Robust Multi-Model Subset Selection

Anthony-Alexander Christidis
Department of Statistics
University of British Columbia
anthony.christidis@stat.ubc.ca

Gabriela Cohen-Freue
Department of Statistics
University of British Columbia
gcohen@stat.ubc.ca

Abstract

Modern datasets in biology and chemistry are often characterized by the presence of a large number of variables, complex correlation structures between them, and outlying samples due to rare biological and chemical profiles. We propose 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 in high-dimensional prediction tasks. The degree to which the models are sparse, diverse and resistant to outliers is driven directly by the data using a cross-validation criterion. We establish the finite-sample breakdown point of the ensembles and the models that comprise them, and we develop a tailored computing algorithm to learn the ensembles 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.

Keywords: Robust methods; High-dimensional data; Ensemble methods; Multi-model optimization.

1 Introduction

With recent advances in technologies to collect and store data in many fields of science and engineering, there is an ever-increasing need for new predictive methods that can handle complex data. For example, advances in genomics technologies allow the simultaneous quantitation of thousands of genes from a patient's sample, revolutionizing the way that scientists can measure pathogenic processes or responses to therapies. Despite their potential to improve diagnostics methods in routine clinical care, these high-dimensional data still present significant challenges (e.g., Byron et al., 2016). Typical datasets contain more features than observations, outliers, complex correlation structures and unnecessary variables. Data with these characteristics are common in fields other than the biomedical sciences. For example, in cheminformatics, outlying data points due to measurement errors or rare chemical profiles may mask the identification of more subtle, yet predictive frequency measurements to accurately predict the concentration of a chemical compound (e.g., Sangiovanni et al., 2019). In this article, we propose a method to generate simultaneously a collection of sparse and diverse predictive models that are resistant to outliers and can be ensembled to improve the performance of single-model sparse and robust methods.

Regression regularized methods have been proposed in the literature to handle data with a large number of predictors relative to the number of samples and to select a subset of predictors to build predictive model, such as the least absolute shrinkage and selection operator (LASSO, Tibshirani, 1996) or the smoothly clipped absolute deviation (SCAD, Fan and Li, 2001) methods. The empirical performance and theoretical properties of these methods have been studied extensively (e.g., Hastie et al., 2019). However, depending on the loss function used, these method may be very sensitive to outliers, which may adversely affect their variable selection and prediction performances. To address this problem, robust statistical procedures typically replace traditional loss functions by robust loss functions that downweight the effect of contaminated samples (Maronna et al., 2019). In recent years, there have been many proposals that combine regularized and robust methods to obtain predictive models that are resistant to outliers. The resulting models have been successfully deployed in a variety of applications using high-dimensional biological and chemical data (Maronna, 2011; Alfons et al., 2013; Smucler and Yohai, 2017; Cohen Freue et al., 2019).

Ensemble methods can be used to generate and aggregate multiple diverse models, and 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 prominent 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. (2023) 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 similar to that achieved by many single-model sparse methods, and the ensembling of the small models outperforms state-of-the-art blackbox ensemble methods on synthetic as well as complex biological and chemical data. However, both the ensembles and the individual models that comprise them are not robust and are thus 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, robustness and diversity within each model are driven directly by the data. We establish the finite-sample breakdown point of these robust ensembles and the individual models that comprise them. We develop a tailored computational algorithm with attractive convergence properties to fit the models. RMSS is shown to outperform state-of-the-art sparse and robust methods in an extensive simulation study and when applied to artificially contaminated biological and chemical data. To the best of our knowledge, this is the first robust ensemble method proposed in the literature. In addition, the flexibility offered by our method can be particularly appealing for practitioners collecting and analyzing high-dimensional complex 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 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 $\mathbf{y} = (y_1, \dots, y_n)^T \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times p}$, and $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ 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 $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)^T \in \mathbb{R}^n$ 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 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 $t \leq \min(n-1, p)$ 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 (CV).

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 methods have been proposed that can be used when $p >> n$, 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 combining robust estimators of pairwise correlations and regression parameters. Penalizing the the least trimmed squares method (LTS, Rousseeuw, 1984) approach, Alfons et al. (2013) introduced the sparse LTS estimator that minimizes the LASSO-penalized sum of h smallest squared residuals, $h \leq \lfloor n/2 \rfloor$. Also inspired by LTS, Thompson (2022) introduced a robust best subset selection (RBSS) by combining the LTS loss with the ℓ_0-penalty for the vector of coefficients. The objective function becomes

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{i \in I} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta} \right)^2 \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\beta}\|_0 \leq t, \\ |I| \geq h, \end{cases} \tag{3}$$

where $I = \{1, \ldots, n\}$ and $|\cdot|$ is the cardinality operator.

Thompson (2022) established the finite-sample breakdown point of RBSS, and developed a computing algorithm. 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, and in the case of gradient boosting the models are fit on residuals rather than the original data.

To generate ensembles of sparse, accurate and diverse models, Christidis et al. (2020) and Christidis et al. (2023) 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 and control the degrees of sparsity and diversity directly, Christidis et al. (2023) introduced a multi-model subset selection (MSS) as a generalization of BSS in (2). The degree of sparsity of the models and diversity between them are chosen by CV and thus driven directly by the data. Despite the high prediction accuracy and interpretability of these ensembles and the models that comprise them, they are very sensitive outliers in the data.

3 Robust Multi-Model Subset Selection

In this Section we introduce our Robust Multi-Model Subset Selection estimator (RMSS) to build an ensemble of strong predictive models from a high-dimensional and sparse complex dataset containing outlying samples. RMSS aims to find $G \geq 2$ robust, sparse and diverse well-performing models that can also be combined into a highly accurate robust ensemble model.

