Int. J. Anal. Appl. (2024), 22:187

International Journal of Analysis and Applications

Development of Two Methods for Estimating High-Dimensional Data in the Case of Multicollinearity and Outliers

Ahmed A. El-Sheikh[1], Mohamed C. Ali[2], Mohamed R. Abonazel[1,*]

[1]Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt

[2]Faculty of Business Administration, Deraya University, Minya, Egypt

*Corresponding author: mabonazel@cu.edu.eg

Abstract. High-dimensional problems involve datasets or models characterized by a substantial number of variables or parameters prevalent across various domains such as statistics, machine learning, optimization, physics, and engineering. Challenges in these scenarios include computational complexity, data sparsity, over-fitting, and the curse of dimensionality. This study introduces two innovative techniques that combine the Random Forest machine learning approach with both the least absolute shrinkage and selection operator and the elastic net, which are statistical methodologies tailored to address high-dimensional challenges. We compared performance evaluations of these hybrid methods against traditional statistical approaches and standalone machine learning methods. The assessment is conducted using goodness-of-fit measures and involves both Monte Carlo simulation and a real-world application. The findings show that the strategies proposed in this study exhibit superior performance compared to conventional approaches when tackling high-dimensional challenges.

1. Introduction

According to the study conducted by [1], advancements in technology across many fields result in the generation of vast quantities of data, comprising millions of samples, instances, and features. The data used in this study are sourced from various domains, including bioinformatics, text mining, and microarray data. These types of data are typically represented as high-

Received Jul. 21, 2024

2020 Mathematics Subject Classification. 62J07.

Key words and phrases. Statistical methods; machine learning methods; LASSO; elastic net; random forests; high-dimensional; dimension reduction.

https://doi.org/10.28924/2291-8639-22-2024-187

ISSN: 2291-8639

dimensional feature vectors. Predicting outcomes in these datasets poses a challenging task within the fields of pattern recognition, bioinformatics, statistical analysis, and machine learning. Computational time and space complexity are both impacted by high-dimensional data during data processing. Typically, most pattern recognition and machine learning methods handle low-dimensional data, which has limitations when confronted with high-dimensional data. In addressing this problem, the utilization of feature selection (FS) assumes a critical role. FS identifies and picks the most relevant characteristics from a large pool of features in high-dimensional data. This process aims to construct a more streamlined model that can achieve higher accuracy in classification. The FS method primarily aims to decrease and eliminate the multidimensional aspect of the data by eliminating irrelevant and redundant information. This process enhances predictive modeling by facilitating improved visualization and comprehension of the data.

[2] proposed an interpretable meta-learning strategy for high-dimensional regression. The elastic net (Enet) algorithm achieves a trade-off between predicting minor effects for a large number of features and significant impacts for a selected selection of features. The proposed approach incorporates a hybrid regularization technique that combines ridge and lasso methods for achieving a balanced regularization effect. Instead of selecting a singular weighting by means of tuning, we aggregate several weightings by employing a stacking approach. The objective was achieved by a method that enhances the ability to make accurate predictions while maintaining the ability to be easily understood and interpreted.

The study conducted by [3] focused on evaluating the predictive efficacy of several advanced multivariate regression techniques. The application utilized clinical and genomic data to make predictions for a wide range of motor and non-motor symptoms observed in patients diagnosed with Parkinson's disease. The researchers concluded that the utilization of stacked multivariate regression, along with their respective alterations, represents a feasible approach for forecasting interrelated outcomes.

[4] proposed two approaches for analysis: the integration of neural networks (NN) with the least absolute shrinkage and selection operator (LASSO) and the coupling of NN with random forests (RF). The performance of conventional approaches, namely ordinary least squares and feed forward NN, was assessed alongside two developed methods through the utilization of Monte Carlo simulation and a real-world application using air quality data in Italy. The results indicated that the approaches provided in this study exhibited superior performance compared to the standard methods.

[5] made enhancements to the random forest algorithm and introduced a novel technique referred to as post-selection boosting RF (PBRF). This technique integrates the RF and LASSO methods, allowing for the dynamic generation of decision trees based on input samples to

produce prediction results without requiring a predetermined number of decision trees for final prediction. In the interim, we ascertain the efficacy of the suggested algorithm in enhancing the performance of the model by conducting simulation tests and analyzing real-world data.

A group of researchers explored the utilization of RF for handling imbalanced data. [6] conducted an extensive empirical assessment of RF concerning imbalanced data. Additionally, RF was employed for variable selection purposes. [7] suggested a heuristic approach for variable selection, relying on data-driven thresholds for decision-making. Meanwhile, [8] introduced a novel method rooted in permutation tests' theoretical framework, meeting specific statistical criteria. Addressing RF uncertainty emerged as a significant research area, with [9] using jackknife and infinitesimal jackknife methods to estimate RF predictors' variance, yielding practical insights. Furthermore, [10] utilized U-statistics to compute limiting distributions and confidence intervals for predictions.

[11]. Several robust estimators were devised to mitigate the impact of atypical data and multicollinearity effects. Initially, a method called ridge least-trimmed squares was discussed. Subsequently, a nonlinear integer programming problem was proposed, utilizing a penalization approach. The tabu search heuristic algorithm was employed to solve the presented optimization problem, which was characterized by its complexity and difficulty. In addition, the robust generalized cross-validation criterion was utilized to identify the most suitable ridge parameter. Our theoretical talks were supported by computationally studying a simulated example and two real-world datasets.

[12] proposed two mixed-integer nonlinear optimization models that can serve as reliable estimators in the presence of both outliers and multicollinearity in the dataset. The models are constructed using penalization methods that metaheuristic algorithms can successfully solve. These schemes can down-weight or disregard atypical data and multicollinearity effects. We confirm that our models offer computational advantages in terms of the flop count. We also employ a robust ridge methodology. Ultimately, three authentic data sets are scrutinized to evaluate the effectiveness of the suggested methodologies.

[13] devised multiple penalized mixed-integer nonlinear programming models for application in high-dimensional regression analysis. The provided matrix approximations possess uncomplicated structures, resulting in reduced computational expenses for the models. Furthermore, the models can be efficiently solved using metaheuristic methods. Numerical tests are conducted to elucidate the performance of the suggested approaches on both simulated and real-world datasets with high dimensions.

In their study, [14] discussed the limitations of classical methods when analyzing high-dimensional data. They subsequently introduced and explained contemporary and widely used approaches for regression analysis of high-dimensional data, such as principal component analysis and penalized methods. Ultimately, a simulation study and analysis of real-world data are conducted to implement and contrast the methodologies above in datasets with a large number of dimensions.

[15] introduced a method for estimating high-dimensional multicollinear data that can be utilized as an alternative. This usage provides a continuous estimation, encompassing the ridge estimator as a specific instance. They analyzed the asymptotic performance of the system as the dimension, denoted by p, approaches infinity while keeping the value of n unchanged. Subject to some minor regularity criteria, the researchers establish the consistency of the proposed estimator and determine its asymptotic features. Several Monte Carlo simulation experiments are conducted to assess their performance, with the aim of analyzing a genetic dataset with high dimensionality.

In their study, [16] sought to enhance the RF algorithm by incorporating suitable penalized regression techniques. Specifically, they aimed to refine the PBRF algorithm through the application of Enet regression. The most efficient method described in this study is referred to as Reducing and Aggregating RF Trees by Enet (RARTEN). The method that has been introduced comprises three distinct steps. The initial stage involves the utilization of the RF algorithm as a predictive model. In the subsequent stage, the Enet technique, which serves as a form of penalized regression, is employed to decrease the number of trees and enhance the performance of both the RF and PBRF models. In the final stage, the chosen trees are consolidated. The statistical performance criteria are utilized to evaluate the outcomes acquired from both the real data and the Monte Carlo simulation. The findings of the simulation study indicate that the Randomized Average Response Tree Ensemble (RARTEN) enhances the precision of both the conventional RF and Wang's proposed method. Specifically, the RARTEN achieves reductions of 7%, 5%, and 8.5% in the linear, nonlinear, and noisy models, respectively. Furthermore, this approach exhibits a substantial decrease in comparison to alternative penalized regression techniques. Furthermore, the empirical findings of our study demonstrate that the strategy suggested herein yields a decrease of nearly 16%, thus affirming the soundness of the proposed model.

The subsequent sections of this work are structured as follows: Section 2 presents the methodology employed in this study. Section 3 discusses the suggested approaches. Section 4 provides an overview of the Monte Carlo simulation study. Section 5 presents the real-data application. Finally, Section 6 concludes this study.

2. Methodology

Firstly, the applicable shrinkage approach was utilized to handle the data. Subsequently, the selected variables were incorporated into the analysis. This paper will provide a brief discussion on the use of shrinkage methods and the RF regression framework for RF trees.

LASSO Regression

One of the penalization techniques proposed by [17] is the LASSO method. It has gained significant popularity in the field of high-dimensional data analysis after the Ridge regression method. The LASSO method can be formulated as an optimization problem, where the optimal value is determined by including the sum of the absolute values of the regression coefficients in the loss function. This method is widely favored for its ability to do variable selection and shrinking simultaneously. The LASSO technique cannot only estimate the coefficients but also produces a coefficient vector with sparsity. LASSO can be characterized as a variant of Ridge regression that employs distinct penalized functions [18]. T& study employs the LASSO approach as a first step for selecting independent variables. The selected variables are subsequently utilized as inputs for the RF method. Additionally, LASSO is employed to reduce the number of RF trees. The accuracy of prediction is enhanced through the process of picking a subset of trees.

One limitation of this method is that the maximum number of trees that can be selected is constrained by the number of samples. It is not feasible to select more trees than the available samples. Suppose there is (X, Y) a dataset so that $X = \left(x_1, \cdots, x_p\right)'$is the independent variable and Y is the dependent variable. The LASSO estimator uses the $\ell_1$norm penalty to obtain an optimal b for the following optimization problem.

$$\hat{\beta}(\lambda) = \underset{\beta}{\arg min} \left(\frac{\|Y - X\beta\|_2^2}{n} + \lambda\|\beta\|_1\right) \tag{2.1}$$

where $\|Y - X\beta\|_2^2$=$\sum_{i=1}^{n}(y_i - (X\beta)_i)^2$, $\|\beta\|_1 = \sum_{j=1}^{p}\left|\beta_j\right|$ and where $\lambda \geq 0$ is a penalty parameter. The estimator has the property that it does variable selection in the sense that $\hat{\beta}(\lambda) = 0$ for some j's (depending on the choice of λ) and $\hat{\beta}_j(\lambda)$can be thought as a shrunken least squares estimator; hence, the name LASSO. LASSO estimator is available in the R package glmnet [19].

Enet regression

While ridge regression is known for shrinking the coefficients of variables without eliminating any variables, and LASSO regression may both shrink variables and choose the most impactful ones simultaneously, it is important to note that these methods may not always be suitable, as discussed in the preceding sections. Thus, [20] proposed a robust approach known as Enet regression, which effectively combines the strengths of both the LASSO and Ridge methods. The Enet is a statistical regularization technique that combines the principles of Ridge regression, which utilizes the ℓ_2-norm, and LASSO regression, which employs the ℓ_1-norm, in order to minimize the loss function. The primary objective of Enet regression is to effectively minimize the coefficients to zero while simultaneously constructing a model that is based on the non-zero coefficients. Certain regression coefficients exhibit a precise value of zero and can be eliminated from the model. The Enet addresses the constraints associated with the LASSO and Ridge methods, namely the restriction of features during variable selection and the risk of overfitting when dealing with a substantial number of predictor variables, respectively. The present study utilizes the Enet technique as a first stage in the process of choosing independent variables. The chosen variables are later employed as inputs for the RF technique. Moreover, the Enet technique is utilized in order to decrease the number of RF trees. The procedure of selecting a subset of trees contributes to the improvement of prediction accuracy. Despite picking a greater number of trees, it exhibits superior performance compared to the LASSO method.

A double penalization using a combination of the $l_1$and l_2-penalties has been proposed by [20]:

$$\hat{\beta}(\lambda_1, \lambda_2) = \underset{\beta}{\arg min} \left(\frac{\|Y - X\beta\|_2^2}{n} + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \right), \tag{2.2}$$

where $\lambda_1, \lambda_2 > 0$ are two regularization parameters and $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$. [20] called the estimator in (2.2) the "naive Enet". Enet estimator is available in the R package "*glmnet*" [19], [21].

RF algorithm

The RF algorithm is a type of ensemble learning method, originally proposed by [22], that involves the creation of M decision trees using the bagging technique. Parallel tree generation is a capability that distinguishes it from boosting, which necessitates sequential generation. The algorithm in question can be employed for both regression and classification tasks. In regression and classification, the prediction and classification tasks include utilizing the mean of trees and the majority of votes, respectively. The RF algorithm employs a framework that bears resemblance to decision trees, wherein the constituent decision trees within the RF are constructed by considering distinct random partitions. To clarify, the mtry predictor is chosen as a potential separator candidate with a value that is smaller than the total number of predictors, denoted as p. In regression tasks, it is commonly set as mtry = p/3, while in classification tasks, it is typically defined as mtry =√p. The R package "randomForest" [23] provides the implementation of RF regression. Figure 1 displays the structure of the RF. The stages involved in constructing a RF, as depicted in the figure, are outlined as follows: [24]

1. The process of generating Bootstrap datasets $(D_1, \cdots, D_M)$ employed to create multiple datasets from the original D dataset.
2. Generate tree structures based on the Bootstrap dataset.

