A comparison of global sensitivity analysis methods for explainable AI with an application in genomic prediction

Stein, B. van; Raponi, E.; Sadeghi, Z.; Bouman, N.; Ham, R.C.H.J. van; Bäck, T.H.W.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 August 2022, accepted 19 September 2022, date of publication 26 September 2022, date of current version 4 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3210175

RESEARCH ARTICLE

A Comparison of Global Sensitivity Analysis Methods for Explainable AI With an Application in Genomic Prediction

BAS VAN STEIN[1], (Member, IEEE), ELENA RAPONI[2,3], ZAHRA SADEGHI[1], NIEK BOUMAN[4], ROELAND C. H. J. VAN HAM[4], AND THOMAS BÄCK[1], (Fellow, IEEE)

[1]Leiden Institute of Advanced Computer Science, Leiden University, 2333 CA Leiden, The Netherlands
[2]TUM School of Engineering and Design, Technical University of Munich, 80333 Munich, Germany
[3]LIP6 Department, Sorbonne Université, 75005 Paris, France
[4]Keygene N.V., 6708 PW Wageningen, The Netherlands

Corresponding author: Bas van Stein (b.van.stein@liacs.leidenuniv.nl)

This work was supported in part by the Project XAIPre of the Research Program Smart Industry 2020 through the Dutch Research Council [Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)] under Project 19455.

ABSTRACT Explainable Artificial Intelligence (XAI) is an increasingly important field of research required to bring AI to the next level in real-world applications. Global sensitivity analysis (GSA) methods play an important role in XAI, as they can provide an understanding of which (groups of) parameters have high influence in the predictions of machine learning models and the output of simulators and real-world processes. In this paper, we conduct a survey into global sensitivity methods in an XAI context and present both a qualitative and a quantitative analysis of these methods under different conditions. In addition to the overview and comparison, we propose an open source application, GSAreport, that allows you to easily generate extensive reports using a carefully selected set of global sensitivity analysis methods depending on the number of dimensions and samples, to gain a deep understanding of the role of each feature for a given model or data set. We finally present the methods discussed in a complex real-world application of genomic prediction and draw conclusions about when to use which GSA methods.

INDEX TERMS Explainable artificial intelligence, global sensitivity analysis, machine learning, plant breeding, genomic prediction.

I. INTRODUCTION

Sensitivity analysis (SA) methods aim to measure the uncertainty in output based on the change in the input. SA methods are a key technology in better understanding the influence and uncertainty of features or parameters in machine learning models, simulators and real-world applications. As SA can potentially give a lot of insights in machine learning models and applications, it can be considered as a model agnostic approach to explainable AI (XAI). Explainable AI gets an increasingly important role in AI research and applications as simply providing inference is in most cases not sufficient anymore. Explanations on how a machine learning model works, what parameters play a role in the prediction and what uncertainties are incorporated in the data and the model are just a few questions that domain experts and engineers want an answer to before deploying AI methods to production. SA plays an important role in the field of XAI as it can answer a part of these questions without being dependent on a certain machine learning model or even a sampling strategy. Especially when the number of model inputs is large, recognizing the factors on which to focus resources in data collection and data-driven modeling efforts becomes crucial. SA is used in many real-world applications, including but not limited to: understanding how chemical models work [1], measuring the influence of input parameters on biological models [2], and understanding and analyzing factors affecting CO2 emissions in the construction industry [3]. SA methods can be divided into *global* and *local* SA methods. Global approaches focus on the variation of all inputs, leading to

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

an overall analysis of the importance of each feature, while local approaches study the local variations around some local points. Global SA (GSA) methods can account for interactions between variables and do not depend on the choice of a nominal point. Local SA (LSA) methods are typically used to explain a particular prediction, while GSA methods are used to explain a model in its entirety. SA approaches can be divided into derivative-based methods, variance-based methods, and density-based methods.

Most XAI methods such as the model agnostic LIME [4] and Shapley values [5] fall in the local method category, as well as many other XAI methods that are model dependent such as Saliency Maps [6], adversarial examples [7], etc. Shapley values and LIME and some other methods, where sets of inputs are switched on and off in order to find which features contribute most to a specific prediction, use the same underlying principles as the much more established GSA methods.

Several attempts have been made in the literature to provide an overview of available GSA methods [8], [9], [10], but to the authors' knowledge they are limited to a small number of dimensions and compare only a few methods. While many papers suggest how to work with many samples in low dimensions, not much has been done for the high dimensional situation. However, the latter scenario is of paramount importance today, as sensitivity analysis plays a critical role in decision making in businesses and economies bombarded by an ever-increasing amount and dimensionality of data.

In this paper, a wide set of popular and different GSA methods or pseudo-GSA methods (i.e., methods that can be used for GSA without this being their main application) are compared both qualitatively and quantitatively by testing their robustness to different sample sizes and their accuracy to a large number of randomly generated linear functions with different dimensions. Suggestions and a real-world example of using GSA methods on very high-dimensional data are given to help data scientists and machine learning experts use these GSA methods as efficiently as possible. An open source library (*GSAreport*) is proposed, that allows users to create GSA reports for any model or data set, the Figures used in Sections III, IV, V and VI are generated by the proposed software.

The paper is organized as follows: In Section II, an overview of related literature is given. Afterwards, the GSA XAI methods compared in this work are presented and explained: Variance-based methods in Section III, Derivative-based methods in Section IV, Density-based methods in Section V, and Model-based methods in Section VI. Then, in Section VII, the first quantitative experiment is introduced to investigate the robustness of the different methods under different sample sizes and number of dimensions. The second experiment in Section VIII focuses on the accuracy of these methods under different conditions for (relatively simple) randomly generated linear functions. In Section IX, we combine the two previous quantitative results with a qualitative comparison of the different methods. In Section X, we propose the open-source software package GSAreport, which allows to easily use the aforementioned XAI and GSA methods to gain a deep understanding of the feature sensitivities of any model or process. We conclude the work with a detailed example of a real-world application dealing with genomic prediction in Section XI, which shows the applicability of these methods, and a summary in Section XII.

II. GLOBAL SENSITIVITY ANALYSIS

While there are already some excellent reviews of sensitivity analysis methods, such as [8] and [9], none of these reviews compare GSA methods in terms of robustness under different sample sizes and number of dimensions and applicability to explainable AI applications. Most related work is limited to a small number of dimensions (typically less than five) and compares only two or three methods. A recent work [10] proposes a meta-function to benchmark different GSA methods in their ability to find ''ground truth'' sensitivity indices. However, the ''ground truth'' was computed with a large sample size and only allows comparison of specific SA methods, since not all SA methods provide similar information (first-order sensitivity scores) as output, making it difficult to compare these algorithms. When working with a small sample size of high-dimensional data, a different approach may be required than when working with many samples of low-dimensional data. Therefore, this study presents and compares common methods of sensitivity analysis, both qualitatively and quantitatively, to finally provide some rules for which methods are better to use and when, and to extend the analysis to the more complex case of high-dimensional data, for which a sample set of relatively small size is available. The review of SA methods in this paper is limited to *Global* Sensitivity measures. Comparing LSA methods would require a completely different approach that is beyond the scope.

For demonstration and comparison purposes, to explain the different methods of SA, in this paper in Sections III, IV, V and VI we use the transformed, multi-modal Rastrigin function (f_3) from the Black-Box Optimization Benchmarking (BBOB) test suite within the COCO framework [11] as the function we want to analyze.

Figure 1 shows a surface plot of the function with $n = 2$ input parameters. In the examples in Sections III, IV, V and VI, $n = 5$ is used.

The following sections provide a comprehensive list of GSA methods, which are divided into four categories: variance-based, derivative-based, density-based, and model-based GSA methods.

III. VARIANCE-BASED METHODS

Variance-based methods are based on the assumption that variance is sufficient to describe the output uncertainty, an assumption made by Andrea *et al.* [12]. Variance-based methods calculate the sensitivity of the input parameters via an ANOVA-like decomposition of the function.

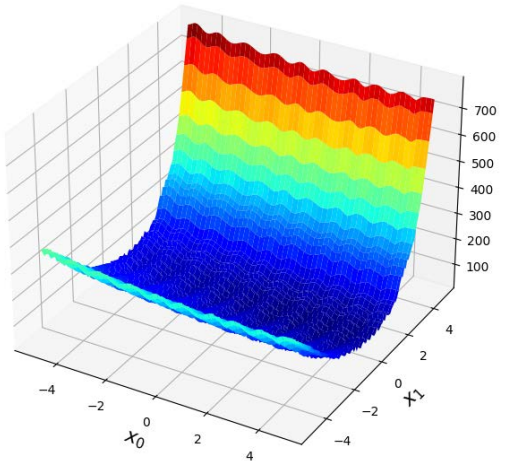

FIGURE 1. Rastrigin function of the BBOB benchmark with $n = 2$. The last parameter (in this case x_1) is the most influential.

A. SOBOL

A very popular and known variance-based SA method is the *Sobol* method [13], also referred to as Sobol indices. The method decomposes the variance of the output of a model or system into fractions that can be attributed to inputs or groups of inputs. The Sobol method can be used to calculate first order sensitivities (the attribution of variance of the output to the variance of a single input (feature)), but also for higher order sensitivities. It is most commonly used for first, second and total order sensitivity analysis.

The Sobol indices are calculated by Equation 1, where $\mathrm{Var}(Y)$ denotes the output variance and the right side of the equation are variance terms decomposed with respect to sets of the input X_i. Since Sobol works with *sets* of the input, it can handle either all individual inputs or groups of inputs.

$$\mathrm{Var}(Y) = \sum_{i=1}^{n} V_i + \sum_{i<j}^{n} V_{ij} + \ldots + V_{12\ldots n}, \tag{1}$$

where

$$V_i = \mathrm{Var}_{X_i}(E_{X_{\sim i}}(Y|X_i)), \tag{2}$$

$$V_{ij} = \mathrm{Var}_{X_{ij}}(E_{X_{\sim ij}}(Y|X_i, X_j)) - V_i - V_j, \tag{3}$$

and $X_{\sim i}$ denotes the set of all input variables except X_i etc.

