ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Master Thesis - Quantitative Finance

Tail Risk Modeling and the GPD: The P > N Case

Author: J.P.P. de Heer
Student ID: 433999

Supervisor: Prof. Dr. P.H.B.F. Franses
Second Assessor: Prof. Dr. C. Zhou

Abstract

This paper compares different techniques to model the GPD parameters dynamically using covariates, for estimating the tail risk of equity returns in a high-dimensional context. Modeling is done with different variable selection, regularization, and dimension reduction techniques within the generalized additive models for location, scale and shape (GAMLSS) framework. The simulation study shows that it is extremely hard to incorporate covariates for the shape parameter of the GPD as gradient boosting is unable to outperform random selection of variables. However, variable selection by gradient boosting performs quite well for the scale parameter, even with high collinearity as well as high dimensionality. The simulation shows that applying stability selection slightly improves the variable selection performance of gradient boosting when there is high collinearity. This result is indifferent to the number of observations. The empirical analysis shows that the models using principal component regression (PCR) overfit heavily, while the gradient boosting and stability selection models do not outperform the benchmark both in-sample and out-of-sample on Value-at-Risk (VaR) estimation. Variable selection of the interaction terms of the informative variable proved to be very hard in simulation, and the interaction terms of the covariates add little to no value to VaR estimation of the S&P 500.

Keywords: EVT, VaR, GAMLSS, GPD, PCR, gradient boosting, stability selection.

Contents

Abbreviations

AQL average quantile loss. 31–38

BM block maxima. 6

ES expected shortfall. 4
EVT extreme value theory. 4, 6

GAIC generalized Akaike information criterion. 6, 7, 11, 17, 30, 34, 35, 38, 39, 46
GAMLSS generalized additive models for location, scale and shape. 1, 4–11, 17, 30, 31, 39, 40
GD global deviance. 10, 12
GPD generalized Pareto distribution. 1, 4–15, 17, 28, 30, 31, 33, 34, 39, 40, 46

IRLS iteratively reweighted least squares. 9

LASSO Least Absolute Shrinkage and Selection Operator. 4, 6

OLS ordinary least squares. 10

PC principal component. 5, 7, 10, 12, 30, 45, 46
PCA principal component analysis. 7
PCR principal component regression. 1, 10, 11, 30–33, 36, 39, 40
POT Peak-Over-Threshold. 4–6

SVD singular value decomposition. 5

TPR true positive rate. 5, 17, 19–26, 37, 40

VaR Value-at-Risk. 1, 2, 4, 30–40

WLS weighted least squares. 10

1 Introduction

Mid-March 2020, the S&P 500 recorded the two largest daily losses of this century and the second and third biggest daily losses since the introduction of the index in 1957, due to the global COVID-19 pandemic. Such extreme losses stress the importance of measuring and managing market risk: the risk of loss of a financial position as a consequence of changes in the value of the underlying financial instruments, such as stocks and bonds. To assess market risk, risk measures such as VaR and expected shortfall (ES) are used. Both statistically model the (extreme) losses, and belong to the branch of statistics called the Extreme value theory (EVT), which aims to model extreme events of a variable of interest, such as large financial asset losses.

The focus on large losses is statistically equivalent to the tail of the assumed underlying distribution of the returns. The Peak-Over-Threshold (POT) method models the exceedances above some high threshold u, which approximately follow the generalized Pareto distribution (GPD; Pickands, 1975). Because the exceedances represent observations from the tail, the GPD is the ideal distribution to model extreme financial asset returns. As acknowledged first by Mandelbrot (1963), one of the stylized facts of financial asset returns is the heavy tails. Other stylized facts of financial asset returns include volatility clustering and gain/loss asymmetry (Cont, 2001). To account for the non-normality characteristics of financial asset returns, the GPD can be made dynamic by incorporating covariates in the estimation of the parameters. Chavez-Demoulin and Davison (2005) are among the first to incorporate covariates when estimating the parameters of the distribution of extremes. Later, Chavez-Demoulin, Embrechts, and Hofert (2016) modeled loss data by letting the parameters of the GPD depend on covariates. The GAMLSS framework allows all the parameters of a distribution of a response variable to depend on a set of explanatory variables, see Rigby and Stasinopoulos (2005). GAMLSS allows for modeling almost all distributions, fitting parametric, semi-parametric, and non-parametric models, making this framework extremely flexible. Moreover, different financial factors (e.g., Fama-French factors), market indices, government bonds, interest rates, forex rates, and commodities ensure that a large number of covariates can be used when modeling the tail risk of financial asset returns.

The natural consequence of high dimensional data when combined with models with a high degree of flexibility, however, is the variable selection problem. This issue has been discussed widely in the literature (Akaike, 1973; Schwarz, 1978; Zou and Hastie, 2005; Tibshirani, 1996; Mayr, Fenske, et al., 2012). Albeit a lot of research is done regarding variable selection, literature about this in an EVT setting is scarce. Hoxha (2021) used gradient boosting for variable selection when estimating tail risk with covariates in the parameters of the GPD, but the high dimensional case remains practically uninvestigated. This paper investigates the tail risk of financial asset returns in a high-dimensional setting, with more covariates than observations. The interaction terms of the covariates are added to ensure a large number of (possible) covariates. We dynamically (i.e. dependent on covariates) model the parameters of the GPD to asses market risk using different methods, and compare these methods. Estimation is done within the GAMLSS framework.

The GAMLSS framework allows for model selection by explicit regularization using either just the negative log-likelihood or information criteria. Other explicit regularization methods that can be used within the GAMLSS framework are Least Absolute Shrinkage and Selection Operator (LASSO; Tibshirani, 1996), Ridge (Hoerl and Kennard, 1970), and Elastic Net regularization (Zou and Hastie, 2005). Explicit regularization is done by adding a penalty to the loss function, depending on the norm (e.g., ℓ^1-norm for LASSO) of the coefficients. Implicit regularization includes boosting techniques for estimation, regularization, and variable selection, such as gradient boosting (Friedman, 2001), adapted to GAMLSS by Mayr, Fenske, et al. (2012). Hoxha (2021) found this technique to work well when used to estimate the tail risk of S&P 500 returns. Still, in the high dimensional setting gradient boosting is prone to unstable results (Meinshausen and Bühlmann, 2010; Mayr, Hofner, and Schmid, 2012b). To ensure a stable set of explanatory variables is selected when using this technique, Hofner, Boccuto, and Göker (2015) combined gradient boosting with stability selection, where the boosted GAMLSS model is repeatedly fitted to a changing subset of the original data and the most stable variables are selected. They find this to work well in high-dimensional settings with more predictors than observations.

If we consider the case with more covariates than observations a dimension reduction problem rather

than a variable selection problem, one can also perform a priori dimension reduction. A basic method to deal with a large number of intertwined explanatory variables is singular value decomposition (SVD), which creates principal components (PCs) as linear combinations of the covariates. Recently, Stasinopoulos, Rigby, Georgikopoulos, et al. (2021) adapted this to the GAMLSS framework.

The simulation study of this paper shows that gradient boosting is a well-performing variable selection technique for the scale parameter of the GPD. For increasing collinearity, stability selection offers a slight improvement in terms of the true positive rates (TPRs). However, this is only true for the scale parameter as variable selection by gradient boosting did not outperform random selection for the covariates of the shape parameter. Also, informative interaction terms were very hard to identify for both gradient boosting and stability selection in the simulation study. The empirical analysis confirmed these terms added little to no value to the risk measures estimation.

This thesis is structured as follows. It starts with a review of the relevant current literature in Section 2, followed by the methodology in Section 3. The methodology starts with an overview of the peak-over-threshold (POT) approach and the generalized Pareto distribution (GPD). It then introduces the generalized additive models for location, scale, and shape and subsequently explains the different variable selection techniques. Then a simulation study and subsequently a sensitivity analysis are performed to select the best hyperparameters in Section 4. In Section 5 an empirical analysis will be conducted on the S&P 500 loss data where we fit and analyze the models selected in the previous section. Section 6 discusses the results and gives directions for further research.

2 Literature review

As described in the introduction, classical extreme value theory (EVT) focuses on the tail of the distribution of a variable of interest, modeling the probabilities of extreme events. The two best-known methods are the block maxima (BM) and peak-over-threshold (POT) methods. The first, classical EVT method, divides the observation period into equal-sized blocks and uses the largest observation of each block for modeling, possibly disregarding important observations. The POT method uses the GPD to model the tails of another distribution and uses all observations above some high threshold u. The latter method is the one used in this paper. To better account for the non-stationary characteristics that almost all financial asset returns exhibit (Cont, 2001), the focus shifts to dynamic EVT, where the incorporation of covariates in the modeling of extremes is an important tool.

Rigby and Stasinopoulos (2005) introduced the generalized additive models for location, scale, and shape (GAMLSS), which allows modeling of up to four parameters of a distribution depending on covariates via a vast number of (semi-)parametric additive functions of covariates. To fit the data by maximizing the (penalized if not fully parametric) likelihood function, GAMLSS uses a generalization of the CG algorithm (Cole and Green, 1992) and the RS algorithm (Rigby and Stasinopoulos, 1996). As we use the highly flexible GAMLSS, and model both the shape and the scale of the GPD in a high dimensional setting, variable selection is paramount. The challenge is not just variable selection, but also for which distribution parameter(s).

Variable selection can be done with regularization, where we add a regularization term to the loss function (e.g., negative log-likelihood) to prevent overfitting. The original GAMLSS algorithm included a method of explicit regularization that selects explanatory variables for each parameter of the distribution based on the generalized Akaike information criterion (generalized Akaike information criterion (GAIC)), where the penalty factor is proportional to the ℓ^0-norm. The GAIC relies on a large number of assumptions and has been criticized for selecting non-informative covariates (Judge, 1985; Hurvich and Tsai, 1989; Anderson and Burnham, 2002), too many variables (Ripley, 2004), as well as being unstable (Flack and Chang, 1987). Moreover, those techniques are infeasible in a high-dimensional setting.

Other explicit regularization methods, where the penalty term is proportional to respectively the ℓ^1-norm and ℓ^2-norm squared are the least absolute shrinkage and selection operator (LASSO; Tibshirani 1996) and ridge regularization (Hoerl and Kennard, 1970). The former produces a sparse model, but when the pairwise correlation among covariates is high, the predictive performance of the latter dominates (Tibshirani, 1996). When this is the case in a high-dimensional setting ($p > n$), LASSO also tends to select one variable of the pairwise correlated group at random (Zou and Hastie, 2005). Moreover, LASSO tends to select uninformative covariates in the early steps, making it impossible to avoid selecting these variables (Su, Bogdan, and Candès, 2017). Ridge, on the other hand, is by construction unable to produce a parsimonious model, since it keeps all covariates in the model.

Implicit regularization can be done by statistical boosting, which aims to combine weak learners into one strong learner to reduce both bias and variance. Mayr, Fenske, et al. (2012) adapted a component-wise boosting algorithm to the GAMLSS framework, named *gamboostLSS*. It fits the negative gradient of the loss function to every covariate separately by simple regression in every iteration. For each distribution parameter, the covariate which produces the biggest decrease in the loss function, i.e. fits best, gets updated. The negative gradient gets recomputed to select the best fitting covariate for the next iteration. Stopping the algorithm before every covariate is updated at least once consequently makes it a variable selection algorithm. Albeit simulations showed that in high-dimensional settings (more covariates than observations: $p > n$) the algorithm performed well on both variable selection and sparsity, it still was prone to unstable results (Mayr, Hofner, and Schmid, 2012b). The performance is similar to LASSO, but with better prediction performance (Hepp et al., 2016).

Hofner, Boccuto, and Göker (2015) combined this component-wise boosting algorithm with stability selection, based on the resampling technique introduced by Meinshausen and Bühlmann (2010). Stability selection creates B random subsamples of half the data, and fits a model to each subsample, generating B different models and thus B sets of selected covariates. The covariates that are selected for the B models with a rate equal to or larger than the specified threshold π_{thr} are the stable covariates and are used to fit

the final model to all data. This proved to improve the selection process and in addition, adds an error control for the number of falsely positively selected covariates. The stability selection algorithm of Hofner and Hothorn (2021) uses the improved stability selection method of Shah and Samworth (2013), which uses complementary pairs of subsamples: if a random subsample of half the data is created, the remaining observations make up the paired complementary subset. This method allows selecting more variables for the same level of error control compared to Hofner, Boccuto, and Göker (2015).

Thomas et al. (2018) modified this to a cyclical gradient boosting algorithm, such that the algorithm updates just one distribution parameter in each iteration. This is done for the combination of the distribution parameter and covariate that result in the biggest loss reduction overall, i.e., the algorithm chooses data-driven which distribution parameter to update in each iteration. Just one stopping hyperparameter has to be specified compared to one for each distribution parameter with the cyclical algorithm. This is more time-efficient than the traditional gamboostLSS algorithm, but more importantly, when combined with stability selection, the selected covariates often had fewer false positives as well as more true positives in the simulations.

Instead of explicit regularization of the full data, we can also apply dimensionality reduction which aims to preserve as much of the relevant information as possible whilst reducing the dimensionality of the data. One of the earliest methods of dimensionality reduction is principal component analysis (PCA; Pearson, 1901). The principal components are uncorrelated linear combinations of the data and have maximum variance, and the weight vectors used for the linear combinations are unit length. The first few principal components explain most of the variance and thus contain most of the relevant information. The principal components can be selected by e.g., the lowest GAIC or their t-values. When the model is fitted, the coefficients of the PCs are transformed back to the covariate coefficients, for interpretation. This results in regularized coefficients, but no variable selection is performed. Stasinopoulos, Rigby, Georgikopoulos, et al. (2021) adapt this technique as a novelty to the GAMLSS setting, where the principal components are used to model the parameters of the distribution of interest. They first expand the number of covariates by adding the first-order interaction effects, increasing the number of covariates from p to $p(p+1)/2$. This makes it a high-dimensional setting as the number of covariates is higher than their number of observations. With these methods, they were able to capture the spread behavior of the Greek-German 10-year government bond yields quite well.

This paper investigates dynamical tail risk modeling in a high-dimensional setting with interrelated explanatory variables by comparing different variable selection, regularization, and dimension reduction techniques. Moreover, it assesses if incorporating first-order interaction terms aids dynamical tail risk modeling.

3 Methodology

In this section, the methodology of the paper will be clarified. First, the peak-over-threshold (POT) will be introduced, together with the related generalized Pareto distribution (GPD). Then the GAMLSS framework, which is used for incorporating covariates in the estimation of the GPD parameters, will be described. Then the variable selection and regularization methods adapted to the GAMLSS framework will be explained.

3.1 Classical EVT: The Peak-Over-Threshold (POT) Method

In this subsection, we start with the notation of Coles et al. (2001). Let $X_1, ..., X_n$ be a sequence of identically and independently distributed variables with unknown distribution function G, and X an arbitrary term in the sequence. Denote threshold $u > 0$, scale parameter $\sigma > 0$ and shape parameter $\xi \in \mathbb{R}$. Then, for large enough u, the exceedances $Y = (X - u)$ conditional on $X > u$, approximately follow the generalized Pareto

distribution, $Y \sim GPD(\sigma, \xi)$, with distribution function

$$F_{\sigma,\xi}(y) = \begin{cases} 1 - \left(1 + \frac{\xi y}{\sigma}\right)^{-1/\xi} & \text{for } \xi \neq 0, \\ 1 - \exp\left(-\frac{y}{\sigma}\right) & \text{for } \xi = 0, \end{cases} \tag{1}$$

and density

$$f_{\sigma,\xi}(y) = \begin{cases} \frac{1}{\sigma}\left(1 + \frac{\xi y}{\sigma}\right)^{(-\frac{1}{\xi}-1)} & \text{for } \xi \neq 0, \\ \frac{1}{\sigma}\exp\left(-\frac{y}{\sigma}\right) & \text{for } \xi = 0, \end{cases} \tag{2}$$

where the support is $y \geq 0$ when $\xi \geq 0$ and $-\sigma/\xi \leq y \leq 0$ when $\xi < 0$. Some remarks are in order concerning the GPD.

First, the shape parameter ξ determines the type of the tail of the underlying distribution we are modeling. When the tails of the underlying distribution are exponentially bounded, such as with the normal distribution, the shape parameter ξ will be equal to zero. On the other hand, $\xi > 0$ will shape heavy-tailed distributions like the t-distribution while the GPD with $\xi < 0$ models the tail of distributions with a finite tail.

Secondly, the choice of the threshold implies a bias-variance tradeoff. A low threshold is likely to violate the assumptions of the asymptotics, increasing the bias. Conversely, a high threshold will lead to fewer available observations and thus a higher variance for the fitted model. The threshold can be chosen with methods like the mean residual life plot. Analysis of the empirical data will shed more light on the optimal choice of the threshold.

Finally, the exceedances approximately follow the GPD, whereas for estimation we assume they follow the GPD exactly.

3.2 Generalized Additive Models for Location, Scale and Shape (GAMLSS)

In this subsection the GAMLSS framework will be explained specifically for the GPD, where we first introduce the full GAMLSS model in Section 3.2.1. In Section 3.2.2, we give an outline of the estimation procedure in detail for the parametric GAMLSS, as our model in Section 3.3 has no random additive terms.

3.2.1 The Full Model

To account for the non-stationary characteristics of financial asset return data, we model the GDP dynamically by letting the parameters of the GPD depend on covariates through a GAMLSS model. Let $\boldsymbol{y} = (y_1, ..., y_n)^\top$ be the vector of exceedances above the threshold u, of length n. Because we want to model dynamically such that we have fitted distribution parameters ξ_i and σ_i for each y_i, we want a model for distribution parameter vectors $\boldsymbol{\xi}$ and $\boldsymbol{\sigma}$ of length n. Let $g_1(\cdot)$ and $g_2(\cdot)$ be known monotonic link functions that relate the GPD parameters ξ and σ to covariates by

$$g_1(\boldsymbol{\xi}) = \boldsymbol{\eta}_1 = \mathbf{X}_1\boldsymbol{\beta}_1 + \sum_{j=1}^{J_1} s_{1j}(\boldsymbol{x}_{1j}), \tag{3}$$

$$g_2(\boldsymbol{\sigma}) = \boldsymbol{\eta}_2 = \mathbf{X}_2\boldsymbol{\beta}_2 + \sum_{j=1}^{J_2} s_{2j}(\boldsymbol{x}_{2j}), \tag{4}$$

where $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are the additive predictor vectors, and where $\boldsymbol{\beta}_k^\top = (\beta_{k1}, ..., \beta_{kJ'_k})$, $k = 1, 2$ is a parameter vector and allows for modelling both distribution parameters as a linear function of each of the J'_k covariates. Moreover, $\mathbf{X}_k$ is a fixed known design matrix with dimensions $n \times J'_k, k = 1, 2$, and $s_{kj}, k = 1, 2$

is a nonparametric smoothing function applied to covariate $\boldsymbol{x}_{kj}, k = 1, 2$. If there are no additive terms such that $J_k = 0$, we have the parametric linear GAMLSS model

$$g_1(\boldsymbol{\xi}) = \boldsymbol{\eta}_1 = \mathbf{X}_1\boldsymbol{\beta}_1, \tag{5}$$

$$g_2(\boldsymbol{\sigma}) = \boldsymbol{\eta}_2 = \mathbf{X}_2\boldsymbol{\beta}_2. \tag{6}$$

The link functions $g_1(\cdot)$ and $g_2(\cdot)$ ensure that the range of the parameter estimates is valid for the GPD. The scale parameter σ is limited to positive values, so has domain $(0, \infty)$ and therefore the link function $g_2(\cdot)$ is the natural logarithm, such that $\boldsymbol{\eta}_2 = g_2(\boldsymbol{\sigma}) = \log(\boldsymbol{\sigma})$. The shape parameter ξ can take any value, has domain $(-\infty, \infty)$ and therefore the link function $g_1(\cdot)$ is the identity function, such that $\boldsymbol{\eta}_1 = g_1(\boldsymbol{\xi}) = \boldsymbol{\xi}$. If we incorporate the same covariates in the linear part of the model for both parameters, i.e., $J_1' = J_2'$, the design matrices will be equal: $\mathbf{X}_1 = \mathbf{X}_2 = \mathbf{X}$.

