\hat{b}_{i(0)} = 0$ and $\hat{\mu}_{i(0)} = 0.5$. **while** *Change in GLL* $> \varepsilon$ **do**

$r \longleftarrow r + 1$

**E-step:**

(i) Compute $\tilde{y}^*_{i(r)} = \left(y_i - \hat{\mu}_{i(r)}\right) g'\left(\hat{\mu}_{i(r)}\right) + g\left(\hat{\mu}_{i(r)}\right)$;

(ii) Let $\hat{f}\left(X_i\right)_{(m)}$ be an estimate of $f\left(X_i\right)$ obtained from a standard RT, GBM, MOB or Ctree algorithm with $\tilde{y}^*_{i(r)} - \hat{\mathbf{b}}_i^\top \mathbf{z}_i$ as responses, $X_i$ as covariates and weights $w_{ij(r)} = \left(v_{ij} g'\left(\hat{\mu}_{ij(r)}\right)^2\right)^{-1}$ for each observation;

**if** *MOB or Ctree* **then**

(iii) Create a set of indicator variables, $\mathbf{I}\left(X_i \in \mathbf{R}_v\right)_{(r)}$, where $v$ ranges over all of the terminal nodes in the fitted tree object;

**else if** *RF or GBM* **then**

(iii) Create a set of indicator variables, $\mathbf{I}\left(X_i \in \mathbf{R}_v\right)_{(r)}$, where $v$ is a rule set using inTrees;

**end**

**M-step:**

(i) Fit the GLMM model for $\eta_i = \sum_{v=1}^{V} \mathbf{I}\left(\mathbf{X}_i \in \mathbf{R}_v\right) c_v + \mathbf{b}_i^\top \mathbf{z}_i$ and extract estimates of the mixed effects $\hat{b}_{i(r)}$ and mean $\hat{\mu}_{i(r)}$.

**end**

## B.5 Prediction

In the previous sections we have shown the similarities and differences between each model. We now briefly discuss one aspect which is (virtually) the same across each of the aforementioned models: prediction.

We distinguish the prediction of two cases:

1. Predicting observations for new clusters/individuals, with no past observations;
2. Predicting future observations for clusters/individuals within the sample.

In the first scenario, the random effects of a cluster is not known. Therefore, each method fixes the estimated random-effects at zero, and only uses the estimated fixed-effects for the prediction. In the second scenario, both the estimated fixed-effects and the estimated random part corresponding to its cluster are used in prediction using the new covariates.

## B.6 Method comparison

### B.6.1 Performance

Mangino and Finch (2021) find that MEgbm (the GBM version of Ngufor et al.'s (2019) MEml model), MERF, and RE-EM attain similar performance to each other, although Capitaine et al. (2021) and Kilham et al. (2018) find that MERF outperforms RE-EM and GLMM in terms of $R^2$, RMSE and estimated bias. Moreover, Hajjem et al. (2014) finds that MERF outperforms MERT. Since GMERT is a rather new methodology, there exists little comparative research on this method compared to methods such as RE-EM, MERT and MERF. The same holds for the comparison of MEml to (G)MERT. Therefore, it is difficult to assess the relative performance of these methods.

### B.6.2 Mathematical properties

Moreover, even though Hajjem et al. (2011, 2017); Sela and Simonoff (2011) and Ngufor et al. (2019) find that their respective mixed-effects models are always better to use than their single-level counterparts (in presence of random effects), an important point of discussion is convergence of these methods. RE-EM, (G)MERT, MERF, and MEml are each based on the premise of expectation-maximization, and Sela and Simonoff (2011) rightfully note that since these methods are not true EM algorithms, the usual properties of the EM algorithm do not necessarily apply. Although a relatively sizeable body of literature exists on the consistency of regression forests/trees and mixed-effects models; unfortunately, there is only little guidance on the consistency of the methods mentioned here in current literature. For example, Capitaine et al. (2021) investigates the consistency properties of MERF, and find that the fitted MERF forest estimations for the response variable and out-of-sample predictions converge when the number of individuals is large enough ($n \to \infty$). However, the convergence of MERF as a whole, since it is based on an iterative EM-algorithm, requires that the inner RF model must be stabilized. This stabilization only occurs for large values of the number of variables randomly drawn before optimizing the split of a node of a tree in the RF, although it remains unclear what "large" entails. It is also unclear how the convergence properties of MERF hold up in case other machine learning methods are used to estimate the fixed-effects component. Moreover, Hajjem et al. (2017) note that convergence of their algorithm might be dependent on, e.g., the structures and magnitudes of fixed- and random effects, but their research lacks concrete inference on consistency of the estimates. Thus, since assessing the mathematical properties of each method goes beyond the scope of this research, we must join Sela and Simonoff (2011) in their suggestion that further research should be conducted on the consistency of $\hat{f}$ and the estimated random effects.

### B.6.3 Model compatibility

A clear disadvantage of RE-EM and (G)MERT, is that these models only allow for tree-based machine learning methods to be used in the estimation of the fixed-effects. MEml allows for the use of any machine learning method, but Ngufor et al. (2019) do not supply any complementary code if one would want to employ any method other than RF, GBM, MOB, or Ctree. In contrast, Hajjem et al. (2014) recently published an updated version of the MERF source-code which has been adjusted in Python to include all types of Python's Sklearn machine learning methods.

In conclusion, based on the (limited) literature on the relative performance of all methods, and compatibility with both tree-based (XGBoost) and non-tree-based (ANNs) machine learning methods, we choose to employ MERF in this analysis.

# C Bayesian Hyperopt

Table 7: Hyperparameters and their search spaces and descriptions for ANN.

| Hyperparameter | Search space | Description |
|---|---|---|
| `learning_rate` | $[\log(0.001), \log(1)]$ (log uniform) | Step size shrinkage used in update to prevent overfitting. After each boosting step, we can directly obtain the weights of new features, and the learning rate shrinks the feature weights to make the boosting process more conservative. |
| `batch_size` | $[11, 15]$ (uniform) | The number of samples that will be propagated through the network, calculated as $2^b$ with $b$ the hyperparameter value. Introducing this parameter is necessary to reduce memory usage and speed up training time, with as disadvantage that the estimate of the gradient might be less accurate. |
| `n_layers` | $[1, 2]$ | Number of hidden layers in the network. |
| `init_dropout` | $[0.0, 1.0]$ (uniform) | Dropout rate in first hidden layer, where dropout implies temporarily removing a node from the network, along with all its incoming and outgoing connections (Srivastava et al., 2014). |
| `mid_dropout` | $[0.0, 1.0]$ (uniform) | Dropout rate in the second hidden layer, if present. |
| `activation` | [ReLu, sigmoid, identity] | Activation function used to progress from layer to layer. The sigmoid and ReLu activation function introduce non-linearity to the neural network. |
| `n_neurons` | $[8, 240]$ (4) | Total number of neurons used in (both) hidden layer(s). |
| `normalization`[1] | [none, minmax, standardization, quantile] | Data normalisation technique, only applied to train data. Minmax transforms the minimum (maximum) value of each regressor to 0 (1) and interpolates remaining values. Standardization removes the mean and scales to unit variance for each feature. The robust quantile normalization transforms the features to follow a uniform or a normal distribution (also robust). |

*Notes:* This table shows which hyperparameters in the ANN model are tuned using Hyperopt, along with their descriptions. The search space is represented within brackets, with the step size in parentheses. [1]Normalization is technically not a hyperparameter, but it is included in this overview for completeness.

