Journal of Big Data

METHODOLOGY Open Access

Generalized Estimating Equations Boosting (GEEB) machine for correlated data

Yuan-Wey Wang[1], Hsin-Chou Yang[2], Yi-Hau Chen[2] and Chao-Yu Guo[1*]

*Correspondence:
cyguo@nycu.edu.tw

[1] Division of Biostatistics and Data Science, Institute of Public Health, College of Medicine, National Yang Ming Chiao Tung University, Taipei, Taiwan
[2] Institute of Statistical Science, Academia Sinica, Taipei, Taiwan

Abstract

Rapid development in data science enables machine learning and artificial intelligence to be the most popular research tools across various disciplines. While numerous articles have shown decent predictive ability, little research has examined the impact of complex correlated data. We aim to develop a more accurate model under repeated measures or hierarchical data structures. Therefore, this study proposes a novel algorithm, the Generalized Estimating Equations Boosting (GEEB) machine, to integrate the gradient boosting technique into the benchmark statistical approach that deals with the correlated data, the generalized Estimating Equations (GEE). Unlike the previous gradient boosting utilizing all input features, we randomly select some input features when building the model to reduce predictive errors. The simulation study evaluates the predictive performance of the GEEB, GEE, eXtreme Gradient Boosting (XGBoost), and Support Vector Machine (SVM) across several hierarchical structures with different sample sizes. Results suggest that the new strategy GEEB outperforms the GEE and demonstrates superior predictive accuracy than the SVM and XGBoost in most situations. An application to a real-world dataset, the Forest Fire Data, also revealed that the GEEB reduced mean squared errors by 4.5% to 25% compared to GEE, XGBoost, and SVM. This research also provides a freely available R function that could implement the GEEB machine effortlessly for longitudinal or hierarchical data.

Keywords: Correlated data, Hierarchical data, Generalized Estimating Equations, Machine learning, Gradient boosting

Introduction

Correlated data measure the dependent variable repeatedly across multiple dimensions, such as longitudinal, clustered, spatial, or multilevel data [17]. Correlated data frequently occur in medicine, public health, and other research fields, requiring specialized statistical approaches to handle the complex correlation structure, avoid potential estimation biases, and ensure the accuracy of estimations.

Generalized Estimating Equations, also known as GEE [6, 12, 13], is a statistical method initially proposed by Liang and Zeger [11]. It extends the framework of Generalized Linear Models (GLMs) and overcomes the assumption of independence among observations, making it particularly useful for handling correlated data. One of the

strengths of GEE is that it only assumes a "working correlation matrix" to describe the correlation structure among observations. This characteristic reduces the need for such restricted distribution assumptions and keeps parameter estimation consistent even when the working correlation matrix is misspecified under mild regularity conditions [7]. The Mixed-effects model is another standard statistical method for correlated data [10, 15]. The difference between GEE and the mixed-effects model is that GEE estimates the population average effects, and the mixed-effects model estimates individual random effects.

The widespread adoption of modern technology and digitization has created vast amounts of data in recent decades. Coupled with the general availability of digital services and advancements in storage technologies, a massive accumulation of data has occurred. This phenomenon has driven the necessity for big data analysis, which has fueled the flourishing development of machine learning (ML). ML aims to construct computer models that could automatically optimize algorithms based on past experiences to predict future outcomes. Conceptually, one can think of ML as having numerous settings of candidate models and using a large amount of past experiential data to guide the computer in finding the model setting that optimizes performance indicators [9].

Currently, some popular supervised ML models include eXtreme Gradient Boosting (XGBoost) [3], Random Forest [8], and Support Vector Machine (SVM) [4]. As one of the fastest-growing technologies, data scientists applied ML in various fields such as finance, marketing, computer vision, aerospace, biomedicine, etc. Every discipline is increasingly utilizing ML for prediction and decision support. Over the past two decades, ML has made significant progress and achievements, from academic research to the most popular commercial applications [16].

Thus, in the era of big data, in addition to traditional statistical methods, ML has provided advanced choices for data analysis. Numerous studies have compared statistical methods to ML, with some articles indicating that ML outperforms statistical methods [2, 14, 19]. However, limited research discusses more complex data structures, such as correlated or hierarchical ones. Therefore, we propose a novel algorithm, the Generalized Estimating Equations Boosting (GEEB) machine, to integrate the gradient boosting technique from ML algorithms into the GEE. Under such hybrid algorithms, we aim to create a new ML model to deal with correlated data, avoid biased estimates, and provide a more accurate prediction.

Materials and methods

Generalized Estimating Equations Boosting Machine

GEE

The core of the new machine is the GEE. Here, we briefly introduce the fundamentals of the GEE. Assume that $y_{ij}, i = 1tok$ and $j = 1ton_i$ represent the jth response of the ith subject, which has a vector of covariates x_{ij}. There are n_i measurements on subject i, and the maximum number of measurements per subject is T. Let the responses of the ith subject be $y_i = [y_{i1}, \ldots, y_{ini}]\prime$ with corresponding means$\mu_i = [\mu_{i1}, \ldots, \mu_{ini}]\prime$.

The marginal mean μ_{ij} of the response y_{ij} is related to a linear predictor through a link function $g(\mu_{ij}) = x_{ij}'\beta$, and the variance of y_{ij} depends on the mean through a variance function $\nu(\mu_{ij})$ for generalized linear models (GLM).

Solving the generalized estimating equations, we could obtain the estimate of the parameter:

$$S(\beta) = \sum_{i=1}^{k} \frac{\partial \mu_i'}{\partial \beta} V_i^{-1}\left(y_i - \mu_i(\beta)\right) = 0$$

where V_i is the working covariance matrix of Y_i

We only require the mean and the covariance of Y_i in the GEE method, we do not need the full specification of the joint distribution of the correlated responses. This feature of the GEE is desirable and leads to a convenient way of analysis since the joint distribution for noncontinuous outcome variables involves high-order associations and is complicated to specify. In addition, the regression parameter estimates are consistent even when the working covariance is incorrectly specified. However, the GEE approach can lead to biased estimates when missing responses depend on previous responses. The "Weighted Generalized Estimating Equations under the MAR Assumption" can provide an unbiased estimate.

Working correlation matrix Suppose $R_i(\alpha)$ is an $n_i \times n_i$ "working" correlation matrix specified by the vector of parameters. The covariance matrix of Y_i is modeled as:

$$V_i = \varphi A_i^{\frac{1}{2}} W_i^{-\frac{1}{2}} R(\alpha) W_i^{-\frac{1}{2}} A_i^{\frac{1}{2}}$$

where A_i is a diagonal matrix $(n_i \times n_i)$ whose jth diagonal element is $\nu(\mu_{ij})$ and W_i is a diagonal matrix $(n_i \times n_i)$ whose jth diagonal is w_{ij}, where w_{ij} is a variable indicating the weight. If not weighted, $w_{ij} = 1$ for all i and j. If $R_i(\alpha)$ is the true correlation matrix of Y_i, then V_i is the true covariance matrix of Y_i.