3.2.2 Estimation of the parametric GAMLSS Model

As the next subsections will make clear, we only use the parametric and boosted GAMLSS model. For the parametric GAMLSS model for the GPD as in Eqs. 5 and 6, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are estimated by maximizing the log-likelihood function ℓ. As we assume that each $y_i \sim GPD(\xi_i, \sigma_i)$ and given the density in Eq. 2, we maximize the following log-likelihood:

$$\ell = \sum_{i=1}^{n} \log\left(f(y_i; \xi_i, \sigma_i)\right) = \sum_{i=1}^{n} \log\left(\frac{1}{\sigma_i}\left(1 + \frac{\xi_i y_i}{\sigma_i}\right)^{-\frac{1}{\xi_i}-1}\right), \tag{7}$$

where $f(\cdot)$ denotes the density of the GPD, again for $\sigma_i > 0$, $\xi_i \in \mathbb{R}$ for $i = 1, ..., n$. The support is again $y_i \geq 0$ when $\xi_i \geq 0$ and $-\sigma_i/\xi_i \leq y_i \leq 0$ when $\xi_i < 0$. Maximization is done by either the RS algorithm, a generalization of the algorithm of Rigby and Stasinopoulos (1996), or by the CG algorithm, a generalization of the algorithm of Cole and Green (1992). The latter uses the first, second, and cross derivatives of the log-likelihood function enabling joint updates of the distribution parameter estimates, whereas the former does not use the cross derivatives and thus updates the distribution parameter estimates iteratively. Although the CG algorithm seems advantageous, it is rather unstable at the beginning and diverges easily. The RS algorithm is in general more stable and in most cases faster, which is why this one is used. The GPD distribution parameter vectors of length n are denoted as $\boldsymbol{\theta}_1 = g_1^{-1}(\boldsymbol{\eta}_1) = \boldsymbol{\xi} = (\xi_1, ..., \xi_n)^\top$ and $\boldsymbol{\theta}_2 = g_2^{-1}(\boldsymbol{\eta}_2) = \boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)^\top$, with $g_1^{-1}(\cdot)$ and $g_2^{-1}(\cdot)$ the inverse of the monotonic link functions. For fitting each distribution parameter $\boldsymbol{\theta}_k$, the so-called inner iteration is used. This inner iteration solves maximum likelihood equations numerically by either Newton-Raphson or the Fisher scoring algorithm. Both algorithms update the estimates of the predictor based on the score and its variance. The score function i.e., the first partial derivative of the log-likelihood with respect to the predictor, is defined as:

$$\boldsymbol{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k} = \left(\frac{\partial \ell}{\partial \boldsymbol{\theta}_k}\right) \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right), \tag{8}$$

with $\circ$ the Hadamard product. The vectors $\partial\ell/\partial\boldsymbol{\eta}_k, \partial\ell/\partial\boldsymbol{\theta}_k,$ and $d\boldsymbol{\theta}_k/d\boldsymbol{\eta}_k$ are all of length n, with elements $\partial\ell_i/\partial\eta_{k,i}, \partial\ell_i/\partial\theta_{k,i}$, and $d\theta_{k,i}/d\eta_{k,i}$, for $i = 1, ..., n$, respectively. The iterative weights $\boldsymbol{w}_k$ are defined as:

$$\boldsymbol{w}_k = -\boldsymbol{f}_k \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right) \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right), \tag{9}$$

where $\boldsymbol{f}_k$ is defined as the expectation of the second derivative of the log-likelihood with respect to parameter k, $\mathbb{E}\left(\partial^2\ell/\partial\boldsymbol{\theta}_k^2\right)$, leading to to the Fisher scoring algorithm. Instead of the actual scoring algorithms, we use the result of McCullagh and Nelder (1989), who prove that iteratively reweighted least squares (IRLS) of the modified response variable $\boldsymbol{z}_k$ on the covariates with weight matrix $\mathbf{W}_k = \text{diag}(\boldsymbol{w}_k)$, gives equal estimates as Fisher scoring in each iteration. The modified response variable to regress is calculated as

$$\boldsymbol{z}_k = \boldsymbol{\eta}_k + \boldsymbol{w}_k^{-1} \circ \boldsymbol{u}_k, \tag{10}$$

with $\boldsymbol{\eta}_k$ the additive predictor, $\boldsymbol{w}_k$ the weight, and $\boldsymbol{u}_k$ the score of distribution parameter $\boldsymbol{\theta}_k$, all vectors of length n. The estimated distribution parameter coefficients are calculated by weighted least squares (WLS) with

$$\hat{\boldsymbol{\beta}}_k = \left(\mathbf{X}_k^\top \mathbf{W}_k \mathbf{X}_k\right)^{-1} \mathbf{X}_k^\top \mathbf{W}_k \boldsymbol{z}_k. \tag{11}$$

With the new estimate of $\boldsymbol{\beta}_k$ we can subsequently update the estimates of the predictor $\boldsymbol{\eta}_k$, and thus the estimates of the distribution parameter $\boldsymbol{\theta}_k$. The new iteration then re-evaluates the score and weight vector at the new estimates. To run the inner iteration, the elements of the weight and parameter vector are initialized to one and zero respectively, i.e., $\boldsymbol{w}_k^{[0]} = \mathbf{1}$ and $\hat{\boldsymbol{\beta}}_k^{[0]} = \mathbf{0}$, for distribution parameter $\boldsymbol{\theta}_k$. Then, the updating sequence for each distribution parameter is given as

$$\hat{\boldsymbol{\beta}}_k^{[m]} \to \hat{\boldsymbol{\eta}}_k^{[m]} \to \hat{\boldsymbol{\theta}}_k^{[m]} \to \boldsymbol{u}_k, \boldsymbol{w}_k\Big|_{\boldsymbol{\eta}_k = \hat{\boldsymbol{\eta}}_k^{[m]}, \boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}_k^{[m]}, \boldsymbol{y} = \boldsymbol{y}} \to \hat{\boldsymbol{z}}_k^{[m]} \xrightarrow{\text{WLS}} \hat{\boldsymbol{\beta}}_k^{[m+1]}.$$

This concludes the inner iteration. The updating iterations are continued until the global deviance (GD), equal to minus two times the log-likelihood evaluated at the estimates of the current iteration, i.e., $\text{GD} = -2 \cdot \hat{\ell}^{[m]}$, converges. The outer iteration is nothing but iteratively fitting both distribution parameters performing the inner iteration. So for the GPD, repeatedly fit a model for $\boldsymbol{\xi}$ given the latest estimate $\hat{\boldsymbol{\sigma}}$, and subsequently, fit a model for $\boldsymbol{\sigma}$ using the new estimate $\hat{\boldsymbol{\xi}}$. Again, this is repeated until the global deviance converges. The pseudocode of the RS algorithm is given in Algorithm 3.1, where it is combined with principal component regression. For the estimation, we use the R package GAMLSS (Rigby and Stasinopoulos, 2007).

3.3 Principal Component Regression in GAMLSS

Although PCR has been around for a while, it was recently adapted to the GAMLSS framework by Stasinopoulos, Rigby, Georgikopoulos, et al. (2021). General PCR consists of three steps, where first, compact in this case, singular value decomposition is performed on the $n \times p$ suitably scaled (zero mean, unit variance) design matrix $\mathbf{X}$ of the p possible covariates:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top, \tag{12}$$

where $\mathbf{U}$ and $\mathbf{V}$ consist of respectively the left and right singular vectors, which are orthogonal such that $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_s$, the identity matrix of size $s = \min\{p, n\}$. Furthermore, $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_s)$ where $\sigma_1 \geq \sigma_2 \geq ... \geq 0$ are the rank-ordered singular values of $\mathbf{X}$. If matrix $\mathbf{X}$ is rank-deficient, at least one of the singular values will be equal to zero. When there are more observations than possible covariates, $n > p$, the matrix $\mathbf{U}$ is rectangular with dimensions $n \times p$ and $\boldsymbol{\Sigma}$ and $\mathbf{V}$ are $p \times p$ square matrices. Conversely, when $p > n$, $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are $n \times n$ square matrices and $\mathbf{V}$ is a $p \times n$ rectangular matrix. Let matrix $\mathbf{X}$ be of rank r which is equal to r non-zero singular values. The first r left singular vectors, columns of $\mathbf{U}$, form an orthonormal basis for the column space, span($\mathbf{X}$), while the first r right singular vectors, columns of $\mathbf{V}$, form an orthonormal basis for the row space, span($\mathbf{X}^\top$). The last $r - k$ right singular vectors i.e., columns of $\mathbf{V}$, provide an orthonormal basis for the null space of $\mathbf{X}$. The scores of the PCs are equal to $\mathbf{T} = \mathbf{X}\mathbf{V} = \mathbf{U}\boldsymbol{\Sigma}$, and the loadings are equal to $\mathbf{P} = \mathbf{V}^\top$. The loadings times the scores are conveniently equal to $\mathbf{T}\mathbf{P} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \mathbf{X}$, i.e., the principal components are linear combinations of the variables in $\mathbf{X}$.

In the second step of PCR, the response variable vector $\boldsymbol{v} = (v_1, ..., v_n)^\top$ is regressed onto the principal components, by treating the principal components scores $\mathbf{T}$ as the design matrix. As the score matrix $\mathbf{T}$ spans the same linear space as $\mathbf{X}$, any linear regression of $\boldsymbol{v}$ on $\mathbf{T}$ or $\mathbf{X}$ should give the same fitted values $\hat{\boldsymbol{v}}$. The ordinary least squares (OLS) parameter estimates, denoted as $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ will be equal to $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{v}$ and $\hat{\boldsymbol{\gamma}} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\boldsymbol{v}$. This gives the relation $\hat{\boldsymbol{v}} = \mathbf{T}\hat{\boldsymbol{\gamma}} = \mathbf{X}\mathbf{V}\hat{\boldsymbol{\gamma}} = \mathbf{T}\hat{\boldsymbol{\beta}} \iff \hat{\boldsymbol{\beta}} = \mathbf{V}\hat{\boldsymbol{\gamma}}$. Matrix $\mathbf{V}$ is also called the rotation matrix, as element $\mathbf{V}_{ij}$ gives us the weight of covariate $i = 1, .., p$ in principal component $j = 1, ..., r$. Since the covariates are demeaned and scaled, the constant of both regression models will be equal to just the mean of response variable v: $\hat{\beta}_0 = \hat{\gamma}_0 = \bar{v}$.

If we do not use all principal components, but only a subset denoted by λ such that we have score matrix $\mathbf{T}_\lambda$, we can use PCR as a model selection technique, which regularizes implicitly. The p possibly correlated

covariates are replaced by $|\lambda| < p$ uncorrelated linear combinations of the original covariates, addressing the problem of multicollinearity. The subset of selected principal components can be chosen by e.g., GAIC where we start with a model with the intercept and the first principal component (corresponding with the largest singular value), and iteratively add the next principal component as long as GAIC decreases. A possible problem is that principal components with lower singular values are eliminated from the model while the disregarded principal components with lower singular values can potentially even contribute more to the reduction of the sum of squares than the selected components (Hadi and Ling, 1998). Therefore we also use the t-statistics of the regression of the principal components on response variable v, selecting only the principal components considered significant and therefore informative, i.e., with a t-statistic greater than two.

To ease interpretation, we can transform the fitted coefficients of the selected principal components back to the coefficients of the original design matrix of the covariates in a third step. The pseudocode for the PCR in GAMLSS with the t-values approach is given in Algorithm 3.1. Estimation and optimization are done with the R package GAMLSS.FOREACH (Stasinopoulos, Rigby, and De Bastiani, 2021).
For the GAIC approach, lines 7 to 9 are replaced with the pseudocode in Algorithm C.1.

3.4 Gradient Boosting and Stability Selection For GAMLSS

As outlined in the literature review, the high flexibility and the large number of potential covariates in the design matrices in Eqs. 3 and 4 make variable selection of paramount importance for GAMLSS. In the case of more covariates than observations, $p > n$, gradient boosting proves to work well as a variable selection technique.

3.4.1 Component-wise gradient boosting

Gradient boosting ensembles weak base-learners to make for a strong prediction model, where each base-learner is a function of one covariate. The base-learners should perform at least slightly better than random guessing, e.g., a simple linear regression model for a linear term. Let $h_{k,j}(\boldsymbol{x}_j)$ denote base-learner j for predictor $\boldsymbol{\eta}_k$, a function of covariate $\boldsymbol{x}_j$. As the base-learner can be of all sorts, such as (non-)linear or smoothing function of the covariate, we can generalize Eqs. 3 and 4. The additive predictor vectors can then be written as a function of the covariates as

$$g_1(\boldsymbol{\xi}) = \boldsymbol{\eta}_1 = \boldsymbol{\xi}_0 + \sum_{j=1}^{p} h_{1,j}(\boldsymbol{x}_j), \tag{13}$$

$$g_2(\boldsymbol{\sigma}) = \boldsymbol{\eta}_2 = \boldsymbol{\sigma}_0 + \sum_{j=1}^{p} h_{2,j}(\boldsymbol{x}_j). \tag{14}$$

where $\boldsymbol{\eta}_0$ and $\boldsymbol{\sigma}_0$ represent the intercept in vector form, and where thus a priori all possible covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_p$ are incorporated in the models for both distribution parameters before gradient boosting.

To estimate the coefficients of the base-learners, gradient boosting minimizes the empirical risk R, which is the loss summed over all observations. As we estimate the parameters of an assumed distribution, the loss function will be the negative log-likelihood of the GPD, and the empirical risk thus the negative log-likelihood function. As we aim to minimize our empirical risk, or negative log-likelihood function, this is equivalent to maximizing the log-likelihood function as in Eq. 7. As an explicit solution is infeasible, we resort to numerical optimization by the noncyclical boosting algorithm for GAMLSS of Thomas et al. (2018).

First, we compute the negative partial derivative of the negative log-likelihood of the GPD with respect to each parameter predictor $\boldsymbol{\eta}_k$ and evaluate it at the estimates of the current iteration m. This is equivalent to the partial derivative of the log-likelihood with respect to the predictor (the score) as in Eq. 8,

$$\boldsymbol{u}_k = \left.\frac{\partial \ell}{\partial \boldsymbol{\eta}_k}\right|_{\boldsymbol{\eta}_k = \hat{\boldsymbol{\eta}}_k^{[m]}, \boldsymbol{y} = \boldsymbol{y}} \quad \text{for } k = 1, 2. \tag{15}$$

Algorithm 3.1: Principal component regression for the GPD within the RS algorithm, t-values approach.

Input:

Response variable $\boldsymbol{y} = (y_1, ..., y_n)^\top$, with assumed distribution $y \sim GPD(\xi, \sigma)$,
p possible explanatory variables with $n \times p$ design matrix $\mathbf{X} = (\boldsymbol{x}_1, ...\boldsymbol{x}_p)$.

Output:

A GAMLSS model with fitted GPD parameters $\hat{\boldsymbol{\theta}}_1 = \hat{\boldsymbol{\xi}}$ and $\hat{\boldsymbol{\theta}}_2 = \hat{\boldsymbol{\sigma}}$ with PCs as covariates.

Initialization:

Derive the log-likelihood function of the GPD, $\ell = \sum_{i=1}^{n} \log\left(\frac{1}{\sigma_i}\left(1 + \frac{\xi_i y_i}{\sigma_i}\right)^{-\frac{1}{\xi_i}-1}\right)$,

Derive $\boldsymbol{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k} = \left(\frac{\partial \ell}{\partial \boldsymbol{\theta}_k}\right) \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right), \boldsymbol{f}_k = \frac{\partial^2 \ell}{\partial \boldsymbol{\theta}_k^2}$ or $\mathbb{E}\left(\frac{\partial^2 \ell}{\partial \boldsymbol{\theta}_k^2}\right), \boldsymbol{w}_k = -\boldsymbol{f}_k \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right) \circ \left(\frac{d\boldsymbol{\theta}_k}{d\boldsymbol{\eta}_k}\right)$,

Scale $\mathbf{X}$ and perform compact SVD: $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$, score matrix $\mathbf{T} = \mathbf{U}\boldsymbol{\Sigma}$ and rotation matrix $\mathbf{P} = \mathbf{V}^\top$. Set $r = \text{rank}(\boldsymbol{\Sigma})$, and subset the first r (non-zero) columns of $\mathbf{T} = \mathbf{T}_{[1:r]}$,

Set $\boldsymbol{w}_k^{[0]} = \mathbf{1}$, $\hat{\boldsymbol{\beta}}_k^{[0]} = \mathbf{0}$, $\hat{\boldsymbol{\eta}}_k^{[0]} = \mathbf{0}$, and $\hat{\boldsymbol{\theta}}_k^{[0]} = g_k^{-1}(\hat{\boldsymbol{\eta}}_k^{[0]})$ for $k = 1, 2$,

Set global deviance change of inner and outer iteration $\Delta\text{GD}_{OUT} = \Delta\text{GD}_{IN} = 1$, $q, m = 0$.

while $|\Delta\text{GD}_{OUT}| > 0.001$ **do**

  for $k = 1, 2$, **do**

    Treat estimate $\hat{\boldsymbol{\theta}}_j, j \neq k$ as given, and:

    while $|\Delta\text{GD}_{IN}| > 0.001$ **do**

      Evaluate score and weights at current estimates, $\boldsymbol{u}_k, \boldsymbol{w}_k\big|_{\boldsymbol{\eta}_k = \hat{\boldsymbol{\eta}}_k^{[m]}, \boldsymbol{\theta}_k = \hat{\boldsymbol{\theta}}_k^{[m]}, \boldsymbol{y} = \boldsymbol{y}}$.

      Update modified response variable $\boldsymbol{z}_k^{[m]} = \hat{\boldsymbol{\eta}}_k^{[m]} + \boldsymbol{w}_k^{-1} \circ \boldsymbol{u}_k$.

      Estimate the PCs coefficients $\hat{\boldsymbol{\gamma}} = \left(\mathbf{T}^\top \mathbf{W}_k \mathbf{T}\right)^{-1} \mathbf{T}^\top \mathbf{W}_k \boldsymbol{z}_k^{[m]}$, with $\mathbf{W}_k = \text{diag}(\boldsymbol{w}_k)$.

      Select subset λ of significant PCs: $\lambda = \{\text{PC}_\text{i} \mid \frac{\hat{\gamma}_\text{i}}{\text{se}(\hat{\gamma}_\text{i})} > 2, i = 1, ..., r\}$,

      Set columns $\boldsymbol{t}_{k,i} = \mathbf{0}$ if $\text{PC}_i \notin \lambda$, resulting in score matrix $\mathbf{T}_\lambda$ of significant PCs.

