Robust Multi-Model Subset Selection

Anthony-Alexander Christidis
Department of Statistics, University of British Columbia
Department of Biomedical Informatics, Harvard Medical School

Gabriela Cohen Freue
Department of Statistics, University of British Columbia

September 8, 2025

Abstract

Outlying observations can be challenging to handle and adversely affect subsequent analyses, especially in data with increasing dimensional complexity. Although outliers are not always undesired anomalies in the data and may possess valuable insights, only methods that are robust to outliers are able to accurately identify them and resist their influence. In this paper, we propose Robust Multi-Model Subset Selection (RMSS), a method that generates an ensemble of sparse and diverse predictive models that are resistant to outliers. We show that the ensembles generally outperform single-model sparse and robust methods. Cross-validation is used to tune parameters to control levels of sparsity, diversity and robustness. We establish the finite-sample breakdown point of the models generated by RMSS, including that of the Robust Best Subset Selection (RBSS) estimator as a special case. In addition, we develop a tailored computing algorithm to learn the ensemble by leveraging recent developments in ℓ_0 optimization. Our extensive numerical experiments on synthetic and artificially contaminated real datasets from bioinformatics and cheminformatics demonstrate the competitive advantage of our method over state-of-the-art single-model methods. The supplementary material contains all theoretical proofs, additional algorithmic and computational details, and the code and data to reproduce our numerical results.

Keywords: Robust methods; High-dimensional data; Ensemble methods; Multi-model optimization; Breakdown point of RBSS, Breakdown point of RMSS

1 Introduction

The rapid growth of digital technologies has led to an explosive increase in data, revolutionizing approaches to modeling and prediction. For example, advances in genomics allow for the simultaneous quantitation of thousands of genes from a patient's sample, which can be used to predict pathogenic processes or responses to therapies (e.g., Byron et al., 2016). Large volumes of data of different types, formats, and structures can be rapidly collected, generated and integrated. Thus, modern datasets are often characterized by the presence of a large number of variables (in columns), of which some are irrelevant or redundant, and in general exceeds the number of observations (in rows). Along with more sophisticated processes of obtaining data comes the inclusion of outlying observations, also known as data contamination. In this article, an outlier refers to a case (or row) in the dataset with observed values differing from those of the bulk of the data. These outliers may arise due to errors in data collection, sensitive detection of rare cases, discrepancies in data sources, or data corruption, among other reasons. Although outliers may contain valuable information, only robust methods resistant to outliers can appropriately identify and handle their influence. In this work, we leverage modern computational tools in ℓ_0-optimization to generate an ensemble of sparse and diverse predictive models that are resistant to outlying rows in the dataset and improve the performance of single-model sparse and robust methods.

Regularized regression methods have been developed to model datasets with many predictors relative to the number of samples, enabling the selection of an optimal subset of predictors for building interpretable predictive models. Prominent examples include the least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996) and the smoothly clipped absolute deviation (SCAD) method (Fan and Li, 2001). However, depending on the loss function, these methods can be very sensitive to outliers, which adversely affects their variable selection and prediction performance. To address this, robust statistical procedures are derived using loss functions that downweight the effect of outlying observations (Maronna et al., 2019). In recent years, many proposals have combined regularization and robustness to obtain predictive models resistant to various types of outliers (Maronna, 2011; Alfons et al., 2013; Smucler and Yohai, 2017; Cohen Freue et al., 2019). Many of these methods have a breakdown point of 50%, meaning that they can provide a robust prediction when the majority of the cases is not contaminated with outliers.

Ensemble methods, which generate and aggregate multiple diverse models, often outperform

single-model methods in high-dimensional prediction tasks. Traditionally, ensemble methods rely on randomization or some form of heuristics to generate diverse models, and are thus considered "blackbox" methods. Some notable examples of these ensemble methods include random forests (RF) (Breiman, 2001), random generalized linear models (RGLM) (Song et al., 2013), gradient boosting (Friedman, 2001) and all its variations (e.g., Bühlmann and Yu, 2003; Chen and Guestrin, 2016). In general, these types of ensemble methods generate a large number of uninterpretable and inaccurate models that are only useful when they are pooled together. More recently, Christidis et al. (2020) and Christidis et al. (2024) proposed methods that generate ensembles comprised of a small number of sparse and diverse models learned directly from the data without any form of randomization or heuristics. Each of the models in these ensembles have a high prediction accuracy, and the ensembling of these small models have been shown to outperform state-of-the-art blackbox ensemble methods on synthetic as well as complex biological and chemical data. However, both these ensembles methods are very sensitive to outliers.

In this article, we introduce a Robust Multi-Model Subset Selection (RMSS) method to generate ensembles comprised of a small number of sparse, robust, and diverse models in a regression setting. The levels of sparsity, diversity, and robustness are driven directly by the data via cross-validation (CV). We establish the finite-sample breakdown point (FSBP) of the models generated by RMSS, as well as the FSBP of RBSS estimator as a special case. To fit these models, we harness recent developments in ℓ_0-optimization to develop a tailored computational algorithm with attractive convergence properties. RMSS is shown to outperform state-of-the-art sparse and robust methods in an extensive simulation study and on biological and chemical datasets with simulated contamination. To the best of our knowledge, this is the first robust ensemble method proposed in the literature. Its flexibility may be particularly valuable for practitioners working with moderately high-dimensional data.

The remainder of this article is organized as follows. In Section 2, we provide a literature review. In Section 3, we introduce RMSS and some of its special cases. In Section 4, we study robustness properties of RBSS and RMSS. In Section 5, we provide a computational algorithm to generate RMSS and establish some of its convergence properties. In Section 6, we present a large simulation study. In Section 7, we apply RMSS on artificially contaminated datasets from bioinformatics and cheminformatics. Concluding remarks are given in Section 8.

2 Literature Review

In this section, we review a variety of predictive methods proposed in the literature that are related to RMSS and introduce important notation.

We consider the usual linear regression setting where a dataset comprised of n observations and p predictor variables can be used to build a predictive model for a response variable of interest. Let $\mathbf{y} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, and $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ be the rows of $\mathbf{X}$ for $1 \leq i \leq n$. We assume a standard linear model

$$y_i = \mu + \mathbf{x}_i^T \boldsymbol{\beta}_0 + \sigma \epsilon_i, \quad 1 \leq i \leq n, \tag{1}$$

where $\mu \in \mathbb{R}$ and $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ are the regression coefficients, and the elements of the noise vector $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T \in \mathbb{R}^n$ are independent and identically distributed with mean zero and variance one. We focus our attention on the high-dimensional setting ($p \gg n$) where the underlying model is sparse, i.e., the number of nonzero elements of the true coefficient vector $\|\boldsymbol{\beta}_0\|_0 \ll p$.

2.1 Single-Model Methods

Several regression methods were proposed to generate sparse predictive models based on only a subset of the predictor variables, particularly needed when p is very large compared to the number of observations n. The Best Subset Selection (BSS) estimator proposed by Garside (1965) was one of the first variable selection method, which can be defined as the solution to the non-convex (and non-differentiable) minimization problem given by

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_0 \leq t, \tag{2}$$

where $0 \leq t \leq \min(n-1, p)$ is the number of nonzero coefficients, which may be chosen by a model selection criterion (see e.g., Mallows, 1973; Akaike, 1974) or by cross-validation.

Since the BSS optimization problem (2) is an NP-hard problem (Welch, 1982), many sparse regularization methods in the form of convex relaxations of (2) were proposed, such as LASSO, Elastic Net (EN) (Zou and Hastie, 2005), and SCAD (Fan and Li, 2001) methods. Although convex relaxations have much lower computational cost, BSS enjoys better estimation and variable

selection properties compared to sparse regularization methods (Shen et al., 2013), and often outperforms regularization methods in high-dimensional prediction tasks (Hastie et al., 2020). In an effort to make BSS computational feasible in high-dimensional settings, Bertsimas et al. (2016) proposed fast and scalable algorithms to generate solutions to the BSS problem (2) directly.

Since most of these methods are based on the squared loss function, they are very sensitive to atypical observations in the data, which may adversely affect their variable selection and prediction performances. Over the last two decades, several robust methods have been proposed that can be used when $p \gg n$, can select only a subset of relevant predictors, and are resistant to outliers. Khan et al. (2007a) and Khan et al. (2007b) were among the first ones to develop robust stepwise and least angle regression (LARS) (Efron et al., 2004) algorithms using robust estimators of pairwise correlations instead of their classical sample counterparts. Alfons et al. (2013) introduced a penalized version of the least trimmed squares estimator (LTS) (Rousseeuw, 1984), called sparseLTS, by minimizing the LASSO-penalized sum of h smallest squared residuals, with $h \leq \lfloor n/2 \rfloor$. Thompson (2022) introduced the Robust Best Subset Selection (RBSS) method by combining the LTS loss with the ℓ_0-penalty for the vector of coefficients and developed an efficient algorithm for its computation. The objective function (2) becomes

$$\min_{\substack{\boldsymbol{\beta} \in \mathbb{R}^p \\ \mathcal{I}' \subseteq \mathcal{I}}} \sum_{i \in \mathcal{I}'} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta}\right)^2 \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\beta}\|_0 \leq t, \\ |\mathcal{I}'| \geq h, \end{cases} \tag{3}$$

where $|\cdot|$ is the cardinality operator, $\mathcal{I} = \{1, \ldots, n\}$ and h is an integer such that $t \leq h \leq n$.

Other sparse robust regression estimators have been later proposed using other loss and penalty functions, including the MM-LASSO (Smucler and Yohai, 2017), PENSE(M) (Cohen Freue et al., 2019) and their adaptive versions (Kepplinger, 2023).

2.2 Ensemble Methods

Ensemble methods have been proposed to generate and aggregate multiple models with appealing performance in high-dimensional prediction tasks. Ueda and Nakano (1996) decomposed the mean squared prediction error (MSPE) of regression ensembles and showed that the variance of an ensemble is largely determined by how correlated its individual models are. Thus, until recently,

most ensemble methods relied on a large number of weak decorrelated models (typically more than 100). For example, decorrelation of the individual trees in RF is achieved by random sampling of the data (i.e., bagging, Breiman, 1996a) and random sampling of the predictors (i.e., the random predictor subspace method, Ho, 1998). Similarly, ensembles from large number of diverse linear models are generated in the RGLM method (Song et al., 2013) and through gradient boosting (Chen and Guestrin, 2016). However, their individual models are not interpretable and have weak predictive accuracy. In addition, the selection of predictors is unreliable if randomization is used.