Table 8: Hyperparameters and their search spaces and descriptions for XGBoost.

| Hyperparameter | Search space | Description |
|---|---|---|
| `learning_rate` | $[\log(0.001); \log(1)]$ (log uniform) | Step size shrinkage used in update to prevent overfitting. After each boosting step, we can directly obtain the weights of new features, and the learning rate shrinks the feature weights to make the boosting process more conservative. |
| `max_depth` | $[2, 6]$ (1) | Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit. Beware that the XGBoost algorithm aggressively consumes memory when training a deep tree. |
| `colsample_bytree` | $[0.5, 1.0]$ (uniform) | The subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed. |
| `reg_lambda` | $[0.0, 1.0]$ (uniform) | L2 regularization term on weights. Increasing this value will make the model more conservative. |
| `subsample` | $[0.5, 1.0]$ (uniform) | Subsample ratio of the training instances. Setting it to 0.5 means that the XGBoost algorithm would randomly sample half of the training data prior to growing trees, which will prevent overfitting. Subsampling will occur once in every boosting iteration. |
| `n_estimators` | $[50, 200]$ (10) | The number of gradient boosted trees. Equivalent to number of boosting rounds. A higher number of estimators usually learns better, which might cause overfitting in turn. |
| `normalization`[1] | [none, minmax, standardization, quantile] | Data normalisation technique, only applied to train data. Minmax transforms the minimum (maximum) value of each regressor to 0 (1) and interpolates remaining values. Standardization removes the mean and scales to unit variance for each feature. The robust quantile normalization transforms the features to follow a uniform or a normal distribution (also robust). |

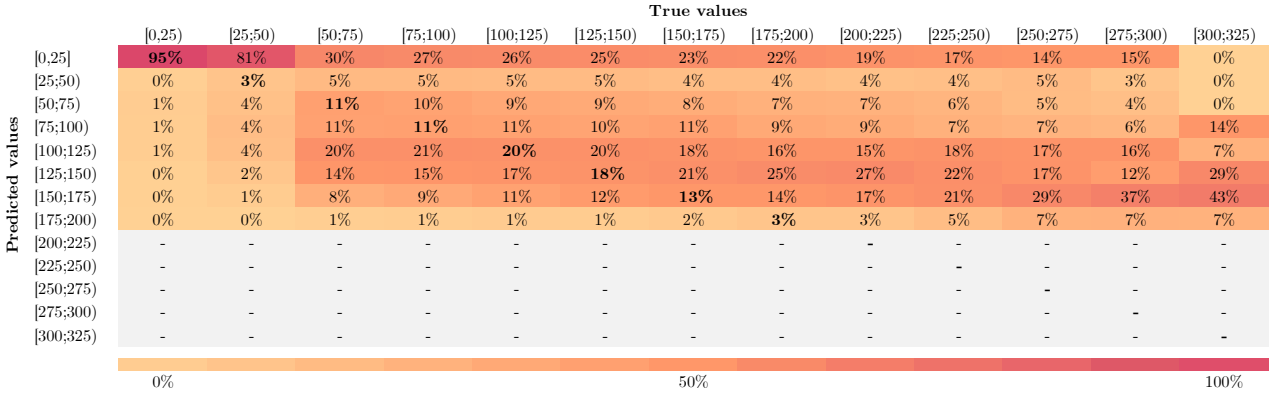
*Notes:* This table shows which hyperparameters in the XGBoost model are tuned using Hyperopt, along with their descriptions. The search space is represented within brackets, with the step size in parentheses.
[1]Normalization is technically not a hyperparameter, but it is included in this overview for completeness.

Table 9: Optimal hyperparameter settings from Bayesian Hyperopt per model.

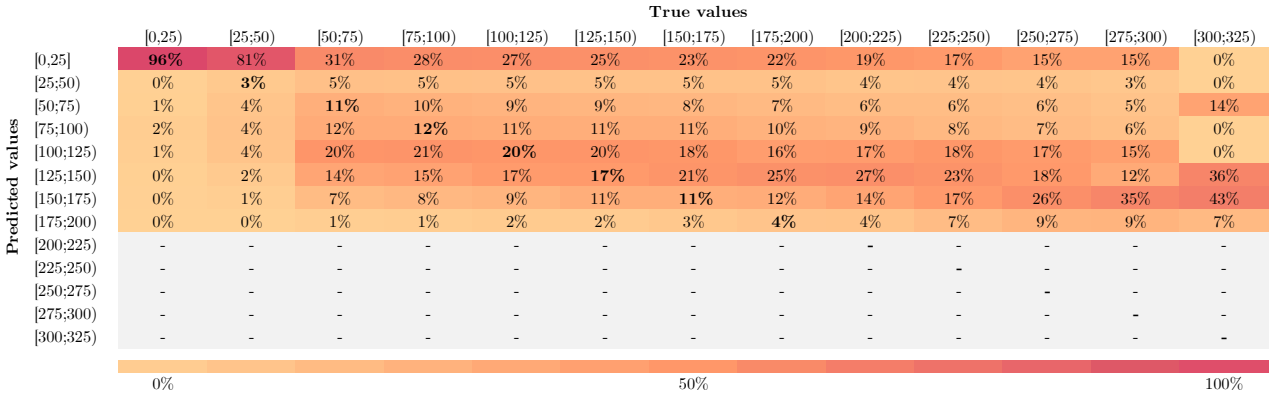
| Hyperparameter | Model | | | | | | |
|---|---|---|---|---|---|---|---|
| | ANN | MeANN* | XGBoost | MeXGB | XGBTweedie | $\text{XGBoost}_{400}$ | $\text{XGBoost}_{350}$ |
| `batch_size` | 13 | 11 | – | – | – | – | – |
| `n_layers` | 2 | 2 | – | – | – | – | – |
| `init_dropout` | 0.0806 | 0.1 | – | – | – | – | – |
| `mid_dropout` | 0.6605 | 0.637 | – | – | – | – | – |
| `activation` | sigmoid | sigmoid | – | – | – | – | – |
| `n_neurons` | 2 | 2 | – | – | – | – | – |
| `learning_rate` | 0.0048 | 0.003 | 0.2783 | 0.5694 | 0.2598 | 0.2791 | 0.0531 |
| `normalization` | stdize[1] | stdize | stdize | none | none | stdize | stdize |
| `max_depth` | – | – | 6 | 6 | 6 | 6 | 9 |
| `colsample_bytree` | – | – | 0.7303 | 0.6270 | 0.9414 | 0.8780 | 0.9796 |
| `reg_lambda` | – | – | 0.4055 | 0.4184 | 0.9260 | 0.7652 | 0.1566 |
| `subsample` | – | – | 0.9852 | 0.9833 | 0.9500 | 0.9698 | 0.9109 |
| `n_estimators` | – | – | 140 | 200 | 180 | 170 | 310 |

*Notes:* This table shows the optimal hyperparameter settings for each of the models evaluated in this paper based on 125 hyperopt iterations, with exception of the MeANN* model, which completed only 77 iterations. The $\text{XGBoost}_{400}$ and $\text{XGBoost}_{350}$ models are XGBoost models with adjusted search space $[50, 400]$ and $[50, 350]$ for `n_estimators`, respectively. For an explanation and search spaces of each hyperparameter, we refer to Tables 7 and 8. [1]Stdize is an abbreviation of standardization.