      Recalculate $\hat{\boldsymbol{\gamma}}_\lambda = \left(\mathbf{T}_\lambda^\top \mathbf{W}_k \mathbf{T}_\lambda\right)^{-1} \mathbf{T}_\lambda^\top \mathbf{W}_k \hat{\boldsymbol{z}}_k^{[m]}$

      Transform to coefficients of the covariates $\hat{\boldsymbol{\beta}}_k^{[m+1]} = \mathbf{V}\hat{\boldsymbol{\gamma}}_\lambda$.

      Update predictor $\hat{\boldsymbol{\eta}}_k^{[m+1]} = \mathbf{X}_k \hat{\boldsymbol{\beta}}_k^{[m+1]}$, and estimate $\hat{\boldsymbol{\theta}}_k^{[m+1]} = g_k^{-1}(\hat{\boldsymbol{\eta}}_k^{[m+1]})$.

      Calculate $\text{GD}_{IN}^{[m+1]} = -2 \cdot \hat{\ell}^{[m+1]}$, evaluated at current estimate $\hat{\boldsymbol{\theta}}_k^{[m+1]}$.

      Set $\Delta\text{GD}_{IN} = \text{GD}_{IN}^{[m+1]} - \text{GD}_{IN}^{[m]}$.

      $m = m + 1$.

    Extract estimate after convergence of inner iteration: $\hat{\boldsymbol{\theta}}_k^{[q+1]} = \hat{\boldsymbol{\theta}}_k^{[m]}$.

  Calculate global deviance $\text{GD}_{OUT}^{[q+1]} = -2 \cdot \hat{\ell}$, at current estimates $\hat{\boldsymbol{\theta}}_k^{[q+1]}, k = 1, 2$.

  Set $\Delta\text{GD}_{OUT} = \text{GD}_{OUT}^{[q+1]} - \text{GD}_{OUT}^{[q]}$.

  $q = q + 1$.

The algorithm fits each base-learner $\hat{h}_{k,j}(\boldsymbol{x}_j)$ to the partial derivatives of the negative log-likelihood with respect to predictor $\boldsymbol{\eta}_k$ in each iteration. The base-learner that fits best, i.e. minimizes the residual sum of squares, is selected:

$$j^* = \underset{j \in 1,\ldots,p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(u_{k,i} - \hat{h}_{k,j}(x_{j,i}) \right)^2 \qquad \text{for } k = 1, 2, \tag{16}$$

with $u_{k,i}$ the partial derivative of the log-likelihood with respect to predictor η_k evaluated at $\eta_{k,i} = \hat{\eta}_{k,i}^{[m]}, y = y_i$, with x_{ij} observation i of covariate j. These are all equal to the i-th element of vectors $\boldsymbol{u}_k, \hat{\boldsymbol{\eta}}_k^{[m]}, \boldsymbol{y}$ and $\boldsymbol{x}_j$, respectively. The cyclical boosting algorithm (Mayr, Fenske, et al., 2012) then, for each distribution parameter iteratively, uses the best fitting base-learner to update the predictor:

$$\hat{\boldsymbol{\eta}}_k^{[m+1]} = \hat{\boldsymbol{\eta}}_k^{[m]} + sl \cdot \hat{h}_{k,j^*}(\boldsymbol{x}_{j^*}) \qquad \text{for } k = 1, 2, \tag{17}$$

where $0 < sl \ll 1$ is the step length (usually set to $sl = 0.1$ as the value is not so important) and $\hat{h}_{k,j^*}$ is the selected, best fitting base-learner for predictor $\boldsymbol{\eta}_k$. If the algorithm is stopped early, at least before all the base-learners are selected, their corresponding covariate is not in the final model and thus variable selection and shrinkage are performed. As long as the number of iterations m is below the specified stopping value m_{stop}, the predictor of every distribution parameter gets updated. This way base-learners that are possibly of much less significance are added to the model compared to base-learners for the other distribution parameters.

This can give problems when the algorithm is combined with stability selection, and a careful choice of m_{stop} for every distribution parameter is needed. Optimization of the hyperparameters thus becomes k-dimensional, with k the number of parameters of the distribution. The noncyclical gradient boosting algorithm of Thomas et al. (2018) solves these problems by updating only one distribution parameter in each iteration, by selecting the *combination* of distribution parameter and base-learner that has the most improvement in the negative log-likelihood. Therefore, the possible improvement in the loss function by each updated predictor is computed first:

$$\Delta\rho_k = -\ell \left(\boldsymbol{y}; \hat{\boldsymbol{\eta}}_k^{[m]} + sl \cdot \hat{h}_{k,j^*}(\boldsymbol{x}_{j^*}) \right), \qquad \text{for } k = 1, 2, \tag{18}$$

where ℓ is the log-likelihood of the GPD as in Eq. 7, and $\hat{h}_{k,j^*}$ is the best fitting base-learner for predictor $\boldsymbol{\eta}_k$. The predictor estimate that gets updated is determined by the *overall* best-fitting base-learner:

$$k^* = \underset{k=1,2}{\operatorname{argmin}} \, \Delta\rho_k, \tag{19}$$

where k^* is the index for the distribution parameters predictor that gets updated. Now we update the predictor with the base-learner that fits best *overall*:

$$\hat{\boldsymbol{\eta}}_k^{[m+1]} = \hat{\boldsymbol{\eta}}_k^{[m]} + sl \cdot \hat{h}_{k,j^*}(\boldsymbol{x}_{j^*}) \qquad \text{if } k = k^*, \tag{20}$$

$$\hat{\boldsymbol{\eta}}_k^{[m+1]} = \hat{\boldsymbol{\eta}}_k^{[m]} \qquad \text{otherwise.} \tag{21}$$

The base-learners are partitioned and sequenced automatically based on the data while fitting, based on just one stopping value m_{stop}. Optimization is greatly eased because of scalar optimization, compared to e.g., grid search for the cyclical algorithm. The pseudocode for the non-cyclical component-wise gradient boosting for the GPD is given in Algorithm 3.2. For estimation and optimization of the noncyclical boosting algorithm, we use the R package GAMBOOSTLSS.

3.4.2 Stability Selection

As outlined in Section 2, many variable selection techniques such as implicit regularization or boosting are known to be unstable or include too many noise variables. Stability selection tries to overcome this by

Algorithm 3.2: Noncyclical component-wise gradient boosting for the GPD, before stability selection

Input:
Response variable $\boldsymbol{y} = (y_1, ..., y_n)^\top$,
Possible covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_p$,
Number of selected covariates q.
Output:
A set of selected covariates $\hat{S}$ for GPD parameters $\boldsymbol{\theta}_1 = \boldsymbol{\xi}$ and $\boldsymbol{\theta}_2 = \boldsymbol{\sigma}$.
Initialization:
Specify a set of base-learners for each GPD distribution parameter $\boldsymbol{\theta}_1 = \boldsymbol{\xi}$ and $\boldsymbol{\theta}_2 = \boldsymbol{\sigma}$, i.e., $h_{k,1}(\cdot), ..., h_{k,p_k}(\cdot)$, where p_k is the cardinality of the set of base-learners specified for $\boldsymbol{\theta}_k$,
Set predictors $\hat{\boldsymbol{\eta}}_1^{[0]} = \hat{\boldsymbol{\eta}}_2^{[0]} = \mathbf{0}$, for $\boldsymbol{\xi}$ and $\boldsymbol{\sigma}$ respectively,
Set $\hat{S} = \emptyset$, $m = 0$.

1 **while** $|\hat{S}| < q$ **do**

2 **for** $k = 1, 2$ **do**

3 Compute partial derivative of the log-likelihood of the GPD with respect to predictor $\boldsymbol{\eta}_k$ and evaluate at current estimates $\hat{\boldsymbol{\eta}}^{[m]}$:

$$\boldsymbol{u}_k = \frac{\partial \ell}{\partial \boldsymbol{\eta}_k}\bigg|_{\boldsymbol{\eta}_k = \hat{\boldsymbol{\eta}}_k^{[m]}, \boldsymbol{y} = \boldsymbol{y}}.$$

4 Fit each of the base-learners $h_{k,1}(\boldsymbol{x}_1), ..., h_{k,p_k}(\boldsymbol{x}_{p_k})$ specified for distribution parameter $\boldsymbol{\theta}_k$ to the partial derivative $\boldsymbol{u}_k$. Select the best-fitting base-learner by the residual sum of squares with respect to the partial derivative:

$$j^* = \underset{j \in 1, ..., p}{\operatorname{argmin}} \sum_{i=1}^{n} \left(u_{k,i} - \hat{h}_{k,j}(x_{ij}) \right)^2.$$

5 Compute the change in the negative log-likelihood:

$$\Delta\rho_k = -\ell \left(\boldsymbol{y}; \hat{\boldsymbol{\eta}}_k^{[m]} + sl \cdot \hat{h}_{k,j^*}(\boldsymbol{x}_{j^*}) \right).$$

6 Select the *overall* best-fitting combination of base-learner and predictor $\hat{\boldsymbol{\eta}}_k$: $k^* = \operatorname{argmin}_{k=1,2} \Delta\rho_k$.

7 Update the estimates of the predictors by:

$$\begin{aligned} \hat{\boldsymbol{\eta}}_k^{[m+1]} &= \hat{\boldsymbol{\eta}}_k^{[m]} + sl \cdot \hat{h}_{k,j^*}(\boldsymbol{x}_{j^*}) && \text{if } k = k^*, \\ \hat{\boldsymbol{\eta}}_k^{[m+1]} &= \hat{\boldsymbol{\eta}}_k^{[m]} && \text{otherwise.} \end{aligned}$$

8 Add the selected base-learner $\hat{h}_{k^*,j^*}$ to the set $\hat{S}$.

repeatedly applying a certain variable selection technique on a large number of random subsamples of the original data, selecting only the covariates that are consistently used for fitting. The approach is rather simple and can be combined with almost every (high-dimensional) variable selection technique. We use a modification (Shah and Samworth, 2013) of the original approach of Meinshausen and Bühlmann (2010), called complementary pairs stability selection.

First, a random subset of half of the observations is selected B times, where each subset complement of the data makes the complementary pair, resulting in $2B$ subsamples of size $n/2$. Then a boosted model is fitted to each of the subsamples until a pre-specified number of q covariates is selected, resulting in $2B$ sets of size q. The relative selection frequency is equal to:

$$\hat{\pi}_j := \frac{1}{2B} \sum_{b=1}^{2B} \mathbb{1}_{j \in \hat{S}_b}, \tag{22}$$

where $\hat{S}_b$ denotes the set of selected covariates by the boosted model, fitted to subsample b, with $\mathbb{1}$ the indicator function. Variable selection is done by specifying threshold value π_{thr}, where covariates are considered stable and subsequently selected if their relative selection frequency is equal to or larger than the threshold value:

$$\hat{S}_{\text{stable}} := \{j : \hat{\pi}_j \geq \pi_{\text{thr}}\}. \tag{23}$$

Hyperparameter q is the number of selected covariates by the boosted model on each subset and π_{thr} the threshold value for the relative selection frequency. The simulation study will shed more light on the optimal choice of these hyperparameters, specifically for the GPD. For estimation and optimization we use the R package STABSEL (Hofner and Hothorn, 2021). The pseudocode for complementary pairs stability selection is given in Algorithm 3.3 (Hofner and Hothorn, 2021).

Algorithm 3.3: Complementary pairs stability selection for component-wise gradient boosting for the GPD parameters in the GAMLSS framework.

Input:
Response variable $\boldsymbol{y} = (y_1, ..., y_n)^\top$,
Possible covariates $\boldsymbol{x}_1, ..., \boldsymbol{x}_p$,
The number of iterations, B, for stability selection,
Number of selected covariates q in each run by noncyclical boosting,
Relative selection rate threshold π_{thr},
Output:
A set of stable covariates $\hat{S}_{\text{stable}}$ for distribution parameters $\theta_1 = \xi$ and $\theta_2 = \sigma$.

for $b = 1, ..., B$ **do**

Split the data by first randomly selecting $[n/2]$ observations, where $[n/2]$ denotes the largest integer $\leq n/2$. The remaining subset of $n - [n/2]$ observations makes the complementary pair.

Run Algorithm 3.2 on both subsets of the data, which outputs the sets of selected base-learners denoted as $\hat{S}_{2b-1}$ and $\hat{S}_{2b}$.

Compute the relative selection frequencies for each

$$\hat{\pi}_j := \frac{1}{2B} \sum_{b=1}^{2B} \mathbb{1}_{j \in \hat{S}_b},$$

Select base-learners with a relative selection frequency of at least π_{thr} for the final model:

$$\hat{S}_{\text{stable}} := \{j : \hat{\pi}_j \geq \pi_{\text{thr}}\}.$$

4 Simulation Study

In this section, we compare the performance of gradient boosting technique *gamboostLSS* to the same technique when combined with stability selection, for variable selection. With these results informed choices can be made about the hyperparameter choice for both models in the empirical application. No hyperparameters need to be set for the GAIC and t-statistics principal component methods in the GAMLSS framework.

4.1 Data-generating process

The extreme observations are generated according to a generalized Pareto distribution (GPD) with linear predictors $\boldsymbol{\eta}_\sigma = -p_{\text{inf}} \cdot \mathbf{1} - \mathbf{X}\boldsymbol{\beta}_1 - \tilde{\mathbf{X}}\boldsymbol{\beta}_2$ and $\boldsymbol{\eta}_\xi = -1.5 \cdot p_{\text{inf}} \cdot \mathbf{1} - \mathbf{X}\boldsymbol{\beta}_3 - \tilde{\mathbf{X}}\boldsymbol{\beta}_4$, with p_{inf} the number of informative variables and with logarithmic link functions such that $\boldsymbol{\sigma} = \exp(\boldsymbol{\eta}_\sigma)$ and $\boldsymbol{\xi} = \exp(\boldsymbol{\eta}_\xi)$.

$$Y_i \sim GPD(\sigma_i, \xi_i), \quad i = 1, ..., n.$$

The observations $\boldsymbol{x}_i = (x_{i1}, ..., x_{ip}), i = 1, ..., n$ were independently drawn from

$$\boldsymbol{x} \sim \frac{1}{2} \cdot \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}),$$

and gathered in the $(n \times p)$ design matrix $\mathbf{X}$. We set the number of variables to $p = 20$ and adding interactions increases the number of variables to $p(p+1)/2 = 210$. We set the number of observations to $n = 209$, resulting in a high-dimensional setting. The interactions are gathered in the $(n \times (p(p-1)/2))$ design matrix $\tilde{\mathbf{X}}$. The number of informative linear and informative interaction variables are equal and varied within $p_{\text{inf}} \in \{1, 2, ..., 10\}$, resulting in 10 different simulation settings, with at most 20 informative variables for both parameters: 10 linear and 10 interaction terms. The coefficients β_{kj}, the elements of $\boldsymbol{\beta}_k$, for $k = 1, ..., 4$, are set to 1 for informative variables and set to 0 for the remaining variables. We use three settings for the covariance matrix $\boldsymbol{\Sigma}$:

1. independent variables, i.e., $\boldsymbol{\Sigma} = \mathbf{I}$,
2. correlated variables drawn from a Toeplitz design with covariance matrix $\boldsymbol{\Sigma}_{kl} = 0.5^{|k-l|}, k, l = 1, ..., p$,
3. highly correlated variables drawn from a Toeplitz design with covariance matrix $\boldsymbol{\Sigma}_{kl} = 0.9^{|k-l|}, k, l = 1, ..., p$.

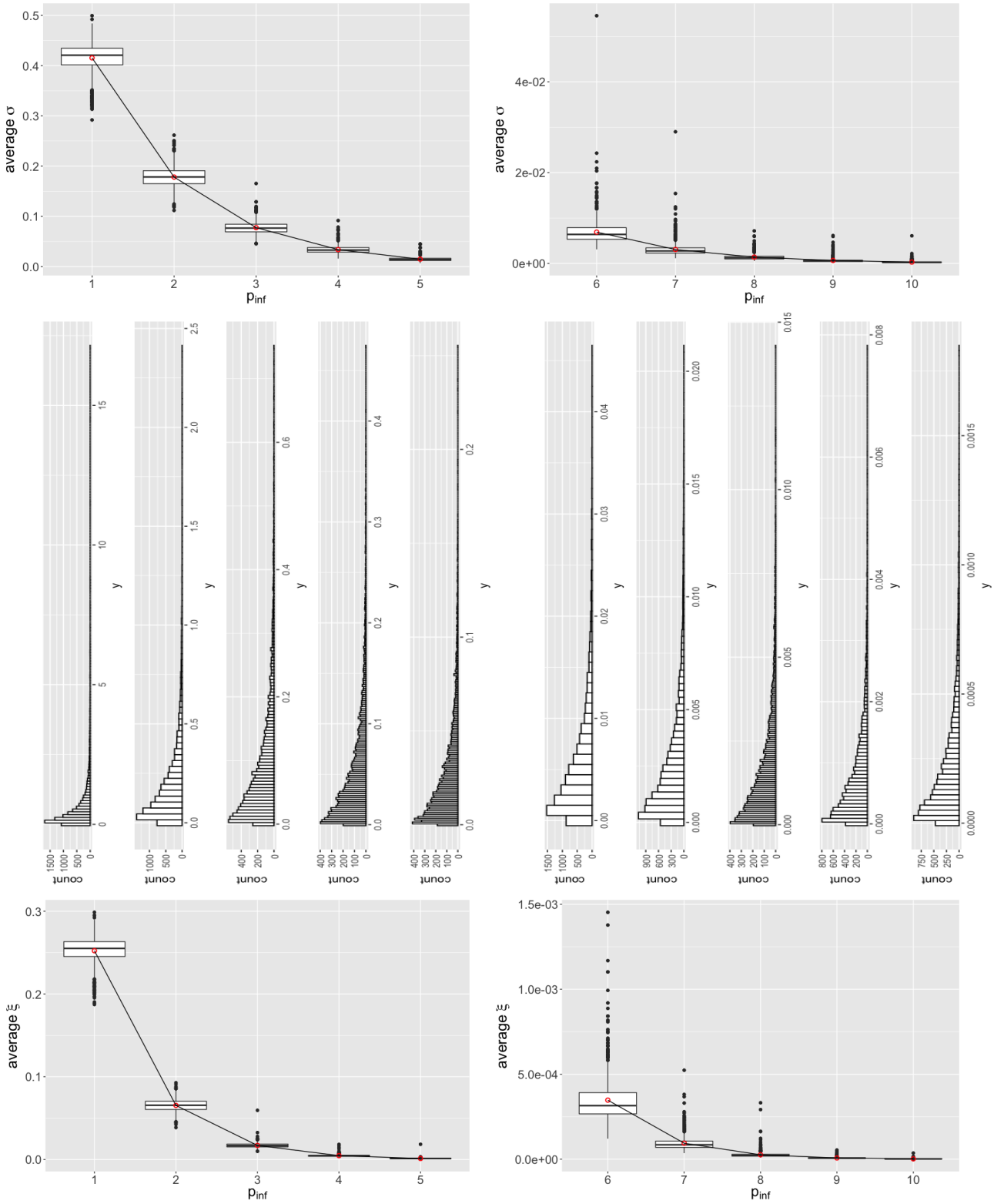
The boxplots and mean of the average σ and ξ in the simulations, with the covariance matrix in setting 2, for each value of p_{inf} together with the histogram of the simulated values using mean parameter values and 10000 observations are given in Figure 1.

