Machine Learning (2023) 112:4995–5055
https://doi.org/10.1007/s10994-023-06384-z

Trimming stability selection increases variable selection robustness

Tino Werner[1]

Received: 22 August 2022 / Revised: 1 August 2023 / Accepted: 8 August 2023 /
Published online: 4 October 2023

Abstract
Contamination can severely distort an estimator unless the estimation procedure is suitably robust. This is a well-known issue and has been addressed in Robust Statistics, however, the relation of contamination and distorted variable selection has been rarely considered in the literature. As for variable selection, many methods for sparse model selection have been proposed, including the Stability Selection which is a meta-algorithm based on some variable selection algorithm in order to immunize against particular data configurations. We introduce the variable selection breakdown point that quantifies the number of cases resp. cells that have to be contaminated in order to let no relevant variable be detected. We show that particular outlier configurations can completely mislead model selection. We combine the variable selection breakdown point with resampling, resulting in the Stability Selection breakdown point that quantifies the robustness of Stability Selection. We propose a trimmed Stability Selection which only aggregates the models with the best performance so that, heuristically, models computed on heavily contaminated resamples should be trimmed away. An extensive simulation study with non-robust regression and classification algorithms as well as with two robust regression algorithms reveals both the potential of our approach to boost the model selection robustness as well as the fragility of variable selection using non-robust algorithms, even for an extremely small cell-wise contamination rate.

Keywords Model selection · Robustness · Sparsity · Stability selection · Breakdown point

Mathematics Subject Classification 62F40 · 62J05 · 62F35 · 62H30

Editor: Derek Greene.

✉ Tino Werner
tino.werner1@uni-oldenburg.de

[1] Institute for Mathematics, Carl von Ossietzky University Oldenburg, Carl-von-Ossietzky-Straße 9-11, 26129 Oldenburg, Lower Saxony, Germany

1 Introduction

Feature selection in machine learning became popular once Tibshirani introduced the Lasso (Tibshirani, 1994). This opened the path for a plethora of feature selection methods in regression [e.g., (Bühlmann & Yu, 2003; Efron et al., 2004; Zou, 2006; Yuan & Lin, 2006; Simon et al., 2013)], classification (Park & Hastie, 2007; Meinshausen, 2007; Van de Geer, 2008), clustering [e.g., (Witten & Tibshirani, 2010; Alelyani et al., 2013) and references therein], ranking (Tian et al., 2011; Lai et al., 2013; Laporte et al., 2014) and sparse covariance or precision matrix estimation (Banerjee et al., 2008; Friedman et al., 2008; Van de Geer, 2016).

It has been observed that single regularized models often tend to overfit, which was the starting point for a more sophisticated concept that combines ensemble models with feature selection, namely Stability Selection (Meinshausen & Bühlmann, 2010). Roughly speaking, in Stability Selection one considers subsamples of the training data, performs feature selection on each subsample and aggregates the models to get a final stable model at the end. Stability Selection can be interpreted as meta-algorithm for which algorithms like the usual Lasso but also Lasso variants and Boosting (Hofner et al., 2015) can enter as base algorithm for feature selection.

All non-ensemble non-robust feature selection algorithms, including standalone Boosting models, usually get distorted once the training data are contaminated, for example, by wrong measurements or by manipulations done by an attacker who is able to intercept the data and, maybe targetedly, alter entries. Safeguarding against such contamination is done by methods of Robust Statistics [e.g., (Maronna et al., 2019; Huber & Ronchetti, 2009; Hampel et al., 2011; Rieder, 1994)]. Although outlier detection strategies [e.g., (Rocke & Woodruff, 1996; Shieh & Hung, 2009; Filzmoser et al., 2008; Rousseeuw & Hubert, 2011; Rousseeuw & Van Den Bossche, 2018)] are important for data cleaning so that a classical algorithm may be applied, robust learning algorithms can directly cope with contaminated data. The main idea [see (Huber & Ronchetti, 2009; Hampel et al., 2011)] is to bound the effects of contamination by replacing unbounded loss functions like the squared loss by a loss function with a bounded derivative or, even better, by a redescender whose derivative tends to zero for large arguments, or to assign less weight to suspicious instances. Robust techniques have successfully entered sparse feature selection like robust Lasso algorithms (Rosset & Zhu, 2007; Chen et al., 2010b, a; Chang et al., 2018), robust L_2-Boosting (Lutz et al., 2008) and Sparse Least Trimmed Squares (SLTS) (Alfons et al., 2013), among several other variants.

In practice, especially for high-dimensional data, cell-wise outliers (Alqallaf et al., 2009) are more realistic than case-wise outliers, so single cells are allowed to be contaminated, independently from whether the other cells in the respective instance are contaminated. In contrast to case-wise contamination, which can be interpreted as measuring all variables for an instance with one single sensor which may be corrupted with a certain probability, cell-wise contamination can be identified with measuring each variable with an individual sensor, and that the sensors are corrupted independently [see (Filzmoser et al., 2020)]. For high-dimensional data, this can cause each instance to be contaminated while the fraction of outliers, measured by the relative part of contaminated cells and not of contaminated rows, can still remain very low. There are already some algorithms that can cope with cell-wise outliers like the robust clustering algorithm that trims outlying cells (García-Escudero et al., 2021), cell-wise robust location and scatter matrix estimation algorithms

(Agostinelli et al., 2015; Leung et al., 2017) and several regression approaches (Bottmer et al., 2022; Leung et al., 2016; Filzmoser et al., 2020).

A common robustness measure is given by the breakdown point. In contrast to the prominent influence curve which quantifies the local robustness of an estimator, i.e., only allowing for an infinitesimal fraction of the data to be contaminated, the breakdown point (BDP), introduced in Hampel (1971, Sect. 6) in a functional version and in Donoho and Huber (1983) in a finite-sample version, studies the global robustness of an estimator. The finite-sample BDP from Donoho and Huber (1983) quantifies the minimum fraction of instances in a data set that guarantees the estimator to "break down" when being allowed to be contaminated arbitrarily while the functional BDP essentially quantifies the minimum Prokhorov distance of the ideal and contaminated distribution that leads to such a breakdown. There has already been a lot of work on BDPs, see for example Rousseeuw (1984, 1985), Davies (1993) and Hubert (1997), Genton (1998), Becker and Gather (1999), Gather and Hilker (1997) or Hubert et al. (2008) which cover location, scale, regression, spatial and multivariate estimators. Recently, BDP concepts for classification (Zhao et al., 2018), multiclass-classification (Qian et al., 2019) and ranking (Werner, 2022b) have been proposed. Another type of breakdown point which relates the quotient of the number of variables and observations to the sparsity of the true model was studied in Donoho and Stodden (2006), illuminating that in high-dimensional settings, classical model selection procedures like Lasso can only find a reliable model provided that the true model is sufficiently sparse.

Despite the recent successes of robust model selection methods and path-breaking theoretical results concerning model selection consistency [see, e.g., (Bühlmann & Van De Geer, 2011)], the question how contamination affects variable selection is seldomly addressed, leaving the connection of the paradigms of sparse model selection (select a small fraction of columns), stability (select variables that are appropriate for the majority of the data, i.e., a majority of the rows) and robustness (focus on a certain "clean" majority of the rows) still opaque. Although the frequency of contamination in real data is high and although the issue that non-aggregated feature selection models are usually unstable and tend to overfit is well-known, combining both stability and robustness seems to have rarely been considered in the literature so far. A robust Stability Selection based on cleaned data has been introduced in Uraibi et al. (2015). The author of Uraibi (2019) applies a weighted LAD-Lasso (Arslan, 2012) as base algorithm for the Stability Selection where the weights are computed according to a robust distance in order to downweigh leverage points. The authors of Park et al. (2019) proposed a robust Stability Selection where the term "robust" however refers to an immunization against a specific regularization parameter of the underlying Lasso models, therefore considers a different goal than we do. Notable work on aggregating robust estimators has been done in Salibián-Barrera and Zamar (2002); Salibián-Barrera et al. (2008) who propose a linear update rule for robust MM-estimators in order to avoid computing it on each of the drawn Bootstrap samples, see also Salibián-Barrera (2006) for fixed-design regressor matrices and Salibián-Barrera et al. (2006) for robust PCA estimators. While these techniques do not consider variable selection, Salibián-Barrera and Van Aelst (2008) extend their method to variable selection where a backward selection strategy according to a minimization of the expected prediction error is applied.

We aim at making a tiny step towards the connection of robustness, stability and sparsity by introducing the variable selection breakdown point, which describes the number of (case-wise or cell-wise) outliers that can make variable selection completely unreliable, and the Stability Selection breakdown point, which corresponds to a sufficiently

high probability that a stable model becomes completely unreliable. We study the relative robustness improvement that a Stability Selection grants, compared to a single model selection algorithm. It turns out that a rank-based Stability Selection where a fixed number of best variables enters the stable model is generally more robust than a threshold-based Stability Selection where all variables whose aggregated selection frequency exceeds the threshold enters the stable model. We propose a trimmed Stability Selection (*TrimStabSel*) and investigate its performance on a large variety of simulated data in comparison with a non-robust Stability Selection, where L_2-Boosting, LogitBoost and SLTS are used model selection algorithms. The numerical results show that even an extremely small cell-wise contamination rate can already have a severe impact on variable selection. Our TrimStabSel can in particular be recommended in settings where the contamination rate is expected to be low but non-zero and in presence of rather high noise levels. Moreover, it can, depending on the particular model selection algorithm, even enhance the performance of a Stability Selection with a robust model selection algorithm.

Our contribution is threefold: (i) we propose BDP definitions for (stable) variable selection; (ii) we propose (oracle) outlier schemes that can completely distort model selection with usually very few cell-wise outliers; (iii) we lift the popular concept of trimming from single instances to whole models that allows for contamination rates exceeding 50% while maintaining the 50%-bound for the standard BDPs of the underlying model selection algorithms.

This paper is organized as follows. Section 2 compiles relevant notions from Robust Statistics which are contamination models and the breakdown point. We also recapitulate the concept of Stability Selection. Section 3 is devoted to the definition of our variable selection BDP and to a discussion on resampling BDPs which are required for our Stability Selection. Our BDP concept for Stability Selection is introduced in Sect. 4. Section 5 presents the Trimmed Stability Selection. Section 6 provides a detailed simulation study that compares the performances of the standard Stability Selection and the Trimmed Stability Selection on simulated data. Most of the figures that depict the simulation results have been moved to the appendix for better readability.

2 Preliminaries

Let $D \in \mathbb{R}^{n\times(p+k)}$ be the data matrix consisting of a regressor matrix $X \in \mathbb{R}^{n\times p}$ and a response matrix $Y \in \mathbb{R}^{n\times k}$. If univariate responses are considered, $Y \in \mathbb{R}^n$ is a response column. We denote the i-th row of X by X_i and the j-th column of X by $X_{\cdot,j}$. The i-th row of Y is denoted by Y_i and for $k = 1$, Y_i denotes the i-th component of Y. Let $n_{\text{sub}} \leq n$ always be the number of instances in subsamples resp. Bootstrap samples.

We always assume that we have a parametric regression or classification model $f_\beta : \mathcal{X} \to \mathcal{Y}$ that maps an $X_i \in \mathcal{X} \subset \mathbb{R}^p$ onto $f_\beta(Y_i) \in \mathcal{Y} \subset \mathbb{R}^k$, with a parameter $\beta \in \mathbb{R}^p$ that is to be inferred.

2.1 Contamination models

We begin with the definition of convex contamination balls (see Rieder (1994, Sect. 4.2) for a general definition of contamination balls).

Definition 1 Let $(\Omega, \mathcal{A})$ be a measurable space. Let $\mathcal{P} := \{P_\theta \mid \theta \in \Theta\}$ be a parametric model where each P_θ is a distribution on $(\Omega, \mathcal{A})$ and where $\Theta \subset \mathbb{R}^p$ is some parameter space. Let P_{θ_0} be the ideal distribution ["model distribution", Rieder (1994, Sect. 4.2)]. A convex contamination model is the family of contamination balls

$$U_c(\theta_0, r) = \{(1-r)_+ P_{\theta_0} + \min(1, r)Q \mid Q \in \mathcal{M}_1(\mathcal{A})\}$$

where $r \in [0, \infty[$ is the contamination radius and for the set $\mathcal{M}_1(\mathcal{A})$ of probability distributions on $\mathcal{A}$.

Definition 1 only considers case-wise (row-wise/instance-wise) outliers, i.e., either a whole row in the regressor matrix or in the response is contaminated or not. A more realistic scenario where the entries (cells) of the regressor matrix are allowed to be perturbed independently (cell-wise outliers) has been introduced in Alqallaf et al. (2009). See also Agostinelli et al. (2015) for the notation.

Definition 2 Let $W \sim P_{\theta_0}$ where P_{θ_0} is a distribution on some measurable space $(\Omega, \mathcal{A})$ for $\Omega \subset \mathbb{R}^p$. Let $U_1, ..., U_p \sim Bin(1, r)$ i.i.d. for $r \in [0, 1]$. Then the *cell-wise convex contamination model* considers all sets

$$U^{cell}(\theta_0, r) := \{Q \mid Q = \mathcal{L}(UW + (I_p - U)\tilde{W})\}$$

where $\tilde{W} \sim \tilde{Q}$ for any distribution $\tilde{Q}$ on $(\Omega, \mathcal{A})$ and for the unit matrix I_p of dimension $p \times p$ and the matrix U with diagonal entries U_j. $\mathcal{L}$ denotes the distribution (law) of the respective random variable.

The authors in Alqallaf et al. (2009) pointed out that if all U_j are perfectly dependent, one either gets the original row or a fully contaminated row which is just the classical convex contamination model in Definition 1. In supervised learning, the X_i or the (X_i, Y_i) take the role of W in Definition 2.

Remark 1 Note that for response matrices, one can similarly construct cell-wise outliers that operate on the response matrix only, and clearly combine it with cell-wise outliers on the regressor matrix to get cell-wise outliers on both the regressor and the response matrix.

In the cell-wise contamination model, the probability that at least one case is contaminated grows with the dimension p. Note that a single contaminated cell already makes an observation an (case-wise) outlier [e.g., (Öllerer & Croux, 2015; Croux & Öllerer, 2016)].

2.1.1 The breakdown point concept

The goal of Robust Statistics is to provide robust estimators, i.e., estimators that tolerate a certain amount of contaminated data without being significantly distorted. As for the term "robustness", in this work, we use the global robustness concept that allows for a large fraction of the data points being contaminated arbitrarily [in contrast to local robustness, corresponding to influence curves (Hampel, 1974), where an infinitesimal amount of contamination is considered]. The minimum fraction of (case-wise) outliers that can lead to a breakdown of the

estimator is called the (case-wise) breakdown point (BDP) of this particular estimator. The finite-sample BDP of Donoho and Huber (1983) is defined as follows.

Definition 3 Let Z_n be a sample consisting of instances $(X_1, Y_1), ..., (X_n, Y_n)$. The *finite-sample breakdown point* of the estimator $\hat{\beta}$ is defined by

$$\varepsilon^*(\hat{\beta}, Z_n) = \min \left\{ \frac{m}{n} \;\middle|\; \sup_{Z_n^m} (||\hat{\beta}(Z_n^m)||) = \infty \right\} \tag{1}$$

where Z_n^m denotes any sample with $(n - m)$ instances in common with the original sample Z_n, so one can arbitrarily contaminate m instances of Z_n, and where $\hat{\beta}(Z_n)$ is the estimated coefficient on Z_n^m.

2.2 Stability selection

The Stability Selection is an ensemble model selection technique that has been introduced in Meinshausen and Bühlmann (2010), mainly with the goal to reduce the number of false positives (non-relevant variables that are selected by the algorithm) and also motivated by the fact that the true predictor set $S_0 \subset \{1, ..., p\}$ is often not derivable by applying a single model selection procedure [cf. Meinshausen and Bühlmann (2010, p. 423)]. In short, one draws B subsamples from the data of usually around $n/2$ instances and performs a model selection algorithm on each subsample which leads to a set $\hat{S}^{(b)} \subset \{1, ..., p\}$ of selected variables for each $b = 1, ..., B$. The next step is to aggregate the selection frequencies of all variables, i.e., the binary indicators whether a particular variable has been selected in a specific predictor set. More precisely, one computes $\hat{\pi}_j := \frac{1}{B} \sum_{b=1}^{B} I(j \in \hat{S}^{(b)})$ for all j.

In the original Stability Selection from Meinshausen and Bühlmann Meinshausen and Bühlmann (2010), one defines a threshold π_{thr} based on an inequality derived in Meinshausen and Bühlmann (2010, Theorem 1) so that the stable model then consists of all variables j for which $\hat{\pi}_j \geq \pi_{thr}$. There are some variants of this Stability Selection, most notably the one in Hofner et al. (2015) that makes it applicable for Boosting while the original one from Meinshausen and Bühlmann (2010) is tailored to algorithms that invoke a regularization term like the Lasso (Tibshirani, 1994) or the Graphical Lasso (Banerjee et al., 2008; Friedman et al., 2008). An excellent implementation of the Stability Selection can be found in the R-packages `mboost` (Hothorn et al., 2017; Hofner et al., 2014; Hothorn et al., 2010; Bühlmann & Hothorn, 2007; Hofner et al., 2015; Hothorn & Bühlmann, 2006) and `stabs` (Hofner & Hothorn, 2017; Hofner et al., 2015; Thomas et al., 2018).

As for the selection of the stable model according to the aggregated selection frequencies, another paradigm that defines a number q of variables that have to enter the stable model so that the q variables with the highest selection frequencies are chosen has been suggested in the literature [e.g., (Zhou et al., 2013; Werner, 2022a)]. The reason is that the threshold-based approach is less intuitive for the user due to the number of stable variables not being predictable in the first place.

3 Breakdown of variable selection

In this section, we first define a BDP for variable selection. Based on this definition, we discuss why this BDP may be very small and outline the path from the robustness of single algorithms concerning variable selection to ensembles of such algorithms.

3.1 Variable selection breakdown point

Donoho and Stodden (2006) already provided a very insightful work where the notion of a "breakdown of model selection" has been introduced. They computed phase diagrams that show under which configurations of the dimensionality of the data and the sparsity level of the true underlying model a successful model selection is possible. More precisely, they derive that the underlying model has to be sufficiently sparse, expressed in the fraction of the true dimensionality q of the model (which is given by the number of non-zero entries of the true parameter β) and the number n of observations, the question whether model selection is possible depends on the fraction n/p for p being the number of predictors.

Our idea also considers to compute a breakdown for model selection, but we restrict ourselves to the standard setting of Robust Statistics where we want to examine how many (case- or cell-wise) outliers can be tolerated for model selection. In order not to confuse our BDP concept with the one in Donoho and Stodden (2006), we call our concept the variable selection breakdown point (VSBDP).

Definition 4 Let D be a data set with n instances and predictor dimension p. Let $k \in \mathbb{N}$ be the dimension of the responses.

(a) The *case-wise variable selection breakdown point (case-VSBDP)* is given by

$$\frac{m^*}{n}, \quad m^* = \min\{m \mid \hat{\beta}_j(Z_n^m) = 0 \; \forall j : \beta_j \neq 0\} \tag{2}$$

where Z_n^m again denotes any sample that has $(n - m)$ instances in common with Z_n.

(b) The *cell-wise variable selection breakdown point (cell-VSBDP)* is given by

$$\frac{\tilde{m}^*}{(p+k)n}, \quad \tilde{m}^* = \min\{m \mid \hat{\beta}_j(\tilde{Z}_{cell}^m) = 0 \; \forall j : \beta_j \neq 0\} \tag{3}$$

where $\tilde{Z}_{cell}^m$ denotes the data set where m cells can be modified arbitrarily and where all other cell values remain as in the original data.

In other words, the VSBDP quantifies the relative number of rows resp. cells that have to be contaminated in order to guarantee that none of the relevant variables are selected. The fraction of outlying cells as a breakdown measure has for example already been considered in Velasco et al. (2020). Note that finite-sample breakdown points usually do not assume knowledge of the true model. We assumed this knowledge however in our definition because otherwise one would have to consider some empirically derived model and tailor the BDP definition to that. In the context of variable selection, such a definition could imply issues, for example, if the computed model is empty. Therefore, we restrict ourselves to the proposed definition here. Let us now formulate a very simple but important result.

Theorem 1 *Let p be the total number of variables and let $q \leq p$ be the true dimension of the underlying model and let again k be the response dimension and n be the number of instances in the data set.*

(a) *Then the cell-VSBDP is at most* $\frac{\min(q,k)}{p+k}$.
(b) *Let $p = p(n)$ so that it grows when n grows. If q or k stays constant, the asymptotic cell-VSBDP is zero.*
(c) *Let $p = p(n)$ and $q = q(n)$ so that both quantities grow when n grows. Let contamination only be allowed on the predictor matrix. Then the asymptotic cell-VSBDP is given by* $\lim_{n\to\infty}\left(\frac{q(n)}{p(n)}\right)$.

Proof

(a) If $q < k$, just replace the entries X_{ij} for all i and for all j corresponding to the relevant variables by zeroes, so that one has to modify qn cells of the data set, more precisely, only of the predictor matrix. Then the originally relevant columns remain without any predictive power and therefore will not be selected. If $q \geq k$, replace all entries of the response matrix with zeroes which would result in an empty model since no predictor column remains correlated with the response, requiring kn outlying cells.

(b)+(c) Directly follows from (a). □

This is a universal result, regardless of the data structure or the applied algorithms. We want to point out that the classical understanding of robustness would only consider the estimated coefficients themselves and call an estimator robust if the coefficients stay bounded. However, if the non-zero coefficients correspond to non-relevant variables, the learning procedure results in a robust fit on noisy variables which will definitely have poor prediction quality on out-of-sample data. Our analysis is based on an interplay between model selection and coefficient estimation so that the ultimate goal is to achieve both subgoals in order to get a reliable model. Let us therefore pose the following statement: *From the perspective of retrieving the correct model, all robust regression and classification models are doomed to have a breakdown point less than q/p.*

When having simulated data and a random cell-wise outlier scheme, it is extremely unlikely that such column-wise outliers that make the true model irretrievable would appear. However, from a practical perspective, one can for example think of an attacker that is aware of (most of) the relevant variables, maybe due to intercepting and analyzing the data first.

3.2 Resampling and robustness

We recapitulate the resampling breakdown point introduced in Berrendero (2007) in a slightly modified version.

Definition 5 Let n be the number of observations in a data set and let B be the number of Bootstrap resp. subsamples with $n_{\text{sub}} \leq n$ observations each. Let $c \in [0, 1]$ be the BDP of each estimator applied on the individual Bootstrap samples resp. subsamples. For a tolerance level $\alpha \in [0, 1]$, the α *resampling BDP for Bootstrap* is given by

$$(\varepsilon^{Boot}(c, n_{\text{sub}}, B, \alpha))^* := \inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid 1 - P(Bin(n_{\text{sub}}, \hat{\varepsilon}) < \lceil cn_{\text{sub}} \rceil)^B > \alpha\} \tag{4}$$

where $\hat{\varepsilon} = m/n$ indicates the empirical rate of outlying rows in the data set, the *α resampling BDP for Subsampling* is given by

$$(\varepsilon^{Subs}(c, n_{\text{sub}}, B, \alpha))^* \inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid 1 - P(Hyp(n, n-m, n_{\text{sub}}) > \lfloor (1-c)n_{\text{sub}} \rfloor)^B > \alpha\} \tag{5}$$

where $Hyp(n, n-m, n_{\text{sub}})$ is the hypergeometrical distribution describing the number of clean instances (successes) for $(n-m) = (1-\hat{\varepsilon})n$ clean instances.