3. Produce a set of M trees. $T_1, \cdots, T_M$
4. Retrieve M expected trees $T_1(z), \cdots, T_M(z)$
5. The final prediction for the entire set of M trees is as follows:

A. Regression $\bar{y} = \frac{1}{M}\sum_{i=1}^{M} T_i(z)$

B. Classification $T(z)$ = majority vote $\{T_i(z)\}_{i=1}^{M}$

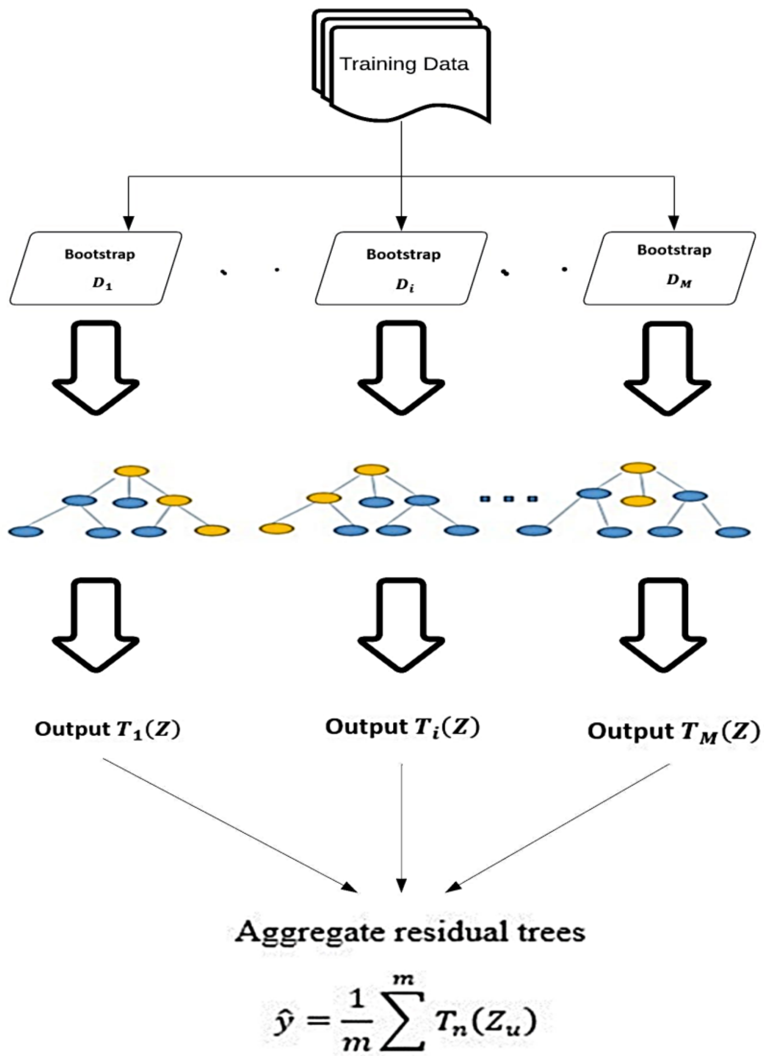

Fig 1: Basic structure of RF

3. Proposed Methods

This section presents two innovative approaches that combine the LASSO and Enet methods with the RF algorithms. The main aim of this integration is to improve the level of congruence, specifically for datasets with a high number of dimensions, in contrast to utilizing RF in isolation. The present study builds upon prior research conducted by [4], who put forth the integration of LASSONN as well as RFNN. Additionally, the work of [24] is referenced, wherein they introduced a novel methodology termed PBRF. The objective of this strategy is to enhance the efficacy of the RF algorithm through the integration of the LASSO method.

Method 1: LASSOPBRF

Step 1: Beginning with the LASSO model

Step 2. The procedure for variable selection in the LASSO model entails the identification and retention of a subset of variables that are considered to be the most pertinent and impactful in forecasting the desired outcome.

Step 3. The selected variables are entered into the RF algorithm.

Step 4. The RF model is employed as a predictive tool.

Step 5. The utilization of the LASSO aims to reduce the number of trees and improve the performance of the RF algorithm.

Step 6. The selected trees are assembled collectively.

Method 2: **EnetRARTEN**

Step 1. The discourse will begin by scrutinizing the Enet paradigm.

Step 2. The procedure for variable selection in the Enet model entails the identification and retention of a subset of variables that are the most pertinent and impactful in forecasting the desired outcome.

Step 3. The selected variables are entered into the RF algorithm.

Step 4: The RF model is utilized as a prediction instrument.

Step 5. The primary objective of utilizing Enet is to reduce the tree count and improve the efficacy of RF.

Step 6. The selected trees have been combined.

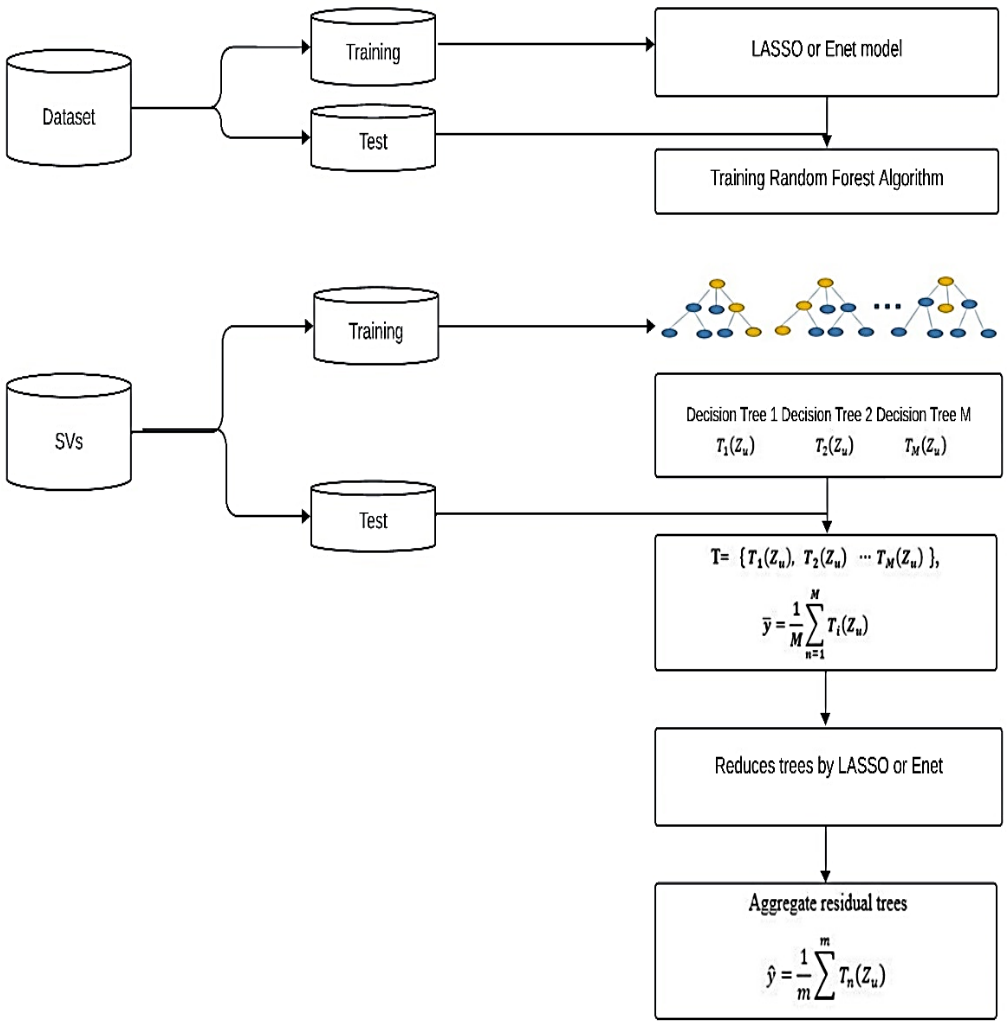

Fig 2: The theoretical underpinning of the suggested methodology

4. Monte Carlo Simulation Study

The primary aim of this work was to conduct a comparative analysis of conventional statistical estimators, namely Enet and LASSO, and a machine learning approach called RF, along with newly introduced estimators such as LASSOPBRF and EnetRARTEN. The analysis was conducted using a Monte Carlo simulation like [25] and [26]. The simulation was conducted using R software version 4, and multiple simulation components were employed to assess the efficacy of the estimators under various conditions (see Table 1). The independent variables utilized in this study were obtained from previous works by [27], [28] and [29]. These variables were generated from a multivariate normal distribution with a mean vector of zero and a covariance matrix denoted as Σ_x. The diagonal elements of Σ_x were assigned a value of 1, whereas the off-diagonal elements were assigned correlation coefficients ρ_x of 0.30, 0.80, 0.85, and 0.90 [30], which indicate the correlation between the independent variables. The errors observed in the study were obtained using a standard normal distribution with outlier rates (OR). Employing two OR, notably 10% and 15%, as reported by [31], [32], [33], [34], respectively. Furthermore, the random forest methodology was utilized, employing different numbers of trees (Ntree), specifically 200, 500, 800, and 1000. The simulation was performed with sample sizes of 58, 100, 250, and 500. It incorporated four independent variables: 100, 450, 500, and 1000. According to [35], it may be observed that. The regression parameters were assigned values of 0.5 and 0.001, as reported by [36] in their reference. The present work aimed to create and employ the LASSOPBRF and EnetRARTEN techniques to provide a comparative analysis. The design of the simulation is depicted in Figure 3, which presents a flowchart.

Table 1 Simulation Factors

Factors	Values
$\boldsymbol{\rho_x}$	0.30,0.80,0.85 and 0.90
OR	0.10 and 0.15
n	58,100,250 and 500
P	100,450,500 and 1000
Ntree	200,500,800 and 1000

Simulation Process

Step 1: Generating Independent Variables

Generating independent variables from a multivariate normal distribution with a correlation between them.

Step 2: Generating Error Terms.

Generate an error term from a standard normal distribution with different ORs (see [37]) of 0.10 and 0.15.

Step 3: Initial Parameters for Regression Coefficients Set initial parameters for β_1 = 0.5 and β_2= 0.001 Step 4: Constructing a High-Dimensional Regression Model (see [38], [39], [40], [41]) Build a regression model using the generated independent variables, error term, and initial parameters. Step 5: Estimation Methods Utilize various estimation methods such as LASSO, Enet, RF, LASSOPBRF, and EnetRARTEN. Each of these methods handles high-dimensional data and regression differently.

Step 6: Calculating criteria mean square error (MSE) and root mean square error (RMSE)

After applying these estimation methods, MSE and RMSE were calculated for each method. These criteria assess the performance of the models in predicting the dependent variable, measuring the average squared differences between predicted and actual values. MSE measures the average squared difference between predicted values and true values. In a Monte Carlo simulation, you would typically have multiple iterations or simulated datasets. For each iteration, suppose you have n observations, and the predicted values are denoted as $\hat{Y}_i$ and the true values are denoted as Y_i for i=1,2,⋯ ,n. The MSE for a single simulation iteration is calculated as:

$$\text{MSE} = \frac{1}{n}\left(\hat{Y}_i - Y_i\right)^2 \tag{4.1}$$

To calculate the MSE over multiple iterations in a Monte Carlo simulation, you would sum up the MSE values obtained in each iteration and divide them by the total number of iterations. RMSE is the square root of MSE and gives a measure of the average magnitude of the error in the same units as the response variable.

$$\text{RMSE} = \sqrt{MSE} \tag{4.2}$$

In this current study, two separate metrics for assessing the accuracy of the estimators were utilized: MSE and RMSE. In addition, each strategy yields data regarding the number of selected variables (#SVs) and the number of selected trees (#STs). The findings of the simulation study, denoted as the simulation results (SRs), were presented in Table 2-13 and Table S1-S12 in the appendix, which displayed data pertaining to a sample size of n = 58, 100, 250, and 500, as well as the number of independent variables p = 100, 450, 500, and 1000. The results comprised several correlation coefficients, two OR, and four Ntree: (0.30, 0.80, 0.85, and 0.90), (0.10 and 0.15), and (200, 500, 800, and 1000).

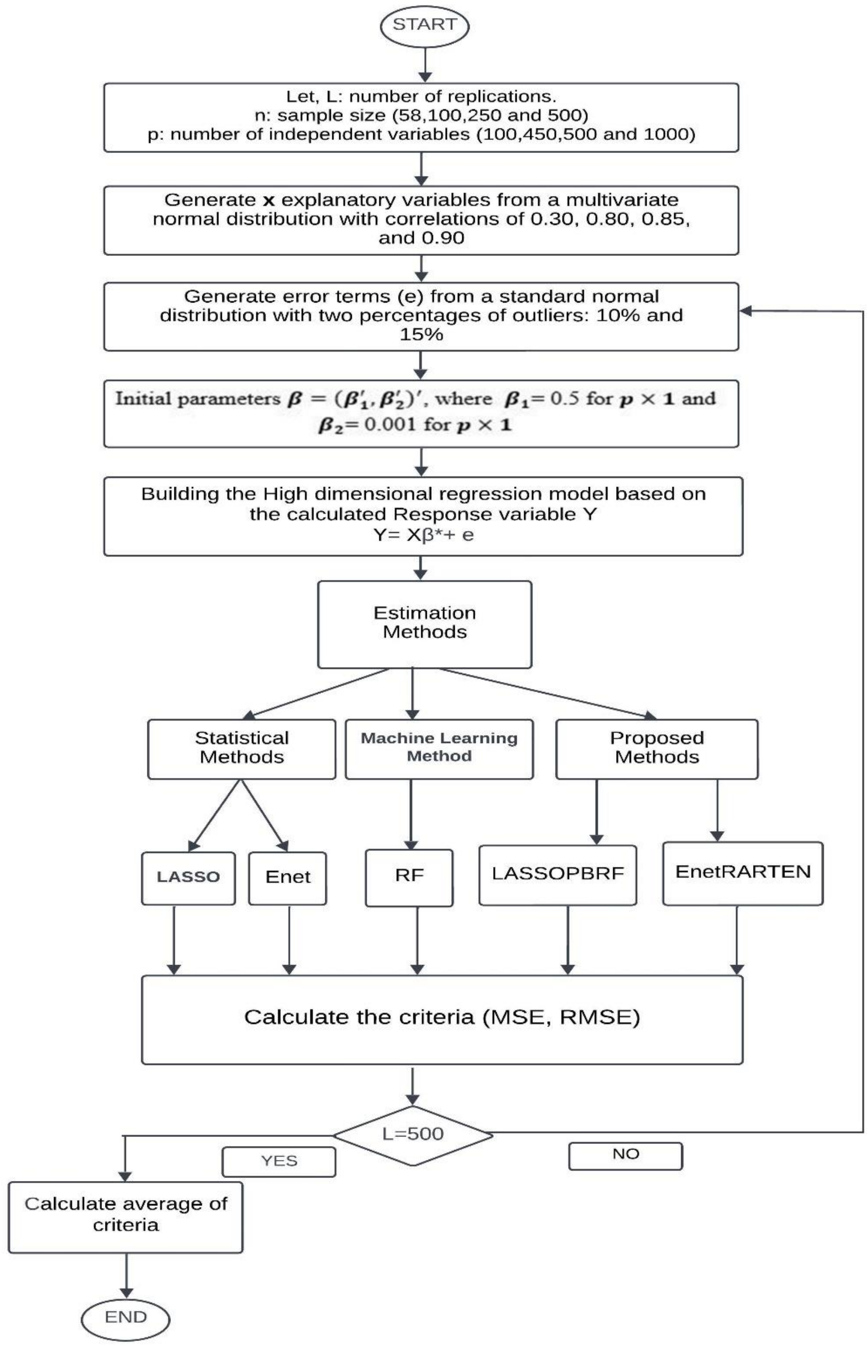

Fig 3: The flowchart depicting the simulation process.