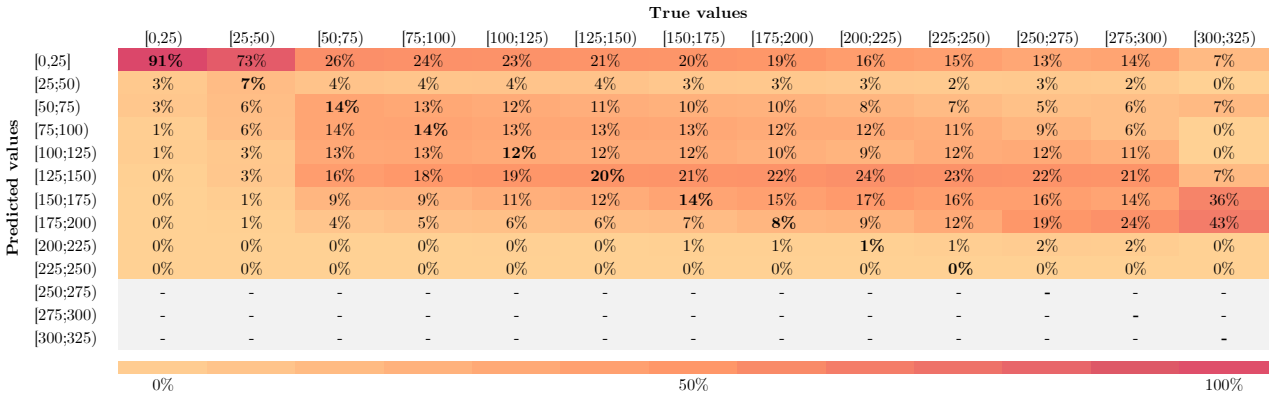
# D Results




(a) ANN

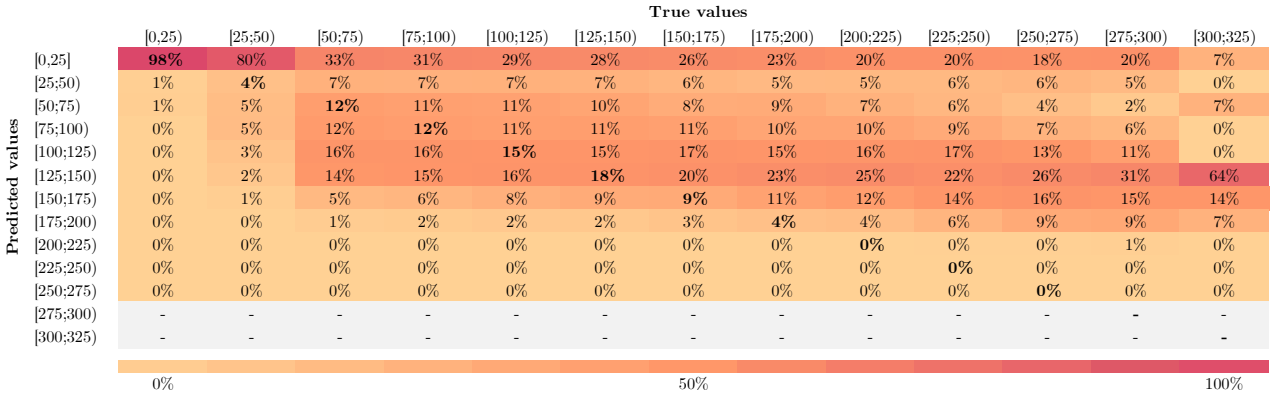



(b) MeANN

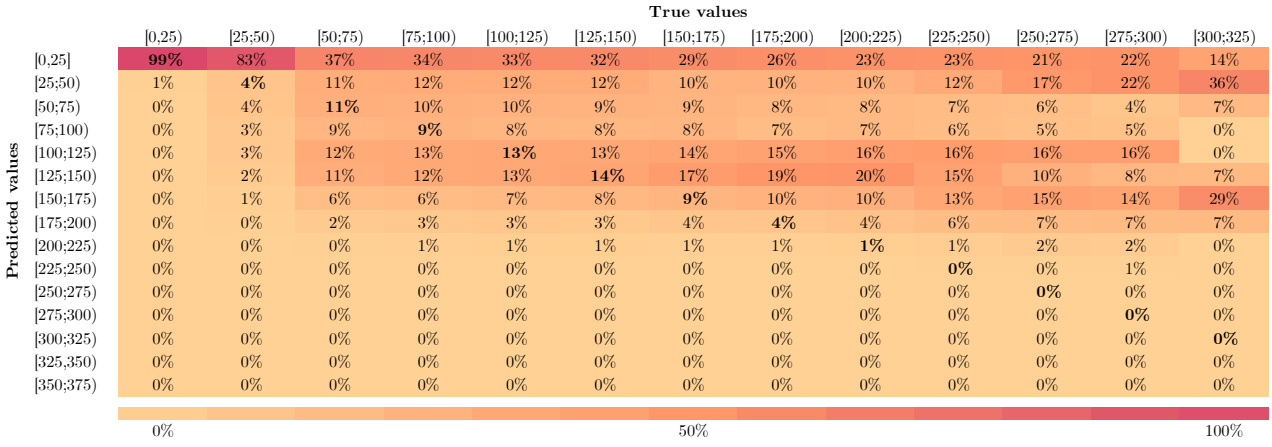



(c) XGBoost

Figure 13: Heatmap of predicted versus true values in percentages, calculated per interval of width 25.




(d) MeXGBoost




(e) TweedieXGBoost

Figure 13 **(continued)**: Heatmap of predicted versus true values in percentages, calculated per interval of width 25.

*Notes:* These figures show the distribution of predicted values versus true values per prediction model in percentages, divided over 13 intervals based on the true values (horizontal axis). That is, each column shows the distribution of the predicted values per interval of the actual values, such that the percentages in each column add up to 100%. In case of perfect prediction, all bold numbers on the diagonal would be 100%. Positive percentages above (below) the diagonal are indicative of under (over) prediction. The gray rows with em dashes (−) are inserted when no predictions lie within the interval specified in that row.