This resampling BDP defines the maximum fraction of contaminated instances in the data so that the probability that a mean aggregation breaks down exceeds the tolerance level [cf. also Camponovo et al. (2012), Filzmoser et al. (2020, Sect. 3.5)]. This definition is important since it lifts the worst-case BDP concept to a probabilistic concept that respects that the worst case is often very unlikely and that solely reporting it would be too pessimistic. Robust aggregation procedures such as Bragging [median aggregation (Berrendero, 2007; Bühlmann, 2012)] and trimmed bagging (Croux et al., 2007) require up to $\lfloor (B+1)/2 \rfloor$ resamples being sufficiently contaminated to break down.

The idea from the rejoinder of Davies and Gather (2005) to map the boundary values of a bounded image set of an estimator to infinite values is important when assessing the effects of resampling on the robustness. A classical example is a correlation estimator which takes only values in the compact interval $[-1, 1]$. It is claimed in Grandvalet (2000) that Bagging never improves the BDP. This is not true if the value set is bounded. We first formally spell out why Bagging is usually not robust, although this fact has already been observed in the literature.

Proposition 1 *Let B be the number of resamples and let S_n be some estimator, mapping onto an unbounded domain, for a data set with n observations. If the (classical) BDP of the estimator is c, so it is for the bagged estimator.*

Proof Since the BDP of the estimator is c, manipulating a relative fraction of c instances in one single resample b suffices to let the estimator $S_n^{(b)}$ on this resample break down, i.e., the norm of the estimated value is fully controlled by the outliers. Then, as the bagged estimator being the empirical mean of all $S_n^{(b)}$, $b = 1, .., B$, its norm is also fully controlled. □

The proof is rather unusual since it assumes that one can targetedly contaminate a selected resample. Usually, the attacker should only have access to the whole training data set. Then, a probabilistic statement in the spirit of the resampling BDP becomes more appropriate.

Proposition 2 *Let B be the number of resamples and let S_n be some estimator, mapping onto an unbounded domain, for a data set with n observations. If the (classical) BDP of the estimator is c, the resampling BDP of the bagged estimator equals $(\varepsilon^{\perp}(c, n_{\text{sub}}, B, \alpha))^*$ for $\perp$= Boot for Bootstrap resp. for $\perp$= Subs for subsampling.*

Proof Evidently, if at least one resample is contaminated to an extent such that the estimator breaks down, due to the unbounded domain and the mean aggregation, the bagged estimator breaks down. By definition of the resampling BDP, this happens with a probability of more than α if the fraction of contaminated rows is $(\varepsilon^{\perp}(c, n_{\text{sub}}, B, \alpha))^*$. □

Remark 2

(a) These results can be trivially extended to the case of cell-wise contamination.
(b) Bagging can nevertheless completely de-robustify the estimator [e.g., (Salibián-Barrera et al., 2008)] if B and the number of outlying instances in the data is high.

As for bounded domains however, the situation is different.

Proposition 3

(a) *Let B be the number of resamples and let S_n be some estimator, mapping onto a bounded domain, for a data set with n observations from which no one takes any of the boundary values. If the (classical) BDP of the estimator is c, the resampling BDP of the bagged estimator is*

$$\inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid P(Bin(B, P(Bin(n_{\text{sub}}, \varepsilon) \geq \lceil cn_{\text{sub}} \rceil)) = B) > \alpha\} \tag{6}$$

for Bootstrapping resp.

$$\inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid P(Bin(B, P(Hyp(n, n - \varepsilon n, n_{\text{sub}}) \leq \lfloor (1 - c)n_{\text{sub}} \rfloor)) = B) > \alpha\} \tag{7}$$

for subsampling.

(b) *For Bragging, the resampling BDP becomes*

$$\inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid P(Bin(B, P(Bin(n_{\text{sub}}, \varepsilon) \geq \lceil cn_{\text{sub}} \rceil)) \geq \lfloor (B + 1)/2 \rfloor) > \alpha\} \tag{8}$$

for Bootstrapping resp.

$$\begin{aligned} &\inf\{\varepsilon \in \{0, 1/n, ..., 1\} \mid \\ &\quad P(Bin(B, P(Hyp(n, n - \varepsilon n, n_{\text{sub}}) \leq \lfloor (1 - c)n_{\text{sub}} \rfloor)) \geq \lfloor (B + 1)/2 \rfloor) > \alpha\} \end{aligned} \tag{9}$$

for subsampling.

Proof

(a) If at least one estimator does not break down, it takes a value in the interior of the domain by assumption. Hence, the aggregated estimated value will also lie in the interior of the domain, so there is no breakdown. A breakdown occurs if and only if all estimators break down to the same boundary value (which, by the worst-case perspective of the BDP concept, can be assumed to be possible).
(b) When taking the median of the estimated values, it suffices that at least the half estimators have broken down to the same boundary value. □

This is an unusual result since the median aggregation (and any other trimmed aggregation) is less robust than the mean aggregation. This artifact, resulting from a wrong notion of robustness for estimation, shows that a breakdown has to be defined very carefully if

the domain is bounded, providing another argument why the concept from the rejoinder of Davies and Gather (2005) to map the boundary values to infinite values is necessary.

4 Stability selection and robustness

Important work on stability of feature selection resp. feature ranking has been done in Nogueira and Brown (2016); Nogueira et al. (2017b, 2017a). Nogueira et al. (2017a), it is pointed out that stable feature selection is either represented by a hard subset selection of the candidate variables or by a ranking of the variable or individual weights which, given some threshold for the ranks resp. the weights, eventually leads to a subset of variables. The cited works propose similarity metrics in order to quantify the stability of feature selection resp. feature ranking. The authors of Nogueira et al. (2017a) consider the stability of feature selection as a robustness measure for the feature preferences. Note that this robustness notion differs from the definition of robustness in the sense of Robust Statistics.

4.1 The stability selection BDP

Having the VSBDP and the resampling BDP defined, we are ready to define a BDP for Stability Selection itself.

Definition 6 Let n be the number of instances in a data set and let n_{sub} be the number of instances in each resample. Let B be the number of resamples for the Stability Selection and let R be the resampling distribution. Let $S^{stab}(\perp, n_{\text{sub}}, B, Z_n)$ denote the stable model derived from B resamples according to $\perp = Boot$ or $\perp = Subs$ with n_{sub} instances from the data set Z_n.

(a) Then the *case-wise Stability Selection BDP (case-Stab-BDP)* for tolerance level α is given by

$$\varepsilon^*_{Stab}(\perp, c, n_{\text{sub}}, B, \alpha) := \min\left\{ \frac{m}{n} \;\middle|\; P_R(\forall j \in S_0 : j \notin S^{stab}(\perp, n_{\text{sub}}, B, Z_n^m)) \geq \alpha \right\} \quad (10)$$

where c represents the case-BDP of the underlying model selection algorithm and where $S_0 \subset \{1, ..., p\}$ again denotes the true predictor set.

(b) Similarly, the *cell-wise Stability Selection BDP (cell-Stab-BDP)* for tolerance level α is given by

$$\tilde{\varepsilon}^*_{Stab}(\perp, \tilde{c}, n_{\text{sub}}, B, \alpha) := \min\left\{ \frac{\tilde{m}}{n(p+k)} \;\middle|\; P_R(\forall j \in S_0 : j \notin S^{stab}(\perp, n_{\text{sub}}, B, \tilde{Z}^m_{cell})) \geq \alpha \right\} \quad (11)$$

with the cell-BDP $\tilde{c}$ of the underlying model selection algorithm.

Intuitively, the Stab-BDP denotes the minimum fraction of outliers required so that the probability that the reported stable model does not contain any relevant variable is sufficiently large. As Stability Selection does not aggregate coefficients but just indicator functions, the influence of a model computed on a single resample is bounded by $1/B$ in the

sense that the aggregated selection frequencies computed on the original data can be at most distorted by $1/B$ (for certain j) if one single resample is (sufficiently) contaminated.

It remains to investigate how the Stab-BDP can be computed in practice. Although the BDP is known for many regression and classification estimators, the computation of the Stab-BDP requires the impact of contamination on variable selection, which is unknown. Therefore, we make assumptions concerning this impact in the next subsection.

4.2 Assumptions on the impact of contamination on variable selection

Relevant variables may not only be suppressed due to contamination as in Theorem 1 but non-relevant variables may also be promoted [cf. Li et al. (2020, 2021)]. To the best of our knowledge, a concise statement on the impact of outliers to model selection itself such as how many relevant variables can be suppressed or how many non-relevant variables can be promoted has not yet been proven in the literature. Therefore, we propose an optimistic and a pessimistic scenario concerning this impact.

Case-wise scenarios: in this scenario, we assume that for a variable selection method with instance-BDP c, a number of $\lceil cn \rceil$ outlying rows in the regressor matrix, the response matrix or the regressor matrix reduced to the relevant columns

- is able to promote any subset of non-relevant variables resp. suppress any subset of relevant variables which means that we assume that this outlier fraction can indeed, at least theoretically, cause all relevant variables to be ignored (*pessimistic case-wise scenario*);
- is able to targetedly suppress all relevant but promote only one single variable (*optimistic case-wise scenario*).

Cell-wise scenarios: we assume that for a cell-BDP of $\tilde{c}$, a fraction of at least $\tilde{c}$ of outlying cells in the regressor matrix, the response matrix or the regressor matrix reduced to the relevant columns

- can promote $\min(p - s_0, \tilde{c})$ non-relevant variables where $s_0 = |S_0|$ resp. suppress any subset of relevant variables (*pessimistic cell-wise scenario*). This scenario is indeed very pessimistic since even in Li et al. (2020, 2021), it seems that one at least has to manipulate the regressor matrix and the response vector which would at least require two outlying cells for an estimator with $\tilde{c} = 0$;
- is able to targetedly suppress all relevant variables but to promote only one single variable (*optimistic cell-wise scenario*).

These scenarios should represent the extreme cases of which impact of contamination on variable selection one may expect. Milder scenarios than the optimistic one seem inappropriate in regard of Theorem 1 where one can easily suppress all relevant variables.

4.3 Robustness of threshold- and rank-based stability selection

We now quantify the robustness of threshold- and rank-based Stability Selection. We assume that there is a fixed selection of instances resp. cells in the data matrix which are contaminated, so we quantify the probability that the Stability Selection breaks down. Note that the computations are only done for the scenarios introduced in the previous subsection, hence representing the range of breakdown probabilities. The true Stab-BDP for a

given data configuration and model selection algorithm may only be approximated empirically, which we do in Sect. 6.

Theorem 2 *Let π be the threshold for the Stability Selection based on B resamples of size n_{sub} from a data set with n instances. Let q be the pre-scribed number of stable variables for the Stability Selection based on B resamples of size n_{sub} from a data set with n instances. Let $\hat{\pi}_j^+$ for $j \in S_0$ resp. $\hat{\pi}_k^-$ for $k \in \{1, ..., p\} \backslash S_0$ be the aggregated selection frequencies on the original data where we assume that $\sum_k I(\hat{\pi}_k^- \geq \hat{\pi}_j^+) \leq q - 1 \; \exists j \in S_0$ for the rank-based Stability Selection resp. $\hat{\pi}_j^+ \geq \pi \; \exists j \in S_0$ for the threshold-based Stability Selection, so there is no immediate breakdown. Let s be the number of relevant variables in the top-q variables. Then, w.l.o.g., let $\hat{\pi}_1^+, ..., \hat{\pi}_s^+$ be the corresponding aggregated selection frequencies of these variables on the original data, similarly, let $\hat{\pi}_1^-, ..., \hat{\pi}_{q-s}^-$ be the aggregated selection frequencies of the $(q - s)$ non-relevant variables (out of $(p - s)$ non-relevant variables in total) among the top−q variables. Let $\hat{\pi}_{q-s+1}^-, ..., \hat{\pi}_q^-$ be the aggregated selection frequencies of the next best s non-relevant variables. Then,*

(a) *In the pessimistic scenario, the rank-based Stability Selection is more robust than the threshold-based Stability Selection in terms of the Stab-BDP if and only if $\pi > 0.5(\max_{j=1,...,s}(\hat{\pi}_j^+) + \min_{k=q-s+1,...,q}(\hat{\pi}_k^-))$;*

(b) *In the optimistic scenario, the rank-based Stability Selection is always more robust than the threshold-based Stability Selection in terms of the Stab-BDP if $\pi > 0.5(\max_{j=1,...,s}(\hat{\pi}_j^+) + \min_{k=q-s+1,...,q}(\hat{\pi}_k^-))$, and the threshold-based Stability Selection is always more robust if $\pi < \min_{k=q-s+1,...,q}(\hat{\pi}_k^-)$.*

Proof See App. A for the proof of (a) for cell-wise contamination and (b).

(a) We distinguish between case-wise and cell-wise outlier configurations. (i) Here, we consider case-wise contamination. Let always a fixed selection of m instances be contaminated. Let c be the case-BDP of the applied model selection algorithm. First, note that for the threshold-based Stability Selection, the selection frequencies of the non-relevant variables are not important, so one does not have to distinguish between the pessimistic and the optimistic scenario. A breakdown is achieved once each relevant variable no longer appears in the stable model, i.e., if all aggregated selection frequencies are lower than the threshold π. Regarding the variable $j^* = \text{argmax}_j(\hat{\pi}_j^+)$, there are more than $\lceil B(\hat{\pi}_{j^*}^+ - \pi) \rceil$ sufficiently contaminated resamples (at least $\lceil cn_{\text{sub}} \rceil$ instances in the resample are contaminated) required since each such resample can decrease the aggregated selection frequency by at most $1/B$. Putting everything together, the Stability Selection breaks down with a probability of

$$P\left(Bin(B, P(Hyp(n, n-m, n_{\text{sub}}) \leq \lfloor (1-c)n_{\text{sub}} \rfloor)) > \lceil B(\max_{j=1,...,s}(\hat{\pi}_j^+) - \pi) \rceil \right) \tag{12}$$

if the resamples are drawn by subsampling and with a probability of

$$P\left(Bin(B, P(Bin(n_{\text{sub}}, m/n) \geq \lceil cn_{\text{sub}} \rceil)) > \lceil B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \pi) \rceil\right) \tag{13}$$

if the resamples are drawn by Bootstrapping. For the rank-based Stability Selection, a breakdown is achieved once each relevant instance has an aggregated selection frequency smaller than the aggregated selection frequencies of q non-relevant variables. Therefore, it suffices to have more than $\lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil$ contaminated resamples since in each one, both the non-relevant variables corresponding to $\hat{\pi}_k^-$, $k = 1, ..., q$, can be promoted and at the same time, the relevant variables can be suppressed, so the distance between the quantities $\max_{j=1,\ldots,s}(\hat{\pi}_j^+)$ and $\min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)$ decreases by $2/B$ steps per contaminated resample. Hence, after $\lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil$ such steps, they are equal or their order relation even has switched, so having one more contaminated resample, the formerly best relevant variable definitely has a lower aggregated selection frequency than the formerly q-th best non-relevant variable. The breakdown probabilities are then

$$\begin{aligned} P(Bin(B, P(Hyp(n, n-m, n_{\text{sub}}) \leq \lfloor (1-c)n_{\text{sub}} \rfloor)) \\ > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil) \end{aligned} \tag{14}$$

if the resamples are drawn by subsampling and

$$\begin{aligned} P(Bin(B, P(Bin(n_{\text{sub}}, m/n) \geq \lceil cn_{\text{sub}} \rceil)) \\ > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil) \end{aligned} \tag{15}$$

if the resamples are drawn by Bootstrapping. □

Remark 3

(i) The cases corresponding to P_1 and P_2 resp. to $\check{P}_1$ and $\check{P}_2$ in the proof of Theorem 2 have to be considered separately since having only contamination in the relevant columns makes it impossible that the whole predictor matrix is contaminated too much provided that $s_0/p < \tilde{c}$. Similarly, if contamination only occurs in the non-relevant columns, a breakdown can still be possible due to promoting effects of the contamination.

(ii) In Theorem 2, we considered the case of univariate responses. For multivariate responses with k response columns, one has to distinguish between the seemingly unrelated regression case (Zellner, 1962) where one fits a model for each response column separately and the general case that the response columns are correlated so that the entire response matrix enters as input for a unified model. In the second case, the probability of a breakdown would be 1 if the relative part of contaminated cells in the response matrix is at least $\tilde{c}$, in the first case however, the relative part of outliers has to be larger than $\tilde{c}$ for one resp. for each response column if a breakdown of the set of the resulting k models is defined in the sense that at least one resp. each of the individual models break down.

Apart from the simple observation that robustness is increased by Stability Selection, we can extract one interesting recommendation for the Stability Selection variant. For the threshold-based Stability Selection, the robustness depends on the difference of the aggregated selection frequency of the best relevant variable and the threshold. It is however not evident

that there even exists such a variable. This problem has been studied for example in Werner (2022a) for noisy data. Due to the combination of a high noise level, a large number p of candidate variables and a rather sparse true model, one often faces the situation that no variable (including non-relevant variables) can pass the threshold, leading to an empty model, i.e., an immediate breakdown in the sense of the VSBDP. As shown in Theorem 2, it depends on how well the aggregated selection frequencies of the best relevant and the best non-relevant variables are spread. Once there is a strong relevant variable j_0 with $\hat{\pi}^+_{j_0} \approx 1$, it follows that the rank-based Stability Selection is better than the threshold-based variant if one follows the recommendation from literature to set π_{thr} at least to 0.5. On the other hand, the condition $\pi < \min_{k=q-s+1,\ldots,q}(\hat{\pi}^-_k)$ from part b) of Theorem 2 is very unlikely to become valid in practice, hence in the optimistic scenario, it should not happen that the threshold-based variant clearly beats the rank-based one in all configurations. Therefore, we recommend to prefer the rank-based Stability Selection over the threshold-based Stability Selection in regard of variable selection robustness. Note that this does not contradict Meinshausen and Bühlmann (2010, Theorem 1) as the number q can be fixed empirically so that an appropriate bound is achieved, where the aggregated selection probability of the q-th best variable replaces the universal threshold.

We learn from Theorem 2 that Stability Selection does not suffer from the numerical instabilities [cf. Salibián-Barrera et al. (2008) for this notion] of resampling that can lead to some resamples having a larger fraction of outlying instances/cells than the whole training data to that extent as standard bagged estimators do.

In contrast to a simple bagged estimator, Stability Selection allows for more than the half of the instances/cells being contaminated without violating equivariance properties [cf. Davies and Gather (2005)] of the underlying algorithm that prevent BDPs from exceeding 0.5. For example, in machine learning, especially regression, one assumes that there is an underlying model from which the clean data have been generated. Even if the relative fraction of outlying instances exceeds a half, say it is 60%, then it is nevertheless a desirable goal to infer the model which describes the correspondence structure of the responses and the predictors of the clean observations. There is no qualitative hindrance to aim at finding the underlying model due to the standard assumption that outliers *do not have structure* and just stem from some unknown, arbitrary distribution.

In order to clarify the argumentation, consider the awkward case-wise contamination situation that the outliers had structure, i.e., additionally to the underlying model $f : \mathcal{X} \to \mathcal{Y}$ that relates the responses and the predictors of the clean observations one had another model $g : \mathcal{X} \to \mathcal{Y}$ from which the outlying instances are generated. In this artificial setting, one would indeed try to infer g instead of f and treat the actual clean instances as the outliers, more precisely, as the outliers w.r.t. the model g.

The main problem, even for a Stability Selection, would be that the probability to draw resamples that are sufficiently contaminated to let the applied algorithm break down would become rather high if the fraction of outlying instances is already rather high on the original data. Therefore, an additional robustification step is necessary which will be discussed in the next section.

5 Robustifying stability selection

In each iteration of SLTS (Alfons et al., 2013), the squared loss for all instances is computed and the "clean subset" that enters the next iteration contains the $h < n$ instances with the lowest in-sample losses. Regarding Stability Selection, the instances in a resample

cannot be treated individually since one aims at aggregating column information from whole resamples. One can however individualize the resamples themselves and the corresponding selected sets of variables. In other words, we aim at lifting the trimming concept from instances in estimation problems to resamples in a model aggregation problem.

5.1 Related approaches

In contrast to (robust) Bagging, we do not combine classifiers nor other learners but predictor sets themselves. Zhang et al. (2017, 2019), general ensemble variable selection techniques are pruned in the sense that the members with the highest prediction errors are trimmed. A very important related algorithm is the fast and robust Bootstrap (FRB), initially introduced in Salibián-Barrera and Zamar (2002); Salibián-Barrera et al. (2008) for regression MM-estimators. The main idea is that standard Bootstrapping of robust estimators would take a long computational time and that a simple uniform Bootstrapping may would result in having resamples with more than half of the instances being contaminated with a considerable probability. They derived a linear update rule for the robust estimator that downweights outlying instances. These weights are derived by a robust MM-estimator which allows for identifying such instances according to the absolute value of their regression residual. Model selection using the FRB has been proposed by Salibián-Barrera and Van Aelst (2008). The idea is to approximate the prediction error of the model built on a particular set of predictors by using FRB and to select the predictor set for which the prediction error was minimal. Their strategy suffers from the lack of scalability for data sets with a large number of predictors since they have to compute the prediction error for all possible models whose total number is $2^p - 1$, and even their backward strategy becomes infeasible for high p.

5.2 Trimmed stability selection

In this paper, we usually intend to measure the quality of the resample-specific models on an in-sample loss basis, similarly as outlying instances are detected by their individual in-sample loss as for example in Alfons et al. (2013). More precisely, if the contamination of a certain resample has caused the algorithm to select wrong variables, the in-sample loss should be high compared to another resample where a sufficiently well model has been selected which contains enough of the true predictors.

Our trimmed Stability Selection works as follows. We generate B resamples of size n_{sub} from the training data and apply the model selection algorithm, for example, Lasso or Boosting, on each resample which selects a model $\hat{S}^{(b)} \subset \{1, ..., p\}$ for $b = 1, ..., B$ and which computes coefficients $\hat{\beta}^{(b)}$. Let $I^{(b)}$ be the index set of the rows that have been selected by the resampling algorithm, i.e., $I^{(b)} \in \{1, ..., n\}^{n_{\text{sub}}}$ for Bootstrapping resp. $I^{(b)} \subset \{1, ..., n\}$ with $|I^{(b)}| = n_{\text{sub}}$ and $I_k^{(b)} \neq I_l^{(b)}\ \forall k \neq l, k, l = 1, ..., n_{\text{sub}}$, for subsampling. For the b-th resample, we compute the in-sample loss

$$L^{(b)} = \frac{1}{n_{\text{sub}}} \sum_{i \in I^{(b)}} L(Y_i, X_i \hat{\beta}^{(b)}) \tag{16}$$

for a loss function $L : \mathcal{Y} \times \mathcal{Y} \to [0, \infty[$. Although, in regression, losses are usually continuous, we cannot exclude that there are ties, for example, if two resamples are identical

or if a trimmed loss is used. In this case, we perform random tie breaking. Note that there is no evidence that the models computed on identical resamples are identical since the model selection algorithm may include stochastic components like the SLTS whose result depends on the randomly chosen initial configurations.

