Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

Algorithms for Robust Predictor Filtering and Evaluation of Their Stability

Boris V. Malozyomov[1], Nikita V. Martyushev[2*], Roman V. Klyuev[3], Anton Y. Demin[2], Svetlana N. Sorokova[2], Egor A. Efremenkov[2], Denis V. Valuev[4], Andrei O. Kotov[2]

[1]Novosibirsk State Technical University, Novosibirsk, Russia
[2]Tomsk Polytechnic University, Tomsk, Russia
[3]Moscow Polytechnic University, Moscow, Russia
[4]Yurga Technological Institute (branch), Tomsk Polytechnic University, Yurga, Russia
E-mail: martjushev@tpu.ru

Received: 11 November 2025; **Revised:** 5 February 2026; **Accepted:** 9 February 2026

Abstract: The paper is devoted to the development of an algorithm for reliable predictor filtering based on Henze-Zirkler statistics, development of an iterative procedure based on Lass regularization. The stability of the algorithms based on modelled and real examples is studied. The description and investigation of existing robust filtering algorithms are given. In the process, two algorithms have been implemented for the study. The second procedure is an improvement of the first algorithm which screens out highly correlated predictors. A comparative analysis with existing filtering algorithms was carried out, and the stability of Henze-Zirkler robust filtering algorithm and the iterative procedure based on Lasso regularization was investigated. As a result of the work, conclusions were drawn about the effectiveness of the Henze-Zirkler robust filtering algorithm and the iterative procedure based on Lasso regularization. In addition, shortcomings in the stability of the algorithms when dealing with categorical data were identified.

***Keywords*:** robust predictor filtering, Henze-Zirkler statistics, regression analysis, lasso regularization

MSC: 62J12

1. Introduction

Objects of human activity are characterized by a large number of different properties and relations between them, which need to be represented in the correct form for their further processing. Usually, each property of an object is called a feature. And the data are presented in the form of the matrix "object-sign", the rows of which are correlated with the analyzed objects or the number of experience and the columns with the values of the studied features [1–3].

Variable selection plays an important role in analyzing high dimensional data. However, in ultra-high dimensional settings, where the number of covariates can grow exponentially as a function of sample size, current variable selection methods may not perform well due to simultaneous issues related to computational feasibility, statistical accuracy, and algorithmic stability [4–6]. A practical approach is to use a filtering procedure to reduce the dimensionality of the feature space to a moderate scale and to implement a variable selection method for the reduced dataset [7, 8].

DOI: https://doi.org/10.37256/cm.7220269142

Developing and researching algorithms that are model-and condition-independent are an urgent problem due to the annually increasing data sizes [9, 10]. And without universal algorithms that will select only important features and filter out correlated ones, it will be more difficult to process this data. One such algorithm is the Henze-Zirkler Sure Independence Screening (HZ-SIS) method [4]. In [6], the authors investigated the stability of the algorithm using additive examples with distributions of heavy tails. The algorithm showed good robustness to outliers [11, 12]. In this paper, the robustness of the algorithm is investigated on a model example with highly correlated predictors and a real example of oil production [13, 14]. Robust feature screening methods are increasingly applied in engineering, energy, and reliability-oriented studies, where high-dimensional measurements and correlated predictors are common. Although the present work focuses on methodological aspects of screening, these application domains provide important motivation and are therefore referenced to illustrate the practical relevance of stable and correlation-aware feature selection.

In practice, two shortcomings of existing screening methods are considered this study. First, marginal screening scores may be unstable under heavy-tailed noise and outliers, which leads to variability of selected subsets across resamples and undermines downstream modelling. Second, in the presence of strongly correlated predictors, in screening procedures redundant variables are often used that dominate the ranking, while omitting weaker but relevant predictors. This reduces interpretability and can degrade predictive models due to multicollinearity. These issues are central in high-dimensional engineering datasets and motivate the combined use of robust HZ-based screening and an explicit correlation-aware iterative refinement.

The aim of this paper is to compose and validate a robust filtering algorithm based on Henze-Zirkler statistics and a procedure for screening out highly correlated predictors to improve the results. The paper makes the following contributions. We provide an implementation-focused study of Henze-Zirkler Sure Independence Screening (HZ-SIS) and systematically evaluate its stability under two practically important sources of instability: heavy-tailed/outlier-contaminated samples and strongly correlated predictors. We propose an iterative extension of HZ-SIS that uses Lasso-based residualization of the remaining predictors with respect to the already selected set, explicitly targeting inter-predictor dependence and reducing redundancy in the selected subset. We compare the baseline and iterative procedures using a controlled simulated example with correlated predictors and a real oil-production dataset, highlighting the moments when each approach succeeds or fails, including the known limitations on categorical/discrete variables. This includes design and stability study of algorithms for relevant feature selection.

This work had the following objectives:

1) To study the HZ-SIS algorithm;

2) To study the procedure for screening out highly correlated predictors based on the lasso regularization method;

3) To develop a software module to solve the previous problems;

4) To investigate the work of the algorithm based on model and real data. The data on oil production volumes are real.

The remainder of the paper is organized as follows. Section 1 reviews classical and model-free screening methods and introduces the baseline HZ-SIS procedure together with the proposed Lasso-based iterative extension for correlation-aware refinement. Section 2 describes the implemented software module and computational setup used for reproducible experiments. Section 3 reports the stability study on a controlled simulated example with correlated predictors and on a real oil-production dataset, including a comparative discussion against representative baseline screening algorithms. Finally, the Conclusion summarizes the main findings, limitations, and practical recommendations.

The methodological contribution of this work can be summarized as follows. First, unlike classical SIS and model-free SIS procedures that rely on marginal correlation or distance-based dependence measures, the proposed adaptation of the Henze-Zirkler Sure Independence Screening (HZ-SIS) employs a non-paranormal transformation combined with a multivariate normality test, allowing independence assessment without specifying a regression model or imposing strong distributional assumptions on predictors and response. This enables robust screening under heavy-tailed distributions and in the presence of outliers, while preserving the sure screening property under weaker conditions.

Second, this paper introduces an iterative regularization-based extension of HZ-SIS that fundamentally differs from existing iterative screening strategies. Instead of reapplying screening directly to the original predictors or residuals of the response variable, the proposed procedure iteratively removes the linear effects of already selected predictors from the

remaining predictors using Lasso regularization. This mechanism explicitly targets multicollinearity between predictors, rather than between predictors and the response, thereby reducing redundancy while increasing the probability of retaining weak but relevant predictors. The combined HZ-SIS and Lasso-based iterative procedure thus fills a methodological gap between robust model-free screening and correlation-aware variable reduction, which has received limited attention in the existing SIS literature.

2. Theoretical aspects of algorithms for robust predictor filtering

Existing sure independence screening methods can be broadly classified into several groups depending on the type of dependence measure they employ. Representative examples considered in this paper include Pearson correlation-based SIS, rank-based Sure Independence Ranking and Screening (SIRS), Distance-Correlation Screening (DC-SIS), Quantile-adaptive Screening (Qa-SIS), and Mean-Variance Screening (MV-SIS), which differ in their dependence measures and assumptions. Classical SIS methods are based on marginal correlation and are typically model-dependent, assuming a correctly specified linear or generalized linear model. Model-free screening approaches replace correlation with alternative dependence measures, including distance correlation, quantile-adaptive measures, or mean-variance criteria, aiming to relax model assumptions. However, these methods still rely on restrictive distributional or smoothness conditions and are often sensitive to heavy-tailed distributions, outliers, or strong inter-predictor correlations. The review below focuses on these classes of screening methods and highlights their theoretical and practical limitations that motivate the use of Henze-Zirkler statistics combined with a non-paranormal transformation.

Initially, to realize the variable selection algorithm, Henze and Zirkler [4] proposed a robust filtering method for linear regression in which the features are tested based on their Pearson correlation coefficients from the response variable [15]. They established the property of robust filtering, that is, all active predictors can be selected with probability approaching unity when the sample size is increased to infinity. He and Wang [5] extended the robust filtering method to generalized linear models, where features are selected based on estimates of regression coefficients in appropriate behavioural models [16, 17].

However, such methods were very much dependent on the constructed model. If the underlying model is constructed correctly, such methods work well, but if not, such methods may produce incorrect results [18]. But in real life, the dimensionality can be very high, and determining the correct model is an impossible task.

Subsequently, methods that were model-independent were developed. In this direction, Zhu et al. [19, 20] proposed the Sure Independence Ranking and Screening (SIRS) method for testing the significant features of multi-index models. Pudjihartono et al. [8, 15, 21] proposed a Distance Correlation Sure Independence Screening (DC-SIS) method in which the features are tested based on their distance correlation with the response variable. It is known that for two random variables, a zero distance correlation coefficient implies independence between them. In [22, 23], the authors proposed a quantile-adaptive sure independence screening method (Quantile adaptive Sure Independence Screening (Qa-SIS)), which uses spline approximation to model the marginal effect of each predictor for quantiles of a given level and then tests the predictors, accordingly. A particular advantage of this method is that it can handle censored data arising from survival analyses [24]. Fan et al. [3, 25] proposed a Mean Variance Sure Independence Screening (MV-SIS) method, in which the dependence of two random variables is measured using the mean variance of the conditional distribution function [26, 27]. This method was originally proposed for a categorical response variable, but it can be extended to problems for which the response variable is continuous by discretization.