In practice, the working correlation matrix is usually unknown, which must be estimated in the iterative fitting process by using the current value of the parameter vector β to compute appropriate functions of the Pearson residual: $e_{ij} = \frac{y_{ij} - \mu_{ij}}{\sqrt{\nu(\mu_{ij})/w_{ij}}}$.

If the working correlation matrix is the identity matrix (I), the GEE reduces to the independence estimating equations. The table from SAS [20] demonstrates the working correlation structure [21].

GEEB

GEEB has a hybrid design with GEE and gradient boosting. The strength of the GEEB algorithm lies in its ability to handle complex relationships in correlated data while optimizing the algorithm further through the gradient-boosting technique.

Gradient Boosting is a prevailing ML algorithm that applies the idea of gradient descent to ensemble learners. In the gradient boosting framework, each iteration builds a new

learner based on the prediction errors calculated by the learner in the previous iteration. When the iterations meet the stopping rule, all the predictions from the iterations are weighted and summed to obtain the final prediction. More specifically, because the loss function is the difference between the predicted and actual values, the goal of gradient boosting is to progressively move towards the minimum value of the loss function by minimizing the prediction errors in each iteration. Negative gradients of the loss function computed at the previous iteration and learning rate determine the direction and range of progressive movement in the current iteration. In other words, the algorithm minimizes the loss function and improves the overall prediction accuracy by updating the model based on the negative gradients.

The GEEB algorithm has four components: an initial setting and three computational steps. The initial stage defines the input dataset and the loss function. The input dataset contains n samples with some input features (x_i) and a continuous output feature (y_i), represented as$\{(x_i, y_i)\}_{i=1}^{n}$. When the dependent variable (y) is continuous, the algorithm defines the loss function as a modified version of the mean squared error:$L\big(y_i, F(x_i)\big) = \frac{1}{2}(y_i - F(x_i))^2$. Here, $L(,)$ represents the loss function, y_i denotes the actual outcome of the i^{th} data point, and $F(x_i)$ represents the model's predicted outcome for the i^{th} data point. Modifying the loss function is crucial as it facilitates more straightforward computation of gradients in subsequent steps.

After defining the input dataset and the loss function, the first step is to compute the initial prediction values of the model. The initial prediction value is a constant number chosen to start the iterations at the most efficient point. In this case, the initial prediction value, denoted as $F_0(x)$, is defined as $F_0(x) = arg\min_{F(x_i)} \sum_{i=1}^{n} L(y_i, F(x_i))$. Thus, when the loss function is defined as $L\big(y_i, F(x_i)\big) = \frac{1}{2}(y_i - F(x_i))^2$, the initial prediction value is set to the mean of the feature values$\left(F_0(x) = \frac{\sum_{i=1}^{n} y_i}{n}\right)$.

The second step of the algorithm involves the iterative model updates for M times. Each iteration consists of five parts: (A), (B), (C), (D), and (E). The iterations continue until the M^{th} iteration converges. Unlike the conventional gradient boosting machines that incorporate all input features, the GEEB randomly selects some input features when building the model to reduce predictive errors. In part (A), a subset of the data is created from the dataset by randomly selecting some features to generate the model. This subset is denoted as$\{(x_i\prime, y_i)\}_{i=1}^{n}$, where $x_i\prime$ represents the selected features and y_i represents the target feature. Part (B) involves calculating the residuals between the true and predicted outcomes of the subset (A). The residuals are computed as $r_{i,m} = -\left[\frac{\partial}{\partial F(x_i\prime)} L\big(y_i, F(x_i\prime)\big)\right]_{F(x_i\prime)=F_{m-1}(x_i\prime)} = y_i - F(x_i\prime), i = 1 \ldots n$. In part (C), the residuals calculated in (B) are used to fit the generalized estimating equations, obtaining the coefficients for this iteration. Part (D) uses the coefficients obtained in (C) to predict the residuals for the entire dataset in this iteration. Finally, in part (E), the progress of this iteration's prediction for the model is updated. The predicted values of the residuals for this iteration, denoted as$p_{i,m}$, are multiplied by the learning rate (ν) and added to the previous overall predicted value$F_{m-1}(x_i)$. The resulting computation represents the prediction for this iteration,$F_m(x_i)$.

After M iterations, the third step is to output the overall prediction results of the model, denoted as $F_M(x_i)$.

The following presents the GEEB algorithm.

Algorithm: Generalized Estimating Equations Boosting Machine

Input: Data $\{(x_i, y_i)\}_{i=1}^n$ and a differentiable loss function $L\big(y_i, F(x_i)\big) = \frac{1}{2}(y_i - F(x_i))^2$

Step 1: Initialize the model with a constant value $F_0(x) = arg\min_{F(x_i)} \sum_{i=1}^n L(y_i, F(x_i))$

Step 2: For m=1 to M

(A) Select features randomly to create a new subset: $\{(x_i', y_i)\}_{i=1}^n$

(B) $r_{i,m} = -\left[\frac{\partial}{\partial F(x_i')} L\big(y_i, F(x_i')\big)\right]_{F(x_i')=F_{m-1}(x_i')}$, $i = 1 \ldots n$

(C) Utilize the subset to fit the GEE model to $r_{i,m}$

(D) Utilize the input dataset to generate a predicted residual $p_{i,m}$ for each subject with the fitted GEE in (C)

(E) Update $F_m(x_i) = F_{m-1}(x_i) + \nu \cdot p_{i,m}$

Step 3: Output $F_M(x_i)$

Materials

This chapter describes the data source, the detailed background of simulation studies, and the application to a real-world dataset, the Forest Fire Data.

Data source

The Institute of Digital Research and Education (IDRE) at the University of California, Los Angeles, published the HDP simulated data in July 2012 [1]. The HDP data is based on a large-scale lung cancer-related study. The correlations exist in its hierarchical structure, which consists of three nested levels: Doctors are nested within hospitals, and patients are nested within doctors. Researchers could adjust the number of hospitals, doctors, and patients according to their research requirements.

The simulated data includes nine different outcomes. For this research, we select tumor size, which follows a Gaussian distribution, as the target output feature. The patient-related features include age (Age), marital status (Married), family history (FamilyHx), smoking history (SmokingHX), sex (Sex), cancer stage (CancerStage), length of stay in hospital (LengthofStay), white blood cell count (WBC), red blood cell count (RBC), body mass index (BMI), interleukin-6 (IL6), and C-reactive protein (CRP). At the doctor level, there is doctor ID (DID), the experience of the doctor (Experience), the quality of the school doctors trained (School), and the number of lawsuits (Lawsuits).

Note that the variable "School" is divided into two categories (top vs. average). Due to the highly imbalanced distribution, the "school" variable may have only one group that introduces errors in estimating the GEE function with the R package. Therefore, we did not include the "school" variable in simulation studies. The hospital-related features include hospital ID (HID) and Medicaid at the given hospital (Medicaid). Consequently, there are 17 predictors in the simulation study. Note that not all 17 features are related to the target response, and these features are noises to the predictive models.