For the set

$$I^{trim}(\gamma) := \left\{ b \in \{1, ..., B\} \;\middle|\; \sum_{b'=1}^{B} I(L^{(b')} \geq L^{(b)}) \leq \lfloor \gamma B \rfloor \right\} \tag{17}$$

with the trimming rate $\gamma \in [0, 1[$, we compute the trimmed aggregated selection frequencies

$$\hat{\pi}_j^{\gamma} = \frac{1}{B - \lfloor \gamma B \rfloor} \sum_{b \in \{1,...,B\} \setminus I^{trim}(\gamma)} I(j \in \hat{S}^{(b)}) \tag{18}$$

for all $j = 1, ..., p$, and let $\hat{\pi}^{\gamma} := (\hat{\pi}_j^{\gamma})_{j=1}^{p}$. The actual identification of the stable model works as usual, based on the $\hat{\pi}_j^{\gamma}$.

As for the choice of the trimming rate γ, note that the main computational time is used for the computation of the individual models $\hat{S}^{(b)}$ which is not affected by γ. Therefore, one can select γ afterwards, for example, using cross-validation so that one computes the stable model corresponding to each element of some grid of trimming rates and finally selects the stable model which corresponds to the smallest cross-validated error.

We also want to elaborate why we suggest to prefer the in-sample losses and not out-of-sample losses, for example, on the remaining instances that are not part of the respective subsample. Assume that one has a clean subsample. Then, the model should perform well on this subsample, i.e., have a low in-sample loss, while the out-of-sample loss would be high if some of the remaining instances contain large outliers. On the contrary, when computing a model on a subsample that mostly contains contaminated instances, the in-sample loss should be high due to the contaminated instances not having a structure that could be well-approximated by the computed model, but the out-of-sample loss on mostly clean instances should be large as well. Therefore, when using the remaining instances and computing the out-of-sample losses, one may not be able to distinguish between clean and contaminated subsamples appropriately. Nevertheless, for illustration, we will also make simulations where the out-of-sample loss is used.

The *Trimmed Stability Selection (TrimStabSel)* is described by the following algorithm:

Algorithm 1 Trimmed Stability Selection

Initialization: Data $D \in \mathbb{R}^{n \times (p+1)}$, number n_{sub} of instances per resample, resampling procedure, number B of resamples, either threshold π_{thr} or number q of stable variables, model selection algorithm, loss function L, trimming rate γ
for $b = 1, ..., B$ **do**
  Draw a resample $D^{(b)} = (X^{(b)}, Y^{(b)}) \in \mathbb{R}^{n_{sub} \times (p+1)}$ from D
  Apply the model selection algorithm to $D^{(b)}$
  Get a predictor set $\hat{S}^{(b)}$ and coefficients $\hat{\beta}^{(b)}$
  Evaluate the in-sample loss $L^{(b)}$ from Eq. 16
end for
Flag the $\lfloor \gamma B \rfloor$ resamples with the highest losses as outlying
Compute the trimmed aggregated selection frequencies as in Eq. 18
Compute the stable model according to π_{thr} or q and $\hat{\pi}^{\gamma}$

Now, we have to analyze the effect of this trimming procedure on the Stab-BDP. We abstain from detailing out each of the cases considered in Theorem 2 again but formulate a universal result which can be easily adapted to all the individual cases.

Theorem 3 *For the robustness gain of TrimStabSel with trimming level γ in comparison with the non-trimmed Stability Selection, let $\lceil K \rceil$ be the (rounded) number of broken models necessary in order to let the respective non-trimmed Stability Selection break down. Assuming that there are k_γ broken models in the set of the $\lfloor \gamma B \rfloor$ trimmed models, $k_\gamma \in \{0, 1, ..., \lfloor \gamma B \rfloor\}$, for TrimStabSel, the number K is replaced by $k_\gamma + (B - \lfloor \gamma B \rfloor)K/B$.*

Proof Since the number of aggregated models decreases from B to $B - \lfloor \gamma B \rfloor$, K has to be multiplied by $(B - \lfloor \gamma B \rfloor)/B$ in order to take the increasing effect of the individual non-trimmed models into account. The only missing feature is the number k_γ of contaminated trimmed models which increases the allowed number of sufficiently contaminated resamples by k_γ. □

6 Simulation study

We now investigate the impact of our proposed outlier scheme on model selection and the performance of TrimStabSel.

We consider a variety of scenarios that differ by n, p, the fraction of outlying cells and the signal to noise ratio (SNR). In all scenarios, there are $s_0 = 5$ relevant variables and the corresponding components of the coefficient β are i.i.d. $\mathcal{N}(4, 1)$-distributed. The cells of the regressor matrix are i.i.d. $\mathcal{N}(5, 1)$-distributed. In the regression settings, the responses are computed by $Y_i = X_i\beta + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ i.i.d. for all i where σ^2 is set so that a specific signal to noise ratio is valid. In the classification settings, we compute $X\beta$ and $\eta_i := \exp(\overline{X_i\beta})/(1 + \exp(\overline{X_i\beta}))$ where $\overline{X_i\beta} := X_i\beta - \text{mean}(X\beta)$. The responses Y_i are drawn according to $Y_i \sim Bin(1, \eta_i)$. As there is no opportunity to simulate a target SNR, we distinguish three situations by drawing $\beta_j \sim \mathcal{N}(\mu, 1)$ so that a higher $|\mu|$ corresponds to a higher

SNR. As for the outlier configuration, we consider different $\tilde{m} \leq n$ and for each $\tilde{m}$, we randomly select $\tilde{m}$ rows of the regressor matrix so that in each of these rows, the value in the cells corresponding to the five relevant variables are replaced by zero. One may consider our contamination scheme as a structured form of cell-wise contamination as the instances are not fully contaminated (as in case-wise contamination setting) but as all cells corresponding to the relevant predictors are contaminated in each selected instance.

We consider different model selection algorithms, namely, L_2-Boosting (Bühlmann & Yu, 2003; Bühlmann & Hothorn, 2007), LogitBoost [see Bühlmann and Van De Geer (2011)] and SLTS (Alfons et al., 2013). The stable model is always derived rank-based with $q = 5$. In practice, one would either choose π_{thr} according to its connection with the per-family error rate and the average number of selected variables in each of the B models (Meinshausen & Bühlmann, 2010; Hofner et al., 2015) or q according to an educated guess for the true dimensionality. If one does not have sufficient knowledge about these components, one may try to find a suitable value for π_{thr} or q empirically by investigating the out-of-sample performance of different stable models (based on different values of π_{thr} resp. q) as suggested in Werner (2022a). As this paper is solely about TrimStabSel, we fix q to the true dimension in our experiments. Another question is how to select the trimming rate γ in practice. The selection of a trimming rate or other hyperparameters that control the robustness of an algorithm is indeed a non-trivial problem which, for example, has been considered in Rieder et al. (2008) from the perspective of efficiency. In practice, for TrimStabSel, assuming that the hyperparameters q or π_{thr} also have to be assessed, we suggest that one may select both γ and q or π_{thr}, respectively, empirically.

We are aware of the fact that a non-robust loss function is problematic in the presence of contamination when one aims at comparing different models (and subsamples here as they are indirectly compared by the models trained on them). This problem cannot even be alleviated using an established robust aggregation such as a trimmed squared loss since the contamination rate may exceed 0.5 on resamples (apart from the problem how to select the trimming rate), or a bounded loss such as Tukey's biweight due to making moderate and large residuals incomparable, maybe hindering the comparison of different subsamples. Therefore, for illustration, we use the squared loss $L(u, u') = (u - u')^2$ for the regression setting and the negative binomial likelihood or the AUC-loss, as implemented in the `Binomial()` and `AUC()` family in the R-package `mboost`, respectively, for the classification setting, as loss function L in order to rank the individual B models.

We evaluate the performance of the Stability Selection variants by computing the mean true positive rate (TPR) over V repetitions where for each $v = 1, ..., V$, we generate an independent data set. Moreover, we compute the fraction of breakdowns, i.e., where no relevant variable has entered the stable model as well as the fraction of cases where the stable model is perfect, i.e., it consists only of the s_0 relevant variables. These quantities are then plotted against the number $\tilde{m}$ of outlying instances. Note that the breakdown rate can be interpreted as empirical version of the probabilities computed in Sect. 4.

6.1 (Trimmed) stability selection with L_2-boosting

The scenario specifications are given in Table 1. We consider the SNRs 1, 2 and 5 and for each SNR and each value for $\tilde{m}$ from the set given in Table 1, we generate $V = 1000$ independent data sets and apply all four Stability Selection variants specified in Table 1. We use the function `glmboost` from the R-package `mboost` (Hothorn et al., 2017) with `family=Gaussian()`, 100 iterations and a learning rate of 0.1.

Table 1 Scenario specification for L_2-boosting and LogitBoost as model selection algorithms

Scen	p	n	$\tilde{m}$	n_{sub}	StabSel	TrimStabSel					
					B	B	γ	B	γ	B	γ
1	25	50	{0,1,...,9}	25	100	100	0.5	100	0.75	100	0.9
2	50	100	{0,1,...,9}	50	100	100	0.5	100	0.75	100	0.9
3	200	200	{0,1,...,20}	100	100	100	0.5	1000	0.9	1000	0.95
4	500	200	{0,1,...,15}	100	100	100	0.75	1000	0.9	1000	0.95

The results in Figs. 1, 2, 3 and 4 show the non-surprising facts that the TPR increases with increasing SNR and that the TPR curves and the curves representing the relative frequency of perfect models decrease with the number of outliers while the curves representing the relative frequency of a breakdown increase. For a low number of outliers, in particular for clean data, the performance of the TrimStabSel variants is usually worse than that of the non-trimmed Stability Selection due to the loss of evidence by trimming (good) models away. A characteristic aspect of all curves is that once contamination occurs, the TrimStabSel variants show better performance than the non-trimmed Stability Selection but that the robustness and performance gain decreases once too many cells are contaminated. The reason is that at some point, the expected number of contaminated subsamples becomes too high so that even TrimStabSel with the configurations in Table 1 is distorted. Note that the exact contamination rate where even TrimStabSel breaks down cannot be computed, but considering scenario 1, $\tilde{m} = 6$ leads to the probability $P(Hyp(50, 44, 25) < 25) \approx 0.989$ to draw a contaminated subsample and therefore to the probability $P(Bin(100, P(Hyp(50, 44, 25) < 25)) \leq 90) \approx 2.05 \times 10^{-7}$ to

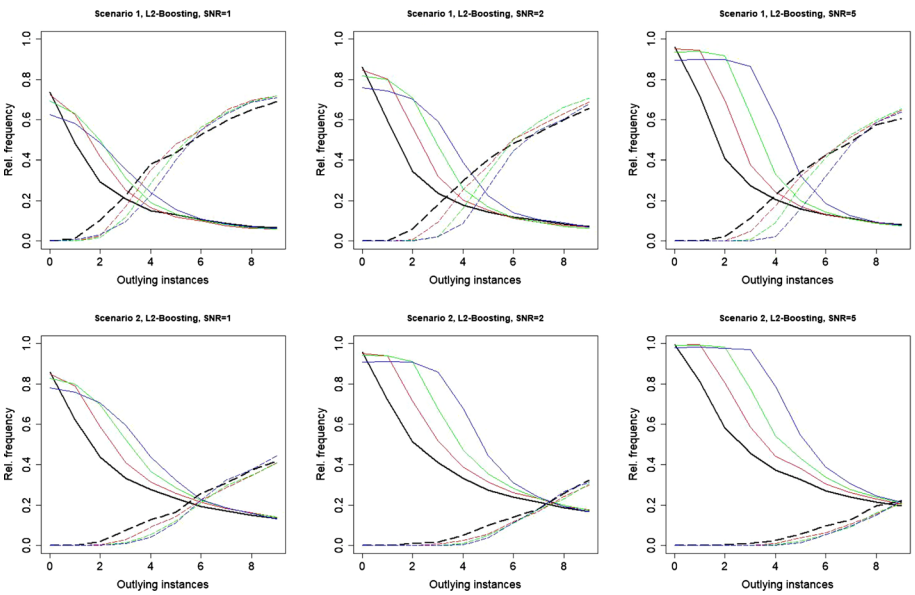

Fig. 1 Results for scenarios 1 and 2 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

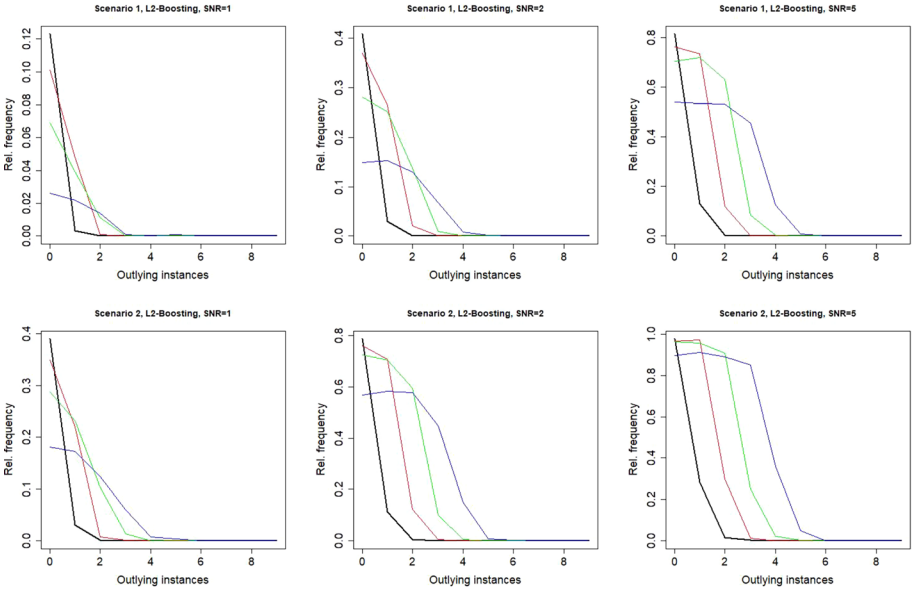

Fig. 2 Relative frequencies of perfect stable models for scenarios 1 and 2 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

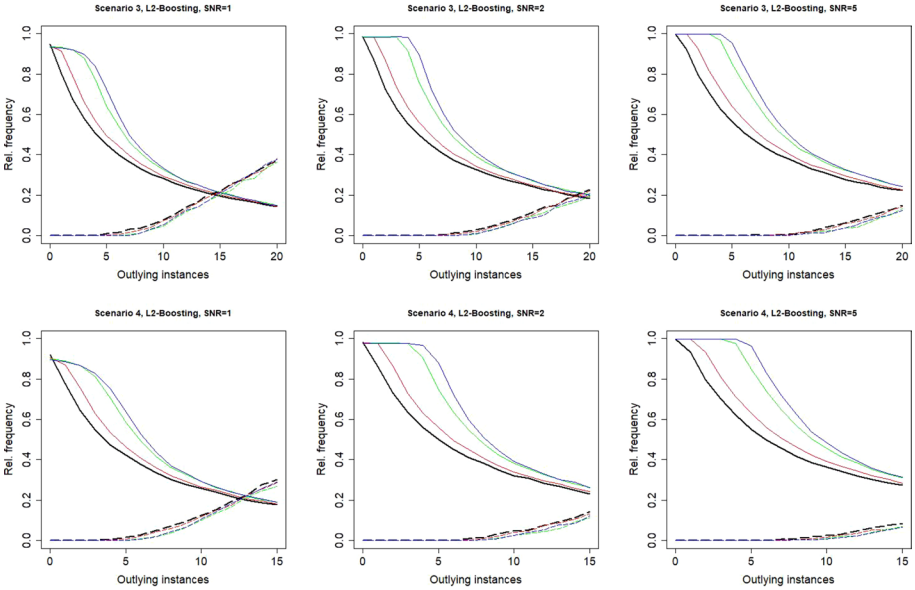

Fig. 3 Results for scenarios 3 and 4 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

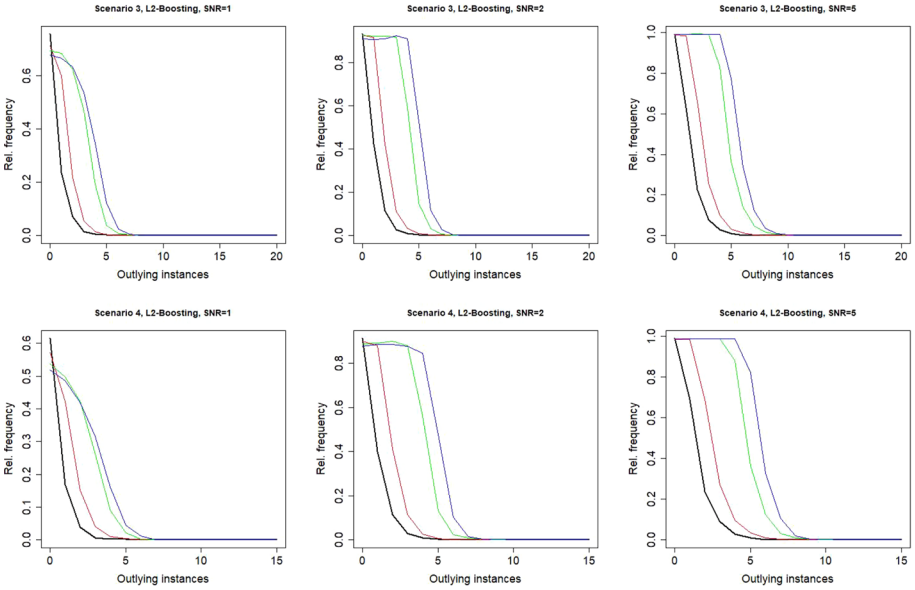

Fig. 4 Relative frequencies of perfect stable models for scenarios 3 and 4 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

have at most 90 out of 100 contaminated subsamples. The inherent robustness of Stability Selection safeguards against a breakdown of the TrimStabSel variants here which would otherwise be very likely.

At small contamination rates, for scenario 1 until around 12% and for scenario 2 until around 7%, TrimStabSel considerably improves model selection, especially for a high SNR. For example, the relative breakdown frequency in scenario 1 with an SNR of 5 and $\tilde{m} = 4$ is around 10 times as high for the non-trimmed Stability Selection as for the third variant of TrimStabSel while the TPR is three times as high and even more than three times as high for $\tilde{m} = 3$. The results in scenario 2–4 look similar as for scenario 1. For an SNR of 5, one can observe even near perfect results for at least the third TrimStabSel variant for low contamination rates up to around 3%.

6.2 (Trimmed) stability selection with LogitBoost

We use the same specifications as in Table 1, but as we cannot targetedly let the data have some specified SNR, we generate the relevant β_j according to a $\mathcal{N}(\mu, 1)$-distribution with $\mu \in \{1, 4, 8\}$ where higher means make, in expectation, the signals stronger and the SNR higher. Again, we use $V = 1000$. We again use `glmboost`, here with `family=Binomial(link='logit')` and let the other hyperparameters be as for L_2 -Boosting.

The results, depicted in Figs. 5, 6, 7 and 8, look similarly as in the previous subsection. One can observe a more compressed shape of the TPR curve for the case $\mu = 1$ while the curves corresponding to the different Stability Selection variants show a considerable

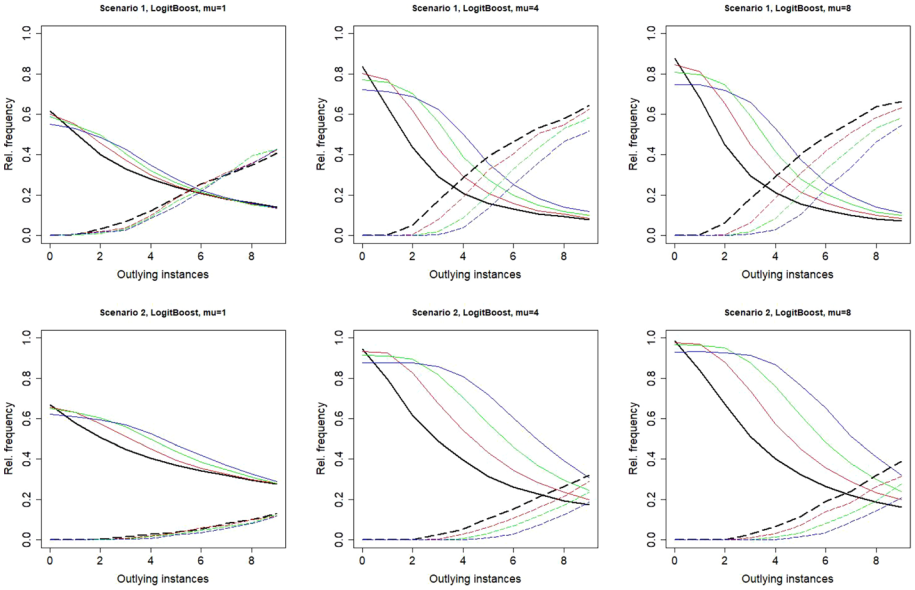

Fig. 5 Results for scenarios 1 and 2 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

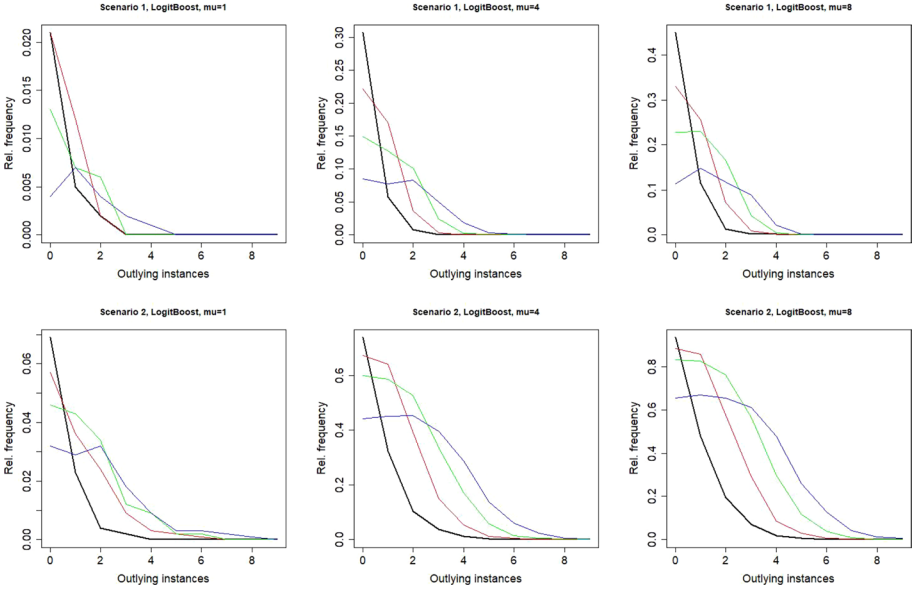

Fig. 6 Relative frequencies of perfect stable models for scenarios 1 and 2 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

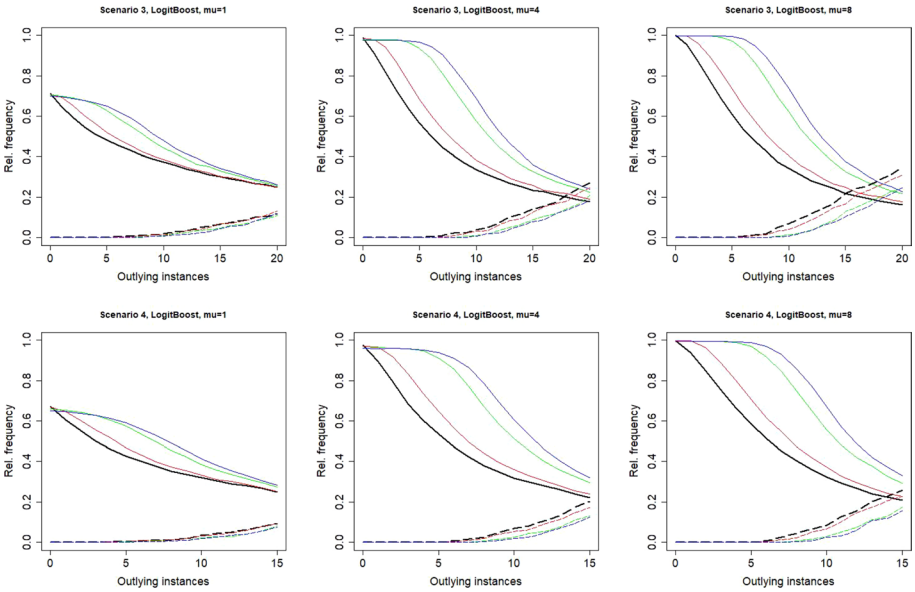

Fig. 7 Results for scenarios 3 and 4 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

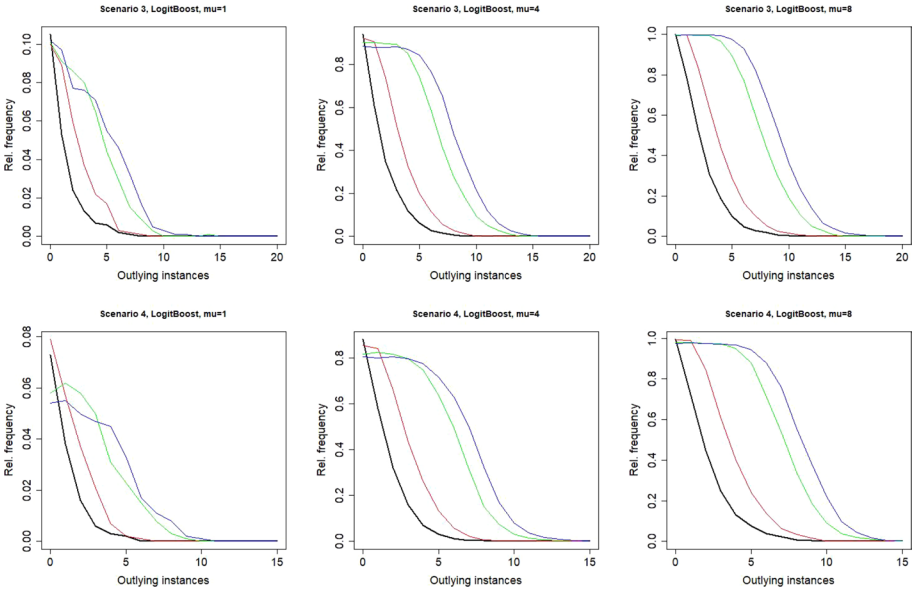

Fig. 8 Relative frequencies of perfect stable models for scenarios 3 and 4 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

margin. In our opinion, these behaviours are a direct consequence of the SNRs. The empirical mean SNRs on our data sets, computed as $Var(X\beta)/Var(Y)$ [the reciprocal value of the noise to signal ratio from Friedman et al. (2001)], is between 30 and 40 for $\mu = 1$ and more than 1200 for $\mu = 8$. Although the interpretation of this SNR is not identical to the interpretation of the SNR in the regression setting, the margins between the curves for high values of μ reflect the behaviour from the previous subsection, here even stronger.