4.2 Choice of hyperparameters

Noncyclical gradient boosting depends, as described in Section 3.4.1, on hyperparameter m_{stop}. To account for different values of p_{inf}, the value set is to $m_{\text{stop}} = 20 \cdot p_{\text{inf}}$. For stability selection q and π_{thr} need to be specified, as described in Section 3.4.2. The number of covariates selected per iteration, q, should be at least as big as the number of informative variables (Meinshausen and Bühlmann, 2010), so we set $q = 2 \cdot p_{\text{inf}} + 1$. For π_{thr}, any value $\in (0.5, 1)$ is potentially acceptable (Hofner, Boccuto, and Göker, 2015), so we set $\pi_{\text{thr}} = 0.51$ as a base case. In Section 4.4 the sensitivity with respect to π_{thr} will be further investigated.

4.3 Results

To evaluate the gradient boosting and stability selection methods of Section 3.4 we look at the percentage of variables that are selected correctly, the TPR. As a base case we use the first setting of the covariance matrix, i.e., $\boldsymbol{\Sigma} = \mathbf{I}$. The number of simulation runs is 1000.



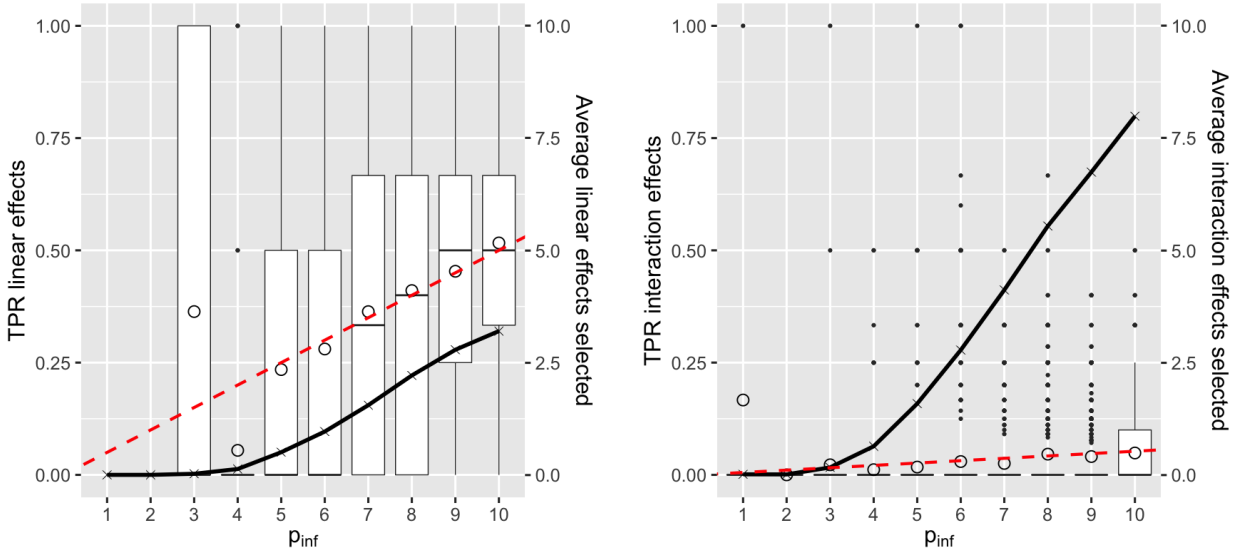
(a) $p_{\text{inf}} = \{1, 2, 3, 4, 5\}$ (b) $p_{\text{inf}} = \{6, 7, 8, 9, 10\}$

Figure 1: Boxplots and mean (red dots) of the average σ and ξ in the 1000 simulations for each value of p_{inf} together with the histogram of the simulated extreme values using mean parameter values and 10000 observations.

Shape parameter $\boldsymbol{\xi}$

The estimation even of a constant shape parameter is known to be difficult (Coles et al., 2001; Park and Kim, 2016), and estimation only gets harder for a dynamic shape parameter in a high-dimensional setting. In a total of 1000 simulation runs with the second setting of the covariance matrix, gradient boosting once selected two covariates for the shape parameter, and sixteen times selected one covariate, all incorrect. Stability selection did not once select a covariate for the shape parameter.

To check the performance of selecting variables for the shape parameter, we change component-wise gradient boosting from noncyclical to cyclical updates, updating every parameter in every iteration, like in the original algorithm (Mayr, Fenske, et al., 2012). This forces the method to select variables for the shape parameter. The results are in Figure 2.



(a) TPR linear covariates

(b) TPR interaction covariates

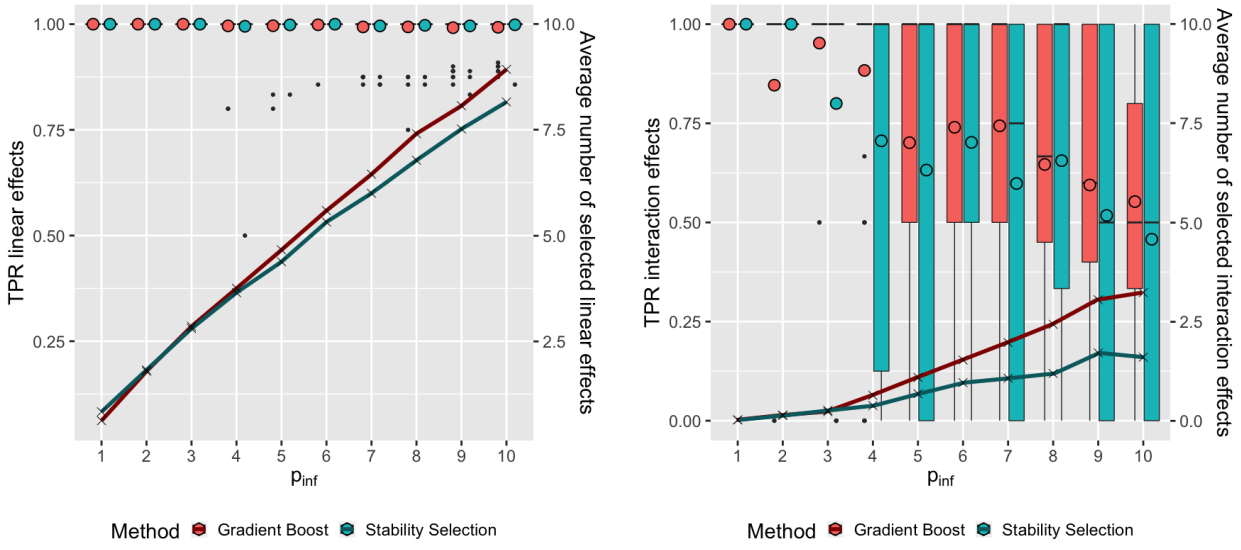
Figure 2: Boxplots and mean (circle) of the TPRs of the selected covariates by the gradient boosting method for the shape parameter, for every setting of p_{inf}. The expected TPR of random variable selection is indicated by the dashed red line. The black line and the secondary vertical axis indicate the average number of selected variables in each of the 1000 simulation runs per setting of p_{inf}.

The slopes of the dashed red lines are equal to $p_{\text{inf}}/10$ and $p_{\text{inf}}/190$ for Figures 2a and 2b respectively, the expected TPR of the linear and interaction terms respectively. We conclude that gradient boosting is unable to outperform random selection of covariates, which is why the noncyclical algorithm hardly, and stability selection never, selects covariates for the shape parameter.

Scale parameter $\boldsymbol{\sigma}$

For both regularization methods, the TPR of the selected linear and interaction terms for σ and the corresponding average number of selected variables in each simulation setting of p_{inf} are displayed in Figure 3.

For the linear covariates of σ both methods perform very well, with the lowest average TPR at 99.26% for gradient boosting and $p_{\text{inf}} = 10$. The stability selection method seems to outperform gradient boosting for p_{inf} high, but the higher number of selected variables by gradient boosting causes bias. The average number of selected linear covariates ranges between 0.62 and 8.93 for gradient boosting and between 0.83 and 8.16 for stability selection. Both averages seem to increase linearly in p_{inf}, supporting the hyperparameter



(a) TPR linear covariates

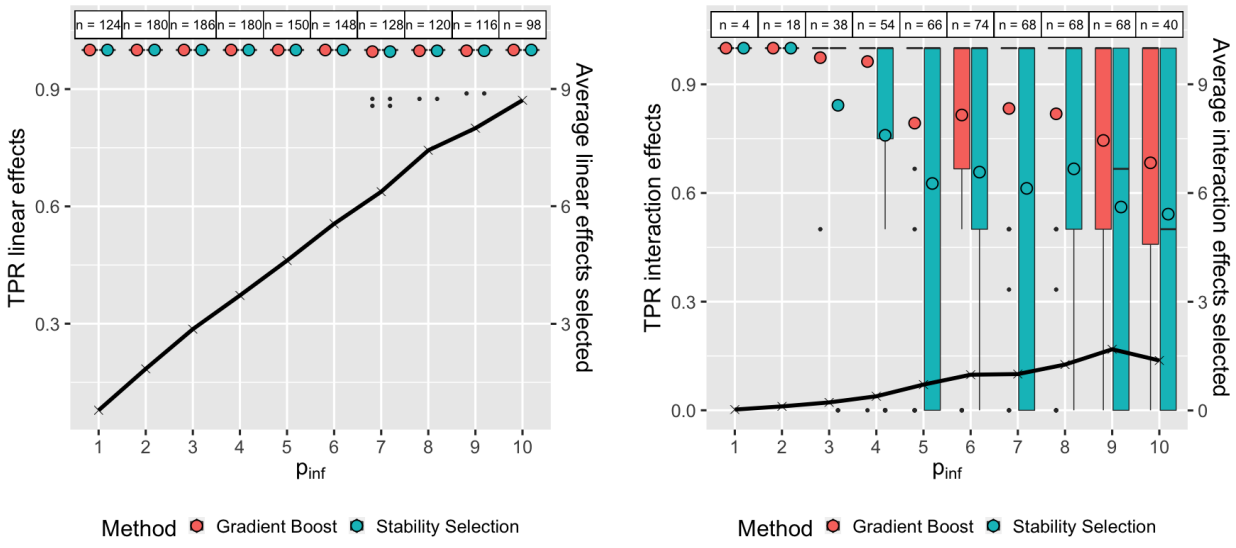
(b) TPR interaction covariates

Figure 3: Comparison by boxplot and mean (circle) of the TPR of the selected linear and interaction terms for σ of the gradient boosting and stability selection method, for every setting of p_{inf}. The secondary vertical axis indicates the average number of selected variables in each of the 100 simulation runs per setting of p_{inf}.

settings. For the interaction terms, the TPR of both methods is more inconsistent as indicated by the large ranges of the boxplots. Moreover, the average TPR shows a decreasing trend for both methods. The average selected number of both linear and interaction terms increases in p_{inf} for both methods. To account for different numbers of selected covariates, Figure 4 shows the results of the simulation runs only where stability selection and gradient boosting selected the same number of linear or interaction terms, the number at the top indicates the number of times this was the case.

For the linear covariates, the methods perform both extremely well and exactly the same. Subsetting the simulation runs on the same number of selected interaction effects eliminates the bias and has a magnifying effect on the outperformance of stability selection by gradient boosting, as gradient boosting selected more interaction terms in Figure 3b. We can conclude that both methods preferably select the easier-to-identify linear covariates, and that stability selection does not offer an improvement to gradient boosting in this setting.

For the interaction terms, two things stand out. The outperformance of stability selection by gradient boosting, while stability selection is essentially repeatedly applying gradient boosting, means the selection of interaction terms is very unstable. Secondly, the average TPR seems to decrease in p_{inf}, while a higher p_{inf} means a higher a priori probability of a true positive. A relative increase in the average number of selected interaction effects disproportionate to the relative increase in p_{inf} is also not the case, and thus does not explain the decrease of TPR in p_{inf}. We conclude that gradient boosting has a hard time selecting the informative interaction effects, becoming even more true for a higher p_{inf}.



(a) TPR linear covariates

(b) TPR interaction covariates

Figure 4: Comparison of the gradient boosting and stability selection method, for every setting of p_{inf} with an equal number of selected covariates, for covariance matrix setting 1.

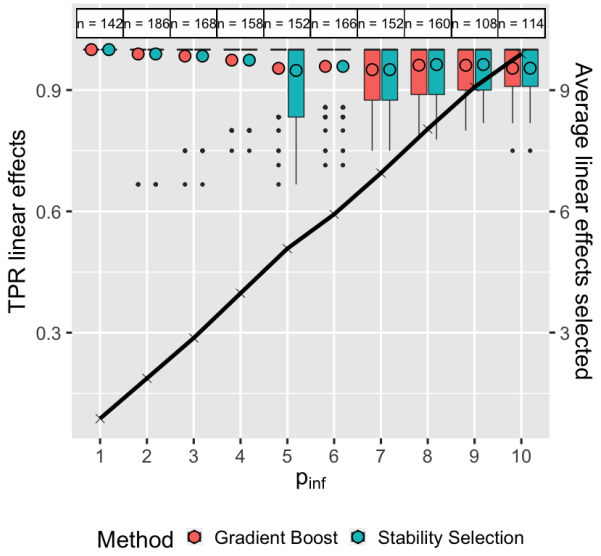
Collinearity

We introduce collinearity through settings 2 and 3 of the covariance matrix. First we use the second setting, $\mathbf{\Sigma}_{kl} = 0.5^{|k-l|}, k, l = 1, ..., p$, where the hyperparameters of the two methods remain the same. We only look at the simulation runs where the number of selected covariates (linear, interaction, or total) is the same for both methods. The results are in Figure 5.

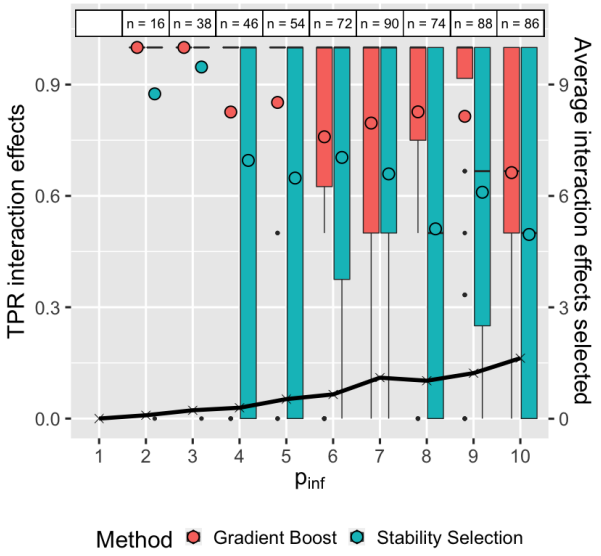
Compared to the results of Figure 4 with the unit matrix as the covariance matrix of the covariates, the average TPRs decrease as expected, except for $p_{\text{inf}} = 1$. In the simulation runs where both methods select the same amount of covariates, there are more linear and almost the same number of interaction terms selected on average compared to without collinearity. Both methods still have an almost equal TPR for the linear covariates, and stability selection now performs worse than gradient boosting for every p_{inf} if we look at the TPR of the interaction terms. The number of cases at the top does not differ much compared to Figure 4, meaning both methods still select the same number of covariates a lot of the time.

Finally, we increase collinearity by setting the covariance matrix to the third setting: $\mathbf{\Sigma}_{kl} = 0.9^{|k-l|}, k, l = 1, ..., p$. The results are in Figure 6.

Further increasing the collinearity surprisingly increases the number of selected variables, for both the linear and interaction terms. It also further decreases the average TPRs. The TPRs seem to decrease slightly in p_{inf}, for both the linear and interaction terms. This is unexpected, as a higher p_{inf} means a higher a priori probability of correctly selecting a covariate. For the first time, stability selection improves the TPR compared to gradient boosting for some settings, but only for the linear covariates and increasingly for higher p_{inf}. Moreover, the number of cases as indicated at the top of Figure 6b is much lower compared to Figure 4b, indicating collinearity magnifies the instability of selecting the interaction terms resulting in lower TPRs for both methods and a decrease in the number of selected interaction terms for stability selection. Stability selection still gets outperformed and there is still a decreasing trend of the TPRs in p_{inf}. We conclude that with the added uncertainty of the high collinearity, the selection of interaction terms becomes even more unstable. Therefore stability selection resorts to mainly selecting linear covariates.

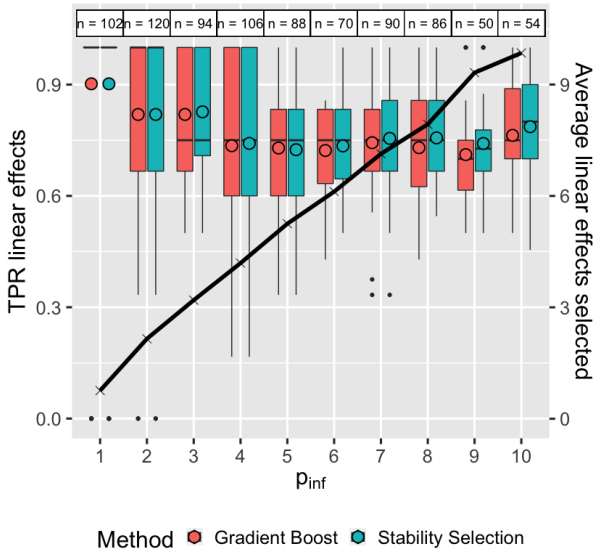


(a) TPR linear covariates

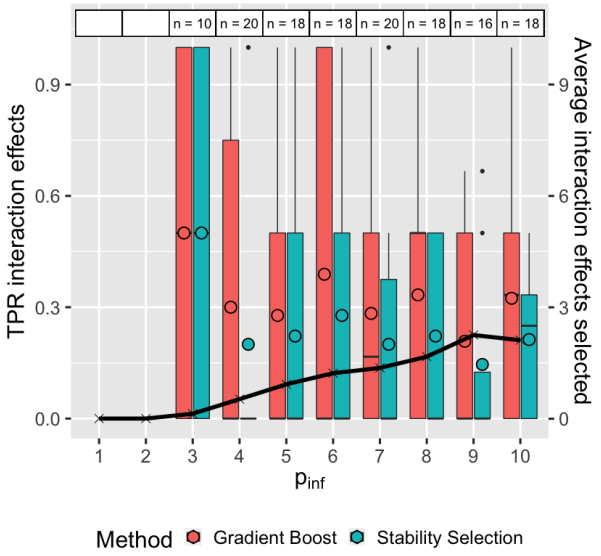


(b) TPR interaction covariates

Figure 5: Comparison of the gradient boosting and stability selection method, for every setting of p_{inf} with an equal number of selected covariates, for covariance matrix setting 2.