An example of the output of the Sobol method is shown in Figures 2 and 3. Sobol can calculate the first, second and total order sensitivity of the parameters. With a graph based plot we can bundle this information in one intuitive visualisation.

B. FAST

Fourier Amplitude Sensitivity Test (FAST) [14] and extended FAST (eFAST) [15] are two well established and popular SA methods. The sensitivity value in FAST is defined based on conditional variances which indicate the individual or joint effects of the uncertain inputs on the output. FAST computes the "main effect" contribution of each input factor to the variance of the output (the first order and total order sensitivity). To calculate the sensitivity FAST uses a specific periodic sampling scheme where the number of harmonics to sum in the Fourier series decomposition has to be provided by the user.

For a formal definition of how FAST calculates the sensitivity we refer to [16].

C. RBD-FAST

Random Balance Designs Fourier Amplitude Sensitivity Test (RBD-FAST) [17] does not depend on a specific sampling scheme and works well with a basic Latin Hypercube Sampling (LHS) scheme. The hybrid RBD-FAST method is computationally much more efficient than classical FAST and gives similar performance according to [17].

An example of the first order (S1) sensitivity calculated by using RBD-FAST for f_3 is given in Figure 4.

IV. DERIVATIVE-BASED METHODS

A. MORRIS

Morris method [18] is a qualitative measure that shows several advantages compared to other SA strategies. First of all, it can work directly on discrete domains. Then, it includes multi-dimensional averaging (i.e., the evaluation of the effect of a factor while the others are also varying) and allows grouping of factors, which makes it particularly suitable to work with very large data sets. Finally, the modified version considering absolute means of the distribution of elementary effects is robust with respect to type II errors.

1) STANDARD MORRIS METHOD

Given a random sampling of a vector $\mathbf{X} = (X_1, X_2, .., X_n)$ from an input space Ω, which is a $n-$dimensional $p-$level grid, Morris method generates a distribution F_i of elementary effects for every i^{th} input. If we consider r randomly sampled inputs, and build a trajectory on the input space by changing one parameter at a time by summing or subtracting a noise term Δ, the elementary effect for the i^{th} factor on the j^{th} trajectory is defined as

$$EE_i(\mathbf{X}^j) = \frac{\begin{array}{c}[Y(X_1^j, \ldots, X_{i-1}^j, X_i^j \pm \Delta, X_{i+1}^j, \ldots, X_n^j) \\ -Y(X_1^j, \ldots, X_{i-1}^j, X_i^j, X_{i+1}^j, \ldots, X_n^j)]\end{array}}{\Delta}. \tag{4}$$

Based on this definition, two sensitivity indices are computed for each factor: the mean of the absolute value of the elementary effects

$$\mu_i^* = \frac{1}{r}\sum_{j=1}^{r} |EE_i(\mathbf{X}^j)|, \tag{5}$$

and the standard deviation of the elementary effects

$$\sigma_i = \sqrt{\frac{1}{r}\sum_{j=1}^{r}\left(EE_i(\mathbf{X}^j) - \frac{1}{r}\sum_{j=1}^{r}(EE_i(\mathbf{X}^j))\right)^2}. \tag{6}$$

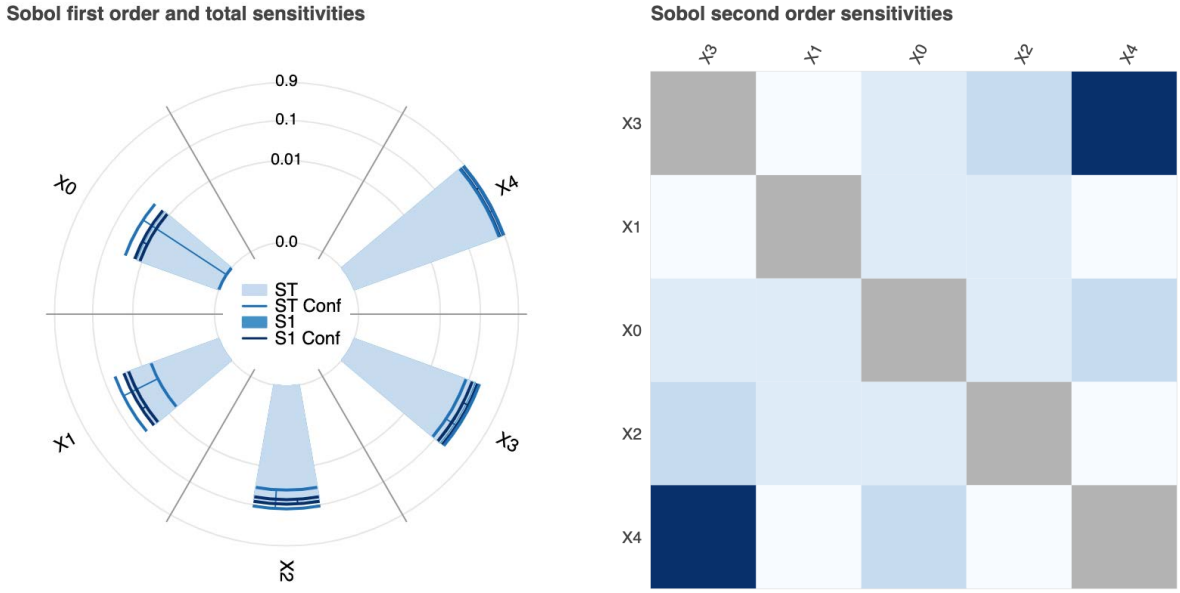

FIGURE 2. Sobol sensitivity indices, on the left, a stacked bar plot of the first (S1) and total sensitivity (ST) for each parameter with the 95% confidence interval for the first (S1 Conf) and total order (ST Conf) respectively as dark blue error bars. On the right the second order sensitivities between two parameters (dark blue is high, white is 0).

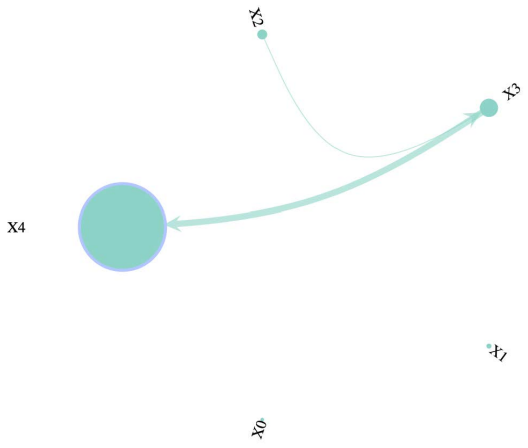

FIGURE 3. Sobol network plot. Each node is a parameter, the size of the node is the total sensitivity index of the parameter, the blue halo around the node is the confidence. The thickness of each edge denotes the secondary interactions.

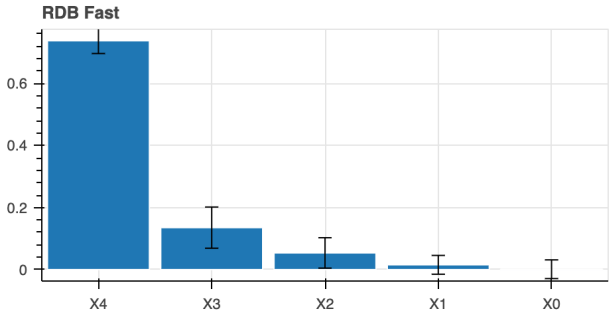

FIGURE 4. RDB Fast first order (S1) sensitivity indices with their confidence as error bars.

The first measures the influence of the i^{th} feature on the output: the larger, the more it contributes to the output variance. The second is a measure of nonlinear and/or interaction effects of the i^{th} feature: a high standard deviation calculated over different trajectories means that the linearity hypothesis between input and output is unlikely. Rather, it indicates non-linear effects on the output due to the i^{th} feature and/or the presence of interactions with other factors. This represents one drawback of Morris method, i.e., it is not possible to distinguish nonlinearity from interaction effects. Results are usually visualized on a σ vs μ^* plot, as in Figure 5. It should be noted that the sensitivity index μ^* belongs to the revised version of the Morris method presented by Campolongo *et al.* [19], which differs from the original definition of μ only in that it presents absolute values. This is to solve the problem that elementary effects of opposite sign cancel each other, which occurs when the model is non-monotonic.

2) MORRIS METHOD WITH GROUPING

Sometimes it is useful to perform sensitivity analysis on groups of input variables to reduce the number of model runs required if the variables belong to the same component of a model or there is reason to believe that they should behave similarly. Morris with groups [19] works similarly to the standard version, with the only difference that all features belonging to the same group are varied simultaneously along a trajectory, and therefore the elementary effects and

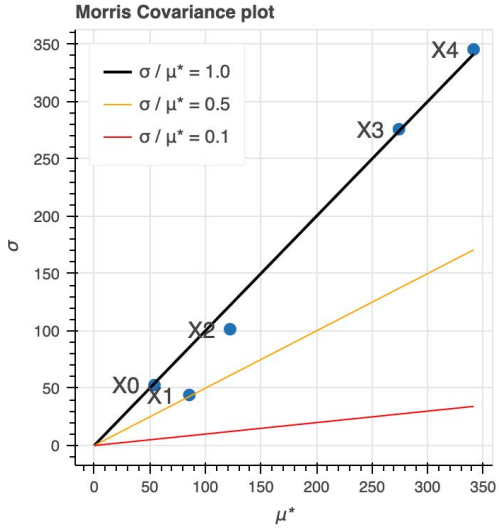

FIGURE 5. Morris σ vs μ^* plane. Plot for sensitivity analysis on five factors for f_3, where two factors seem to be influential, with eventual interaction effects. This plot allows for distinguishing three main categories of factors: non influential (small values of both μ^* and σ), with linear and/or additive effects (relatively large μ^* and relatively small σ), and with nonlinear and/or interaction effects (relatively small μ^* and relatively large σ).