6.3 (Trimmed) stability selection with SLTS

Due to the inherent robustness of SLTS, we additionally allow for situations with high contamination radii. More precisely, we use the set $\{0, 1, ..., 10, 15, 20, 25\}$ for scenario 1, $\{0, 1, ..., 10, 15, 20, ..., 45, 50\}$ for scenario 2, $\{0, 1, ..., 20, 30, ..., 70\}$ for scenario 3 and $\{0, 1, ..., 15, 20, 30, ..., 70\}$ for scenario 4.

We only consider regression scenarios here, i.e., we use `family=Gaussian()` in the `sparseLTS` function from the R-package `robustHD` (Alfons, 2016). We use a trimming rate in SLTS of 0.25 and for the penalty parameter, we propose the grid $\{0.05, 0.55, ..., 4.55\}$ and let the best element be chosen data-driven by a winsorization strategy corresponding to `mode='fraction'` in `sparseLTS`. Due to the computational complexity, we set $V = 100$. The remaining configurations are as in Table 1.

The results in Figs. 9, 10, 11 and 12 show that the region where the non-trimmed Stability Selection leads to a better performance than the trimmed variants is considerably extended in contrast to the former experiments. This can be explained by the fact that SLTS with an internal trimming rate of 25% itself has a BDP of around 25% while

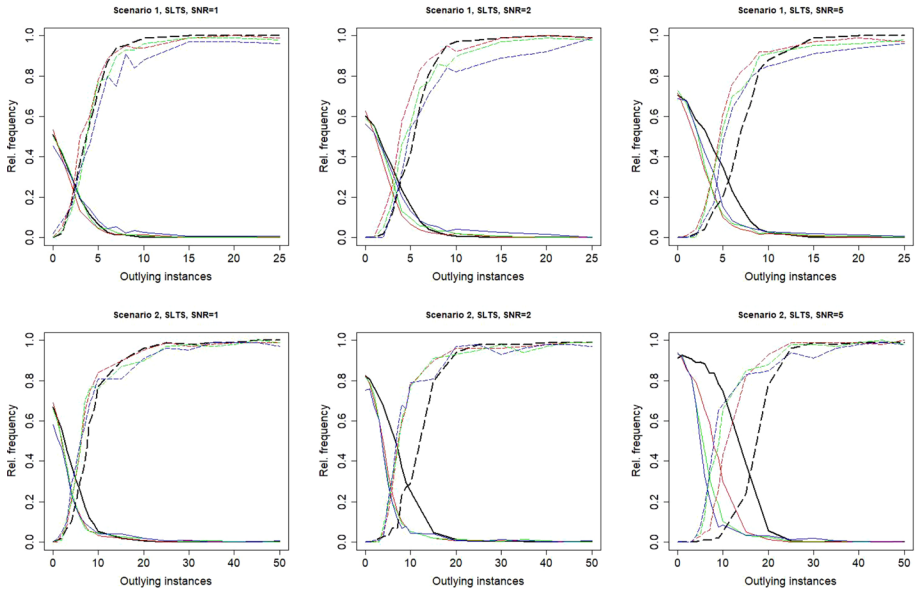

Fig. 9 Results for scenarios 1 and 2 with SLTS as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

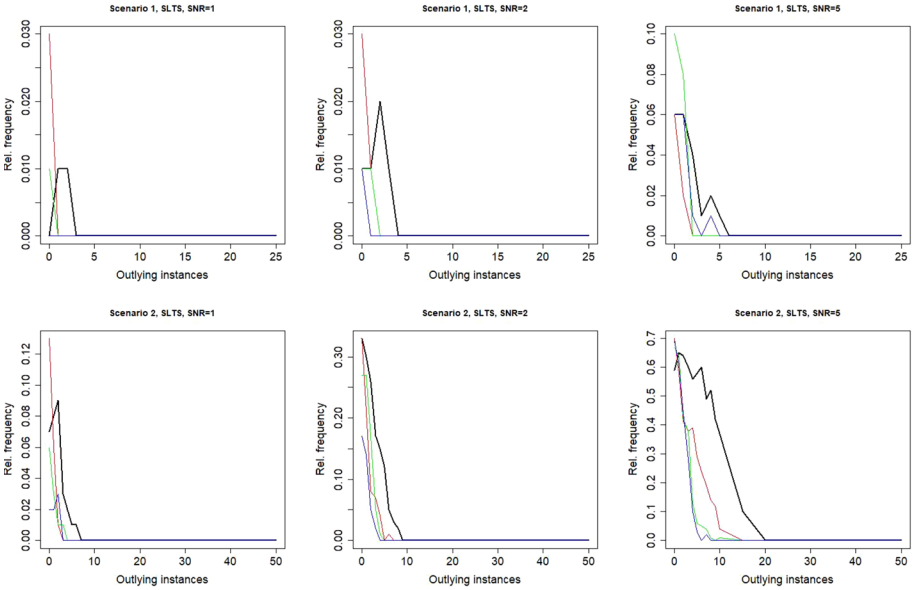

Fig. 10 Relative frequencies of perfect stable models for scenarios 1 and 2 with SLTS as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

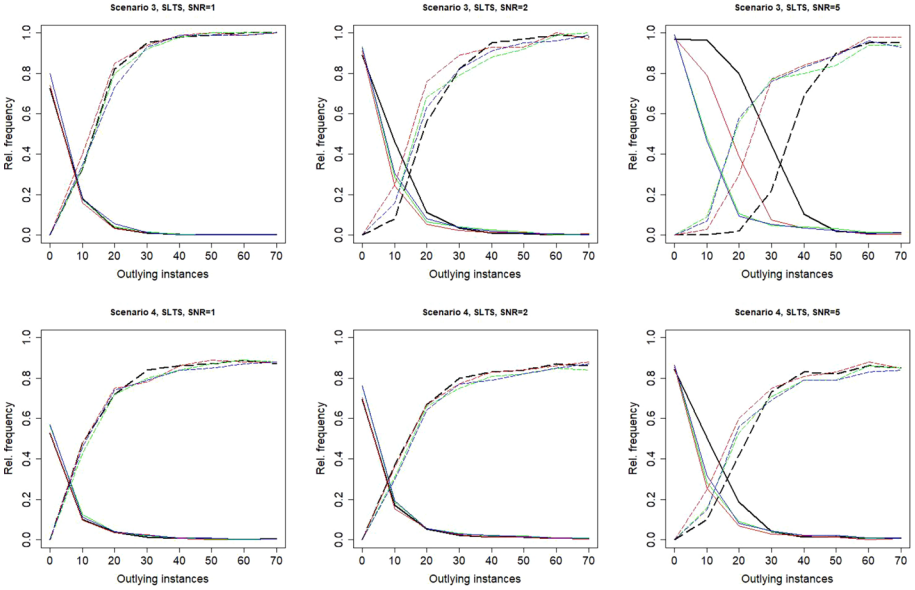

Fig. 11 Results for scenarios 3 and 4 with SLTS as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

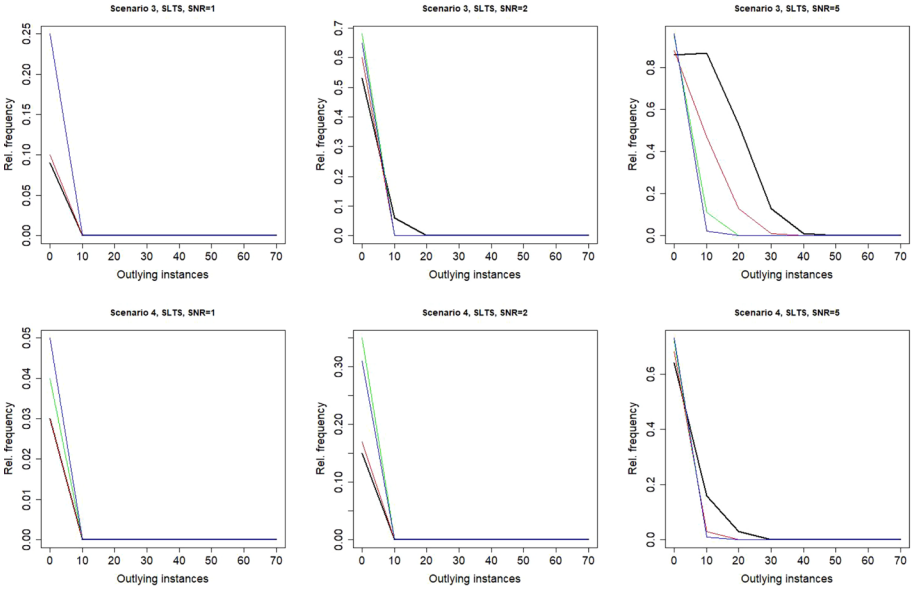

Fig. 12 Relative frequencies of perfect stable models for scenarios 3 and 4 with SLTS as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

L_2-Boosting and LogitBoost have a BDP of 0. Therefore, the trimmed Stability Selection pays off late, more precisely, once the contamination rate is sufficiently high to let SLTS break down on many subsamples, which is even more delayed for high SNR values due to the better performance of the underlying model selection in these cases. Also, the performance loss of TrimStabSel for low contamination rates is more considerable for low SNR values than for high SNR values.

It should be noted that the TPR starts with lower values for $\tilde{m} = 0$ than in the previous experiments. In particular, the relative fraction of perfect models is very low for scenario 1, even without contamination and with an SNR of 5, in contrast to the experiments with L_2-Boosting. The reason could be that, similarly as in TrimStabSel where the trimming decreases the evidence and makes the stable model therefore more fragile, trimming instances away in SLTS decreases the evidence of the fitted model and therefore its performance. The loss in efficiency of robust methods seems to carry over to model selection itself.

One can observe that trimming eventually pays off, but due to the fraction of broken models already being very high resp. the mean TPR already being very low, the improvement itself granted by TrimStabSel may no longer be reasonable as the relative fraction of broken models is still very high, for example, in scenario 2 with an SNR of 2, 99% of the models have broken down for the non-trimmed Stability Selection while for the third version of TrimStabSel, still 89% of the models have broken down.

Experiments with a higher B and α so that a larger number of models is considered for aggregation showed a slight improvement of TrimStabSel, but we omit detailed figures here as they essentially resemble the ones before.

Finally, we directly compare the performance of the non-trimmed Stability Selection with SLTS with that of the TrimStabSel variants with L_2-Boosting in order to check whether TrimStabSel can be beneficial or if the traditional Stability Selection with a robust model selection algorithm is always superior. This is done exemplarily for scenarios 2 and 3 and depicted in Fig. 13. One can observe that for high signal-to-noise ratios, the standard Stability Selection with SLTS is nearly always superior, both in terms of mean TPR and mean breakdown rate, than any TrimStabSel variant with L_2-Boosting. Interestingly, for low signal-to-noise ratios, while the robustness in terms of VSBDP is higher for the Stability Selection with SLTS than for the TrimStabsel variants with L_2-Boosting, the precision in terms of mean TPR is superior for the TrimStabSel variants with a high trimming rate, indicating that is strongly depends not only on the contamination rate but also on the noise level which model aggregation strategy should be selected. Summarizing, TrimStabSel with a non-robust model selection algorithm can outperform the traditional Stability Selection with a robust model selection algorithm.

6.4 Threshold-based stability selection

We repeat all experiments with L_2-Boosting and LogitBoost with a threshold-based Stability Selection with a fixed threshold of $\pi_{thr} = 0.75$. In addition to the standard setting $q = 5$ for the number of variables in the rank-based Stability Selection, we also run simulations with $q = 8$ for a better comparison with the threshold-based variant as the true number of

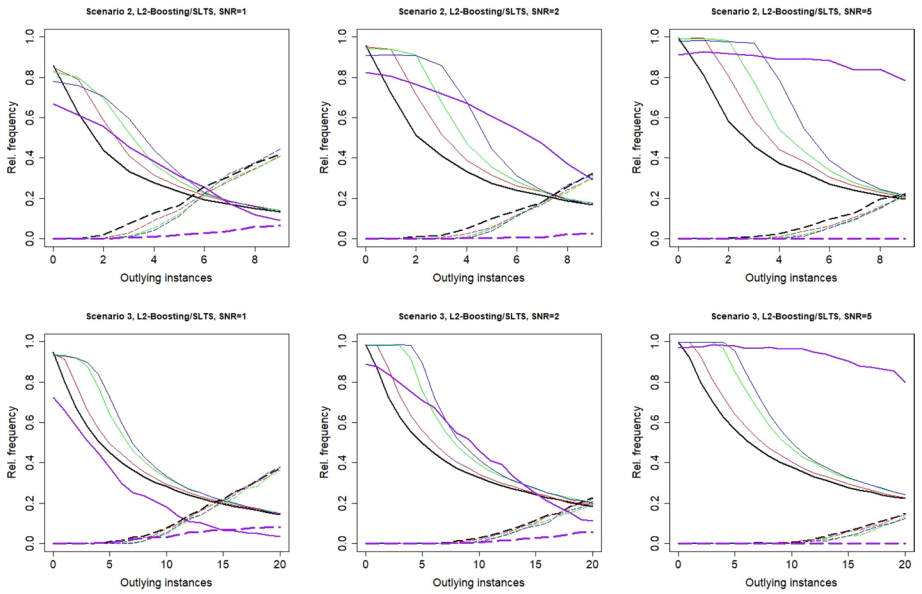

Fig. 13 Results for scenarios 2 and 3 with the traditional Stability Selection with SLTS (thick purple lines) and the TrimStabSel variants with L_2-Boosting. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively, all with L_2-Boosting as model selection algorithm (Color figure online)

variables is not known in practice, so $q = 5$ grants the rank-based variant an unfair advantage here.

One can observe in Figs. 14, 15, 16, 17, 18, 19, 20, 21, 22 and 23 that the characteristics of the curves are similar as in the rank-based case. However, one can clearly see that the performance of the threshold-based variant is usually worse than the rank-based variant as the TPRs are lower and the relative breakdown frequencies attain high values more quickly, which corresponds to our theoretical results in Theorem 2. As expected, the TPR slightly increases for the case $q = 8$ (note that the accuracy does decrease) and that the empirical breakdown rate slightly decreases in comparison with the case $q = 5$ for the rank-based variant. Nevertheless, the results are still considerably better than for the threshold-based variant.

6.5 Large outliers

We repeat the experiments for scenarios 2 and 3 with L_2-Boosting and SLTS as model selection algorithms with an alternative contamination scheme where we replace cells with the value 50 instead of replacing them with zeroes. As the cells are generated from a $\mathcal{N}(5, 1)$-distribution, these values will definitely appear as large outliers. From here on, we only use the rank-based variant due to the reduced performance of the threshold-based variant experienced in the previous subsections.

One can observe in Figs. 26, 27, 28, 29, 30 and 31 that the curves show a similar behaviour as for the case of replacing the values of the contaminated cells with zeroes, but the individual curves corresponding to the four Stability Selection variants are much better separated now. It should be noted that the usage of SLTS as model selection algorithm leads to superior performance than a TrimStabSel with L_2-Boosting as model selection algorithm here.

6.6 Out-of-sample loss

We repeat all experiments for scenario 2 and 3 with L_2-Boosting, LogitBoost and SLTS as model selection algorithms where TrimStabSel is based on the out-of-sample losses and not on the in-sample losses. As for the out-of-sample losses, we always consider the instances in the training data that were not selected in the subsample on which the respective model has been trained.

It is revealed in Figs. 32 and 33 that it has a significant effect whether one considers the in-sample loss or the out-of-sample loss for TrimStabSel when using L_2-Boosting as model selection algorithm. While the TrimStabSel variants with the in-sample losses showed superior performance than the non-trimmed Stability Selection, the performance of the TrimStabSel variants with the out-of-sample performance is inferior to that of the non-trimmed Stability Selection, confirming our argumentation for using the in-sample losses here. As for LogitBoost as model selection algorithm, Figs. 34 and 35 indicate that this observation also holds here but that the difference between the performances is much smaller than in Figs. 32 and 33, maybe resulting from a generally higher signal-to-noise ratio.

Finally, the curves Figs. 36 and 37 are very similar to their counterparts in Figs. 9, 10, 11 and 12, indicating that with a robust model selection algorithm, it may not be important whether to consider the in-sample or the out-of-sample losses in the TrimStabSel. A possible reason is that, due to using the usual squared loss when computing the in-sample and

out-of-sample performance of the individual models, robust methods that avoid to overfit outliers result in an in-sample performance that is worse than that of a model computed by a non-robust method. At the same time, due to avoiding to overfit, the out-of-sample performance may, on the contrary, be better than for non-robust methods, making the trimming procedure less susceptible for the choice of whether to use the in-sample or the out-of-sample loss.

6.7 Out-of-sample loss and large outliers

We combine the out-of-sample loss approach with the situation of large outliers.

One can observe in Figs. 38, 39, 40 and 41 that the combination of large outliers and the consideration of the out-of-sample loss for the validation of the individual models significantly decreases the performance of both the non-trimmed Stability Selection and the trimmed variants compared to the scenario where the out-of-sample loss is combined with the situation of replacing cells with zeroes. The comparison of the results here with the situation where the in-sample loss is used in the presence of large outliers is highly interesting as Figs. 26, 27, 28, 29, 30 and 31 have revealed the benefit of the trimmed variants, according to the in-sample loss, in this case, while using the out-of-sample loss is clearly disadvantageous in Figs. 38, 39, 40 and 41.

Again, Figs. 42 and 43 indicate that TrimStabSel with SLTS as model selection algorithm is indifferent whether the in-sample or the out-of-sample loss is considered, as the curves look very similar to those in Figs. 30 and 31.

6.8 RRBoost

We apply the RRBoost algorithm (Ju & Salibián-Barrera, 2021) as an alternative robust model selection algorithm. We use the implementation in the R-package `RRBoost` (Ju & Salibián-Barrera, 2020). In our experiments, we used the default configurations of the `Boost` function in the package `RRBoost`.

The results in Figs. 44, 45, 46 and 47 show interesting differences between the case where SLTS is used as model selection algorithm, as shown in Figs. 9, 10, 11 and 12. One can observe in scenario 2 that the Stability Selection with RRB is much more robust and achieves a much higher mean TPR than the Stability Selection with SLTS. Moreover, the TrimStabSel with a low trimming rate achieves similar performance concerning mean TPR than the non-trimmed Stability Selection, while TrimStabsel with a high trimming rate achieves a better robustness than the non-trimmed Stability Selection, in contrast to the Stability Selection with SLTS where trimming generally worsened the performance. In scenario 3, it turns out that TrimStabSel with a high trimming rate is also beneficial for the mean TPR. Similar observations can be made for the relative frequency of perfect models, which is higher with RRB than with SLTS as model selection algorithm and where trimming is beneficial in scenario 3 but not in scenario 2.

For scenario 2 with an SNR of 5, we also applied TrimStabSel with RRB for the cases of out-of-sample loss, large outliers, and their combination. Figures 46 and 47 reveal that it depends on the contamination rate whether trimming pays off, in contrast to the case of SLTS as model selection algorithm were trimming never resulted in better performance for this particular scenario, as shown in Fig. 30. Similar observations can be made for the case of using out-of-sample loss where trimming is considerably better for rather high contamination rates concerning the robustness, in harsh contrast to the case with SLTS as shown in

Fig. 32. The results for the case of the out-of-sample loss in combination with large outliers is comparable. Interestingly, the rate of perfect models is generally better for the case of SLTS than when using RRB.

6.9 AUC-Boosting

Finally, we use AUC-Boosting which considers the AUC-loss, i.e., 1 minus the AUC. Since this loss function is not differentiable, a smooth approximation is used [cf. Hothorn et al. (2017)]. The results, depicted in Figs. 48 and 49 for the in-sample loss and in the top right part of Figs. 50 and 51 show almost flat curves for the trimmed Stability Selection. A closer inspection revealed that AUC-Boosting tends to overfit, leading to a perfect in-sample performance so that all samples are indistinguishable, so that selection of the "best" $(1-\gamma)$-fraction is just done randomly.