(a) TPR linear covariates



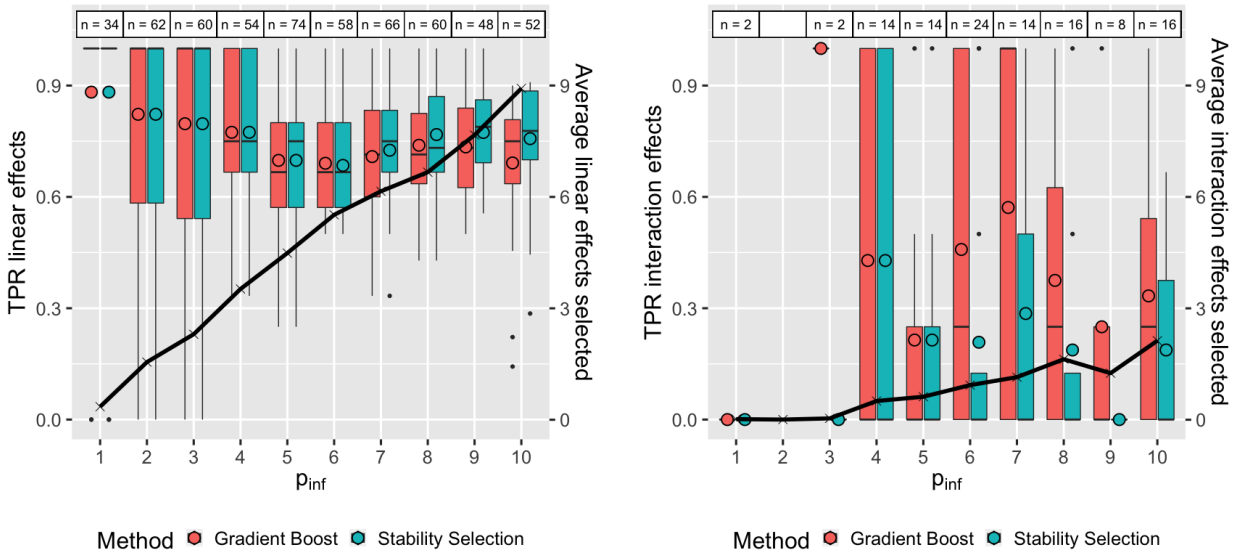
(b) TPR interaction covariates

Figure 6: Comparison of the gradient boosting and stability selection method, for every setting of p_{inf} with an equal number of selected covariates, for covariance matrix setting 3.

4.4 Sensitivity Analysis

Number of Observations n

To analyze the sensitivity to n, we both increase and decrease the number of observations. First, we decrease the number of observations from 209 to 105, almost half the number of covariates. We use the third setting of the covariance matrix, so the results are comparable to Figure 6. The number of simulation runs is 1000. The results are in Figure 7.



(a) TPR linear covariates

(b) TPR interaction covariates

Figure 7: Comparison of the gradient boosting and stability selection method, for every setting of p_{inf} with an equal number of selected covariates, for low number of observations n.

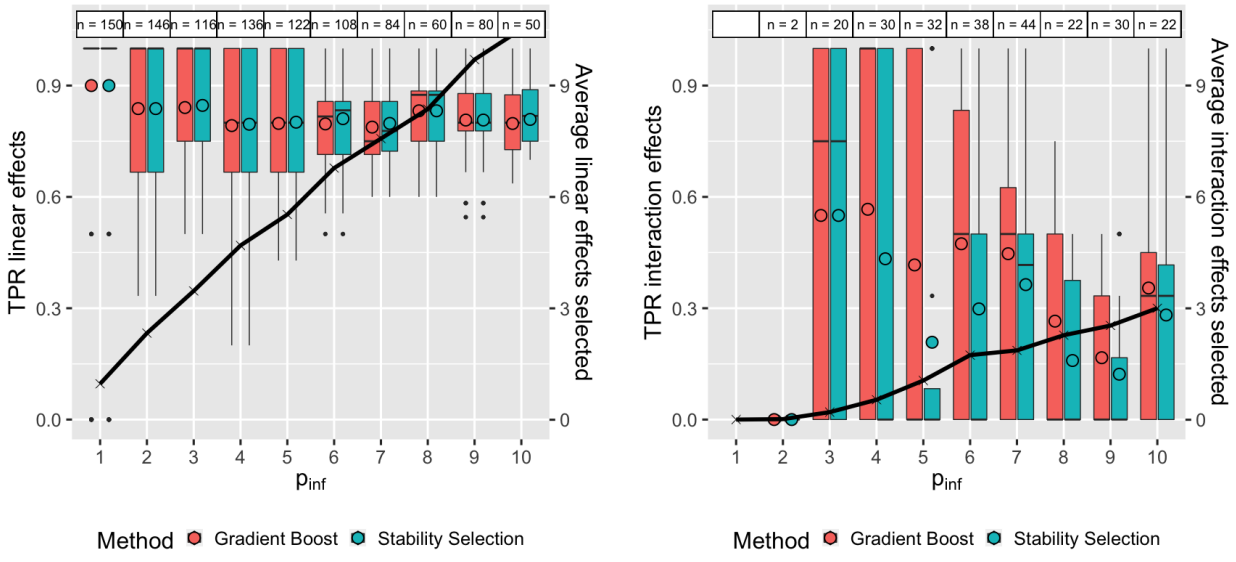
The number of selected linear covariates is slightly lower when compared to Figure 6a. Reducing the number of observations seems to magnify the differences in the TPRs of Figure 6. For the linear covariates and $p_{\text{inf}} > 6$, stability selection performs better than gradient boosting with the differences bigger now the number of observations is low. For the interaction terms, gradient boosting still performs better but again the differences in TPR are bigger than before. The TPRs are still not increasing in p_{inf}: both gradient boosting and stability selection have a harder time selecting the right linear covariates.

Next, we increase the number of observations to 420, double the number of covariates. We use the third setting of the covariance matrix, so the results are comparable with Figures 6 and 7. The results are in Figure 8.

The number of selected variables increases for both the linear and interaction terms, compared to the base case of Figure 6. The TPRs do not change much: they appear to be the same as for the regular n for the linear covariates, and also for the interaction terms the TPRs are not significantly higher. The number of selected variables however is higher, and thus the number of correctly selected variables also. Doubling the number of observations increases the performance of both methods, but the performance difference between the methods does not change.

Relative Selection Rate Threshold π_{thr}

Meinshausen and Bühlmann (2010) propose to use $\pi_{\text{thr}} \in (0.6, 0.9)$, while Hofner, Boccuto, and Göker (2015) state that any value $\in (0.5, 1)$ is potentially acceptable. To analyze the sensitivity with respect to the relative



(a) TPR linear covariates

(b) TPR interaction covariates

Figure 8: Comparison of the gradient boosting and stability selection method, for every setting of p_{inf} with an equal number of selected covariates, for high number of observations n.

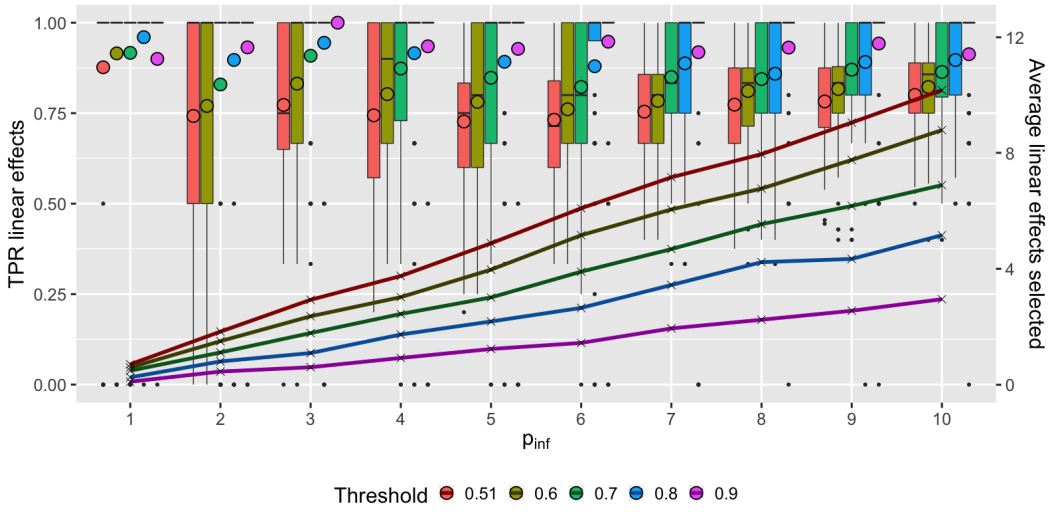
selection threshold, we let $\pi_{\text{thr}} \in \{0.51, 0.6, 0.7, 0.8, 0.9\}$, with covariance matrix setting 3 and 100 simulation runs per setting of p_{inf}. The results are in Figure 9.

For the linear covariates, The number of selected covariates is still almost linear in p_{inf}, just with a different slope for every different threshold. There is no clear pattern in the TPRs for every threshold, with $p_{\text{inf}} = 1$ producing a remarkable result where the highest threshold does not produce the highest average TPR. This means that the selection rate of an uninformative linear covariate is higher than that of the only informative linear covariate, which in turn means gradient boosting regularly selects an uninformative linear covariate before the informative. For $p_{\text{inf}} > 1$, a higher threshold results in a higher TPR on average. Until $p_{\text{inf}} = 6$, it seems advantageous to increase the threshold to improve the TPRs, but the difference in variables selected should be taken into account. Especially, for p_{inf} high, the TPRs differences are small but the difference in linear covariates selected is very large. Taking this into account we conclude that a lower threshold will select more informative variables.

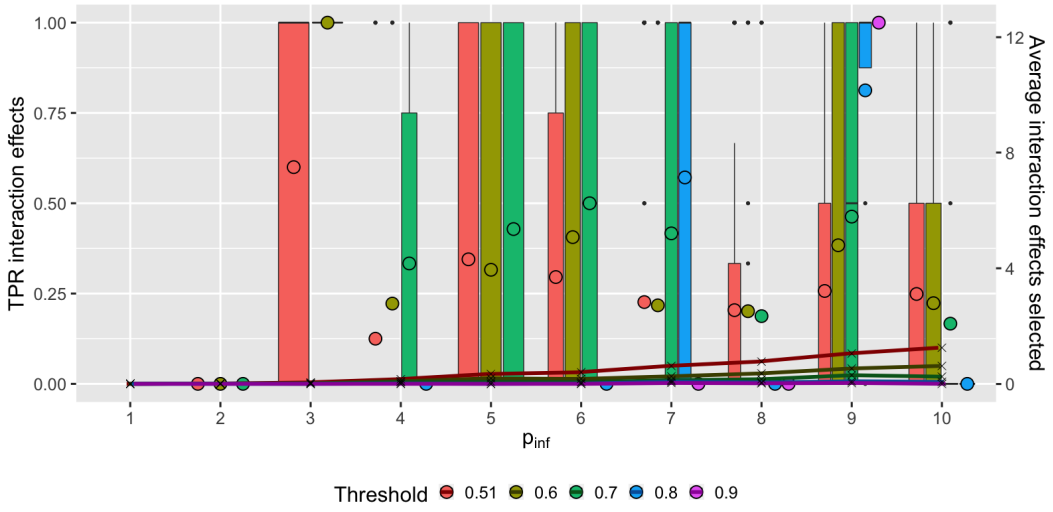
For the interaction terms, the highest threshold selects an interaction covariate in just one of the 1000 simulation runs, underlining the preference of gradient boosting to select the linear covariates. The results are unstable, sometimes higher thresholds return higher TPRs on average, but that is not the case for $p_{\text{inf}} = 7, 8$ and 10.

4.5 Conclusion

The hyperparameter settings lead to an average selected number of linear covariates that are very close to the real number of informative variables, for both the stability selection and the gradient boosting method. Cyclical gradient boosting does not outperform random guessing for both the linear and the interaction terms for the shape parameter, which underlines how difficult it is to model the shape parameter. The noncyclical version of gradient boosting as in the simulation therefore rarely selects covariates for the shape parameter and is incorrect when it does. The performance of both methods is very good for the scale parameter on the other hand, and not too sensitive to the number of observations. Without collinearity, both methods have TPRs close to perfect for the linear covariates.



(a) TPR linear covariates



(b) TPR interaction covariates

Figure 9: TPRs for different thresholds of the stability selection technique.

The informative interaction terms are very hard to identify resulting in low and unstable TPRs for both methods. Without collinearity, stability selection does not offer an improvement to gradient boosting. For the linear terms, it does offer a very small improvement when we increase collinearity and this remains valid for both a lower and higher number of observations. This also resembles the context of the empirical study the best, as Chmielewski et al. (2015) find lagged and unlagged correlations above 0.98 between stock indices around the world. For the stability selection threshold choice, the sensitivity analysis shows a higher threshold increases the TPR for linear covariates, but the average number of selected linear covariates decreases a lot. If we take that into account, models with a lower threshold select more informative linear covariates than with higher thresholds, despite the slightly lower TPRs. This could be an advantage in an empirical setting. For a relatively large number of (informative) variables, a low threshold seems advantageous.

5 Empirical Analysis

Using the simulation study results, we fit all models to empirical data.

5.1 Data Description

The negative returns, or losses, of the S&P 500 index make up the response variable. The time series is from September 1st, 2001 till December 31st, 2021 based on availability, resulting in 5113 daily observations excluding public holidays. Missing values due to e.g., public holidays are linearly interpolated. Figure 10 shows the plots of both the log losses and the excess log losses.

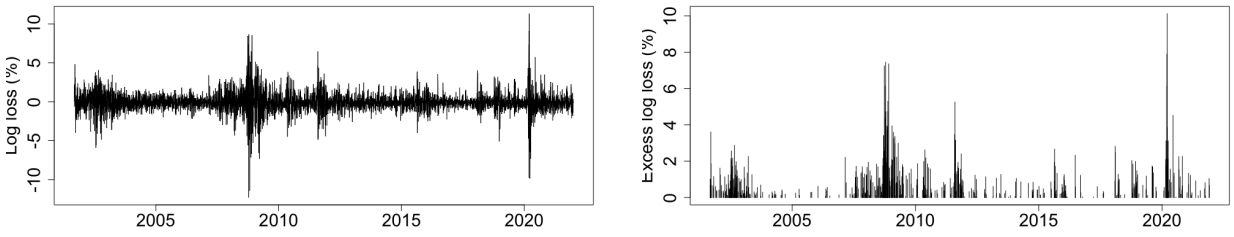


Figure 10: S&P 500 log losses and excess log losses.

The descriptive statistics of the losses are presented in Table 1.

Table 1: Descriptive statistics for the S&P 500 log losses and excess log losses in percentages.

	n	Mean	Median	SD	Min	Max	D1	D9	Skew.	Kurtosis
S&P 500 losses	5113	-0.044	-0.070	1.223	-12.307	11.319	-1.217	1.201	-0.065	12.318
S&P 500 excess losses	512	0.999	0.6133	1.207	0.001	10.118	0.106	2.191	3.064	13.411

The covariates consist of the Russel 2000 index and 12 stock indices of the largest trading partners of the United States, together with the corresponding 8 exchange rates to the U.S. dollar. Other covariates are the commodities gold, silver, crude oil, copper, platinum, and natural gas. Also included are the risk factors of the Fama-French 5-factor model (Fama and French, 1993), except for the market return, together with the corresponding risk-free rate. All covariates are log returns. Furthermore, the market yield on the U.S. Treasury securities at 3-month, 6-month, 1, 3, 5, 10, and 20-year constant maturity and the Effective Federal Funds Rate (EFFR) are included, in log percentages, as well as the log percentage changes of the

CBOE Volatility Index. We also consider all interaction terms, resulting in a total of $(41)(41+1)/2 = 861$ covariates. All descriptive statistics are in Table 2.

Table 2: Descriptive statistics for covariates, in percentages.

	Mean	SD	Min	Max	Skewness	Kurtosis	JB-test (p-value)
Indices							
FTSE 100	0.007	1.172	-11.512	9.384	-0.312	8.785	16524 (0.000)
Russell 2000	0.031	1.541	-15.399	8.976	-0.592	7.973	13842 (0.000)
AEX	0.009	1.387	-11.376	10.028	-0.173	7.668	12551 (0.000)
DAX	0.023	1.453	-13.055	10.797	-0.125	6.222	8260 (0.000)
GSPTSE	0.022	1.073	-13.176	11.295	-1.008	20.588	91171 (0.000)
SSE	0.026	1.531	-9.256	9.401	-0.419	5.091	5671 (0.000)
CAC 40	0.010	1.418	-13.098	10.595	-0.164	6.989	10429 (0.000)
Nikkei 225	0.015	1.444	-12.111	13.235	-0.430	6.918	10353 (0.000)
VIX	-0.011	7.235	-35.059	76.825	1.046	6.613	10249 (0.000)
KOSPI	0.029	1.313	-11.172	11.284	-0.434	6.562	9334 (0.000)
IPC Mexico	0.041	1.187	-7.266	10.441	-0.029	5.762	7075 (0.000)
BSE Sensex	0.062	1.391	-14.102	15.990	-0.308	11.176	26690 (0.000)
FTSE MIB	0.000	1.530	-18.546	10.874	-0.582	9.578	19831 (0.000)
HSI	0.018	1.404	-13.582	13.407	-0.060	8.492	15365 (0.000)
Currency Rates							
USDEUR	0.005	0.567	-3.003	4.621	0.112	3.046	1988 (0.000)
INRUSD	0.006	0.433	-3.756	3.792	-0.044	9.304	18442 (0.000)
CNYUSD	-0.005	0.156	-2.019	1.816	0.233	20.381	88539 (0.000)
JPYUSD	-0.001	0.597	-5.216	3.343	-0.331	5.028	5479 (0.000)
KRWUSD	-0.004	0.655	-13.222	10.135	-0.501	53.672	613915 (0.000)
MXNUSD	0.015	0.706	-5.960	8.114	0.827	11.465	28587 (0.000)
USDGBP	0.001	0.581	-8.169	4.435	-0.611	11.213	27102 (0.000)
CADUSD	-0.005	0.556	-5.072	3.807	-0.092	5.644	6793 (0.000)
Commodities							
Gold	0.035	1.111	-9.821	8.643	-0.369	5.466	6481 (0.000)
Silver	0.031	2.019	-19.546	12.196	-0.911	7.364	12259 (0.000)
Crude Oil	0.029	2.618	-28.221	31.963	0.039	18.830	75542 (0.000)
Natural Gas	0.007	3.334	-19.899	32.375	0.601	5.825	7537 (0.000)
Copper	0.035	1.714	-11.693	11.769	-0.203	4.277	3932 (0.000)
Platinum	0.013	1.365	-12.347	11.176	-0.580	6.380	8957 (0.000)
Security Market Yields							
U.S. 3MO	1.211	1.440	0.000	5.060	1.279	0.680	1493 (0.000)
U.S. 6MO	1.313	1.472	0.020	5.193	1.229	0.522	1346 (0.000)
U.S. 1Y	1.415	1.446	0.040	5.164	1.104	0.213	1047 (0.000)
U.S. 3Y	1.867	1.329	0.100	5.126	0.731	-0.527	515 (0.000)
U.S. 5Y	2.308	1.250	0.190	5.098	0.425	-0.856	310 (0.000)
U.S. 10Y	2.972	1.157	0.519	5.297	0.044	-1.049	236 (0.000)
U.S. 20Y	3.511	1.197	0.866	5.874	-0.040	-1.190	303 (0.000)
Interest Rates							
EFFR	1.297	1.506	0.040	5.269	1.306	0.706	1561 (0.000)
Fama-French Factors							
SMB	0.006	0.614	-4.625	5.572	0.107	4.317	3980 (0.000)
HML	-0.005	0.730	-5.087	6.532	0.352	9.532	19463 (0.000)
RMW	0.015	0.451	-2.758	3.218	0.078	2.748	1614 (0.000)
CMA	0.004	0.343	-2.634	2.430	-0.053	3.720	2950 (0.000)
RF	0.005	0.006	0.000	0.022	1.301	0.791	1575 (0.000)

All stock index and commodities log returns have a positive mean and thus a positive return over the sample period. The FTSE MIB is the worst performing stock index. The VIX has the highest standard deviation by some margin, which is not unexpected as it is an index for volatility itself. Furthermore, all covariates are not normally distributed as indicated by very significant p-values on the Jarque-Bera test. The simulation showed that collinearity affects the performance of variable selection for gradient boosting, and therefore a correlation plot is given in Figure A.1. To aid the comparison of the stock indices, we use the log returns of the S&P 500, instead of the log losses, in the correlation plot.