Although model-free robust variable filtering methods avoid specification of a specific model structure, they still rely, to a greater or lesser extent, on some assumptions for predictor and response variables [28, 29]. For example, DC-SIS requires that the probabilities of both the predictors and the response variable at the tails of the distribution be subexponentially uniform. That is, in practice, the response variable and predictors must be uniformly bounded or follow a multivariate normal distribution. Qa-SIS requires the derivative of the conditional quantile function to satisfy the Lipschitz condition and the conditional density function to be uniformly bounded for each trait [30, 31].

A new method of robust filtering free from model assumptions was investigated in the work. In [13, 32], the method proposed works based on the non-paranormal transformation and the Henze-Zirkler criterion. Compared to existing methods, the proposed method requires fewer assumptions to guarantee the robust filtering property and thus works more robustly. Moreover, most existing approaches do not explicitly address the stability of screening results in the presence of strongly correlated predictors, which remains a critical practical issue in high-dimensional applications.

Compared to DC-SIS, Qa-SIS, MV-SIS, and SIRS, the proposed HZ-SIS approach differs primarily in the way the dependence is assessed (Table 1). DC-SIS relies on distance correlation and requires tail conditions for both predictors and response. Qa-SIS depends on smoothness assumptions for conditional quantiles, and MV-SIS involves discretization or categorical extensions. SIRS is designed for specific multi-index structures. In contrast, HZ-SIS leverages a non-paranormal transformation combined with a multivariate normality test, allowing detection of general dependence patterns under weaker model assumptions. Simultaneously, the proposed iterative extension explicitly addresses inter-predictor correlation, which is not directly handled in these alternatives.

Table 1. Correspondence of feature names and titles in studies

Method	Dependence measure	Key assumptions	Main limitations
SIS	Pearson correlation	Linear model, finite moments	Sensitive to outliers and correlation
SIRS	Rank-based	Specific index structure	Limited to certain model classes
DC-SIS	Distance correlation	Tail conditions, boundedness	Sensitive to heavy tails
Qa-SIS	Quantile regression	Smooth conditional quantiles	Requires tuning, model assumptions
MV-SIS	Mean-variance	Discretization or categorical setup	Information loss
HZ-SIS (this work)	HZ normality test after NPN	Continuous variables	Not suitable for categorical data

Before the main filtering algorithm, we have the following data as input: $X = (X_1, \ldots, X_p)$, where p is the number of features; Y is the response variable that depends on X, $f = (y \mid x)$, the density function of Y for a given X; n is the sample size.

First, we need to transform the response variable and each of the predictors into normal random variables using the nonparanormal transformation [33]. And then it is possible to test the relationship between the response variable and the predictors using the Henze-Zirkler criterion. Without specifying the parametric form for the regression model, we define the sets of active predictors and inactive predictors as follows: $D = \{k : (y \mid x)$ is functionally dependent on$\}$, $I = \{k : f(y \mid x)$ is functionally independent of $X_k\}$.

Under high dimensionality, it becomes difficult to identify a set of active predictors. Therefore, it was proposed to first define a moderately sized set of variables that included all the elements of D, and then apply the variable selection method to this moderately sized set to correctly and accurately determine D. If $(y \mid x)$ is functionally dependent on X, then Y and X_k are usually also indirectly dependent. It follows that a moderately sized set of variables can be constructed by selecting only predictors that are indirectly related to Y. This procedure is called independence checking. Under the condition of partial orthogonality, i.e., $\{y_i \in D\}$ is independent of $\{x_j,\ j \in I\}$, it can be shown that $(y \mid x)$ is functionally dependent on x_k if and only if Y and X_k are indirectly dependent [34].

To implement the independence test, we need to find a metric to measure the marginal relationship between each predictor (X_k) and the response variable (Y). Let $(\cdot)$ denote the distribution function of the response variable (Y), and let $F_k(\cdot)$ denote the distribution function of the predictor. The non-paranormal transformation in this case is as follows:

$$(Y) = F^{-1}(F_y(Y)),\ T_k(X) = F^{-1}(F_k(X_k)),\ k = 1, \ldots, p, \tag{1}$$

where F denotes the distribution function of the standard normal distribution.

The transformation described in the formula (1) applies to continuous random variables of general form. It can be easily seen that Y is independent of X_k if and only if $(Y),\ T_k()$ obey the bivariate normal distribution of $N_2(0,\ 2)$, where I_2 denotes a 2×2 unit matrix. This can be tested using a multivariate statistical criterion for normality testing, such as the Henze-Zirkler criterion with a known covariance structure. If $(Y), T_k()$ do not obey the distribution of $N_2(0,\ 2)$, then the statistic of the Henze-Zirkler criterion tends to take a larger value [35]. We can use the estimated non-paranormal transformation since usually F_y and F_k are unknown.

It should be noted that the proposed HZ-SIS methodology is inherently designed for continuous predictors. Since the Henze-Zirkler test evaluates departures from multivariate normality after a non-paranormal transformation, its assumptions are structurally violated for discrete or categorical variables with a small number of unique values. In such cases, the transformation produces highly concentrated values, leading to inflated test statistics and unstable screening results. This limitation is revisited and empirically illustrated in the experimental sections. The HZ-SIS framework is primarily suited for continuous predictors. After the non-paranormal transformation, the Henze-Zirkler statistics assesses whether $(T(Y),\ T_k(X_k))$ follows a bivariate normal distribution. For discrete or categorical predictors with few unique values (e.g., binary 0/1), the transformed samples remain highly concentrated and can exhibit ties and nonsmooth empirical distributions. This structurally violates the normality-based assumptions behind the Henze-Zirkler test and may inflate the HZ statistics, leading to unstable rankings and the artificial dominance of low-cardinality predictors (especially under strong class imbalance). This effect is later illustrated in the real-data study.

In this study, stability is understood as the weak sensitivity of the screening result to small perturbations of the data distribution, including the presence of heavy-tailed noise and isolated outliers. The robustness of the HZ-based statistics follows from its construction via characteristic functions with exponentially decaying weights, which suppress the influence of extreme observations. In contrast to correlation-based measures relying on second moments, the HZ statistic remains well defined and bounded for distributions with heavy tails.

As shown in Henze and Zirkler [4] and subsequent studies, the HZ test possesses affine invariance and bounded influence behavior, which implies that marginal screening based on this statistics is less sensitive to deviations from Gaussianity and to heavy-tailed contamination. Therefore, although a full asymptotic robustness proof is beyond the scope of this applied study, the observed stability of HZ-SIS under heavy-tailed sampling in Section 3 is theoretically supported by the known robustness properties of the underlying statistics.

2.1 *Henze-Zirkler independent screening algorithm*

In [13], the authors break the algorithm into three steps.

Step 1 is executed on each predictor and response variable; for example, we will take the response variable (Y). The truncated empirical distribution function must be found, which is constructed by the formula:

$$\tilde{F}_y = \begin{cases} \delta_n : \hat{F}_y(t) < \delta_n; \\ \hat{F}_y(t) : \delta_n \leq \hat{F}_y(t) \leq 1 - \delta_n; \\ 1 - \delta_n : \hat{F}_y(t) > 1 - \delta_n, \end{cases} \tag{2}$$

where $\hat{F}_y(t) = \dfrac{1}{n}\sum_{i=1}^{n} 1_{(y_i \leq t)}$ is the distribution function, and $\delta_n = \dfrac{1}{4} n^{-\frac{1}{4}} (\pi\ \log\ n)^{-\frac{1}{2}}$ is the standard truncation parameter.

The truncation parameter (δ) controls the clipping of extreme empirical quantiles in the non-paranormal transformation. Its primary role is to stabilize the transformation in finite samples by preventing the empirical distribution function from attaining values too close to 0 or 1. This would otherwise lead to infinite values after applying the Gaussian quantile function. From a statistical perspective, δ regulates the bias-variance trade-off of the transformation. Smaller values retain more tail information, but increase sensitivity to outliers, whereas larger values improve robustness at the cost of tail

resolution. In this study, δ is chosen as a deterministic function of the sample size, following standard recommendations in the non-paranormal literature, ensuring asymptotic consistency while maintaining numerical stability.

We convert the values of the response variable to standard random normal variables [36] using a non-paranormal transformation:

$$\tilde{T}(y_i) = F^{-1}\left(\tilde{F}_y(y_i)\right),\ i = 1,\ \ldots,\ n,$$

where y_i is the i-th observation.

Step 2. For each predictor, we calculate the Henze-Zirkler statistics:

$$\tilde{w}_k^* = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} e^{-\frac{\beta^2}{2}d_{ij}} - \frac{2}{n(1+\beta^2)}\sum_{i=1}^{n} e^{-\frac{\beta^2}{2\left(1+\beta^2\right)}d_i} + \frac{1}{1+2\beta^2},$$

where $\beta = \frac{(1.25n)^{\frac{1}{6}}}{\sqrt{2}}$ is a smoothing parameter that corresponds to the optimal window width for nonparametric kernel density estimation with Gaussian kernel. $d_{ij} = \left(\tilde{T}_k(x_{ki}) - \tilde{T}_k\left(x_{kj}\right)\right)^2 + \left(\tilde{T}_y(y_i) - \tilde{T}_y\left(y_j\right)\right)^2$; $d_i = \tilde{T}_k(x)_{ki}^2 + \tilde{T}_y(y)_i^2$; $\tilde{T}_k(x_{ki})$ and $\tilde{T}_y(y)_i$ denote the i-th implementation of $\tilde{T}_k(X_k)$ and $\tilde{T}_y(Y)$, respectively.