For the top left and bottom part of Figs. 50 and 51, the out-of-sample losses were used, resulting in a considerable improvement of the performance when combining AUC-Boosting with TrimStabSel instead of the traditional Stability Selection.

7 Conclusion

We intended to make a step towards the unification of sparse model selection, robustness and stability in order to lift the understanding of robustness from the rows of a data matrix to the columns and investigated how contamination can affect model selection. We started with the introduction of the variable selection breakdown point and an outlier scheme which allows a very small number of contaminated cells to completely distort variable selection, making a robustification in the usual sense that provides coefficients whose norm is always bounded obsolete if no relevant variable is considered.

We extended the notion of the resampling breakdown point, which quantifies the relative fraction of outlying instances so that the probability that a resample is contaminated too much exceeds some threshold, by the Stability Selection BDP which we computed for different scenarios where we postulate different effects of outliers onto model selection due to the absence of concrete results in literature. Our analysis reveals that a Stability Selection where the stable model is given by the best q variables for a pre-defined q can be expected to be more robust than the standard threshold-based Stability Selection.

Finally, we propose a Trimmed Stability Selection which considers only the best resamples, based on the in-sample losses, when aggregating the models. A simulation study reveals the potential of this Trimmed Stability Selection to robustify model selection, although it evidently inherits the necessity to find appropriate hyperparameter configurations. The simulations also prove the alarming fragility of variable selection, even for an very low number of outlying cells, if the outliers are targetedly placed onto the relevant columns. In particular, regarding the rapid performance decrease of the non-trimmed Stability Selection with L_2-Boosting as model selection algorithm, one has to keep in mind that even 2 resp. 5 outlying instances in scenario 1 and 2 resp. 3 and 4 suffice to let nearly no stable model be perfect, accompanied with a somewhat decreased mean TPR, so the cell-wise contamination rates range from 25/100500 in scenario 4 to 10/1275 in scenario 1.

We recommend to consider a trimmed Stability Selection with a non-robust model selection algorithm in situations where the contamination rate can be expected to be low and the noise level to be high. In such settings, the trimmed Stability Selection with a non-robust

model selection algorithm like L_2-Boosting shows a significant improvement concerning mean TPR, breakdown rate and the relative number of perfect stable models in contrast to the non-trimmed Stability Selection, and can even outperform the Stability Selection combined with a robust model selection algorithm, while being very easy to implement. This avoids the application of robust model selection algorithms which are computationally more expensive and which alone do not follow the stability paradigm which would in fact necessitate to even apply a Stability Selection with a robust model selection algorithm. In cases with larger contamination rates, one however cannot avoid the application of much more expensive robust model selection algorithms, but our simulations revealed that aggregating the resulting models with TrimStabSel instead of the traditional Stability Selection can considerably improve the performance. One should however note that it depends on the model selection algorithm whether an improvement can be achieved. It turns out that one should not combine TrimStabSel with trimming algorithms such as SLTS due to the reduced evidence by trimming training instances in each of the subsamples, reducing their comparability.

We want to emphasize that in our experiments, a robust model selection algorithm seems to show inferior performance than a non-robust model selection algorithm if applied on clean data. Although it is well-known that robust algorithms are less efficient in terms of asymptotic covariance, this loss in efficiency seems to carry over to variable selection itself.

Future research is necessary in order to study the potential of outliers for targeted variable promotion or suppression further. Although our proposed outlier schemes seem to be artificial so that they most probably would not occur by chance, one has to be aware of the attacking paradigm emerging from the deep learning community. Similarly as popular situations where models have to be inferred before attacks can be crafted (see, e.g., Papernot et al. 2017), one could intercept a data transfer, try to detect relevant variables and suppress them targetedly or try to detect certainly non-relevant variables (for example by Sure Independence Screening, see Fan & Lv 2008) in order to targetedly promote them.

A Proof of Theorem 2

Proof

(a) (ii) Now, we consider cell-wise contamination. Let, for $i = 1, ..., n$, a fixed selection of c_i cells in instance i be contaminated, let $Z_l, l = 0, 1, ..., p + 1$, be the number of instances for with $c_i = l$ and let $\tilde{m} := \sum_i c_i$. Let further $Z^{rel}_{l'}, l' = 1, ..., s_0$, denote the number of instances with l' outlying cells in the relevant columns. Let $\check{m}$ be the number of outliers in the response column. Let $\tilde{c}$ be the cell-BDP of the applied model selection algorithm.

First, note that for both the rank-based and the threshold-based Stability Selection, the following two facts obviously hold: The probability that the model selection breaks down in terms of the VSBDP is 1 if the fraction of cell-wise outliers exceeds $\tilde{c}$ in the relevant columns or in the response column; and it is 0 if $c_i \leq \lfloor \tilde{c}(p + 1) \rfloor \ \forall i$ resp. 1 if $c_i > \lfloor \tilde{c}(p + 1) \rfloor \ \forall i$.

Otherwise, there are three ways how to achieve a breakdown as already mentioned when defining the cell-wise scenarios: 1.) The probability that, due to subsampling, the fraction of cell-wise outliers in the whole data matrix becomes at least $\tilde{c}$; 2.) The analogous probability for the set of relevant col-

umns; 3.) The probability that a fraction of at least $\tilde{c}$ of the responses are contaminated which also can cause the wrong variables to be selected according to the assumption in the cell-wise scenarios.

Let

$$p_1 := \sum_{z_0,...,z_{p+1}: \sum_l l z_l \geq \lceil \tilde{c} n_{\text{sub}}(p+1) \rceil} f_{Z,n_{\text{sub}}}(z_0, ..., z_{p+1})$$

where $f_{Z,n_{\text{sub}}}$ for $Z = (Z_0, ..., Z_{p+1})$ represents the probability function of a multivariate hypergeometric distribution with values in $\{(z_0, ..., z_{p+1}) \mid z_0 + ... + z_{p+1} = n_{\text{sub}}\}$, i.e., $f_{Z,n_{\text{sub}}}(z_0, ..., z_{p+1})$ is the probability that when sampling n_{sub} instances without replacement, one gets z_0 out of the Z_0 instances without cell-wise outliers, z_1 out of the Z_1 instances with one cell-wise outlier and so forth. Moreover, let

$$p_2 := \sum_{\tilde{z}_0,...,\tilde{z}_{s_0}: \sum_l l \tilde{z}_l \geq \lceil \tilde{c} n_{\text{sub}} s_0 \rceil} f_{Z^{rel},n_{\text{sub}}}(\tilde{z}_0, ..., \tilde{z}_{s_0})$$

for $Z^{rel} = (Z_0^{rel}, ..., Z_{s_0}^{rel})$. The probabilities p_1 and p_2 correspond to one resample, so we define P_1 and P_2 corresponding to a sufficient contamination of sufficiently many resamples, leading to the probabilities

$$P_1 := P(Bin(B, p_1) > \lceil B(\max(\hat{\pi}_j^+) - \pi) \rceil) \tag{19}$$

and

$$P_2 := P(Bin(B, p_2) > \lceil B(\max(\hat{\pi}_j^+) - \pi) \rceil) \tag{20}$$

that the threshold-based Stability Selection breaks down due to (1) or (2). As for (3), denote the quantity in Eq. (12), where m is replaced by $\check{m}$, by P_3, so that we finally get the probability $\min(P_1, P_2, P_3)$ that the model selection breaks down if the resamples are drawn by subsampling.

Analogously, by the same arguments, the probability of a VSBDP for the threshold-based Stability Selection if the resamples are drawn by Bootstrapping is given by $\min(\check{P}_1, \check{P}_2, \check{P}_3)$ for

$$\check{P}_1 := P(Bin(B, \check{p}_1 > \lceil B(\max(\hat{\pi}_j^+) - \pi) \rceil) \tag{21}$$

where

$$\check{p}_1 := \sum_{z_0,...,z_{p+1}: \sum_l l z_l \geq \lceil \tilde{c} n_{\text{sub}}(p+1) \rceil} f_{Mult}(z_0, ..., z_{p+1})$$

where f_{Mult} represents the density of a multinomial distribution with parameters $(Z_0 / \sum_l Z_l, ..., Z_{p+1} / \sum_l Z_l)$ and n_{sub}; for

$$\check{P}_2 := P(Bin(B, \check{p}_2 > \lceil B(\max(\hat{\pi}_j^+) - \pi) \rceil) \tag{22}$$

where

$$\check{p}_2 := \sum_{\tilde{z}_0,\ldots,\tilde{z}_{s_0}:\sum_l l\tilde{z}_l \geq \lceil \tilde{c} n_{\text{sub}} s_0 \rceil} f_{Mult}(\tilde{z}_0,\ldots,\tilde{z}_{s_0});$$

and for $\check{P}_3$ as in Eq. (13) where m is replaced by $\check{m}$.

By the same arguments as in i) for the rank-based Stability Selection and for the threshold-based Stability Selection with cell-wise contamination above, the probability that the rank-based Stability Selection breaks down is given by $\min(P_1, P_2, P_3)$ for

$$P_v := P(Bin(B, p_v) > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-))\rceil) \tag{23}$$

for $v = 1, 2$ and for P_3 as in Eq. (14), where m is replaced by $\check{m}$, if the resamples are drawn by subsampling; and it is given by $\min(\check{P}_1, \check{P}_2, \check{P}_3)$ for

$$\check{P}_v := P(Bin(B, \check{p}_v) > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-))\rceil) \tag{24}$$

and $\check{P}_3$ as in Eq. (15), where m is replaced by $\check{m}$, if the resamples are drawn by Bootstrapping.

Now, we are ready to compare the robustness of threshold- and rank-based Stability Selection by simply comparing the right-hand sides inside the $P(\cdot)$ -brackets in Eqs. (12) and (14), (13) and (15), (19), (20) and (23) and (21), (22) and (24), respectively, indicating that both variants are equally robust if and only if

$$\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \pi = 0.5(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)),$$

directly proving statement a).

(b) The threshold-based Stability Selection has already been covered in a).

(i) Again, we first consider case-wise contamination.

As for rank-based Stability Selection, it is indeed important that in the optimistic scenario, we can only targetedly promote one non-relevant variable. Therefore, the breakdown of rank-based Stability Selection depends on the terms s, q, $\max_j(\hat{\pi}_j^+)$ and all $\hat{\pi}_k^-$ for $k = 1, \ldots, q$, so one cannot make a universal precise statement. However, there are two extreme cases. If for

$$\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-) =: \Delta$$

the difference between $\max_j(\hat{\pi}_j^+)$ and $\hat{\pi}_k^-$ for $(q-1)$ indices k from $\{1, \ldots, q\}$ is exactly $\Delta/2$, except for $k^* := \text{argmin}_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)$ for which it is Δ, then more than $0.5\lceil B\Delta \rceil$ contaminated samples suffice for a breakdown if the variable corresponding to $\hat{\pi}_{k^*}^-$ is promoted in each of these resamples since the same reasoning as in a) applies, i.e., the difference of the selection probabilities for the worst non-relevant and best relevant variable decreases with a step size of $2/B$. The other extreme case is that all $\hat{\pi}_k^-$ are equal. Then, if $s > \lceil B\Delta \rceil$, even after promoting each of these non-relevant variables in one single contaminated resample does not suffice for a breakdown since there will be at least one remaining one whose aggregated selection frequency was still not promoted. In that case, we can treat $\hat{\pi}_k^-$ (in general, $\min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)$) as

threshold so that the results for the threshold-based Stability Selection are applicable, so the relevant variables have to be suppressed in so many resamples such that the selection frequency of the best of them finally crosses the threshold. Hence, the probability of a VSBDP of rank-based Stability Selection here lies in the interval

$$\begin{aligned}[P(Bin(B, P(Hyp(n, n-m, n_{\text{sub}}) \leq & \lfloor (1-c)n_{\text{sub}} \rfloor)) \\ & > \lceil B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil), \\ P(Bin(B, P(Hyp(n, n-m, n_{\text{sub}}) \leq & \lfloor (1-c)n_{\text{sub}} \rfloor)) \\ & > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil)]\end{aligned} \tag{25}$$

if the resamples are drawn by subsampling and lies in the interval

$$\begin{aligned}[P(Bin(B, P(Bin(n_{\text{sub}}, m/n) \geq & \lceil cn_{\text{sub}} \rceil)) \\ & > \lceil B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil), \\ P(Bin(B, P(Bin(n_{\text{sub}}, m/n) \geq & \lceil cn_{\text{sub}} \rceil)) \\ & > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil)]\end{aligned} \tag{26}$$

if the resamples are drawn by Bootstrapping.

(ii) Now, we consider cell-wise contamination.

As for the rank-based variant, by the same arguments as before, it lies in one of the intervals

$$\begin{aligned}&[P(Bin(B, p_v) > \lceil B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil), \\ &P(Bin(B, p_v) \geq \lceil cn_{\text{sub}} \rceil)) > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil)]\end{aligned}$$

for $v = 1, 2$ with p_1 and p_2 as above or in the interval given in Eq. (25), where m is replaced by $\check{m}$, if the resamples are drawn by subsampling; lies in one of the intervals

$$\begin{aligned}[P(Bin(B, \check{p}_v) > & \lceil B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil), P(Bin(B, \check{p}_v) \\ & > \lceil 0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) \rceil)]\end{aligned}$$

for $v = 1, 2$ with $\check{p}_1$ and $\check{p}_2$ as before; or in the interval given in Eq. (26), where m is replaced by $\check{m}$, if the resamples are drawn by Bootstrapping. These statements are slightly more tricky than for the threshold-based variant since one has intervals. However, for a concrete data set and a concrete model selection algorithm, one value in the respective intervals is realized so that the probability of a breakdown of the Stability Selection is the minimum.

Again, the statement b) is proven by comparing the right-hand sides in the respective $P(\cdot)$-brackets. Due to the interval statements, we can only consider the cases where one variant is definitely more robust than the other one. The rank-based variant is definitely more robust than the threshold-based variant if $0.5B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) > B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-))$.

The converse relation is true if $B(\max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \min_{k=q-s+1,\ldots,q}(\hat{\pi}_k^-)) < \max_{j=1,\ldots,s}(\hat{\pi}_j^+) - \pi$. Simple arithmetic leads to the statements of b).

□

B Further simulation results

B1 Threshold-based stability selection

See Appendix Figs. 14, 15, 16, 17, 18, 19, 20 and 21.

Below, we depict the results of the rank-based Stability Selection with $q = 8$ (see Appendix Figs. 22, 23).

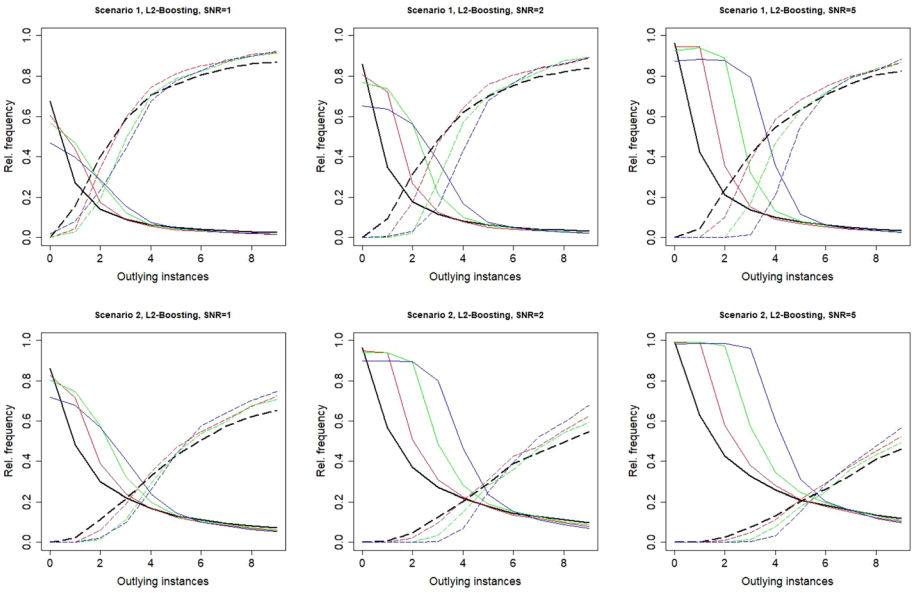

Fig. 14 Results for scenarios 1 and 2 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

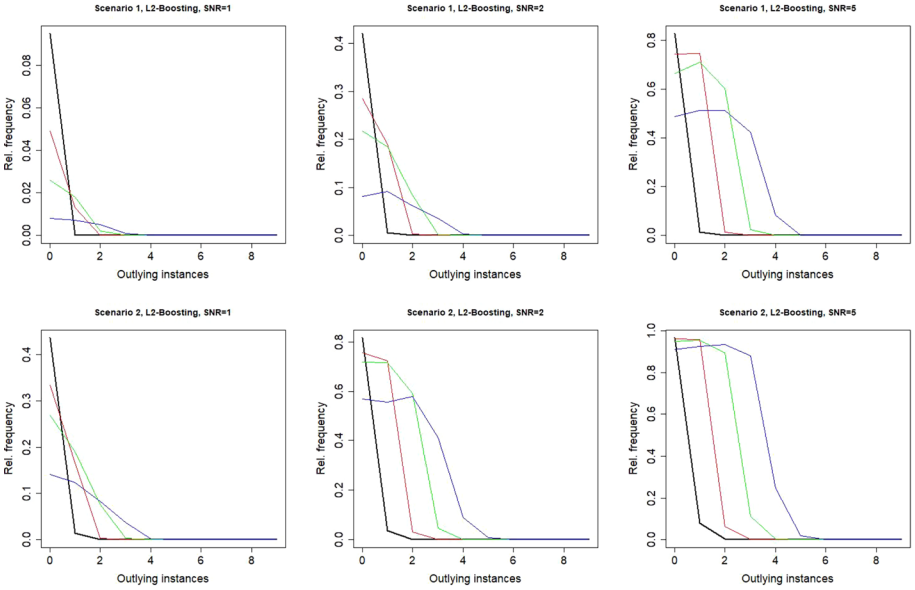

Fig. 15 Relative frequencies of perfect stable models for scenarios 1 and 2 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

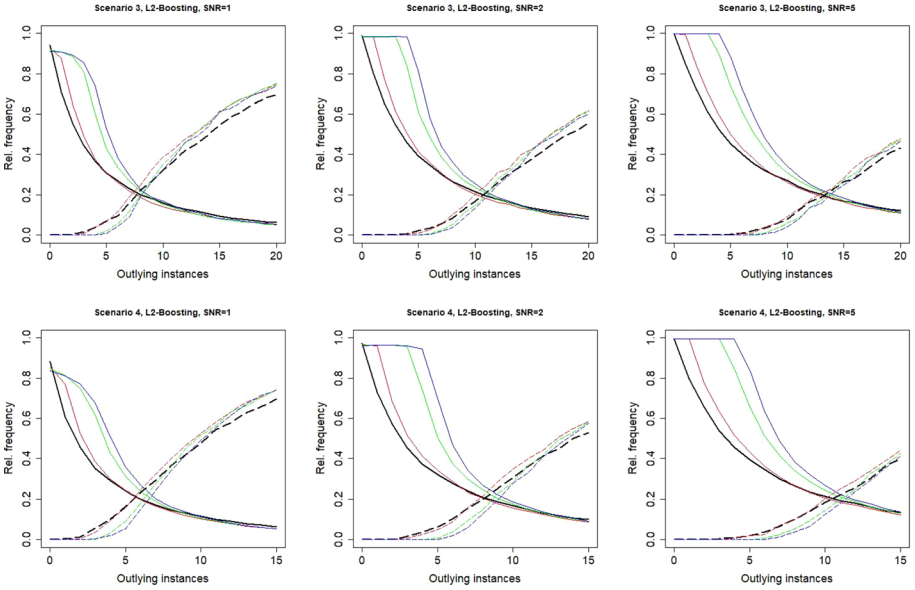

Fig. 16 Results for scenarios 3 and 4 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

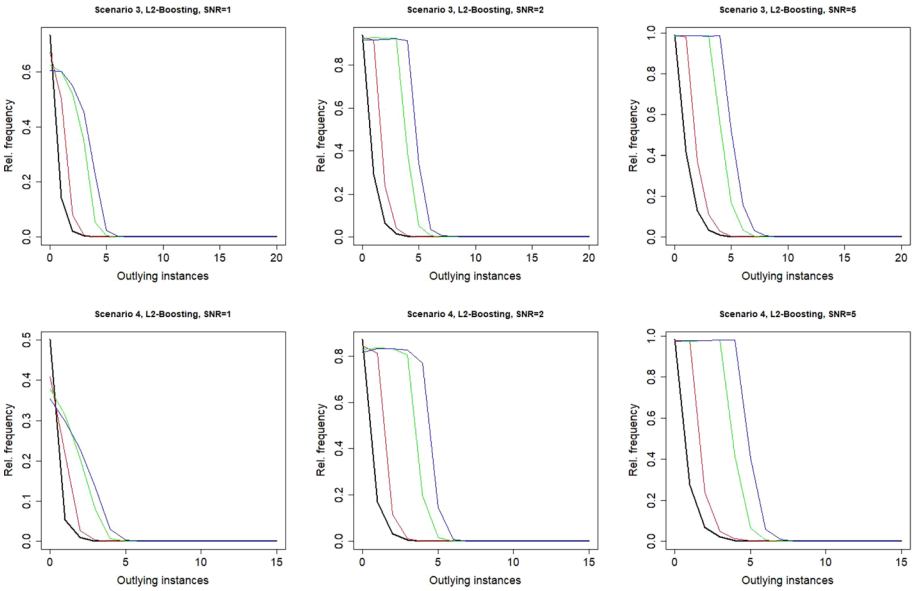

Fig. 17 Results for scenarios 3 and 4 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

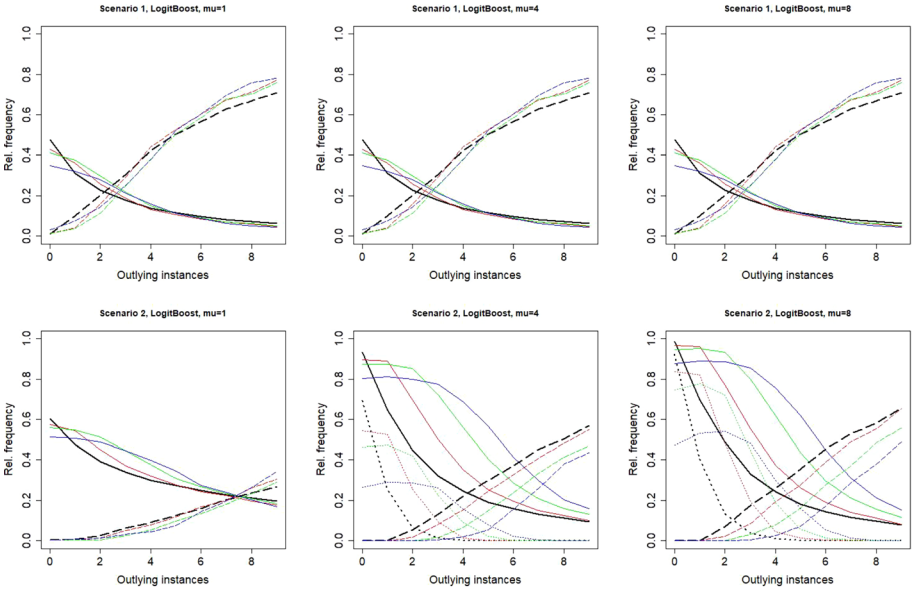

Fig. 18 Results for scenarios 1 and 2 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