To generate ensembles of sparse, accurate and diverse models, Christidis et al. (2020) and Christidis et al. (2024) relied on the principle of the multiplicity of good models (McCullagh and Nelder, 1989). Christidis et al. (2020) proposed a method called Split-Regularized Regression (SplitReg) that splits the set of predictors into groups and builds a set of sparse models by minimizing an objective function that encourages sparsity within each group and diversity among them. To alleviate using the multi-convex relaxation of SplitReg, Christidis et al. (2024) introduced a multi-model subset selection (MSS) as a generalization of BSS in (2). Despite their high prediction accuracy, these methods are very sensitive to outliers in the data.

3 Robust Multi-Model Subset Selection

In this section we introduce our Robust Multi-Model Subset Selection (RMSS) estimator, designed to build an ensemble of $G \geq 2$ robust and sparse predictive models from moderately high-dimensional data possibly containing outlying samples.

Let β_j^g denote the coefficient for predictor j in model g, for $1 \leq j \leq p$ and $1 \leq g \leq G$. Let $\boldsymbol{\beta}^g = (\beta_1^g, \ldots, \beta_p^g)^T \in \mathbb{R}^p$ be the coefficient vector for model g, and let $\boldsymbol{\beta}_{j\cdot} = (\beta_j^1, \ldots, \beta_j^G)^T \in \mathbb{R}^G$ be the vector of coefficients for predictor j across all G models. For a fixed number of models $G \geq 2$, and tuning constants t, u, and h, RMSS solves the constrained optimization problem:

$$\min_{\substack{\boldsymbol{\beta}^1,\ldots,\boldsymbol{\beta}^G \in \mathbb{R}^p \\ \mathcal{I}^1,\ldots,\mathcal{I}^G \subseteq \mathcal{I}}} \sum_{g=1}^{G} \sum_{i \in \mathcal{I}^g} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta}^g\right)^2 \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\beta}^g\|_0 \leq t, & 1 \leq g \leq G, \\ \left\|\boldsymbol{\beta}_{j\cdot}\right\|_0 \leq u, & 1 \leq j \leq p, \\ |\mathcal{I}^g| \geq h, & 1 \leq g \leq G. \end{cases} \tag{4}$$

Each subset $\mathcal{I}^g \subseteq \mathcal{I} = \{1, \ldots, n\}$ is comprised of at least h indices of the observations used to estimate model g, with $t < h \leq n$. The parameter h controls the robustness of each model by using only the h observations with the smallest sum of squared residuals. The tuning constant t controls the number of predictors selected for each model with $t \leq \min(n-1, p)$. The tuning constant u sets the maximum number of times a predictor can be shared among models with $u \leq G$, thus controlling diversity. In our framework, these constants are chosen by CV (see Section 5.3).

A nice feature of RMSS is that the individual models in the ensemble may be fit on different subsamples (i.e., different $\mathcal{I}^g$ subsets). By fitting models on different observation subsets, the ensemble can better exploit uncontaminated portions of the data. Moreover, the combination of this flexible subsampling and variable selection framework, and the multi-model structure of RMSS may help to provide some resistance to cellwise contamination, where individual cells in the data matrix are contaminated (Alqallaf et al., 2009; Raymaekers and Rousseeuw, 2024).

Since RMSS builds multiple models on different subsets of predictors, contaminated cells in certain observations do not affect all models. On average, each submodel faces a lower effective contamination rate, so that for low levels of cellwise contamination and small model sizes the propagation may be controlled at levels that casewise subsampling-based estimators are able to withstand. The subsampling nature of RMSS further adds flexibility in adapting to different contamination patterns across predictor subsets. Overall, the ensemble can still recover the underlying relationships and the ensemble potentially use more observations relative to single-model robust methods, reducing information loss from the training data.

We now show that RMSS reduces to several well-known estimators as special cases, depending on the choice of tuning constants.

Proposition 1 *(Relationship of RMSS to Other Estimators) RMSS generalizes several other subset selection methods:*

(I) If $h = n$, RMSS is equivalent to the non-robust Multi-Model Subset Selection (MSS) method.

(II) If $u = G$, the diversity constraint is inactive, and the solution for each of the G models is the optimal solution to the RBSS problem in (3).

(III) If $u = G$ and $h = n$, RMSS is equivalent to BSS in (2).

(IV) If $u = G$ and $t = p < n - 1$, RMSS is equivalent to LTS estimator.

The proof, given in the supplementary material, follows directly from the simplification of the RMSS problem in (4) under specific parameter settings. For instance, setting $u = G$ renders the diversity constraint inactive, making the optimization problem separable across the G models and thus equivalent to solving the RBSS problem independently for each. The other cases follow from similar arguments. Since the tuning parameters (t, u, h) can be chosen by CV, RMSS can easily adapt to data with different characteristics, including cases with few predictors or no outliers.

In this article, we generate ensembles using simple model averaging, where the coefficients of the ensemble, $\bar{\boldsymbol{\beta}}$, are the average of the estimated coefficients $\hat{\boldsymbol{\beta}}^g$ of the G models. However, other methods can also be implemented, including weighted model averaging methods (Breiman, 1996b) or model aggregation methods (Biau et al., 2016).

4 Finite-Sample Breakdown Point

This section establishes the finite-sample breakdown point (FSBP) of the RBSS and RMSS coefficient estimators. The FSBP is a standard robustness measure defined by Donoho and Huber (1983) that indicates the smallest fraction of contaminated observations needed to render an estimator meaningless.

Definition 1 *(Finite-Sample Breakdown Point) Let $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ be a sample of size n. The finite-sample breakdown point (FSBP) of an estimator T at the sample $\mathbf{Z}$ is defined as the smallest fraction of contaminated points m/n that make the estimator's norm unbounded. Formally,*

$$\varepsilon^*(T; \mathbf{Z}) = \frac{1}{n} \min \left\{ m \in \{1, \ldots, n\} : \sup_{\tilde{\mathbf{Z}}_m} \left\| T(\tilde{\mathbf{Z}}_m) \right\|_2 = \infty \right\},$$

where the supremum is taken over all corrupted samples $\tilde{\mathbf{Z}}_m$ obtained by replacing m points of $\mathbf{Z}$.

While Thompson (2022) studied the FSBP of the value of the objective function, the FSBP of RBSS coefficient estimator has not been derived. Theorem 1 establishes this result for RBSS, which corresponds to RMSS when $G = 1$, as well as for the general case $G \geq 1$ defining RMSS.

Theorem 1 *(FSBP of RBSS and Individual RMSS Models) Let $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ be a sample of size n with $p > 1$ covariates (for simplicity, assume a model without intercept). Consider a tuning*

constant t *such that* $2t \leq n$ *and let* $q = \min\{2t, p\}$. *Assume that any subset of* k *covariates, with* $t \leq k \leq q$, *form a submatrix* $\mathbf{X}_K$ *with points in general position. If* $\left\lceil \frac{n+k+1}{2} \right\rceil \leq h \leq n$, *then:*

(I) The finite-sample breakdown point of RBSS coefficient estimator $\hat{\boldsymbol{\beta}}$ *is:*

$$\varepsilon^* \left(\hat{\boldsymbol{\beta}}; \mathbf{Z} \right) = \frac{n - h + 1}{n}.$$

(II) The finite-sample breakdown point of each coefficient estimator $\hat{\boldsymbol{\beta}}^g$ *of RMSS estimator is:*

$$\varepsilon^* \left(\hat{\boldsymbol{\beta}}^g; \mathbf{Z} \right) = \frac{n - h + 1}{n}, \quad 1 \leq g \leq G.$$

The proof of both parts are provided in the supplementary material. We first establish the result for the RBSS estimator, a special case of RMSS (Proposition 1). Since RBSS reduces to LTS when $t = p < n$, our proof for the lower bound of the FSBP of RBSS builds on the classical LTS breakdown point geometric argument, adapting it to the sparse setting. In our context, the number of predictors p may exceed the sample size n and the estimators are derived under ℓ_0-sparsity constraints, e.g., to have at most t nonzero components. More specifically, for the proofs of the lower bounds of the FSBP of RBSS and RMSS, we adapted Rousseeuw's geometrical construction (Rousseeuw, 1984) by projecting onto the coordinate subspace defined by the selected predictors. The proofs of the upper bounds are constructive and analogous to that of Alfons et al. (2013) but adapted to our objective functions with different constraints.

It is important to note two key differences between the FSBP of LTS and those of RBSS and RMSS. First, the general position condition is more conservative for the sparse case. Since the variables selected by RBSS and RMSS may be affected by the contamination, we require any set of k covariates, with $t \leq k \leq \min\{2t, p\}$, to have their observations in general position. If $2t \geq p$, and under the condition that $2t < n$, the condition reduces to that of LTS. Second, the FSBP of RBSS and RMSS depend on the "actual number of parameters" selected by these estimators, rather than on p. Interestingly, the dependency of the breakdown point on the effective dimension was also noted by Maronna (2011) in his analysis of Ridge S- and MM-estimators.

From Theorem 1, if $h = n$ RBSS and RMSS corresponds to the BSS and MSS, respectively, demonstrating that their breakdown points are $1/n$ and thus not resistant to any contamination level. If the trimming parameter h is chosen by CV, the extent to which RMSS model parameters

are resistant to outliers is data-driven.

5 Computing Algorithm

A brute-force evaluation of every possible predictor and observation combination in RMSS is computationally infeasible, as shown by a combinatorial analysis in the supplementary material. We therefore propose a practical algorithm that searches over a three-dimensional grid of the tuning parameters (t, u, h) in (4).