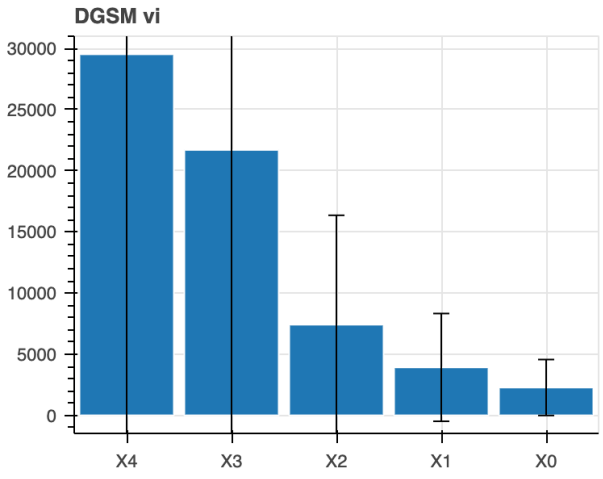

FIGURE 6. DGSM sensitivity analysis on five factors for f_3, where two factors are identified to be the most influential.

sensitivity measures are calculated for each group instead of for each factor:

$$EE_U(\mathbf{X}^j) = \frac{[Y(X_1^j, \ldots, X_{i_1}^j \pm \Delta, X_{i_2}^j \pm \Delta, \ldots, X_k^j) - Y(X_1^j, \ldots, X_k^j)]}{\Delta}, \quad (7)$$

where the features X_{i_1}, X_{i_2}, and X_{i_3} belong to the same group $U = (X_{i_1}, X_{i_2}, X_{i_3})$, and $\mathbf{X}^j$ is the random sample generated to initiate the $j-$th trajectory. Hence, the idea is to move all factors of the same group simultaneously and produce an overall sensitivity measure relative to a group, rather than to a single factor.

B. DGSM

Derivative-based Global Sensitivity Measures (DGSM) [20], [21] can be seen as a the generalization of the Morris method. By averaging local derivatives using Monte Carlo or Quasi Monte Carlo sampling methods, DGSM can be more accurate than the Morris method because the elementary effects are evaluated as strict local derivatives with much smaller increments compared to the Δ term in Morris. In addition, the local derivatives are evaluated at selected points throughout the uncertainty range rather than at points belonging to a fixed grid. Compared to Sobol's sensitivity indices, DGSM is computationally less expensive. Let f be a differential function defined in the unit hypercube H^k. Local sensitivity measures are defined as the limit version of the elementary effects in Morris when $\Delta \longrightarrow 0$:

$$E_i(\mathbf{X}^j) = \frac{\partial f(\mathbf{X}^j)}{\partial x_i}. \quad (8)$$

If $\partial f/\partial x_i \in L_2$, meaning that it is a a square-integrable function, the most used DGSM measure is defined as the mean value of $(\partial f/\partial x_i)^2$:

$$\nu_i = \int_{H^k} \left(\frac{\partial f(\mathbf{X})}{\partial x_i}\right)^2 d\mathbf{X}. \quad (9)$$

The higher the measure ν_i, the more influential the $i-$th factor. DGSM also support grouping of factors. For further details, the reader is referred to [21].

An example of ν_i indices found by DGSM is given in Figure 6.

V. DENSITY-BASED METHODS

Density-based SA methods consider the entire Probability Density Function (PDF) of the model output in order to calculate the sensitivity of the inputs and their interactions. Density-based SA methods are popular for their ability to overcome certain limitations associated with the interpretation of variance-based measures in the presence of dependencies among the model inputs. However, their estimation runs the risk of becoming infeasible when the number of model inputs is large (high dimensionality) or when the computing time of the model or function takes longer than a few minutes. Below we discuss two of the most popular density-based SA methods, DELTA and PAWN.

A. DELTA

The DELTA (δ) [22] method is a Densitity based SA method that is independent of the sampling generation method. The method provides both the first order sensitivity and the δ (similar to the total sensitivity) for each input parameter. DELTA aims at assessing the influence of the entire input distribution on the entire output distribution without reference to a particular moment of the output. The moment independent sensitivity indicator δ for a factor X_i is calculated using

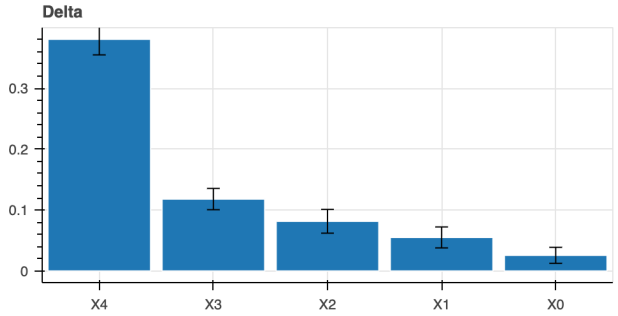

FIGURE 7. δ sensitivities plot, indicating X_4 as most influential.

Equation 10.

$$\delta_i = \frac{1}{2} \int f_{X_i}(x) \left[\int |f_Y(y) - f_{Y|X_i=x}(y)| dy \right] dx, \quad (10)$$

where

- $\mathbf{X}$ is a vector of input factors and $x \in X$;
- X_i is a fixed input variable;
- $f_{X_i}(x)$ is the marginal probability density function of the input factor X_i;
- $f_Y(y)$ the cumulative probability density function of the model output Y;
- $f_{Y|X_i=x}(y)$ is the conditional density of Y given that one of the parameters, X_i, assumes a fixed value.

An example of the calculated δ values for the BBOB f_3 function is presented in Figure 7.

B. PAWN

PAWN [23] is a density-based GSA method (named after the authors) that aims to provide Density-based SA metrics in a more efficient way. The key idea is to characterise output distributions by their Cumulative Distribution Functions (CDF), which are easier to derive than PDFs. An advantage of PAWN is that sensitivity indices can be computed not just over the entire range of variation of the output, but also over a sub-range. This can be useful in applications where one is interested in a specific region of the output distribution. The PAWN method provides not only the mean sensitivity of each parameter but also the minimum, median, maximum and standard deviation of the sensitivity. In Figure 8 the minimum, median and maximum sensitivity index for each parameter of the f_3 function are shown.

VI. MODEL-BASED METHODS

Machine learning models, such as decision trees and linear models can also provide insights in the importance of input features. The most commonly used machine learning models for this purpose are *linear models* and *Random Forests* [24], [25]. Note that providing these insights is not the main purpose of these techniques and comes as a convenient by-product. Each of these ML models also come with a set of assumptions that the data needs to fit to, for example it would not make much sense to use Linear Regression on highly non-linear functions in order to estimate sensitivity of the input features.

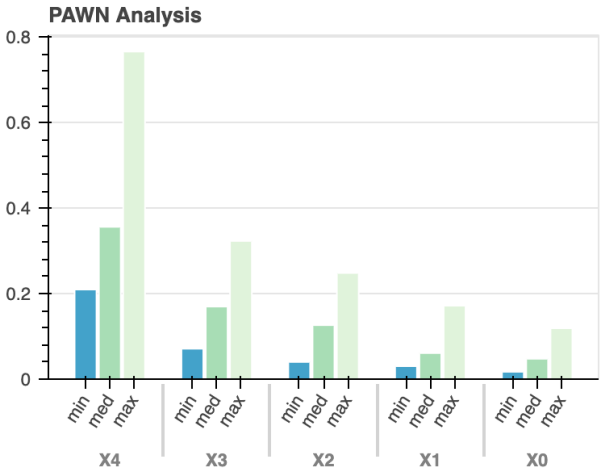

FIGURE 8. PAWN minimum, median and maximum sensitivity index on f_3.

A. LINEAR MODELS

Linear regression is a linear approach for modeling the relationship between the input features and the output. Using multiple linear regression (more than one variable), a set of coefficients of the linear function are fit to the data by minimizing the residual sum of squares between the observed targets and the predicted targets. The learned coefficients immediately represent the relative importances of the input variables. In the experiments in this work we scale the coefficients between 0 and 1 to resemble feature importance indices.

In general the linear regression model fits a function in the form of Equation 11.

$$y = \beta + \sum_{i=1}^{n} \beta_i X_i + \epsilon, \quad (11)$$

where y is the dependent variable, β, is the intercept of the model, X_i corresponds to the i^{th} explanatory variable of the model, and ϵ is the random error with expectation 0 and variance σ^2.

In Equation 12 the function fitted on a Latin Hypercube design of experiments with 512 samples of f_3 is given, showing the largest (negative) coefficient for x_4.

$$y = -142 + 4.3\, X_0 - 7.9 X_1 - 11.3\, X_2 - 0.1\, X_3 - 38.9\, X_4 \quad (12)$$

B. RANDOM FOREST

Random Forest [26] is a very popular Machine Learning technique that learns multiple decision trees on random subsets of input features. The method is quite robust, needs little to no hyper-parameter tuning and can handle mixed-integer data, making it a very flexible and powerful model. As a bonus, using Random Forests one can calculate the variable

importance. The importance of an input feature is computed as the normalized total reduction of the criterion brought by that feature and it is also known as the Gini importance. This importance however is known to be misleading for very high cardinality features (inputs with many unique values). It is also important to note that the Random Forest is not an interpretable model such as decision trees, as looking at individual trees inside a forest can be highly misleading due to the random feature subsets.

C. SHAPLEY AND SHAP

Shapley values is a cooperative game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations. The idea is that the input parameters are different ''players'' in a game (the real function that we are interested in). One of the fundamental properties of Shapley values is that they always sum up to the difference between the game outcome (prediction of a model, output of a function) when all players are present and the game outcome when no players are present.