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 $\boldsymbol{\beta}^g = (\beta_1^g, \beta_2^g, \dots, \beta_p^g)^T \in \mathbb{R}^p$ be the vector of coefficients of model g, and $\boldsymbol{\beta}_{j\cdot} = (\beta_j^1, \beta_j^2, \dots, \beta_j^G)^T \in \mathbb{R}^G$ be the vector of coefficients of predictor j across the G models. For a fixed number of models $G \geq 2$, RMSS solves the constrained optimization problem

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

where the subsets $I^{(g)} \subseteq I = \{1, \dots, n\}$, $1 \leq g \leq G$ indicate the subset of samples used to estimate model g. Similar to MSS, the parameters $t \leq \min(n-1, p)$ and $u \leq G$ are chosen by CV such that the degrees of sparsity of the models and diversity between them are data-driven. The parameter $h \leq n$ determines the number of smallest residuals used to fit each model. In our algorithm, this argument can be predetermined by the user or selected by CV. Unless there is some a priori knowledge of the degree of data contamination, we recommend to use CV.

Since only a subset of the predictor variables may be contaminated for any given sample, a nice feature of RMSS is that the individual models in the ensembles may be fit on different samples (i.e., different subsets $I^{(g)}$). RMSS may potentially use more samples relative to RBSS or other single-model sparse and robust methods, reducing the loss of information from the training data.

We now address some special cases of RMSS for different configurations of the tuning parameters. If $h = n$ in (4) it follows immediately that RMSS is equivalent to MSS for the same values of t and u. RMSS can also be seen as a generalization of RBSS.

Proposition 1 *If $u = G$ in (4), then there is no diversity among the individual models in RMSS and the solution to each of these models is the optimal solution to RBSS in (3) with sparsity and robustness parameters t and h, respectively.*

The proof of Proposition 1 provided in the supplementary material follows directly from the fact that if there is no restriction on the sharing of predictors. Thus, the minimum loss for each model is achieved by the RBSS optimal solution with the same tuning parameters t and h. Corollary 1 below follows immediately from Proposition 1.

Corollary 1 *If $u = G$ in (4), then if*

(I) $h = n$, RMSS is equivalent to BSS, and

(II) $t = p < n - 1$, RMSS is equivalent to LTS.

Since the tuning parameters t, u and h are chosen by CV, RMSS can easily adapt to data with different characteristics, e.g., data with very few predictors or without any outliers.

In this article, we generate ensembles using the simple model averaging method, where the coefficients of an 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 the 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 of RMSS ensembles, a standard robustness measure defined by Donoho and Huber (1983) that indicates the smallest fraction of contaminated observations needed to render the estimator meaningless. The mathematical definition of the finite-sample breakdown point is given in Definition 1 below.

Definition 1 *Let $(\mathbf{X}, \mathbf{y})$ be an uncontaminated sample of size n, and denote by $(\mathbf{X}_c, \mathbf{y}_c)$ the contaminated samples of $(\mathbf{X}, \mathbf{y})$ with $1 \leq m \leq n$ contaminated samples. Let $T(\mathbf{X}, \mathbf{y})$ be some estimator of data $(\mathbf{X}, \mathbf{y})$. The finite-sample breakdown point of $T(\mathbf{X}, \mathbf{y})$ is given by*

$$B\left(T|\mathbf{X}, \mathbf{y}\right) = \min_{1 \leq m \leq n} \left\{ \frac{m}{n} : \sup_{(\mathbf{X}_c, \mathbf{y}_c)} \left\| T\left(\mathbf{X}, \mathbf{y}\right) - T\left(\mathbf{X}_c, \mathbf{y}_c\right) \right\|_2 = \infty \right\}. \quad (5)$$

In Theorem 1 and Corollary 2, we establish the finite-sample breakdown point of RMSS ensembles and the individual models that comprise them, respectively. The proofs are provided in the supplementary material.

Theorem 1 *Let $T(\mathbf{X}, \mathbf{y})$ be the optimal value of the objective function of RMSS in (4). Then, $T(\mathbf{X}, \mathbf{y})$ has finite-sample breakdown point*

$$B\left(T|\mathbf{X}, \mathbf{y}\right) = \frac{n-h+1}{n}.$$

Corollary 2 *Let $T^g(\mathbf{X}, \mathbf{y})$ be the trimmed sum of squares of model g,*

$$T^g\left(\mathbf{X}, \mathbf{y}\right) = \sum_{i \in \hat{I}^{(g)}} \left(y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}^g\right)^2, \quad 1 \leq g \leq G,$$

where $\hat{I}^{(g)}$ and $\hat{\boldsymbol{\beta}}^g$ are the optimal subset of samples and vector of coefficients for model g in RMSS (4), respectively. Then $T^g(\mathbf{X}, \mathbf{y})$ has finite-sample breakdown point

$$B\left(T^g|\mathbf{X}, \mathbf{y}\right) = \frac{n-h+1}{n}, \quad 1 \leq g \leq G.$$

From Theorem 1 and Corollary 2, it follows that RMSS ensembles and the individual models that comprise them are resistant to up to $n-h$ contaminated samples. Since $h = n$ corresponds to the MSS ensemble method of Christidis et al. (2023), the breakdown point of MSS ensembles and the individual models that comprise them is $1/n$ and are thus not resistant to any contamination level. If the trimming parameter h is chosen by CV, the extent to which RMSS ensembles and their individual models are resistant to outliers is data-driven.

5 Computing Algorithm

The evaluation of every possible combination of predictors in RMSS is not feasible, even for a low-dimensional case in which predictors are not shared between models (see a combinatorics result in the supplementary material). Thus, we propose an algorithm to search over a three-dimensional grid of the tuning parameters h, t and u, reducing the computational cost of the CV.

We first center and scale the response $\mathbf{y}$ and the columns of the design matrix $\mathbf{X}$ using their medians and median absolute deviations, respectively. Estimated coefficients are then returned to their original scale. We then generate disjoint subsets of predictors for the particular case $u = 1$ (Section 5.1) and generate solutions for any combination of values in the three-dimensional grid of h, t and u (Section 5.2). We also developed a three-dimensional neighborhood search to sequentially improve incumbent solutions generated by our algorithm (see supplementary material). However, numerical experiments show only marginal improvements in prediction and variable selection performances. Nevertheless, this option is available in our software.