The smoothing parameter (β) determines the sensitivity of the Henze-Zirkler statistics to deviations from multivariate normality. Smaller values of β emphasize local discrepancies, making the test more sensitive to subtle dependence structures, whereas larger values produce stronger smoothing and favor detection of global departures from normality. In the context of HZ-SIS, β therefore directly influences the ranking of predictors by controlling how strongly marginal dependence between each predictor and the response is penalized. The choice of β as a function of the sample size follows established theoretical results for the Henze-Zirkler test and is adopted in this case to ensure stable behavior across different data scales.

Conceptually, our iterative procedure differs from previously published iterative SIS variants in that it does not iterate screening on response residuals alone. Instead, it iteratively residualizes the unselected predictors against the already selected set via Lasso, so that predictors whose signal is explained by the current selection lose priority, which directly addresses multicollinearity within the feature space.

Step 3. We choose the set of predictors with the highest values of $\tilde{w}_k^*$, i.e.:

$$\hat{D} = \left\{k : \tilde{w}_k^* > cn^{-k},\ 1 \le k \le p\right\}, \tag{3}$$

where c and k are the set threshold values.

However, c and k are usually hard to determine, other feature selection methods can be used to set the size of $\hat{D}$ in the formula (4) equal to $\left[\frac{n}{\log n}\right]$, where $[z]$ denotes the integer part of z.

In [37], the authors investigated the HZ-SIS algorithm using an additive example that tests the stability of algorithms d distributions with heavy tails. The authors modelled an example where the number of features was 2,000, the volume of each sample was 200, and the number of experiments was 100.

The authors' results show that HZ-SIS performs well compared to the conventional SIS. HZ-SIS selects all 4 relevant features 80% of the time, while SIS does not select all 4 relevant features simultaneously in all experiments. Such results show that when selected by the SIS algorithm, the value of the response variable will differ in most cases from the predicted value of the response variable without feature selection. From this additive example, it can be seen that HZ-SIS performs well in feature selection in heavy-tailed examples. The algorithm is well robust to outliers and has low sensitivity to

outliers and inhomogeneities in the sample. In this paper, we also investigate the algorithm for robustness to the inclusion of highly correlated predictors, as well as using a real-world example about oil production [38].

2.2 *Regularization-based iterative procedure*

The following problem may arise when using the HZ-SIS algorithm: some predictors that do not provide useful information may be highly correlated with important predictors and have a higher priority for HZ-SIS selection than that of other important predictors [39]. Correlation is the relationship between two or more random variables. If the features are highly correlated with each other, they provide the same information for the model [40]. If such features are not reduced, the overfitting of the model occurs, which negatively affects the results. Therefore, one tries to reduce the number of such features [41].

To address this problem, in [42], the authors proposed iterative versions of SIS algorithms. The idea is to iteratively apply large-scale feature selection followed by moderate careful variable selection. SIS works with linear regression, so the residuals can be used to eliminate certain highly correlated predictors through iterations. In [43], the authors propose selecting at each iteration some subset of k_1 variables $\left(A_1 = \left\{X_{i1}, \ldots, X_{ik_1}\right\}\right)$ using the SIS-based model selection method. By applying regularization, they obtain an n -dimensional vector of residuals from the response regression on $X_{i_1}, \ldots, ik_1$. The next step processes these results and applies the same method as in the previous step to the remaining predictors of $p-k_1$, obtaining a set of k_2 variables $\left(A_2 = \left\{X_{i_1}, \ldots, X_{ik_2}\right\}\right)$. Continuing the algorithm, they obtain the A set from the union of A_i subsets of D size.

This helps to address the problem with highly correlated predictors, not excluding informative predictors. Unlike response-residual iterations in classical iterative SIS, the key mechanism here is predictor-residualization, i.e., removing the linear component of each remaining predictor explained by the selected subset, which suppresses redundant correlated variables in subsequent screening steps. Similar to SIS, the correlation between predictors is eliminated, so for HZ-SIS this can be accounted for using the following procedure:

Step 1: Apply HZ-SIS to the original data and select $A_{(1)}$ predictors, where $p_{(1)} < \left[\frac{n}{\log(n)}\right]$. We denote the set of selected predictors by $A_{(1)}$.

Step 2: Treat each predictor $\notin A_{(1)}$ as a response variable and construct a Lasso regression for it, including the $A_{(1)}$ set as input variables. We will then calculate the residuals of the regression.

Step 3: Apply HZ-SIS to the residuals calculated in step 2. Suppose that $p_{(2)}$ residuals are selected and the corresponding set of variables is denoted by $A_{(2)}$.

We update the fully selected set of $A_{(1)} \cup A_{(2)}$ predictors.

We update the fully selected set of $A_{(1)} \cup A_{(2)}$ predictors.

Step 4: Repeat steps (2) and (3) m^{-1} times until the number of predictors $\left(p_{(1)} + \ldots + p_{(m)}\right)$, exceeding $\left[\frac{n}{\log(n)}\right]$, is completely selected. As a result, we obtain the set of selected predictors $A_{(1)} \cup \ldots \cup A_{(m)}$.

Obviously, if a predictor is highly correlated with some already selected predictors, it will not be selected by this procedure. This increases the chance to select true predictors that are weakly dependent on the response [44]. It is suggested to work with the original predictors in step 2.

2.3 *Lasso regularization*

Regularization is a general method that consists in imposing additional constraints on the sought parameters, which can prevent the model from being overly complex.

The general principle of regularization methods is described as

$$\min_{\hat{\beta}} RSS + \text{ Fine},$$

where *RSS* is the residual sum of squares. The Lasso regression method consists in introducing an additional regularization term into the model optimization functional, which often results in a more stable solution.

At the same time, Lasso regression is defined as

$$\min_{\hat{\beta}} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda (|\hat{\beta}_1| + |\hat{\beta}_2| + \ldots + |\hat{\beta}_k|), \tag{4}$$

where y_i is the response value, $\hat{y}_i$ is the predicted response value, λ is the regularization parameter, $\hat{\beta}_i$ is the element of the coefficient vector.

The choice of the regularization parameter (λ) for the penalty is a relevant problem. A poor regularization parameter can reduce the stability of the algorithm and the results may be incorrect. There are different ways to choose the regularization parameter, but in this paper we will vary the regularization parameter to investigate the stability [45, 46]. The values of the regularization parameter (λ) considered in this study are not intended to provide an optimal predictive model, but rather to explore the sensitivity of the iterative screening procedure to different degrees of shrinkage. Small values of λ correspond to weak regularization and approximate ordinary least squares, leading to limited removal of correlated predictors. Conversely, large values of λ enforce strong sparsity and effectively suppress predictors that are linearly explainable by the already selected set. Examining a wide range of λ thus allows assessment of the robustness of the proposed iterative procedure with respect to multicollinearity control rather than tuning λ for predictive accuracy.

3. Description of the software module of robust predictor filtering

3.1 *General information*

The research programme consists of several modules implementing different functionality and connected together. In total, there are 6 modules in the programme. Each module is completely written in Python [47]. The programme also has a graphical interface for more convenient work. To ensure reproducibility of all simulation results, fixed random seeds were used for data generation and algorithmic procedures. In particular, pseudo-random number generators in NumPy and Python standard libraries were initialized with predefined seeds prior to each experimental run, ensuring that identical inputs lead to identical screening outcomes across independent executions.

To work with the programme, it is necessary to have the Python Anaconda language distribution and Visual Studio Code source code editor installed. In addition, to run it, the PySimpleGUI library must be installed in the terminal to display the graphical interface [48]. This can be done in two ways: pip3 install pysimplegui, and pip install pysimplegui. From a computational perspective, the HZ-SIS procedure requires $O\left(p \cdot n^2\right)$ operations due to pairwise distance computations for each predictor, which is comparable to other kernel-based screening methods. The iterative regularization-based extension introduces additional Lasso regressions, leading to an overall complexity of approximately $O\left(m \cdot p \cdot n^2\right)$, where m is the number of iterations. In practice, m remains small, and the screening stage substantially reduces dimensionality, making the approach feasible for moderately large p. The implementation is fully vectorized where possible and demonstrated to run efficiently on high-performance computing infrastructure.

3.2 *Functional purpose*

The program is designed to investigate an independent screening algorithm based on Henze-Zirkler statistics and a correlated predictor reduction algorithm based on Lasso regularization [49].

3.3 *Description of the logical structure*

In the programme, each module has its own functionality.

The main module is Window.py, which is designed to create the Graphical User Interface (GUI) and send data to the Run.py module for processing. After the user enters the correct data for the algorithm to work, the module saves this data and calls the Run() function of the main.py module.

Main.py is the main module where the algorithm is executed and data is processed and written to files. There are the following functions in main.py.

The HZ-SIS() function performs the algorithm described earlier in Subsection 1.1. At the first step of the function, the data are transformed into standard random normal variables. For this purpose, the empirical_func() function is called, which constructs a truncated empirical distribution function. At the second step, the weights w() are calculated by calling calculate_w() . The values of the weights are then sorted in a descending order by calling shellSort() from the sorting.py module.

The second algorithm is performed in the LRL() function to select weakly correlated predictors. At the first step of the algorithm described in Subsection 1.2, the sets of A and $X \notin A$ are constructed in the function. The second step performs Lasso regularization from the standard Python sklearn library. At the third step of the function, the HZ-SIS algorithm is executed and some more predictors are selected. Steps 2 and 3 are executed until the size of the set exceeds $\left[\frac{n}{\log(n)}\right]$. The output is the set of predictors and the values of the weights.