Our procedure begins by robustly standardizing the data using medians and median absolute deviations. We then generate initial disjoint predictor subsets (Section 5.1) to serve as a "warm start" for an algorithm that computes solutions over the entire (t, u, h) grid (Section 5.2). While we also developed an optional neighborhood search to refine these solutions (see supplementary material), its marginal impact on performance did not justify its computational cost for the main procedure. Finally, all estimated coefficients are returned to their original scale.

5.1 Initial Predictor Subsets

To initialize the main RMSS algorithm, we require a high-quality, robust partition of predictors into G disjoint subsets. Christidis et al. (2024) proposed a multi-model forward selection algorithm that selects G disjoint subsets of predictors to initialize the algorithm of MSS. However, their approach is sensitive to outliers in the data. Thus, in this section we propose a robust multi-model stepwise selection procedure to select disjoint initial subsets of predictors.

Algorithm 1 generalizes the robust forward stepwise regression algorithm of Khan et al. (2007a) to multiple models. For a single-model framework, they showed that the stepwise forward search algorithm depends only the sample means, variances and correlations between the variables in the model. Thus, to make the algorithm robust to outlying observations, they proposed replacing these sample estimators by robust counterparts to compute the residual sum of squares and partial F-rules. In the supplementary material, we explicitly show this connection, which in turn enabled us to derive the computationally efficient and novel recursive formulas shown in Proposition 2. Consistently with our algorithm and code, these formulas specify the rule for selecting the best single predictor from the candidate set.

Proposition 2 *(Correlation-Based Formulas) Let* $\hat{\mathbf{r}}_{\mathbf{y}} = (\hat{r}_{y1}, \ldots, \hat{r}_{yp})^T \in \mathbb{R}^p$ *be the vector of robust correlations between the response and predictors, and let* $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}$ *be the robust correlation matrix of the predictors. For a single model* g *with current active set* $\mathcal{J}^g \subseteq \mathcal{J} = \{1, \ldots, p\}$*, the following rules determine how the optimal* k*-th predictor,* j_k*, is selected from a candidate set* $\mathcal{C} \subseteq \mathcal{J}$ *and how the robust Residual Sum of Squares (rRSS) is calculated:*

- *If the model is empty (*$k = 1$ *and* $\mathcal{J}^g = \emptyset$*):*

$$j_1 = \arg\max_{j \in \mathcal{C}} |\hat{r}_{yj}|, \qquad rRSS_1 = n(1 - \hat{r}^2_{yj_1}).$$

- *If the model is not empty (*$k \geq 2$ *and* $\mathcal{J}^g = \{j_1, \ldots, j_{k-1}\}$*):*

$$j_k = \arg\max_{j \in \mathcal{C}} \left| \frac{P_{yj \cdot j_1 \ldots j_{k-1}}}{\sqrt{P_{jj \cdot j_1 \ldots j_{k-1}}}} \right|, \qquad rRSS_k = rRSS_{k-1} - \frac{P^2_{yj_k \cdot j_1 \ldots j_{k-1}}}{P_{j_k j_k \cdot j_1 \ldots j_{k-1}}}.$$

The partial covariance terms P *are computed recursively as follows.*

For $k = 2$*:*

$$\begin{cases} P_{yj \cdot j_1} = n\left(\hat{r}_{yj} - \hat{\Sigma}_{jj_1}\hat{r}_{yj_1}\right), \\ P_{j_a j_b \cdot j_1} = n\left(\hat{\Sigma}_{j_a j_b} - \hat{\Sigma}_{j_a j_1}\hat{\Sigma}_{j_b j_1}\right). \end{cases}$$

For $k \geq 3$*:*

$$\begin{cases} P_{yj \cdot j_1 \ldots j_k} = P_{yj \ldots j_{k-1}} - \frac{P_{jj_k \cdots j_{k-1}}}{P_{j_k j_k \cdots j_{k-1}}} P_{yj_k \ldots j_{k-1}}, \\ P_{j_a j_b \ldots j_k} = P_{j_a j_b \ldots j_{k-1}} - \frac{P_{j_a j_k \cdots j_{k-1}} P_{j_b j_k \cdots j_{k-1}}}{P_{j_k j_k \cdots j_{k-1}}}. \end{cases}$$

Algorithm 1 describes our multi-model initialization procedure, which uses the formulas from Proposition 2 within a competitive framework. The goal of this framework is to produce disjoint initial predictor sets, which encourages diversity and provides a more effective "warm start" for the main RMSS optimization.

In our implementation, we compute robust correlation estimates via the state-of-the-art Detect Deviating Cells (DDC) method of Rousseeuw and Bossche (2018), which can be scaled to high-dimensional settings using properties of product moments (Raymaekers and Rousseeuw, 2021). Following in Khan et al. (2007a), Proposition 2 shows how these robust estimates are used to compute robust residual sum of squares (rRSS) and robust partial F-rules. Due to post-inference problems and the use of plug-in estimators, the resulting methodology can not be used to perform

formal F-tests with valid p-values from the F-distribution. In Algorithm 1 these rules and p-values are only used to develop a selection and a stopping criterion.

At each iteration, the algorithm employs a greedy strategy: it identifies the best candidate for each model, then assigns the single predictor with the smallest robust partial F-test p-value across all models to its corresponding model, removing it from the candidate pool $\mathcal{C}$. The saturation threshold, $\gamma \in (0, 1)$, controls variable addition during initialization. While a selective approach (e.g., $\gamma = 0.05$) is reasonable, an inclusive strategy is also justified. Since the primary objective is to generate a rich warm start and the final model sparsity is rigorously controlled later by the tuning parameter t, a non-restrictive value (e.g., $\gamma = 1$) can be used. This allows the stepwise procedure to populate the initial models generously until they reach their maximum size or the predictor pool is exhausted.

Algorithm 1 Robust Multi-Model Stepwise Selection

Input: Robust correlation vector of the response $\hat{\mathbf{r}}_{\mathbf{y}} \in \mathbb{R}^p$, robust correlation matrix of predictors $\hat{\mathbf{\Sigma}} \in \mathbb{R}^{p \times p}$, number of models $G \geq 2$, and saturation threshold $\gamma \in (0, 1)$.

Initialize: The set of candidates $\mathcal{C} = \{1, \ldots, p\}$, and for each model the set of model predictors $\mathcal{J}^g = \emptyset$ and the model saturation indicator $M_g = \text{FALSE}$, $1 \leq g \leq G$.

1: Repeat the following steps until $M_g = \text{TRUE}$ for all g, $1 \leq g \leq G$:

1.1. For each model g satisfying $M_g = \text{FALSE}$:

1.1.1: Using Proposition 2 with candidate set $\mathcal{C}$, identify candidate predictor j_g that maximizes the decrease in rRSS.

1.1.2: Calculate the p-value γ_g from the robust partial F-rule using the decrease in rRSS for adding predictor j_g.

1.1.3: If $\gamma_g \geq \gamma$, set $M_g = \text{TRUE}$.

1.2. Identify the unsaturated model g^* with the smallest p-value γ_{g^*}.

1.3. If $\gamma_{g^*} < \gamma$:

1.3.1: Update the set of predictors for model g^*: $\mathcal{J}^{g^*} \leftarrow \mathcal{J}^{g^*} \cup \{j_{g^*}\}$.

1.3.2: Update the set of candidate predictors $\mathcal{C} \leftarrow \mathcal{J} \setminus \{j_{g^*}\}$.

1.3.3: If $|\mathcal{J}^{g^*}| = n - 1$, set $M_{g^*} = \text{TRUE}$.

2: Return the sets of model predictors $\mathcal{J}^g$, $1 \leq g \leq G$.

5.2 Computing RMSS over a Grid of Tuning Constants

To develop an efficient computing algorithm to compute RMSS over a grid of t, u and h values, we recast (4) in its equivalent form using auxiliary variables $\boldsymbol{\eta}^g = (\eta_1^g, \ldots, \eta_n^g)^T \in \mathbb{R}^n$, $1 \leq g \leq G$, with 0 entries for the smallest h residuals of each model g, and the corresponding residuals otherwise. For any t, u and h values, the equivalent reformulation of RMSS is given by

$$\min_{\substack{\boldsymbol{\beta}^1,\ldots,\boldsymbol{\beta}^G \in \mathbb{R}^p \\ \boldsymbol{\eta}^1,\ldots,\boldsymbol{\eta}^G \in \mathbb{R}^p}} \sum_{g=1}^{G} \mathcal{L}_n\left(\boldsymbol{\beta}^g, \boldsymbol{\eta}^g | \mathbf{y}, \mathbf{X}\right) \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\beta}^g\|_0 \leq t, & 1 \leq g \leq G, \\ \left\|\boldsymbol{\beta}_{j\cdot}\right\|_0 \leq u, & 1 \leq j \leq p, \\ \|\boldsymbol{\eta}^g\|_0 \leq n-h, & 1 \leq g \leq G, \end{cases} \tag{5}$$

where the loss function $\mathcal{L}_n$ used for each model g is given by

$$\mathcal{L}_n\left(\boldsymbol{\beta}^g, \boldsymbol{\eta}^g | \mathbf{y}, \mathbf{X}\right) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^g - \boldsymbol{\eta}^g\|_2^2 . \tag{6}$$

The gradients $\nabla_{\boldsymbol{\beta}}\mathcal{L}_n(\boldsymbol{\beta}, \boldsymbol{\eta}|\mathbf{y}, \mathbf{X})$ and $\nabla_{\boldsymbol{\eta}}\mathcal{L}_n(\boldsymbol{\beta}, \boldsymbol{\eta}|\mathbf{y}, \mathbf{X})$ are both Lipschitz continuous with Lipschitz constants $\ell_{\boldsymbol{\beta}} = 2\|\mathbf{X}^T\mathbf{X}\|_2^2$ and $\ell_{\boldsymbol{\eta}} = 2$, respectively (see proofs in the supplementary material).

We now introduce some notation needed to outline our computing algorithms.

Definition 2 *(Projection Operator) For any $v \in \mathbb{R}^p$ and $r \in \mathbb{R}$, the projection operator $\mathcal{P}(v; r)$, defined as*

$$\mathcal{P}\left(v; r\right) \in \underset{w \in \mathbb{R}^p}{\arg\min} \|w - v\|_2^2 \quad \textit{subject to} \quad \|w\|_0 \leq r \tag{7}$$

retains the r largest elements in absolute value of the vector v.