5.1 Initial Predictor Subsets

Algorithm 1 generalizes the robust forward stepwise regression algorithm of Khan et al. (2007a) to multiple models based on G disjoint subsets of predictors $J^{(g)} \subseteq J = \{1, \ldots, p\}$, $1 \leq g \leq G$. The proposed method is built upon the fact that stepwise forward search algorithm depends only the vector of sample correlations between the response and predictors and the sample correlation matrix of $\mathbf{X}$ (see lemmas to prove this claim in the supplementary material). Thus, Khan et al. (2007a) replaced these multivariate sample correlation and covariance estimators by robust estimators. In our implementation, we use the fast and robust estimates obtained via the state-of-the-art "Detect Deviating Cells" (DDC) method of Rousseeuw and Bossche (2018), which can be scaled to high-dimensional settings by exploiting the "g-product" moment trick (Raymaekers and Rousseeuw, 2021). We denote the resulting estimates $\hat{\mathbf{r}}_{\mathbf{y}}$ and $\hat{\mathbf{\Sigma}}$.

Algorithm 1 Robust Multi-Model Stepwise Regression

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

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

1: Repeat the following steps until $\gamma^* \geq \gamma$ or $T^{(g)} = \text{TRUE}$ for all $1 \leq g \leq G$:
    1.1: For each model $g$ satisfying $T^{(g)} = \text{FALSE}$:
        1.1:1: Identify candidate predictor $j^{(g)}$ that maximizes the decrease in RSS when combined
               with the variables in $J^{(g)}$.
        1.1:2: Calculate the $p$-value $\gamma^{(g)}$ of predictor $j^{(g)}$ in the enlarged model via the $F$-test for
               nested model comparison.
        1.1:3: If $\gamma^{(g)} \geq \gamma$ set $T^{(g)} = \text{TRUE}$.
    1.2: Identify the unsaturated model $g^*$ with the smallest $p$-value $\gamma^{(g^*)}$.
    1.3: If $\gamma^{(g^*)} < \gamma$:
        1.3:1: Update the set of predictors for model $g^*$: $J^{(g^*)} = J^{(g^*)} \cup \{j^{(g^*)}\}$.
        1.3:2: Update the set of candidate predictors $J = J \setminus \{j^{(g^*)}\}$.
        1.3:3: If $|J^{(g^*)}| = n - 1$, set $T^{(g^*)} = \text{TRUE}$.
2: Return the sets of model predictors $J^{(g)}$, $1 \leq g \leq G$.
```

Initially, all models are empty, i.e., they do not contain any of the predictor variables. At each iteration of the algorithm, the candidate predictor variable that maximizes the improvement in goodness-of-fit of each unsaturated model is identified and its p-value is computed via the partial F-test for nested model comparison. If the smallest p-value falls below the significance threshold $\gamma \in (0,1)$ then the model corresponding to the smallest p-value is updated by adding its optimal candidate predictor to its set of model predictors. This predictor is also removed from the set of candidate predictors for subsequent iterations. The algorithm iterates until each model either contains $n-1$ predictor variables or is devoid of any statistically significant candidate predictor.

5.2 Computing an Initial Grid of Solutions

To develop an efficient computing algorithm to generate solutions for RMSS, we can recast (4) in its equivalent form using auxiliary variables $\boldsymbol{\eta}^g = (\eta_1^g, \ldots, \eta_n^g)^T \in \mathbb{R}^n$, $1 \leq g \leq G$, given by

$$\min_{\boldsymbol{\beta}^1,\ldots,\boldsymbol{\beta}^G \in \mathbb{R}^p} \sum_{g=1}^{G} \mathcal{L}_n\left(\boldsymbol{\beta}^g, \boldsymbol{\eta}^g | \mathbf{y}, \mathbf{X}\right) \quad \text{subject to} \quad \begin{cases} \|\boldsymbol{\beta}^g\|_0 \leq t, & 1 \leq g \leq G, \\ \|\boldsymbol{\beta}_{j\cdot}\|_0 \leq u, & 1 \leq j \leq p, \\ \|\boldsymbol{\eta}^g\|_0 \leq n-h, & 1 \leq g \leq G. \end{cases} \tag{6}$$

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

$$\mathcal{L}_n\left(\boldsymbol{\beta}, \boldsymbol{\eta} | \mathbf{y}, \mathbf{X}\right) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\eta}\|_2^2 . \tag{7}$$

It can be proved that the gradients of $\mathcal{L}_n$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$, $\nabla_{\boldsymbol{\beta}}\mathcal{L}_n(\boldsymbol{\beta}, \boldsymbol{\eta}|\mathbf{y}, \mathbf{X})$ 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. The proofs are relayed to the supplementary material.

Essential to our computing algorithm are two projection operators which we define below.

Definition 2 *For any vector $v \in \mathbb{R}^p$ and scalar $t \in \mathbb{R}$ the projection operator $\mathcal{P}(v;t)$ is defined as*

$$\mathcal{P}\left(v;t\right) \in \underset{\substack{w \in \mathbb{R}^p \\ \|w\|_0 \leq t}}{\arg\min} \|w - v\|_2^2 . \tag{8}$$

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

$$\mathcal{Q}\left(v; t, S\right) \in \underset{\substack{w \in \mathbb{R}^p \\ \|w\|_0 \leq t \\ \{j \in J : w_j \neq 0\} \subseteq S}}{\arg\min} \quad \|w - v\|_2^2 \, . \tag{9}$$

The operator $\mathcal{P}(v; t)$ retains the t largest elements in absolute value of the vector v, while the operator $\mathcal{Q}(v; t, S)$ retains the t largest elements in absolute value of the vector v that belong to the set S. Note that both $\mathcal{P}(v; t)$ and $\mathcal{Q}(v; t, S)$ are set-valued maps since more than one possible permutation of the indices $J = \{1, \ldots, p\}$ and $\{j \in J : j \in S\}$ may exist. Moreover, let $S^{(g)}$ be the subsets of predictors that are used in at most $u - 1$ models excluding model g,