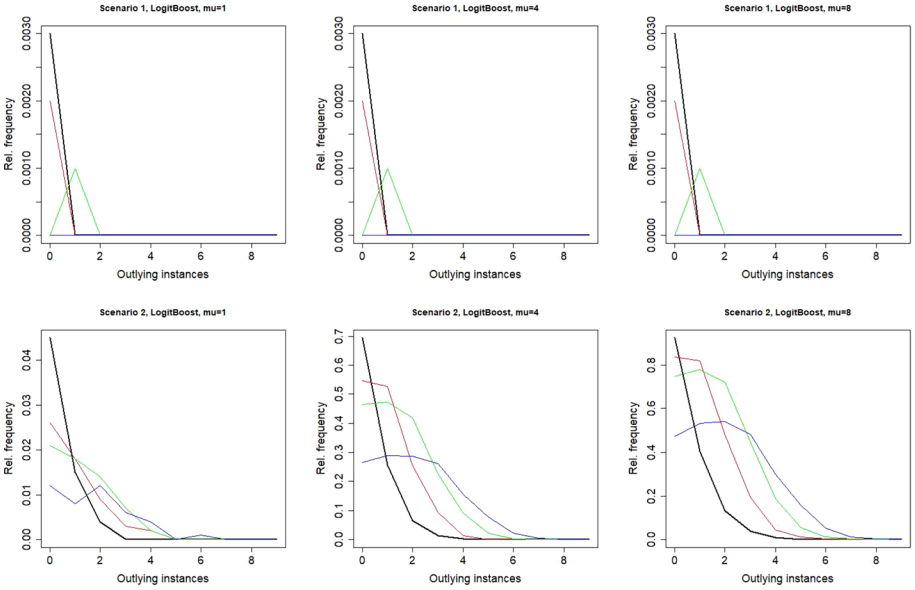

Fig. 19 Results for scenarios 1 and 2 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

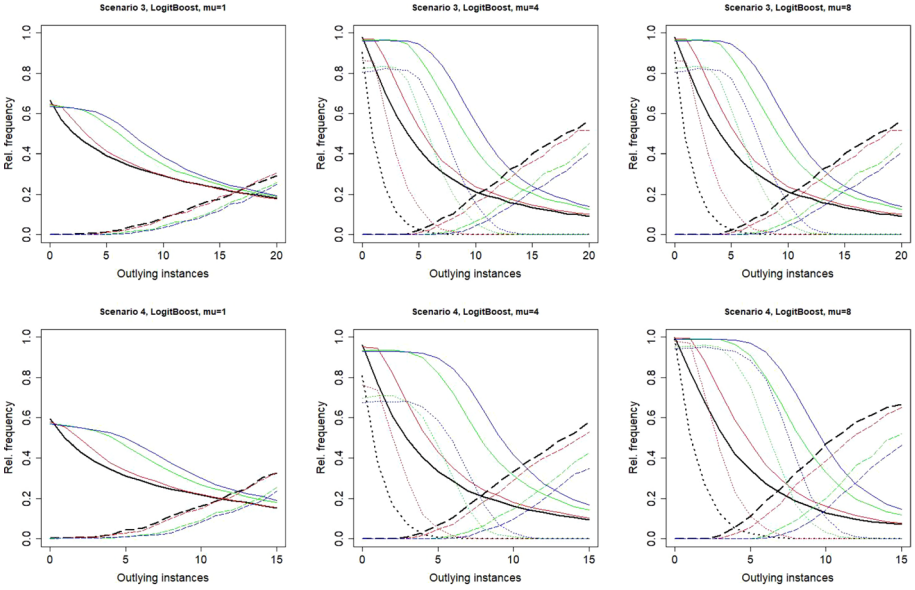

Fig. 20 Results for scenarios 3 and 4 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

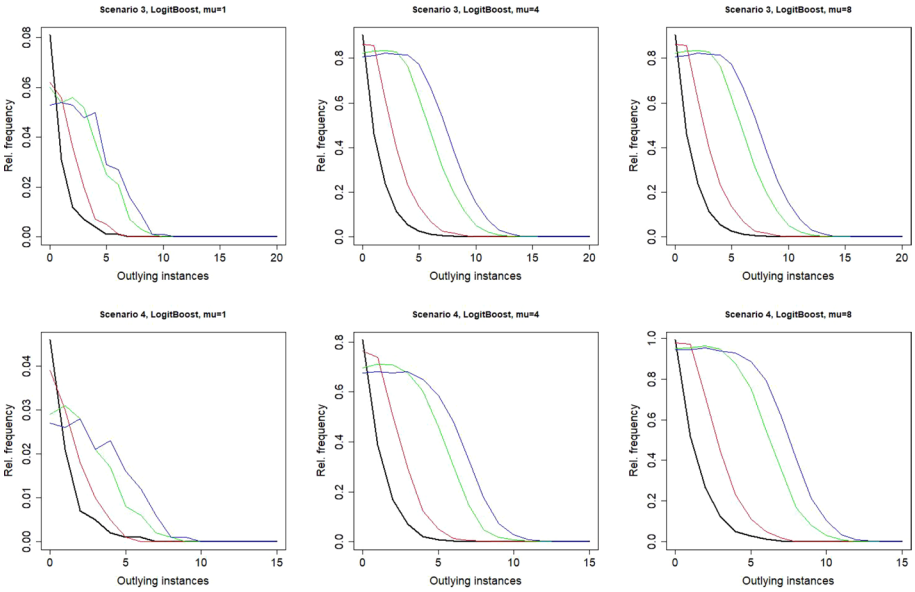

Fig. 21 Results for scenarios 3 and 4 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

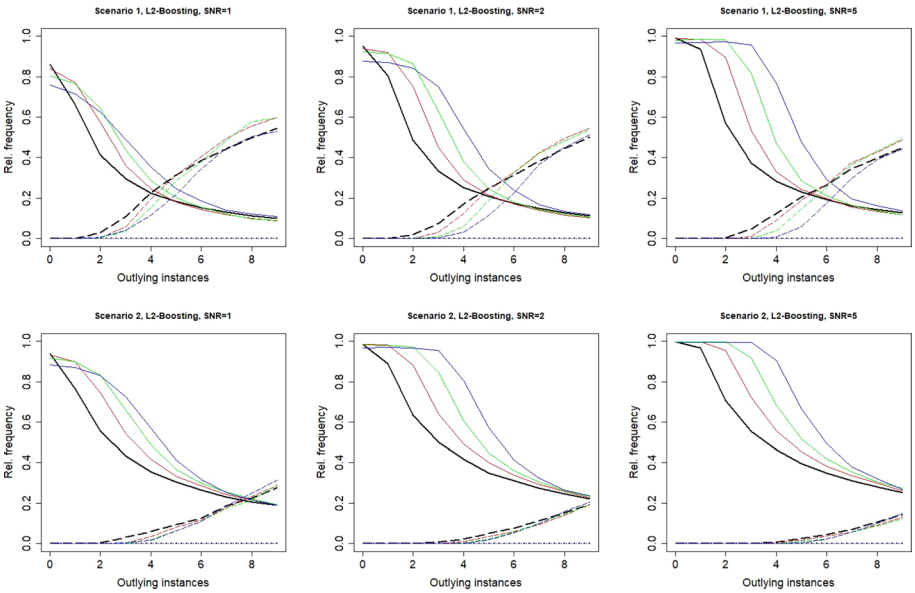

Fig. 22 Results for scenarios 1 and 2 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

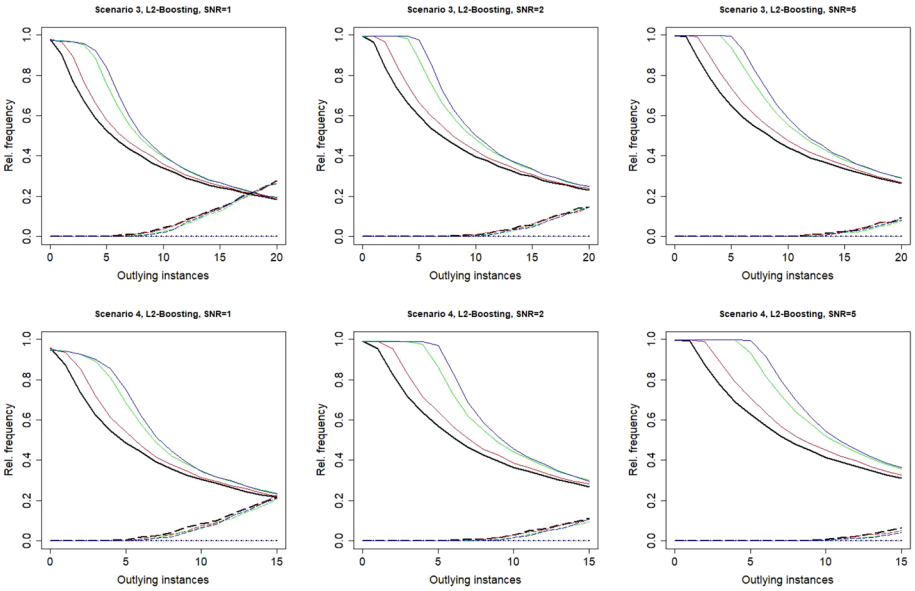

Fig. 23 Results for scenarios 3 and 4 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B2 Less sparse true models

We repeat the experiments for L_2-Boosting in Scenario 3 with less sparse true models, i.e., with $s_0 = 20$ instead of $s_0 = 5$, both for the threshold- and the rank-based Stability Selection.

Fig. 24 reveals that the rank-based and the threshold-based variant indeed can lead to very different results in this situation. While the rank-based variant already shows a decreased performance in contrast to the case $s_0 = 5$, the performance of the threshold-based variant is very poor, most likely due to the fact that due to the higher number of relevant variables, it could be more likely to miss some in each of the models, decreasing the relative selection frequencies for the Stability Selection so that more relevant variables are discarded in the threshold-based variant.

Figure 25 reveals that the relative frequency of perfect models has significantly decreased which is not surprising as it can be assumed to be much more difficult to retrieve all 20 relevant variables in the stable model without selecting any noise variable than finding only 5 relevant variables without selecting any noise variable.

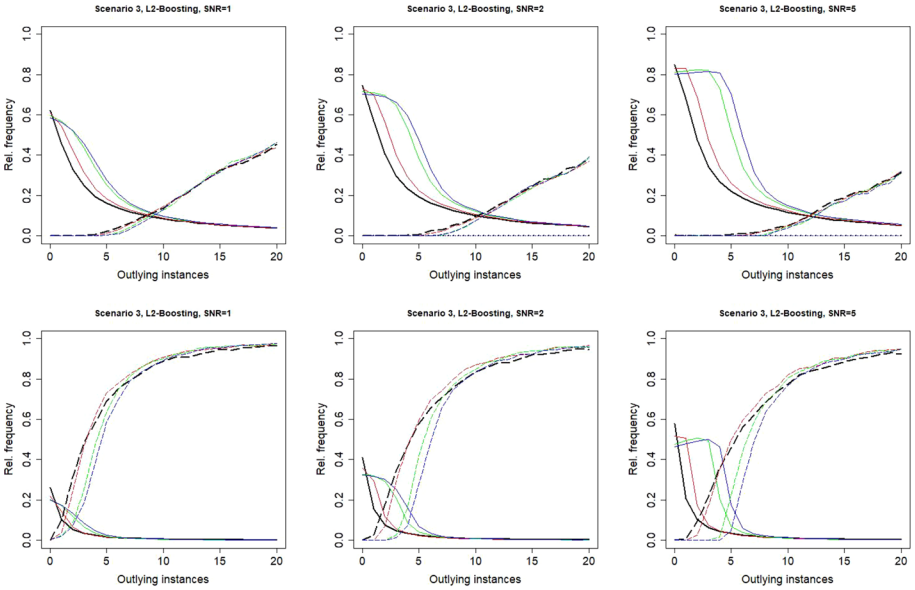

Fig. 24 Results for scenario 3 with L_2-Boosting as model selection algorithm. Upper row: Rank-based Stability Selection; Lower row: Threshold-based Stability Selection. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

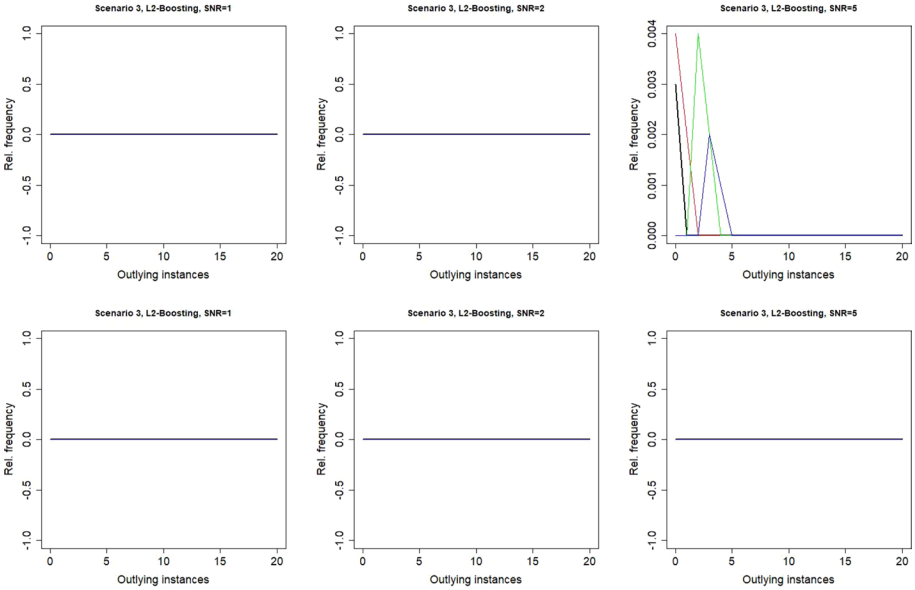

Fig. 25 Results for scenario 3 with L_2-Boosting as model selection algorithm. Upper row: Rank-based Stability Selection; Lower row: Threshold-based Stability Selection. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B3 Large outliers

See Appendix Figs. 26, 27, 28, 29, 30 and 31

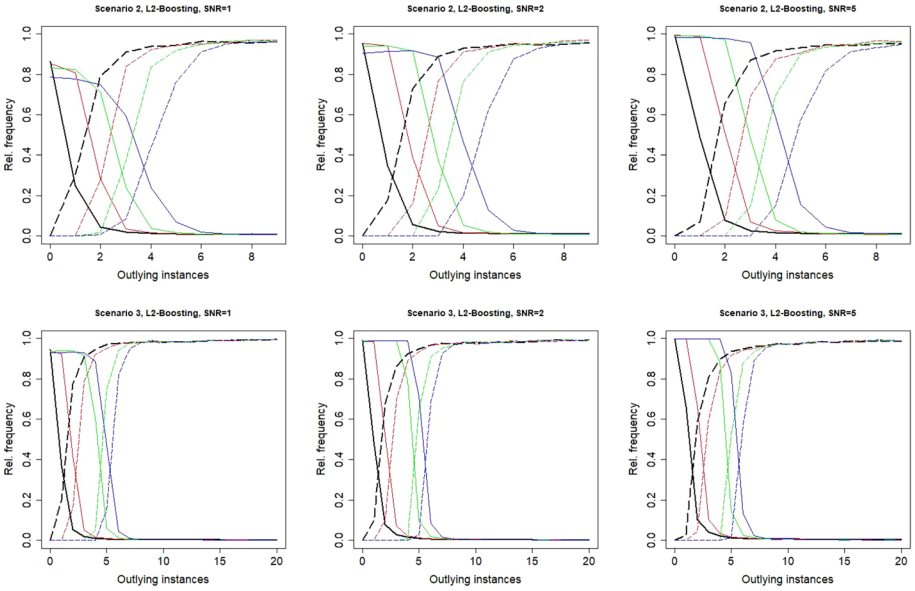

Fig. 26 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

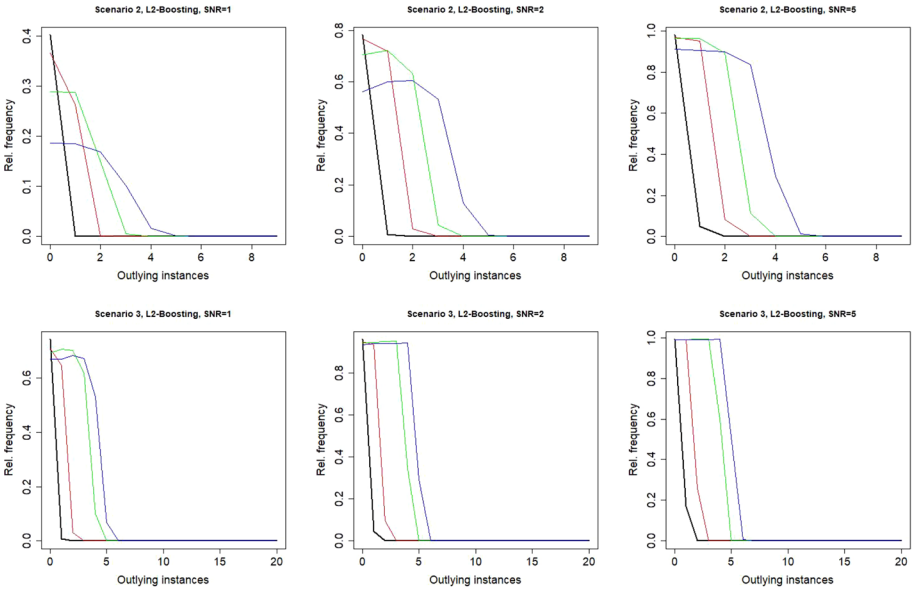

Fig. 27 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

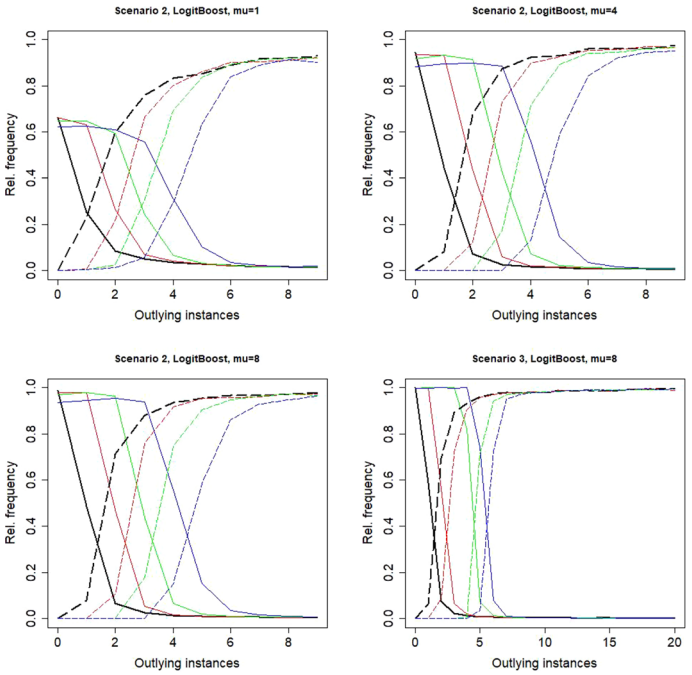

Fig. 28 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

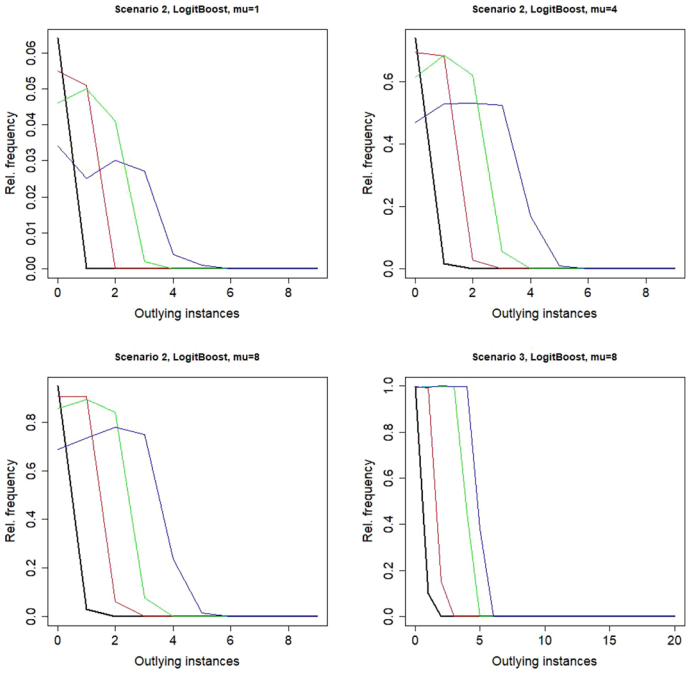

Fig. 29 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

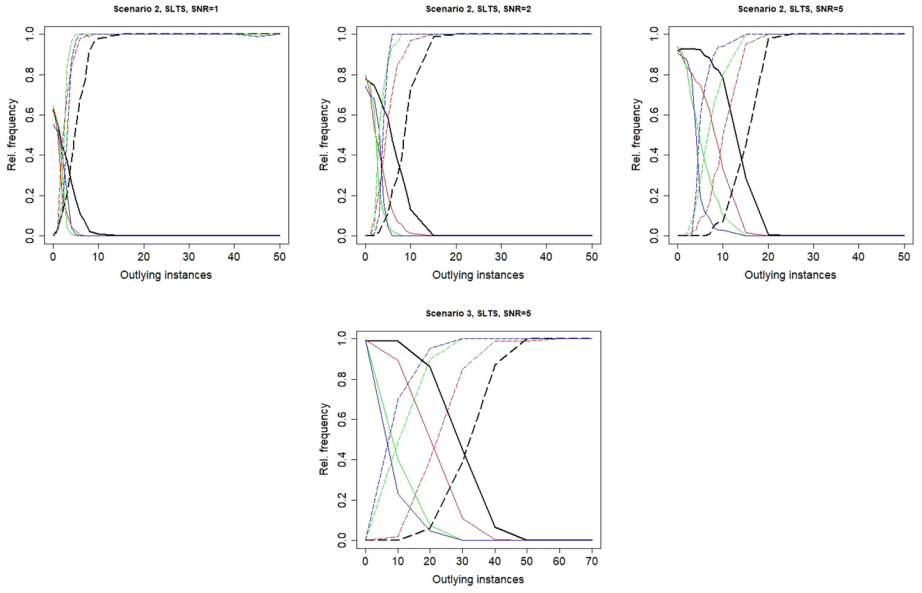

Fig. 30 Results for scenarios 2 and 3 with SLTS as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

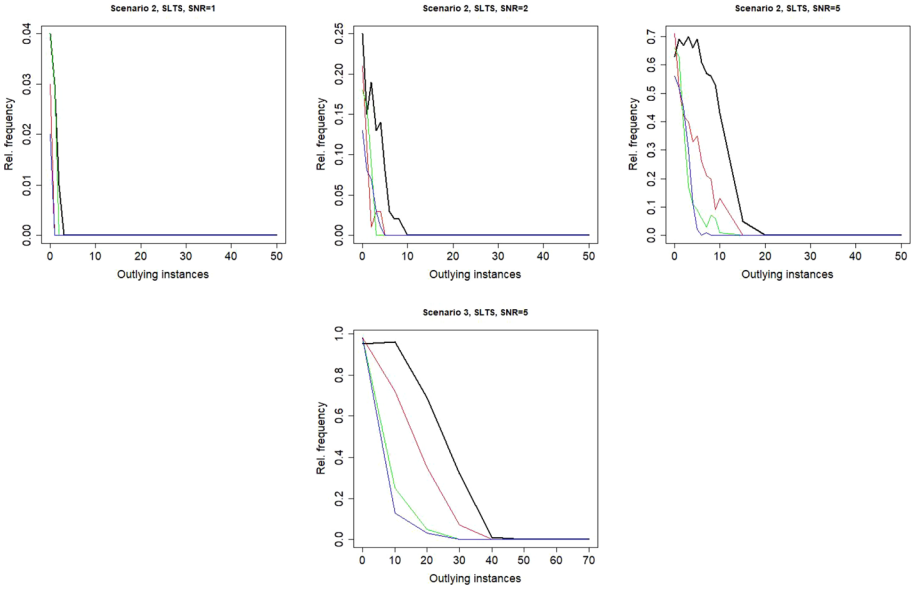

Fig. 31 Results for scenarios 2 and 3 with SLTS as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B4 Out-of-sample loss

See Appendix Figs. 32, 33, 34, 35, 36 and 37.