While Shapley values, and specifically SHapley Additive exPlanations (SHAP) [5] are mostly used to explain single predictions (local SA), they can also be used to generate global SA by averaging the Shapley values over all instances of the data set the model was trained on. Calculating the exact solution of Shapley values is usually infeasible as the computation cost is exponential with the number of features. The approximation of Shapley features by taking a subset of random samples is much faster (though still computationally expensive with many features). The advantage of using SHAP for global feature analysis is that the method not only provides an importance but also the direction of the change that the feature has on the target variable. Unfortunately, SHAP requires both a model of the data and the complete training data set to calculate the Shapley values, making it impractical for large data sets or to analyse a fixed design of experiments (such as in our experimental setup). Due to these limitations we do not include the SHAP method in our experimental comparison. However, when performing the feature analysis of a machine learning model, and especially a tree based model, we can use the TreeSHAP [27] method that exploits the information present in the trees and gives a deep understanding of the features and the features in respect with the training data.

In Figure 9 a summary of the Shapley values for all features over all instances in the data (Latin Hypercube Design of Experiments with 512 samples) is visualised for the f_3 function. This plot immediately makes it clear that x_4 is again the most influential parameter, but also that low values of x_4 seem to contribute towards high values in the function, and the other way around (negative correlation as also detected by the Linear Regression).

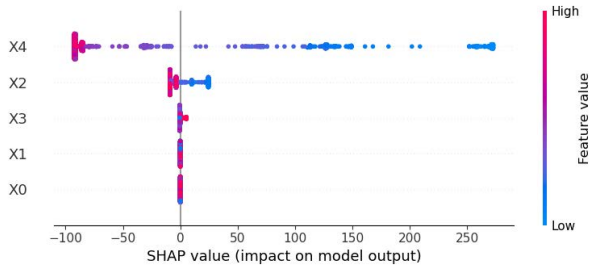

FIGURE 9. SHAP summary plot of feature importances of f_3. Red values correspond to high feature values and blue to low feature values. Each dot shows the SHAP values for an instance in the training data set. In this case it is clear that X_4 has the highest influence on the model outcome.

VII. COMPARISON IN ROBUSTNESS

In the first quantitative experiment, we conduct a large set of experiments to access the robustness of the different SA methods with regards to their primary parameter, namely the sample size. Especially when working with expensive black box optimization or learning feature interactions on expensive real-world problems (expensive in terms of computational power or license costs etc.), a small sample size is preferred. However, not all methods work equally well with a small number of samples, of course also depending on the characteristics of the function and the dimensionality of the problem. Due to different required sampling techniques for each method it is impossible to use *exactly* the same sample sizes for comparison. We therefore work with the following *base* sample sizes (B_n) (from 2^7 to 2^{15} samples) and the number of dimensions 2, 4, 8, 16, 32 and 64. Morris, Sobol (without second order indices) and DGSM requires $x \cdot (d+1)$ samples (where d is the number of dimensions and x a user defined number), and FAST requires $x \cdot d$ samples. All other methods can be set to a Latin Hypercube sampling with any number of samples n. In order to make a fair comparison between the methods the number of samples should differ as little as possible. We therefore use the following real sample sizes N for each method using the base sample sizes ($b \in B_n$) mentioned before.

- Morris: $N = b/d \cdot (d+1)$
- Sobol: $N = b/d \cdot (d+1)$
- DGSM: $N = b/d \cdot (d+1)$
- Fast: $N = b/d \cdot d$

Where d is the number of dimensions. All other methods use exactly b samples. This means that only Morris, Sobol and DGSM have d more samples, but since the number of samples is much larger than the dimensionality in this experiment the effect of this can be neglected.

In the following experiment we run seven SA methods (DGSM, Delta, FAST, Morrris, RDB-FAST, Sobol and PAWN), the simple statistical method Pearson correlations and two model-based methods (linear regression and random forest). We test these methods over a wide range of sample sizes (from 2^7 to 2^{15} samples) for the 24 different noiseless BBOB functions [11]. Each method that requires a specific sampling scheme uses this specified scheme, while other methods such as Pearson, RDB-FAST, random forest and linear regression use Latin Hypercube sampling. We repeat these experiments 10 times with different random seeds and

FIGURE 10. Sensitivity analysis on f_3 in 2, 4 and 8 dimensions. Each plot shows the results using a different algorithm. The sample size is on the x-axis and the scaled sensitivity on the y-axis. The shaded area denotes the standard deviation from the 10 runs with different random seed.

report back on the average and standard deviation of the results. To compare the different methods we scale the sensitivity indices for methods that do not give a sensitivity between 0 and 1 by dividing the sensitivities by the sum of all variable sensitivities. For the SA methods we only look at the $S1$ scores, the first order sensitivity indices. The experiment is repeated for different dimensions to investigate whether the different dimensionality requires more samples and whether certain methods perform better in high dimensions than others.

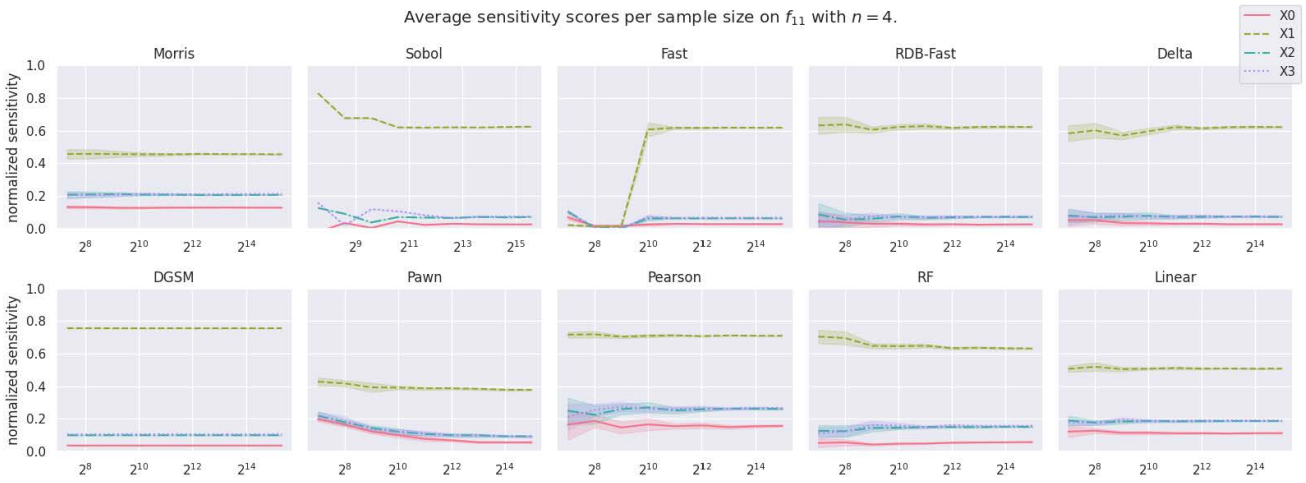

FIGURE 11. Sensitivity analysis on f_{11} ($n = 4$), showing the mean and standard deviation over 10 runs for all different methods and sample sizes.

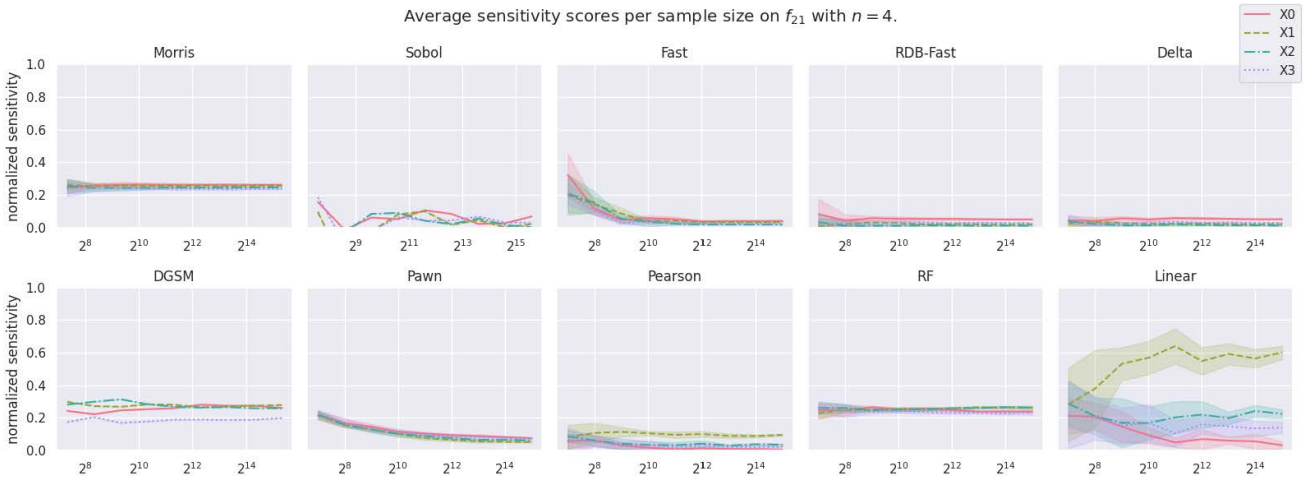

FIGURE 12. Sensitivity analysis on f_{21} ($n = 4$), showing the mean and standard deviation over 10 runs for all different methods and sample sizes.

A. ROBUSTNESS: RESULTS

Figure 10 shows the mean sensitivity of each parameter for different sample sizes and methods. From these plots, it is immediately apparent that some algorithms do not perform well for a relatively small sample size (below $2^{10} = 1024$ samples). The robustness of the results under different random seeds is also interesting to notice for the model-based approaches, which have much larger variance than, for example Sobol.

To compare the methods in terms of robustness (rather than validity, since we do not know the ground truth sensitivities of all BBOB functions), we compute the mean squared error between the sensitivities of each sample size (for each function) and the largest sample size. In doing so, we assume that the method with the largest sample size should be the most stable (and accurate). In this way, we can summarize the stability of the algorithms across all functions and sample sizes. Table 1 shows the results of this experiment. In bold is the algorithm per number of dimensions that performs most robust. Note that this experiment does not identify the ''best performing'' SA method, as an algorithm that would always output a constant value would be very robust but not correct.