For each Xk predictor, HZ-SIS computes the Henze-Zirkler statistics using pairwise distances across nnn observations, which requires $O\left(n^2\right)$ operations per predictor. Therefore, the overall screening stage scales as $O\left(pn^2\right)$ (plus $O(n\log(n))$ for sorting). The iterative regularization-based procedure repeats screening over several iterations and additionally fits Lasso models during residualization. If the iterative procedure runs for mmm iterations with a screened subset size on the order of $O(n\log(n))$, the total cost is approximately $O\left(mpn^2\right)$ for the repeated HZ computations, plus the cost of Lasso fits. This is typically subdominant for moderate n and depends on the solver and sparsity. These estimates explain why dimensionality reduction by screening is essential for feasibility in high-dimensional settings.

The select_predictors() function creates a $\hat{D}$ set depending on the user's choice, which includes the most relevant predictors. If option 1 was selected in the GUI, the size of the set is equal to $\left[\frac{n}{\log(n)}\right]$ or the number of predictors. If option 2 was selected, then predictors whose weights exceed cn^{-}, where c and are predetermined, are selected. If option 3 is selected, the user inputs the power of the set himself $(\hat{D})$.

The save_data() function is designed to save the results to a file. At the output, two files are created in the programme folder: result.csv and result2.csv. The Run() function is run from the graphical interface and is designed to generate or load data with their further processing in the HZ-SIS function.

The config.py module is responsible for linking the global variables between all other modules. It contains global variables that are responsible for the sample size, the number of predictors, the number of experiments, etc. There are also two functions: Beta() and Delta(), which calculate β and δ, respectively.

The distribution.py module builds a standard empirical distribution function. The initializer.py module is responsible for creating random data or, if the user already has data, saving it for processing by the program. The csv_read() function reads data from the files that the user has specified in the GUI fields. The generate_sample() function creates random numbers obeying a continuous distribution on the interval of $[-2.5;\ 2.5]$. The generate_int_sample() function creates random integers on the interval of $[-100;\ 100]$. The genereate_model_sample() function models an example of 30 features according to a certain principle.

3.4 *Technical means used*

Programme development and research of the programme on real data are carried out using the terminal server of Novosibirsk State Technical University (NSTU) Math, as it has better characteristics than Personal Computer (PC) does [50]. The server has the following characteristics:

92 compute threads based on a pair of AMD EPYC 7402 processors clocked up to 4 GHz ;

256 GB of Random Access Memory (RAM) with Error Correction Support (ECS);

A pair of 256 GB SATA SSDs for the operating system;

Four 2.5 TB SATA SSDs for user files;

64-bit operating system.

All computations were performed using Python 3.x with the Anaconda distribution. The core dependencies include NumPy, Pandas, SciPy, and scikit-learn, with Lasso regression implemented via the standard scikit-learn library. The versions of the libraries used are consistent with those specified in the official documentation at the time of the study, ensuring compatibility and reproducibility of the reported results.

From a computational perspective, the HZ-SIS procedure requires $O\left(p \cdot n^2\right)$ operations due to pairwise distance computations for each predictor, which is comparable to other kernel-based screening methods. The iterative regularization-based extension introduces additional Lasso regressions, leading to an overall complexity of approximately $O(m \cdot p \cdot n^2)$, where m is the number of iterations. In practice, m remains small, and the screening stage substantially reduces dimensionality, making the approach feasible for moderately large p. The implementation is fully vectorized where possible and demonstrated to run efficiently on high-performance computing infrastructure.

The software was validated using a two-stage procedure. First, correctness was verified on simulated datasets with a controlled correlation structure and known relevant predictors, where the expected screening behavior can be analytically anticipated. Second, external validation was performed based on a real oil-production dataset by reproducing and extending results reported in previously published studies. Consistency of the obtained rankings and selection stability across repeated runs confirmed the correctness of the implementation and the absence of numerical artefacts.

4. Study of predictors of reliable filtering algorithms

4.1 *Research on a model case study*

In this paper, stability of a screening method is understood as the invariance of the selected predictor set under repeated sampling from the same data-generating mechanism. Operationally, we assess stability by repeated experiments (or repeated subsamples) and record selection frequencies of each predictor. A method is considered more stable if its selection frequencies for truly relevant predictors are consistently high across repetitions while the inclusion of redundant/irrelevant predictors remains consistently low. This frequency-based assessment corresponds to a resampling variability perspective commonly used in stability analysis of variable selection.

First, it is possible to simulate a test case. This example helps to investigate the stability of the algorithm when the feature set also includes highly correlated predictors [51, 52].

The relevant features of $x_1^{(m)}$, $x_2^{(m)}$, $x_3^{(m)}$, $x_5^{(m)}$, $x_6^{(m)}$ are modelled as independent random variables with a standard normal distribution. Relevant features of ${x_4}^{(m)}$, ${x_7}^{(m)}$ were found from the relations:

$$x_4^{(m)} = x_3^{(m)} + e_1, \quad x_7^{(m)} = x_5^{(m)} + x_6^{(m)} + e_2,$$

where e_1, e_2 are independent random variables with the standard normal distribution.

The response was defined as

$$y = \sum_{i=1}^{7} x_i^{(m)} + e_3,$$

where e_3 is a normally distributed random variable with zero mathematical expectation and standard deviation equal to 0, 1 independent of e_1, e_2 and of ${x_i}^{(m)}$.

We also add redundant features that are modelled as correlating with the main predictors.

$$x_i^{(r)} = x_i^{(m)} + i, \quad i = 1, \ldots, 7, \quad x_i^{(r)} = x_{i-7}^{(m)} + i, \quad i = 8, \ldots, 14.$$

We add irrelevant predictors as random noise: $\xi_1, \ldots, \xi_5$ are independent random variables with the standard normal distribution. We also include $\varepsilon_1, \ldots, \varepsilon_4$ as noise.

Each random variable is modelled with a volume of 1,000. As a result, the set of attributes from which the selection was made includes:

Relevant predictors from which the response is calculated, ${x_1}^{(m)} = {x_7}^{(m)}$;

Redundant features correlated with relevant features, ${x_1}^{(r)} = {x_{14}}^{(r)}$;

Irrelevant features, $\xi_1, \ldots, \xi_5, \varepsilon_1, \ldots, \varepsilon_4$.

There are 30 attributes in total. 25 experiments are simulated. The number of repetitions was chosen as a compromise between statistical representativeness and computational cost, given that each repetition involves $O\left(pn^2\right)$ screening computations. Preliminary experiments show that increasing the number of repetitions beyond this level does not qualitatively change the observed ranking stability patterns, while substantially increasing computation time. To facilitate data processing, all features can be represented in the form of $i = \overline{0,\ 29}$.

To compare the results with HZ-SIS, we will take an algorithm based on the Pearson's correlation coefficient [53]. This SIS-type baseline is included as a widely used reference, but it has well-known shortcomings that motivate robust alternatives. First, Pearson correlation reflects only linear marginal association and can miss nonlinear dependence patterns. Second, correlation-based ranking is sensitive to outliers and heavy-tailed noise, which may distort scores and reduce selection stability. Third, in the presence of strongly correlated predictors, SIS tends to prioritize redundant variables that carry similar information, which can mask weaker but truly relevant predictors and lead to multicollinearity in downstream models. These limitations clarify the need for HZ-based robust screening and for the proposed correlation-aware iterative refinement. We will find the correlation coefficient for each characteristic X with Y by the formula:

$$r_{xy} = \frac{\operatorname{cov}(X,\ Y)}{\sqrt{s_x^2 s_y^2}},$$

where $\operatorname{cov}(X,\ Y)$ is the covariance of X, and Y, ${s_x}^2$, ${s_y}^2$ are the sample variances.

Then we sort the moduli of correlation coefficients of each trait in a descending order and select 7 traits for each experiment.

Figure 1 shows the frequencies of occurrence of each feature over 25 experiments. The results in Figure 1 show that both HZ-SIS and the Pearson correlation algorithm do not include irrelevant features in the set. In addition, the algorithms rarely include redundant features, but also most of the relevant features are not included in the main set. X3, X6 signs, which were modelled as correlated with other predictors, as well as X0, which is correlated with X10 and X13, X20, which are correlated with X6, are consistently included in the set.

To complement the visual analysis, the selection frequencies can be interpreted in terms of standard screening performance metrics. Since the set of truly relevant predictors is known in the simulated model, the average selection frequency of relevant predictors corresponds to an empirical True Positive Rate (TPR), while the frequency, with which irrelevant predictors enter the selected set, reflects the False Discovery Rate (FDR). From this perspective, the baseline HZ-SIS demonstrates elevated FDR in the presence of strongly correlated predictors, whereas the iterative regularization-based procedure yields higher TPR for weakly relevant predictors and substantially lower inclusion of irrelevant variables. Therefore, the frequency plots implicitly summarize quantitative screening performance across repetitions.

The dominance of certain predictors in Figure 1 can be explained by the presence of strong correlations between relevant and redundant variables. In such cases, marginal screening methods tend to prioritize predictors that exhibit the strongest indirect association with the response, even if this association is primarily inherited through correlation with

truly relevant variables. As a result, correlated predictors may consistently dominate the selection frequency, while other relevant but less correlated predictors are omitted. This behavior illustrates a well-known limitation of marginal screening approaches in high-dimensional settings, where redundancy can mask true relevance.