Real-world data

Cortez and Morais [5] published the Forest Fire Data. This dataset covers the period from January 2000 to December 2003 and includes records of forest fires in the Montesinho Natural Park in northeastern Portugal. Multiple institutions collected the data and encompassed numerous variables, such as the Fire Weather Index (FWI) [22], spatial, temporal, and weather-related information.

We generated a new feature for "season" to construct the third-level hierarchical structure. In this way, the day is nested within the month, and the month is nested within season. Note that "season" was derived from the "month" variable. The four seasons are (1) Spring, from March to May; (2) Summer, from June to August; (3) Autumn, from September to November; and (4) Winter, from December to February. As a result, there are 14 variables (refer to Table 1), and the sample size is 517.

Experiment

We examine the consistency and accuracy of (1) the new machine GEEB, (2) the statistical method GEE, (3) the SVM, and (4) XGBoost under different hierarchical structures and sample sizes. Regarding the hyperparameter of SVM, the kernel is a radial basis function (RBF). Hyperparameters of XGB are objective = 'reg:squarederror', nrounds = 50, and verbose = 0. Other settings of hyperparameters yielded similar

Table 1 The preprocessed Forest Fire Data attributes

Attribute	Description	Value
X	x-axis coordinate	[1,9]
Y	y-axis coordinate	[2,9]
Season	Season of the year	1: spring, 2: summer, 3: fall, 4: winter
Month	Month of the year	jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
Day	Day of the week	mon, tue, wed, thu, fri, sat, sun
FFMC	Fine Fuel Moisture Code	[18.7, 96.2]
DMC	Duff Moisture Code	[1.1, 291.3]
DC	Drought Code	[7.9, 860.6]
ISI	Initial Spread Index	[0.0, 56.1]
temp	Temperature (in °C)	[2.2, 33.3]
RH	Relative humidity (in %)	[15, 100]
wind	Wind speed (in km/h)	[0.4, 9.4]
rain	The accumulated precipitation within the previous 30 min (in mm/m^2)	[0.0, 6.4]
area	Total burned area (in ha)	[0.00, 1090.84]

results. When developing simulation studies, we tuned the SVM and XGB with different parameter settings, such as the max_depth and learning_rate for the XGB. The results could be better or worse. In each scenario, there are 1000 repetitions. Each repetition could find its best parameter setting, but the comparisons are similar. Therefore, we used the most common settings for SVM and XGB.

Simulation studies

The core concept of the GEEB machine involves the random selection of features. Note that validation sets in the training data could find the optimal proportion. However, we randomly select eight proportions (30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100%) of features in GEEB denoted as Model 1 to Model 8. The new approach is more promising if the GEEB outperforms other methods without an optimal proportion obtained by ten-fold cross-validation. Table 2 presents detailed model settings.

Next, we examine the impact of sample size on the predictions. Therefore, three sample sizes were defined: (A) small, (B) medium, and (C) large. Due to the random variation ± 1 set for the number of doctors and patients, the sample size is approximately estimated as the mean value. The three estimated sizes with the minimum and maximum in the brackets are (A) small sample: 200 [72, 405], (B) medium sample: 500 [200, 720], (C) large sample: 1000 [650, 1500]. Because the results are consistent from 72 to 1,500 patients, we did not increase the sample size after 1,500. Since a small sample size introduces more statistical issues than big data, the simulation study suggests that a minimum of 72 subjects is sufficient to implement the GEEB.

Additionally, we explore the impact of different hierarchical data structures and consider five scenarios in three different sample sizes: (1) a structure with a small number of hospitals, followed by some doctors, and then more patients in a ratio of 1:3:5. (2) A structure with a small number of hospitals, followed by more doctors and then a more significant number of patients, in a more disparate ratio of 1:5:9. (3) An equal number of hospitals, doctors, and patients in a balanced ratio 1:1:1. (4) A structure with many hospitals, followed by some doctors, and then a small number of patients in a ratio of 5:3:1. (5) A structure with even more hospitals, followed by some doctors and a few patients, with a more extreme ratio of 9:5:1. Tables 3, 4, 5 show a detailed summary of the hierarchical structures.

Finally, the research framework diagram in Fig. 1 indicates the study flow. In the beginning, the input datasets undergo data preprocessing, and then the dataset is split into an

Table 2 Settings of Model 1–Model 8

Model	Content
Model 1	GEEB with randomly selected 30% features
Model 2	GEEB with randomly selected 40% features
Model 3	GEEB with randomly selected 50% features
Model 4	GEEB with randomly selected 60% features
Model 5	GEEB with randomly selected 70% features
Model 6	GEEB with randomly selected 80% features
Model 7	GEEB with randomly selected 90% features
Model 8	GEEB with all features

Table 3 Parameter settings of a small sample size

(A) 200 [72, 405]				
Scenario	**Ratio**	**N of hospitals**	**N of doctors**	**N of patients**
Scenario A1	1:3:5	2	6:8	11:13
Scenario A2	1:5:9	2	7:9	14:16
Scenario A3	1:1:1	6	5:7	5:7
Scenario A4	5:3:1	12	6:8	1:3
Scenario A5	9:5:1	15	7:9	1:3

Table 4 Parameter settings of a medium sample size

(B) 500 [200, 720]				
Scenario	**Ratio**	**N of hospitals**	**N of doctors**	**N of patients**
Scenario B1	1:3:5	3	9:11	15:17
Scenario B2	1:5:9	2	10:12	19:21
Scenario B3	1:1:1	8	7:9	7:9
Scenario B4	5:3:1	16	9:11	2:4
Scenario B5	9:5:1	20	10:12	1:3

Table 5 Parameter settings of a large sample size

(C) 1000 [650, 1500]				
Scenario	**Ratio**	**N of hospitals**	**N of doctors**	**N of patients**
Scenario C1	1:3:5	4	11:13	19:21
Scenario C2	1:5:9	3	13:15	24:26
Scenario C3	1:1:1	10	9:11	9:11
Scenario C4	5:3:1	20	11:13	3:5
Scenario C5	9:5:1	25	13:15	2:4

80% training set to build the models. The remaining 20% of the data is the testing set that evaluates the predictive performance. Lastly, we record the predictive performance in every scenario.

Regarding other hyperparameters, the learning rate for the GEEB model is set to 0.1. The number of iterations for the GEEB model is set to 100 since the convergence takes many iterations. Lastly, the simulation of the HDP dataset repeats 1000 times for each parameter setting.