B. ROBUSTNESS TO DIFFERENT SAMPLE SIZES

From the results in Table 1 it is easy to see that Morris is the most stable method over all different sample sizes and benchmark functions. The model-based methods are also surprisingly robust, as well as DGSM. However, it is important to note that this does not mean that the Morris method is better overall. Most methods show significant convergence when larger sample sizes are used. It is also clear that for most methods the number of samples needed scales exponentially with the dimensionality of the problem, Morris and Linear

TABLE 1. Mean square error over all functions and sample sizes per method, algorithm and dimensionality. The MSE is calculated by using the largest sample size as the ground truth. Bold values indicate the most stable method in the given dimensionality.

Method	2d	4d	8d	16d	32d	64d
Morris	$\mathbf{0.151E^{-3}}$	$\mathbf{0.272E^{-3}}$	$\mathbf{0.275E^{-3}}$	$\mathbf{0.186E^{-3}}$	$\mathbf{0.095E^{-3}}$	$\mathbf{0.040E^{-3}}$
Sobol	$1.317E^{-3}$	$2.266E^{-3}$	$2.464E^{-3}$	$6.215E^{-3}$	$7.403E^{-3}$	$37.974E^{-3}$
Fast	$55.629E^{-3}$	$26.276E^{-3}$	$77.174E^{-3}$	$74.949E^{-3}$	$84.465E^{-3}$	$82.204E^{-3}$
RDB-Fast	$0.758E^{-3}$	$0.814E^{-3}$	$0.719E^{-3}$	$0.618E^{-3}$	$0.544E^{-3}$	$0.513E^{-3}$
Delta	$4.781E^{-3}$	$1.681E^{-3}$	$0.738E^{-3}$	$0.333E^{-3}$	$0.246E^{-3}$	$0.213E^{-3}$
DGSM	$2.921E^{-3}$	$4.169E^{-3}$	$0.932E^{-3}$	$0.549E^{-3}$	$0.338E^{-3}$	$0.177E^{-3}$
Pawn	$2.973E^{-3}$	$4.239E^{-3}$	$5.515E^{-3}$	$6.242E^{-3}$	$6.958E^{-3}$	$7.582E^{-3}$
Pearson	$1.030E^{-3}$	$1.277E^{-3}$	$1.414E^{-3}$	$1.464E^{-3}$	$1.504E^{-3}$	$1.513E^{-3}$
RF	$1.984E^{-3}$	$1.152E^{-3}$	$0.835E^{-3}$	$0.477E^{-3}$	$0.292E^{-3}$	$0.114E^{-3}$
Linear	$6.419E^{-3}$	$4.783E^{-3}$	$1.917E^{-3}$	$0.557E^{-3}$	$0.189E^{-3}$	$0.071E^{-3}$

being exceptions. When we look at a highly multi-modal function f_{11} in Figure 11 and function f_{21} in Figure 12, we see that more variance occurs in the results and that for very complex functions the sensitivities collapse and little differences per variable are observed.

VIII. COMPARISON OF ACCURACY

The second experiment uses a linear function generator to benchmark the above algorithms. In this experiment, we know the ground truth sensitivity indices because we know the coefficients of the linear function. We sample the coefficients of the ''effective dimensions'', the influential parameters, uniformly randomly between 0 and 100 and take samples between 0 and 1. Moreover, we define problems with total dimension higher than the number of important variables to simulate variables that have no influence. Since the ground truth is known, we use the Kendall's tau metric for the highest ranked features by taking the top ''effective dimensions'' to verify that the methods can correctly predict which variables are important and in the right order. Kendall's tau is a measure of the correspondence between two rankings. Values close to 1 indicate strong agreement, values close to -1 indicate strong disagreement. We used the tau-b version of Kendall's tau, which accounts for ties.

Note that in this experiment we use only linear functions, so the Linear Regression method has a clear advantage. The goal of this experiment is twofold: first, we want to check which algorithms perform best in low and extremely high dimensions with relatively small sample sizes. Second, we want to see the performance when the number of effective dimensions is relatively small but the actual number of parameters is large. We also track the computation time for each algorithm (only the sensitivity analysis part, not sampling) to get a good overview of the average runtime per algorithm for different sample sizes and number of dimensions.

We consider the following tuples of effective dimensions and total dimensions respectively: $(2, 2)$, $(6, 8)$, $(8, 16)$, $(16, 32)$, $(32, 128)$, $(64, 128)$, $(64, 1024)$, $(256, 1024)$, $(128, 8192)$, $(4048, 8192)$. The sample sizes considered are $(128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768)$. Each experiment is repeated 10 times with different random seeds. The values given are the averages of these runs. We did not include TreeSHAP because it is model dependent and computationally more expensive.

A. ACCURACY: RESULTS

From Figure 13 and Figure 14, it can be seen that most methods require many more samples as the number of total dimensions increases. In Figure 14, one can see that FAST in particular requires a minimum number of samples, which depends on the number of features to be analyzed (the gray areas did not have enough samples to execute the algorithm). Two clear winners emerge from the tau statistics, namely the linear model and the Morris method, both of which perform very well even when the number of dimensions is high and the sample size is low. Note that the linear model has an advantage in this setup since only linear functions are considered. In real-world applications, this is unlikely to be the case and therefore the linear model would be less useful. Sobol and FAST perform well when the sample size is sufficiently large. In terms of efficiency, the Delta and RF methods require the most computational effort and Pearson the least (although it is less effective and also relies heavily on the linearity assumption).

IX. A QUALITATIVE COMPARISON OF SA METHODS

In this paper, we present methods that are global and model-free, i.e., they are independent of assumptions about the model, such as linearity, additivity, etc. Although the main goal is to provide model-based analytical tools to study how uncertainties in model output are related to uncertainties in input, quantitative methods are useful whenever the goal is to understand the extent to which a particular factor is more important than another. Variance-based measures are an example of quantitative methods. The choice between quantitative and qualitative methods usually depends on the costs and characteristics of the case study under investigation. Relatedly, there are several desirable characteristics for GSA methods:

- Computation of first, second, and total order sensitivities. Based on the study purposes, a GSA method should be able to compute different sensitivity indexes: the

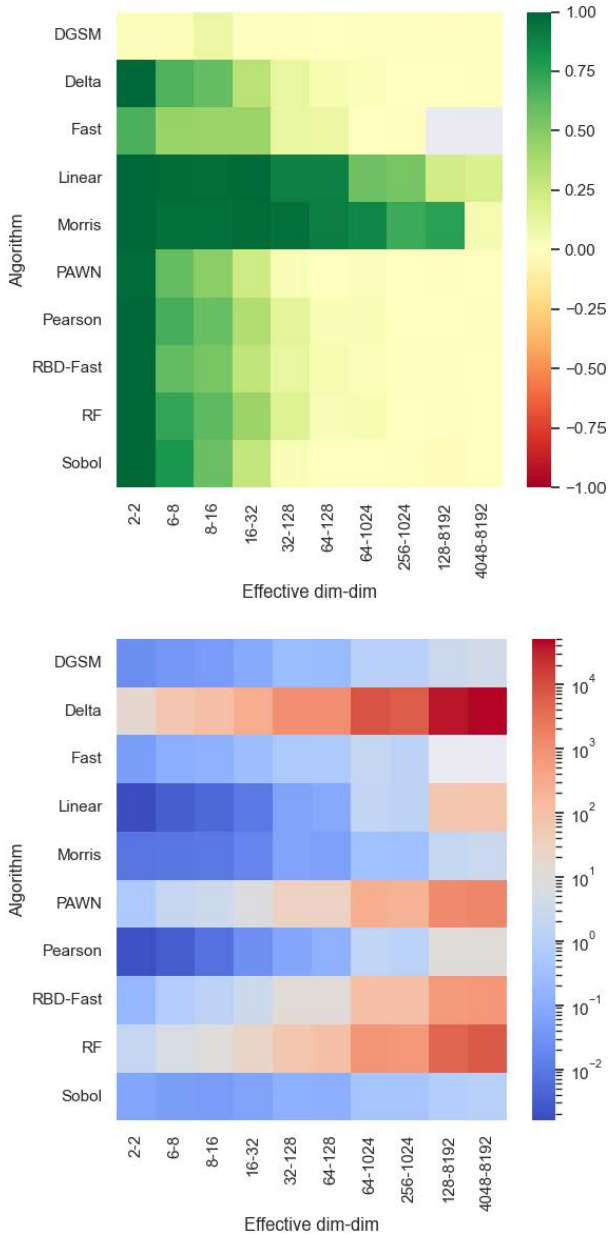

FIGURE 13. On top a visualisation of the Kendall's tau (averaged over all sample sizes and random seeds) per algorithm (y-axis) against the number of effective dimensions and real dimensions (x-axis). A tau value of 1 represents total agreement with the ground truth ranking and −1 a total disagreement. Below the logarithmic execution time for the same combinations.

first-order index, also called main effect index, which measures the effect on the output variance of varying a factor alone, the second-order index, which measures the fractional contribution of factor interactions to the output variance, and the total sensitivity index, which measures the influence of a variable jointly with all its interactions;

- Estimation of the direction of the effect. The method is able to forecast not only the importance of a factor, but also provides the direction of the change that a factor has on the target variable;
- Providing a confidence indication. A GSA method can also provide information about the variation in the estimate of the measure predicted to state the importance of a factor;
- Being able to treat grouped factors. When the dimensionality of the problem is high and some insight about similarities between factors is available, this property is essential to perform GSA while keeping the number of model runs low;
- Being model independent. The method should be able to work on non-linear, non-additive models, and to estimate interaction effects between different factors;
- Being independent of the sampling scheme. Regardless of the sampling on which the GSA is performed, it should be able to produce consistent results;
- Including multidimensional averaging. A global SA method should be able to evaluate the effect of a factor while the others are varying from their nominal value.