$$S^{(g)} = \left\{ j \in J : \sum_{\substack{h=1 \\ h \neq g}}^{G} \mathbb{I}\left(j \in J^{(h)} \right) \leq u - 1 \right\}, \tag{10}$$

where $J^{(g)} = \{j \in J : \hat{\boldsymbol{\beta}}_j^g \neq 0\}$ as defined in Section 5.1 and the vector $\hat{\boldsymbol{\beta}}^g = (\hat{\beta}_1^g, \ldots, \hat{\beta}_p^g)^T \in \mathbb{R}^p$ contains the current coefficient estimates for model g, $1 \leq g \leq G$. Lastly, denote $\mathbf{X}_S \in \mathbb{R}^{n \times |S|}$ the submatrix of $\mathbf{X}$ with column indices $S \subseteq J = \{1, \ldots, p\}$.

In Algorithm 2, we outline the steps to perform projected subset block gradient descent (PS-BGD) given starting values $(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g)$, $1 \leq g \leq G$, and generate solutions for RMSS with fixed parameters t, u and h. The algorithm alternates between updates of the model parameters $\hat{\boldsymbol{\beta}}^g$ and $\hat{\boldsymbol{\eta}}^g$ one model at a time until convergence is achieved. The convergence of the iterative procedure in step 2 of Algorithm 2 is established below (see the proof in the supplementary material).

Proposition 2 *For each model g in (6), step 2 of Algorithm 2 generates a converging sequence for the pair ($\hat{\boldsymbol{\beta}}^g$, $\hat{\boldsymbol{\eta}}^g$) and the inequalities*

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

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

Algorithm 2 generates solutions for fixed tuning parameters t, u and h given starting values. 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 $T = \{t_1, \ldots, t_q\}$, $U = \{1, \ldots, G\}$ and $H = \{h_1, \ldots, h_r\}$, 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 $(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g)$, $1 \leq g \leq G$, tuning parameters $t$, $u$ and $h$, and tolerance parameter $\epsilon > 0$.

1: Initialize the sets of model predictors $J^{(g)} = \{j \in J : \tilde{\boldsymbol{\beta}}_j^g \neq 0\}$, $1 \leq g \leq G$.

2: Repeat the following steps for each model $g$, $1 \leq g \leq G$:
    2.1: Update the the set of allowed predictor $S^{(g)}$ via (10) and the Lipschitz constant $\ell_{\boldsymbol{\beta}^{(g)}} = 2\|\mathbf{X}_{S^{(g)}}^T \mathbf{X}_{S^{(g)}}\|_2$.
    2.2: Perform one update for each of the current estimates $(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g)$ via
         $\hat{\boldsymbol{\beta}}^g \in \mathcal{Q}\left(\tilde{\boldsymbol{\beta}}^g - \frac{1}{L_{\boldsymbol{\beta}^{(g)}}} \nabla_{\boldsymbol{\beta}} \mathcal{L}_n\left(\boldsymbol{\beta}, \hat{\boldsymbol{\eta}}^g | \mathbf{y}, \mathbf{X}\right)\Big|_{\boldsymbol{\beta}=\tilde{\boldsymbol{\beta}}^g}; S^{(g)}, t\right)$
         $\hat{\boldsymbol{\eta}}^g \in \mathcal{P}\left(\tilde{\boldsymbol{\eta}}^g - \frac{1}{L_{\boldsymbol{\eta}}} \nabla_{\boldsymbol{\eta}} \mathcal{L}_n\left(\hat{\boldsymbol{\beta}}^g, \boldsymbol{\eta} | \mathbf{y}, \mathbf{X}\right)\Big|_{\boldsymbol{\eta}=\tilde{\boldsymbol{\eta}}^g}; n-h\right)$
         with $L_{\boldsymbol{\beta}^{(g)}} \geq \ell_{\boldsymbol{\beta}^{(g)}}$ and $L_{\boldsymbol{\eta}} \geq \ell_{\boldsymbol{\eta}}$ until $\mathcal{L}_n(\tilde{\boldsymbol{\beta}}^g, \tilde{\boldsymbol{\eta}}^g | \mathbf{y}, \mathbf{X}) - \mathcal{L}_n(\hat{\boldsymbol{\beta}}^g, \hat{\boldsymbol{\eta}}^g | \mathbf{y}, \mathbf{X}) \leq \epsilon$.
    2.3: Update the model predictors $J^{(g)} = \{j \in J : \hat{\beta}_j^g \neq 0\}$ and compute the set of model samples $I^{(g)} = \{i \in I : \hat{\eta}_i^g = 0\}$.
    2.4: Compute the final model coefficients
         $\hat{\boldsymbol{\beta}}^g = \underset{\substack{\boldsymbol{\beta} \in \mathbb{R}^p \\ \beta_j = 0, j \notin J^{(g)}}}{\arg\min} \sum_{i \in I^{(g)}} \left(y_i - \mathbf{x}_i^T \boldsymbol{\beta}\right)^2, \quad 1 \leq g \leq G.$

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

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 $T = \{t_1, \ldots, t_q\}$, $U = \{1, \ldots, G\}$ and $H = \{h_1, \ldots, h_r\}$, and tolerance parameter $\epsilon > 0$.

1: Use Algorithm 1 initialize the sets of model predictors $J_{\text{INIT}}^{(g)}$, $1 \leq g \leq G$, and compute the initial model coefficients
   $\hat{\boldsymbol{\beta}}_{\text{INIT}}^g = \underset{\substack{\boldsymbol{\beta} \in \mathbb{R}^p \\ \beta_j = 0, j \notin J_{\text{INIT}}^{(g)}}}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$

2: For each combination of $t \in T$ and $h \in H$:
    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 $(\hat{\boldsymbol{\beta}}_{\text{INIT}}^g, \mathbf{0}_n)$ 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 $t \in T$, $u \in U$ and $h \in H$.
```

5.3 Selection of Tuning Parameters

We use 5-fold CV to select the final combination of t, u and h from the grids of candidates T, U and H. Since the test folds may contain outliers, we use the fast and robust τ-estimator of location (Maronna and Zamar, 2002) on the prediction residuals of the test folds for each combination of t, u and h. Our final selection is the combination with the smallest robust location estimate.