To provide a quantitative summary of these results, we additionally interpret selection frequencies as empirical true positive and false positive rates. The relevant predictors, consistently selected across experiments, correspond to high true positive rates, whereas inclusion of redundant or irrelevant predictors reflects false positives. In the considered model example, the HZ-SIS algorithm exhibits elevated false positive rates for predictors, highly correlated with relevant features, whereas the iterative regularization-based procedure substantially reduces this effect, yielding more stable true positive recovery across repeated samples.

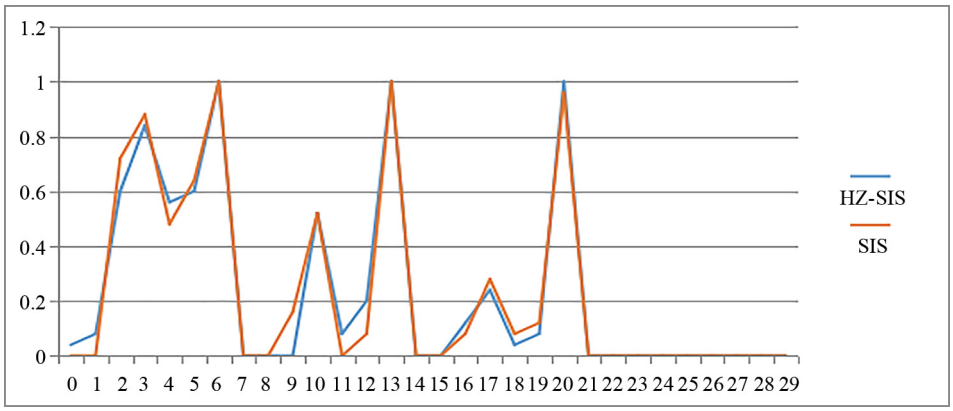

Figure 1. Comparison of frequencies of algorithms in the set of $\hat{D}$

Figure 2 shows that the fraction results are different in different experiments, indicating the instability of the HZ-SIS algorithm. However, in most cases, the algorithm consistently selects one set of predictors.

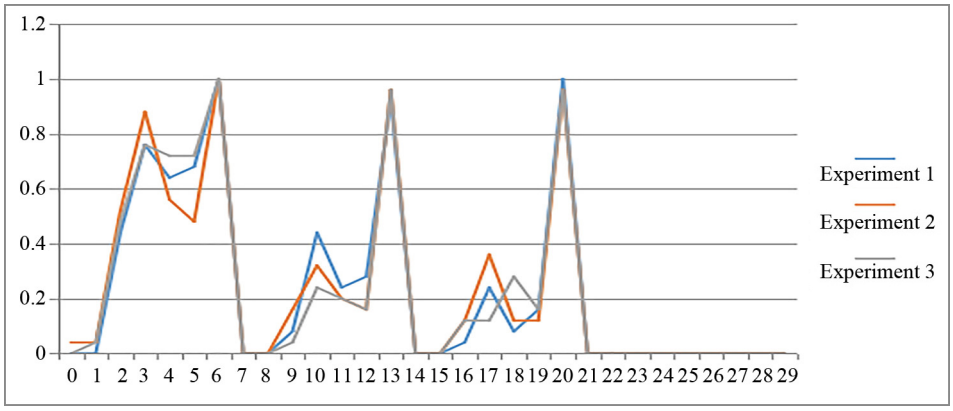

Figure 2. Comparison of results using different samples of the HZ-SIS algorithm

The iterative procedure should be investigated using the same model example. Since the procedure involves Lasso regularization, we will vary the λ parameter described in the formula (5).

The results in Figure 3 show that when $\lambda = 1$, in most cases the first 3 relevant features are selected except the second one. We can also observe that redundant features are almost not selected and irrelevant features are not selected at all. When $\lambda = 0.0001$, the Lasso regression is a regular Minimum Norm Criterion (MNC) and both redundant features and

irrelevant features are selected into the final set. When $\lambda = 10$, the regularization-based procedure behaves almost the same as the conventional HZ-SIS does, which does not optimize the result well.

The iterative regularization-based procedure mitigates this dominance by explicitly removing the linear effects of already selected predictors from the remaining feature space. By applying Lasso regression at each iteration, predictors that are highly correlated with the selected set are effectively residualized and thus lose priority in subsequent screening steps. This mechanism allows the algorithm to progressively reveal weaker but genuinely relevant predictors that would otherwise be overshadowed by redundant variables in a single-pass screening framework. From a practical perspective, this behavior is crucial for high-dimensional screening tasks, as it improves interpretability of the selected feature set and reduces redundancy without requiring explicit pairwise correlation filtering.

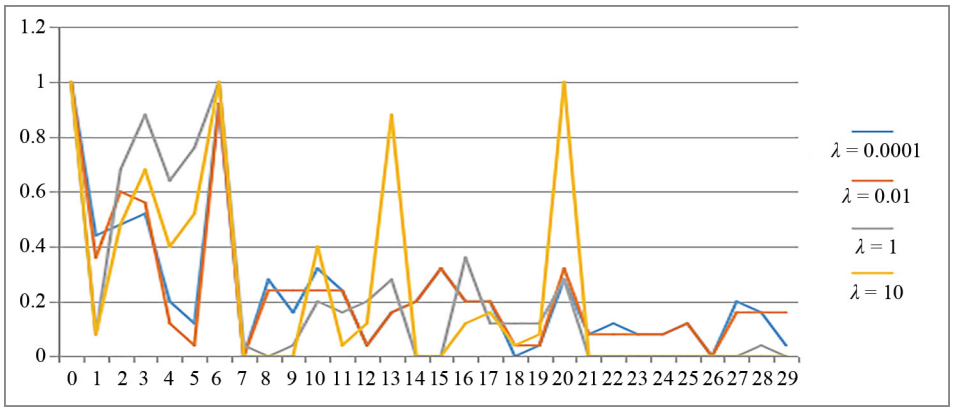

Figure 3. Comparison of results with different parameters of λ

In [54], the authors propose a feature selection algorithm based on correlations. The authors have an optimization problem of linear programming when analyzing the problem, which they propose to solve by the Binary Cut and Branch Method (BCBM). The results obtained in the paper must be compared with the results obtained by the Lasso regularization-based procedure with the regularization parameter of $\lambda = 1$.

The Modified Bootstrap Omitting Voting (MBOV) method presented in Figure 4 has been simulated 1,000 times, while the regularization procedure has been simulated only 25 times. But it can be observed that the frequency of entering the set of relevant features is higher for the Iterative Bootstrap Omitting Voting (IBOV) than that for the iterative procedure.

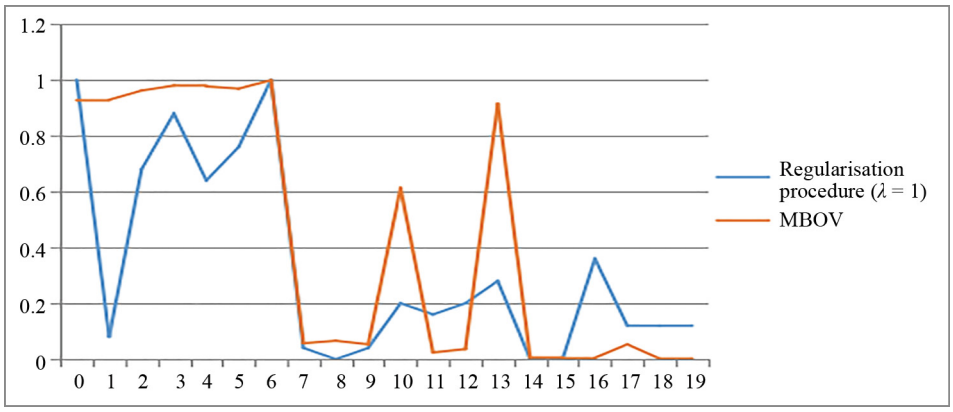

Figure 4. Comparison of results with another algorithm

In general, based on the model example, the HZ-SIS algorithm selects not the relevant features themselves, but those that are highly correlated with them. The algorithm performs almost as well as SIS does and performs worse than the correlation-based algorithm described in [55] does, which selects only 2 features that are correlated with the relevant ones. The Lasso regularization-based iterative procedure performs better, but it is also unstable because it involves direct selection of predictors, i.e. sequential inclusion of each predictor in the set.

Although the empirical comparison focuses on SIS and MBOV-type algorithms, the observed behavior of HZ-SIS and its iterative extension are consistent with known limitations of other model-free screening methods such as DC-SIS, Qa-SIS, and MV-SIS. In particular, these approaches rely on marginal dependence measures that do not explicitly address inter-predictor correlation. This explains their tendency to retain redundant features in settings that are similar to the present model example. The proposed method can thus be viewed as complementary to these screening strategies, with an explicit mechanism for correlation-aware refinement.

These observations highlight the practical implications of the proposed approach for real-world high-dimensional problems, where correlated predictors are ubiquitous. Stable recovery of relevant features in such settings is essential not only for predictive performance, but also for interpretability and subsequent model-based analysis. The empirical results demonstrate that the iterative HZ-SIS procedure provides a more coherent and reliable screening outcome compared to one-step marginal methods.

4.2 *A real-life case study*

In [56], the authors investigate feature selection based on annual oil production data for 2000-2019. In this paper, a sample on oil production associated with a carbonate field and a horizontal well type is studied. Table 2 shows the traits that are needed to investigate this sample.