Table 2: SRs when n=58, P=450, $\rho_x = 0.90$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR =10%					
200	LASSO	297.188	17.239	-	139
	Enet	250.767	15.835	-	239
	RF	179.154	13.384	200	450
	LASSOPBRF	47.739	6.909	121	139
	EnetRARTEN	46.618	6.827	165	293
500	LASSO	308.384	17.56	-	139
	Enet	262.758	16.209	-	296
	RF	175.006	13.228	500	450
	LASSOPBRF	49.021	7.001	149	139
	EnetRARTEN	45.926	6.776	350	296
800	LASSO	305.128	17.467	-	137
	Enet	293.066	17.119	-	301
	RF	179.953	13.414	800	450
	LASSOPBRF	49.491	7.035	166	137
	EnetRARTEN	46.044	6.785	518	301
1000	LASSO	314.413	17.731	-	137
	Enet	297.79	17.256	-	300
	RF	181.144	13.459	1000	450
	LASSOPBRF	49.85	7.06	175	137
	EnetRARTEN	46.206	6.797	572	300
OR=15%					
200	LASSO	322.029	17.945	-	139
	Enet	263.456	16.231	-	295
	RF	178.763	13.37	200	450
	LASSOPBRF	48.302	6.95	123	139
	EnetRARTEN	47.157	6.867	168	295
500	LASSO	302.869	17.403	-	140
	Enet	248.826	15.774	-	295
	RF	182.89	13.523	500	450
	LASSOPBRF	48.834	6.988	151	140
	EnetRARTEN	46.612	6.827	347	295
800	LASSO	308.787	17.572	-	138
	Enet	305.66	17.483	-	306
	RF	183.696	13.553	800	450
	LASSOPBRF	49.497	7.035	167	138
	EnetRARTEN	46.708	6.834	475	306
1000	LASSO	332.73	18.24	-	140
	Enet	285.081	16.884	-	295
	RF	170.786	13.068	1000	450
	LASSOPBRF	49.819	7.058	175	140
	EnetRARTEN	46.329	6.806	608	295

Table 3: SRs when n=58, P=450, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	443.573	21.061	-	135
	Enet	327.542	18.098	-	288
	RF	230.59	15.185	200	450
	LASSOPBRF	53.899	7.341	121	135
	EnetRARTEN	51.685	7.189	163	288
500	LASSO	436.711	20.897	-	137
	Enet	361.911	19.023	-	287
	RF	227.865	15.095	500	450
	LASSOPBRF	54.768	7.4	147	137
	EnetRARTEN	50.412	7.1	348	287
800	LASSO	435.853	20.877	-	133
	Enet	402.219	20.055	-	299
	RF	225.712	15.023	800	450
	LASSOPBRF	55.673	7.461	162	133
	EnetRARTEN	50.835	7.129	472	299
1000	LASSO	402.715	20.067	-	135
	Enet	363.095	19.055	-	296
	RF	221.846	14.894	1000	450
	LASSOPBRF	55.885	7.475	173	135
	EnetRARTEN	50.946	7.137	545	296
OR= 15%					
200	LASSO	435.227	20.862	-	134
	Enet	411.981	20.297	-	297
	RF	239.156	15.464	200	450
	LASSOPBRF	53.903	7.341	122	134
	EnetRARTEN	51.516	7.177	165	297
500	LASSO	425.279	20.622	-	133
	Enet	393.555	19.838	-	294
	RF	234.53	15.314	500	450
	LASSOPBRF	54.914	7.41	147	133
	EnetRARTEN	51.222	7.157	338	294
800	LASSO	421.755	20.536	-	136
	Enet	348.322	18.663	-	290
	RF	232.517	15.248	800	450
	LASSOPBRF	55.437	7.445	159	136
	EnetRARTEN	50.8	7.127	495	290
1000	LASSO	434.959	20.855	-	135
	Enet	345.728	18.593	-	286
	RF	216.464	14.712	1000	450
	LASSOPBRF	56.072	7.488	167	135
	EnetRARTEN	51.173	7.153	594	286

Table 4: SRs when n=100, P=100, $\rho_x = 0.90$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	8.732	2.955	-	34
	Enet	26.003	5.099	-	82
	RF	11.995	3.463	200	100
	LASSOPBRF	1.942	1.393	97	34
	EnetRARTEN	1.908	1.381	167	82
500	LASSO	8.54	2.922	-	34
	Enet	27.41	5.235	-	82
	RF	11.757	3.428	500	100
	LASSOPBRF	1.925	1.387	137	34
	EnetRARTEN	1.868	1.367	352	82
800	LASSO	8.586	2.93	-	34
	Enet	24.303	4.929	-	81
	RF	11.867	3.444	800	100
	LASSOPBRF	1.966	1.402	156	34
	EnetRARTEN	1.878	1.37	489	81
1000	LASSO	8.41	2.9	-	34
	Enet	24.964	4.996	-	82
	RF	11.761	3.429	1000	100
	LASSOPBRF	1.935	1.391	167	34
	EnetRARTEN	1.85	1.36	581	82
OR=15%					
200	LASSO	10.878	3.298	-	32
	Enet	32.974	5.742	-	82
	RF	13.614	3.689	200	100
	LASSOPBRF	2.249	1.499	98	32
	EnetRARTEN	2.193	1.48	167	82
500	LASSO	10.804	3.286	-	32
	Enet	31.732	5.633	-	80
	RF	13.496	3.673	500	100
	LASSOPBRF	2.197	1.482	138	32
	EnetRARTEN	2.135	1.461	368	80
800	LASSO	10.775	3.282	-	32
	Enet	32.181	5.672	-	81
	RF	13.663	3.696	800	100
	LASSOPBRF	2.217	1.489	161	32
	EnetRARTEN	2.145	1.464	506	81
1000	LASSO	11.048	3.323	-	32
	Enet	32.389	5.691	-	81
	RF	13.27	3.642	1000	100
	LASSOPBRF	2.205	1.485	171	32
	EnetRARTEN	2.136	1.461	587	81

Table 5: SRs when n=100, P=100, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	9	3	-	38
	Enet	32.927	5.738	-	83
	RF	13.016	3.607	200	100
	LASSOPBRF	2.115	1.454	102	38
	EnetRARTEN	2.074	1.44	172	83
500	LASSO	8.971	2.995	-	38
	Enet	29.317	5.414	-	81
	RF	12.463	3.53	500	100
	LASSOPBRF	2.057	1.434	142	38
	EnetRARTEN	2.005	1.416	349	81
800	LASSO	9.016	3.002	-	38
	Enet	30.91	5.559	-	82
	RF	12.614	3.551	800	100
	LASSOPBRF	2.065	1.437	164	38
	EnetRARTEN	2.006	1.416	485	82
1000	LASSO	9.09	3.014	-	38
	Enet	31.745	5.634	-	84
	RF	12.826	3.581	1000	100
	LASSOPBRF	2.09	1.445	175	38
	EnetRARTEN	2.023	1.422	577	84
OR=15%					
200	LASSO	11.568	3.401	-	36
	Enet	37.133	6.093	-	82
	RF	15.273	3.908	200	100
	LASSOPBRF	2.377	1.541	102	36
	EnetRARTEN	2.337	1.528	171	82
500	LASSO	11.677	3.417	-	36
	Enet	36.706	6.058	-	80
	RF	15.12	3.888	500	100
	LASSOPBRF	2.346	1.531	145	36
	EnetRARTEN	2.311	1.52	348	80
800	LASSO	11.881	3.447	-	35
	Enet	40.665	6.376	-	83
	RF	14.906	3.86	800	100
	LASSOPBRF	2.367	1.538	166	35
	EnetRARTEN	2.308	1.519	517	83
1000	LASSO	12.076	3.475	-	35
	Enet	39.252	6.265	-	81
	RF	14.458	3.802	1000	100
	LASSOPBRF	2.333	1.527	179	35
	EnetRARTEN	2.293	1.514	555	81

Table 6: SRs when n=100, P=500, $\rho_x = 0.90$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	302.842	17.402	-	180
	Enet	481.547	21.944	-	344
	RF	241.34	15.535	200	500
	LASSOPBRF	36.09	6.007	156	180
	EnetRARTEN	35.776	5.981	170	344
500	LASSO	294.468	17.16	-	179
	Enet	438.314	20.935	-	324
	RF	242.021	15.557	500	500
	LASSOPBRF	35.683	5.973	201	179
	EnetRARTEN	34.498	5.873	352	324
800	LASSO	296.848	17.229	-	180
	Enet	473.245	21.754	-	343
	RF	235.379	15.342	800	500
	LASSOPBRF	35.682	5.973	220	180
	EnetRARTEN	34.621	5.884	511	343
1000	LASSO	308.037	17.55	-	181
	Enet	475.214	21.799	-	338
	RF	222.854	14.928	1000	500
	LASSOPBRF	35.547	5.962	229	181
	EnetRARTEN	34.656	5.886	601	338
OR=15%					
200	LASSO	307.703	17.541	-	180
	Enet	431.102	20.763	-	325
	RF	227.24	15.074	200	500
	LASSOPBRF	36.145	6.012	156	180
	EnetRARTEN	35.893	5.991	172	325
500	LASSO	306.76	17.514	-	180
	Enet	447.764	21.16	-	336
	RF	231.139	15.203	500	500
	LASSOPBRF	36.081	6.006	199	180
	EnetRARTEN	35.039	5.919	354	336
800	LASSO	295.663	17.194	-	178
	Enet	464.414	21.55	-	333
	RF	259.9	16.121	800	500
	LASSOPBRF	36.088	6.007	220	178
	EnetRARTEN	34.803	5.899	528	333
1000	LASSO	305.232	17.47	-	180
	Enet	462.902	21.515	-	335
	RF	237.77	15.419	1000	500
	LASSOPBRF	35.898	5.991	229	180
	EnetRARTEN	34.559	5.878	596	335

Table 7: SRs when n=100, P=500, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	400.549	20.013	-	177
	Enet	688.353	26.236	-	344
	RF	275.189	16.588	200	500
	LASSOPBRF	39.74	6.303	158	177
	EnetRARTEN	39.367	6.274	173	344
500	LASSO	402.27	20.057	-	177
	Enet	684.351	26.16	-	345
	RF	256.635	16.02	500	500
	LASSOPBRF	39.648	6.297	202	177
	EnetRARTEN	38.086	6.171	367	345
800	LASSO	406.623	20.164	-	176
	Enet	604.228	24.581	-	328
	RF	265.181	16.284	800	500
	LASSOPBRF	39.697	6.3	220	176
	EnetRARTEN	37.951	6.16	497	328
1000	LASSO	412.425	20.308	-	177
	Enet	701.249	26.481	-	346
	RF	263.435	16.23	1000	500
	LASSOPBRF	40.123	6.334	231	177
	EnetRARTEN	38.359	6.193	632	346
OR=15%					
200	LASSO	402.133	20.053	-	177
	Enet	634.995	25.199	-	329
	RF	276.751	16.635	200	500
	LASSOPBRF	40.21	6.341	159	177
	EnetRARTEN	39.563	6.289	174	329
500	LASSO	393.016	19.825	-	176
	Enet	653.031	25.554	-	334
	RF	294.198	17.152	500	500
	LASSOPBRF	39.782	6.307	204	176
	EnetRARTEN	38.313	6.19	357	334
800	LASSO	405.292	20.131	-	176
	Enet	697.443	26.409	-	346
	RF	260.909	16.152	800	500
	LASSOPBRF	40.193	6.339	222	176
	EnetRARTEN	38.378	6.195	505	346
1000	LASSO	403.866	20.096	-	177
	Enet	665.138	25.79	-	342
	RF	276.042	16.614	1000	500
	LASSOPBRF	39.959	6.321	228	177
	EnetRARTEN	38.029	6.166	609	342

Table 8: SRs when n=100, P=1000, $\rho_x = 0.90$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	3565.049	59.708	-	970
	Enet	1983.852	44.54	-	638
	RF	826.965	28.757	200	1000
	LASSOPBRF	166.249	12.893	195	970
	EnetRARTEN	167.698	12.949	173	638
500	LASSO	3683.887	60.695	-	975
	Enet	1928.831	43.918	-	623
	RF	914.308	30.237	500	1000
	LASSOPBRF	162.495	12.747	437	975
	EnetRARTEN	164.248	12.815	365	623
800	LASSO	3551.094	59.591	-	977
	Enet	1788.971	42.296	-	593
	RF	847.715	29.115	800	1000
	LASSOPBRF	161.643	12.713	608	977
	EnetRARTEN	164.368	12.82	509	593
1000	LASSO	3541.304	59.508	-	975
	Enet	1798.726	42.411	-	605
	RF	926.376	30.436	1000	1000
	LASSOPBRF	162.262	12.738	718	975
	EnetRARTEN	164.152	12.812	642	605
OR=15%					
200	LASSO	3632.82	60.272	-	977
	Enet	1938.682	44.03	-	620
	RF	927.066	30.447	200	1000
	LASSOPBRF	167.434	12.939	195	977
	EnetRARTEN	167.884	12.957	172	620
500	LASSO	3612.021	60.1	-	974
	Enet	1899.24	43.58	-	613
	RF	893.155	29.885	500	1000
	LASSOPBRF	163.123	12.771	437	974
	EnetRARTEN	165.236	12.854	360	613
800	LASSO	3581.256	59.843	-	975
	Enet	1939.093	44.035	-	631
	RF	863.664	29.388	800	1000
	LASSOPBRF	162.036	12.729	608	975
	EnetRARTEN	163.813	12.798	539	631
1000	LASSO	3622.049	60.183	-	977
	Enet	1988.673	44.594	-	651
	RF	853.157	29.208	1000	1000
	LASSOPBRF	162.553	12.749	693	977
	EnetRARTEN	164.986	12.844	594	651