Definition 3 *(Projected Subset Operator) For any $v \in \mathbb{R}^p$, $\mathcal{S} \subseteq \mathcal{J} = \{1, \ldots, p\}$ and $r \in \mathbb{R}$, the projected subset operator $\mathcal{Q}(v; \mathcal{S}, r)$, defined as*

$$\mathcal{Q}\left(v; \mathcal{S}, r\right) \in \underset{w \in \mathbb{R}^p}{\arg\min} \ \|w - v\|_2^2 \quad \textit{subject to} \quad \begin{cases} \|w\|_0 \leq r \\ \{j \in \mathcal{J} : w_j \neq 0\} \subseteq \mathcal{S} \end{cases} \tag{8}$$

retains the r largest elements in absolute value of the vector v that belong to the subset $\mathcal{S}$.

Note that both $\mathcal{P}(v;r)$ and $\mathcal{Q}(v;\mathcal{S},r)$ are set-valued maps since more than one possible permutation of the indices $\mathcal{J}=\{1,\ldots,p\}$ and $\{j\in\mathcal{J}:j\in\mathcal{S}\}$ may exist.

For a given set of multi-model coefficient estimates $\hat{\boldsymbol{\beta}}^g=(\hat{\beta}_1^g,\ldots,\hat{\beta}_p^g)^T\in\mathbb{R}^p$, $1\leq g\leq G$, let $\mathcal{J}^g=\{j\in\mathcal{J}:\hat{\boldsymbol{\beta}}_j^g\neq 0\}$ be the subsets of predictors' indices included in each model and $\mathcal{S}_u^{(g)}$ be the subsets of predictors' indices used in at most $u-1$ models excluding model g,

$$\mathcal{S}_u^g=\left\{j\in\mathcal{J}:\sum_{\substack{l=1\\ l\neq g}}^{G}\mathbb{I}\left(j\in\mathcal{J}^l\right)\leq u-1\right\}. \tag{9}$$

Lastly, denote $\mathbf{X}_{\mathcal{S}}\in\mathbb{R}^{n\times|\mathcal{S}|}$ the submatrix of $\mathbf{X}$ with column indices $\mathcal{S}\subseteq\mathcal{J}=\{1,\ldots,p\}$.

For fixed tuning constants t, u and h and given some starting values $(\tilde{\boldsymbol{\beta}}^g,\tilde{\boldsymbol{\eta}}^g)$, $1\leq g\leq G$, Algorithm 2, outlines the steps to perform a projected subset block gradient descent (PSBGD) to generate robust multi-model estimates. For each model at a time, the algorithm alternates between updates of the coefficient estimates $\hat{\boldsymbol{\beta}}^g$ and updates of the subsamples selected given the size of the updated residuals in $\hat{\boldsymbol{\eta}}^g$ until convergence is achieved.

Proposition 3 below establishes the convergence of the Algorithm 2 (see the proof in the supplementary material).

Proposition 3 *(Model-Wise Convergence of PSBGD Algorithm) For each model g in (5), step 1 of Algorithm 2 generates a converging sequence for the pair $(\hat{\boldsymbol{\beta}}^g,\hat{\boldsymbol{\eta}}^g)$ and the inequalities*

$$\left\|\hat{\boldsymbol{\beta}}^g-\mathcal{Q}\left(\hat{\boldsymbol{\beta}}^g-\frac{1}{L_{\boldsymbol{\beta}^g}}\nabla_{\boldsymbol{\beta}}\mathcal{L}_n\left(\boldsymbol{\beta},\hat{\boldsymbol{\eta}}^g|\mathbf{y},\mathbf{X}\right)\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}^g};\mathcal{S}_u^g,r\right)\right\|_2^2\leq\delta,$$
$$\left\|\hat{\boldsymbol{\eta}}^g-\mathcal{P}\left(\hat{\boldsymbol{\eta}}^g-\frac{1}{L_{\boldsymbol{\eta}}}\nabla_{\boldsymbol{\eta}}\mathcal{L}_n\left(\hat{\boldsymbol{\beta}}^g,\boldsymbol{\eta}|\mathbf{y},\mathbf{X}\right)\Big|_{\boldsymbol{\eta}=\hat{\boldsymbol{\eta}}^g};n-h\right)\right\|_2^2\leq\delta,$$

can be achieved in $O(1/\delta)$ iterations.

In Algorithm 3, we outline the steps to generate the solutions $(\hat{\boldsymbol{\beta}}^g[t,u,h],\hat{\boldsymbol{\eta}}^g[t,u,h])$, $1\leq g\leq G$, for any t, u, and h over the grids $\mathcal{T}=\{t_1,\ldots,t_q\}$, $\mathcal{U}=\{1,\ldots,G\}$ and $\mathcal{H}=\{h_1,\ldots,h_r\}$, respectively.

Algorithm 2 Projected Subset Block Gradient Descent (PSBGD)

Input: Matrix of predictor variables $\mathbf{X} \in \mathbb{R}^{n\times p}$, response vector $\mathbf{y} \in \mathbb{R}^n$, starting values $(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g)$, $1 \le g \le G$, tuning parameters t, u and h, and tolerance parameter $\epsilon > 0$.
Initialize: For each model g, the set of model predictors $\mathcal{J}^g = \{j \in \mathcal{J} : \tilde{\boldsymbol{\beta}}^g_j \neq 0\}$, $1 \le g \le G$.

1: Repeat the following steps for each model g, $1 \le g \le G$:

1.1: Create the set of predictors' indices $\mathcal{S}^g_u$ via (9) and compute the Lipschitz constant $\ell_{\boldsymbol{\beta}^g} = 2\|\mathbf{X}^T_{\mathcal{S}^g_u}\mathbf{X}_{\mathcal{S}^g_u}\|_2$.

1.2: Update, sequentially, current estimates $\tilde{\boldsymbol{\beta}}^g$ and $\tilde{\boldsymbol{\eta}}^g$ via

$$\hat{\boldsymbol{\beta}}^g \in \mathcal{Q}\left(\tilde{\boldsymbol{\beta}}^g - \frac{1}{L_{\boldsymbol{\beta}^g}}\nabla_{\boldsymbol{\beta}}\mathcal{L}_n\left(\boldsymbol{\beta}, \hat{\boldsymbol{\eta}}^g|\mathbf{y}, \mathbf{X}\right)\Big|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}^g}; \mathcal{S}^g_u, r\right)$$
$$\hat{\boldsymbol{\eta}}^g \in \mathcal{P}\left(\tilde{\boldsymbol{\eta}}^g - \frac{1}{L_{\boldsymbol{\eta}}}\nabla_{\boldsymbol{\eta}}\mathcal{L}_n\left(\hat{\boldsymbol{\beta}}^g, \boldsymbol{\eta}|\mathbf{y}, \mathbf{X}\right)\Big|_{\boldsymbol{\eta}=\tilde{\boldsymbol{\eta}}^g}; n-h\right)$$

with $L_{\boldsymbol{\beta}^g} \ge \ell_{\boldsymbol{\beta}^g}$ and $L_{\boldsymbol{\eta}} \ge \ell_{\boldsymbol{\eta}}$, and repeat until $\mathcal{L}_n(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g|\mathbf{y}, \mathbf{X}) - \mathcal{L}_n(\hat{\boldsymbol{\beta}}^g, \hat{\boldsymbol{\eta}}^g|\mathbf{y}, \mathbf{X}) \le \epsilon$.

1.3: Update the model predictors $\mathcal{J}^g = \{j \in \mathcal{J} : \hat{\beta}^g_j \neq 0\}$ and compute the set of model subsamples $\mathcal{I}^g = \{i \in \mathcal{I} : \hat{\eta}^g_i = 0\}$.

1.4: Compute the final model coefficients

$$\hat{\boldsymbol{\beta}}^g = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \sum_{i\in\mathcal{I}^g}\left(y_i - \mathbf{x}_i^T\boldsymbol{\beta}\right)^2 \quad \text{subject to} \quad \beta_j = 0, j \notin \mathcal{J}^g, \quad 1 \le g \le G.$$

2: Return the pairs $(\hat{\boldsymbol{\beta}}^g, \hat{\boldsymbol{\eta}}^g)$, $1 \le g \le G$.

5.3 Selection of Tuning Parameters

We use 5-fold CV to select the final combination of (t, u, h) from the candidate grids $\mathcal{T}$,$\mathcal{U}$, and $\mathcal{H}$. Since the test folds may contain outliers, we use the fast and robust τ-scale estimator of prediction residuals to evaluate performance for each parameter combination (Maronna and Zamar, 2002). Our final model is the one corresponding to the combination with the smallest robust scale estimate. This robust CV approach provides stable parameter selection even in high-noise scenarios where traditional CV using mean squared error may fail. As previously mentioned, the combination of multiple models ($G > 1$) with the diversity constraint ($u < G$) provides natural regularization that helps prevent overfitting and enhances stability, which is reflected in the strong performance of RMSS across a range of signal-to-noise scenarios in our simulations.

The tuning parameters in RMSS have clear interpretations that guide their practical selection.

Algorithm 3 Decrementing Diversity PSBGD

Input: Design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, response vector $\mathbf{y} \in \mathbb{R}^n$, grids of tuning parameters $\mathcal{T} = \{t_1, \ldots, t_q\}$, $\mathcal{U} = \{1, \ldots, G\}$ and $\mathcal{H} = \{h_1, \ldots, h_r\}$, and tolerance parameter $\epsilon > 0$.

1: Use Algorithm 1 to obtain the sets of predictors' indices $\mathcal{J}^g$, $1 \leq g \leq G$, and compute the initial model coefficients

$$\hat{\boldsymbol{\beta}}^g_{\text{init}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to} \quad \beta_j = 0, j \notin \mathcal{J}^g.$$

2: For each combination of $t \in \mathcal{T}$ and $h \in \mathcal{H}$:

2.1: Compute the pairs $(\hat{\boldsymbol{\beta}}^g[t, 1, h], \hat{\boldsymbol{\eta}}^g[t, 1, h])$, $1 \leq g \leq G$, using Algorithm 2 initialized with $(\hat{\boldsymbol{\beta}}^g_{\text{init}}, \mathbf{0}_n)$ where $\mathbf{0}_n = (0, \ldots, 0)^T \in \mathbb{R}^n$, $1 \leq g \leq G$.