The fine grids $T = \{1, \ldots, n-1\}$ and $H = \{\lfloor n/2 \rfloor + 1, \ldots, n\}$ may be used for the sparsity and robustness parameters, respectively. But less dense grids can also be used to speed up computation. Oftentimes, a priori information is available about the level of contamination of the data which may guide the choice for H. Our method has the flexibility to choose h by CV or set it to a fix value. We illustrate this feature in the simulation section. In general, the grid $U = \{1, \ldots, G\}$ should be fixed since it is required for the computation for solutions in Algorithm 3.

5.4 Software

The implementation of robust multi-model stepwise regression as outlined in Algorithm 1 is available on CRAN (R Core Team, 2022) in the `R` package `robStepSplitReg` (Christidis and Cohen-Freue, 2023b). Our implementation to fit RMSS ensembles via Algorithms 2 - 3 is also available on CRAN in the `R` package `RMSS` (Christidis and Cohen-Freue, 2023a), which can also generate RBSS models by setting $G = 1$ in the appropriate argument. The source code of `RMSS` is written in `C++` for optimization purposes and multithreading via OpenMP (Chandra et al., 2001) is available in the package to further speed up computations.

6 Simulations

In this Section, we investigate the performance of RMSS against robust and sparse methods in an extensive simulation study where the data is contaminated with leverage points, i.e., samples 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 in bioinformatics and cheminformatics (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 (1), where the $\mathbf{x}_i \in \mathbb{R}^p$ are multivariate normal with zero mean and correlation matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$, and the ϵ_i are standard normal. To generate disjoint blocks of correlated of active predictors (i.e., with nonzero coefficients) we use 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. We set the number of blocks of correlated predictors so that each block is comprised of 25 predictors, the sample size $n = 50$ and the number of predictors $p = 500$. Based on our preliminary numerical experiments, alternative choices for n and p where $p \gg n$ lead to similar conclusions. For each p, we consider the proportion of active variables equal to $\zeta \in \{0.1, 0.2, 0.4\}$. 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 $m = \lfloor \tau n \rfloor$ 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 $\Theta_i \sim \mathcal{N}(\mathbf{0}_\mathbf{p}, 0.01 \times \mathbf{I}_\mathbf{p})$ and $\mathbf{a} = \tilde{\mathbf{a}} - (1/p)\tilde{\mathbf{a}}^T \mathbf{1}_\mathbf{p}$ where $\mathbf{I}_\mathbf{p}$ is the p-dimensional identity matrix, $\mathbf{0}_\mathbf{p} = (0, \ldots, 0)^T \in \mathbb{R}^p$, $\mathbf{1}_\mathbf{p} = (1, \ldots, 1)^T \in \mathbb{R}^p$, 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 the contaminated observations. Our preliminary experiments showed that the effect of k_{lev} on estimators examined was almost the same for any $\mathrm{k}_{\text{lev}} > 1$, hence we fixed $\mathrm{k}_{\text{lev}} = 2$. We also found that the position the vertical outliers affects the estimators much more, and that the performance of non-robust estimators degraded significantly for any $\mathrm{k}_{\text{slo}} \geq 100$, hence we fixed $\mathrm{k}_{\text{slo}} = 100$. We consider the contamination proportions of $\tau = 0$ (no contamination), $\tau = 0.15$ (moderate contamination) and $\tau = 0.3$ (high contamination).

6.3 Methods

Our simulation study compares the prediction and variable selection accuracy of five 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**), proposed by (Thompson, 2022), with `RMSS` package.

6. Robust Multi-Model Subset Selection (**RMSS**) with $G = 10$ models, proposed in this paper, with `RMSS` package.

Details about the selection of their tuning parameters of each estimator is given in the supplementary material. To reduce the computational cost of RMSS in our extensive simulation study, we use the candidate grid $T = \{0.3n, 0.4n, 0.5n\} = \{15, 20, 25\}$ for the sparsity tuning parameter. We use a grid of values $H = \{(1 - (\tau + 0.1))n, (1 - (\tau + 0.05))n, (1 - \tau)n\}$ where $\tau \in \{0, 0.15, 0.3\}$ and determine the robustness tuning parameter h by CV. We use the default grid $U = \{1, \dots, G\}$ with $G = 10$ for the diversity parameter as required by our computing algorithm. RBSS is computed at the same time as RMSS in a single function call of the `RMSS` package by fixing $u = G$ (see Proposition 1). While better empirical results may be obtained with RMSS when combined with a larger number of models and more refined grids for the tuning parameters, we find that even with our suboptimal settings RMSS is competitive with state-of-the-art sparse and robust methods. To select tuning parameters we use 10-fold CV for EN and 5-fold CV for the robust methods.

6.4 Performance Measures

For each combination of the sparsity parameter ζ, SNR and contamination level τ, we randomly generate $N = 50$ training sets and a large (uncontaminated) independent test set of size $r = 2{,}000$. In each replication of a particular simulation configuration, we fit the methods on the training sets and we compute the MSPE using the independent test set. The reported MSPEs are relative to the variance of the irreducible error σ^2, i.e., the best possible result is 1. We also compute the recall (RC) and precision (PR) which are 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 regression coefficients, respectively. We use the ensemble fit $\bar{\boldsymbol{\beta}}$ for RMSS. RC and PR range between 0 and 1 and large values are desirable.