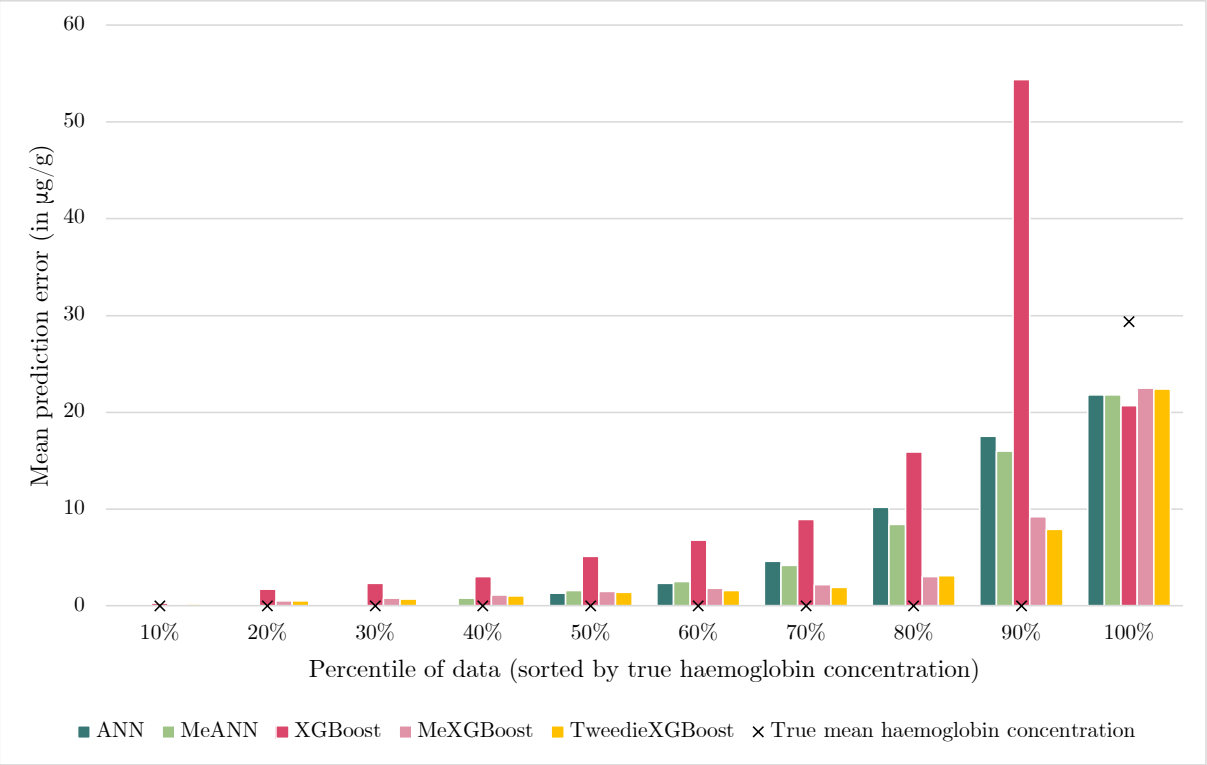



Figure 14: Mean prediction errors per model (represented by the bars) and true median haemoglobin concentrations (represented by the crosses) by tenth percentile.

*Notes:* This figure is similar to Figure 8 using means instead of medians. To create this figure, we first sort the true dependent variable in the test