Table 9: SRs when n=100, P=1000, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	5040.912	70.999	-	975
	Enet	2820.093	53.104	-	639
	RF	1070.905	32.724	200	1000
	LASSOPBRF	180.204	13.424	196	975
	EnetRARTEN	181.78	13.482	174	639
500	LASSO	4975.152	70.534	-	978
	Enet	2624.633	51.231	-	607
	RF	1053.161	32.452	500	1000
	LASSOPBRF	176.089	13.269	419	978
	EnetRARTEN	177.555	13.324	367	607
800	LASSO	4976.801	70.546	-	976
	Enet	2600.776	50.997	-	607
	RF	1085.889	32.952	800	1000
	LASSOPBRF	175.429	13.244	573	976
	EnetRARTEN	176.479	13.284	553	607
1000	LASSO	5015.59	70.82	-	977
	Enet	2966.625	54.466	-	655
	RF	1005.624	31.711	1000	1000
	LASSOPBRF	175.8	13.258	666	977
	EnetRARTEN	176.694	13.292	636	655
OR=15%					
200	LASSO	5047.877	71.048	-	978
	Enet	2875.285	53.621	-	639
	RF	1039.723	32.244	200	1000
	LASSOPBRF	181.609	13.476	196	978
	EnetRARTEN	182.663	13.515	174	639
500	LASSO	5040.924	70.999	-	977
	Enet	2643.354	51.413	-	620
	RF	1117.461	33.428	500	1000
	LASSOPBRF	175.839	13.26	422	977
	EnetRARTEN	178.749	13.369	356	620
800	LASSO	5017.64	70.835	-	974
	Enet	2688.079	51.846	-	634
	RF	1039.473	32.24	800	1000
	LASSOPBRF	176.008	13.266	576	974
	EnetRARTEN	177.792	13.333	521	634
1000	LASSO	5078.959	71.266	-	976
	Enet	2704.21	52.002	-	616
	RF	1085.147	32.941	1000	1000
	LASSOPBRF	175.748	13.257	663	976
	EnetRARTEN	177.905	13.338	629	616

Table 10: SRs when n=250, P=500, $\rho_x = 0.90$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	170.584	13.06	-	272
	Enet	343.623	18.537	-	366
	RF	80.386	8.965	200	500
	LASSOPBRF	10.811	3.288	141	272
	EnetRARTEN	10.48	3.237	174	366
500	LASSO	171.611	13.1	-	275
	Enet	351.667	18.752	-	379
	RF	75.056	8.663	500	500
	LASSOPBRF	10.055	3.171	256	275
	EnetRARTEN	9.715	3.116	387	379
800	LASSO	164.254	12.816	-	269
	Enet	355.199	18.846	-	376
	RF	75.989	8.717	800	500
	LASSOPBRF	9.99	3.16	295	269
	EnetRARTEN	9.551	3.09	563	376
1000	LASSO	172.983	13.152	-	273
	Enet	352.275	18.768	-	376
	RF	76.449	8.743	1000	500
	LASSOPBRF	9.95	3.154	324	273
	EnetRARTEN	9.51	3.083	673	376
OR=15%					
200	LASSO	138.612	11.773	-	245
	Enet	376.873	19.413	-	381
	RF	81.516	9.028	200	500
	LASSOPBRF	11.282	3.359	121	245
	EnetRARTEN	10.831	3.291	171	381
500	LASSO	143.316	11.971	-	250
	Enet	371.364	19.27	-	381
	RF	80.289	8.96	500	500
	LASSOPBRF	10.653	3.263	216	250
	EnetRARTEN	10.053	3.17	392	381
800	LASSO	139.938	11.829	-	247
	Enet	351.045	18.736	-	377
	RF	77.02	8.776	800	500
	LASSOPBRF	10.523	3.244	245	247
	EnetRARTEN	9.853	3.139	562	377
1000	LASSO	138.91	11.786	-	247
	Enet	346.992	18.627	-	366
	RF	78.729	8.872	1000	500
	LASSOPBRF	10.543	3.247	264	247
	EnetRARTEN	9.898	3.146	667	366

Table 11: SRs when n=250, P=500, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	262.086	16.189	-	284
	Enet	465.346	21.571	-	371
	RF	109.735	10.475	200	500
	LASSOPBRF	14.719	3.836	157	284
	EnetRARTEN	14.485	3.806	178	371
500	LASSO	258.321	16.072	-	287
	Enet	495.85	22.267	-	381
	RF	107.287	10.357	500	500
	LASSOPBRF	13.867	3.723	295	287
	EnetRARTEN	13.573	3.684	390	381
800	LASSO	260.14	16.128	-	287
	Enet	503.395	22.436	-	382
	RF	109.264	10.452	800	500
	LASSOPBRF	13.698	3.701	345	287
	EnetRARTEN	13.275	3.643	569	382
1000	LASSO	256.947	16.029	-	285
	Enet	469.513	21.668	-	376
	RF	108.192	10.401	1000	500
	LASSOPBRF	13.727	3.705	364	285
	EnetRARTEN	13.132	3.623	702	376
OR=15%					
200	LASSO	225.359	15.011	-	267
	Enet	495.011	22.248	-	373
	RF	110.742	10.523	200	500
	LASSOPBRF	15.121	3.888	145	267
	EnetRARTEN	14.722	3.836	180	373
500	LASSO	218.924	14.796	-	266
	Enet	502.686	22.42	-	380
	RF	106.683	10.328	500	500
	LASSOPBRF	14.357	3.789	254	266
	EnetRARTEN	13.798	3.714	392	380
800	LASSO	224.083	14.969	-	268
	Enet	498.561	22.328	-	375
	RF	107.114	10.349	800	500
	LASSOPBRF	14.216	3.77	299	268
	EnetRARTEN	13.624	3.691	561	375
1000	LASSO	218.82	14.792	-	266
	Enet	459.676	21.44	-	368
	RF	110.782	10.525	1000	500
	LASSOPBRF	14.21	3.769	317	266
	EnetRARTEN	13.616	3.69	678	368

Table 12: SRs when n=500, P=1000, $\rho_x = 0.90$

Tree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	1321.594	36.353	-	665
	Enet	1838.355	42.876	-	750
	RF	273.101	16.525	200	1000
	LASSOPBRF	38.595	6.212	178	665
	EnetRARTEN	39.191	6.26	169	750
500	LASSO	1307.498	36.159	-	669
	Enet	1611.448	40.142	-	722
	RF	282.13	16.796	500	1000
	LASSOPBRF	36.13	6.01	425	669
	EnetRARTEN	36.42	6.034	387	722
800	LASSO	1330.144	36.471	-	671
	Enet	2143.279	46.295	-	799
	RF	243.636	15.608	800	1000
	LASSOPBRF	35.628	5.968	618	671
	EnetRARTEN	35.833	5.986	600	799
1000	LASSO	1283.096	35.82	-	664
	Enet	2107.65	45.909	-	797
	RF	302.231	17.384	1000	1000
	LASSOPBRF	35.508	5.958	687	664
	EnetRARTEN	35.948	5.995	681	797
OR=15%					
200	LASSO	1160.754	34.069	-	639
	Enet	1995.808	44.674	-	756
	RF	275.029	16.584	200	1000
	LASSOPBRF	39.648	6.296	172	639
	EnetRARTEN	39.668	6.298	173	756
500	LASSO	1162.319	34.092	-	637
	Enet	1962.345	44.298	-	740
	RF	267.891	16.367	500	1000
	LASSOPBRF	36.78	6.064	399	637
	EnetRARTEN	36.796	6.066	401	740
800	LASSO	1142.438	33.799	-	635
	Enet	1971.459	44.401	-	762
	RF	241.095	15.527	800	1000
	LASSOPBRF	36.276	6.023	563	635
	EnetRARTEN	36.399	6.033	607	762
1000	LASSO	1192.556	34.533	-	644
	Enet	2008.23	44.813	-	752
	RF	260.158	16.129	1000	1000
	LASSOPBRF	36.121	6.01	631	644
	EnetRARTEN	36.365	6.03	671	752

Table 13: SRs when n=500, P=1000, $\rho_x = 0.85$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	1861.055	43.139	-	670
	Enet	2655.342	51.53	-	755
	RF	360.52	18.987	200	1000
	LASSOPBRF	49.662	7.047	185	670
	EnetRARTEN	49.751	7.053	178	755
500	LASSO	1871.101	43.256	-	668
	Enet	2569.485	50.69	-	745
	RF	351.04	18.736	500	1000
	LASSOPBRF	45.742	6.763	450	668
	EnetRARTEN	46.027	6.784	409	745
800	LASSO	1909.951	43.702	-	671
	Enet	2522.624	50.225	-	744
	RF	329.483	18.151	800	1000
	LASSOPBRF	45.352	6.734	650	671
	EnetRARTEN	45.895	6.774	610	744
1000	LASSO	1880.454	43.364	-	668
	Enet	3072.312	55.428	-	821
	RF	369.11	19.212	1000	1000
	LASSOPBRF	45.169	c6.72	716	668
	EnetRARTEN	45.413	6.738	738	821
OR=15%					
200	LASSO	1736.702	41.673	-	653
	Enet	2556.775	50.564	-	744
	RF	366.22	19.136	200	1000
	LASSOPBRF	50.055	7.075	180	653
	EnetRARTEN	50.511	7.107	175	744
500	LASSO	1752.195	41.859	-	657
	Enet	2953.814	54.349	-	793
	RF	341.723	18.485	500	1000
	LASSOPBRF	46.456	6.815	440	657
	EnetRARTEN	46.715	6.834	407	793
800	LASSO	1758.659	41.936	-	652
	Enet	2335.699	48.329	-	717
	RF	324.006	18	800	1000
	LASSOPBRF	45.682	6.758	627	652
	EnetRARTEN	46.049	6.785	617	717
1000	LASSO	1802.231	42.452	-	665
	Enet	2826.899	53.168	-	772
	RF	341.067	18.468	1000	1000
	LASSOPBRF	45.646	6.756	712	665
	EnetRARTEN	45.756	6.764	742	772

Based on the data provided in tables 2 and 3, with a sample size of n = 58 and independent variables equal to 450 and considering different rates of correlation (0.85 and 0.90), rates of outliers (10% and 15%), and four different values for Ntrees (200, 500, 800, and 1000), the following conclusions can be drawn: 1. Enet selects more independent variables than LASSO. 2. Enet has a lower minimum MSE and RMSE than LASSO. 3. Random Forest (RF) cannot select independent variables, but it has a lower minimum MSE and RMSE than LASSO and Enet. 4. The two proposed methods are superior to LASSO, Enet, and RF in terms of MSE and RMSE. 5. EnetRARTEN selects a larger number of trees than LASSOPBRF and has a lower minimum MSE and RMSE than all other methods.

Based on the data provided in tables 4 and 5, the study was conducted with a sample size of 100. The independent variables were set at 100, with correlation rates of 0.85 and 0.90. Additionally, two different rates of outliers were considered: 10% and 15%. The study also included four different values for the Ntrees: 200, 500, 800, and 1000. Enet is found to choose a greater number of independent variables compared to LASSO. Additionally, Enet exhibits greater values of MSE and RMSE than LASSO. RF, on the other hand, is unable to select independent variables but still achieves lower MSE and RMSE values than both LASSO and Enet. Therefore, the two proposed methods outperform LASSO, Enet, and RF. Furthermore, EnetRARTEN selects a larger number of trees than LASSOPBRF and demonstrates the lowest MSE and RMSE among all methods.

Based on the data provided in tables 6 and 7, the analysis was conducted using a sample size of 100. The independent variables were set at 500, with correlation rates of 0.85 and 0.90. Additionally, two different rates of outliers were considered: 10% and 15%. The analysis was performed using four different values for Ntrees: 200, 500, 800, and 1000. Enet is found to choose a greater number of independent variables compared to LASSO. Additionally, Enet exhibits greater values of MSE and RMSE than LASSO. RF, on the other hand, is unable to select independent variables but still achieves lower values of minimum MSE and RMSE than both LASSO and Enet. Consequently, the two proposed methods outperform LASSO, Enet, and RF. Furthermore, EnetRARTEN selects a higher number of trees than LASSOPBRF and demonstrates lower values of minimum MSE and RMSE compared to all other methods.

Based on the data provided in tables 8 and 9, the study was conducted with a sample size of 100. The independent variables were set at 1000, with correlation rates of 0.85 and 0.90. Two different rates of outliers, 10% and 15%, were also considered. Additionally, four different values of Ntrees were used: 200, 500, 800, and 1000. Enet is found to choose a greater number of independent variables compared to LASSO. Additionally, Enet exhibits lower values of MSE and RMSE than LASSO. RF, on the other hand, is unable to select independent variables, but it still demonstrates lower MSE and RMSE than both LASSO and Enet. Therefore, the two proposed methods outperform LASSO, Enet, and RF. Furthermore, EnetRARTEN selects a higher number of trees than LASSOPBRF and achieves the lowest MSE and RMSE among all the methods.

Based on the data provided in tables 10 and 11, the study was conducted with a sample size of 250. The independent variables were set at 500, with correlation rates of 0.85 and 0.90. Additionally, two different rates of outliers were considered, namely 10% and 15%. The analysis was performed using four different Ntrees values: 200, 500, 800, and 1000. Enet is found to choose a greater number of independent variables compared to LASSO. Additionally, LASSO exhibits lower minimum MSE and RMSE values than Enet. RF, on the other hand, is unable to select independent variables but still demonstrates lower minimum MSE and RMSE values than both LASSO and Enet. Consequently, the two proposed methods outperform LASSO, Enet, and RF. Furthermore, EnetRARTEN selects a higher number of trees than LASSOPBRF and achieves lower minimum MSE and RMSE values than all other methods.