Table 2. Correspondence of the feature name and the title in studies

Field in the matrix	Column in research
Signs	
"Was under download or transferred under download" 0/1	PUMPING
Breed group	FORMATION
Geological and geophysical data (GGD)	
Absolute permeability (μm^2)	PERMEABILITY
Porosity (d.unit)	POROSITY
Initial oil saturation (d.u.s.)	INIT_SATUR
Initial unsaturated thickness (m)	INIT_NET_PAY
Parameters for the year of the start of the selection	
Area of drainage zone (ha)	DRAINAGE_AREA
Initial oil reserves (thousand tonnes)	INIT_IN_PLACE
Current balance oil reserves (thousand tonnes)	CUR_IN_PLACE
Initial mobile oil reserves (thousand tonnes)	INIT_MOBILE
Current mobile oil reserves (thousand tonnes)	CUR_MOBILE
Waterflood coverage ratio (d.unit)	WATERFLOOD_COEF
Distance to the nearest injection pump (m)	DELIVERY_DISTANCE
Min flow rate of neighbouring wells (tpd)	MIN_DEBIT
Average flow rate of neighbouring wells (tpd)	AVG_DEBIT
Max flow rate of neighbouring wells (tonnes per day)	MAX_DEBIT
Min water cut of neighbouring wells (d.u.)	MIN_WC
Average watercut of neighbouring wells (d.u.)	AVG_WC
Water content of neighbouring wells (d.unit)	MAX_WC

Prior to analysis, the dataset was examined for missing values; observations with incomplete records were excluded, resulting in a fully observed sample. Categorical variables were encoded as binary indicators, consistent with their original representation in the data source. Continuous predictors were used without additional scaling, as the non-paranormal transformation in HZ-SIS standardizes marginal distributions. The plausibility of the selected predictors was additionally assessed in consultation with domain specialists in oil production, who confirmed that the retained features correspond to physically and operationally meaningful factors influencing production performance.

In [**?**], the authors use a filtering algorithm based on the construction of Bayesian Neural Networks (BNNs). We use these research results, for comparison with the HZ-SIS algorithm and the iterative procedure.

The results in Table 3 show that all three algorithms select 5 identical features. However, the HZ-SIS algorithm also selects features that the Bootstrap Noise Screening (BNS)-based algorithm does not select. If we compare it with the iterative procedure, the feature matches with the BNS are 7 out of 10. This means that the iterative procedure has improved the results of the HZ-SIS algorithm and selects features that are more relevant.

Table 3. Comparison of attributes included in the set

Feature number	BNS	HZ-SIS	Iterative procedure
1	PUMPING	PUMPING	PUMPING
2	FORMATION	FORMATION	FORMATION
3	PERMEABILITY	DRAINAGE_AREA	DRAINAGE_AREA
4	INIT_SATUR	PERMEABILITY	PERMEABILITY
5	INIT_NET_PAY	AVG_DEBIT	DELIVERY_DISTANCE
6	DRAINAGE_AREA	MAX_DEBIT	AVG_DEBIT
7	INIT_IN_PLACE	DELIVERY_DISTANCE	INIT_SATUR
8	CUR_IN_PLACE	MIN_WC	INIT_IN_PLACE
9	AVG_DEBIT	MIN_DEBIT	INIT_MOBILE
10	MAX_DEBIT	AVG_WC	MIN_WC

The results must be compared for the selected features. We will vary the regularization parameter in Lasso. We can notice that in this example in Table 4, the regularization parameter does not change much in the feature selection. This shows that this example is robust to different Lasso regularization parameters.

Table 4. Comparison of results for different parameters (λ) for the oil production example

Feature number	$\lambda = 10^{e-10}$	$\lambda = 0.0001$	$\lambda = 10$
1	PUMPING	PUMPING	PUMPING
2	INIT_IN_PLACE	INIT_IN_PLACE	MIN_WC
3	FORMATION	FORMATION	FORMATION
4	AVG_DEBIT	AVG_DEBIT	INIT_SATUR
5	DRAINAGE_AREA	DRAINAGE_AREA	DRAINAGE_AREA
6	PERMEABILITY	PERMEABILITY	PERMEABILITY
7	DELIVERY_DISTANCE	DELIVERY_DISTANCE	DELIVERY_DISTANCE
8	INIT_SATUR	INIT_SATUR	AVG_DEBIT
9	INIT_MOBILE	INIT_MOBILE	MIN_DEBIT
10	MIN_WC	MIN_WC	AVG_WC

Table 5 shows the results of predictor selection based on different algorithms. HZ-SIS and Lasso select PUMPING and FORMATION features first. This is because there are few unique values (only 0 and 1) in the samples for these

features. Moreover, the number of zeros in these attributes exceeds 95% of the entire sample. When calculating the weights for these features, the values are large and therefore the algorithm selects them into a finite set. However, in practice, it is rarely necessary to select features with a large number of zero values.

Table 5. Comparison of the first 10 selected features of different algorithms

Feature number	HZ-SIS	Lasso	SIS
1	PUMPING	PUMPING	AVG_DEBIT
2	MIN_WC	INIT_IN_PLACE	MIN_WC
3	MIN_DEBIT	INIT_MOBILE	AVG_WC
4	FORMATION	FORMATION	MAX_DEBIT
5	DRAINAGE	DRAINAGE_AREA	MIN_DEBIT
6	PERMEABILITY	PERMEABILITY	INIT_SATUR
7	AVG_DEBIT	DELIVERY_DISTANCE	INIT_NET_PAY
8	MAX_DEBIT	AVG_DEBIT	DELIVERY_DISTANCE
9	DELIVERY_DISTANCE	INIT_SATUR	PERMEABILITY
10	AVG_WC	MIN_WC	MAX_WC

Tables 6 and 7 show the correlation coefficients of highly correlated features. It can be seen that the SIS algorithm selects the highly correlated features (AVG_DEBIT, MAX_DEBIT, MIN_DEBIT) first. The HZ-SIS algorithm also includes these features in the set, but they are not first. Including all correlated features in the set affects the predicted response value and is an incorrect result. At the same time, the iterative regularization procedure does not include the features correlated with each other in the set, which gives a good result. From this we can conclude that this procedure is the most robust to the occurrence of highly correlated predictors.

Table 6. Correlation coefficients for some traits

	MIN_WC	AVG_WC	MAX_WC
MIN_WC	1	0.8230003498515732	0.5307901249762745
AVG_WC	0.8230003498515732	1	0.8963299607505395
MAX_WC	0.5307901249762745	0.8963299607505395	1

Table 7. Correlation coefficients for some attributes

	INIT NET PAY	INIT IN PLACE	CUR IN PLACE	INIT MOBILE	CUR MOBILE
INIT_NET_PAY	1	0.795502768559	0.778881224989	0.78494913049	0.74455089566
INIT_IN_PLACE	0.795502768559	1	0.996496188161	0.97576076291	0.95799844298
CUR_IN_PLACE	0.778881224989	0.996496188161	1	0.97358097098	0.96722195776
INIT_MOBILE	0.784949130498	0.975760762918	0.973580970982	1	0.98900921226
CUR MOBILE	0.744550895663	0.957998442989	0.967221957766	0.98900921226	1

5. Conclusion

In the work, two algorithms were implemented. The second procedure is an improvement of the first algorithm, which screens out highly correlated predictors. A comparative analysis with existing filtering algorithms was carried out,

and the stability of the Henze-Zirkler robust filtering algorithm and the Lasso regularization-based iterative procedure was investigated. Specifically, we compared a Pearson correlation-based SIS baseline as a standard marginal screening method, a Bayesian Neural Network as the (BNN)-based screening approach reported in the referenced oil-production study, and a correlation-driven optimization-based selector (MBOV/IBOV-type), used in the simulated setting. The empirical results show that HZ-SIS improves robustness to outliers/heavy tails compared to SIS, while the proposed Lasso-based iterative extension further reduces redundancy by suppressing highly correlated predictors. This leads to the appearance of more stable and interpretable selected subsets on both the simulated correlated-design example and the real oil-production data.

The results obtained demonstrate the effectiveness of the proposed HZ-SIS screening procedure and its Lasso-based iterative extension for data with correlated predictors and heavy-tailed noise. This conclusion is supported by experiments on a controlled simulated example and on a real oil-production dataset, where improved stability and reduced redundancy of the selected predictors were observed in comparison with baseline screening methods. In addition, the developed software implementation enabled reproducible validation of the proposed approach on both model and real data, confirming its practical applicability within the scope considered in this study.

Algorithms were developed and the stability of these algorithms was investigated using examples. The following conclusions were drawn as a result of this research. The HZ-SIS algorithm is very sensitive to correlated features. The correlated features will be selected more often without selecting other relevant predictors. The HZ-SIS algorithm is also sensitive if there are many repeated values in the samples. Then the values of feature weights become large and are selected as relevant. The iterative procedure based on Lasso regularization performs more stable as compared to HZ-SIS. In addition, the iterative procedure is robust to correlated features and to varying the regularization parameter. The iterative procedure is robust to outliers and shows stable results even for samples with heavy tails.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Cui H, Li R, Zhong W. Model-free feature screening for ultrahigh dimensional discriminant analysis. *Journal of the American Statistical Association*. 2015; 110(510): 630-641. Available from: https://doi.org/10.1080/01621459.2014.920256.

[2] Fan J, Lv J. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 2008; 70(5): 883-911. Available from: https://doi.org/10.1111/j.1467-9868.2008.00674.x.

[3] Fan J, Song R. Sure independence screening in generalised linear models with NPdimensionality. *The Annals of Statistics*. 2010; 38(6): 3567-3604. Available from: https://doi.org/10.1214/10-AOS798.