2.2: For $u = 2, \ldots, G$:

2.2.1: Compute the pairs $(\hat{\boldsymbol{\beta}}^g[t, u, h], \hat{\boldsymbol{\eta}}^g[t, u, h])$, $1 \leq g \leq G$, using Algorithm 2 initialized with $(\hat{\boldsymbol{\beta}}^g[t, u-1, h], \hat{\boldsymbol{\eta}}^g[t, u-1, h])$, $1 \leq g \leq G$.

3: Return the pairs $(\hat{\boldsymbol{\beta}}^g[t, u, h], \hat{\boldsymbol{\eta}}^g[t, u, h])$, $1 \leq g \leq G$, for all combinations of $t \in \mathcal{T}$, $u \in \mathcal{U}$ and $h \in \mathcal{H}$.

The subsample size h, controls the robustness of each model by using only the h observations that produce the smallest sum of squared residuals. This tuning parameter can be set based on the anticipated level of contamination in the data (for example, $h = 0.75n$ corresponds to 25% contamination) or chosen by CV from a grid (e.g., $\mathcal{H} = \{\lfloor n/2 \rfloor + 1, \ldots, n\}$). The sparsity parameter, t, controls model complexity for each model with $t \leq \min(n-1, p)$ and less dense grids can be used to speed computation. The diversity parameter, u, controls the number of predictors shared across models, where $u = 1$ enforces maximum diversity. Typically, $u = 1$ (completely distinct models) or $u = 2$ (predictors shared at most by 2 models) perform well in our simulation studies.

For the ensemble size G, we recommend starting with $G = 5$ for $p \ll 500$ and $G = 10$ otherwise, as our simulations show diminishing returns beyond these choices due to increased computational cost without proportional performance gains. The CV procedure scales as $O(G \times \text{CV-folds} \times |\mathcal{T}| \times |\mathcal{U}| \times |\mathcal{H}|)$, making it computationally feasible for typical problems with modest grid sizes.

5.4 Software

The implementation of robust multi-model stepwise selection outlined in Algorithm 1 is available on CRAN (R Core Team, 2022) in the `R` package `robStepSplitReg` (Christidis and Cohen-Freue, 2025). The implementation of Algorithms 2 - 3 to fit RMSS ensembles is also available on CRAN in the `R` package `RMSS` (Christidis and Cohen-Freue, 2023), which generates RBSS if $G = 1$. The source code of `RMSS` is written in `C++`, and multithreading via OpenMP (Chandra et al., 2001) is available in the package to further speed up computations. A compressed archive containing a detailed `README` file, all `R` scripts and data files necessary to reproduce the numerical results is available as supplementary material. To improve accessibility, all files are also archived on Zenodo (`https://doi.org/10.5281/zenodo.16915351`).

6 Simulations

In this section, we investigate the performance of RMSS against robust and sparse methods as well as blackbox ensemble methods in an extensive simulation study where the data is contaminated in both the predictor space and the response. We also use a block correlation structure between predictor variables to mimic as closely as possible the behavior of many modern datasets (Zhang and Coombes, 2012).

6.1 Simulation of Uncontaminated Data

In each setting of our simulation study, we generate the uncontaminated data from the linear model

$$y_i = \mu + \mathbf{x}_i^T \boldsymbol{\beta}_0 + \sigma \epsilon_i, \quad 1 \leq i \leq n,$$

with independent and identically distributed $\mathbf{x}_i \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$ and $\epsilon_i \sim N(0, 1)$, sample size $n = 50$ and the number of predictors $p = 500$. Alternative choices for n and p where $p \gg n$ lead to similar conclusions and are omitted for conciseness.

We examine models with different proportions ζ of active predictors, with $\zeta \in \{0.1, 0.2, 0.4\}$. Without loss of generality, we construct the correlation matrix of the active predictors as a block

matrix, with each block corresponding to 25 predictors, a within-block correlation $\rho_1 = 0.8$ and a between-block correlation $\rho_2 = 0.2$. The non-active predictors are independent and uncorrelated with the active ones.

For simplicity, the intercept is set to $\mu = 0$. The coefficients of the active variables, $\{\beta_j : \beta_j \neq 0, 1 \leq j \leq p\}$, are randomly generated from the random variable $(-1)^Z \times U$, where Z is Bernoulli distributed with parameter 0.7 and U is uniformly distributed on the interval $(0, 5)$.

The noise parameter σ is computed based on the desired signal to noise ratio, $\text{SNR} = \boldsymbol{\beta}_0' \boldsymbol{\Sigma} \boldsymbol{\beta}_0 / \sigma^2$. We consider SNRs of 0.5 (low signal), 1 (moderate signal), 2 (high signal), which correspond to proportions of variance explained $\text{PVE} = \text{SNR}/(\text{SNR}+1)$ of 33.3%, 50% and 66.7%, respectively.

6.2 Data Contamination

We contaminate the first $m = \lfloor \alpha n \rfloor$ samples $(\mathbf{x}_i, y_i)$ according to the model proposed in Maronna (2011). The regression outliers are introduced by replacing the predictors $\mathbf{x}_i$ with

$$\tilde{\mathbf{x}}_\mathbf{i} = \Theta_i + \frac{\mathrm{k}_{\text{lev}}}{\sqrt{\mathbf{a}^T \boldsymbol{\Sigma}^{-1} \mathbf{a}}} \mathbf{a}, \quad 1 \leq i \leq m,$$

where $\Theta_i \sim \mathcal{N}(\mathbf{0_p}, 0.01 \times \mathbf{I_p})$ and $\mathbf{a} = \tilde{\mathbf{a}} - (1/p)\tilde{\mathbf{a}}^T \mathbf{1_p}$ where $\mathbf{I_p}$ is the p-dimensional identity matrix, $\mathbf{0_p} = (0, \ldots, 0)^T \in \mathbb{R}^p$, $\mathbf{1_p} = (1, \ldots, 1)^T \in \mathbb{R}^p$, and the entries of $\tilde{a}_j$ of $\tilde{\mathbf{a}}$ follow a uniform distribution on the interval $(-1, 1)$, $1 \leq j \leq p$. The parameter k_{lev} controls the distance in the direction most influential for the estimator.

We also contaminate the observation in the response by altering the regression coefficient

$$\tilde{y}_i = \tilde{\mathbf{x}}_\mathbf{i}^T \tilde{\boldsymbol{\beta}}, \quad \tilde{\beta}_j = \begin{cases} \beta_j(1 + \mathrm{k}_{\text{slo}}), & \beta_j \neq 0, \\ \mathrm{k}_{\text{slo}} \left\| \boldsymbol{\beta} \right\|_\infty, & \text{otherwise}, \end{cases} \quad 1 \leq i \leq m.$$

The parameters k_{lev} and k_{slo} control the position of contaminated observations. Preliminary experiments showed that estimator performance was nearly unchanged for any $k_{\text{lev}} > 1$, so we fixed $k_{\text{lev}} = 2$. In contrast, the position of vertical outliers had a much stronger impact: non-robust estimators degraded markedly for any $k_{\text{slo}} \geq 100$, so we fixed $k_{\text{slo}} = 100$. We consider contamination proportions $\alpha = 0$ (none), $\alpha = 0.15$ (moderate), and $\alpha = 0.3$ (high).

6.3 Methods

Our simulation study compares the prediction and variable selection accuracy of eight methods. All computations were carried out in `R` using the implementations listed below

1. Elastic Net (**EN**, Zou and Hastie, 2005), with `glmnet` package (Friedman et al., 2010).
2. Adaptive **PENSE** (Kepplinger, 2023), with `pense` package (Kepplinger et al., 2023).
3. EN Penalized Huber (**HuberEN**, Yi and Huang, 2017), with `hqreg` package (Yi, 2017).
4. Sparse LTS (**SparseLTS**, Alfons et al., 2013), with `robustHD` package (Alfons, 2021).
5. Robust Best Subset Selection (**RBSS**, Thompson, 2022), with `RMSS` package.
6. Robust Multi-Model Subset Selection (**RMSS**) with $G = 10$ models, proposed in this paper, with `RMSS` package.
7. Random GLM (**RGLM**, Song et al., 2013), with `RMSS` package.
8. Random Forest (**RF**, Breiman, 2001), with `randomForest` package (Liaw and Wiener, 2002).

To reduce computational cost in our extensive simulation study, we used modest tuning grids for RMSS. For the sparsity parameter, we used the candidate grid $\mathcal{T} = \{15, 20, 25\}$. The robustness parameter h was selected via CV from the grid $\mathcal{H} = \{(1-(\alpha+0.1))n, (1-(\alpha+0.05))n, (1-\alpha)n\}$, where α is the true contamination level. We used the default grid $\mathcal{U} = \{1, \ldots, G\}$ with $G = 10$ for the diversity parameter. RBSS was computed concurrently by evaluating the $u = G$ case (see Proposition 1). While more refined grids could yield further improvements, these settings proved sufficient for RMSS to be highly competitive. We used 5-fold CV for the robust methods and 10-fold CV for EN; additional details are in the supplementary material.

6.4 Performance Measures

For each simulation configuration (i.e., each combination of sparsity ζ, SNR, and contamination α), we generated $N = 50$ training sets and a single, large, uncontaminated test set of size 2,000. For each replication, we fit the methods on the training data and evaluated performance on the test set.

We report three performance metrics. The Mean Squared Prediction Error (MSPE) is reported relative to the irreducible error variance, σ^2, making the optimal value 1. For variable selection, we compute Recall (RC) and Precision (PR), defined as:

$$\text{RC} = \frac{\sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0, \hat{\beta}_j \neq 0)}{\sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0)}, \quad \text{PR} = \frac{\sum_{j=1}^{p} \mathbb{I}(\beta_j \neq 0, \hat{\beta}_j \neq 0)}{\sum_{j=1}^{p} \mathbb{I}(\hat{\beta}_j \neq 0)},$$

where $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}$ are the true and estimated coefficients, respectively. For RMSS, we use the averaged ensemble fit. We do not report RC and PR for RGLM and RF, as their dense nature makes these metrics uninformative. For both RC and PR, higher values are better.