Based on the data from tables 12 and 13, with a sample size of 500 and independent variables set at 1000, we observed different rates of correlation (0.85 and 0.90) and two rates of outliers (10% and 15%). We also tested four different values for Ntrees: 200, 500, 800, and 1000. Our findings indicate that Enet selects more independent variables than LASSO. Additionally, LASSO has the lowest values for MSE and RMSE compared to Enet. RF, on the other hand, cannot select independent variables but still has lower MSE and RMSE than LASSO and Enet. Overall, the two proposed methods (Enet and RF) outperform LASSO, Enet, and RF in terms of MSE and RMSE. Furthermore, EnetRARTEN selects a larger number of trees compared to LASSOPBRF and also achieves the lowest MSE and RMSE among all the methods.

Overall Conclusions:

RF

- Selected all independent variables regardless of correlation or outliers.

- Showed the minimum MSE and RMSE compared to classical statistical methods (LASSO and Enet).

Enet:

- Demonstrated higher selection of independent variables and numbers of trees than LASSO in various scenarios compared to other methods.
- Had a higher selection of independent variables and trees than Random Forest in the EnetRARTEN case.
- Showed better performance in terms of variable selection compared to LASSO but did not achieve the lowest MSE and RMSE compared to all methods.

LASSO:

- Had a lower selection of independent variables and numbers of trees compared to Elastic Net and Random Forest.
- Did not achieve the lowest MSE and RMSE compared to all methods.

EnetRARTEN:

- Showed a high selection of independent variables and numbers of trees compared to LASSO, Enet, and RF in all cases.
- Achieved minimum MSE and RMSE compared to all other methods.

LASSOPBRF:

- Showed a lower MSE and RMSE compared to RF, LASSO, and Enet.

EnetRARTEN exhibited the lowest MSE and RMSE among all methods.

RF performed consistently well, selecting all independent variables and showing minimal MSE and RMSE compared to classical statistical methods (LASSO and Enet).

It is important to ensure the clarity of the conclusions, especially in terms of methodology and the specifics of the analysis, to maintain accuracy and avoid misinterpretation.