In addition to this set of characteristics that a GSA method can support, the results of the first and second quantitative experiments are included in Table 2 to provide a complete comparative overview of the different GSA methods. This overview allows a researcher to select the appropriate methods under different circumstances and requirements. For example, if we have a fixed design of experiments that is generated by an unknown sampling process with 10 dimensions, the DELTA or PAWN methods seem to be a good choice. However, if we have an expensive simulator, no design of experiments yet, and a large number of dimensions, then the Morris method would probably be a better choice.

X. OPEN SOURCE GSA REPORTING SOFTWARE

In order to make Global Sensitivity Analysis methods and to combine the information that can be extracted with this large variety of methods and automatically select the right methods, we propose the *GSAreport* application. With *GSAreport* one can create a detailed and interactive HTML report for a specific dataset, machine learning model or real-world process. The application selects the best GSA methods to use based on the number of dimensions and samples per dimension. If the number of dimensions is over 64, Sobol and PAWN are omitted. If the number of samples per dimension is less than 50, Sobol, Delta, and PAWN are not applied. These rules follow the observations from Table 2. FAST and DGSM are not included in the application due to their relatively low performance. The *GSAreport* application can be easily installed by either downloading the executable, using the provided Docker image, or setting up the Python dependencies. No programming skills are required to use the package. The software generally works with two different steps, the sampling step and the analysis and report generation step. In the sampling step, the software can generate different designs of experiments required for the application of all

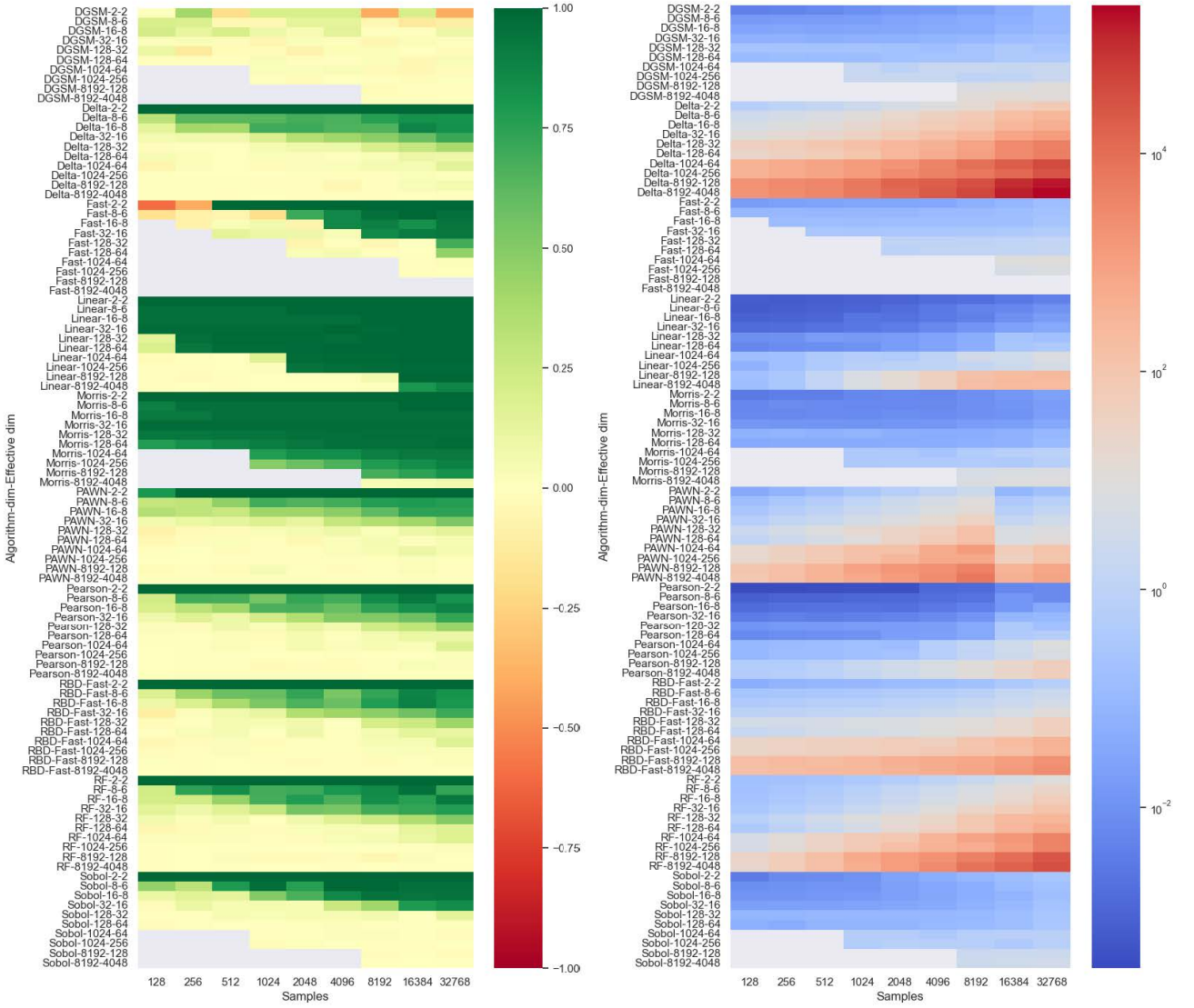

FIGURE 14. On the left a visualisation of Kendall's tau per combination of number of effective dimensions, total number of dimensions and algorithm (*y*-axis) against the sample size (averaged over all random seeds). On the right the logarithmic execution time for the same combinations. Grey areas mean that the number of samples was insufficient to carry out the algorithm at all.

applicable global SA methods. These designs of experiments are stored as csv files and can be used as input for real-world or model evaluations. The evaluations should be stored in the same order and format. Alternatively, an existing data set with input and output samples can be provided. In this case, a surrogate model (Random Forest) is built to allow for methods that require a specific sampling scheme. In the second step, the data is loaded and the various global SA methods are executed. The resulting sensitivity indices are displayed interactively in a concise reporting template with references and tips for each method. The software is freely available on GitHub.[1] The GitHub repository contains a *readme* and a link to the documentation page explaining all the details needed for installation and usage of the reporting tool (e.g., format of raw data, parameter setting, problem definition).

XI. REAL-WORLD USE CASE: IDENTIFYING IMPORTANT SNPs IN DNA DATA

The Morris method is here applied to provide an example of GSA on a specific real-world scenario dealing with genomic prediction, i.e., prediction of phenotype (output) based on genomic data (input). The target of the application is to understand which parts of the genome, i.e., Single Nucleotide Polymorphisms (SNPs), have a major role in the predicted outcome. Among the other GSA techniques, the Morris method was chosen for this particular application because it can handle the very large number of discrete variables characterizing this application study and includes

[1] https://github.com/Basvanstein/GSAreport

TABLE 2. Comparison of different GSA and XAI methods based on different aspects and support. A + indicates that the feature is present for a method. For the performance in accuracy and time a distinction is made between the best methods (++), good methods (+), worse methods (−) and the worst method for a given situation (- -). *The TreeSHAP method is model dependent, however classical Shapley values are model independent. ** The linear model has an advantage due to the experimental setup and is therefore not included in this overview regarding accuracy performance. * The TreeSHAP method was not included.**

	Variance-based			Derivative-based		Density-based		Model-based		
	Sobol	Fast	RBD-FAST	Morris	DGSM	DELTA	PAWN	Linear	RF	TreeSHAP
First order sens.	+	+	+	+	+	+		+		
Second order sens.	+			+						
Total order sens.	+	+		+	+	+	+		+	+
Direction of effect								+		+
Confidence indication	+	+	+	+	+	+	+			
Grouping support	+			+	+					
Model independence	+	+	+	+	+	+	+			*
Sampling scheme independence			+			+	+	+	+	+
No min. sample size required							+	+	+	+
Multidimensional averaging	+	+	+	+	+	+	+	+	+	+
Performance in low dim.	+	-	+	++	- -	+	+	**	+	***
Performance in high dim.	-	- -	+	++	-	+	-	**	+	***
Performance with small sample sizes.	-	- -	+	++	-	-	-	**	-	***
Computational efficiency.	+	-	+	+	+	- -	-	++	-	***

multi-dimensional averaging, i.e., is able to account for interaction effects when ranking factors. It also allows for working with groups of factors, which is advantageous when dealing with large data sets.

A. BACKGROUND

Plant breeding is the science-driven creative process that uses principles from a variety of sciences to develop new plant varieties and improve the genetic potential of plants. The process involves combining parental plants to obtain the next generation with the best characteristics. Selection of genetically superior genotypes among a huge amount of recombinant and segregating progenies is an essential but complex procedure in plant breeding [28]. Advances in computer modeling and simulation have provided many advantages compared to conventional plant breeding and have been essential for breeders to make critical decisions in the design of their breeding programs [29]. However, to determine the genetic potential of individuals or families in the field and choose the best genotypes that exhibit desirable traits, cyclic crossing and selection procedures are required, which involve extensive field experiments and computational resources. As a consequence of the limit in available resources, plant breeders are often forced to reduce the number of plants that can be fully grown. It is in this context that GSA can play a relevant role, trying to select the plants that contain SNPs in the state that contributes most to the phenotype.

In studies focused on single nucleotide polymorphisms (SNPs), which are not limited to the field of plant breeding, scientists use genotyping analysis to try to untangle the complex relationships between genotype and phenotype [30]. When genetic variation occurs, meaning that the value of one or more SNPs in a sequence varies, SNP contributions can be independent or influence/modulate the effect of each other.