The stock indices show positive intercorrelation, with some strongly correlated like the German DAX index with the Dutch AEX index and the American S&P 500 index with the Russel 2000 index. The same is true for the commodities and Fama-French risk factors small minus big (SMB) and high minus low (HML). The indices, commodities, and risk factors all are negatively correlated with the VIX, and the stock indices all have a positive but weak correlation with the commodities, SMB and HML. Conversely, there is a weak negative correlation between the exchange rates to the US dollar and the stock indices, and a weak positive correlation between the VIX and those exchange rates. The profitability risk factor (RMW) has similar correlations to the exchange rates. The market yield on U.S. Treasury securities with constant maturity and the risk-free rate (RF) logically have strong positive intercorrelations. However, they are uncorrelated to almost all other covariates.

5.2 Threshold choice

The Peak-Over-Threshold method states that the excesses of i.i.d. variables over a high threshold u approximately follow the GPD with distribution function $G_{\xi,\sigma}$. The choice of the threshold u implies a bias-variance tradeoff. A low threshold is likely to violate the assumptions of the asymptotics, increasing the bias. Conversely, a high threshold will lead to fewer available observations and thus a higher variance for the fitted model. The mean excess function $e(u)$ as described by Embrechts, Mikosch, and Klüppelberg (1997) can be plotted to make a substantiated choice for the threshold. For a random variable X with distribution function $G_{\xi,\sigma}$,

$$e(u) = E\left(X - u | X > u\right) = \frac{\sigma + \xi u}{1 - \xi}, \quad \xi < 1, \tag{24}$$

so $e(u)$ is linear in threshold u. The empirical mean excess function given sample $X_1, ..., X_n$ is defined as

$$e_n(u) = \frac{1}{N_u} \sum_{i \in \Delta_n(u)} (X_i - u), \quad u > 0, \tag{25}$$

where $N_u = \text{card}\{i : i = 1, ..., n, X_i > u\} = \text{card}\Delta_n(u)$. Based on the linearity of the mean excess function, u should be chosen such that $e_n(u)$ is approximately linear for $x \geq u$. The empirical mean excess function for the S&P 500 log losses is plotted in Figure 11.

The full empirical mean excess function appears to be somewhat linear, and most threshold choices seem to be fine. The 80% and 90% quantiles are indicated as blue lines, which results in 1023 or 512 threshold exceedances of the losses respectively. A higher threshold is less likely to violate the assumptions of the threshold, and Section 4.4 showed that the variable selection performance of the machine learning models is not too sensitive to a low number of observations. Therefore, we set $u = 1.2\%$, such that $P(X \leq u) = 0.9$.

5.3 Hyperparameter choice

Before we fit all the models, the hyperparameters of the gradient boosting and stability selection need to be set. For gradient boosting, the optimal number of boosting iterations m_{stop} will be determined based on cross-validated empirical predictive risk based on 25 folds as proposed by Mayr, Hofner, and Schmid (2012a), with the results in Figure 12.

The optimal $m_{\text{stop}} = 1491$, which results in model GB_{rich}, with 74 selected covariates for the scale parameter and an intercept for both the shape and scale parameter. Producing rich models despite early

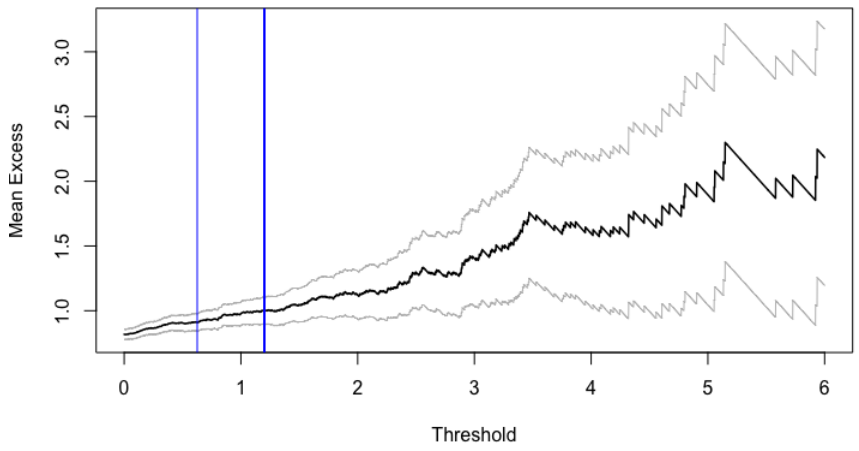


Figure 11: Mean excess plot for S&P 500 losses. The blue lines indicate the 80% and 90% quantiles.

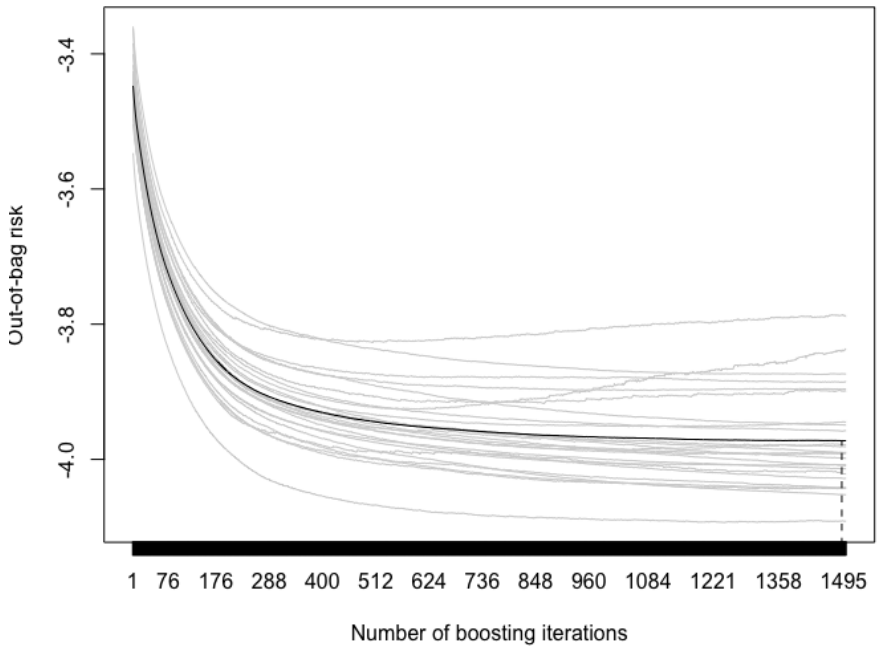


Figure 12: Out-of-bag risk of 25-fold cross-validation, optimal number of iterations indicated.

stopping and regularization is a known problem for boosting (Mayr, Hofner, and Schmid, 2012a; Meinshausen and Bühlmann, 2010). Therefore, based on Figure 12, we also set $m_{\text{stop}} = 400$ as the average out-of-bag predictive risk does not increase much compared to $m_{\text{stop}} = 1491$, but the number of selected covariates decreases to 18 for the scale parameter. This model is indicated as $\text{GB}_{\text{sparse}}$.

For stability selection, we set the number of selected covariates per run $q = 20$, such that in theory all 18 covariates and the two intercepts selected by the $\text{GB}_{\text{sparse}}$ model can be selected in every run, following Meinshausen and Bühlmann (2010). Together with relative selection frequency threshold $\pi_{\text{thr}} = 0.51$, stability selection selects 7 covariates for the scale parameter and an intercept for both the shape and the scale parameter. Stability selection only performs variable selection, so a model still needs to be fitted. To prevent overfitting we fit a gradient boosting model, but with just the selected stable covariates and with optimal $m_{\text{stop}} = 1997$, again based on cross-validated empirical predictive risk. This model is indicated as SS_{opt}. We also fully fit a regular GAMLSS model with just the 7 selected stable covariates, indicated as model SS_{full}, to investigate the effect of regularization. This results in four models that can be regarded as machine learning models. For the models using PCR in GAMLSS, no hyperparameters need to be set. The PCR model using GAIC selects the first 8 principal components to minimize the GAIC, while the t-VAL model selects 130 PCs with significant t-values. The characteristics of all models are summarized in Table B.1 and the selected covariates by the four machine learning models are presented in Table B.2.

5.4 Tail Risk and Performance Measures

Value-At-Risk

Because the GPD for the excess losses is used to estimate the tail of the underlying distribution F of the log losses, the VaR is estimated as the quantile of the fitted GPD distribution. Using the notation of McNeil, Frey, and Embrechts (2005), for $x \geq u$,

$$\bar{F}(x) = \bar{F}(u)P(X > x|X > u) = \bar{F}(u)\bar{H}_{\sigma,\xi}(x-u), \tag{26}$$

where $\bar{F}(u)$ is the threshold exceedance probability, and $H_{\sigma,\xi}$ the distribution function of the GPD as in Equation 1. If we know $\bar{F}(u)$, this formula can be inverted to obtain the VaR for confidence level $\alpha \geq F(u)$, equal to

$$\text{VaR}_\alpha = q_\alpha(F) = u + \frac{\sigma}{\xi}\left(\left(\frac{1-\alpha}{\bar{F}(u)}\right)^{-\xi} - 1\right). \tag{27}$$

with $q_\alpha(F) = F^{\leftarrow}(\alpha)$ the quantile function of F and u, ξ and σ the threshold, and shape and scale parameter respectively of the GPD. We can estimate this using the GPD parameters estimated by the model.

Performance measures

To measure the performance of the $\widehat{\text{VaR}}_\alpha$, we define the hit function as

$$I_i(\alpha) = \begin{cases} 1 & \text{if } x_i > \widehat{\text{VaR}}_{\alpha,i}, \\ 0 & \text{if } x_i \leq \widehat{\text{VaR}}_{\alpha,i}, \end{cases} \tag{28}$$

with x_i the realized log loss i. The number of violations, $I(\alpha) = \sum_{i=1}^{n} I_i(\alpha)$ can be compared to the empirical number of violations, $(1-\alpha)n$, to give an indication if the model has a tendency to over or underestimate. To evaluate the accuracy, we use two backtests to see if the hit function satisfies two properties. The first property, the unconditional coverage property, is tested by the Kupiec (1995) proportion of failures test, with the test statistic equal to

$$POF = 2\ln\left[\left(\frac{1-\hat{\alpha}}{1-\alpha}\right)^{n-I(\alpha)}\left(\frac{\hat{\alpha}}{\alpha}\right)^{I(\alpha)}\right] \sim \chi^2(1), \tag{29}$$

where $\hat{\alpha} = \frac{1}{n} I(\alpha)$. This restricts the number of allowed violations and indicates if the number of violations is significantly too high or low. The second, independence property restricts the timing of these violations. Consecutive realizations of the hit function should be independent, tested by the Christoffersen (1998) test for independence, which first calculates the probabilities of a violation conditional on the previous realization of the hit function,

$$\begin{aligned} N_{kj} &= \sum_{i=2}^{n} \mathbb{1}_k(I_{i-1}(\alpha)) \mathbb{1}_j(I_i(\alpha)) && \text{for } k, j = 0, 1, \\ \pi_k &= \frac{N_{k1}}{N_{k0} + N_{k1}} && \text{for } k = 0, 1, \\ \pi &= \frac{N_{01} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}, \end{aligned}$$

with $\mathbb{1}$ the indicator function. Under the null hypothesis, the probability of a violation should be independent of the hit function the day before, i.e. $\pi_0 = \pi_1$, resulting in the test statistic

$$LR = -2 \ln \left[(1-\pi)^{N_{00}+N_{10}} \pi^{N_{01}+N_{11}} \right] + 2 \ln \left[(1-\pi_0)^{N_{00}} \pi_0^{N_{01}} (1-\pi_1)^{N_{10}} \pi_1^{N_{11}} \right] \sim \chi^2(1). \tag{30}$$

Both backtests treat all data as binary, discarding relevant information. To measure the performance of the estimated VaR using all information, we calculate the quantile loss function as used in quantile regression,

$$L(\widehat{\text{VaR}}_{\alpha,i}, x_i) = \max \left[\alpha(x_i - \widehat{\text{VaR}}_{\alpha,i}), (\alpha - 1)(x_i - \widehat{\text{VaR}}_{\alpha,i}) \right], \tag{31}$$

with again x_i the realized log loss i. We use this loss function to calculate the average quantile loss (AQL). As Section 5.2 sets $u = 1.2\%$ such that $P(X \leq u) = 0.9$, we have the unconditional threshold exceedance probability $\bar{F}(u) = 0.1$. For all models, the estimated VaRs that exceed the highest empirical loss are set equal to that loss.

5.5 Results

In this section, we evaluate the performance of the different models. We look at the violations, AQL and the backtests and compare them to a benchmark model. First Section 5.5.1 presents the results for the subset of excess losses, while Section 5.5.2 looks at all losses and fits a model which also predicts exceedances.

5.5.1 Performance of VaR Estimation For Excess Log Losses

We start off by fitting the stability selection, gradient boosting, and PCR models to the excess log losses in Figure 10. Because the excess losses exceed the threshold by construction, we have $u = 0$ and hence $\bar{F}(u) = 1$, simplifying the VaR of Equation 27.

Benchmark Model

To compare the models, we also set up a benchmark model. This benchmark model will be a fully fitted regular GAMLSS model for the GPD distribution. Because the regular GAMLSS models can not fit high-dimensional data, we only use the linear terms and not the interaction terms of the variables as the covariates. Therefore, besides acting as a benchmark, comparing with this model sheds light on the added value of the interaction terms, as the GAIC and t-VAL models are also fully fitted GAMLSS models, but with both the linear and interaction terms of the variables as the covariates.

Performance

The in-sample results of the VaR estimation of all models for the excess log losses and the benchmark are in Table 3. The Christoffersen test is not conducted as the subset of excess log losses is not a valid time series.

Table 3: In-sample VaR estimation performance of all models and the benchmark model for excess losses.

	GB_{sparse}	GB_{rich}	SS_{opt}	SS_{full}	t-VAL	GAIC	Benchmark	Empirical
$\widehat{\text{VaR}}_{95\%}$								
Violations	**342**	374	351	353	393	471	359	256
Kupiec	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	-
AQL	0.219	0.188	0.231	0.233	**0.171**	0.189	0.221	-
$\widehat{\text{VaR}}_{97.5\%}$								
Violations	68	74	123	**131**	90	6	121	128
Kupiec	(0.000)	(0.000)	(0.682)	**(0.684)**	(0.000)	(0.000)	(0.538)	-
AQL	0.127	0.120	0.138	0.138	0.149	**0.063**	0.137	-
$\widehat{\text{VaR}}_{99\%}$								
Violations	0	0	0	0	**1**	0	0	51
Kupiec	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	-
AQL	0.127	0.119	0.106	0.103	0.130	**0.101**	0.106	-

At a confidence level of 95%, all models significantly underestimate the VaR, resulting in many violations and Kupiec backtest p-values close to zero. The PCR models produce relatively low AQLs, while the gradient boosting models outperform the benchmark model which in turn outperforms the stability selection models.

At a confidence level of 97.5%, the stability selection models and the benchmark model produce similar violations and do not significantly over- or underestimate the VaR, while the PCR models and gradient boosting models do overestimate. The former models also produce similar AQLs and are outperformed by the gradient boosting models. The GAIC model has the lowest AQL by some margin, but the extremely low number of violations indicates overfitting.

At the highest confidence level, all models significantly overestimate the $\text{VaR}_{99\%}$ which results in just one violation. The GAIC model again produces the lowest AQL, while the machine learning models (GB_{sparse}, GB_{rich}, SS_{opt}, SS_{full}) collectively do not outperform the benchmark. Between those models, the stability selection models produce lower AQL, but the reverse is true for the lower confidence levels of 95% and 97.5%.

The rich gradient boosting model performs similarly to or better than the sparse model on all confidence levels for both the AQL and the Kupiec backtest, an indication that the performance is sensitive to the choice of m_{stop} for gradient boosting. The performance of the fully fitted and shrinked (optimal) coefficients stability selection models SS_{opt} and SS_{full} is very similar. Once the stable covariates are selected, shrinking does not add value to in-sample estimation.

Out-Of-Sample Performance

We also evaluate the out-of-sample performance to detect overfitting. The data consist of 103 observations from January 3rd, 2022 till May 31st 2022, containing 26 excess log losses of the S&P 500 above the threshold $u = 1.2\%$, a considerably higher rate than the in-sample rate of 0.1. The mean and standard deviation of the excess losses are 0.863% and 0.807% respectively, lower than the in-sample mean and standard deviation of 0.999% and 1.207% reported in Table 1. The minimum and maximum excess losses are 0.006% and 2.759% respectively. The out-of-sample performance of the risk measure estimation of all models, the benchmark, and the empirical violations are presented in Table 4.

The performance in terms of violations is fairly better than the in-sample performance for most models. The estimated VaR_α of all the machine learning models only produces significant Kupiec p-values at the

Table 4: Out-of-sample VaR estimation performance of all models and the benchmark model for excess losses.

	GB_{sparse}	GB_{rich}	SS_{opt}	SS_{full}	t-VAL	GAIC	Benchmark	Empirical
$\widehat{VaR}_{95\%}$								
Violations	19	21	19	19	**13**	16	19	13
Kupiec	(0.016)	(0.001)	(0.016)	(0.016)	**(0.695)**	(0.237)	(0.016)	-
AQL	0.228	**0.178**	0.207	0.199	0.368	0.283	0.187	-
$\widehat{VaR}_{97.5\%}$								
Violations	**6**	**6**	**6**	**6**	9	11	5	7
Kupiec	**(0.822)**	**(0.822)**	**(0.822)**	**(0.822)**	(0.131)	(0.054)	(0.819)	-
AQL	0.087	0.079	0.075	**0.070**	0.339	0.284	**0.070**	-
$\widehat{VaR}_{99\%}$								
Violations	0	0	0	0	8	**5**	0	3
Kupiec	**(0.237)**	**(0.237)**	**(0.237)**	**(0.237)**	(0.001)	(0.158)	**(0.237)**	-
AQL	0.072	0.060	**0.056**	**0.056**	0.288	0.201	0.058	-

95% confidence level, compared to significant p-values and thus over and underestimation at all confidence levels for the gradient boosting and at the 95% and 99% confidence levels for stability selection in-sample. They also produce lower AQL at all confidence levels compared to the in-sample result, except GB_{sparse} at 95% confidence, which is just slightly higher. While in-sample at all confidence levels one of the PCR models produces the lowest AQL, the reverse is true out-of-sample, and this relative outperformance by the machine learning models and the benchmark increases for higher confidence levels. Although violations for the PCR models are only significantly too high for the t-VAL model at the 99% confidence level, the deteriorated out-of-sample AQLs indicate overfitting by the PCR models. Between those models, the t-VAL model seems to overfit the heaviest, evidenced by a higher AQL at all confidence levels.

The machine learning models and the benchmark produce almost equal violations, only significantly underestimating the $VaR_{95\%}$. In terms of AQL none of the machine learning models outperforms the benchmark on all confidence levels. Between the machine learning models, the stability selection models produce lower AQLs than the gradient boosting models for the highest confidence levels. Between the stability selection models, SS_{full} just slightly outperforms SS_{opt} on the out-of-sample AQL, while there was no outperformance in-sample. This confirms that if variable selection leads to a sparse model, shrinking does not add value to the estimation.