6.5 Results

In Table 1, we report the average and lowest MSPE rank of the six methods over the nine possible combinations of the SNR and sparsity level for each contamination proportion. The top performance in each column of a table are highlighted in bold fonts.

In the $\tau = 0$ (no contamination) case, RMSS consistently outperformed the non-robust EN and achieved the best overall rank. RMSS was also very stable in the no contamination case and never ranked worse than second place among the six methods. This demonstrates the ability of RMSS to adapt to the level of contamination of the data. In the $\tau = 0.15$ (moderate contamination) case PENSE and RMSS traded the top rank over every possible combination of SNR and sparsity level, with PENSE achieving the highest rank more often. However in the $\tau = 0.30$ (high contamination) case, PENSE's performance deteriorated while RMSS achieved the top rank in every possible configuration of the simulation. RBSS was not competitive with RMSS and PENSE, except in the high contamination case where it achieved the second best average rank behind RMSS.

In Figure 1, we plot the MSPE of PENSE, RBSS and RMSS over the $N = 50$ random training sets over different contamination and sparsity levels for SNR $= 1$ (moderate signal). It is evident that in the moderate contamination case PENSE and RMSS perform very similarly for all sparsity level, while for the high contamination case RMSS significantly outperforms PENSE and RBSS for any sparsity level. Similar conclusion are obtained for all three SNRs considered.

In Table 2, we report the average and lowest RC and PR rank of the six methods over the nine possible combinations of the SNR and sparsity level for each contamination proportion. RMSS achieved the best average RC rank over all contamination levels, while PENSE was the second best performing method over all contamination levels. In terms of PR, RBSS was the best performing method. In particular, when the data was contaminated as RBSS always had the best PR while RMSS always had the second best PR.

In Figure 2, we plot the RC and PR of PENSE, RBSS and RMSS over the $N = 50$ random training sets over different contamination and sparsity levels for a moderate signal. RMSS with

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

	MSPE Rank					
	$\tau = 0$		**$\tau = 0.15$**		**$\tau = 0.3$**	
Method	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**
EN	3.2	4	6.0	6	4.7	5
PENSE	2.4	5	**1.2**	**2**	4.1	5
HuberEN	3.2	4	4.4	5	2.8	4
SparseLTS	5.7	6	3.3	4	6.0	6
RBSS	5.1	6	4.2	5	2.4	4
RMSS	**1.3**	**2**	1.8	**2**	**1.0**	**1**

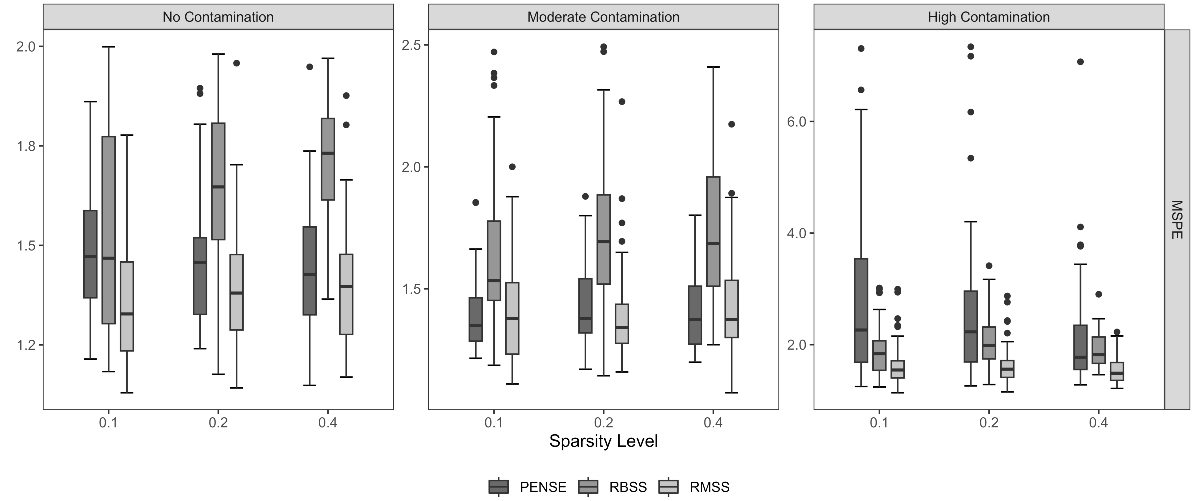

Figure 1: MSPE of PENSE, RBSS and RMSS over $N = 50$ random training sets over different contamination and sparsity levels for SNR = 1.

$G = 10$ models outperformed PENSE and RBSS in terms of RC over all contamination levels when $\zeta = 0.1$ and 0.2, and was competitive with PENSE when $\zeta = 0.4$. In our numerical experiments, RMSS achieved even superior RC when we used more than $G = 10$ models. RBSS generally had the best performance in terms of PR. However, the RC of RBSS was subpar over any combination of sparsity, contamination and SNR level. RMSS combined a high RC with a high PR which remained stable across all configurations of our simulation study, demonstrating a steady desirable performance in a simulation that combined a high-dimensional block correlation structure with regression outliers. The advantageous performance of RMSS in terms of RC and PR was also observed in the low and high SNR scenarios (see results in the supplementary material).