set and create ten buckets of equal size. We then calculate the median prediction error per bucket, per model – this represents the height of each bar. The black crosses denote the median of the true dependent variable per bucket.

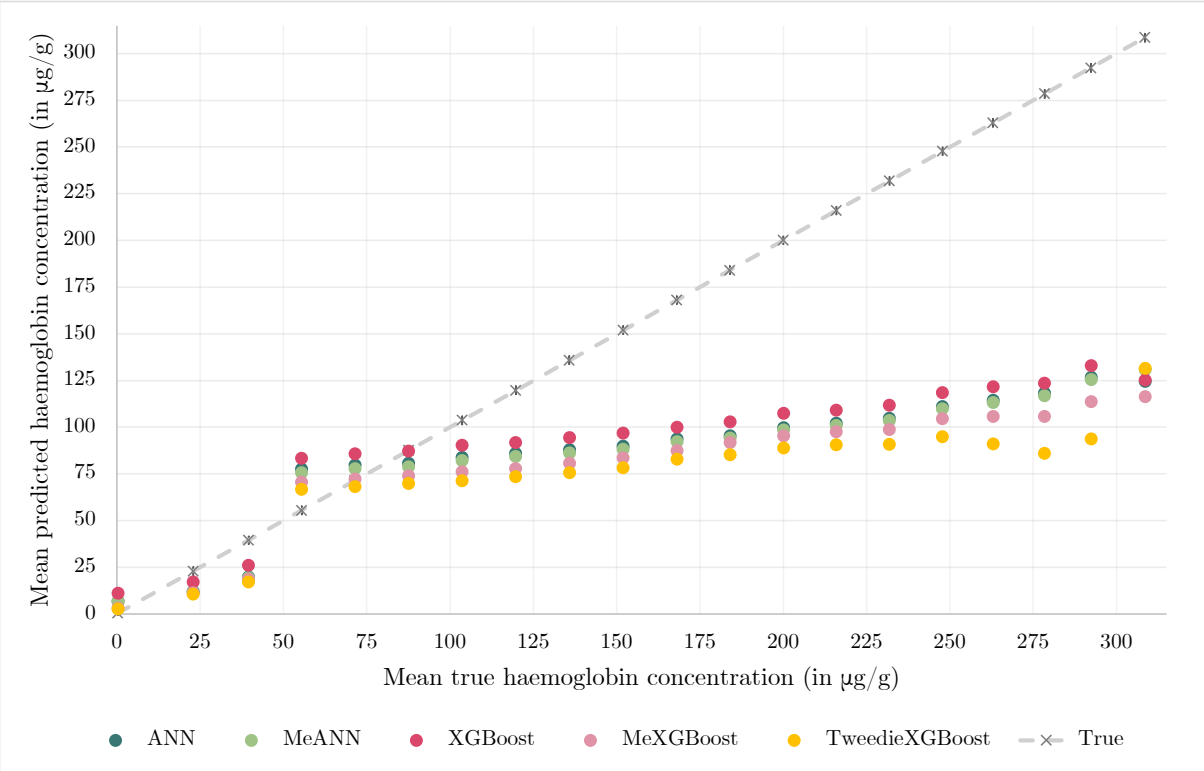
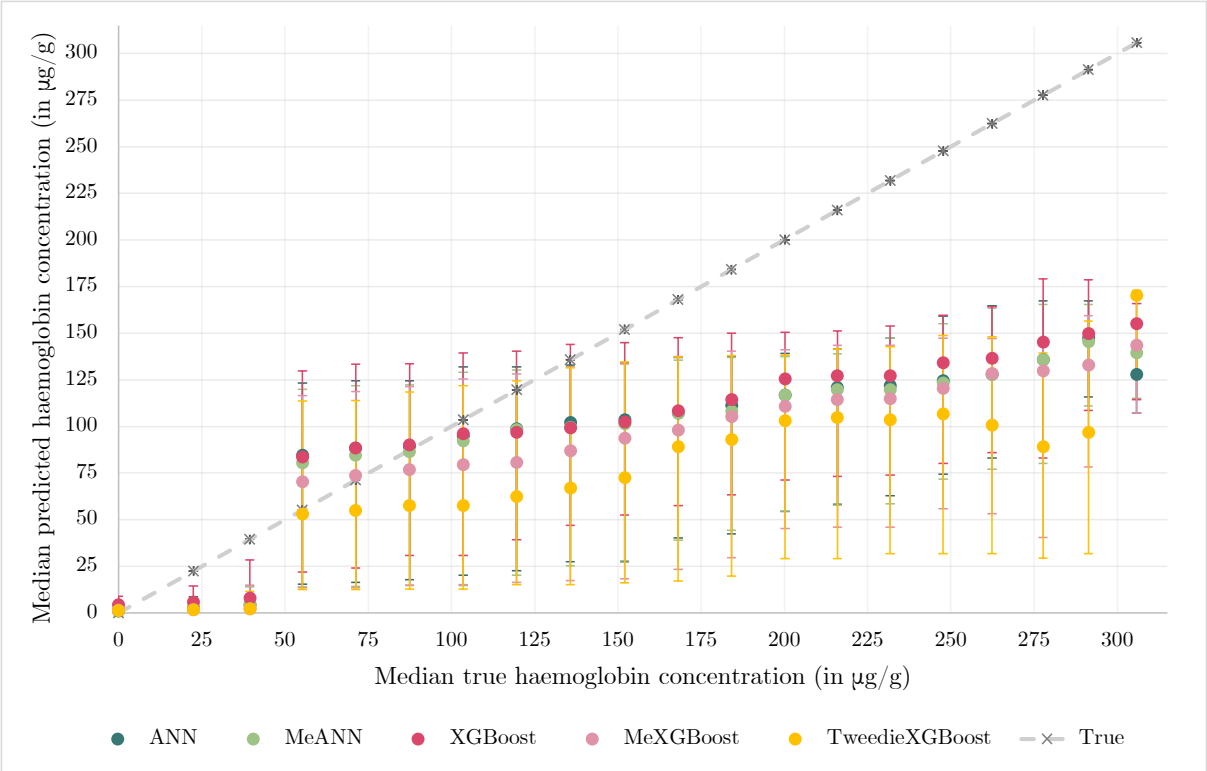
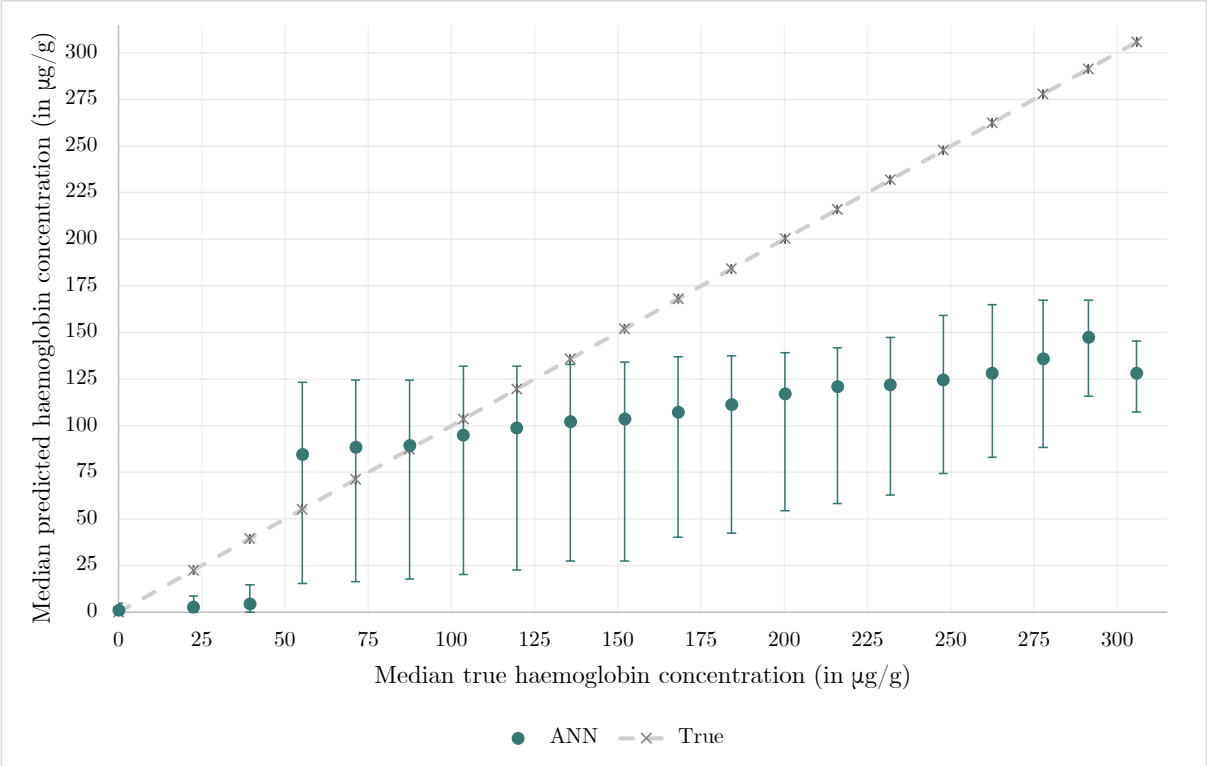