The correlation matrix varies in every repetition in each scenario. Take scenario A1, for example, 11–13 patients are nested within 6 to 8 doctors, who are nested within two hospitals. At the doctor level, the dimension of the correlation matrix could be 11×11, 12×12, or 13×13. At the hospital level, the size is 6×6, 7×7, or 8×8. Because there are 1000 repetitions, A1 yields 6000 correlation matrixes. The simulation study generates 6000*15 = 9000 correlation matrixes (15 Scenarios: A1-A5, B1-B5, C1-C5). In the Additional file 1: Table S1, we display one correlation matrix of scenario A1 at the doctors level. Additional file 1: Table S2 shows an example data in scenario C1.

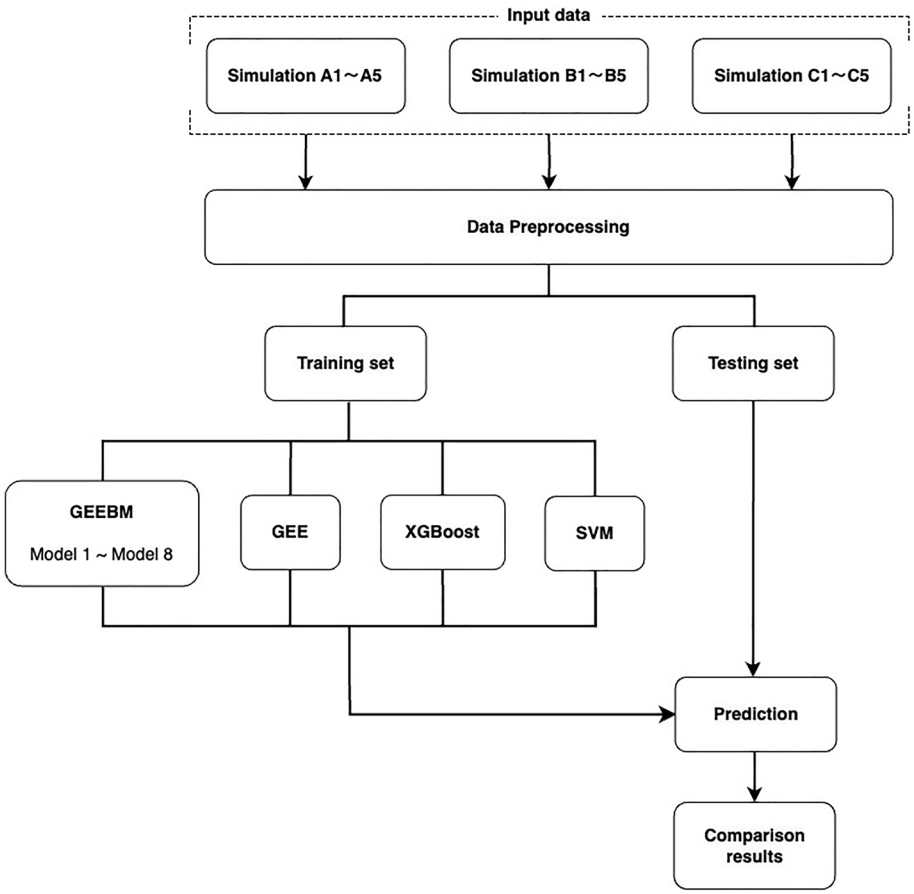

Fig. 1 Research framework diagram

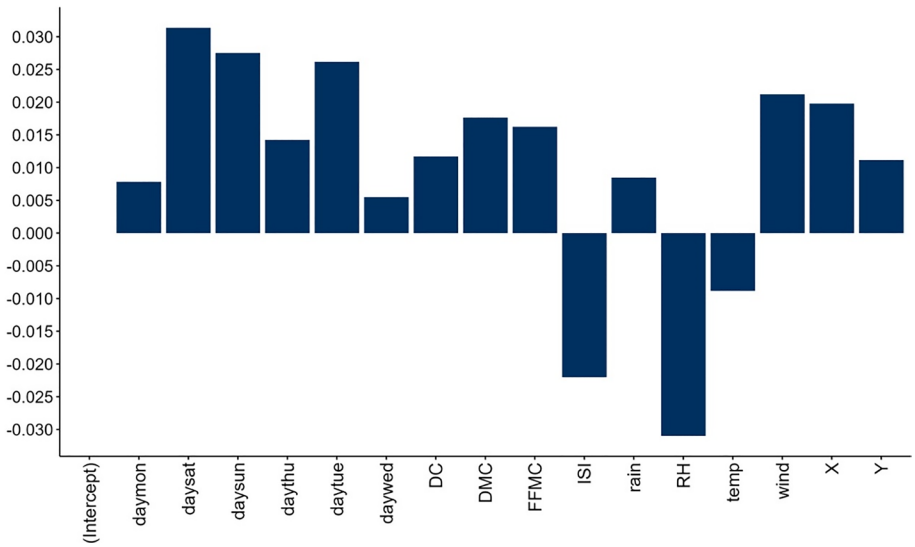

Fig. 2 Visualization of feature importance of the Forest Fire Data (single run)

Application to a real-world data

In the Forest Fire Data, the preprocessing step standardized all input features. We split the data into 80% training and 20% testing to evaluate the performance. Due to the stochastic nature of data splitting and feature selection, we will repeat the analysis one or 100 times. Tables 13, 14 and Fig. 2 reveal the analysis results and Feature Importance.

Evaluation metric

The Mean Square Error (MSE) measures the performance since the output feature is Gaussian. The MSE is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - F(x_i))^2$$

The formula represents the expected value of the squared difference between the true values (y_i) and the predicted values ($F(x_i)$) for each subject in a dataset of size n. Therefore, a smaller MSE indicates that the model's overall predicted results are closer to the actual values, meaning better performance, and vice versa.

Results

Computer simulations and applications to the Forest Fire Data are implemented by R version 4.2.1 R Core Team [18]. R: A language and environment for statistical computing. R Foundation for Statistical Computing). A computer with 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50 GHz and 16 GB of RAM in a 64-bit platform is used to implement all the experiments.

Simulation results

The new model GEEB performed well according to simulation results. The GEEB, with the selection of all features, is consistently superior to the benchmark GEE in Tables 6, 7, 8. With a suitable random feature selection proportion, the GEEB has a further improvement.