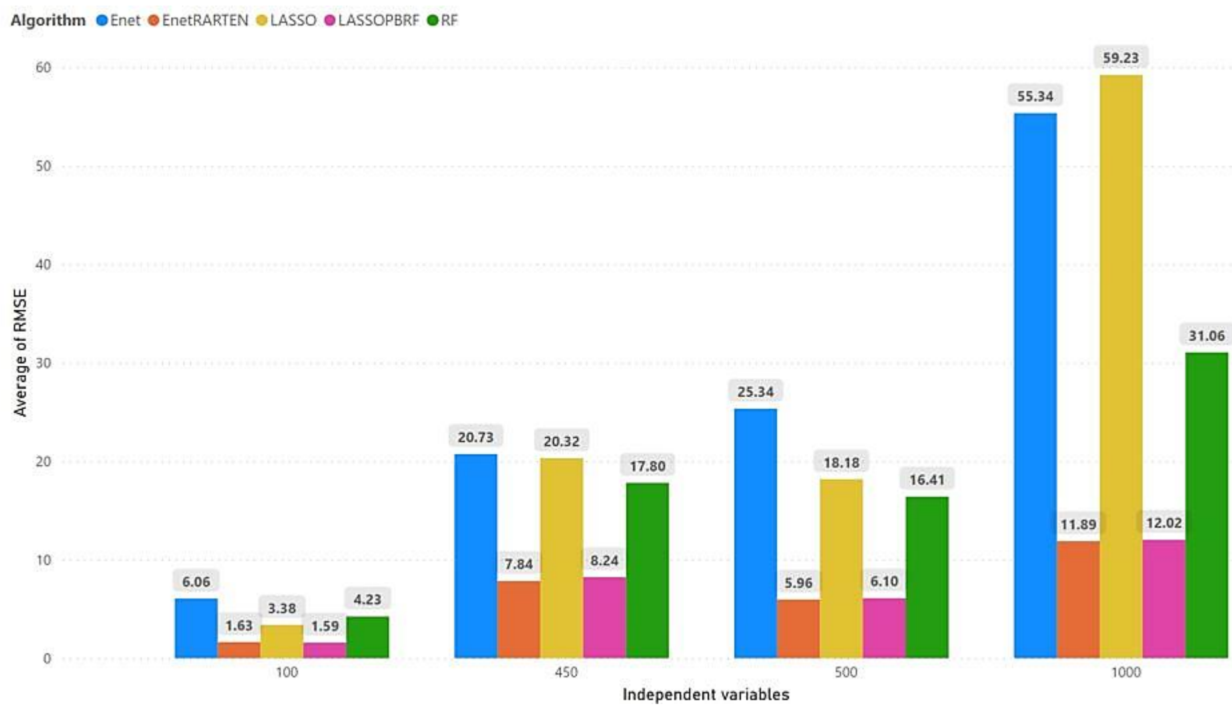

Fig 4: RMSE of methods at different levels of independent variable

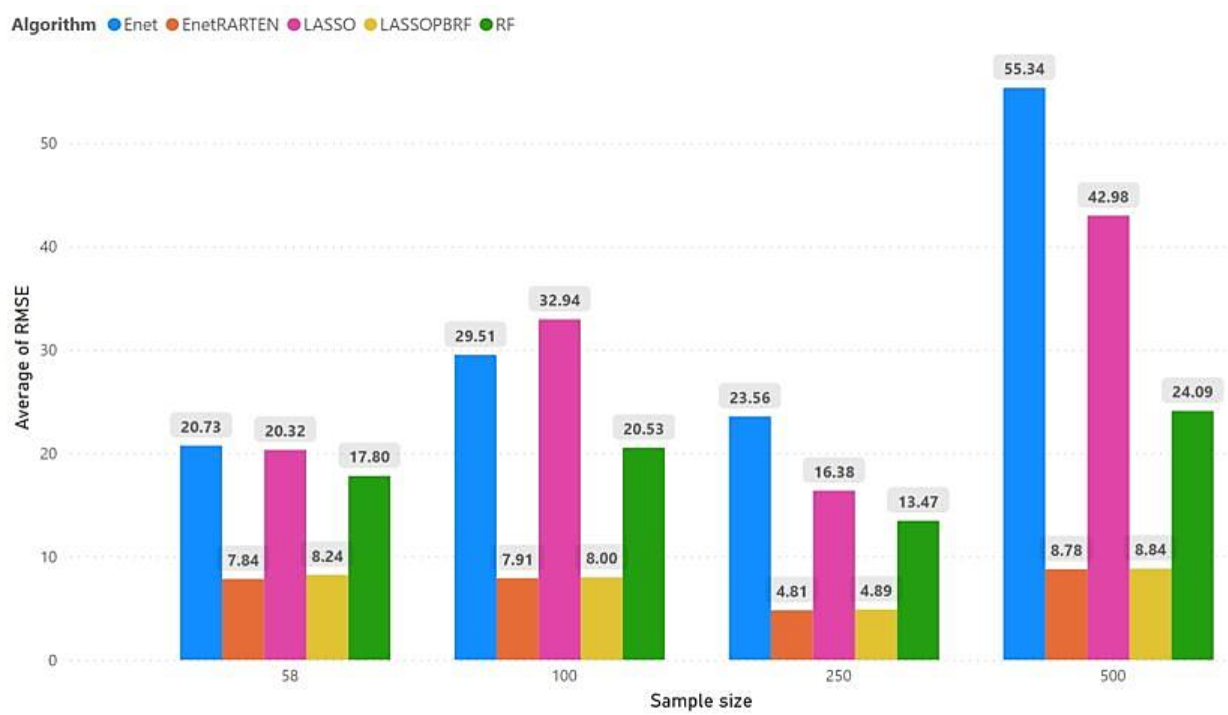

Fig 5: RMSE of methods at different levels of sample size

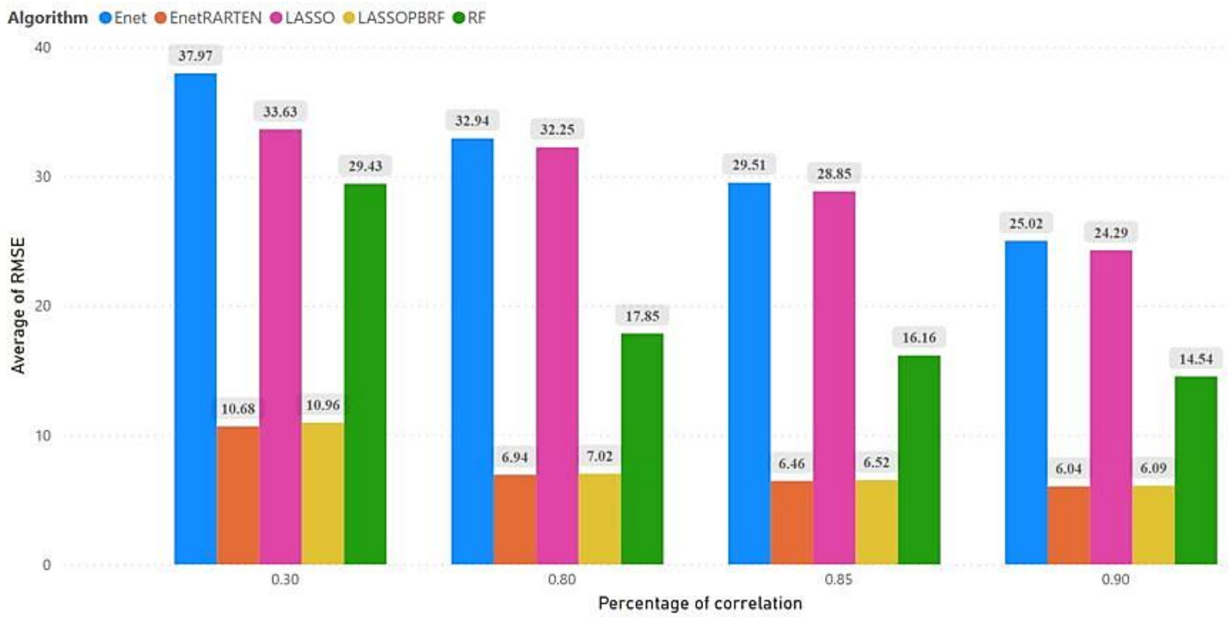

Fig 6: RMSE of methods at different levels of percentage of correlation

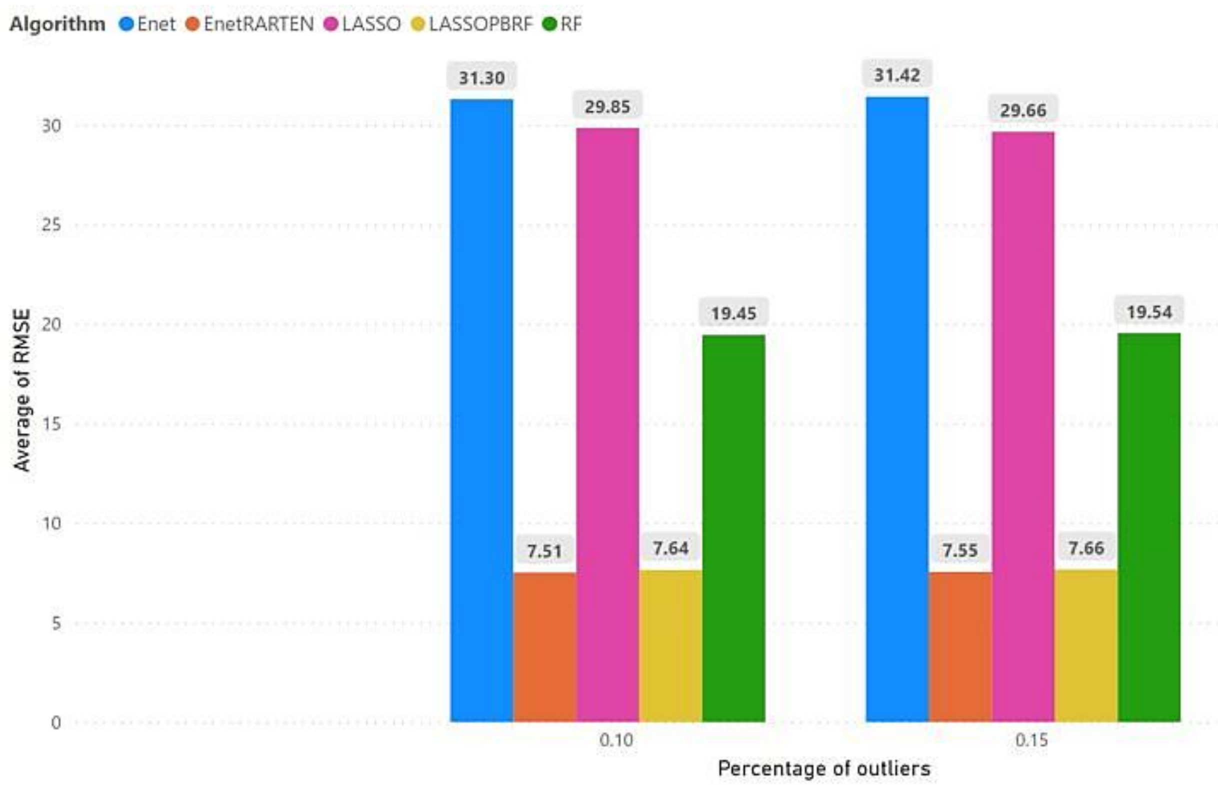

Fig 7: RMSE of methods at two levels of percentage of outliers

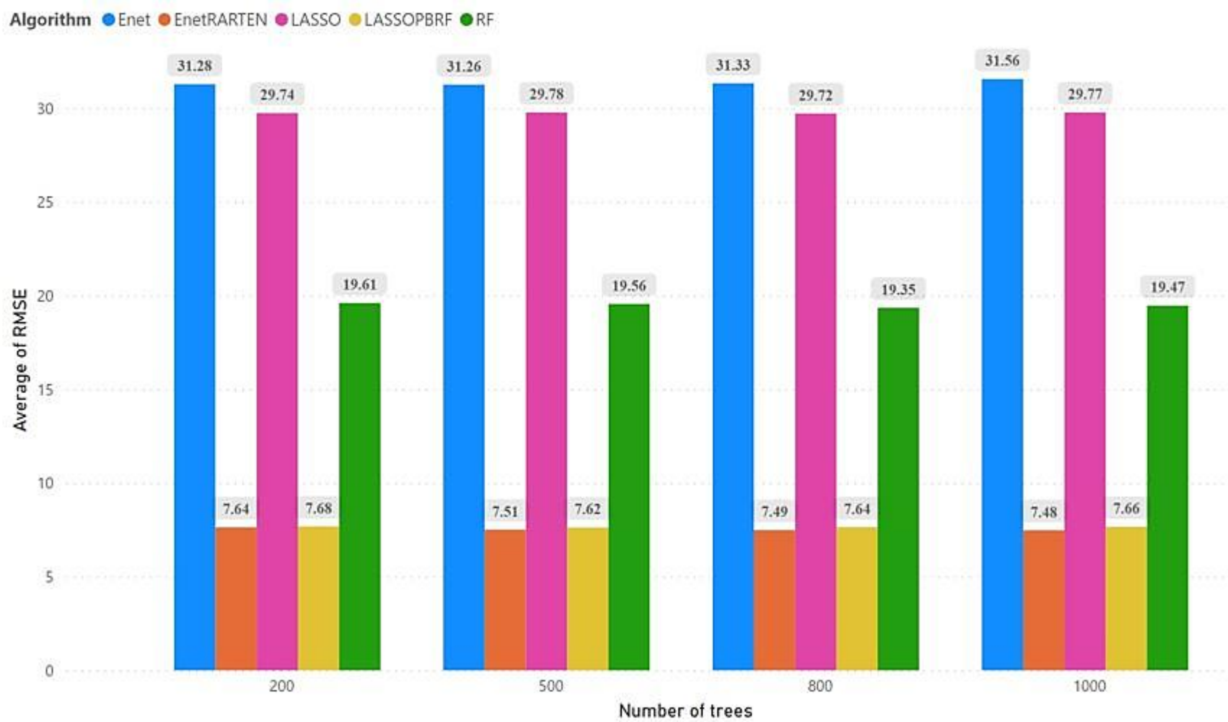

Fig 8: RMSE of methods at different levels of a number of trees

Figure 4 shows that the AMSE of RF is less than that of LASSO and Enet, and the proposed methods EnetRARTEN and LASSOPBRF are better than the classical methods LASSO, Enet, and RF. Finally, EnetRARTEN is better than all methods at any level of independent variables.

Figure 5 shows that the AMSE of RF is less than that of LASSO and Enet, and the proposed methods EnetRARTEN and LASSOPBRF are better than the classical methods LASSO, Enet, and RF. Finally, EnetRARTEN is better than all methods at any level of sample size.

Figure 6 shows that the AMSE of RF is less than that of LASSO and Enet, and the proposed methods EnetRARTEN and LASSOPBRF are better than the classical methods LASSO, Enet, and RF. Finally, EnetRARTEN is better than all methods at any level of percentage correlation.

Figure 7 shows that the AMSE of RF is less than that of LASSO and Enet, and the proposed methods EnetRARTEN and LASSOPBRF are better than the classical methods LASSO, Enet, and RF. Finally, EnetRARTEN is better than all methods at two percentages of the outlier.

Figure 8 shows that the AMSE of RF is less than that of LASSO and Enet, and the proposed methods EnetRARTEN and LASSOPBRF are better than the classical methods LASSO, Enet, and RF. Finally, EnetRARTEN is better than all methods at any value of the number of trees.

Overall Summary:

EnetRARTEN Superiority: EnetRARTEN consistently displayed the minimum RMSE across various parameters, including the sample size, independent variable levels, correlation, outlier levels, and number of trees. This shows EnetRARTEN's robust performance and superiority compared to Enet, LASSO, RF, and LASSOPBRF across diverse conditions and factors in the analysis.

5. Real data application

The data pertaining to a production process were systematically observed during a specified period. [42] employed the data above in their analysis. Four hundred samples were collected for analysis, causing the inclusion of 468 unique independent variables to explain the resultant outcome. To guarantee the maintenance of confidentiality, the data accessible at the URL https://cstat.tuwien.ac.at/data is provided. R-Data has undergone a process of anonymization through the application of centering and scaling techniques. For the sake of simplicity, the time-series nature of the data will not be taken into consideration in the subsequent analysis. A training set comprising randomly picked samples seventy percent of the sample size. Various methods were employed for fitting, and the evaluation was conducted on the remaining 30% of the test data. The primary aim of our investigation was to discover the independent variables that exerted the most substantial influence on the prediction of the dependent variable. In order to accomplish this aim, we used a model or variable-selection method.

Suppose you have a dataset with actual observed values Y_i and corresponding predicted values $\hat{Y}_i$ generated by a model. MSE is calculated by taking the average of the squared differences between predicted and actual values for all data points:

$$\text{MSE} = \frac{1}{n}\left(\hat{Y}_i - Y_i\right)^2, \tag{5.1}$$

where n sample size of the dataset and Y_i are the observed values and $\hat{Y}_i$ are the predicted values generated by a model. The RMSE is calculated as the square root of MSE, allowing for interpretation in the same units as the dependent variable:

$$\text{RMSE} = \sqrt{MSE} \tag{5.2}$$

Table 14 Goodness fit measure for real data application

Ntree	Algorithm	MSE	RMSE	#ST	#SV
200	LASSO	0.545	0.738	-	33
	Enet	0.543	0.737	-	78
	RF	0.475	0.689	200	468
	LASSOPBRF	0.077	0.277	125	33
	EnetRARTEN	0.073	0.27	198	78
500	LASSO	0.576	0.759	-	33
	Enet	0.543	0.737	-	78
	RF	0.483	0.695	500	468
	LASSOPBRF	0.073	0.27	122	33
	EnetRARTEN	0.071	0.267	377	78
800	LASSO	0.571	0.755	-	33
	Enet	0.543	0.737	-	78
	RF	0.487	0.698	800	468
	LASSOPBRF	0.072	0.268	131	33
	EnetRARTEN	0.071	0.266	512	78
1000	LASSO	0.571	0.755	-	33
	Enet	0.543	0.737	-	78
	RF	0.487	0.698	1000	468
	LASSOPBRF	0.073	0.271	117	33
	EnetRARTEN	0.067	0.26	535	78

The findings presented in Table 14 demonstrate that the Enet method outperforms both LASSO and RF in selecting independent variables. Specifically, Enet considers all independent variables and decision trees in its selection process. The proposed methodologies, namely LASSOPBRF and EnetRARTEN, exhibited superior performance compared to the conventional statistical approaches (Enet and LASSO) as well as the RF method, as evidenced by their lower MSE and RMSE values. Both the LASSOPBRF and EnetRARTEN methods were employed to determine the smallest number of independent variables and trees. Among the Enet, LASSO, RF, and LASSOPBRF models, EnetRARTEN had the lowest MSE and RMSE. The EnetRARTEN model incorporated a greater number of independent variables and trees compared to the LASSOPBRF method.

6. Conclusions

The phenomenon known as the curse of dimensionality poses a substantial obstacle in the context of challenges characterized by a high number of dimensions. As the number of dimensions increases, the volume of the space experiences exponential growth, leading to a decrease in data density. The presence of sparsity in a dataset has the potential to result in overfitting, a phenomenon in which a model has strong performance on the training data but struggles to effectively generalize to unseen data. To accomplish this objective, the study conducted a comparative analysis of the performance of two proposed approaches, namely LASSOPBRF and EnetRARTEN, in comparison to conventional statistical methods (Enet and LASSO) and a machine learning method known as RF. This analysis was carried out using both a Monte Carlo simulation and a real-world application that utilized a production dataset. In summarizing the principal findings of the simulation study, it was seen that the EnetRARTEN approach had superior goodness of fit in comparison to the other methods. (2) EnetRARTEN had superior performance compared to all other methods, as evidenced by its attainment of the lowest values for MSE and RMSE. (3) In contrast to LASSOPBRF and EnetRARTEN, RF picked a greater number of variables and decision trees. Based on the obtained results, it can be inferred that the EnetRARTEN technique is the suggested approach due to its consistent demonstration of lower MSE and RMSE values in comparison to the Enet, LASSO, RF, and LASSOPBRF methods. This indicates the usefulness of the EnetRARTEN method in effectively addressing the challenges posed by multicollinearity and outlier influences. In conclusion, the research emphasizes the significance of employing high-dimensional methodologies, particularly EnetRARTEN, to enhance the precision of statistical models when confronted with intricate datasets that encompass multicollinearity and outlier effects. The analysis of the real-world application revealed several significant findings. Firstly, the RF method employed all independent variables in its analysis, utilizing what is known as the full model. In contrast, both LASSOPBRF and EnetRARTEN showed higher values for metrics such as MSE and RMSE. Moreover, the EnetRARTEN method demonstrated superior performance when compared to Enet, LASSO, RF, and LASSOPBRF, achieving the lowest values of MSE and RMSE.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Appendix

Table S1: SRs when n=58, P=450, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	499.858	22.357	-	129
	Enet	412.707	20.315	-	282
	RF	290.939	17.056	200	450
	LASSOPBRF	60.946	7.806	118	129
	EnetRARTEN	58.037	7.618	166	282
500	LASSO	508.441	22.548	-	129
	Enet	430.856	20.757	-	284
	RF	286.677	16.931	500	450
	LASSOPBRF	62.157	7.884	144	129
	EnetRARTEN	57.155	7.56	342	284
800	LASSO	532.312	23.071	-	129
	Enet	483.059	21.978	-	298
	RF	286.446	16.924	800	450
	LASSOPBRF	63.651	7.978	152	129
	EnetRARTEN	56.941	7.545	475	298
1000	LASSO	520.052	22.804	-	129
	Enet	445.454	21.105	-	291
	RF	261.52	16.171	1000	450
	LASSOPBRF	64.186	8.011	160	129
	EnetRARTEN	56.846	7.539	584	291
OR=15%					
200	LASSO	518.475	22.77	-	129
	Enet	446.275	21.125	-	288
	RF	298.954	17.29	200	450
	LASSOPBRF	60.914	7.804	117	129
	EnetRARTEN	58.185	7.627	165	288
500	LASSO	510.749	22.599	-	131
	Enet	482.137	21.957	-	303
	RF	275.894	16.61	500	450
	LASSOPBRF	61.954	7.871	143	131
	EnetRARTEN	56.994	7.549	325	303
800	LASSO	513.874	22.668	-	130
	Enet	443.485	21.059	-	290
	RF	280.345	16.743	800	450
	LASSOPBRF	63.159	7.947	156	130
	EnetRARTEN	57.117	7.557	473	290
1000	LASSO	532.79	23.082	-	130
	Enet	463.811	21.536	-	284
	RF	278.376	16.684	1000	450
	LASSOPBRF	64.413	8.025	159	130
	EnetRARTEN	58.164	7.626	533	284

Table S2: SRs when n=58, P=450, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	411.447	20.284	-	107
	Enet	663.265	25.753	-	284
	RF	668.279	25.851	200	450
	LASSOPBRF	108.62	10.422	100	107
	EnetRARTEN	99.235	9.961	162	284
500	LASSO	413.63	20.337	-	106
	Enet	652.349	25.541	-	276
	RF	644.809	25.393	500	450
	LASSOPBRF	111.211	10.545	117	106
	EnetRARTEN	95.133	9.753	317	276
800	LASSO	406.022	20.15	-	107
	Enet	692.483	26.315	-	290
	RF	655.851	25.609	800	450
	LASSOPBRF	115.27	10.736	125	107
	EnetRARTEN	96.248	9.81	430	290
1000	LASSO	407.708	20.191	-	106
	Enet	677.662	26.031	-	289
	RF	658.888	25.668	1000	450
	LASSOPBRF	115.178	10.732	129	106
	EnetRARTEN	95.606	9.777	530	289
OR=15%					
200	LASSO	405.216	20.129	-	107
	Enet	608.675	24.671	-	274
	RF	701.013	26.476	200	450
	LASSOPBRF	108.193	10.401	100	107
	EnetRARTEN	99.009	9.95	162	274
500	LASSO	397.348	19.933	-	107
	Enet	699.823	26.454	-	296
	RF	683.089	26.135	500	450
	LASSOPBRF	112.61	10.611	117	107
	EnetRARTEN	96.908	9.844	325	296
800	LASSO	404.875	20.121	-	106
	Enet	668.65	25.858	-	280
	RF	684.866	26.169	800	450
	LASSOPBRF	115.53	10.748	125	106
	EnetRARTEN	94.837	9.738	459	280
1000	LASSO	408.296	20.206	-	106
	Enet	708.982	26.626	-	290
	RF	681.828	26.111	1000	450
	LASSOPBRF	115.58	10.75	129	106
	EnetRARTEN	95.103	9.752	519	290

Table S3: SRs when n=100, P=100, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	10.038	3.168	-	41
	Enet	36.314	6.026	-	83
	RF	14.416	3.796	200	100
	LASSOPBRF	2.286	1.512	104	41
	EnetRARTEN	2.257	1.502	172	83
500	LASSO	9.733	3.119	-	41
	Enet	33.462	5.784	-	83
	RF	14.822	3.85	500	100
	LASSOPBRF	2.228	1.492	145	41
	EnetRARTEN	2.213	1.487	349	83
800	LASSO	9.78	3.127	-	41
	Enet	38.882	6.235	-	85
	RF	14.437	3.799	800	100
	LASSOPBRF	2.236	1.495	165	41
	EnetRARTEN	2.187	1.479	508	85
1000	LASSO	9.893	3.145	-	41
	Enet	38.079	6.17	-	84
	RF	14.476	3.804	1000	100
	LASSOPBRF	2.207	1.485	181	41
	EnetRARTEN	2.175	1.474	579	84
OR=15%					
200	LASSO	12.285	3.505	-	38
	Enet	42.738	6.537	-	83
	RF	16.895	4.11	200	100
	LASSOPBRF	2.536	1.592	105	38
	EnetRARTEN	2.507	1.583	174	83
500	LASSO	11.9397	3.4554	-	38
	Enet	31.5659	5.6183	-	82
	RF	15.7006	3.9624	500	100
	LASSOPBRF	2.4726	1.5724	148	38
	EnetRARTEN	2.472	1.5722	349	82
800	LASSO	11.762	3.429	-	38
	Enet	44.823	6.695	-	83
	RF	16.063	4.007	800	100
	LASSOPBRF	2.486	1.576	170	38
	EnetRARTEN	2.48	1.575	498	83
1000	LASSO	12.153	3.486	-	39
	Enet	44.905	6.701	-	85
	RF	16.705	4.087	1000	100
	LASSOPBRF	2.447	1.564	178	39
	EnetRARTEN	2.439	1.561	582	85