However, GB_{rich} outperforms GB_{sparse} on the AQL both in-sample and out-of-sample for all confidence levels. Gradient boosting models thus are sensitive to m_{stop} even for small differences in the out-of-bag CV risk. As there is no outperformance by the gradient boosting models on the Kupiec backtest, the GB_{rich} model seems preferable.

As the machine learning models do not consistently outperform the benchmark on the out-of-sample VaR estimation, the overfitting problem of the benchmark is not tackled. In-sample, the GAIC model outperforms the benchmark on AQL, but the out-of-sample surge of AQL of the GAIC and t-VAL model prove these models increase overfitting rather than tackle it. These results also question the added value of the interaction terms in the models, which will be investigated later.

5.5.2 Performance of VaR Estimation For Log Losses

In reality, the log losses that exceed threshold u are unknown in advance, so we need to estimate the threshold exceedance probability $\bar{F}(u)$. We model this dynamically using covariates too. Embrechts, Mikosch, and Klüppelberg (1997) suggest that for an i.i.d. sample, the excesses over a high threshold can be modeled by the GPD, where the number of exceedances independently follows a Poisson process, $N_t \sim Pois(\lambda t)$. The

time between exceedances thus follows the exponential distribution. The threshold exceedance probability $\bar{F}(u)$ is then given by $1 - \exp(-\lambda)$, with λ the intensity parameter of the exponential distribution. We dynamically model the intensity parameter of the exponential distribution with the previously mentioned covariates and their interaction terms using gradient boosting with $m_{\text{stop}} = 1995$, which minimizes the out-of-bag risk of 25-fold cross-validation. The model selects 4 linear and 11 interaction terms, specified in Tables B.1 and B.2. The fitted $\hat{\lambda}$ give estimated threshold exceedance probabilities. The estimated VaR_α is only valid for $\alpha \geq F(u)$ or equivalently $\bar{F}(u) \geq 1 - \alpha$. Therefore if the estimated threshold exceedance probability, $1 - \exp(-\hat{\lambda}) \geq 1 - \alpha$, it will produce a valid $\widehat{\text{VaR}}_\alpha$, and it thus predicts an exceedance. For every $1 - \exp(-\hat{\lambda}) < 1 - \alpha$, it predicts no exceedance. The in-sample and out-of-sample true negatives, true positives, false negatives, and false positives (TN, TP, FN, FP) and the performance of the prediction of exceedances of this gradient boosting model are reported in Table 5.

Table 5: Prediction performance gradient boosting exponential model.

	TN	TP	FN	FP	Accuracy	Sensitivity	Specificity
In-sample							
$\alpha = 95\%$	4549	496	16	52	0.987	0.989	0.969
$\alpha = 97.5\%$	4537	498	14	64	0.985	0.986	0.973
$\alpha = 99\%$	4517	505	7	84	0.982	0.982	0.986
Out-of-sample							
$\alpha = 95\%$	77	24	2	0	0.981	1.000	0.923
$\alpha = 97.5\%$	77	26	0	0	1.000	1.000	1.000
$\alpha = 99\%$	76	26	0	1	0.990	0.987	1.000

The fitted prediction model has high accuracy, sensitivity, and specificity in-sample, and performs even better out-of-sample for the two highest confidence levels. The unconditional expectation of the conditional exceedance probabilities is equal to 9.39% in-sample and 21.46% out-of-sample, compared to the exceedance rates of 10% in-sample and 25.42% out-of-sample respectively. For the 16 false negatives for $\alpha = 95\%$, the highest realized log loss which was predicted no threshold exceedance equals 1.39% or 0.19% above the threshold u in-sample, and 1.22% or 0.02% above threshold u for 2 false negatives out-of-sample. The model predicts an exceedance if $1 - \exp(-\hat{\lambda}) \geq 1 - \alpha$, an undesirable feature as the number of predicted exceedances increases in α. Solving this means a trade-off between specificity and sensitivity as more predicted exceedances lead to fewer false negatives but more false positives. From a risk management perspective, specificity is arguably more significant, but this is not investigated in this paper as the main focus is on dynamically modeling the GPD distribution.

For the predicted threshold exceedances, the VaR_α is now equal to

$$\text{VaR}_\alpha = u + \frac{\sigma}{\xi}\left(\left(\frac{1-\alpha}{1-e^{-\lambda}}\right)^{-\xi} - 1\right), \tag{32}$$

which can be estimated using the fitted GPD parameters $\hat{\xi}$, $\hat{\sigma}$ by the different models together with the fitted $\hat{\lambda}$. The threshold $u = 1.2\%$. The in-sample performance of the estimated VaR for all models using the conditional exceedance probability is reported in Table 6, for the predicted exceedances.

The model estimates a VaR_α for the predicted threshold exceedances instead of the realized excess losses which are a priori unknown. The threshold exceedance probability was equal to $\bar{F}(u) = 1$ for excess losses but is now estimated as $\bar{F}(u) = 1 - \exp(-\hat{\lambda})$, decreasing the $\widehat{\text{VaR}}_\alpha$ of each predicted excess loss if $1 - \exp(-\hat{\lambda}) < 1$. This increases the number of violations of the $\widehat{\text{VaR}}_\alpha$ at all confidence levels, while the false negatives decrease the number of $\widehat{\text{VaR}}_\alpha$ violations. This results finally in more $\widehat{\text{VaR}}_\alpha$ violations for the benchmark and machine learning models.

In comparison to Table 3 the AQL is lower for all models at all levels but not comparable as we now estimate for the log losses instead of the subset of excess log losses. The performance of the GAIC model

Table 6: In-sample performance of the VaR estimation of all models and the benchmark model for the log losses.

	GB_{sparse}	GB_{rich}	SS_{opt}	SS_{full}	t-VAL	GAIC	Benchmark	Empirical
$\widehat{\text{VaR}}_{95\%}$								
Violations	400	427	409	412	79	**274**	415	256
Kupiec	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	**(0.244)**	(0.000)	-
Christoffersen	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	**(0.191)**	(0.000)	-
AQL	0.100	0.098	0.099	0.099	0.112	**0.096**	0.099	-
$\widehat{\text{VaR}}_{97.5\%}$								
Violations	116	**123**	171	180	63	119	172	128
Kupiec	(0.282)	**(0.664)**	(0.000)	(0.000)	(0.000)	(0.424)	(0.000)	-
Christoffersen	(0.040)	**(0.757)**	(0.000)	(0.000)	(0.000)	(0.338)	(0.001)	-
AQL	0.038	**0.037**	0.039	0.040	0.067	0.046	0.039	-
$\widehat{\text{VaR}}_{99\%}$								
Violations	32	34	34	34	44	**54**	34	51
Kupiec	(0.004)	(0.010)	(0.010)	(0.010)	(0.305)	**(0.689)**	(0.010)	-
Christoffersen	(0.013)	(0.030)	(0.030)	(0.030)	(0.412)	**(0.806)**	(0.030)	-
AQL	**0.015**	**0.015**	**0.015**	**0.015**	0.037	0.019	**0.015**	-

stands out, with no significant over- or underestimation of the VaR_α and dependence in the violations at all confidence levels according to the Kupiec and Christoffersen backtests.

The Christoffersen backtest for independence yields similar results as the Kupiec backtest in terms of significance. Only the $\widehat{\text{VaR}}_{97.5\%}$ of GB_{sparse} produces no significant overestimation according to the Kupiec backtest, but the violations are significantly dependent. We investigate this visually with the plot of the predicted and actual exceedances, and $\widehat{\text{VaR}}_{97.5\%}$ violations of the GB_{sparse} model in Figure 13.

It looks like the low log excess losses are overrepresented in the violations, but the mean of the violations is almost equal to the mean of all excess losses. The median of the violations is even higher than the median of all excess losses, so this does not explain the significant dependence. It can be explained by the characteristics of the empirical data itself, as the Ljung-Box test with lag 1 for the absolute log losses ($p < 0.001$) and squared log losses ($p < 0.001$) indicate volatility clustering. This means the realized excess log losses, and consequently, the predicted excess log losses because of high prediction accuracy, are clustered, and so are the violations.

In Table 3 one of the PCR models, GAIC and t-VAL, produced the lowest AQL at every confidence level, but when combined with the conditional threshold exceedance probability they now are outperformed by the other models and the outperformance increases for higher confidence levels. This is another indication of overfitting, especially for the t-VAL model which is much worse than the GAIC model.

The differences between the AQL of the stability selection and gradient boosting models in Table 3 seem to disappear, with all models producing very similar results without consistent outperformance. Only the difference in violations of the $\widehat{\text{VaR}}_{97.5\%}$ is still large, with significantly too many violations for the stability selection model. GB_{rich} performs at least equal to the benchmark on all measures at the two highest confidence levels, while there is no outperformance of the benchmark for the other models.

Between the gradient boosting models, GB_{rich} not only outperforms GB_{sparse} on AQL like in Table 3, but also performs equal or better on the backtests. The conditional threshold exceedance probability does not change much for the stability selection models. Both the full and optimal model perform similarly, with no outperformance between them.

Finally, we look at the out-of-sample performance of the models, combined with the conditional threshold exceedance probability, reported in Table 7.
Just like before the number of violations increases for the benchmark and the machine learning models, which results in underestimation of the $\text{VaR}_{97.5\%}$ compared to the Table 4, although not significantly. However,

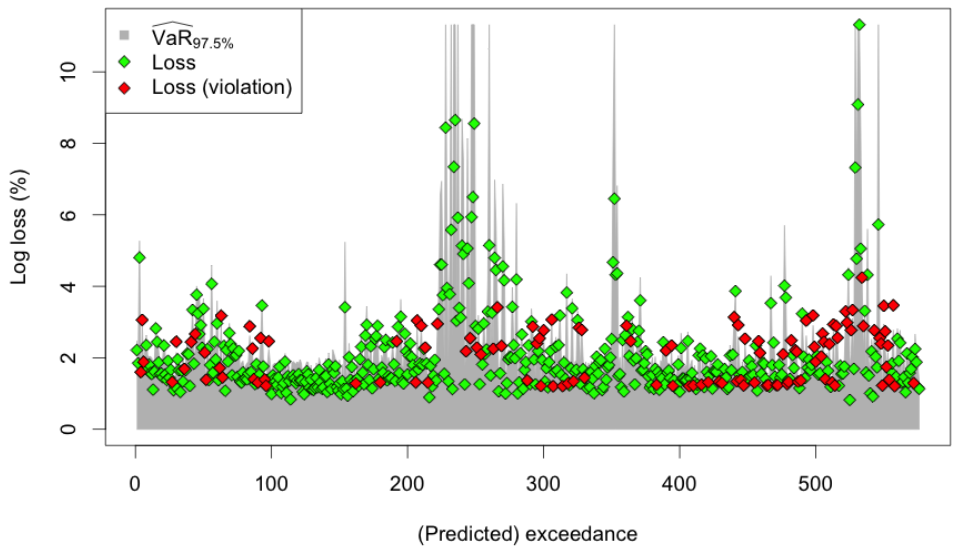


Figure 13: $\widehat{\text{VaR}}_{99\%}$ of the SS_{full} model with realized log losses.

Table 7: Out-of-sample performance of the VaR estimation of all models and the benchmark model for the log losses.

	$\text{GB}_{\text{sparse}}$	GB_{rich}	SS_{opt}	SS_{full}	t-VAL	GAIC	Benchmark	Empirical
$\widehat{\text{VaR}}_{95\%}$								
Violations	23	24	23	22	**9**	18	23	13
Kupiec	(0.007)	(0.003)	(0.007)	(0.014)	**(0.212)**	(0.157)	(0.007)	-
Christoffersen	(0.001)	(0.001)	(0.001)	(0.002)	**(0.446)**	(0.094)	(0.004)	-
AQL	0.259	**0.239**	0.252	0.248	0.315	0.272	0.242	-
$\widehat{\text{VaR}}_{97.5\%}$								
Violations	10	9	10	10	**7**	13	8	7
Kupiec	(0.187)	(0.337)	(0.187)	(0.187)	**(0.841)**	(0.019)	(0.557)	-
Christoffersen	(0.141)	**(0.612)**	(0.418)	(0.418)	(0.585)	(0.010)	(0.425)	-
AQL	0.101	0.102	0.099	0.098	0.188	0.152	**0.097**	-
$\widehat{\text{VaR}}_{99\%}$								
Violations	**3**	**3**	**3**	**3**	5	8	**3**	3
Kupiec	**(0.806)**	**(0.806)**	**(0.806)**	**(0.806)**	(0.180)	(0.006)	**(0.806)**	-
Christoffersen	**(0.886)**	**(0.886)**	**(0.886)**	**(0.886)**	(0.315)	(0.012)	**(0.886)**	-
AQL	0.043	0.044	**0.042**	**0.042**	0.089	0.069	0.043	-

the number of violations of the $\widehat{\text{VaR}}_{99\%}$ are now very close to the empirical value. The PCR models again perform the worst based on the AQL, with the t-VAL model estimating extremely poor.

The differences between the benchmark and machine learning models AQLs are a lot smaller compared to Table 4, and the violations are still very similar for those models. All those violations are not significantly too high, too low, or dependent for the 97.5% and 99% confidence levels and still none of the models outperforms the benchmark on all confidence levels, just like in Table 4.

Just like with unconditional threshold exceedance probabilities, the stability selection models produce lower AQL for the higher confidence levels compared to the gradient boosting models. Between the stability selection models, SS_{full} again produces equal or lower AQL, but the differences are still very small.

Between the gradient boosting models, GB_{rich} outperformed GB_{sparse} on AQL on all confidence levels out-of-sample, but with conditional exceedance probabilities this advantage does not hold. The AQLs for both models are also much closer at all confidence levels.

5.6 Sensitivity with respect to relative selection frequency threshold

In Section 4.4 we evaluated the sensitivity of stability selection with respect to π_{thr} by comparing the TPR for different thresholds. Empirically, we evaluate this sensitivity by looking at the performance of the SS_{full} model for different π_{thr}. First of all, the relative selection frequencies $\hat{\pi}$ of the 10 most selected covariates are presented in Figure 14.

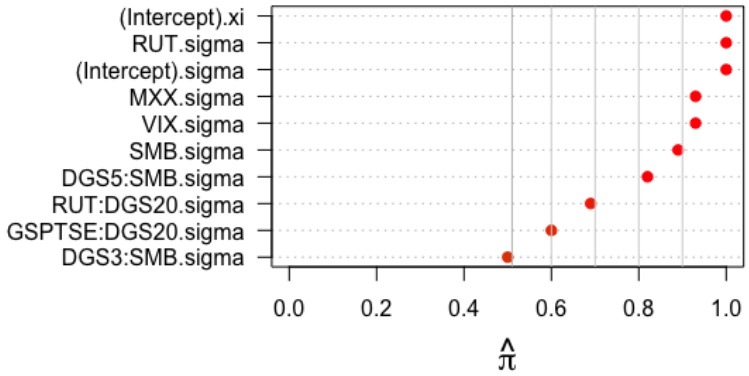


Figure 14: Relative selection frequencies for the 10 most selected covariates.

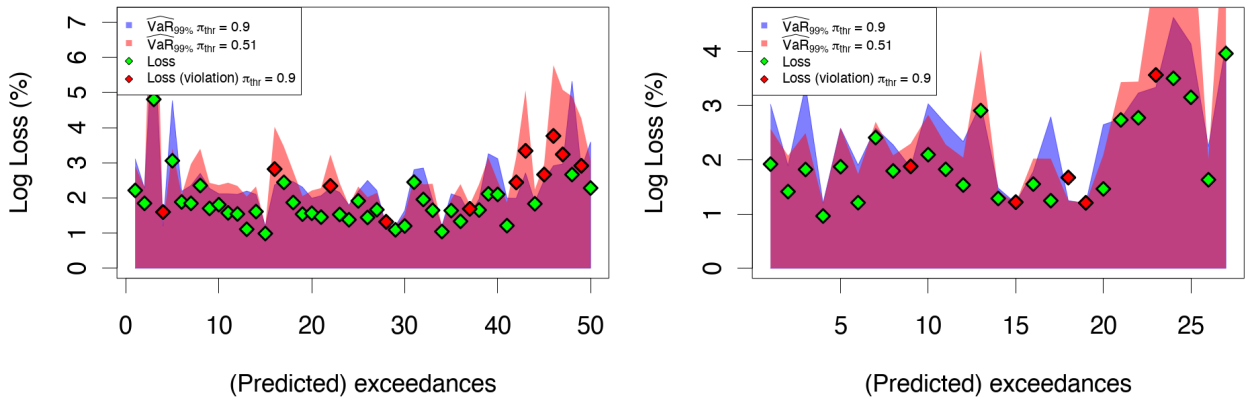
The first thing to notice is that the intercept for ξ and the intercept and Russel 2000 index for σ are selected in each run of the algorithm. The grey vertical lines show that $\pi_{thr} \in \{0.51, 0.7, 0.9\}$ results in three different sets of (stable) covariates, ranging from three to seven covariates, plus intercepts. We fully fit the stability selection model for the different π_{thr}, with the conditional threshold exceedance probability. The in-sample and out-of-sample performance of the estimated VaR at a 99% confidence level are presented in Table 8.

Table 8: Performance of the SS_{full} model for different π_{thr}.

	$\pi_{thr} = 0.51$	$\pi_{thr} = 0.7$	$\pi_{thr} = 0.9$	Empirical
$\widehat{\text{VaR}}_{99\%}$ *In-sample*				
Violations	34	34	**65**	51
Kupiec	(0.010)	(0.010)	**(0.061)**	-
Christoffersen	**(0.030)**	**(0.030)**	(0.001)	-
AQL	**0.015**	**0.015**	0.017	-
$\widehat{\text{VaR}}_{99\%}$ *Out-of-sample*				
Violations	**3**	**3**	5	3
Kupiec	**(0.806)**	**(0.806)**	(0.180)	-
Christoffersen	**(0.886)**	**(0.886)**	(0.315)	-
AQL	**0.042**	0.043	0.043	-

Setting $\pi_{thr} = 0.9$ results in the highest AQL and too many violations both in-sample and out-of-sample,

although not significantly according to the Kupiec backtest. The violations produced by the other two thresholds are equal, significantly too few in-sample but the exact right number of violations out-of-sample. We also investigate visually, by plotting the $\text{VaR}_{99\%}$ estimations and the realized log losses for the model with $\pi_{\text{thr}} = 0.51$ and $\pi_{\text{thr}} = 0.9$ in-sample and out-of-sample in Figure 15.



(a) In-sample (first 50).

(b) Out-of-sample.

Figure 15: Out-of-sample $\widehat{\text{VaR}}_{99\%}$ for the (predicted) threshold exceedances against the benchmark together with realized log losses.

In Figure 15a the model with $\pi_{\text{thr}} = 0.9$ severely underestimates the $\text{VaR}_{99\%}$ with 13 violations where 5 (10% of 50) are expected, and the model with $\pi_{\text{thr}} = 0.51$ seems to capture the movements of the realized log losses better. Together with the results of the backtests, we conclude a low threshold is preferable. This is in line with the simulation results in Section 4.4.

5.7 Added value interaction terms

The GAIC and t-VAL models do not perform variable selection, but they are regularized by subsetting the principal components. Section 5.5 shows that these models are outperformed on AQL out-of-sample by the fully fitted benchmark, with only the linear terms and not the interaction terms of the variables as covariates. Adding interaction terms without variable selection thus magnifies the effect of overfitting, and regularization based on significant t-values or lowest GAIC of the principal components does not make up for it. To evaluate if the interaction terms itself add value for VaR estimation, we fit the models $\text{GB}_{\text{sparse}}$ and GB_{rich} both to the full dataset and to a subset with just the linear covariates. The results are reported in Table 9.