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**					
	$\tau = 0$		$\tau = 0.15$		$\tau = 0.3$		$\tau = 0$		$\tau = 0.15$		$\tau = 0.3$	
Method	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**	**Avg**	**Low**
EN	5.4	6	5.7	6	5.3	6	2.1	4	5.7	6	5.3	6
PENSE	2.3	**3**	1.8	**2**	1.9	**2**	5.0	5	3.0	3	3.0	3
HuberEN	2.3	**3**	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	4.1	5	3.1	4	3.3	3	**1.2**	**2**	**1.0**	**1**	**1.0**	**1**
RMSS	**1.7**	**3**	**1.2**	**2**	**1.1**	**2**	3.0	4	2.0	2	2.0	2

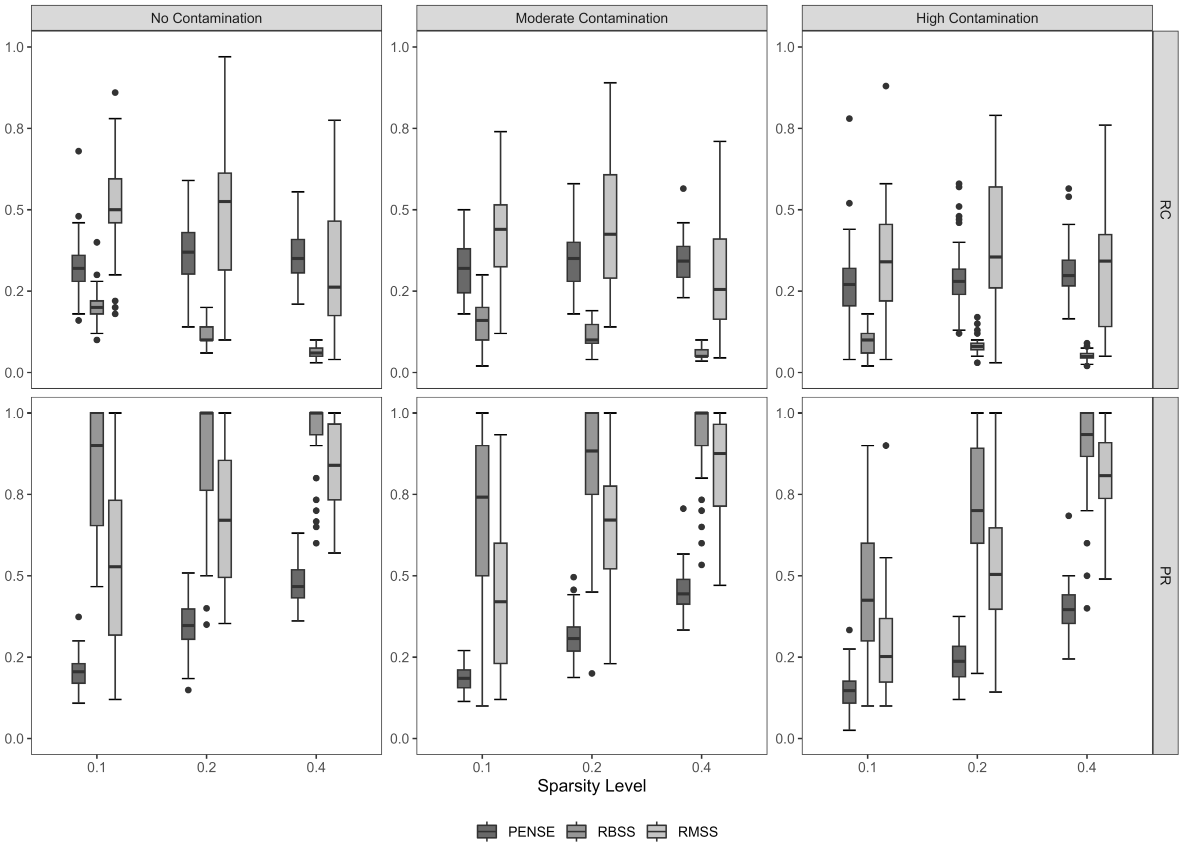

Figure 2: RC and PR of PENSE, RBSS and RMSS over $N = 50$ random training sets over different contamination and sparsity levels for SNR = 1.

6.6 Computing Times

The average computing times of the R function calls over all configurations of our simulation is given in Table 3. The time of RMSS also includes the computation of RBSS since the latter is simultaneously computed by setting $u = G$ in (4). Our implementation of RMSS generates $G = 10$ sparse and robust models simultaneously and still achieved a lower average computing time than PENSE, which generates a single sparse and robust model, with a smaller number of initial estimates than the default in its R implementation. Moreover, RMSS must perform CV over three tuning parameters compared to only one for all the single-model sparse and robust methods.

The computing time of RMSS would increase significantly if it uses a neighborhood search strategy. In general, we find that the neighborhood search improves the solutions $\hat{\boldsymbol{\beta}}^g$ $(1 \leq g \leq G)$ in terms of variable selection and minimizing the objective function (4) but not in terms of prediction.

Table 3: Computation time of R function calls for the methods in CPU seconds. CPU seconds are on a 2.7 GHz Intel Xeon processor in a machine running Linux 7.8 with 125 GB of RAM.

Method	EN	PENSE	HuberEN	SparseLTS	RMSS
Time	0.1	281.8	0.1	169.5	239.7

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. Many modern datasets with biological or chemical information are high-dimensional and display some form of data contamination. 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). The analysis of such dataset require statistical tools that can handle a large number of measurements and potentially contaminated samples. 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

In a study analyzing the genetic basis of Bardet-Biedl syndrome (BBS), Scheetz et al. (2006) performed mutation and functional studies and identified TRIM32 (tripartite motif-containing protein 32) as a gene whose expression highly correlates with the incidence of BBS. The R package abess (Zhu et al., 2022) contains a dataset with the expression of TRIM32 and $p = 500$ genes for 120 mammalian-eye tissue samples, which is a subset of the original dataset analyzed by Scheetz et al. (2006). The $p = 500$ genes were selected from the 18,976 available genes based on their marginal correlation with TRIM32 and are used to predict the expression of TRIM32.