Figure 15: Mean predicted versus true values per model, calculated based on 16 intervals.

*Notes:* This figure is similar to Figure 10 using means instead of medians. Thus, to create the figure above, we first sort the data set into 16 intervals, each with a width of 20 micrograms haemoglobin per gram of faeces, to reduce the number of data points. We then plot the mean of the true dependent variable against its five predicted counterparts for each of these intervals. The gray 45-degree line is a line of reference and shows all points where predicted values are exactly equal to true values. The crosses on this dashed line denote the means and medians of the true dependent variable per interval.

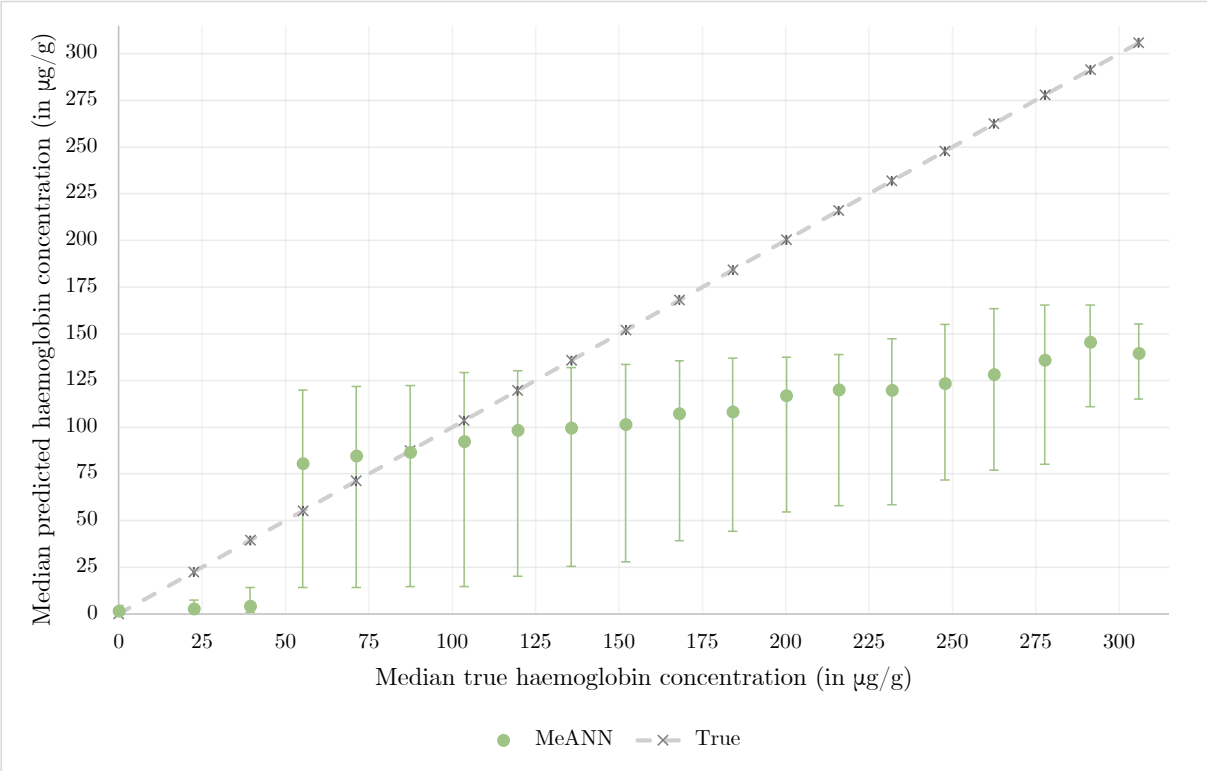



(a) All models

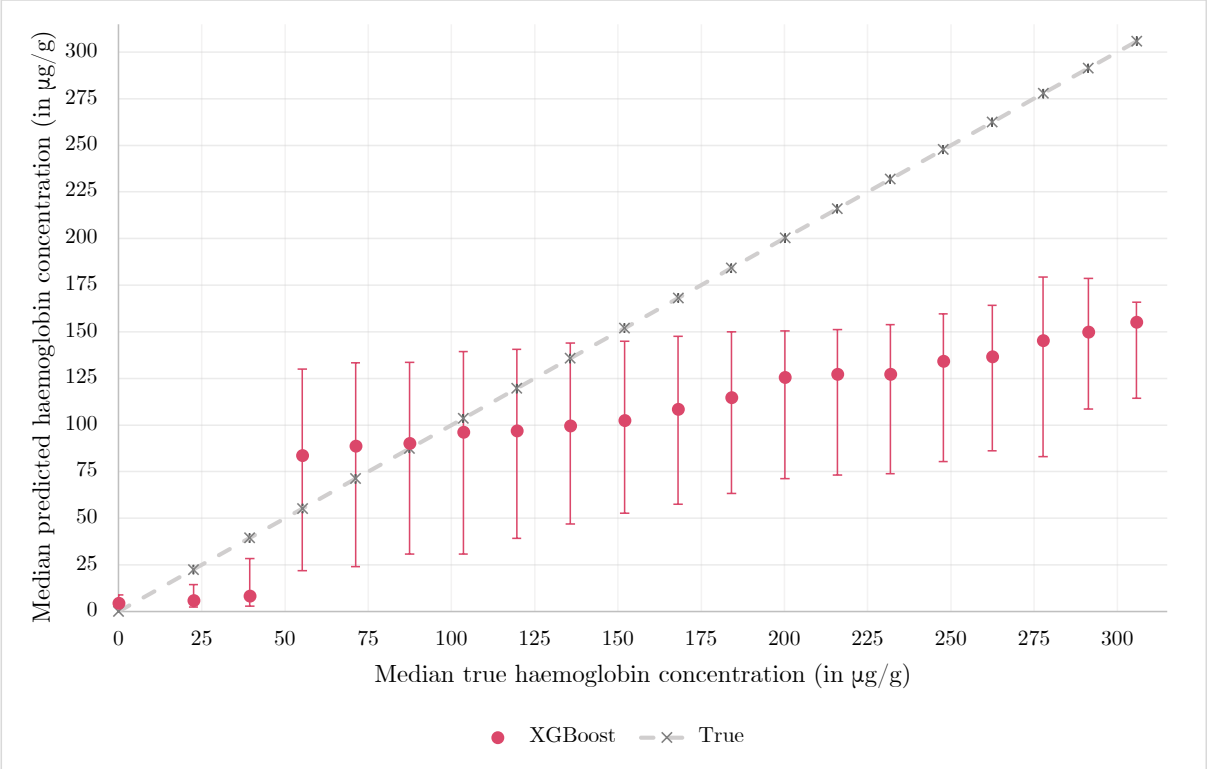



(b) ANN

Figure 16: Median predicted versus true values per model, calculated based on intervals of 20 micrograms haemoglobin per gramme of faeces, presented with interquartile ranges.