We discovered that three factors, the proportion of random feature selection, sample size, and hierarchical structure, impact model performance. The GEEB and GEE

Table 6 Simulation results of GEEB and GEE with a small sample size

		A1	A2	A3	A4	A5
GEEB	Model 1	99.01835516	95.70521460	101.94608001	107.97686232	102.39713434
	Model 2	99.48542540	95.80135649	102.09502575	108.60056002	102.65618209
	Model 3	100.08083977	96.03755343	102.45040297	109.33152319	103.01510905
	Model 4	100.29922203	96.13009608	102.55449120	109.53844516	103.11828510
	Model 5	100.48693837	96.21545902	102.67153983	109.73710182	103.22835607
	Model 6	100.51547112	96.22897508	102.68758588	109.76956898	103.24548101
	Model 7	100.53191486	96.23657640	102.69762296	109.78681880	103.25546115
	Model 8	100.53314727	96.23727716	102.69849263	109.78816782	103.25626760
GEE		100.53386987	96.23774067	102.69909140	109.78894802	103.25681055

Table 7 Simulation results of GEEB and GEE with a medium sample size

		B1	B2	B3	B4	B5
GEEB	Model 1	94.65760827	93.53285239	96.41856104	99.08916344	98.34733859
	Model 2	94.59880891	93.35856425	96.29459362	98.93853969	98.26241921
	Model 3	94.62586832	93.37919664	96.32194755	98.90918803	98.29750455
	Model 4	94.67019540	93.40186582	96.35473784	98.92699583	98.33037117
	Model 5	94.70640839	93.43028503	96.38531918	98.94879956	98.36590894
	Model 6	94.71352895	93.43571434	96.39169739	98.95170612	98.37284096
	Model 7	94.71741330	93.43884560	96.39519533	98.95420109	98.37652539
	Model 8	94.71776080	93.43916545	96.39550407	98.95448441	98.37686670
GEE		94.71798990	93.43940338	96.39570954	98.95471693	98.37712698

Table 8 Simulation results of GEEB and GEE with a large sample size

	C1	C2		C3	C4	C5
GEEB	Model 1	94.02394485	93.74645504	95.17534831	96.28861670	95.84736003
	Model 2	93.81084305	93.56541182	94.98002746	96.09494579	95.61835417
	Model 3	93.71743381	93.48196280	94.90869049	96.02364853	95.51996369
	Model 4	93.71736127	93.48402925	94.90935079	96.03299968	95.51834122
	Model 5	93.72468587	93.49760871	94.92597152	96.04065702	95.52506366
	Model 6	93.72760097	93.50035054	94.92810249	96.04305602	95.52754263
	Model 7	93.72897819	93.50179523	94.92965753	96.04428952	95.52844997
	Model 8	93.72912492	93.50193532	94.92981440	96.04445820	95.52858655
GEE		93.72922513	93.50204072	94.92992517	96.04458325	95.52869609

Table 9 Simulation results of GEEB with Model 3, GEE, XGBoost, and SVM concerning the hierarchical structure

	GEEB with Model 3		GEE		SVM		XGBoost	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
A1	100.08083977	26.95404996	100.53386987	27.01249069	112.12767754	28.54215119	112.45269936	29.44731637
B1	94.62586832	15.18795180	94.71798990	15.19590222	97.14783528	14.83062427	97.25146238	14.91062380
C1	93.71743381	10.99309471	93.72922513	11.00408210	90.40457748	9.85050686	89.93082567	9.61089664
A2	96.03755343	22.65889734	96.23774067	22.72473623	104.40697571	22.75368670	104.88445968	23.75001617
B2	93.37919664	16.73492992	93.43940338	16.74627032	95.02183905	15.85076210	96.11347731	15.67838128
C2	93.48196280	11.11303043	93.50204072	11.11896335	86.94844371	9.21310427	88.78844514	9.10177358
A3	102.45040297	24.35281831	102.69909140	24.41306419	118.07327362	27.03897431	114.26875449	26.96785652
B3	96.32194755	14.34927426	96.39570954	14.33870888	106.81904823	15.78853772	100.11787803	14.83848559
C3	94.90869049	10.16099839	94.92992517	10.15832247	101.35317458	10.98376852	92.15629237	9.65414782
A4	109.33152319	27.73652633	109.78894802	27.96769900	133.14030348	34.12907936	128.82575872	34.37172626
B4	98.90918803	14.57846063	98.95471693	14.59775801	121.15671344	18.36387217	110.17394078	17.18216334
C4	96.02364853	10.37016019	96.04458325	10.37370905	114.01629456	12.36584604	99.61407890	10.44030172
A5	103.01510905	21.90171557	103.25681055	21.95059436	127.62577943	26.34114910	120.12543242	25.97924278
B5	98.29750455	14.52335686	98.37712698	14.54756006	124.33607693	18.58431286	112.72063341	17.06058754
C5	95.51996369	9.40236520	95.52869609	9.41868433	116.96334753	11.41679679	100.50386001	9.88936982

perform better in A1–A2, B1–B2, and C1–C2 in Tables 6, 7, 8. These scenarios have fewer hospitals, a moderate number of doctors, and more patients.

Models within the same hierarchical structure perform better as the dataset increases (B1 vs. A1, C1 vs. B1, ..., etc.). Although the optimal feature selection proportion varies with different sample sizes and hierarchical structures, we suggest that Model 3 (with a 50% random feature selection) demonstrates consistent and satisfying predictive results (Tables 6, 7, 8). Therefore, without tenfold cross-validation searching for the optimal ratio, we adopted Model 3 in Tables 9, 10, 11, 12 for the GEEB. The GEEB with Model 3 shows a superior MSE than the SVM and XGBoost (Tables 9, 10, 11, 12).

In addition to the main results mentioned above, the subsequent sections have details in three aspects. First, we explore the impact of different random feature selection proportions in the GEEB. Secondly, we examine the influence of sample size. Lastly, we see how the predictive ability differs among the GEEB, GEE, XGBoost, and SVM.

Table 10 Simulation results of GEEB with Model 3, GEE, SVM, and XGBoost with a small sample size

	N of clusters	GEEB with Model 3		GEE		SVM		XGBoost	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
A1	[12, 16]	100.08083977	26.95404996	100.53386987	27.01249069	112.12767754	28.54215119	112.45269936	29.44731637
A2	[14, 18]	96.03755343	22.65889734	96.23774067	22.72473623	104.40697571	22.75368670	104.88445968	23.75001617
A3	[30, 42]	102.45040297	24.35281831	102.69909140	24.41306419	118.07327362	27.03897431	114.26875449	26.96785652
A4	[72, 96]	109.33152319	27.73652633	109.78894802	27.96769900	133.14030348	34.12907936	128.82575872	34.37172626
A5	[105, 135]	103.01510905	21.90171557	103.25681055	21.95059436	127.62577943	26.34114910	120.12543242	25.97924278

Table 11 Simulation results of GEEB with Model 3, GEE, SVM, and XGBoost with a medium sample size

	N of clusters	GEEB Model 3		GEE		SVM		XGBoost	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
B1	[27, 33]	94.62586832	15.18795180	94.71798990	15.19590222	97.14783528	14.83062427	97.25146238	14.91062380
B2	[20, 24]	93.37919664	16.73492992	93.43940338	16.74627032	95.02183905	15.85076210	96.11347731	15.67838128
B3	[56, 72]	96.32194755	14.34927426	96.39570954	14.33870888	106.81904823	15.78853772	100.11787803	14.83848559
B4	[144, 176]	98.90918803	14.57846063	98.95471693	14.59775801	121.15671344	18.36387217	110.17394078	17.18216334
B5	[200, 240]	98.29750455	14.52335686	98.37712698	14.54756006	124.33607693	18.58431286	112.72063341	17.06058754