Between the models with all terms and just the linear terms, there is no collective outperformance on all confidence levels both in-sample and out-of-sample for the backtests or the AQL. The differences between the models with all terms and the models with the linear terms in the number of $\text{VaR}_{97.5\%}$ violations in-sample stand out, with no clear cause. The Christoffersen test shows these violations are not independent for the sparse models, while the violations are not significantly too high or low. These significant p-values on the Christoffersen test can be explained by volatility clustering as stated before.

In-sample, the sparse gradient boosting models with all terms produces equal or slightly lower AQLs and equal or higher p-values for the backtests on all confidence levels, outperforming its model equivalents with just the linear terms. Remarkably, the reverse is true out-of-sample as the sparse gradient boosting model with linear terms produces a more accurate number of violations, higher p-values on the backtests, and lower AQL on all confidence levels.

The rich gradient boosting model with all terms in-sample produces equal or lower AQLs and equal or higher p-values for the backtests on all confidence levels compared to the rich gradient boosting model with

Table 9: In-sample and out-of-sample comparison of the estimated VaR of gradient boosting models with just the linear terms, and with all terms.

	In-sample					Out-of-sample				
	All terms		Linear terms			All terms		Linear terms		
	$GB_{sp.}$	GB_{rich}	$GB_{sp.}$	GB_{rich}	Emp.	$GB_{sp.}$	GB_{rich}	$GB_{sp.}$	GB_{rich}	Emp.
$\widehat{VaR}_{95\%}$										
Violations	**400**	427	**400**	414	256	23	24	**22**	24	13
Kupiec	(0.000)	(0.000)	(0.000)	(0.000)	-	(0.007)	(0.003)	**(0.014)**	(0.003)	-
Christof.	(0.000)	(0.000)	(0.000)	(0.000)	-	(0.001)	(0.001)	**(0.002)**	(0.001)	-
AQL	0.100	**0.098**	0.101	0.099	-	0.259	**0.239**	0.257	0.252	-
$\widehat{VaR}_{97.5\%}$										
Violations	116	**123**	145	164	128	10	9	**8**	9	7
Kupiec	(0.282)	**(0.664)**	(0.132)	(0.002)	-	(0.187)	(0.337)	**(0.557)**	(0.337)	-
Christof.	(0.040)	**(0.757)**	(0.012)	(0.008)	-	(0.141)	(0.612)	**(0.425)**	(0.263)	-
AQL	0.038	**0.037**	0.038	0.039	-	0.101	0.102	0.098	**0.096**	-
$\widehat{VaR}_{99\%}$										
Violations	32	**34**	32	**34**	51	3	3	3	3	3
Kupiec	(0.004)	**(0.010)**	(0.004)	**(0.010)**	-	(0.806)	(0.806)	(0.806)	(0.806)	-
Christof.	(0.013)	**(0.030)**	(0.013)	**(0.030)**	-	(0.886)	(0.886)	(0.886)	(0.886)	-
AQL	0.015	0.015	0.015	0.015	-	0.043	0.044	0.043	**0.042**	-

linear terms. Although the performance in-sample is slightly better for the models with all terms compared to their equivalents with just the linear terms, we have to conclude this does not hold out-of-sample. If anything, the models with just the linear terms perform better out-of-sample. We conclude that adding the interaction terms does not increase the performance of the gradient boosting models.

6 Conclusion

This paper compares models for high-dimensional market tail risk by fitting the GPD with dynamical parameters using the machine learning techniques gradient boosting, gradient boosting combined with stability selection, and PCR techniques. The high-dimensional characteristic stems from adding the interaction terms of the covariates.

Stasinopoulos, Rigby, Georgikopoulos, et al. (2021) propose to use interaction terms of economic variables to model complex economic relationships and provide a new method to fit distributional regression models to interrelated high-dimensional data, by adapting PCR to the GAMLSS framework. This technique performs implicit regularization by subsetting principal components but does not perform variable selection. Both variants, based on significant t-values and lowest GAIC, perform reasonably well in-sample, but unsatisfactory out-of-sample due to severe overfitting. Both models perform far worse than the benchmark model without the interactions, so adding interactions seems to increase overfitting, and regularization by subsetting the principal components does not solve this.

The simulation proves that variable selection for the shape parameter of the GPD is extremely difficult, which is in line with the models fitted to the empirical data selecting only the intercept for the shape parameter. The simulation shows that gradient boosting is good at selecting informative covariates for the shape parameter. For high collinearity, applying stability selection offers a slight improvement. This improvement is bigger for a lower number of observations, although gradient boosting itself is not too sensitive to the number of observations. Although the empirical data exhibits high collinearity, applying stability selection did not improve the estimation of the tail risk when compared to the gradient boosting, although it has similar performance and fewer selected variables. After applying stability selection, the regularized and fully fitted model perform similarly, so after variable selection, regularization is not so important. For

gradient boosting, the rich model using optimal m_{stop} outperforms the sparse model in-sample, but this is not true out-of-sample. We conclude that the choice of m_{stop} is important for in-sample results, but if economic interpretation is preferred one should lower m_{stop} as long as the out-of-bag-risk does not increase too much, yielding a sparser model. Because gradient boosting performs variable selection, it eases the (economic) interpretation of the models and it is also applicable to non-linear models. Gradient boosting both with and without stability selection should be preferred over the PCR methods for tail risk estimation. The large out-of-sample performance difference could indicate gradient boosting is superior to PCR for GAMLSS, but further research could show if PCR is of value as a method to handle high-dimensional data for GAMLSS.

In the simulation the informative interaction terms for the scale parameter proved to be hard to identify, resulting in both low and unstable TPRs of the selected covariates. In the empirical setting, adding the interaction terms of the variables as covariates did not improve tail risk estimation, neither for the PCR models nor the gradient boosting models. On top of worsening interpretation, adding interaction terms does not add value to tail risk modeling when using the GPD.

Further research could try different methods to obtain high-dimensional daily time series by e.g. adding lagged variables. Also, modeling the parameters as non-linear functions of the covariates is outside the scope of this paper but it could increase risk measure estimation and is well within the capabilities of gradient boosting and GAMLSS. One could also change the time interval of the time series to estimate e.g. intraday VaR which is useful for high-frequency trading.

References

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. *Selected Papers of Hirotugu Akaike*. New York, NY: Springer New York, pp. 199–213.

Anderson, D. R. and K. P. Burnham (2002). Avoiding pitfalls when using information-theoretic methods. *Journal of Wildlife Management* 66, pp. 912–918.

Chavez-Demoulin, V. and A. C. Davison (2005). Generalized additive modelling of sample extremes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 54.1, pp. 207–222.

Chavez-Demoulin, V., P. Embrechts, and M. Hofert (2016). An Extreme Value Approach for Modeling Operational Risk Losses Depending on Covariates. *Journal of Risk and Insurance* 83.3, pp. 735–776.

Chmielewski, L. J., M. Janowicz, L. Ochnio, and A. Orłowski (2015). "Clusterization of Indices and Assets in the Stock Market". *Intelligent Data Engineering and Automated Learning – IDEAL 2015*. Ed. by K. Jackowski, R. Burduk, K. Walkowiak, M. Wozniak, and H. Yin. Cham: Springer International Publishing, pp. 541–550. ISBN: 978-3-319-24834-9.

Christoffersen, P. F. (1998). Evaluating Interval Forecasts. *International Economic Review* 39.4, pp. 841–862.

Cole, T. and P. Green (1992). Smoothing reference centile curves: the LMS method and penalized likelihood. English. *Statistics in Medicine* 11, pp. 1305–1319. ISSN: 0277-6715.

Coles, S., J. Bawa, L. Trenner, and P. Dorazio (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer Series in Statistics. Springer. ISBN: 9781852334598.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* 1, pp. 223–236.

Embrechts, P., T. Mikosch, and C. Klüppelberg (1997). *Modelling Extremal Events: For Insurance and Finance*. Berlin, Heidelberg: Springer-Verlag. ISBN: 3540609318.

Fama, E. F. and K. R. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33.1, pp. 3–56. ISSN: 0304-405X.

Flack, V. F. and P. C. Chang (1987). Frequency of Selecting Noise Variables in Subset Regression Analysis: A Simulation Study. *The American Statistician* 41.1, pp. 84–86.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29.5, pp. 1189–1232.

Hadi, A. S. and R. F. Ling (1998). Some Cautionary Notes on the Use of Principal Components Regression. *The American Statistician* 52.1, pp. 15–19. ISSN: 00031305.

Hepp, T., M. Schmid, O. Gefeller, E. Waldmann, and A. Mayr (2016). Approaches to Regularized Regression - A Comparison between Gradient Boosting and the Lasso. *Methods of information in medicine* 55 5, pp. 422–430.

Hoerl, A. E. and R. W. Kennard (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12.1, pp. 55–67. ISSN: 00401706.

Hofner, B., L. Boccuto, and M. Göker (2015). Controlling false discoveries in high-dimensional situations: Boosting with stability selection. *BMC Bioinformatics* 16, p. 144.

Hofner, B. and T. Hothorn (2021). *Stabs: Stability Selection with Error Control*. R package version 0.6-4.

Hoxha, X. (2021). "Variable Selection in Tail Risk Modeling of Equity Returns". MA thesis.

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika* 76.2, pp. 297–307. ISSN: 0006-3444.

Judge, G. (1985). *The Theory and practice of econometrics*. 2nd ed. Wiley series in probability and mathematical statistics. New York: Wiley. ISBN: 047189530X.

Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models.

Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *The Journal of Business* 36.

Mayr, A., N. Fenske, B. Hofner, T. Kneib, and M. Schmid (2012). Generalized additive models for location, scale and shape for high dimensional data—a flexible approach based on boosting. *Journal of the Royal Statistical Society Series C* 61.3, pp. 403–427.

Mayr, A., B. Hofner, and M. Schmid (2012a). The Importance of Knowing When to Stop A Sequential Stopping Rule for Component-wise Gradient Boosting. *Methods of information in medicine* 51, pp. 178–86.

— (2012b). The Importance of Knowing When to Stop: A Sequential Stopping Rule for Component-wise Gradient Boosting. *Methods of information in medicine* 51, pp. 178–86.

McCullagh, P. and J. A. Nelder (1989). *Generalized linear models*. 2nd ed. Chapman and Hall/CRC. ISBN: 9780412317606.

McNeil, A. J., R. Frey, and P. Embrechts (2005). *Quantitative risk management: concepts, techniques and tools*.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.4, pp. 417–473.

Park, M. H. and J. H. Kim (2016). Estimating extreme tail risk measures with generalized Pareto distribution. *Computational Statistics Data Analysis* 98, pp. 91–104. ISSN: 0167-9473.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572.

Pickands, J. (1975). Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics* 3, pp. 119–131. ISSN: 00905364.

Rigby, R. A. and D. M. Stasinopoulos (1996). A semi-parametric additive model for variance heterogeneity. *Statistics and Computing* 6, pp. 57–65.

— (2005). Generalized additive models for location, scale and shape,(with discussion). *Applied Statistics* 54, pp. 507–554.

Rigby, R. A. and D. M. Stasinopoulos (2007). Generalized Additive Models for Location Scale and Shape (GAMLSS) in R. *Journal of Statistical Software* 23.7, pp. 1–46.

Ripley, B. D. (2004). "SELECTING AMONGST LARGE CLASSES OF MODELS".

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics* 6.2, pp. 461–464. ISSN: 00905364.

Shah, R. D. and R. J. Samworth (2013). Variable selection with error control: another look at stability selection. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 75.1, pp. 55–80. ISSN: 13697412, 14679868.

Stasinopoulos, M., B. Rigby, and F. De Bastiani (2021). *gamlss.foreach: Parallel Computations for Distributional Regression*. R package version 1.1-3.

Stasinopoulos, M., R. Rigby, N. Georgikopoulos, and F. De Bastiani (2021). Principal component regression in GAMLSS applied to Greek-German government bond yield spreads. *Statistical Modelling*, pp. 1–19.

Su, W., M. Bogdan, and E. Candès (2017). False Discoveries Occur Early on the Lasso Path. *The Annals of Statistics* 45.5, pp. 2133–2150. ISSN: 00905364.

Thomas, J., A. Mayr, B. Bischl, M. Schmid, A. Smith, and B. Hofner (2018). Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates. *Statistics and Computing* 28, pp. 1–15.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1, pp. 267–288. ISSN: 00359246.

Zou, H. and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2, pp. 301–320. ISSN: 13697412, 14679868.

Appendices

A Figures

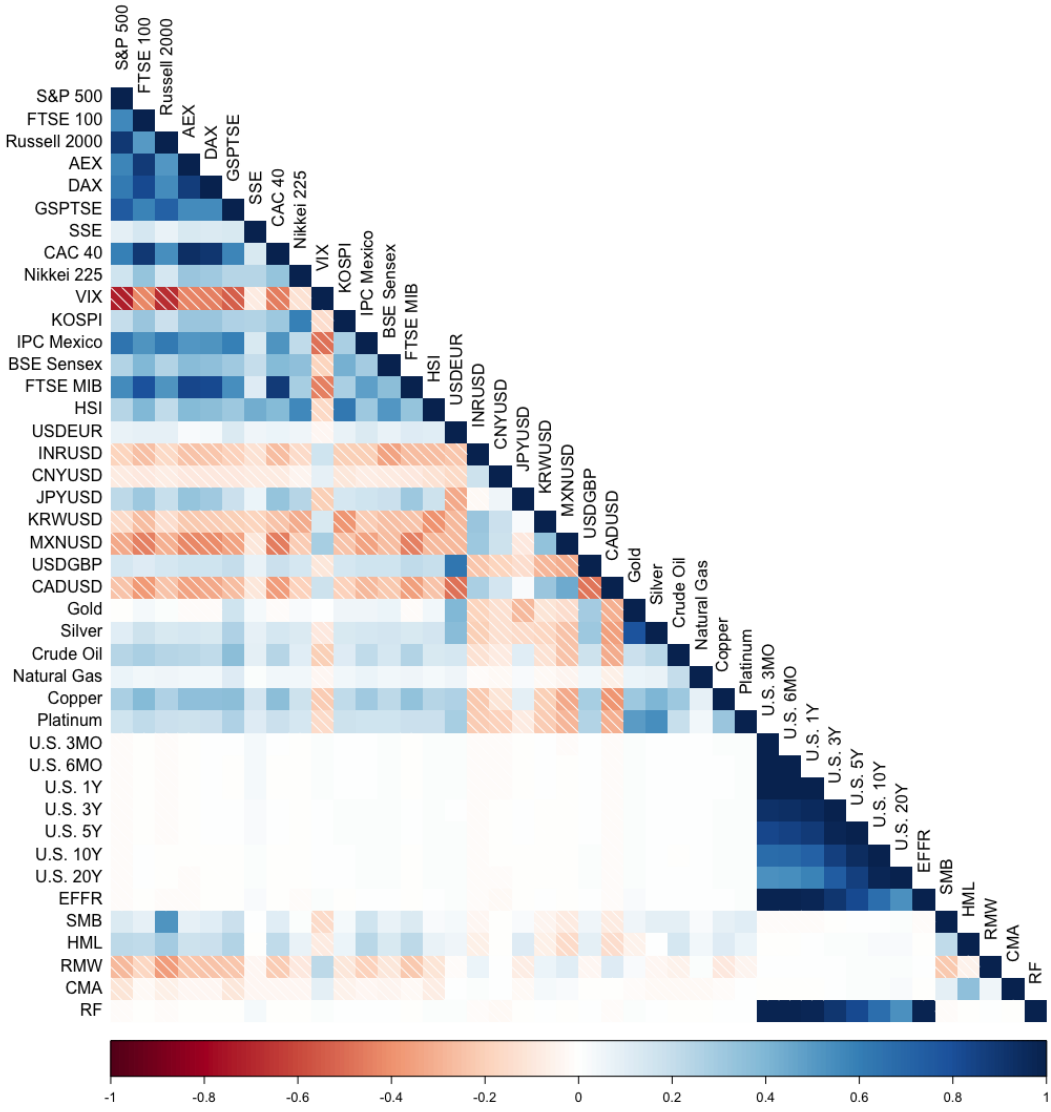


Figure A.1: Correlations between the S&P 500 log losses and the covariates.

B Models

Table B.1: Characteristics of the models fitted on empirical data.

Model	Number of covariates	Linear terms	Hyperparameter settings
$\text{GB}_{\text{sparse}}$	18	7	$m_{\text{stop}} = 400$
GB_{rich}	75	12	$m_{\text{stop}} = 1491$
SS_{opt}	7	4	$q = 0.51$, $\pi_{\text{thr}} = 0.51 + m_{\text{stop}} = 1997$ (gradient boosting)
SS_{full}	7	4	$q = 0.51$, $\pi_{\text{thr}} = 0.51$ + regularly fully fitted
GAIC	First 8 PCs		-
t-VAL	130 PCs		-
Benchmark	41 (fully fitted)	41	-
Exponential	15	4	$m_{\text{stop}} = 1995$

Table B.2: Selected variables by the machine learning models.

Model	Selected variables
$\text{GB}_{\text{sparse}}$	RUT, GDAXI, VIX, MXX, HG, SMB, RMW, RUT:BSESN, RUT:DGS20, GDAXI:DGS20, GSPTSE:SI, GSPTSE:DGS20, SS:PL, DEXUSEU:RMW, DEXCHUS:HML, DEXUSUK:GC, SI:NG, DGS5:SMB.
GB_{rich}	RUT, GDAXI,GSPTSE,VIX, MXX,HSI,DEXINUS,CL,HG,DGS10,SMB,RMW. + 63 interaction terms.
SS_{opt}	RUT, VIX, MXX, SMB, RUT:DGS20, DGS20:GSPTSE, SMB:DGS5
SS_{full}	RUT, VIX, MXX, SMB, RUT:DGS20, DGS20:GSPTSE, SMB:DGS5
Exponential	RUT, VIX, SMB, RMW,FTSE:DGS20, FTSE:CMA, RUT:DGS20, SS:FTSEMIB, VIX:DGS20, KS:CMA, DEXUSEU:CMA, DEXMXUS:DGS3MO, DGS20:SMB, DGS20:CMA, RMW:CMA

C Algorithms

Algorithm C.1: Lines 7-11 of the RS algorithm for principal component regression for the GPD, GAIC approach.

7 **for** $\lambda = 1, 2, ..., r$ **do**

8 Estimate first λ PCs coefficients: $\hat{\boldsymbol{\gamma}}_\lambda = \left(\mathbf{T}_\lambda^\top \mathbf{W}_k \mathbf{T}_\lambda\right)^{-1} \mathbf{T}_\lambda^\top \mathbf{W}_k \boldsymbol{z}_k^{[m]}$, with $\mathbf{W}_k = \text{diag}(\boldsymbol{w}_k)$ and $\mathbf{T}_\lambda = \mathbf{T}_{[1:\lambda]}$.

9 Compute fitted values $\hat{\boldsymbol{z}}_{k,i}^{[m]} = \mathbf{T}_\lambda \hat{\boldsymbol{\gamma}}_\lambda$.

10 Get local $GAIC_\lambda = \sum_{i=1}^n w_{k,i}(z_{k,i}^{[m]} - \hat{z}_{k,i}^{[m]})^2 + \log(n) \cdot (\lambda + 1)$.

11 The λ that corresponds to the minimum GAIC is chosen.