B. METHODOLOGY

The GSA methodology based on the Morris method comprises several building blocks, which are illustrated in Figure 15. Using a breeding simulation platform, several generations of random mating are simulated to create a population of 1000 plants. In this simulation the genotype of a plant is represented using 27205 SNPs, split over 5 chromosomes. The exact location of a SNP on a chromosome is sampled uniformly at random. To calculate the trait value (phenotype) of a plant we assume an epistatic trait model based on 20 SNPs, listed in Table 4. In an epistatic trait model the contribution of a SNP to the trait is based on both the states of that SNP as well as the state of (some) other SNPs through interactions. This population is split randomly, 80% for training and 20% for testing. A Convolutional Neural Network model (CNN) is trained on this data set. In accord with the literature [31] and in order to be able to draw sound statistics on the results, the training is repeated 10 times by initializing with different random seeds. Once the models are trained, a Morris statistical analysis is performed for each of them, by following the steps described in Section IV-A: a sample matrix is generated by considering 10 initial random designs and varying one SNP at a time on a k-dimensional 3-level (0,1,2) grid, where $k = 27205$ is the number of SNPs, and, as a consequence, the number of factors of our sensitivity analysis. Once the Morris sample matrix is generated, the phenotype associated to each genotype, meaning the trait value associated to each sequence of SNPs, is evaluated on the CNN prediction models. Finally, the GSA is performed, by considering both the standard and the grouping version of the Morris method.

C. ANALYSIS SETUP

We make use of the Python packages of tensorflow 2.6.0 and keras 2.6.0 to train the CNN model. Here, we use conic convolutions, meaning that the number of filters get

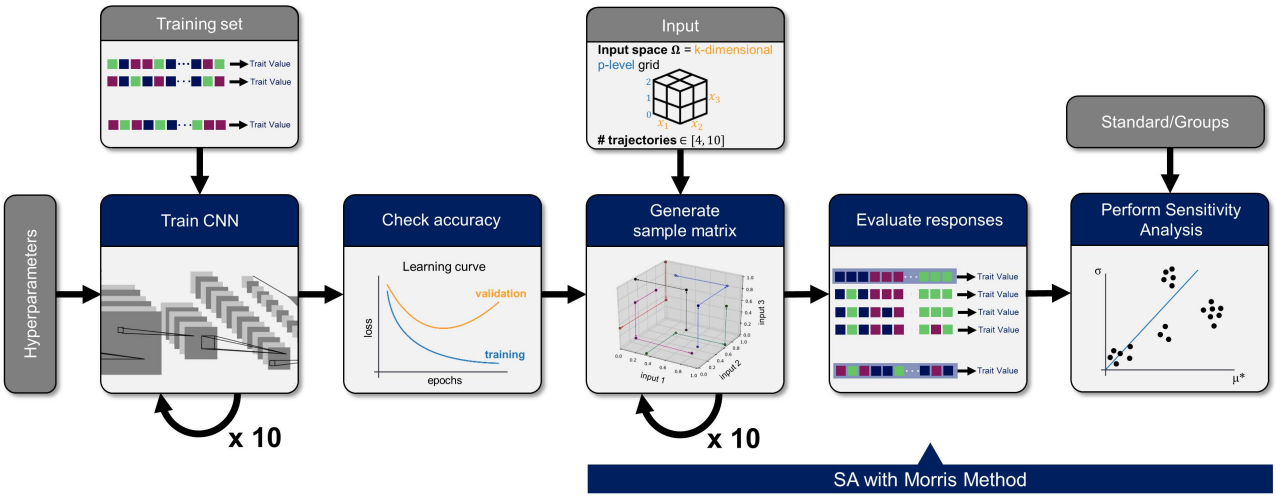

FIGURE 15. Flowchart of the sensitivity methodology for finding contributing SNPs in genomic prediction models using Morris GSA strategy on machine learning model (CNN) predictions.

TABLE 3. Overview of the hyperparameters used to train the machine learning (CNN) model used for the Morris GSA. The CNN is initialized with a maximum number of 1000 epochs, patience 20, and batchsize 64.

CNN hyperparameter overview		
UnitNumber	The number of nodes in the first dense layer.	64
FilterNumber	The number of filters in the first convolutional layer.	64
KernelSize	The size of the filters in the convolutional layers.	9
PoolStride	The stride between regions that are pooled together in the max pooling layers, after each convolutional layer.	5
DropRate	The ratio of nodes that are randomly set to zero during training. A form of regularization.	0.4
ConvolutionShape	The relative sizes of consecutive convolutional layers. "Conic", consecutive layers have double the number of filters.	conic
LearningRate	The initial step size of weight updates during training.	0.0001
ConvolutionNumber	The number of consecutive convolutional layers.	2
EmbeddingDimension	The number of nodes of the embedding layer that represents the input data.	10
DenseNumber	The number of dense layers.	1
LambdaRegularization	The strength of L2 weight regularization during training. The value is scaled by the batch size.	2.5

doubled for every consecutive layer, for example 32-64-128. For embedding we just use a (learnable) lookup matrix to turn the SNP values (0,1,2) into vectors of size *EmbeddingDimension* $= 10$. We do not make use of any special embedding technique designed for NLP/words. An overview of the hyperparameters that have been used in this work are presented in Table 3. Once the CNN models are trained, these are used to evaluate the samples composing the trajectories generated by the Morris GSA method. We apply the Morris method in different configurations:

- Standard Morris: Sensitivity indices are calculated for each SNP (feature), as shown in the Equations (5) and (6);
- Morris in groups: The features are divided into groups of 200 SNPs, giving a total of 136 groups. The last group contains five additional SNPs to cover the whole set of features, as we have a total of 27205 SNPs. The sensitivity indices are calculated for each group for which the elementary effect is defined as in Eq. (7);
- Morris in groups based on correlations: The indices are calculated for groups of SNPs that do not have the same cardinality. They are indeed consecutive SNPs, but the groups are distinguished based on drops in pairwise correlation between consecutive SNPs. Different correlation biases are taken into account.

The extremely high dimensionality of the problem would benefit from a larger sample matrix. However, this would dramatically increase the computational costs, given that $r(k+1)$ samples, and hence model evaluations, are needed to compute the sensitivity indices. Therefore, in all cases, we compute a sampling matrix consisting of $r = 10$ trajectories, in agreement with other examples from the literature.

D. RESULTS

In our first experiment using the standard version of the Morris method, the values of the sensitivity measures are obtained by performing 272060 model evaluations ($k = 27205$ and $r = 10$), where each factor can vary among three levels (0, 1, 2). The ground truth of the simulated data evidences 20 SNPs (features) contributing to the final value, given in the Table 4. The ground truth contains a dense range of features between 25500 and 26800. Let us take a look at Figure 16, which provides an example of (μ^*, σ) distribution for the 27205 SNPs under analysis. Although we obtained different points distributions over the (μ^*, σ) plane for different CNN training random seeds, all of them present on the top right corner SNPs belonging to the same range. Moreover, if features belonging

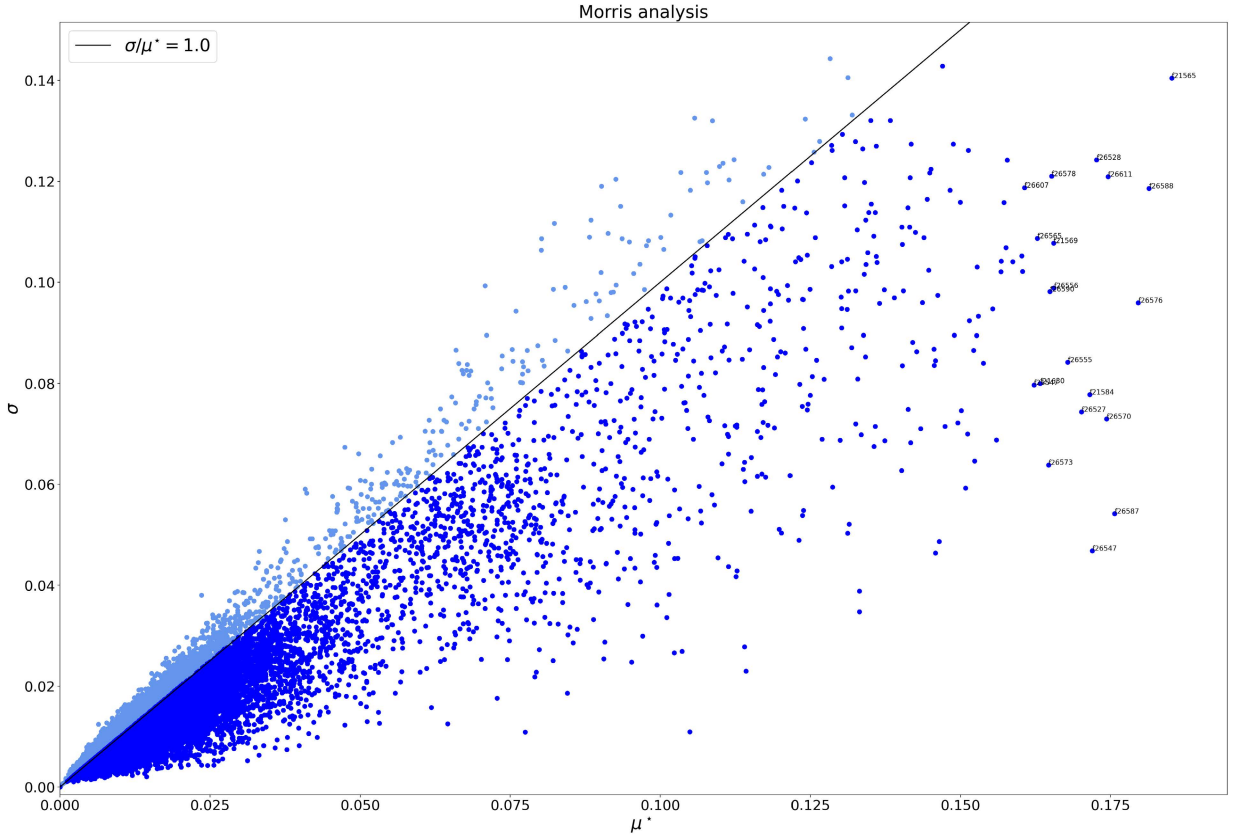

FIGURE 16. Morris σ versus μ^* plane, showing the sensitivity indices values for each of the 27205 SNPs. The black line shows the reference line which presents 1:1 relationship between σ and μ^*. SNPs on the left side (in light blue) are non-influential or influential with nonlinear or interacting effects, while SNPs on the right side (in plain blue) are more influential with linear and/or additive effects. Strongly influent variables are labeled.