Table S4: SRs when n=100, P=100, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	13.647	3.694	-	51
	Enet	41.696	6.457	-	85
	RF	31.679	5.628	200	100
	LASSOPBRF	3.953	1.988	93	51
	EnetRARTEN	4.512	2.124	171	85
500	LASSO	13.386	3.658	-	52
	Enet	45.061	6.712	-	84
	RF	31.864	5.644	500	100
	LASSOPBRF	3.484	1.866	114	52
	EnetRARTEN	4.362	2.088	330	84
800	LASSO	14.075	3.751	-	51
	Enet	43.173	6.57	-	83
	RF	32.391	5.691	800	100
	LASSOPBRF	3.315	1.82	121	51
	EnetRARTEN	4.339	2.083	434	83
1000	LASSO	14.219	3.77	-	51
	Enet	44.939	6.703	-	83
	RF	32.036	5.66	1000	100
	LASSOPBRF	3.228	1.796	126	51
	EnetRARTEN	4.3	2.073	490	83
OR=15%					
200	LASSO	16.703	4.087	-	48
	Enet	45.286	6.729	-	82
	RF	34.168	5.845	200	100
	LASSOPBRF	4.042	2.01	89	48
	EnetRARTEN	4.739	2.176	175	82
500	LASSO	16.386	4.048	-	48
	Enet	47.939	6.923	-	81
	RF	34.499	5.873	500	100
	LASSOPBRF	3.548	1.883	106	48
	EnetRARTEN	4.591	2.142	323	81
800	LASSO	16.855	4.105	-	48
	Enet	47.998	6.928	-	81
	RF	33.106	5.753	800	100
	LASSOPBRF	3.408	1.846	116	48
	EnetRARTEN	4.534	2.129	433	81
1000	LASSO	16.43	4.053	-	48
	Enet	47.786	6.912	-	82
	RF	33.828	5.816	1000	100
	LASSOPBRF	3.324	1.823	119	48
	EnetRARTEN	4.578	2.139	478	82

Table S5: SRs when n=100, P=500, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	478.391	21.872	-	172
	Enet	786.95	28.052	-	332
	RF	325.412	18.039	200	500
	LASSOPBRF	45.733	6.762	159	172
	EnetRARTEN	44.707	6.686	176	332
500	LASSO	468.2	21.638	-	171
	Enet	859.176	29.312	-	341
	RF	331.965	18.22	500	500
	LASSOPBRF	45.526	6.747	202	171
	EnetRARTEN	43.646	6.606	350	341
800	LASSO	478.857	21.882	-	172
	Enet	884.055	29.733	-	347
	RF	318.795	17.854	800	500
	LASSOPBRF	45.997	6.782	220	172
	EnetRARTEN	43.609	6.603	519	347
1000	LASSO	485.505	22.034	-	172
	Enet	805.106	28.374	-	337
	RF	313.992	17.719	1000	500
	LASSOPBRF	45.941	6.777	229	172
	EnetRARTEN	43.256	6.576	599	337
OR=15%					
200	LASSO	487.27	22.074	-	173
	Enet	818.441	28.608	-	334
	RF	319.499	17.875	200	500
	LASSOPBRF	45.318	6.732	159	173
	EnetRARTEN	44.997	6.708	175	334
500	LASSO	479.443	21.896	-	172
	Enet	780.243	27.932	-	332
	RF	318.116	17.835	500	500
	LASSOPBRF	45.776	6.765	203	172
	EnetRARTEN	43.724	6.612	372	332
800	LASSO	479.339	21.893	-	172
	Enet	765.38	27.665	-	327
	RF	325.634	18.045	800	500
	LASSOPBRF	45.882	6.773	221	172
	EnetRARTEN	43.521	6.597	502	327
1000	LASSO	473.195	21.753	-	171
	Enet	829.763	28.805	-	338
	RF	330.61	18.182	1000	500
	LASSOPBRF	46.03	6.784	228	171
	EnetRARTEN	43.383	6.586	594	338

Table S6: SRs when n=100, P=500, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	419.378	20.479	-	137
	Enet	1046.329	32.347	-	327
	RF	755.306	27.483	200	500
	LASSOPBRF	99.946	9.997	144	137
	EnetRARTEN	96.741	9.836	175	327
500	LASSO	422.718	20.56	-	137
	Enet	1043.168	32.298	-	329
	RF	763.128	27.624	500	500
	LASSOPBRF	102.699	10.134	179	137
	EnetRARTEN	94.553	9.723	350	329
800	LASSO	430.728	20.754	-	137
	Enet	1112.722	33.357	-	330
	RF	757.185	27.517	800	500
	LASSOPBRF	103.826	10.189	194	137
	EnetRARTEN	93.149	9.651	494	330
1000	LASSO	410.599	20.263	-	138
	Enet	1081.133	32.88	-	329
	RF	743.289	27.263	1000	500
	LASSOPBRF	104.952	10.244	200	138
	EnetRARTEN	93.278	9.658	566	329
OR=15%					
200	LASSO	419.173	20.474	-	136
	Enet	1110.111	33.318	-	328
	RF	771.038	27.768	200	500
	LASSOPBRF	100.17	10.009	144	136
	EnetRARTEN	97.453	9.872	175	328
500	LASSO	426.02	20.64	-	137
	Enet	994.529	31.536	-	319
	RF	765.572	27.668	500	500
	LASSOPBRF	101.989	10.099	180	137
	EnetRARTEN	94.171	9.704	350	319
800	LASSO	426.26	20.646	-	137
	Enet	1089.636	33.009	-	327
	RF	750.056	27.387	800	500
	LASSOPBRF	104.263	10.21	194	137
	EnetRARTEN	93.26	9.657	476	327
1000	LASSO	424.593	20.605	-	137
	Enet	1074.683	32.782	-	333
	RF	765.614	27.669	1000	500
	LASSOPBRF	104.827	10.238	201	137
	EnetRARTEN	92.705	9.628	576	333

Table S7: SRs when n=100, P=1000, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	6339.119	79.618	-	975
	Enet	3723.027	61.016	-	668
	RF	1187.731	34.463	200	1000
	LASSOPBRF	198.352	14.083	196	975
	EnetRARTEN	198.988	14.106	174	668
500	LASSO	6456.204	80.35	-	976
	Enet	3394.143	58.259	-	610
	RF	1264.367	35.557	500	1000
	LASSOPBRF	192.066	13.858	409	976
	EnetRARTEN	194.318	13.939	371	610
800	LASSO	6289.279	79.304	-	976
	Enet	3405.385	58.355	-	632
	RF	1173.856	34.261	800	1000
	LASSOPBRF	193.627	13.915	543	976
	EnetRARTEN	194.018	13.929	540	632
1000	LASSO	6281.299	79.254	-	976
	Enet	3312.643	57.555	-	611
	RF	1271.024	35.651	1000	1000
	LASSOPBRF	193.536	13.911	610	976
	EnetRARTEN	194.238	13.936	628	611
OR=15%					
200	LASSO	6398.12	79.988	-	977
	Enet	3473.805	58.938	-	630
	RF	1260.374	35.501	200	1000
	LASSOPBRF	197.709	14.06	196	977
	EnetRARTEN	199.248	14.115	177	630
500	LASSO	6336.328	79.601	-	975
	Enet	3329.283	57.699	-	621
	RF	1210.776	34.796	500	1000
	LASSOPBRF	193.803	13.921	405	975
	EnetRARTEN	195.165	13.97	367	621
800	LASSO	6426.608	80.166	-	977
	Enet	3530.922	59.421	-	626
	RF	1233.966	35.127	800	1000
	LASSOPBRF	192.879	13.888	536	977
	EnetRARTEN	195	13.964	514	626
1000	LASSO	6290.025	79.309	-	975
	Enet	3253.726	57.041	-	603
	RF	1266.156	35.583	1000	1000
	LASSOPBRF	193.714	13.918	622	975
	EnetRARTEN	195.585	13.985	617	603

Table S8: SRs when n=100, P=1000, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	8345.816	91.355	-	714
	Enet	4428.716	66.548	-	610
	RF	3039.101	55.128	200	1000
	LASSOPBRF	406.205	20.154	176	714
	EnetRARTEN	399.146	19.978	168	610
500	LASSO	8290.178	91.05	-	715
	Enet	4555.003	67.49	-	619
	RF	3092.877	55.613	500	1000
	LASSOPBRF	413.709	20.339	212	715
	EnetRARTEN	391.164	19.777	356	619
800	LASSO	8257.499	90.87	-	726
	Enet	4291.87	65.512	-	585
	RF	2953.834	54.349	800	1000
	LASSOPBRF	417.997	20.444	230	726
	EnetRARTEN	387.653	19.688	496	585
1000	LASSO	8382.053	91.553	-	730
	Enet	4565.024	67.564	-	613
	RF	2938.105	54.204	1000	1000
	LASSOPBRF	419.922	20.492	240	730
	EnetRARTEN	386.632	19.662	586	613
OR=15%					
200	LASSO	8315.567	91.189	-	723
	Enet	4550.122	67.454	-	613
	RF	2974.945	54.543	200	1000
	LASSOPBRF	403.265	20.081	177	723
	EnetRARTEN	399.295	19.982	169	613
500	LASSO	8409.82	91.705	-	717
	Enet	4580.982	67.682	-	604
	RF	3041.809	55.152	500	1000
	LASSOPBRF	414.515	20.359	211	717
	EnetRARTEN	390.629	19.764	357	604
800	LASSO	8322.242	91.226	-	725
	Enet	4513.337	67.181	-	599
	RF	2919.553	54.032	800	1000
	LASSOPBRF	420.007	20.494	231	725
	EnetRARTEN	388.114	19.7	503	599
1000	LASSO	8427.911	91.803	-	720
	Enet	4246.089	65.162	-	586
	RF	2971.641	54.512	1000	1000
	LASSOPBRF	420.139	20.497	238	720
	EnetRARTEN	382.262	19.551	605	586

Table S9: SRs when n=250, P=500, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	328.647	18.128	-	290
	Enet	605.426	24.605	-	378
	RF	144.281	12.011	200	500
	LASSOPBRF	18.949	4.353	168	290
	EnetRARTEN	18.576	4.31	182	378
500	LASSO	336.229	18.336	-	291
	Enet	621.573	24.931	-	389
	RF	141.136	11.88	500	500
	LASSOPBRF	17.852	4.225	321	291
	EnetRARTEN	17.574	4.192	390	389
800	LASSO	338.211	18.39	-	289
	Enet	663.803	25.764	-	389
	RF	140.009	11.832	800	500
	LASSOPBRF	17.838	4.223	368	289
	EnetRARTEN	17.435	4.175	561	389
1000	LASSO	340.175	18.443	-	292
	Enet	656.827	25.628	-	390
	RF	137.616	11.731	1000	500
	LASSOPBRF	17.789	4.217	394	292
	EnetRARTEN	17.229	4.15	689	390
OR=15%					
200	LASSO	298.767	17.284	-	278
	Enet	555.842	23.576	-	369
	RF	146.505	12.103	200	500
	LASSOPBRF	19.419	4.406	158	278
	EnetRARTEN	19.094	4.369	180	369
500	LASSO	306.787	17.515	-	278
	Enet	591.305	24.316	-	373
	RF	145.428	12.059	500	500
	LASSOPBRF	18.356	4.284	289	278
	EnetRARTEN	17.924	4.233	403	373
800	LASSO	299.441	17.304	-	278
	Enet	606.153	24.62	-	379
	RF	139.938	11.829	800	500
	LASSOPBRF	18.301	4.278	339	278
	EnetRARTEN	17.646	4.2	570	379
1000	LASSO	299.006	17.291	-	278
	Enet	623.189	24.963	-	381
	RF	138.078	11.75	1000	500
	LASSOPBRF	18.255	4.272	357	278
	EnetRARTEN	17.601	4.195	684	381

Table S10: SRs when n=250, P=500, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	381.773	19.539	-	269
	Enet	781.938	27.963	-	374
	RF	525.887	22.932	200	500
	LASSOPBRF	68.781	8.293	180	269
	EnetRARTEN	68.154	8.255	192	374
500	LASSO	397.238	19.93	-	269
	Enet	824.353	28.711	-	377
	RF	515.563	22.706	500	500
	LASSOPBRF	67.691	8.227	330	269
	EnetRARTEN	66.164	8.134	416	377
800	LASSO	389.787	19.743	-	269
	Enet	839.53	28.974	-	377
	RF	507.419	22.525	800	500
	LASSOPBRF	67.82	8.235	378	269
	EnetRARTEN	65.649	8.102	582	377
1000	LASSO	387.977	19.697	-	269
	Enet	799.142	28.269	-	375
	RF	516.7	22.731	1000	500
	LASSOPBRF	67.882	8.239	397	269
	EnetRARTEN	65.173	8.073	693	375
OR=15%					
200	LASSO	393.698	19.841	-	268
	Enet	812.3	28.5	-	377
	RF	524.335	22.898	200	500
	LASSOPBRF	69.173	8.317	179	268
	EnetRARTEN	68.981	8.305	192	377
500	LASSO	394.781	19.869	-	268
	Enet	752.195	27.426	-	367
	RF	515.274	22.699	500	500
	LASSOPBRF	67.613	8.222	330	268
	EnetRARTEN	66.331	8.144	414	367
800	LASSO	390.689	19.765	-	268
	Enet	836.303	28.918	-	378
	RF	515.153	22.696	800	500
	LASSOPBRF	68.073	8.25	377	268
	EnetRARTEN	65.894	8.117	572	378
1000	LASSO	389.652	19.739	-	268
	Enet	859.782	29.322	-	381
	RF	510.658	22.597	1000	500
	LASSOPBRF	68.249	8.261	396	268
	EnetRARTEN	65.733	8.107	669	381