Table 12 Simulation results of GEEB with Model 3, GEE, SVM, and XGBoost with a large sample size

	N of clusters	GEEB Model 3		GEE		SVM		XGBoost	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
C1	[44, 52]	93.71743381	10.99309471	93.72922513	11.00408210	90.40457748	9.85050686	89.93082567	9.61089664
C2	[39, 45]	93.48196280	11.11303043	93.50204072	11.11896335	86.94844371	9.21310427	88.78844514	9.10177358
C3	[90, 110]	94.90869049	10.16099839	94.92992517	10.15832247	101.35317458	10.98376852	92.15629237	9.65414782
C4	[220, 260]	96.02364853	10.37016019	96.04458325	10.37370905	114.01629456	12.36584604	99.61407890	10.44030172
C5	[325, 375]	95.51996369	9.40236520	95.52869609	9.41868433	116.96334753	11.41679679	100.50386001	9.88936982

The proportion of random feature selection

In the small sample scenarios (A1 to A5), Table 6 shows that the GEEB consistently outperforms the GEE even when all features are included (Model 8). Therefore, the boosting technique improved the accuracy compared to the conventional statistical approach. The optimal model is identified as Model 1, with a random feature selection proportion of 30% ($MSE_{M1,A1} = 99.01835516$, $MSE_{M1,A2} = 95.70521460$, $MSE_{M1,A3} = 101.94608001$, $MSE_{M1,A4} = 107.97686232$, $MSE_{M1,A5} = 102.39713434$).

Moving on to the medium sample scenarios (B1 to B5) in Table 7, Model 2 (40%) and Model 3 (50%) exhibit the most favorable results of GEEB ($MSE_{M2,B1} = 94.59880891$, $MSE_{M2,B2} = 93.35856425$, $MSE_{M2,B3} = 96.29459362$, $MSE_{M3,B4} = 98.90918803$, $MSE_{M2,B5} = 98.26241921$). In Table 8, large sample scenarios (C1 to C5) indicate that the optimal models of GEEB are Model 3 (50%) and Model 4 (60%), ($MSE_{M4,C1} = 93.71736127$, $MSE_{M3,C2} = 93.48196280$, $MSE_{M3,C3} = 94.90869049$, $MSE_{M3,C4} = 96.02364853$, $MSE_{M4,C5} = 95.51834122$). These results demonstrate the impact of the proportion of random feature selection on the GEEB across various sample sizes.

In Tables 6, 7, 8, Model 8 (GEEB with all features) outperforms GEE even without random feature selection, showing better predictive results than the GEE model. Furthermore, in the small sample scenarios (Table 6), the MSE in Models 1 to 7 is smaller than the numbers in Model 8. The comparisons demonstrate the improved GEEB through random feature selection. Depending on the sample size, the optimal random feature selection proportion falls between 30 to 60%. Table 7 (B2–B4) and 8 (C1-C5) show a curved pattern when the percentage is too small. Models 1 or 2 may encounter a higher MSE compared to Model 8 of the GEEB. For example, in B2, the best scenario is Model 2 (40% features, $MSE_{M2,B2} = 93.35856425$), whereas a lower selection proportion Model 1 (30% features, $MSE_{M1,B2} = 93.53285239$) performs worse than Model 2 and even underperforms the GEE ($MSE_{GEE,B2} = 93.43940338$). Similarly, in C1, the best situation is Model 4 (60% features, $MSE_{M4,C1} = 93.71736127$), whereas a lower selection proportion Model 1 (30% features, $MSE_{M1,C1} = 94.02394485$) performs worse than Model 4 and even underperforms the GEE ($MSE_{GEE,C1} = 93.72922513$). We think that the information content of 30% or 40% of the features is insufficient to provide accurate predictions.

In conclusion, although each sample size has its optimal range of random feature selection proportions, we recommend the following: (1) random feature selection is a hyperparameter for the proposed GEEB machine. The optimal selection proportion falls between 30 to 60% through simulation studies. Hence, one can find the optimal hyperparameter through techniques such as a validation set or k-fold cross-validation. (2) A 50% selection proportion consistently demonstrates stable and excellent performance across all scenarios. In cases where it is impossible to employ any validation technique, this research suggests setting the random feature selection proportion to 50%. Thus, the GEEB function in the R language has a default random feature selection proportion set to 50%.

The following two sections (Tables 9, 10, 11, 12) will explore the impact of sample size and hierarchical structure using the GEEB with Model 3.

Sample size

In Table 9, all models, including the GEEB, GEE, SVM, and XGBoost, perform better as the sample size increases. The GEEB with Model 3 in the small sample A1 has $\text{MSE}_{M3,A1} = 100.08083977$. In the medium sample scenario B1, it is $\text{MSE}_{M3,B1} = 94.62586832$. In the large sample scenario C1, it is $\text{MSE}_{M3,C1} = 93.71743381$. The pattern indicates an improvement in reduced errors as the sample size increases. For GEE, the MSE in A1 is $\text{MSE}_{GEE,A1} = 100.53386987$, in B1, it is $\text{MSE}_{GEE,B1} = 94.71798990$, and in C1, it is $\text{MSE}_{GEE,C1} = 93.72922513$. For SVM, in A1, it has $\text{MSE}_{SVM,A1} = 112.12767754$, in B1, it has $\text{MSE}_{SVM,B1} = 97.14783528$, and in C1, it has $\text{MSE}_{SVM,C1} = 90.40457748$. For XGBoost, in A1, it has $\text{MSE}_{XGBoostA1} = 112.45269936$, in B1, it has $\text{MSE}_{XGBoost,B1} = 97.25146238$, and in C1, it has $\text{MSE}_{XGBoost,C1} = 89.93082567$. In summary, under the same hierarchical structure, increasing the sample size leads to a decreasing trend in MSE, indicating improved predictive ability. Besides, SVM and XGBoost are more sensitive to sample size variations than the GEEB and GEE.

Hierarchical structure

Five types of hierarchical structures show the impact on the MSE. In Tables 10, 11, 12, there is a significant contrast between the first (ratio = 1:3:5) and fourth (ratio = 5:3:1), as well as the second (ratio = 1:5:9) and fifth (ratio = 9:5:1) hierarchical structures across all scenarios with the same dataset size. All models perform better in the first and second hierarchical structures, where the setting presents fewer hospitals, a moderate number of doctors, and the highest number of patients. In contrast, they do not perform well in the fourth and fifth hierarchical structures, when the data have more hospitals and minimal patients. Regarding the third hierarchical structure, where the data involves an equal number of hospitals, doctors, and patients, the predictive capability lies between all models.

Note that the second and fifth hierarchical structures demonstrate more extreme ratios, implying a more significant disparity in the size of hospitals, doctors, and patients. These situations investigate whether the models would exhibit more extreme MSEs. However, only in the small sample scenario A could we observe relatively notable differences. We did not see significant differences in the medium and large sample scenarios. The XGBoost and SVM are more sensitive hierarchical structures than the GEEB and GEE.