The normalized gene expression levels in the uncontaminated dataset are all below 10 in magnitude. We randomly split the samples $N = 50$ times in a training set of size $n = 50$ and a test set of size $m = 70$. We contaminate 25% of the samples of each training set by replacing the expression of TRIM32 and 100 randomly selected predictor genes with a normal random variable with mean 25 and standard deviation 1. We evaluate the MSPE of the same six methods used in Section 6 on the uncontaminated test set. For RMSS we fix $h = \lfloor 0.75n \rfloor = 37$ but keep the grid $T = \{0.3n, 0.4n, 0.5n\} = \{15, 20, 25\}$ for the sparsity parameter. For the other methods we use the same configurations as in Section 6.

The MSPE and standard deviation (SD) of the MSPE reported in Table 5 are relative to the lowest value attained by the six methods, thus the best possible value is 1. The best performance for each measure is highlighted in bold fonts. RMSS achieved the best performance in terms of MSPE with the lowest MSPE variability. SparseLTS was the closest competitor but its MSPE was still 7% higher than the MSPE of RMSS. The individual $G = 10$ models of RMSS also achieved a high prediction accuracy with an average MSPE only 10% higher than the MSPE of RBSS. This observation indicates that each individual model correctly identified the outlying samples (see Corollary 2). As expected, the EN completely deteriorated with the addition outliers to the data.

Table 4: MSPE and SD of the six methods relative to the best performance for the artificially contaminated BBS bioinformatic dataset.

Method	EN	PENSE	HuberEN	SparseLTS	RBSS	RMSS
MSPE	> 25	1.11	1.26	1.07	1.77	**1.00**
SD	> 100	1.40	1.64	1.26	4.11	**1.00**

Beyond the good predictive performance of RMSS, the ensembles can potentially uncover genes that are not identified by the other methods. In particular, in the presence of high-dimensional data there are multiple models comprised of different subsets of predictors that can each achieve a high prediction accuracy. This phenomenon is known as the "the multiplicity of good models" in the statistical literature (see relevant discussions in McCullagh and Nelder, 1989). Thus single-model sparse and robust methods may potentially discard important genes from the decision-making process. On the BBS dataset, no gene was selected more than 50% of the time by PENSE or SparseLTS over the $N = 50$ random training sets, while 30 genes were selected more than 50% of the time by RMSS. Moreover, the genes most often selected by PENSE and SparseLTS were often selected by RMSS. For example, PENSE selected gene at probe `13704291_at` most often, and this gene was selected the same number of times by RMSS. Conversely, the genes most often selected by RMSS were seldom selected by PENSE or SparseLTS. For example, the gene at probe `1374809_at` was selected 72% of the time by RMSS and only 2% and 4% of the time by PENSE and SparseLTS, respectively.

7.2 Cheminformatics Data

We analyze the glass dataset from Lemberge et al. (2000) for which the goal is to predict the concentration of the chemical compound Na2O based on its frequency measurements obtained from an electron probe X-ray microanalysis (EPXMA). After removing variables with little variation, the dataset is comprised of $p = 486$ frequency measurements for $n = 180$ samples. We split the data into $N = 50$ training sets of size $n = 50$ and test sets of size $m = 130$. We contaminate the training sets in the same way as we did for the BBS data, and compute the MSPE of the same methods using the uncontaminated test sets.

In Table 4, we report the MSPE and SD of the MSPE of the methods relative to the best performing method. RMSS again achieved the best MSPE by far as SparseLTS coming was the second method with an MSPE 54% larger. RMSS was also the most stable method in terms of prediction accuracy. The $G = 10$ individual models of RMSS achieved an MSPE similar to RBSS (only 3% larger). The good predictive performance of RMSS is not restricted to the chemical compound Na2O and may be observed in more compounds available in Lemberge et al. (2000).

RMSS also uncovered frequency measurements that may be relevant to predict the concentration of Na2O that were missed by PENSE and SparseLTS. The frequency measurement most often selected by RMSS (92% of the time over the $N = 50$ random training sets) was only selected 2% of the time by both PENSE and SparseLTS. On the other hand, the frequency measurements

Table 5: MSPE and SD of the six methods relative to the best performance for the artificially contaminated glass cheminformatic dataset.

Method	EN	PENSE	HuberEN	SparseLTS	RBSS	RMSS
MSPE	> 150	1.90	10.91	1.54	2.58	**1.00**
SD	> 300	2.62	7.04	1.28	3.86	**1.00**

most often selected by PENSE and SparseLTS were often selected by RMSS. In fact, the frequency measurement most often selected by PENSE was selected only 30% of the time, and this same frequency measurement was selected 90% of the time by RMSS.

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 high-dimensional datasets possibly containing outlying observations. To the best of our knowledge, this is the first method proposed for this aim. The degree to which the models are sparse, diverse and robust is driven directly by the data based on a CV criterion. We established the finite-sample breakdown point of the ensembles and the individual models within the ensembles. 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 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.

Code

Code to reproduce the numerical results of this manuscript is available in the following public GitHub repository:

https://github.com/anonymousTechno/RMSS.git

Conflict of Interests

The authors declare no potential conflict of interests.

Acknowledgement

Most of the numerical results were generated using GCF's computational infrastructure funded by the Canada Foundation for Innovation (CFI). Authors were supported by NSERC Discovery grant and the Data Science Institute of the University of British Columbia.

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.

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 l 2 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 (2023a). *RMSS: Robust Multi-Model Subset Selection*. R package version 1.1.1.

Christidis, A. and G. Cohen-Freue (2023b). *robStepSplitReg: Robust Stepwise Split Regularized Regression*. R package version 1.1.0.

Christidis, A.-A., S. V. Aelst, and R. Zamar (2023). Multi-model subset selection.

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. *Ann. Statist. 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.

Hastie, T., R. Tibshirani, and M. Wainwright (2019). *Statistical learning with sparsity: the lasso and generalizations.* Chapman and Hall/CRC.

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 2.2.0.

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.

Mallows, C. L. (1973). Some comments on cp. *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.

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.

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.