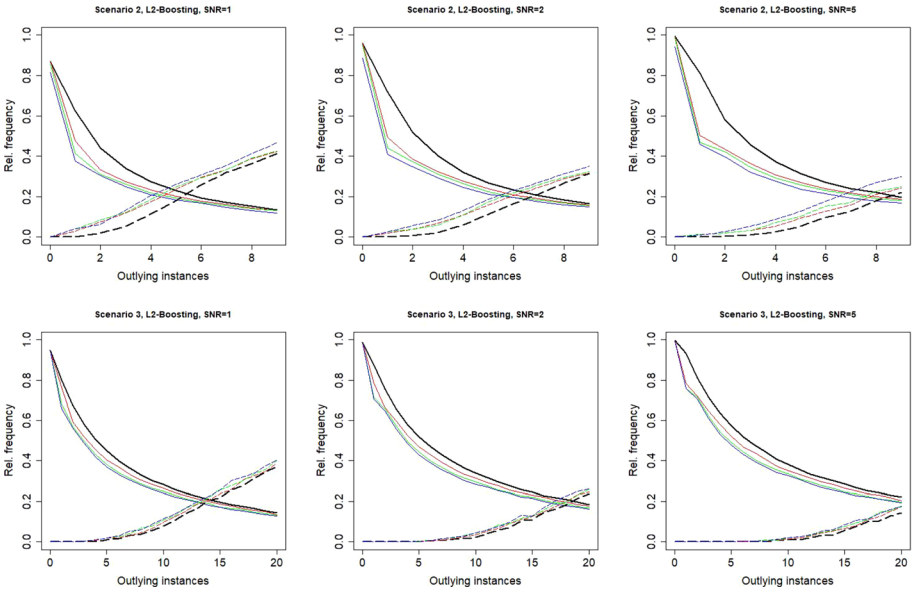

Fig. 32 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

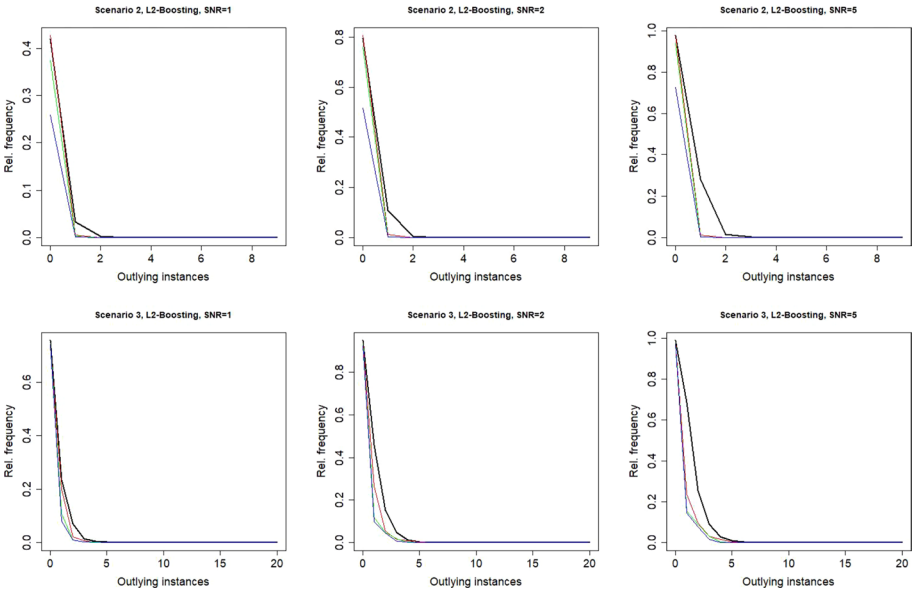

Fig. 33 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

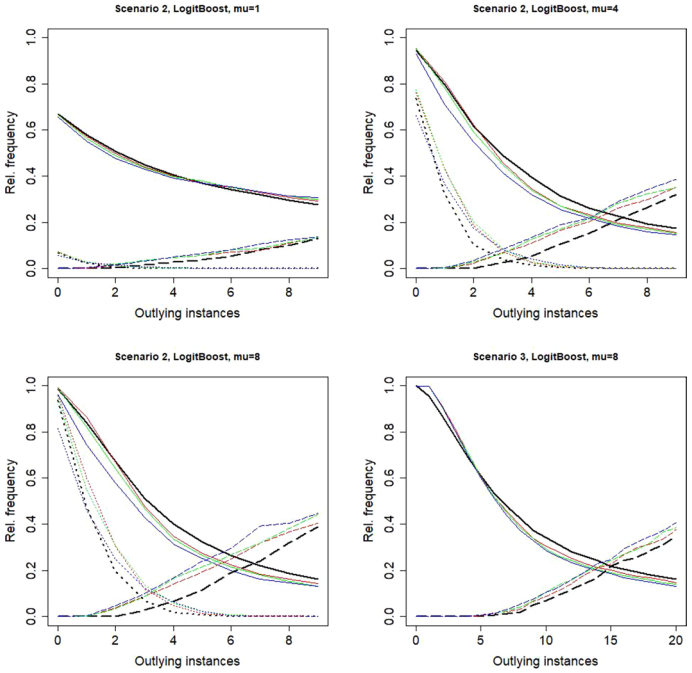

Fig. 34 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

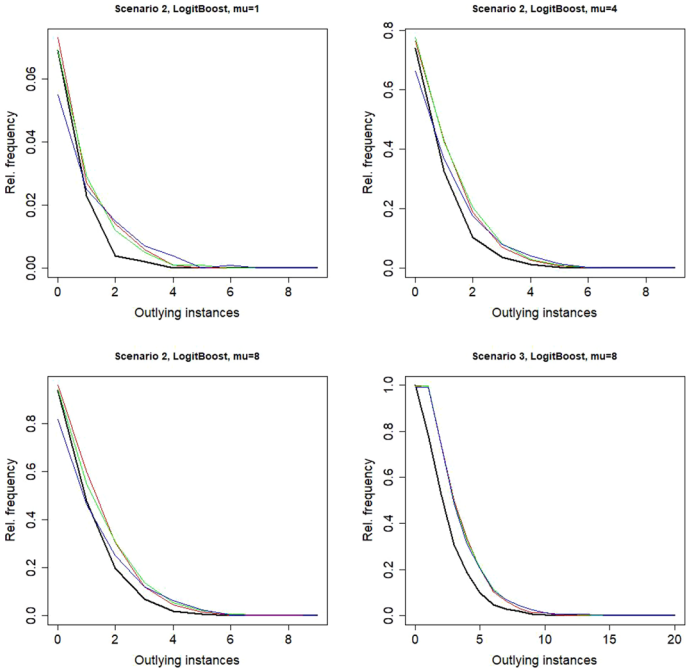

Fig. 35 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

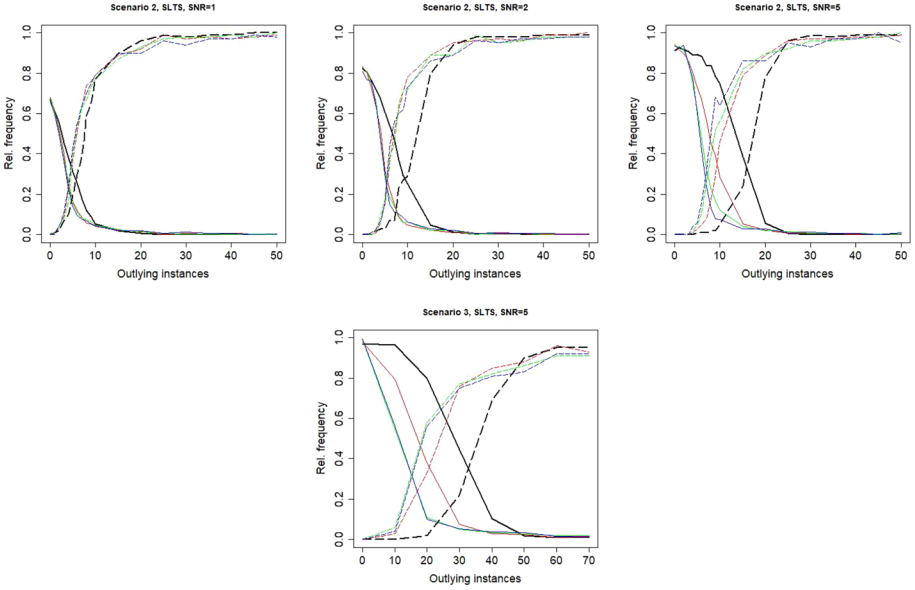

Fig. 36 Results for scenarios 2 and 3 with SLTS as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

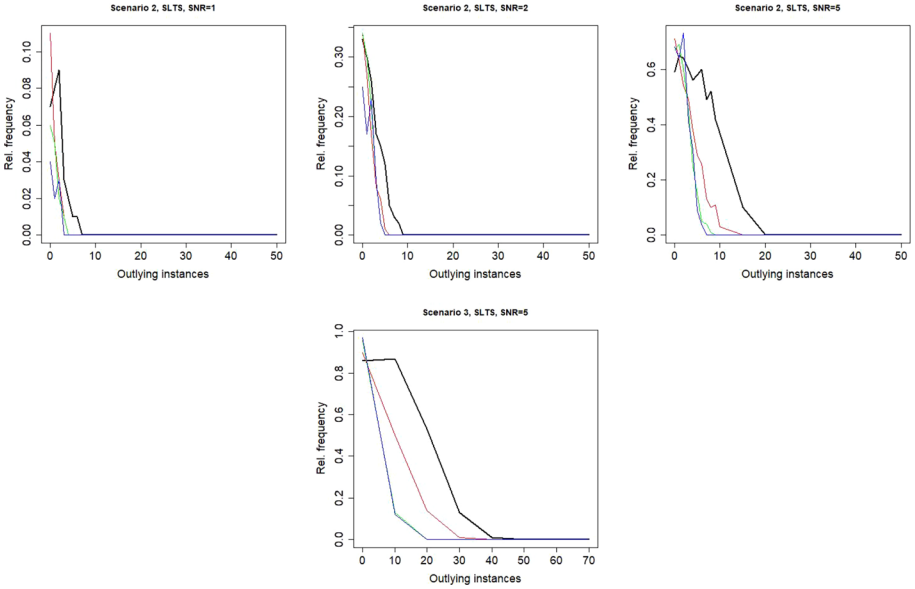

Fig. 37 Results for scenarios 2 and 3 with SLTS as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B5 Out-of-sample loss and large outliers

See Appendix Figs. 38, 39, 40, 41, 42 and 43.

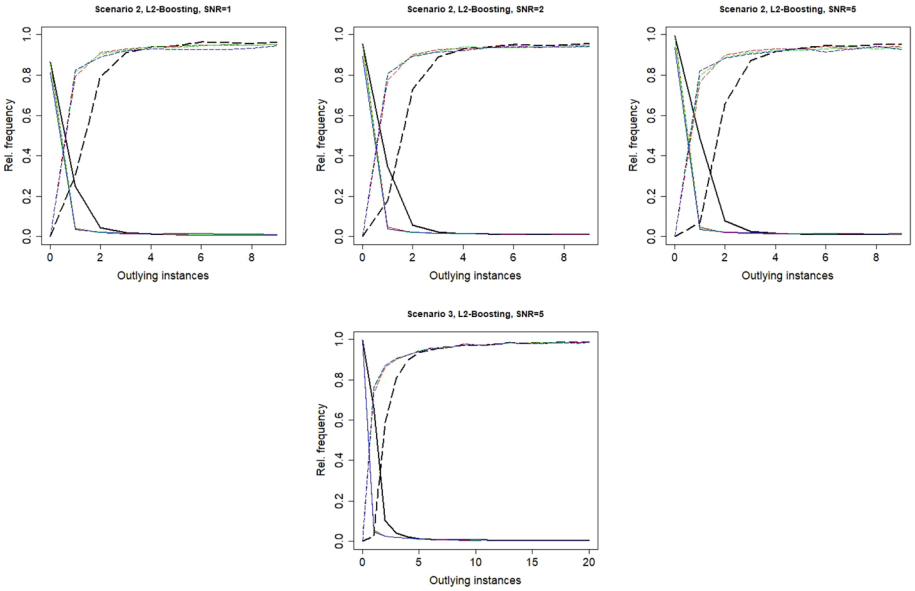

Fig. 38 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

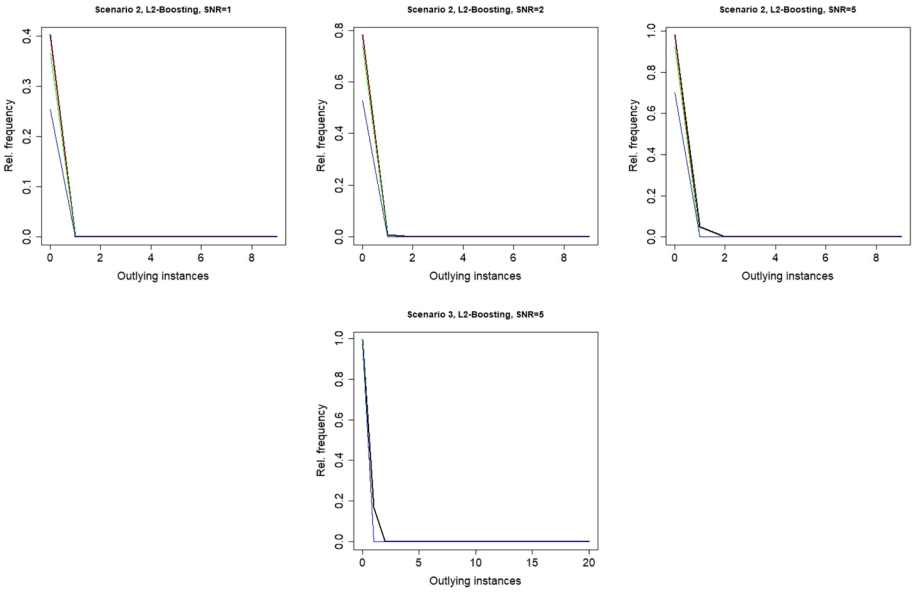

Fig. 39 Results for scenarios 2 and 3 with L_2-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

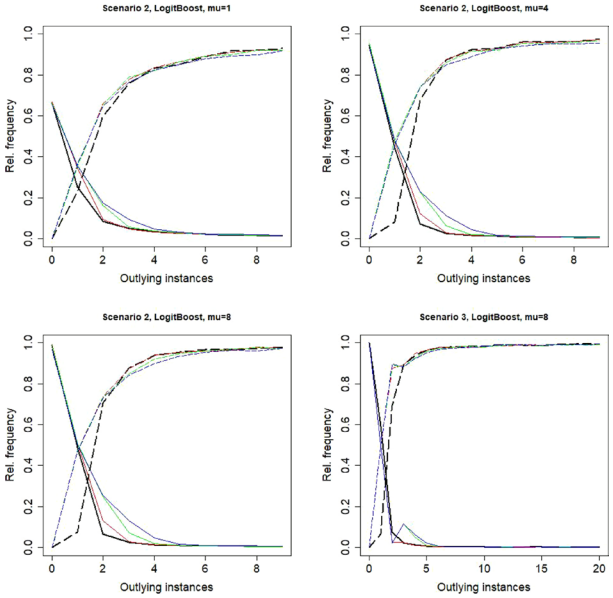

Fig. 40 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

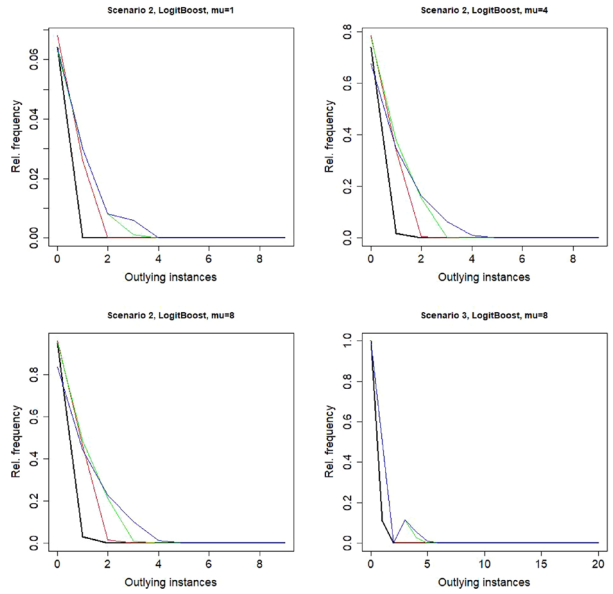

Fig. 41 Results for scenarios 2 and 3 with LogitBoost as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

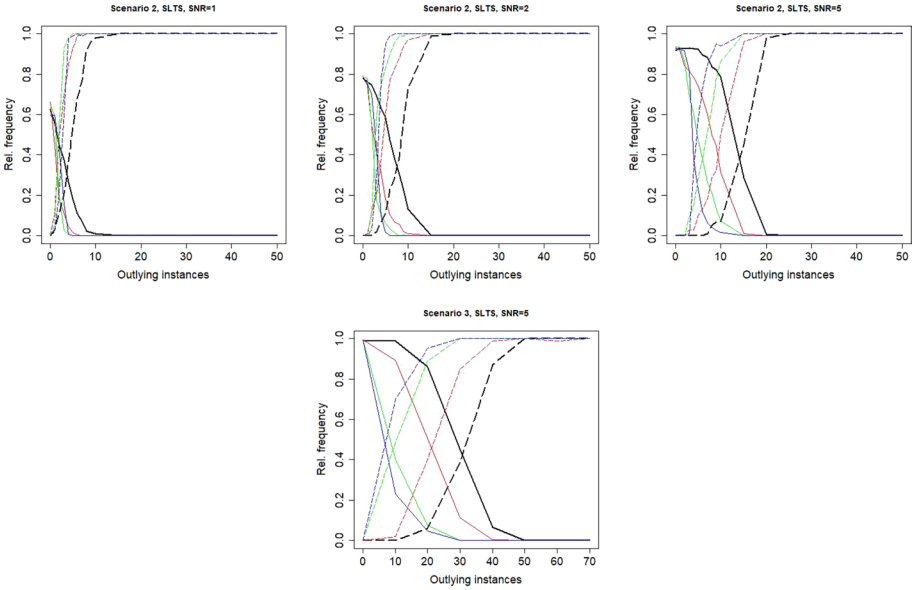

Fig. 42 Results for scenarios 2 and 3 with SLTS as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

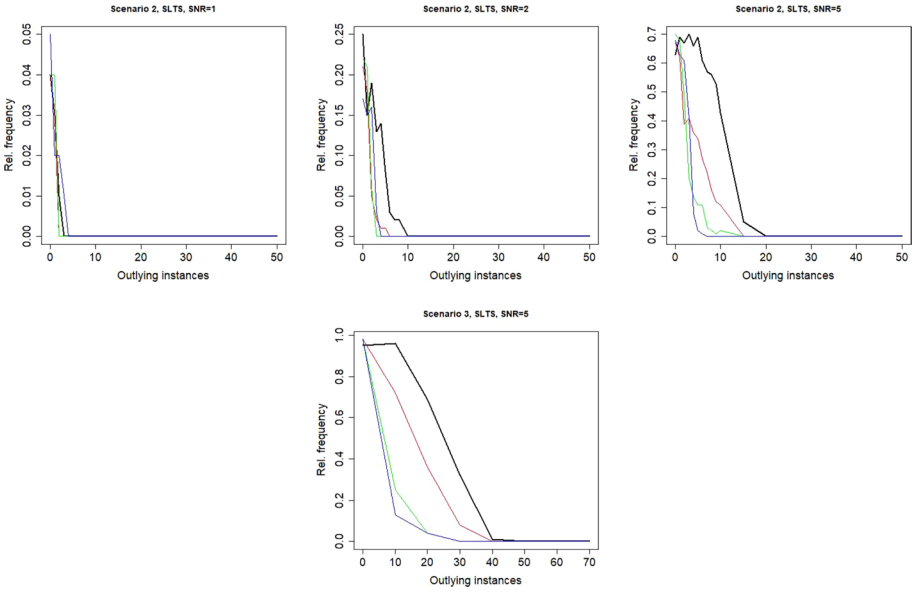

Fig. 43 Results for scenarios 2 and 3 with SLTS as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B6 RRBoost

See Appendix Figs. 44, 45, 46 and 47.