[4] Henze N, Zirkler B. A class of invariant consistent tests for multivariate nor-mality. *Communications in Statistics-Theory and Methods*. 1990; 19(10): 3595-3617. Available from: https://doi.org/10.1080/03610929008830400.

[5] He X, Wang L, Hong HG. Correction: Quantile-adaptive model-free variable screening for high-dimensional heterogeneous data. *The Annals of Statistics*. 2013; 41(1): 342-369. Available from: https://doi.org/10.1214/13-AOS1087.

[6] Li R, Zhong W, Zhu L. Feature screening via distance correlation learning. *Journal of the American Statistical Association*. 2012; 107(499): 1129-1139. Available from: https://doi.org/10.1080/01621459.2012.695654.

[7] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*. 2011; 12(85): 2825-2830.

[8] Pudjihartono N, Fadason T, Kempa-Liehr AW, O'Sullivan JM. A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics*. 2022; 2: 927312. Available from: https://doi.org/10.3389/fbinf.2022.927312.

[9] Wang M, Fu W, Hao S, Tao D, Wu X. A survey on large-scale machine learning. *IEEE Transactions on Knowledge and Data Engineering*. 2022; 34(6): 2574-2594. Available from: https://doi.org/10.1109/TKDE.2020.3011295.
[10] Sarker IH. Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*. 2021; 2: 160. Available from: https://doi.org/10.1007/s42979-021-00592-x.
[11] Xue J, Liang F. A robust model-free feature screening method for ultrahigh-dimensional data. *Journal of Computational and Graphical Statistics*. 2017; 26(4): 803-813. Available from: https://doi.org/10.1080/10618600.2017.1328364.
[12] Peña D, Yohai VJ. A review of outlier detection and robust estimation in high-dimensional data. *Statistical Methods and Applications*. 2023; 32: 1-25. Available from: https://doi.org/10.1016/j.ecosta.2023.02.001.
[13] Liu JY, Zhong W, Li RZ. A selective overview of feature screening for ultrahigh-dimensional data. *Science China Mathematics*. 2015; 58(10): 2033-2054. Available from: https://doi.org/10.1007/s11425-015-5062-9.
[14] Henze N, Zirkler B. A class of invariant consistent tests for multivariate normality. *Communications in Statistics-Theory and Methods*. 1990; 19(10): 3595-3617.
[15] Li R, Zhong W, Zhu L. Feature screening via distance correlation learning. *Journal of the American Statistical Association*. 2012; 107(499): 1129-1139. Available from: https://doi.org/10.1080/01621459.2012.695654.
[16] Chen X, Chen X, Wang H. Robust feature screening for ultra-high dimensional right censored data via distance correlation. *Computational Statistics & Data Analysis*. 2018; 119: 118-138. Available from: https://doi.org/10.1016/j.csda.2017.10.004.
[17] Wang T, Zheng L, Li Z, Liu H. A robust variable screening method for high-dimensional data. *Journal of Applied Statistics*. 2017; 44(10): 1839-1855. Available from: https://doi.org/10.1080/02664763.2016.1238044.
[18] Bühlmann P, van de Geer S. High-dimensional inference in misspecified linear models. *Electronic Journal of Statistics*. 2015; 9: 1449-1473. Available from: https://doi.org/10.1214/15-EJS1041.
[19] Zhu LP, Li L, Li R, Zhu LX. Model-free feature screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*. 2011; 106(496): 1464-1475. Available from: https://doi.org/10.1198/jasa.2011.tm10563.
[20] Young G. High-dimensional statistics: A non-asymptotic viewpoint, Martin J. Wainwright, Cambridge University Press, 2019. *International Statistical Review*. 2020; 88(1): 258-261. Available from: https://doi.org/10.1111/insr.12370.
[21] Kong J, Wang S, Wahba G. Using distance covariance for improved variable selection with application to learning genetic risk models. *Statistics in Medicine*. 2015; 34(10): 1708-1720. Available from: https://doi.org/10.1002/sim.6441.
[22] Azizyan M, Singh A, Wasserman L. Density-sensitive semisupervised inference. *Annals of Statistics*. 2013; 41(2): 751-771. Available from: https://doi.org/10.1214/13-AOS1092.
[23] Qu L, Wang X, Sun L. Variable screening for varying coefficient models with ultrahigh-dimensional survival data. *Computational Statistics & Data Analysis*. 2022; 172: 107498. Available from: https://doi.org/10.1016/j.csda.2022.107498.
[24] Zhang J, Wang Q, Wang X. Surrogate-variable-based model-free feature screening for survival data under the general censoring mechanism. *Annals of the Institute of Statistical Mathematics*. 2022; 74(2): 379-397. Available from: https://doi.org/10.1007/s10463-021-00801-7.
[25] Fan J, Feng Y, Song R. Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*. 2011; 106(494): 544-557. Available from: https://doi.org/10.1198/jasa.2011.tm09779.
[26] Székely GJ, Rizzo ML, Bakirov NK. Measuring and testing dependence by correlation of distances. *Annals of Statistics*. 2007; 35(6): 2769-2794. Available from: https://doi.org/10.1214/009053607000000505.
[27] Barut E, Fan J, Verhasselt A. Conditional sure independence screening. *Journal of the American Statistical Association*. 2016; 111(515): 1266-1277. Available from: https://doi.org/10.1080/01621459.2015.1092974.
[28] Zambom AZ, Matthews GJ. Sure independence screening in the presence of missing data. *Statistical Papers*. 2021; 62(2): 817-845. Available from: https://doi.org/10.1007/s00362-019-01115-w.
[29] Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*. 1996; 58(1): 267-288. Available from: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.
[30] Wang H. Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*. 2009; 104(488): 1512-1524. Available from: https://doi.org/10.1198/jasa.2008.tm08516.

[31] Zhao SD, Li Y. Principled sure independence screening for Cox models with ultra-high-dimensional covariates. *Journal of Multivariate Analysis*. 2012; 105(1): 397-411. Available from: https://doi.org/10.1016/j.jmva.2011.08.002.

[32] Liu H, Lafferty J, Wasserman L. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*. 2009; 10: 2295-2328.

[33] Hoff PD. Extending the rank likelihood for semiparametric copula estimation. *Annals of Applied Statistics*. 2007; 1(1): 265-283. Available from: https://doi.org/10.1214/07-AOAS107.

[34] Mecklin CJ, Mundfrom DJ. A Monte Carlo comparison of the type I and type II error rates of tests of multivariate normality. *Journal of Statistical Computation and Simulation*. 2005; 75(2): 93-107. Available from: https://doi.org/10.1080/0094965042000193233.

[35] Liu H, Fang H, Ming Y, Lafferty J, Wasserman L. High-dimensional semiparametric Gaussian copula graphical models. *Annals of Statistics*. 2012; 40(4): 2293-2326. Available from: https://doi.org/10.1214/12-AOS1037.

[36] Chang J, Tang C, Wu Y. Marginal empirical likelihood and sure independence feature screening. *Annals of Statistics*. 2013; 41: 2123-2148. Available from: https://doi.org/10.1214/13-AOS1139.

[37] Xu C, Chen J. The sparse MLE for ultrahigh-dimensional feature screening. *Journal of the American Statistical Association*. 2014; 109(507): 1257-1269. Available from: https://doi.org/10.1080/01621459.2013.879531.

[38] Werneck RO, Prates R, Moura R, Gonçalves MM, Castro M, Soriano-Vargas A, et al. Data-driven deep-learning forecasting for oil production and pressure. *Journal of Petroleum Science and Engineering*. 2022; 210: 109937. Available from: https://doi.org/10.1016/j.petrol.2021.109937.

[39] Bühlmann P, van de Geer S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Berlin: Springer; 2011.

[40] Dormann CF, Elith J, Bacher S, Buchmann C, Carl G, Carré G, et al. Collinearity: A review of methods to deal with it and a simulation study evaluating their performance. *Ecography*. 2013; 36(1): 27-46. Available from: https://doi.org/10.1111/j.1600-0587.2012.07348.x.

[41] Guyon I, Elisseeff A. An introduction to variable and feature selection. *Journal of Machine Learning Research*. 2003; 3: 1157-1182.

[42] Drukker DM, Liu D. Posis: Command for the sure-independence-screening Neyman orthogonal estimator. *Stata Journal*. 2025; 25(3): 561-586. Available from: https://doi.org/10.1177/1536867X251365455.

[43] Fan J, Samworth R, Wu Y. Ultrahigh dimensional variable selection: Beyond the linear model. *Journal of Machine Learning Research*. 2009; 10: 2013-2038.

[44] Bühlmann P, Kalisch M, Meier L. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*. 2014; 1: 255-278. Available from: https://doi.org/10.1146/annurev-statistics-022513-115545.

[45] Tibshirani R, Hastie T, Friedman J. Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software*. 2010; 33(1): 1-22. Available from: https://doi.org/10.18637/jss.v033.i01.

[46] Tibshirani R. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 1996; 58(1): 267-288. Available from: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[47] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. *Nature*. 2020; 585: 357-362.

[48] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. 2020; 17(3): 261-272. Available from: https://doi.org/10.1038/s41592-019-0686-2.

[49] Ishwaran H, Kogalur UB, Gorodeski EZ, Minn AJ, Lauer MS. High-dimensional variable selection for survival data. *Journal of the American Statistical Association*. 2010; 105(489): 205-217. Available from: https://doi.org/10.1198/jasa.2009.tm08622.

[50] Zou H. The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association*. 2006; 101(476): 1418-1429. Available from: https://doi.org/10.1198/016214506000000735.