6.5 Results

In Table 1, we report the average and lowest MSPE rank of the eight methods over the nine SNR and sparsity level combinations for each contamination proportion. The top two performances in each column are highlighted in bold.

In the uncontaminated case ($\alpha = 0$), RGLM achieved the best overall rank, followed by RMSS and RF. RMSS consistently outperformed the non-robust EN, demonstrating its ability to adapt well to clean data. For moderate contamination ($\alpha = 0.15$), PENSE and RMSS were the clear top performers, ranking first and second, respectively, across every simulation setting. As shown in Figure 1, their prediction performances were very similar under this scenario. However, in the high contamination case ($\alpha = 0.30$), PENSE's performance deteriorated while RMSS achieved the top rank in every configuration. The non-robust ensemble methods' predictive performance degraded completely under both moderate and high contamination.

To better contextualize these summary results, Figure 1 plots the MSPE of the top robust competing methods (PENSE, RBSS, and RMSS) from the $N = 50$ random training sets for a moderate SNR. It is evident that while PENSE and RMSS perform very similarly at moderate contamination levels, RMSS significantly outperforms both PENSE and RBSS when the contamination level is high. This conclusion holds across all three SNR levels considered in our study.

Table 2 summarizes the variable selection performance, reporting the average Recall (RC) and Precision (PR) ranks for the six relevant methods across all simulation settings. RMSS consistently achieved the best average RC rank across all contamination levels, with PENSE

Table 1: Average and lowest MSPE rank of the eight methods over the nine SNR and sparsity level combinations for each contamination proportion.

	MSPE Rank					
	$\boldsymbol{\alpha = 0}$		$\boldsymbol{\alpha = 0.15}$		$\boldsymbol{\alpha = 0.3}$	
Method	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**
EN	5.3	6	5.9	6	4.6	5
PENSE	4.2	7	**1.0**	**1**	3.9	5
HuberEN	5.0	6	4.4	5	**2.4**	**4**
SparseLTS	7.9	8	3.3	4	7.0	8
RBSS	6.9	8	4.3	6	3.1	5
RMSS	**2.7**	**4**	**2.0**	**2**	**1.0**	**1**
RGLM	**1.1**	**2**	8.0	8	7.7	8
RF	2.9	5	7.0	7	6.3	7

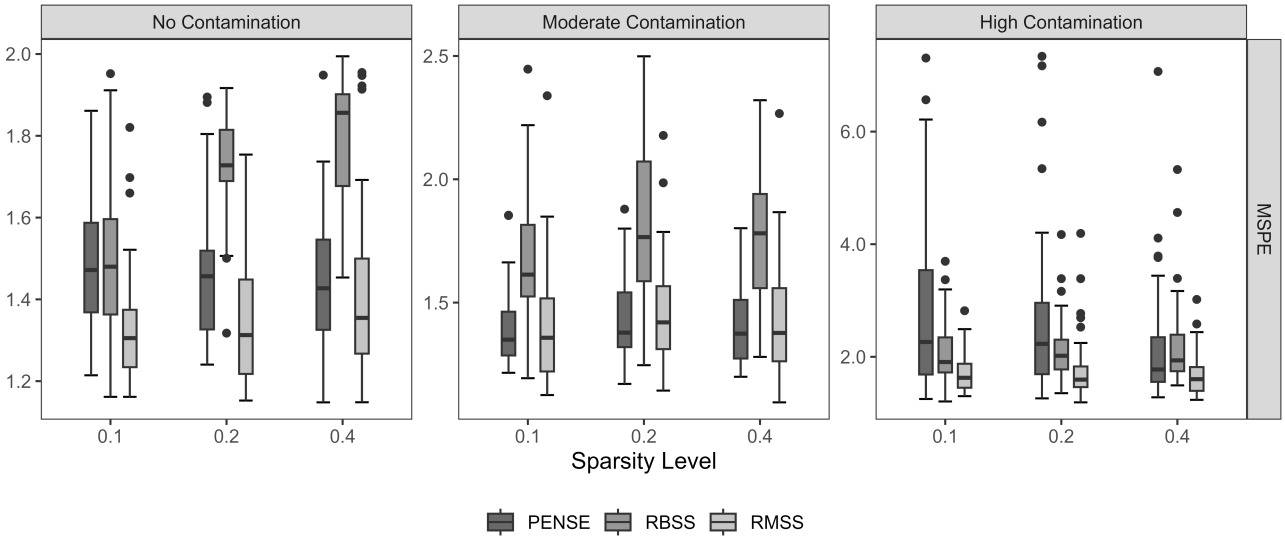

Figure 1: MSPE of PENSE, RBSS and RMSS over $N = 50$ random training sets over different contamination and sparsity levels for SNR $= 1$.

as the next best performer. In terms of PR, RBSS performed best, followed by RMSS in the contaminated scenarios.

Figure 2 provides a more detailed view for a moderate SNR. RMSS generally outperformed its competitors in RC, especially for lower-sparsity models ($\zeta = 0.1, 0.2$), a result that improved

further in experiments with more than $G = 10$ models. While RBSS consistently achieved the highest PR, this came at the cost of poor RC across all settings. In contrast, RMSS provided the best balance, combining high RC with high PR across all configurations of our simulation study. This demonstrates its ability to reliably identify true predictors while controlling for false positives, even in complex data with block correlation and regression outliers.

Table 2: Average and lowest RC and PR rank of the six methods over the nine SNR and sparsity level combinations for each contamination proportion.

	RC Rank						**PR Rank**					
	$\boldsymbol{\alpha = 0}$		$\boldsymbol{\alpha = 0.15}$		$\boldsymbol{\alpha = 0.3}$		$\boldsymbol{\alpha = 0}$		$\boldsymbol{\alpha = 0.15}$		$\boldsymbol{\alpha = 0.3}$	
Method	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**
EN	5.6	6	5.7	6	5.3	6	**2.2**	**4**	5.7	6	5.3	6
PENSE	2.6	**3**	**1.9**	**2**	**1.7**	**2**	5.0	5	3.0	3	3.0	3
HuberEN	**2.1**	4	5.1	6	5.7	6	3.7	**4**	5.3	6	5.7	6
SparseLTS	5.4	6	4.1	5	4.0	4	6.0	6	4.0	4	4.0	4
RBSS	3.9	4	3.1	**4**	3.0	**3**	**1.1**	**2**	**1.0**	**1**	**1.0**	**1**
RMSS	**1.4**	**2**	**1.1**	**2**	**1.3**	**2**	3.0	**4**	**2.0**	**2**	**2.0**	**2**

6.6 Computing Times

Table 3 reports the average computation time for each method across all simulation configurations. Notably, our RMSS implementation, which simultaneously generates $G = 10$ robust models and performs a 3D cross-validation, achieved a lower average computing time than PENSE. This is particularly efficient given that PENSE generates only a single model and performs CV over one parameter. The reported time for RMSS also includes the computation of RBSS, as it is generated concurrently by setting $u = G$.

To assess scalability for larger datasets, we examined how the computing time of RMSS varies with sample size n and number of predictors p. We used a challenging simulation scenario (low signal, high contamination, high proportion of true predictors) to provide a realistic performance benchmark. Figure 3 displays the run time as a function of $p \in \{250, \ldots, 1000\}$ for different sample sizes $n \in \{50, 100, 200\}$. Results show mean elapsed time across replications. As previously noted

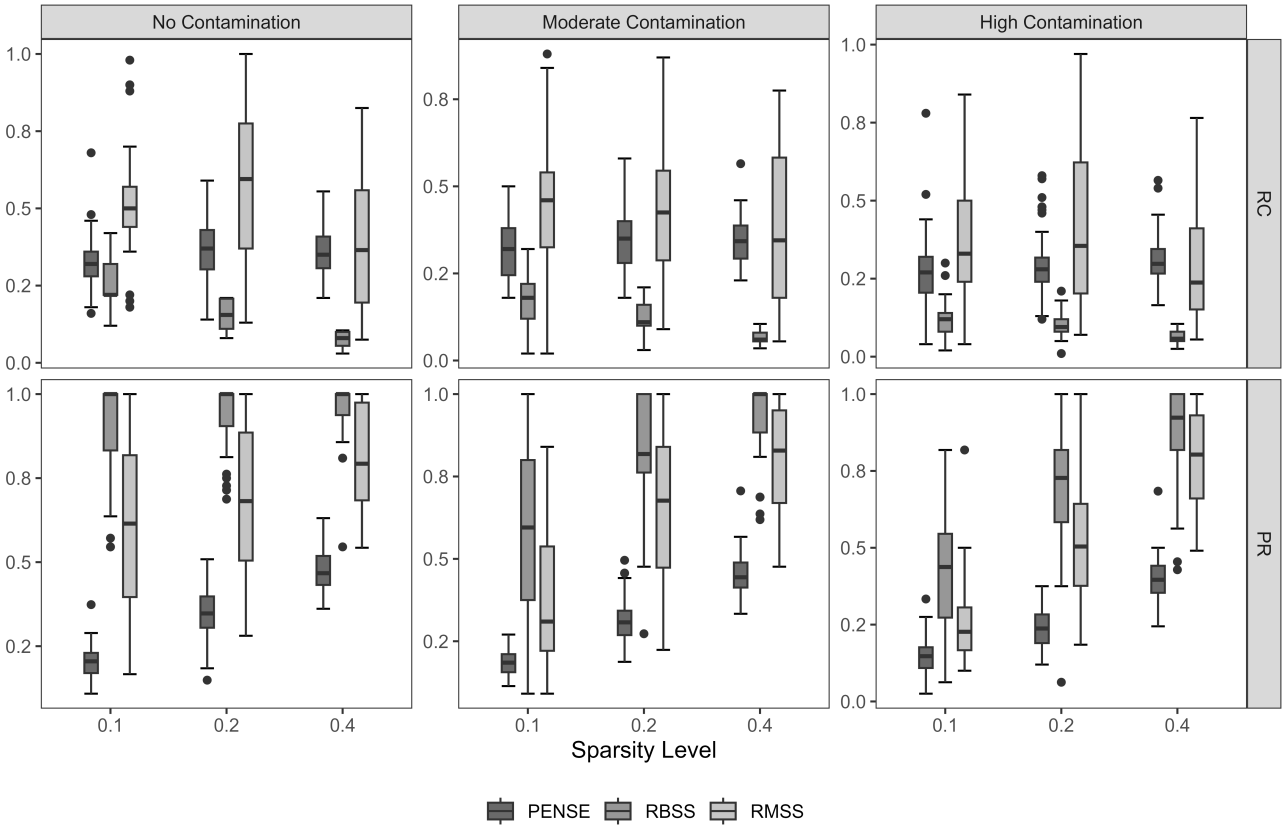

Figure 2: RC and PR of PENSE, RBSS and RMSS over $N = 50$ random training sets over different contamination and sparsity levels for SNR = 1.