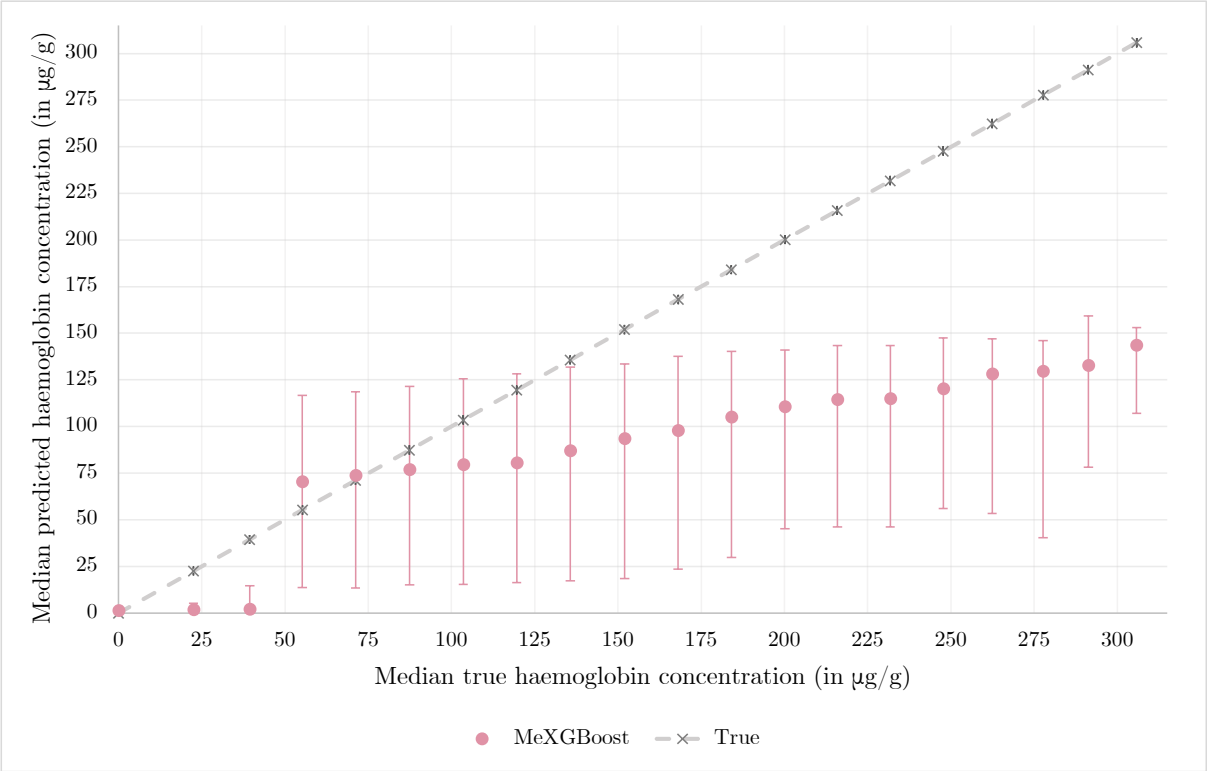



(c) MeANN

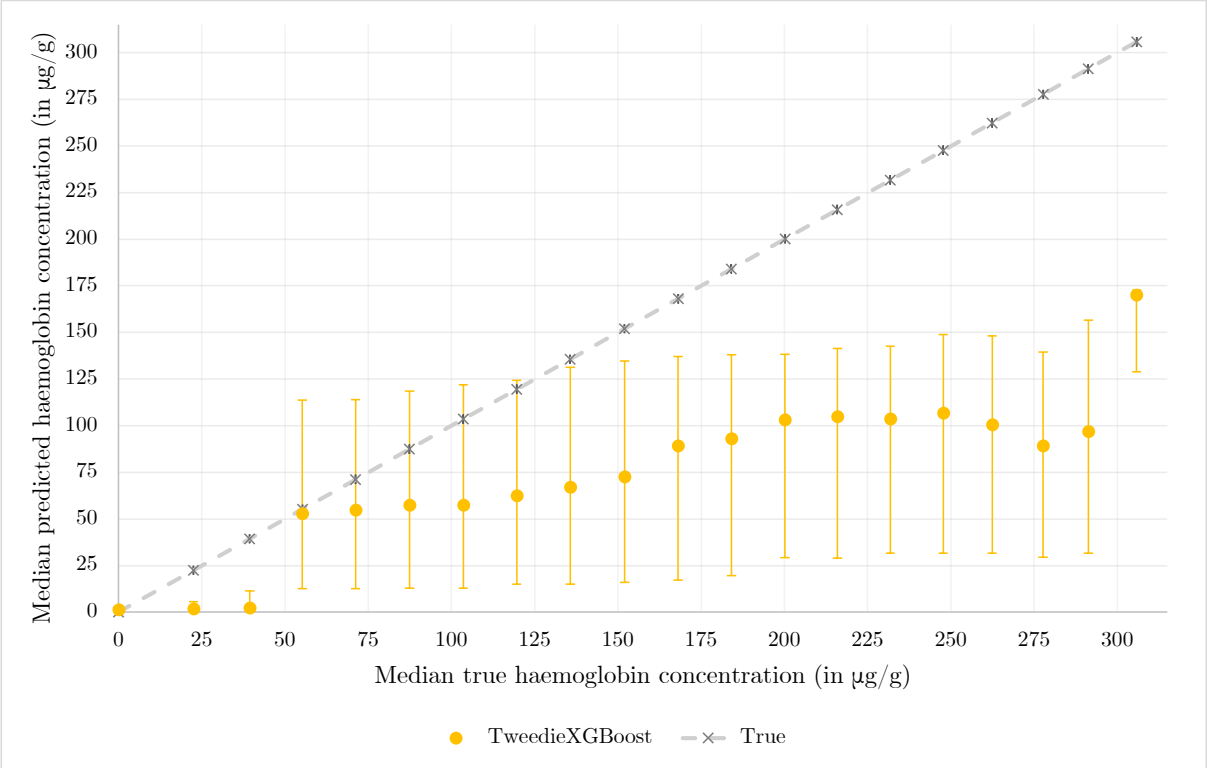



(d) XGBoost

Figure 16 **(continued)**: Median predicted versus true values per model, calculated based on intervals of 20 micrograms haemoglobin per gramme of faeces, presented with interquartile ranges.



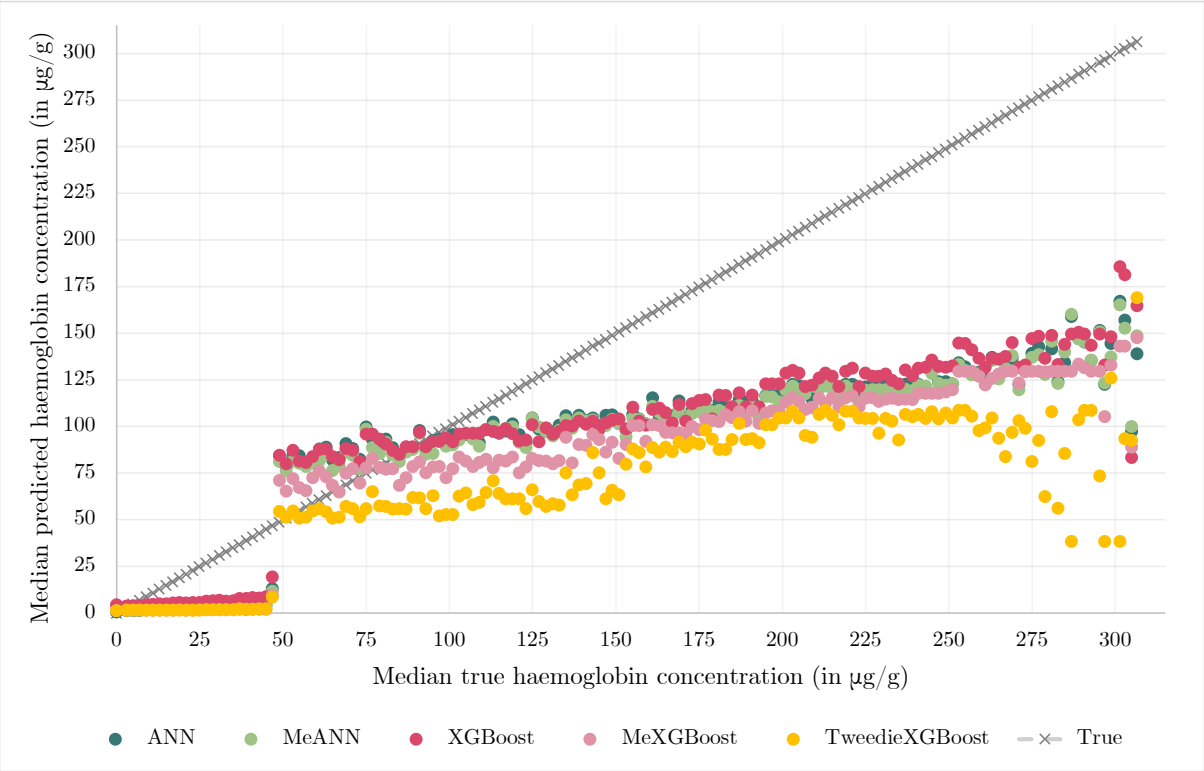

(e) MeXGBoost



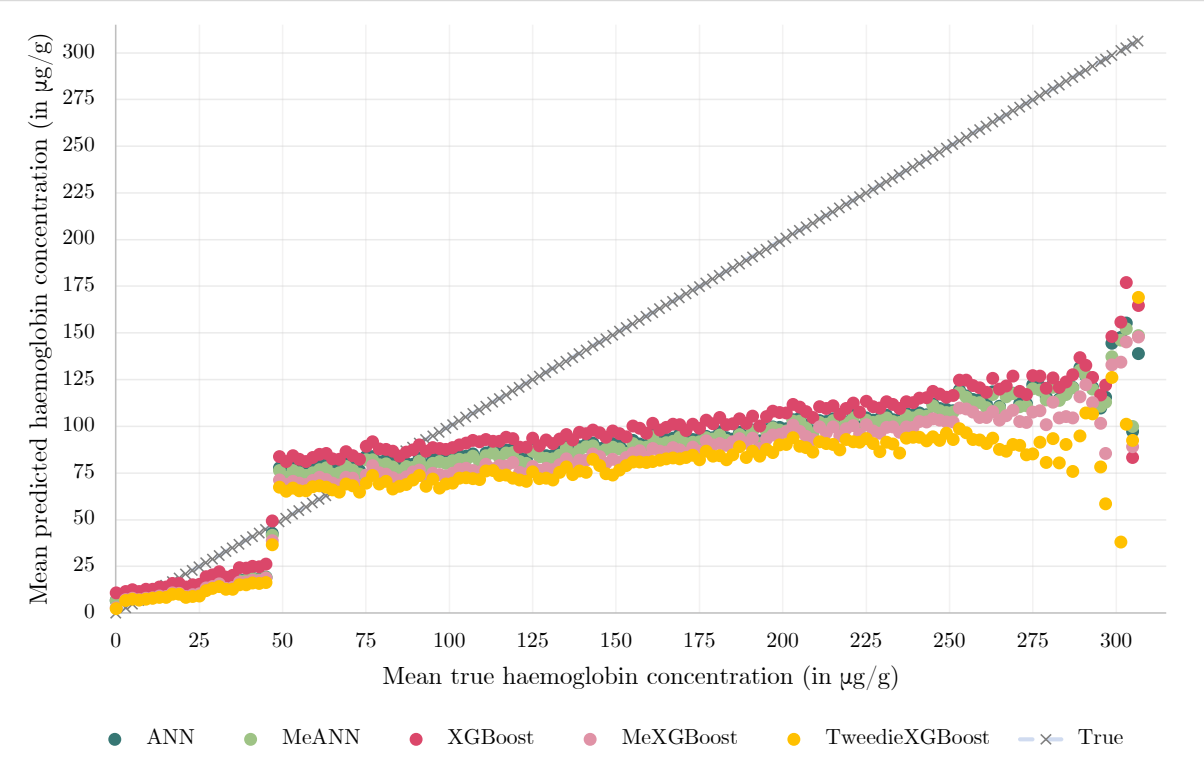

(f) TweedieXGBoost

Figure 16 **(continued)**: Median predicted versus true values per model, calculated based on intervals of 20 micrograms haemoglobin per gramme of faeces, presented with interquartile ranges.