In the small sample size under the first and fourth structure, the GEEB with Model 3 yields $\text{MSE}_{M3,A1} = 100.08083977$ and $\text{MSE}_{M3,A4} = 109.33152319$; in the medium sample, it shows $\text{MSE}_{M3,B1} = 94.62586832$ and $\text{MSE}_{M3,B4} = 98.90918803$; in the large sample, it achieves $\text{MSE}_{M3,C1} = 93.71743381$ and $\text{MSE}_{M3,C4} = 96.02364853$. Similarly, GEE demonstrates similar behavior ($\text{MSE}_{GEE,A1} = 100.53386987, \text{MSE}_{GEE,A4} = 109.78894802$, $\text{MSE}_{GEE,B1} = 94.71798990, \text{MSE}_{GEE,B4} = 98.95471693, \text{MSE}_{GEE,C1} = 93.72922513$, $\text{MSE}_{GEE,C4} = 96.04458325$). However, the SVM and XGBoost show more differences with changes in hierarchical structures. For instance, in the first and fourth scenarios, the SVM yields $\text{MSE}_{SVM,A1} = 112.12767754$ and $\text{MSE}_{SVM,A4} = 133.14030348$ in the small sample, $\text{MSE}_{SVM,B1} = 97.14783528$ and $\text{MSE}_{SVM,B4} = 121.15671344$ in the

medium sample, and $\text{MSE}_{SVM,C1} = 90.40457748$ and $\text{MSE}_{SVMC4} = 114.01629456$ in the large sample. The trend is similar for XGBoost ($\text{MSE}_{\text{XGBoost},A1} = 112.45269936$, $\text{MSE}_{\text{XGBoost},A4} = 128.82575872$, $\text{MSE}_{\text{XGBoost},B1} = 97.25146238$, $\text{MSE}_{\text{XGBoost},B4} = 110.17394078$, $\text{MSE}_{\text{XGBoost},C1} = 89.93082567$, $\text{MSE}_{\text{XGBoost},C4} = 99.61407890$). Thus, we observe significant differences between the first and fourth scenarios when using the SVM and XGBoost. It may be because the two ML models are not well-suited to handle hierarchical data, leading to their inferior performance in the fourth hierarchical structure, which has relatively enormous clusters.

Furthermore, according to the increasing number of clusters, we observed that SVM and XGBoost show a clear inverse relationship in both medium and large samples. Besides, as the number of clusters increases, the predictive performance of SVM and XGBoost decreases, indicating worse performance with more clusters and more pronounced inter-cluster correlations. In contrast, the GEEB and GEE demonstrate consistent and satisfying predictions.

The SVM and XGBoost can outperform the GEEB in scenarios C1 and C2 for larger sample sizes (Table 12). The reason may be with fewer clusters and relatively large data sizes, SVM and XGBoost can overlook the inter-cluster correlation structure and treat the data as independent.

Results of the Forest Fire Data

According to the simulation study, the GEEB model with 50% random feature selection demonstrates consistent and improved predictive performance. Therefore, we adopt the GEEB with Model 3 as the default model in real-world data analysis.

The Forest Fire Data analyses for each method are shown in Table 13, indicating that the GEEB with Model 3 exhibits the minimum MSE compared to the GEE, SVM, and XGBoost. The MSE of GEE is approximately 4.5% higher than the GEEB. The XGBoost is about 25.2% higher than the GEEB. Therefore, the GEEB has a decent improvement compared to the most famous statistical model for correlated data and the most promising ML approaches, SVM and XGBoost. Feature Importance of the GEEB with Model 3 is in Table 14, and the visualization is in Fig. 2.

Discussions

In this study, we propose a new ML strategy named the Generalized Estimating Equations Boosting (GEEB) machine. This method integrates the gradient boosting technique with the gold standard model for correlated data, the GEE. Computer simulations confirmed that the GEEB outperforms the GEE. In most situations, GEEB performs better than the famous SVM and XGBoost. Besides, the GEEB demonstrates the best prediction for the Forest Fire Data. Therefore, our findings suggest: (1) the gradient boosting technique enables the GEEB to outperform the GEE model. (2) Although the XGBoost and SVM are known for their excellent predictive ability, they may not perform well

Table 13 Applications to the Forest Fire Data

GEEB Model 3		GEE		SVM		XGBoost	
Mean	SD	Mean	SD	Mean	SD	Mean	SD
2.02196796	0.26742275	2.11245426	0.40181580	2.25342487	0.40144105	2.58312353	0.33828778

Table 14 Feature importance of the Forest Fire Data (single run/averaged 100 runs)

	Feature importance (single run)	Averaged feature importance (100 runs)	
		Mean	SD
(Intercept)	$-1.84015877 \times 10^{-17}$	$-9.44600470 \times 10^{-19}$	$7.34268206 \times 10^{-18}$
daymon	0.00783564	0.00964657	0.00640527
daysat	0.03137851	0.01906195	0.00788899
daysun	0.02750425	0.01370748	0.00790682
daythu	0.01424024	0.00145060	0.00662915
daytue	0.02618028	0.01721314	0.00760060
daywed	0.00548812	0.00884680	0.00685072
DC	0.01169941	0.01493954	0.00856180
DMC	0.01763062	0.01490499	0.00925210
FFMC	0.01624885	0.00877412	0.00541663
ISI	−0.02200907	−0.02049532	0.00602142
rain	0.00847838	0.00169441	0.00866076
RH	−0.03097815	−0.01785837	0.00823336
temp	−0.00883744	0.00113329	0.01025085
wind	0.02119821	0.02607868	0.00648451
X	0.01978786	0.01864735	0.00676956
Y	0.01116722	0.00258028	0.00731071

with hierarchical data. Treating subjects as independent failed to capture the correlation structure.

This research also provides the code that computes all research results. The geebm() is an R function that implements the GEEB machine. This function has seven arguments: *formula*, *id*, *iteration*, *feature_rate*, *lrate*, *standardize*, and *data*. Note that *formula* must be specified in the format "response ~ predictors" to list the predictors (input features) and response variable (output feature) in the dataset. *id* is a vector that identifies the clusters and can support multiple levels arranged in the order of multilayer structure. *iteration* is an integer representing the number of iterations, set to default at 100 iterations. *feature_rate* represents the proportion of random feature selection. When set to 1, it uses all features; by default, it is set to 0.5, using half of the features. *lrate* is a hyperparameter for the learning rate, with a default value of 0.1. *standardize* determines whether features are standardized, and the default does not perform standardization. *data* is used to input the training dataset. For example, when training the model with the Forest Fire Data in this study, the function would be: *geebm(area ~ X + Y + FFMC + DMC + DC + ISI + temp + RH + wind + rain + day, id = c("season","month"), iteration = 100, feature_rate = 0.5, lrate = 0.1, standardize = T, data = Dataset).*

The GEEB is also inspired by the Random Forest that incorporates Bootstrap while randomly selecting features. However, the results were not satisfying. Our research aims to compare the GEEB with other benchmark ML and statistical models in correlated data. When deciding which ML models to include, we primarily considered models that are widely discussed and used in academia and industry and frequently win in various

data science competitions. Therefore, we included the XGBoost, SVM, and Random Forest. However, when conducting the simulation studies, we discovered that the randomForest package in R cannot handle datasets with more than 53 categories. Since each doctor within each hospital is treated as a separate category and there are other categorical features such as gender and cancer stage, this number exceeds the limitation. Therefore, we must exclude Random Forest in the comparison as it does not apply to the hierarchical dataset.