[51] Zhao P, Yu B. On model selection consistency of LASSO. *Journal of Machine Learning Research*. 2006; 7: 2541-2563.

[52] Candès EJ, Tao T. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*. 2007; 35(6): 2313-2351. Available from: https://doi.org/10.1214/009053606000001523.

[53] Bosnić Z, Kononenko I. Comparison of approaches for estimating reliability of individual regression predictions. *Data and Knowledge Engineering*. 2008; 67(3): 504-516. Available from: https://doi.org/10.1016/j.datak.2008.08.001.

[54] Hall P, Li Q, Racine JS. Nonparametric estimation of regression functions in the presence of irrelevant regressors. *Review of Economics and Statistics*. 2007; 89(4): 784-789. Available from: https://doi.org/10.1162/rest.89.4.784.

[55] Al-Anazi AF, Gates ID. Support vector regression for porosity prediction in a heterogeneous reservoir: A comparative study. *Computers & Geosciences*. 2010; 36(12): 1494-1503. Available from: https://doi.org/10.1016/j.cageo.2010.03.022.

[56] Magris M, Iosifidis A. Bayesian learning for neural networks: An algorithmic survey. *arXiv:2211.11865*. 2022. Available from: https://doi.org/10.48550/arXiv.2211.11865.

Appendix A. Program code

The above code represents the basic algorithms implemented. The experiments reported in this paper can be reproduced by following these steps. 1. Install the required Python environment and dependencies. 2. Initialize the configuration parameters specified in the config.py module, including the sample size, the number of predictors, and regularization parameters. 3. Generate synthetic or load real data using the initializer module. 4. Execute the Run() function to perform HZ-SIS and the iterative regularization-based screening. 5. Analyze the output files containing selected predictors and corresponding weights. All algorithmic components required for replication are provided in this appendix.

Algorithm 1: HZ-SIS (Henze-Zirkler Sure Independence Screening)

Input: data matrix $X \in R^{n\times p}$, response vector $Y \in R^n$, truncation parameter δ, smoothing parameter β, target size d.

Step 1 (Non-paranormal transform): For Y and each predictor X_k, compute truncated empirical CDF with truncation δ, then transform values by Gaussian quantile function F^{-1}.

Step 2 (HZ scoring): For each predictor $k = 1, \ldots, p$, compute the Henze-Zirkler statistic w_k on the bivariate sample $(T(Y), T(X_k))$ using β.

Step 3 (Screening): Sort predictors by w_k in descending order and return the top ddd predictors (or those above the threshold rule used in the paper).

Output: ranked predictor list and weights $\{w_k\}$.

Algorithm 2: Iterative HZ-SIS with Lasso-based residualization

Input: X, Y, parameters of Algorithm 1, Lasso penalty λ, iteration limit mmm, target total selected size d.

Step 1: Run Algorithm 1 on (X, Y) to obtain an initial selected set $A(1)$.

Step 2 (Residualize remaining predictors): For each predictor $X_j \notin A^{(1)}$, fit a Lasso regression of X_j on predictors in $A^{(1)}$ using penalty λ; compute residuals $R_j = X_j - X_j$.

Step 3: Apply Algorithm 1 to the residual matrix R with the same response Y to select a new subset $A^{(2)}$.

Step 4: Update the selected set $A \leftarrow A \cup A^{(2)}$. Repeat Steps 2-4 until $|A| \geq d$ or the maximum number of iterations is reached.

Output: final selected set A and associated screening weights.

```
import math import numpy as np import pandas as pd import config
from scipy.stats import norm from sklearn.linear_model import Lasso from distribution import ECDF
from sorting import shellSort
from initializer import csv_read, generate_sample, generate_int_sample, generate_model_sample
def trim_up(x): if (x < config.delta): return config.delta return x
def trim_down(x): if (x > 1 - config.delta): return 1 - config.delta return x
#1 def empirical_func(sample): ordered_sample = sorted(sample)
  #Build the empirical distribution function (ECDF) ecdf = ECDF(ordered_sample)
  #To truncate the top and bottom EFRs
  F_tilda = list(map(trim_up, ecdf))
  F_tilda = list(map(trim_down, F_tilda))
  #Find quantiles of Gaussian distribution temp = [norm.ppf(x) for x in F_tilda]
  ratio = dict(zip(ordered_sample, temp))
  #Find a T with a tilde, preserving the original order of sampling
  T_tilda = [] for i in range(0, len(sample)):
T_tilda.append(ratio[sample[i]])
  return T_tilda
#2 Calculating statistics def calculate_w(X, Y): a1, a2, a3 = 0, 0, 0,  0 n = len(X)  for  i  in range(n): for  j  in
range(n): a1 += math.pow(math.e, -config.beta * config.beta/2*((X[i] - X[j])**2 + (Y[i] - Y[j])**2)))
  for i in range(n): a2 += math.pow(math.e, -config.beta * config.beta / (2*(1 + config.beta**2))* (X[i]**2 + Y[i]**2)))
  a3 = 1 / (1 + 2*config.beta**2)
```

```
    return (1/n**2) * a1 - 2/(n * (1 + config.beta**2)) * a2 + a3
  def select_predictors(w, X, fname): n = X.shape[0] p = X.shape[1] D = [0] * config.p
    if(config.mode == 1): if(len(w) < n * math.log(n)):
        threshold = len(w) else:
         threshold = round(n * math.log(n)) for i in range(threshold): D[int(X.columns[i][1:])] += 1 elif(config.mode
== 2):
    c = 20
    k = 3 check = c * math.pow(n, -k) for i in range(p): if (w[i] > check): D[int(X.columns[i][1:])] += 1 else:
    threshold = config.D_length for i in range(threshold): D[int(X.columns[i][1:])] += 1
  with open(fname, 'a') as file: for i in X.columns: file.write(i + ' ') file.write('\n') for i in w: file.write(str(i) + ' ')
file.write('\n')
  return D
  def HZ_SIS(X, Y): n = X.shape[0] p = X.shape[1]
    #1
    T_x = X.copy() for i in X.columns: T_x[i] = empirical_func(T_x[i])
    T_y = empirical_func(Y)
    #2
    w = [] for i in X.columns:
    w.append(calculate_w(T_x[i], T_y))
    w, result = shellSort(w, X)
    #3
    return w, result
  def LRL(X, weights, result_y):
     #threshold =  round (X .  shape[0] / math.log (X .  shape[0])) #if (threshold >X .  shape[1]):
     # threshold = X.shape[1]
     limit =  math.ceil(threshold/2)
     #flag = limit
     limit =  1  flag = limit
     #1 Choose predictors for the set A
     A = X.iloc[:, 0: limit].copy()
     A_ = X.iloc[:, limit:].copy()
     w = weights[:limit] threshold = config.D_length while flag < threshold: #limit = math.ceil(limit / 2) flag  += limit
#if flag > threshold:
       # limit = flag - threshold
       data = [] #2 for i in range(A_.shape[1]):
  x = A.copy()
         y = A_.iloc[:, i : i+1 ].copy()
         lasso = Lasso(alpha =0.0000000001 , tol =0.001  ).fit (  x, y  )
        y _predict =  lasso.predict (x)
         temp =[]  for  j  in range(len(y)): temp.append((y.values[j][0] - y_predict[j])) data.append(temp)
           #3
       result_x = pd.DataFrame(data, dtype = float).T result_x.columns = A_.columns
      w_, dataframe = HZ_SIS(result_x, result_y) for i in range(round(limit)): w.append(w_[i])
      for i in range(limit):
      A = pd.concat([A, X[dataframe.columns[i]]], axis=1)
      A_ = pd.DataFrame() for i in range(limit, dataframe.shape[1]):
        A_ = pd.concat([A_, X[dataframe.columns[i]]], axis=1)
    D = select_predictors(w, A, 'weights2.csv') save_data(w, A, 'result2.csv') return D
```

```
def save_data(w, X, fname): with open(fname, ′w′) as file: for i in X.columns: file.write(i + ′ ′) file.write(′\n′) for i
in w:
        file.write(str(i) + ′ ′) file.write(′\n\ n\n′) for i in range(X.shape[0]): for j in X.values[i]:
    file.write(str(j) + ′ ′)
        file.write(′\n′)
    def CorrPirson(X, Y): r = np.corrcoef(X.T, Y.T) return r[0][1]
    def Run(_self = 0):
      if _self == 1:
    = csv_read(config.NameX, 1)
    = csv_read(config.NameY, 2)
    config.N = X.shape[0] config. p = X.shape[1] config.M = 1
    config.delta = config.Delta(config.N)
    config.beta = config.Beta(config.N)
      P = [0] * config.p
      P1 = [0] * config.p
      for i in range(0, config.M): if _self == 0:
          X, Y=generate_model_sample()
      #Basic algorithm w, dataframe = HZ_SIS(X, Y)
      D = select_predictors(w, dataframe, ′weights1.csv′)
      save_data(w, dataframe, ′result1.csv′)
       #Auxiliary algorithm
       D1 =LRL(X, w, Y)
       for j in range(config.p):
    if(D[j]==1):     P[j]+=1  for  j  in range(config.p):  if(D 1[j]==1)：
          P1[j] +=  1
      for i in range(config.p):
    P[i] /=config. M
        P1[i] /= config.M
      with open(′probs.txt′, ′w′) as file:
        for i in P: file.write(str(i) + ′ ′) file.write(′\n′) for i in P1: file.write(str(i) + ′ ′)
        file.write(′\n′)
      return  P
```