Table 3: Computation time of `R` function calls for the methods in CPU seconds. CPU seconds are on a 2.1 GHz Intel Xeon Platinum 8468 processor in a machine running CentOS Linux 7.9 with 32 GB of RAM.

Method	EN	PENSE	HuberEN	SparseLTS	RMSS
Time	0.2	329.2	0.1	131.8	137.7

for other estimators based on mixed-interger optimization (Thompson, 2022; Bertsimas et al., 2016), RMSS remains computationally tractable up to dimensions as large as p = 1000, despite the expected increase in runtime with more predictors.

The reported times do not include the optional neighborhood search strategy described in the supplementary material since we found that it significantly increases the computational cost with only marginal gains in prediction accuracy. Overall, this analysis offers a guidance on the computational times expected for datasets of varying dimensions and highlights the practical

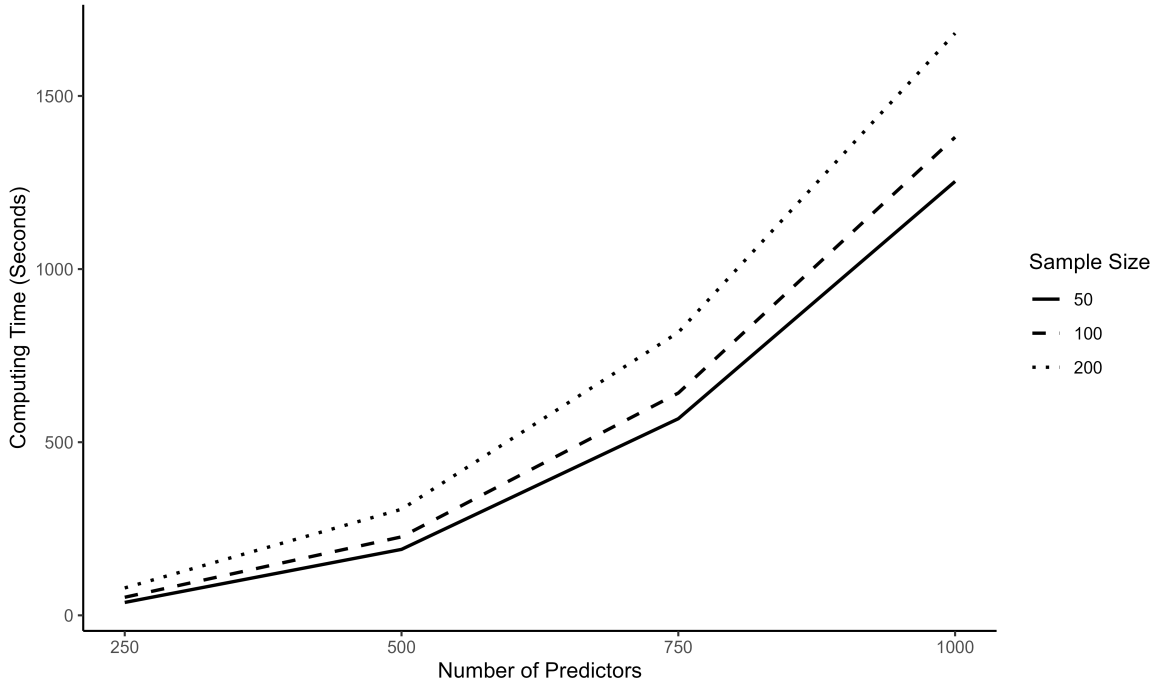

Figure 3: Computing time versus number of predictors ($p = 250, 500, 750, 1{,}000$) for different sample sizes ($n = 50, 100, 200$). Results show mean elapsed time across replications.

scalability of our algorithm.

7 Contamination of Bioinformatics and Cheminformatics Data

In biomedical sciences new deoxyribonucleic acid (DNA) microarray and ribonucleic acid (RNA) sequencing technologies allow for an increase in the type and volume of the genomics data collected (e.g., Byron et al., 2016). In chemistry, innovative microscopic technologies allow for the collection of data on the composition of chemical compounds and molecules. Outliers are not rare in these datasets. For example, in genomics, many datasets contain atypical observations obtained from samples with poor measurement quality or incorrect reads (e.g., Sangiovanni et al., 2019). In cheminformatics the topic of high-dimensionality and robustness has gained a lot of attention in recent years due to the emerging fields of computer-aided drug design and computational toxicology among others (see e.g. Basak and Vracko, 2022).

In this section we artificially contaminate real bioinformatic and cheminformatic datasets to

evaluate the performance of RMSS and the other methods in situations that mimic real applications. We also show that RMSS can uncover some predictors that may be relevant to predict the outcome of interest but that may not be picked up by single-model sparse robust methods.

7.1 Bioinformatics Data

We first analyze data from a study on Bardet-Biedl syndrome (BBS) by Scheetz et al. (2006), using a dataset from the `R` package `abess` (Zhu et al., 2022). The task is to predict the expression of the gene TRIM32 using $p = 500$ other genes from 120 mammalian-eye tissue samples. We randomly split the data 50 times into training ($n = 50$) and test sets ($n = 70$). We then contaminated 25% of samples in each training set by replacing the TRIM32 expression and 100 predictor genes with large, atypical values.

The relative MSPE and standard deviation (SD) in Table 4 show that RMSS achieved the best performance with the lowest variability. PENSE, the closest competitor, had an MSPE 7% higher than RMSS. As expected, the non-robust EN method failed completely, indicating the RMSS ensemble effectively handles data contamination.

Table 4: MSPE and SD relative to the best performance for the artificially contaminated BBS dataset ($p = 500$).

Method	EN	PENSE	HuberEN	SparseLTS	RBSS	RMSS
MSPE	> 25	1.07	1.17	1.09	1.74	**1.00**
SD	> 75	1.15	1.22	1.37	2.18	**1.00**

Beyond its predictive accuracy, RMSS provides a more comprehensive view of relevant predictors. This is valuable in high-dimensional settings due to the "multiplicity of good models" phenomenon (McCullagh and Nelder, 1989). Single-model methods may overlook important genes; indeed, on the BBS dataset, both PENSE and SparseLTS exhibited unstable variable selection, with no gene selected in over 50% of the runs. In contrast, RMSS identified 14 genes with over 50% selection frequency. It not only captured predictors favored by others but also consistently identified genes they missed, such as AAK1 (selected in 68% of RMSS runs vs. only 2% for PENSE and SparseLTS).

To evaluate performance in a higher dimensional setting, we conducted an additional analysis

using the original microarray data from the GSE5680 dataset (Barrett et al., 2005). After standard preprocessing, we created a feature set of $p = 1{,}275$ predictors by selecting the 50 genes most correlated with TRIM32 and including all their pairwise interactions. This biologically-motivated approach creates a challenging $p/n \approx 25$ scenario. We applied the same 25% contamination framework to this data.

The results in Table 5 show that RMSS continues to achieve excellent prediction performance. Despite the significant increase in dimensionality, RMSS maintains its robustness advantage over competing methods, confirming its applicability to modern genomic datasets where predictors greatly exceed the sample size.

Table 5: MSPE and SD relative to the best performance for the artificially contaminated BBS dataset with interactions ($p = 1,275$).

Method	EN	PENSE	HuberEN	SparseLTS	RBSS	RMSS
MSPE	> 50	1.08	1.94	1.21	1.35	**1.00**
SD	> 50	3.93	2.84	1.96	2.06	**1.00**

7.2 Cheminformatics Data

We next analyze the glass dataset from Lemberge et al. (2000), where the goal is to predict the concentration of the chemical compound Na2O from $p = 486$ frequency measurements obtained via electron probe X-ray microanalysis (EPXMA). Using 50 random splits, we created training sets ($n = 50$) and test sets ($n = 130$). We applied the same 25% contamination scheme to the training sets and evaluated performance on the uncontaminated test sets.

As shown in Table 6, which reports relative MSPE and its standard deviation, RMSS again achieved the best predictive performance by a large margin. PENSE, the next best method, had an MSPE 32% larger than RMSS. The individual models within the RMSS ensemble also performed well, achieving an MSPE comparable to the single-model RBSS solution. This strong predictive performance was observed for other chemical compounds in the full dataset as well.

The superior variable selection of RMSS was also evident. The ensemble consistently identified important frequency measurements that PENSE selected unreliably. This is likely because PENSE's elastic net penalty tends to select only one predictor from a correlated group, whereas

Table 6: MSPE and SD relative to the best performance for the artificially contaminated glass cheminformatics dataset. SparseLTS was omitted as it consistently returned an empty model.

Method	EN	PENSE	HuberEN	RBSS	RMSS
MSPE	> 100	1.32	8.46	2.45	**1.00**
SD	> 75	1.21	1.62	2.34	**1.00**

RMSS's ensemble structure can accommodate such predictors by assigning them to different models. This was clear in the results: the top predictor for RMSS (86% selection rate) was chosen by PENSE in only 12% of runs. Further, a feature PENSE selected in just 26% of runs was captured far more consistently by the RMSS ensemble (78% selection rate). This highlights the ability of RMSS to provide a more robust and complete feature set, particularly under multicollinearity.