*Notes:* These figures show the median predicted versus true values calculated per interval. For a description of the construction of these intervals, see the notes to Figure 10. The gray 45-degree line is a line of reference, and shows all points where predicted values are exactly equal to true values. The confidence intervals denote the range between the 25% quantile to the 75% quantile.



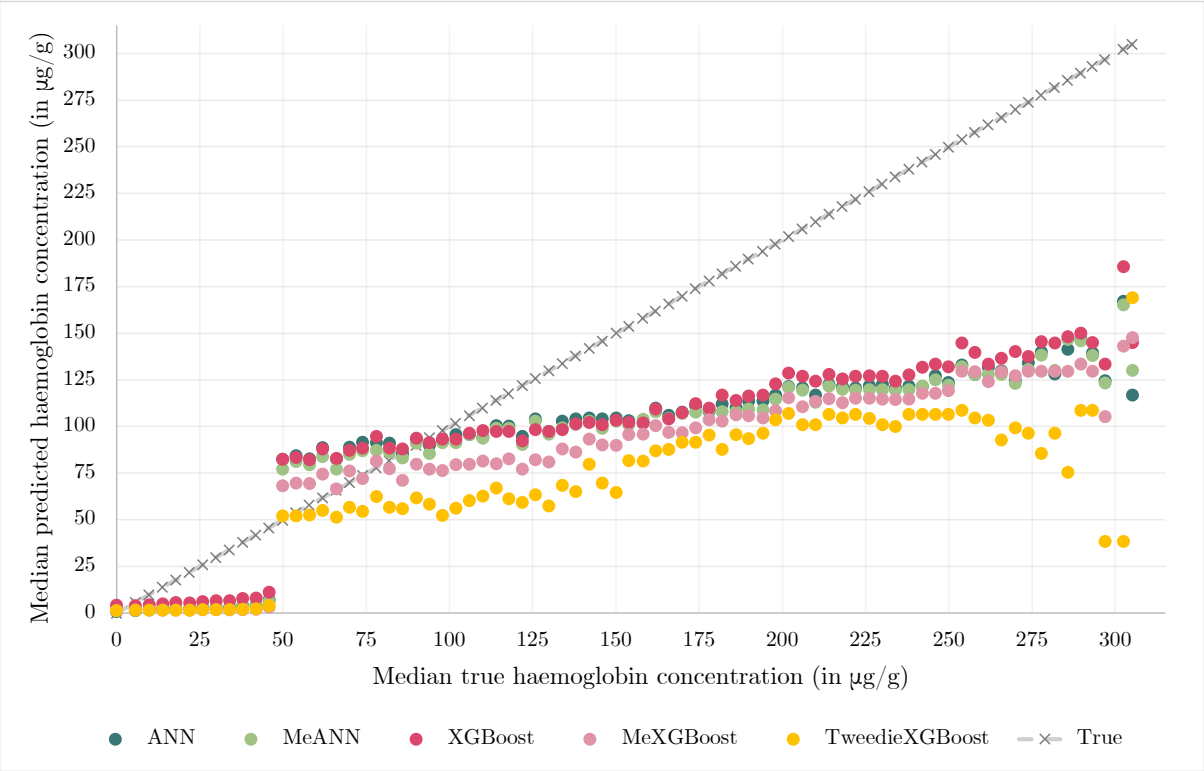

(a) Median



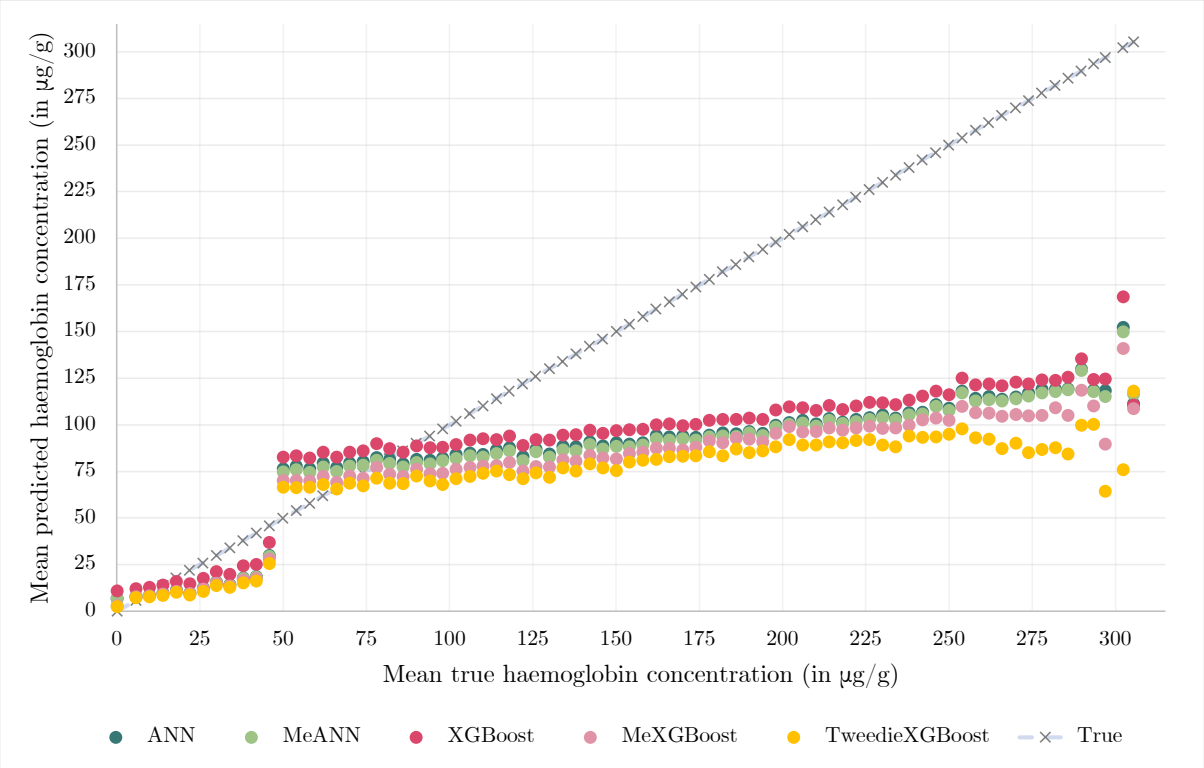

(b) Mean

Figure 17: Median (A) and mean (B) predicted versus true values per model, calculated based on intervals of 2 micrograms haemoglobin per gramme of faeces.

*Notes:* These figures show the median and mean predicted versus true values, calculated per interval. The construction of intervals are similar to that described in the notes of Figure 10, with an intervals of width 2 instead of 20. The gray 45-degree line is a line of reference, and shows all points where predicted values are exactly equal to true values.




(a) Median




(b) Mean

Figure 18: Median (A) and mean (B) predicted versus true values per model, calculated based on intervals of 4 micrograms haemoglobin per gramme of faeces.

*Notes:* These figures show the median and mean predicted versus true values, calculated per interval. The construction of intervals are similar to that described in the notes of Figure 10, with an intervals of width 4 instead of 20. The gray 45-degree line is a line of reference, and shows all points where predicted values are exactly equal to true values.