Table S11: SRs when n=500, P=1000, $\rho_x = 0.80$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	2344.31	48.418	-	667
	Enet	3556.755	59.638	-	775
	RF	487.469	22.078	200	1000
	LASSOPBRF	62.275	7.891	188	667
	EnetRARTEN	62.54	7.908	180	775
500	LASSO	2359.815	48.577	-	664
	Enet	3768.472	61.387	-	807
	RF	473.777	21.766	500	1000
	LASSOPBRF	58.08	7.621	463	664
	EnetRARTEN	58.6	7.655	426	807
800	LASSO	2272.547	47.671	-	664
	Enet	3427.746	58.546	-	738
	RF	407.74	20.192	800	1000
	LASSOPBRF	57.04	7.552	670	664
	EnetRARTEN	57.413	7.577	652	738
1000	LASSO	2322.336	48.19	-	665
	Enet	3023.95	54.99	-	700
	RF	468.432	21.643	1000	1000
	LASSOPBRF	57.012	7.55	740	665
	EnetRARTEN	57.061	7.553	756	700
OR=15%					
200	LASSO	2221.654	47.134	-	656
	Enet	3217.188	56.72	-	737
	RF	484.344	22.007	200	1000
	LASSOPBRF	62.562	7.909	187	656
	EnetRARTEN	63.434	7.964	180	737
500	LASSO	2308.197	48.043	-	661
	Enet	3711.533	60.922	-	769
	RF	422.436	20.553	500	1000
	LASSOPBRF	58.493	7.648	459	661
	EnetRARTEN	58.993	7.68	419	769
800	LASSO	2318.551	48.151	-	662
	Enet	3160.647	56.219	-	720
	RF	466.171	21.59	800	1000
	LASSOPBRF	57.494	7.582	664	662
	EnetRARTEN	57.892	7.608	644	720
1000	LASSO	2297.004	47.927	-	659
	Enet	3334.79	57.747	-	754
	RF	461.022	21.471	1000	1000
	LASSOPBRF	57.448	7.579	726	659
	EnetRARTEN	57.521	7.584	747	754

Table S12: SRs when n=500, P=1000, $\rho_x = 0.30$

Ntree	Algorithm	MSE	RMSE	#ST	#SV
OR=10%					
200	LASSO	2121.678	46.061	-	575
	Enet	4105.331	64.072	-	744
	RF	1613.557	40.169	200	1000
	LASSOPBRF	216.679	14.72	192	575
	EnetRARTEN	216.186	14.703	197	744
500	LASSO	2154.761	46.419	-	574
	Enet	4154.013	64.451	-	745
	RF	1592.892	39.911	500	1000
	LASSOPBRF	208.717	14.447	468	574
	EnetRARTEN	208.667	14.445	450	745
800	LASSO	2125.489	46.103	-	574
	Enet	4276.956	65.398	-	746
	RF	1591.201	39.889	800	1000
	LASSOPBRF	209.461	14.472	645	574
	EnetRARTEN	207.58	14.407	686	746
1000	LASSO	2133.461	46.189	-	575
	Enet	4906.392	70.045	-	790
	RF	1578.005	39.724	1000	1000
	LASSOPBRF	209.051	14.458	706	575
	EnetRARTEN	205.481	14.334	784	790
OR=15%					
200	LASSO	2117.779	46.019	-	573
	Enet	4697.491	68.538	-	784
	RF	1581.203	39.764	200	1000
	LASSOPBRF	216.767	14.723	192	573
	EnetRARTEN	215.491	14.679	198	784
500	LASSO	2129.988	46.151	-	573
	Enet	4496.684	67.057	-	771
	RF	1597.818	39.972	500	1000
	LASSOPBRF	210.054	14.493	469	573
	EnetRARTEN	209.774	14.483	456	771
800	LASSO	2117.508	46.016	-	572
	Enet	4735.219	68.812	-	772
	RF	1599.241	39.99	800	1000
	LASSOPBRF	209.077	14.459	647	572
	EnetRARTEN	207.602	14.408	665	772
1000	LASSO	2083.419	45.644	-	572
	Enet	4709.021	68.622	-	780
	RF	1613.062	40.162	1000	1000
	LASSOPBRF	209.053	14.458	708	572
	EnetRARTEN	207.4	14.401	760	780

References

[1] G. Manikandan, S. Abirami, An Efficient Feature Selection Framework Based on Information Theory for High Dimensional Data, Appl. Soft Comp. 111 (2021), 107729. https://doi.org/10.1016/j.asoc.2021.107729.

[2] A. Rauschenberger, E. Glaab, M.A. van de Wiel, Predictive and Interpretable Models via the Stacked Elastic Net, Bioinformatics 37 (2020), 2012–2016. https://doi.org/10.1093/bioinformatics/btaa535.

[3] A. Rauschenberger, E. Glaab, Predicting Correlated Outcomes from Molecular Data, Bioinformatics 37 (2021), 3889–3895. https://doi.org/10.1093/bioinformatics/btab576.

[4] A.A. El-Sheikh, M.R. Abonazel, M.C. Ali, Proposed Two Variable Selection Methods for Big Data: Simulation and Application to Air Quality Data in Italy, Commun. Math. Biol. Neurosci. 2022 (2022), 16. https://doi.org/10.28919/cmbn/6908.

[5] H. Wang, G. Wang, Improving Random Forest Algorithm by Lasso Method, J. Stat. Comp. Simul. 91 (2020), 353–367. https://doi.org/10.1080/00949655.2020.1814776.

[6] T.M. Khoshgoftaar, M. Golawala, J.V. Hulse, An Empirical Study of Learning from Imbalanced Data Using Random Forest, in: 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), IEEE, Patras, Greece, 2007: pp. 310–317. https://doi.org/10.1109/ICTAI.2007.46.

[7] R. Genuer, J.M. Poggi, C. Tuleau-Malot, Variable Selection Using Random Forests, Pattern Recogn. Lett. 31 (2010), 2225–2236. https://doi.org/10.1016/j.patrec.2010.03.014.

[8] A. Hapfelmeier, K. Ulm, A New Variable Selection Approach Using Random Forests, Comp. Stat. Data Anal. 60 (2013), 50–69. https://doi.org/10.1016/j.csda.2012.09.020.

[9] S. Wager, T. Hastie, B. Efron, Confidence Intervals for Random Forests: The Jackknife and the Infinitesimal Jackknife, J. Mach. Learn. Res. 15 (2014), 1625-1651.

[10] L. Mentch, G. Hooker, Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests, arXiv preprint arXiv:1404.6473, 2014. https://doi.org/10.48550/arXiv.1404.6473.

[11] M. Roozbeh, S. Babaie-Kafaki, Z. Aminifard, Improved High-Dimensional Regression Models with Matrix Approximations Applied to the Comparative Case Studies with Support Vector Machines, Optim. Methods Softw. 37 (2022), 1912–1929. https://doi.org/10.1080/10556788.2021.2022144.

[12] M. Roozbeh, S. Babaie-Kafaki, Z. Aminifard, Two Penalized Mixed–Integer Nonlinear Programming Approaches to Tackle Multicollinearity and Outliers Effects in Linear Regression Models, J. Ind. Manage. Optim. 17 (2021), 3475-3491. https://doi.org/10.3934/jimo.2020128.

[13] M. Roozbeh, S. Babaie-Kafaki, Z. Aminifard, Improved High-Dimensional Regression Models with Matrix Approximations Applied to the Comparative Case Studies with Support Vector Machines, Optim. Methods Softw. 37 (2022), 1912–1929. https://doi.org/10.1080/10556788.2021.2022144.

[14] M. Maanavi, M. Roozbeh, Regression Analysis Methods for High-dimensional Data, Andishe 25 (2021), 69–90.

[15] M. Arashi, M. Norouzirad, M. Roozbeh, N.M. Khan, A High-Dimensional Counterpart for the Ridge Estimator in Multicollinear Situations, Mathematics 9 (2021), 3057. https://doi.org/10.3390/math9233057.

[16] Z. Farhadi, H. Bevrani, M.-R. Feizi-Derakhshi, Improving random forest algorithm by selecting appropriate penalized method, Communications in Statistics - Simulation and Computation 53 (2022) 4380–4395. https://doi.org/10.1080/03610918.2022.2150779.

[17] R. Tibshirani, Regression Shrinkage and Selection Via the Lasso, J. R. Stat. Soc. Ser. B: Stat. Methodol. 58 (1996), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[18] M. Amini, M. Roozbeh, Improving the Prediction Performance of the LASSO by Subtracting the Additive Structural Noises, Comp. Stat. 34 (2018), 415–432. https://doi.org/10.1007/s00180-018-0849-0.

[19] J. Friedman, T. Hastie, N. Simon, R. Tibshirani, Package glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models, ver. 2.0, 2016. https://cran.r-project.org/web/packages/glmnet.

[20] H. Zou, T. Hastie, Regularization and Variable Selection Via the Elastic Net, J. R. Stat. Soc. Ser. B: Stat. Methodol. 67 (2005), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[21] A.S. Al-Jawarneh, M.T. Ismail, A.M. Awajan, A.R.M. Alsayed, Improving Accuracy Models Using Elastic Net Regression Approach Based on Empirical Mode Decomposition, Comm. Stat. – Simul. Comp. 51 (2020), 4006–4025. https://doi.org/10.1080/03610918.2020.1728319.

[22] L. Breiman, Random Forests, Mach. Learn. 45 (2001), 5–32. https://doi.org/10.1023/a:1010933404324.

[23] A. Liaw, Package 'randomforest', University of California, Berkeley, CA, USA, 2018.

[24] I.H. Witten, E. Frank, M.A. Hall, What's It All About, in: Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 338, (2011).

[25] M.R. Abonazel, A.R.R. Alzahrani, A.A. Saber, I. Dawoud, E. Tageldin, A.R. Azazy, Developing Ridge Estimators for the Extended Poisson-Tweedie Regression Model: Method, Simulation, and Application, Sci. Afr. 23 (2024), e02006. https://doi.org/10.1016/j.sciaf.2023.e02006.

[26] A.H. Youssef, M.R. Abonazel, E.G. Ahmed, Robust M Estimation for Poisson Panel Data Model with Fixed Effects: Method, Algorithm, Simulation, and Application, Stat., Optim. Inf. Comp. 12 (2024), 1292–1305. https://doi.org/10.19139/soic-2310-5070-1996.

[27] M. R. Abonazel, A Practical Guide for Creating Monte Carlo Simulation Studies Using R, Int. J. Math. Comp. Sci. 4 (2018), 18-33.

[28] M.R. Abonazel, R.A. Farghali, Liu-Type Multinomial Logistic Estimator, Sankhya B 81 (2018), 203–225. https://doi.org/10.1007/s13571-018-0171-4.

[29] M.R. Abonazel, S.M. El-Sayed, O.M. Saber, Performance of Robust Count Regression Estimators in the Case of Overdispersion, Zero Inflated, and Outliers: Simulation Study and Application to German Health Data, Commun. Math. Biol. Neurosci. 2021 (2021), 55. https://doi.org/10.28919/cmbn/5658.

[30] M.M. Abdelwahab, M.R. Abonazel, A.T. Hammad, A.M. El-Masry, Modified Two-Parameter Liu Estimator for Addressing Multicollinearity in the Poisson Regression Model, Axioms 13 (2024), 46. https://doi.org/10.3390/axioms13010046.

[31] M.R. Abonazel, Handling Outliers and Missing Data in Regression Models Using R: Simulation Examples, Acad. J. Appl. Math. Sci. 6 (2020), 187–203. https://doi.org/10.32861/ajams.68.187.203.

[32] M.R. Abonazel, O.M. Saber, A Comparative Study of Robust Estimators for Poisson Regression Model with Outliers, J. Stat. Appl. Prob. 9 (2020), 279-286. http://dx.doi.org/10.18576/jsap/090208.

[33] M.R. Abonazel, I. Dawoud, Developing Robust Ridge Estimators for Poisson Regression Model, Concurr. Comp.: Pract. Exper. 34 (2022), e6979. https://doi.org/10.1002/cpe.6979.

[34] A.R. Azazy, M.R. Abonazel, A.M. Shafik, T.M. Omara, N.M. Darwish, A Proposed Robust Regression Model to Study Carbon Dioxide Emissions in Egypt, Comm. Math. Biol. Neurosci. 2024 (2024), 86. https://doi.org/10.28919/cmbn/8673.

[35] D. Rossell, D. Telesca, Nonlocal Priors for High-Dimensional Estimation, J. Amer. Stat. Assoc. 112 (2017), 254-265. https://doi.org/10.1080/01621459.2015.1130634.

[36] H. Binder, W. Sauerbrei, P. Royston, Comparison Between Splines and Fractional Polynomials for Multivariable Model Building with Continuous Covariates: A Simulation Study with Continuous Response, Stat. Med. 32 (2013), 2262-2277. https://doi.org/10.1002/sim.5639.

[37] A. Lukman, O. Arowolo, K. Ayinde, Some Robust Ridge Regression for Handling Multicollinearity and Outlier, Int. J. Sci.: Basic Appl. Res. 16 (2014), 192-202.

[38] I. Dawoud, F.A. Awwad, E. Tag Eldin, M.R. Abonazel, New Robust Estimators for Handling Multicollinearity and Outliers in the Poisson Model: Methods, Simulation and Applications, Axioms 11 (2022), 612. https://doi.org/10.3390/axioms11110612.

[39] E.R. Lee, J. Cho, K. Yu, A Systematic Review on Model Selection in High-Dimensional Regression, J. Korean Stat. Soc. 48 (2019), 1-12. https://doi.org/10.1016/j.jkss.2018.10.001.

[40] I. Dawoud, M.R. Abonazel, Robust Dawoud–Kibria Estimator for Handling Multicollinearity and Outliers in the Linear Regression Model, J. Stat. Comp. Simul. 91 (2021), 3678–3692. https://doi.org/10.1080/00949655.2021.1945063.

[41] S. Li, T.T. Cai, H. Li, Transfer Learning for High-Dimensional Linear Regression: Prediction, Estimation and Minimax Optimality, J. R. Stat. Soc. Ser. B: Stat. Methodol. 84 (2021), 149–173. https://doi.org/10.1111/rssb.12479.

[42] P. Filzmoser, K. Nordhausen, Robust Linear Regression for High-Dimensional Data: An Overview, WIREs Comp. Stat. 13 (2020), e1524. https://doi.org/10.1002/wics.1524.