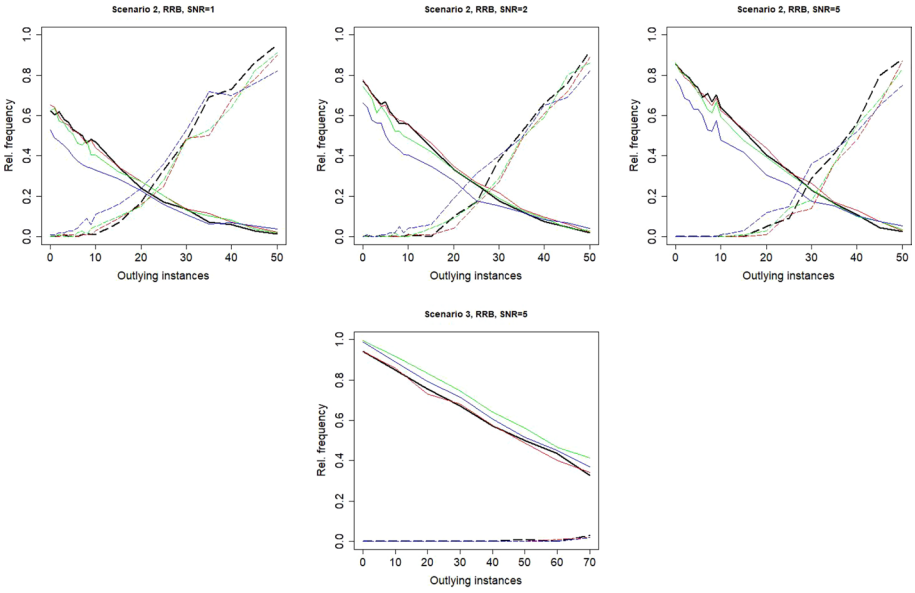

Fig. 44 Results for scenarios 2 and 3 with robust Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

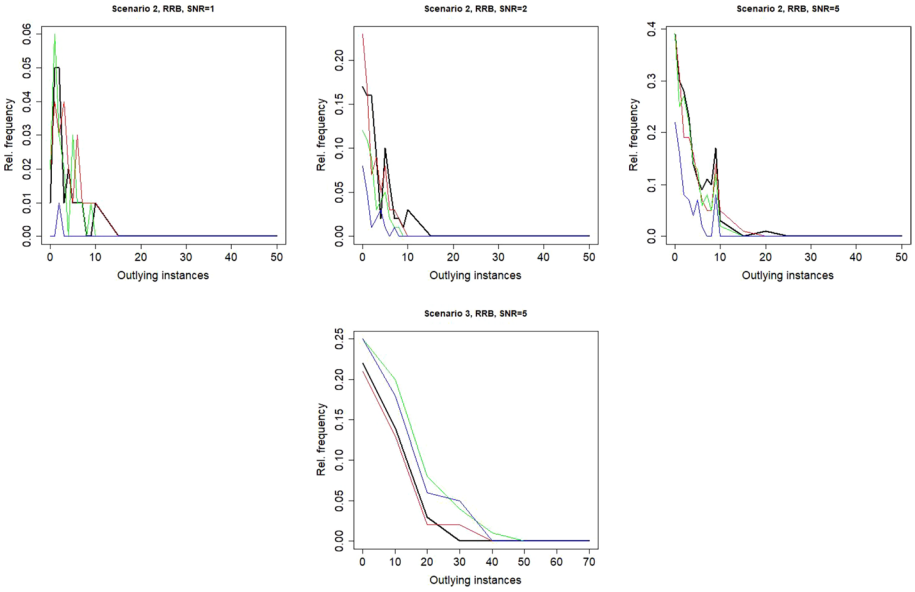

Fig. 45 Results for scenarios 2 and 3 with robust Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

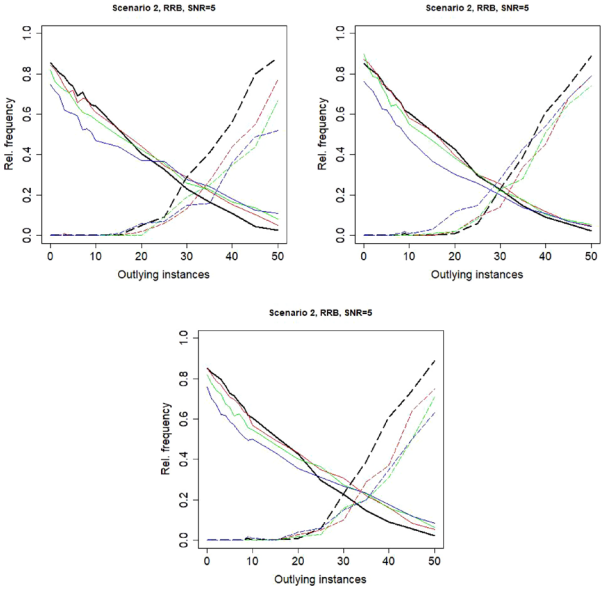

Fig. 46 Results for scenario 2 with robust Boosting as model selection algorithm. Top left: Out-of-sample losses were used; top right: Large outliers were generated; bottom: Out-of-sample losses in combination with large outliers. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

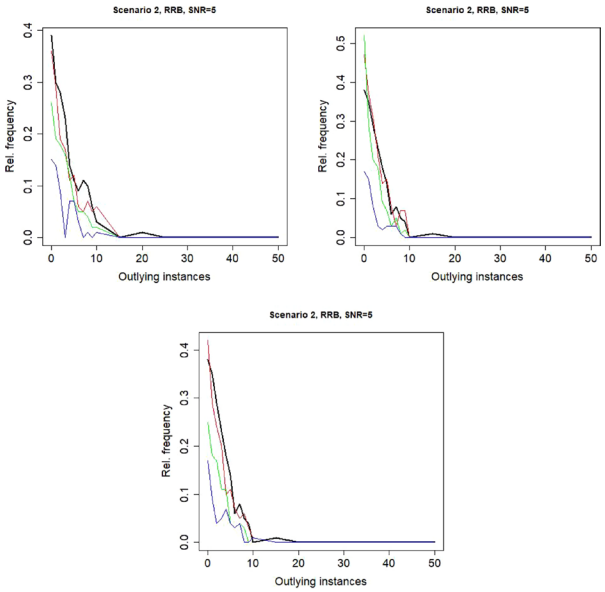

Fig. 47 Results for scenario 2 with robust Boosting as model selection algorithm. Top left: Out-of-sample losses were used; top right: Large outliers were generated; bottom: Out-of-sample losses in combination with large outliers. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

B7 AUC-Boosting

See Appendix Figs. 48, 49, 50 and 51.

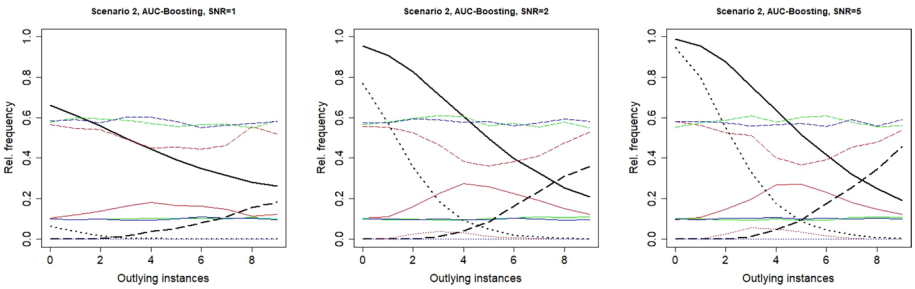

Fig. 48 Results for scenario 2 with AUC-Boosting as model selection algorithm. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

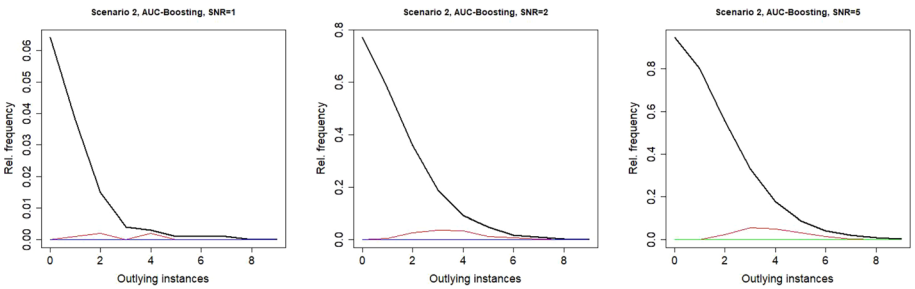

Fig. 49 Results for scenario 2 with AUC-Boosting as model selection algorithm. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

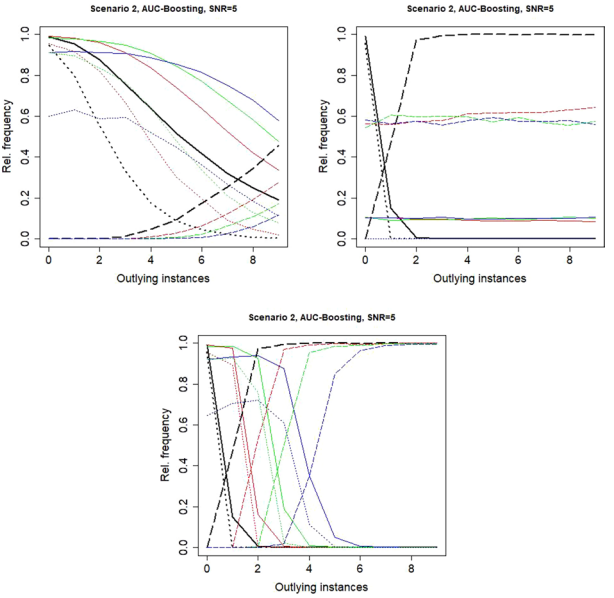

Fig. 50 Results for scenario 2 with AUC-Boosting as model selection algorithm. Top left: Out-of-sample losses were used; top right: Large outliers were generated; bottom: Out-of-sample losses in combination with large outliers. Solid lines represent the TPR, dashed lines the relative frequencies of a breakdown. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

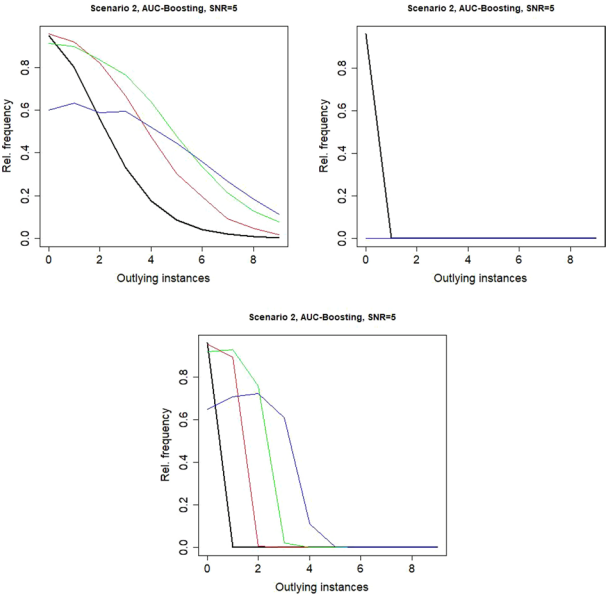

Fig. 51 Results for scenario 2 with AUC-Boosting as model selection algorithm. Top left: Out-of-sample losses were used; top right: Large outliers were generated; bottom: Out-of-sample losses in combination with large outliers. The black lines correspond to the non-trimmed Stability Selection and the red, green and blue lines to the first, second and third configuration of TrimStabSel, as specified in Table 1, respectively (Color figure online)

Acknowledgements I thank all the anonymous referees for their valuable comments that helped to significantly improve the quality of the paper.

Author Contributions Not applicable.

Funding Open Access funding enabled and organized by Projekt DEAL. Not applicable.

Availability of data and material Not applicable.

Code Availability The R-Code used for the simulations is available upon request.

Declarations

Conflict of interest Not applicable.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Agostinelli, C., Leung, A., Yohai, V. J., & Zamar, R. H. (2015). Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. *Test, 24*(3), 441–461.

Alelyani, S., Tang, J., & Liu, H. (2013). Feature selection for clustering: a review. *Data Clustering: Algorithms and Applications, 29*(110–121), 144.

Alfons, A. (2016). robustHD: Robust Methods for High-Dimensional Data. R package version 0.5.1. https://CRAN.R-project.org/package=robustHD

Alfons, A., Croux, C., & Gelper, S. (2013). Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics, 7*(1), 226–248.

Alqallaf, F., Van Aelst, S., Yohai, V. J., & Zamar, R. H. (2009). Propagation of outliers in multivariate data. *The Annals of Statistics, 37*(1), 311–331.

Arslan, O. (2012). Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression. *Computational Statistics & Data Analysis, 56*(6), 1952–1965.

Banerjee, O., Ghaoui, L. E., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research, 9*, 485–516.

Becker, C., & Gather, U. (1999). The masking breakdown point of multivariate outlier identification rules. *Journal of the American Statistical Association, 94*(447), 947–955.

Berrendero, J. R. (2007). The bagged median and the bragged mean. *The American Statistician, 61*(4), 325–330.

Bottmer, L., Croux, C., & Wilms, I. (2022). Sparse regression for large data sets with outliers. *European Journal of Operational Research, 297*(2), 782–794.

Bühlmann, P. (2012). Bagging, boosting and ensemble methods. In *Handbook of computational statistics* (pp. 985–1022). Springer.

Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: Methods, theory and applications*. Springer.

Bühlmann, P., & Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science, 22*(4), 477–505.
Bühlmann, P., & Yu, B. (2003). Boosting with the L_2 loss: Regression and Classification. *Journal of the American Statistical Association, 98*(462), 324–339.
Camponovo, L., Scaillet, O., & Trojani, F. (2012). Robust subsampling. *Journal of Econometrics, 167*(1), 197–210.
Chang, L., Roberts, S., & Welsh, A. (2018). Robust lasso regression using Tukey's biweight criterion. *Technometrics, 60*(1), 36–47.
Chen, X., Wang, Z.J., & McKeown, M.J. (2010b). Asymptotic analysis of the Huberized lasso estimator. In *2010 IEEE international conference on acoustics speech and signal processing (ICASSP)* (pp. 1898–1901). IEEE.
Chen, X., Wang, Z. J., & McKeown, M. J. (2010). Asymptotic analysis of robust lassos in the presence of noise with large variance. *IEEE Transactions on Information Theory, 56*(10), 5131–5149.
Croux, C., & Öllerer, V. (2016). Robust and sparse estimation of the inverse covariance matrix using rank correlation measures. In *Recent advances in robust statistics: Theory and applications*, (pp. 35–55). Springer.
Croux, C., Joossens, K., & Lemmens, A. (2007). Trimmed bagging. *Computational statistics & data analysis, 52*(1), 362–368.
Davies, P. (1993). Aspects of robust linear regression. *The Annals of Statistics, 21*(4), 1843–1899.
Davies, P. L., & Gather, U. (2005). Breakdown and groups. *The Annals of Statistics, 33*(3), 977–1035.
Donoho, D. L., & Huber, P. J. (1983). The notion of breakdown point. *A Festschrift for Erich L. Lehmann*, 157–184.
Donoho, D., & Stodden, V. (2006). Breakdown point of model selection when the number of variables exceeds the number of observations. In *The 2006 IEEE international joint conference on neural network proceedings* (pp. 1916–1921). IEEE.
Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics, 32*(2), 407–499.
Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70*(5), 849–911.
Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., & Verdonck, T. (2020). Cellwise robust M regression. *Computational Statistics & Data Analysis, 147*, 106944.
Filzmoser, P., Maronna, R., & Werner, M. (2008). Outlier identification in high dimensions. *Computational Statistics & Data Analysis, 52*(3), 1694–1711.
Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning* (Vol. 1). Springer.
Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics, 9*(3), 432–441.
García-Escudero, L. A., Rivera-García, D., Mayo-Iscar, A., & Ortega, J. (2021). Cluster analysis with cellwise trimming and applications for the robust clustering of curves. *Information Sciences, 573*, 100–124.
Gather, U., & Hilker, T. (1997). A note on Tyler's modification of the MAD for the Stahel–Donoho estimator. *Annals of statistics, 25*(5), 2024–2026.
Genton, M. G. (1998). Spatial breakdown point of variogram estimators. *Mathematical Geology, 30*(7), 853–871.
Grandvalet, Y. (2000). Bagging down-weights leverage points. In *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. Neural computing: New challenges and perspectives for the new millennium* (Vol. 4, pp. 505–510). IEEE.
Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (2011). *Robust statistics: The approach based on influence functions* (Vol. 114). Wiley.
Hampel, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics, 42*(6), 1887–1896.
Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association, 69*(346), 383–393.
Hofner, B., & Hothorn, T. (2017). stabs: Stability selection with error control. R package version 0.6-3. https://CRAN.R-project.org/package=stabs
Hofner, B., Boccuto, L., & Göker, M. (2015). Controlling false discoveries in high-dimensional situations: Boosting with stability selection. *BMC Bioinformatics, 16*(1), 1–17.
Hofner, B., Mayr, A., Robinzonov, N., & Schmid, M. (2014). Model-based boosting in R: A hands-on tutorial using the R package mboost. *Computational Statistics, 29*(1–2), 3–35.
Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., & Hofner, B. (2017). mboost: Model-based boosting. R package version 2.8-1. https://CRAN.R-project.org/package=mboost
Hothorn, T., & Bühlmann, P. (2006). Model-based boosting in high dimensions. *Bioinformatics, 22*(22), 2828–2829.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., & Hofner, B. (2010). Model-based boosting 2.0. *Journal of Machine Learning Research, 11*, 2109–2113.
Huber, P. J., & Ronchetti, E. (2009). *Robust statistics*. Wiley.
Hubert, M. (1997). The breakdown value of the L_1 estimator in contingency tables. *Statistics & Probability Letters, 33*(4), 419–425.
Hubert, M., Rousseeuw, P. J., & Van Aelst, S. (2008). High-breakdown robust multivariate methods. *Statistical Science, 23*(1), 92–119.
Ju, X., & Salibián-Barrera, M. (2020). RRBoost: A robust boosting algorithm. R package version 0.1. https://CRAN.R-project.org/package=RRBoost
Ju, X., & Salibián-Barrera, M. (2021). Robust boosting for regression problems. *Computational Statistics & Data Analysis, 153*(1), 107065.
Lai, H., Pan, Y., Liu, C., Lin, L., & Wu, J. (2013). Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers, 62*(6), 1221–1233.
Laporte, L., Flamary, R., Canu, S., Déjean, S., & Mothe, J. (2014). Nonconvex regularizations for feature selection in ranking with sparse SVM. *IEEE Transactions on Neural Networks and Learning Systems, 25*(6), 1118–1130.
Leung, A., Yohai, V., & Zamar, R. (2017). Multivariate location and scatter matrix estimation under cellwise and casewise contamination. *Computational Statistics & Data Analysis, 111*, 59–76.
Leung, A., Zhang, H., & Zamar, R. (2016). Robust regression estimation and inference in the presence of cellwise and casewise contamination. *Computational Statistics & Data Analysis, 99*, 1–11.
Li, F., Lai, L., & Cui, S. (2020). On the adversarial robustness of feature selection using LASSO. In *2020 IEEE 30th international workshop on machine learning for signal processing (MLSP)* (pp. 1–6). IEEE.
Li, F., Lai, L., & Cui, S. (2021). On the adversarial robustness of LASSO based feature selection. *IEEE Transactions on Signal Processing, 69*, 5555–5567.
Lutz, R. W., Kalisch, M., & Bühlmann, P. (2008). Robustified L_2 boosting. *Computational Statistics & Data Analysis, 52*(7), 3331–3341.
Maronna, R. A., Martin, R. D., Yohai, V. J., & Salibián-Barrera, M. (2019). *Robust statistics: Theory and methods (with R)*. Wiley.
Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis, 52*(1), 374–393.
Meinshausen, N., & Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72*(4), 417–473.
Nogueira, S., & Brown, G. (2016). Measuring the stability of feature selection. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 442–457). Springer.
Nogueira, S., Sechidis, K., & Brown, G. (2017b). On the use of Spearman's rho to measure the stability of feature rankings. In *Iberian conference on pattern recognition and image analysis* (pp. 381–391). Springer.
Nogueira, S., Sechidis, K., & Brown, G. (2017). On the stability of feature selection algorithms. *Journal of Machine Learning Research, 18*(1), 6345–6398.
Öllerer, V., & Croux, C. (2015). Robust high-dimensional precision matrix estimation. In *Modern nonparametric, robust and multivariate methods* (pp. 325–350). Springer.
Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506–519).
Park, M. Y., & Hastie, T. (2007). L_1-Regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 69*(4), 659–677.
Park, H., Yamada, M., Imoto, S., & Miyano, S. (2019). Robust sample-specific stability selection with effective error control. *Journal of Computational Biology, 26*(3), 202–217.
Qian, C., Tran-Dinh, Q., Fu, S., Zou, C., & Liu, Y. (2019). Robust multicategory support matrix machines. *Mathematical Programming, 176*(1–2), 429–463.
Rieder, H. (1994). *Robust asymptotic statistics* (Vol. 1). Springer.
Rieder, H., Kohl, M., & Ruckdeschel, P. (2008). The cost of not knowing the radius. *Statistical Methods & Applications, 17*(1), 13–40.
Rocke, D. M., & Woodruff, D. L. (1996). Identification of outliers in multivariate data. *Journal of the American Statistical Association, 91*(435), 1047–1061.
Rosset, S., & Zhu, J. (2007). Piecewise linear regularized solution paths. *Annals of statistics, 35*(3), 1012–1030.
Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association, 79*(388), 871–880.
Rousseeuw, P. J. (1985). Multivariate estimation with high breakdown point. *Mathematical Statistics and Applications, 8*(37), 283–297.
Rousseeuw, P. J., & Van Den Bossche, W. (2018). Detecting deviating data cells. *Technometrics, 60*(2), 135–145.

Rousseeuw, P. J., & Hubert, M. (2011). Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(1), 73–79.
Salibián-Barrera, M. (2006). Bootstrapping MM-estimators for linear regression with fixed designs. *Statistics & Probability Letters, 76*(12), 1287–1297.
Salibián-Barrera, M., & Van Aelst, S. (2008). Robust model selection using fast and robust bootstrap. *Computational Statistics & Data Analysis, 52*(12), 5121–5135.
Salibián-Barrera, M., Van Aelst, S., & Willems, G. (2006). Principal components analysis based on multivariate MM estimators with fast and robust bootstrap. *Journal of the American Statistical Association, 101*(475), 1198–1211.
Salibián-Barrera, M., Van Aelst, S., & Willems, G. (2008). Fast and robust bootstrap. *Statistical Methods and Applications, 17*(1), 41–71.
Salibián-Barrera, M., & Zamar, R. H. (2002). Bootrapping robust estimates of regression. *The Annals of Statistics, 30*(2), 556–582.
Shieh, A. D., & Hung, Y. S. (2009). Detecting outlier samples in microarray data. *Statistical Applications in Genetics and Molecular Biology, 8*(1), 1–24.
Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of Computational and Graphical Statistics, 22*(2), 231–245.
Thomas, J., Mayr, A., Bischl, B., Schmid, M., Smith, A., & Hofner, B. (2018). Gradient boosting for distributional regression: Faster tuning and improved variable selection via noncyclical updates. *Statistics and Computing, 28*(3), 673–687.
Tian, Y., Shi, Y., Chen, X., & Chen, W. (2011). AUC maximizing support vector machines with feature selection. *Procedia Computer Science, 4*, 1691–1698.
Tibshirani, R. (1994). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, 58*, 267–288.
Uraibi, H. S. (2019). Weighted lasso subsampling for high dimensional regression. *Electronic Journal of Applied Statistical Analysis, 12*(1), 69–84.
Uraibi, H. S., Midi, H., & Rana, S. (2015). Robust stability best subset selection for autocorrelated data based on robust location and dispersion estimator. *Journal of Probability and Statistics, 2015*, 1–8.
Van de Geer, S. A. (2008). High-dimensional generalized linear models and the lasso. *The Annals of Statistics, 36*(2), 614–645.
Van de Geer, S. A. (2016). *Estimation and testing under sparsity*. Springer.
Velasco, H., Laniado, H., Toro, M., Leiva, V., & Lio, Y. (2020). Robust three-step regression based on comedian and its performance in cell-wise and case-wise outliers. *Mathematics, 8*(8), 1259.
Werner, T. (2022a). Loss-guided stability selection. arXiv preprint arXiv:2202.04956.
Werner, T. (2022). Quantitative robustness of instance ranking problems. *Annals of the Institute of Statistical Mathematics, 75*(2), 1–34.
Witten, D. M., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association, 105*(490), 713–726.
Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68*(1), 49–67.
Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American Statistical Association, 57*(298), 348–368.
Zhang, C., Wu, Y., & Zhu, M. (2019). Pruning variable selection ensembles. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 12*(3), 168–184.
Zhang, C.-X., Zhang, J.-S., & Yin, Q.-Y. (2017). A ranking-based strategy to prune variable selection ensembles. *Knowledge-Based Systems, 125*, 13–25.
Zhao, J., Yu, G., & Liu, Y. (2018). Assessing robustness of classification using angular breakdown point. *Annals of statistics, 46*(6B), 3362.
Zhou, J., Sun, J., Liu, Y., Hu, J., & Ye, J. (2013). Patient risk prediction model via top-k stability selection. In *Proceedings of the 2013 SIAM international conference on data mining* (pp. 55–63). SIAM.
Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association, 101*, 1418–1429.