Integrating the concept of gradient boosting and using the statistical model GEE as the base learner, combined with a random feature selection, the proposed novel approach GEEB has several advantages. Compared to the ML model XGBoost, which also utilizes Gradient Boosting, GEEB performs better in most scenarios. GEEB can handle such data more effectively, resulting in improved predictive performance. Furthermore, compared to using the GEE model alone, after conducting 1000 simulations, we observed that GEEB achieves more accurate predictions.

Limitations

There are some limitations in this study. Firstly, the simulated data used in this study is based on the publicly available HDP dataset from UCLA, and the investigation of the impact of the level of variable correlations, such as weak to high correlations, has not been further explored.

Secondly, in this study, we investigate the predictions of tumor size. The underlining techniques of GEEB are GEE and gradient boosting, both of which support classification tasks. However, this research focused on regression tasks only. The performance of the GEEB under other types of output features is unknown.

Here, we only roughly categorized the structures into three types: (1) fewer hospitals, followed by some doctors and more patients; (2) more hospitals, followed by some doctors and fewer patients; and (3) an equal number of hospitals, doctors, and patients. The study also considered varying sample sizes, including more extreme cases. Consequently, we examined five hierarchical structures: 1:3:5, 1:5:9, 1:1:1, 5:3:1, and 9:5:1. While this design provides initial insights, we could explore more detailed hierarchical structures in future works.

Future research topics

The corresponding theoretical work and simulation studies are great topics for future research with dichotomous, ordinal, or categorical nominal correlated datasets.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40537-023-00875-5.

Additional file 1: Table S1. An Example of A1. **Table S2.** An Example of C1.

Additional file 2. This research also provides the code that computes all research results. The geebm() is an R function that implements the GEEB machine. This function has seven arguments: *formula, id, iteration, feature_rate, lrate, standardize*, and *data*. Note that *formula* must be specified in the format "response ~ predictors" to list the predictors (input features) and response variable (output feature) in the dataset. *id* is a vector that identifies the clusters and can support multiple levels arranged in the order of multilayer structure. *iteration* is an integer representing the number of iterations, set to default at 100 iterations. *feature_rate* represents the proportion of random feature selection. When set to 1, it uses all features; by default, it is set to 0.5, using half of the features. *lrate* is a hyperparameter for the

learning rate, with a default value of 0.1. *standardize* determines whether features are standardized, and the default does not perform standardization. *data* is used to input the training dataset. For example, when training the model with the Forest Fire Data in this study, the function would be: *geebm(area~X+Y+FFMC+DMC+DC+ISI+temp+RH+wind+rain+day, id=c("season","month"), iteration=100, feature_rate=0.5, lrate=0.1, standardize=T, data=Dataset)*.

Author contributions
YWW first draft manuscript, simulations, analyses, Tables, Figures, and R code. HCY provided critical comments on the methods and manuscript. YHC provided critical comments on the methods and manuscript. CYG proposed the research concept, supervised the project, additional simulations, modified and completed the manuscript, and revisions.

Funding
The National Science and Technology Council supports this work. Grant ID: 111-2118-M-A49-005 and 112-2118-M-A49-003.

Availability of data and materials
Not applicable. It is a computer simulation study.

Code availability
R codes are included as Additional file 2 for publication.

Declarations

Ethics approval and consent to participate
Not applicable. It is a computer simulation study.

Consent for publication
All authors read and approved the final manuscript for publication.

Competing interests
The authors declare that they have no conflicts of interest related to the subject matter or materials discussed in this article.

Received: 11 October 2023 Accepted: 27 December 2023
Published online: 22 January 2024

References

1. Bruin J (2012). R advanced: simulating the hospital doctor patient dataset. https://stats.idre.ucla.edu/r/codefragments/mesimulation/. Accessed 27 Oct 2022.
2. Caruana, R. and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. Proceedings of the 23rd international conference on Machine learning. 2006.
3. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
4. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20:273–97.
5. Cortez P, Morais AJR. A data mining approach to predict forest fires using meteorological data. 2007.
6. Diggle P, Diggle PJ, Heagerty P, Liang K-Y, Zeger S. Analysis of longitudinal data. Oxford: Oxford University Press; 2002.
7. Hardin JW, Hilbe JM. Generalized estimating equations. Boca Raton: CRC Press; 2012.
8. Ho TK. Random decision forests. Proceedings of 3rd international conference on document analysis and recognition, IEEE. 1995.
9. Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science. 2015;349(6245):255–60.
10. Laird NM, Ware JH. Random-effects models for longitudinal data. Biometrics. 1982;38:963–74.
11. Liang K-Y, Zeger SL. Longitudinal data analysis using generalized linear models. Biometrika. 1986;73(1):13–22.
12. Lipsitz SR, Fitzmaurice GM, Orav EJ, Laird NM. Performance of generalized estimating equations in practical situations. Biometrics. 1994;50:270–8.
13. Lipsitz SR, Kim K, Zhao L. Analysis of repeated categorical data using generalized estimating equations. Stat Med. 1994;13(11):1149–63.
14. Louppe G, Wehenkel L, Sutera A, Geurts P. Understanding variable importances in forests of randomized trees. Adv Neural Inf Process Syst. 2013;26.
15. McCulloch CE, Searle SR. Generalized, linear, and mixed models. Hoboken: Wiley; 2004.
16. OpenAI. ChatGPT (Mar 14 version) [Large language model]. 2023. https://chat.openai.com/chat.
17. Peter X-KS, Song K. Correlated data analysis: modeling, analytics, and applications. Berlin: Springer; 2007.
18. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, V., Austria. 2014. http://www.R-project.org/.

19. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, Liu PJ, Liu X, Marcus J, Sun M. Scalable and accurate deep learning with electronic health records. NPJ Digit Med. 2018;1(1):18.
20. SAS Institute Inc 2013. SAS/ACCESS® 9.4 Interface to ADABAS: Reference. Cary, N. S. I. I.
21. Stokes ME, Davis CS, Koch GG. Categorical data analysis using SAS. 3rd ed. Cary: SAS Institute Inc; 2012.
22. Taylor SW, Alexander ME. Science, technology, and human factors in fire danger rating: the Canadian experience. Int J Wildland Fire. 2006;15(1):121–35.

Publisher's Note