8 Summary and Future Works

In this article, we introduce RMSS, a data-driven method to build an ensemble of sparse and robust models to predict a response of interest and select important predictors from moderately high-dimensional datasets possibly containing outlying observations. To the best of our knowledge, this is the first method proposed for this aim. The levels of sparsity, diversity and robustness of each model are driven directly by the data based on a CV criterion. We established the finite-sample breakdown point of RBSS and RMSS. To bypass the NP-hard computational complexity of RMSS, we developed a tailored computing algorithm with a local convergence property by leveraging recent developments in the ℓ_0-optimization literature. Our extensive numerical experiments on synthetic and real data demonstrate the excellent performance of RMSS relative to state-of-the-art sparse and robust methods in moderately high-dimensional prediction tasks when the data is also contaminated. We also showed how RMSS can potentially uncover important predictor variables that may be discarded by single-model sparse and robust methods.

Since RMSS can potentially uncover predictor variables that are not picked up by single-model methods, the addition of interaction terms may potentially further increase the competitive advantage of RMSS over single-model sparse and robust methods. For example, in the -omics sciences where interactions between genes or proteins may drive the outcome of interest. The empirical

performance of RMSS can be improved further by considering alternative ways to combine the models in the ensembles other than the simple model average we used in this article. Our work can be extended by considering other robust loss functions to build sparse robust models.

With the growing emphasis on interpretable statistical and machine learning algorithms in the literature and in real data applications, our proposal will potentially pave the way for the development of other robust ensemble methods. A potential bottleneck in this area of research is the high computational cost of such methods, thus new optimization tools will be needed to render such ensemble methods feasible in practice.

Conflict of Interests

The authors declare no potential conflict of interests.

Supplementary Materials

Supplementary Document: A PDF document containing detailed theoretical proofs for all propositions and the theorem for the finite-sample breakdown point, extended derivations for the robust stepwise selection algorithm, a full convergence analysis of the PSBGD algorithm, and additional details on the optional neighborhood search, combinatorial complexity, and simulation tuning parameters. (`RMSS_supplementary.pdf`, PDF file)

Code and Data for Reproducibility: A compressed archive containing all `R` scripts and data files necessary to reproduce the numerical results presented in the simulation studies and real-data applications. A detailed `README.pdf` file is included, which provides clear instructions on package dependencies, file organization, and how to run the scripts to generate the results for each section of the paper. (`RMSS_reproducibility.zip`, ZIP archive)

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control 19*(6), 716–723.

Alfons, A. (2021). robustHD: An R package for robust regression with high-dimensional data. *Journal of Open Source Software 6*(67), 3786.

Alfons, A., C. Croux, and S. Gelper (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 226–248.

Alqallaf, F., S. Van Aelst, V. J. Yohai, and R. H. Zamar (2009). Propagation of outliers in multivariate data. *The Annals of Statistics*, 311–331.

Barrett, T., T. O. Suzek, D. B. Troup, S. E. Wilhite, W.-C. Ngau, P. Ledoux, D. Rudnev, A. E. Lash, W. Fujibuchi, and R. Edgar (2005). Ncbi geo: mining millions of expression profiles—database and tools. *Nucleic acids research 33*(suppl_1), D562–D566.

Basak, S. C. and M. Vracko (2022). *Big Data Analytics in Chemoinformatics and Bioinformatics: With Applications to Computer-Aided Drug Design, Cancer Biology, Emerging Pathogens and Computational Toxicology*. Elsevier.

Bertsimas, D., A. King, and R. Mazumder (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics 44*(2), 813–852.

Biau, G., A. Fischer, B. Guedj, and J. D. Malley (2016). Cobra: A combined regression strategy. *Journal of Multivariate Analysis 146*, 18–28.

Breiman, L. (1996a). Bagging predictors. *Machine Learning 24*(2), 123–140.

Breiman, L. (1996b). Stacked regressions. *Machine Learning 24*(1), 49–64.

Breiman, L. (2001, October). Random forests. *Machine Learning 45*(1), 5–32.

Bühlmann, P. and B. Yu (2003). Boosting with the l2-loss: Regression and classification. *Journal of the American Statistical Association 98*(462), 324–339.

Byron, S. A., K. R. Van Keuren-Jensen, D. M. Engelthaler, J. D. Carpten, and D. W. Craig (2016). Translating rna sequencing into clinical diagnostics: opportunities and challenges. *Nature Reviews Genetics 17*(5), 257–271.

Chandra, R., L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald (2001). *Parallel programming in OpenMP*. Morgan Kaufmann.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Christidis, A. and G. Cohen-Freue (2023). *RMSS: Robust Multi-Model Subset Selection*. R package version 1.2.1.

Christidis, A. and G. Cohen-Freue (2025). *robStepSplitReg: Robust Stepwise Split Regularized Regression*. R package version 2.0.0.

Christidis, A.-A., S. V. Aelst, and R. Zamar (2024). Multi-model subset selection. *Computational Statistics and Data Analysis*. In press.

Christidis, A.-A., L. Lakshmanan, E. Smucler, and R. Zamar (2020). Split regularized regression. *Technometrics 62*(3), 330–338.

Cohen Freue, G. V., D. Kepplinger, M. Salibián-Barrera, and E. Smucler (2019). Robust elastic net estimators for variable selection and identification of proteomic biomarkers.

Donoho, D. L. and P. J. Huber (1983). The notion of breakdown point. *A festschrift for Erich L. Lehmann 157184*.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of Statistics 32*(2), 407–499.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), 1189–1232.

Friedman, J. H., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1.

Garside, M. (1965). The best sub-set in multiple regression analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 14*(2-3), 196–200.

Hastie, T., R. Tibshirani, and R. Tibshirani (2020). Best subset, forward stepwise or lasso? analysis and recommendations based on extensive comparisons. *Statistical Science 35*(4), 579–592.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*(8), 832–844.

Kepplinger, D. (2023). Robust variable selection and estimation via adaptive elastic net s-estimators for linear regression. *Computational Statistics & Data Analysis 183*, 107730.

Kepplinger, D., M. Salibián-Barrera, and G. Cohen Freue (2023). *pense: Penalized Elastic Net S/MM-Estimator of Regression*. R package version 2.2.2.

Khan, J. A., S. Van Aelst, and R. H. Zamar (2007a). Building a robust linear model with forward selection and stepwise procedures. *Computational Statistics & Data Analysis 52*(1), 239–248.

Khan, J. A., S. Van Aelst, and R. H. Zamar (2007b). Robust linear model selection based on least angle regression. *Journal of the American Statistical Association 102*(480), 1289–1299.

Lemberge, P., I. De Raedt, K. H. Janssens, F. Wei, and P. J. Van Espen (2000). Quantitative analysis of 16–17th century archaeological glass vessels using pls regression of epxma and μ-xrf data. *Journal of Chemometrics: A Journal of the Chemometrics Society 14*(5-6), 751–763.

Liaw, A. and M. Wiener (2002). Classification and regression by randomforest. *R News 2*(3), 18–22.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics 15*(4), 661–675.

Maronna, R. A. (2011). Robust ridge regression for high-dimensional data. *Technometrics 53*(1), 44–53.

Maronna, R. A., R. D. Martin, V. J. Yohai, and M. Salibián-Barrera (2019). *Robust statistics: theory and methods (with R)*. John Wiley & Sons.

Maronna, R. A. and R. H. Zamar (2002). Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics 44*(4), 307–317.

McCullagh, P. and J. A. Nelder (1989). Monographs on statistics and applied probability. *Generalized Linear Models 37*.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Raymaekers, J. and P. J. Rousseeuw (2021). Fast robust correlation for high-dimensional data. *Technometrics 63*(2), 184–198.

Raymaekers, J. and P. J. Rousseeuw (2024). Challenges of cellwise outliers. *Econometrics and Statistics*.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American statistical association 79*(388), 871–880.

Rousseeuw, P. J. and W. V. D. Bossche (2018). Detecting deviating data cells. *Technometrics 60*(2), 135–145.

Sangiovanni, M., I. Granata, A. S. Thind, and M. R. Guarracino (2019). From trash to treasure: detecting unexpected contamination in unmapped ngs data. *BMC bioinformatics 20*(4), 1–12.

Scheetz, T. E., K.-Y. A. Kim, R. E. Swiderski, A. R. Philp, T. A. Braun, K. L. Knudtson, A. M. Dorrance, G. F. DiBona, J. Huang, T. L. Casavant, et al. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences 103*(39), 14429–14434.

Shen, X., W. Pan, Y. Zhu, and H. Zhou (2013). On constrained and regularized high-dimensional regression. *Annals of the Institute of Statistical Mathematics 65*(5), 807–832.

Smucler, E. and V. J. Yohai (2017). Robust and sparse estimators for linear regression models. *Computational Statistics & Data Analysis 111*, 116–130.

Song, L., P. Langfelder, and S. Horvath (2013). Random generalized linear model: a highly accurate and interpretable ensemble predictor. *BMC Bioinformatics 14*(1), 5.

Thompson, R. (2022). Robust subset selection. *Computational Statistics & Data Analysis*, 107415.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodological) 58*(1), 267–288.

Ueda, N. and R. Nakano (1996). Generalization error of ensemble estimators. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, Volume 1, pp. 90–95. IEEE.

Welch, W. J. (1982). Algorithmic complexity: three np-hard problems in computational statistics. *Journal of Statistical Computation and Simulation 15*(1), 17–25.

Yi, C. (2017). *hqreg: Regularization Paths for Lasso or Elastic-Net Penalized Huber Loss Regression and Quantile Regression*. R package version 1.4.1.

Yi, C. and J. Huang (2017). Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics 26*(3), 547–557.

Zhang, J. and K. R. Coombes (2012). Sources of variation in false discovery rate estimation include sample size, correlation, and inherent differences between groups. *BMC Bioinformatics 13*(S13), S1.

Zhu, J., X. Wang, L. Hu, J. Huang, K. Jiang, Y. Zhang, S. Lin, and J. Zhu (2022). abess: a fast best-subset selection library in python and r. *The Journal of Machine Learning Research 23*(1), 9206–9212.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodological) 67*(2), 301–320.