TABLE 4. Ground truth of simulated data. 20 features are assumed to have a large effect on the variance of the target value. Their indices are listed.

Indices			
408	7233	20639	25522
2501	10967	21595	25814
3813	13324	22115	26534
5545	14388	24665	26552
6654	16895	25519	26781

to other ranges are denoted as influential ones, such features (or some highly correlated close neighbors) always belong to the ground truth list. Although the graphical representation on the (μ^*, σ) plane allows for considering together the values of both the sensitivity indices and hence appreciate the relative influence of the inputs, it becomes rather impracticable in the case of many (hundreds or thousands) factors. Moreover, it is not possible to show averaged results in our case, where we perform GSA on 10 different CNN model predictions. Hence, to draw sound conclusions, we present the two line-plots in Figure 17, which depict the trend of the μ^* sensitivity index calculated according to the standard Morris method and Morris with groups, respectively. The index μ^* allows for the detection of the most important factors without necessarily inspecting the σ index. Using the Morris method with groups, a total of $k = 138$ groups are defined and $r = 10$ trajectories are generated, resulting in a total of 1390 model evaluations. Thus, grouping factors into subgroups allows us to run a much cheaper GSA while providing results that are easily readable and consistent with the full SA for all factors. From the plots, it can be observed that all the major peaks detect some SNPs belonging to the ground truth, but not the other way around, meaning that some ground truth features are located in ranges with low μ^* according to Morris analysis. However, the highest peaks are located in correspondence to the neighbours with the highest density of features belonging to the ground truth, highlighting space for further investigation in localized portions of the factor domain. In addition, the most important features/groups seem to influence the importance of the neighbouring features/group, hence suggesting a correlated behaviour between them. This is because neighboring SNPs on the same chromosome are very likely to be inherited together.

To take advantage of the biological reality for SNP interaction, a different logic for grouping factors based on neighbouring is used. We base the splitting of factors into different groups based on pairwise correlation between consequent SNPs, computed on the training data. In particular, we evaluate correlations based on Pearson product-moment

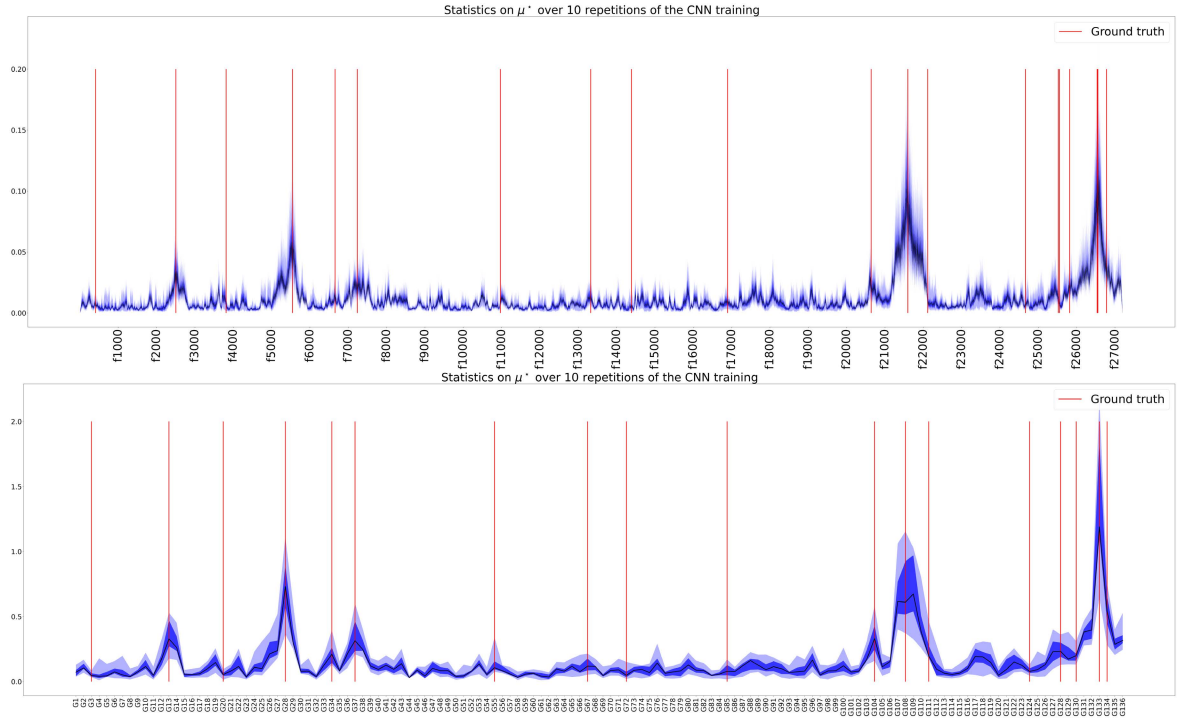

FIGURE 17. μ^* lineplot according to Morris sensitivity analysis standard (top) and for groups of 200 input variables (bottom). The absolute mean against feature/group number is plotted. The black line is the median over 10 random seeds for the CNN training. The dark blue area corresponds to lower-upper quartile range and the light blue one to the whole min-max range of the computed values for μ^*. Red vertical lines correspond to the features belonging to the ground truth in the first subplot, and to groups containing the relevant features according to the ground truth in the second one.

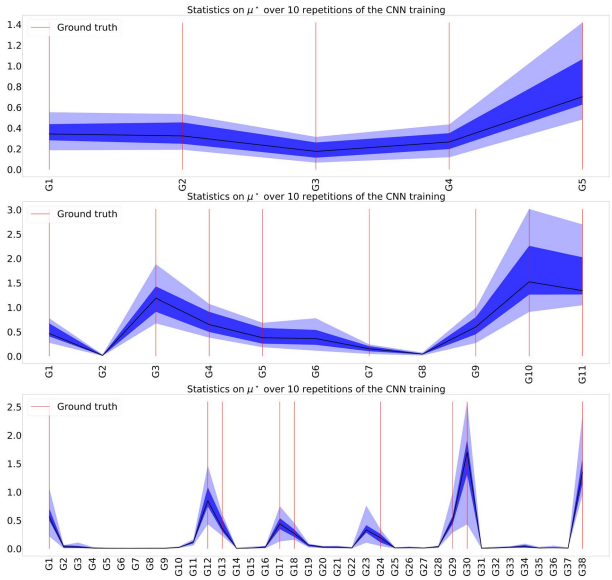

FIGURE 18. μ^* line plot from the Morris sensitivity analysis with groups defined according to drops in pairwise correlation between SNPs. The absolute mean against group number is plotted. The black line is the median over 10 random seeds for the CNN training. The dark and light blue areas correspond to lower-upper quartile range and to the min-max range of the computed values for μ^*, respectively. Red vertical lines indicate the groups containing the ground truth relevant features. Top: $R = 0.90$ case. Center: $R = 0.98$ case. Bottom: $R = 0.99$ case.

correlation coefficients $R_{ij} = \text{Cov}(X_i, X_j)/(\sigma_i \sigma_j)$, where Cov is the covariance function and σ_i is the standard deviation of a random variable X_i. Given three thresholds for this coefficient, namely $R = 0.90, 0.98, 0.99$, we split the factors sequence to different groups when the pairwise correlation of one factor with one of its two neighbours falls below this threshold. Hence, we obtain 5, 11, and 38 different groups for the three thresholds, respectively. Figure 18 shows the averaged lineplots representing the values calculated for the μ^* sensitivity measure for all feature groups.

It is evident that by choosing $R = 0.90$ we have SNPs from the ground truth following into each group. In this case, the 5 groups are most likely exactly the 5 chromosomes we used for simulating the data. According to our SA, high values of μ^* correspond to high number of relevant SNPs belonging to a group. In fact, the fifth group, which is the one with the highest μ^* value, is composed of the SNPs ranging from index 20886 and 27205, hence it also contains the highest number of features listed in Table 4. By choosing $R = 0.98$ some groups do not contain SNPs from the ground truth (G2 and G8). These groups are irrelevant according to the Morris SA. On the other hand, G6, ranging from feature 14564 to 16757 does not contain any ground truth SNP either, but has some relevance according to the results of the SA. In any case, according to the data from the ground truth, G11 = [23665, 27205] is the more dense group, and it has a sufficiently high value for μ^* to attract our attention and be worth further and more focused SA. By choosing $R = 0.99$ we have even higher agreement between experimental results and ground truth. In fact, all the groups that

are considered relevant according to our SA also contain some SNPs in Table 4. However, it has to be stressed that the Morris method defining groups based on pairwise correlation between SNPs creates factors groups of different cardinalities and hence tends to favour bigger groups. Except for the first subplot in Figure 18 where all groups have similar cardinality, bigger groups are assigned bigger values of μ^*, which makes sense as we vary more factors at a time in the definition of the Morris elementary effect, and hence we have a higher probability of impacting the variance of the output. However, larger groups also imply a higher probability of containing relevant features, which would support the results of the sensitivity analysis. In any case, the results confirm that the modeler can easily select a subset of factors that play a minor role in the model, i.e., that are responsible for only a small percentage of the total output variance, thus preparing the ground for model simplification. For example, the sensitivity analysis shows that there are groups of factors that have almost no influence despite their size.

XII. SUMMARY AND OUTLOOK

A thorough comparison of Global Sensitivity Analysis (GSA) methods and similar working XAI methods that can be used for the same purpose of identifying important variables is presented. Two quantitative experiments are conducted to determine the robustness and accuracy of the different methods under different circumstances. A qualitative evaluation of the different methods is made, distinguishing which methods should be used and when. In addition to the overview and comparison, an open-source software package is proposed that allows experts and non-experts to easily work with a wide range of GSA methods to gain a better understanding of their machine learning models, simulators, and real-world processes. From our experiments, we can conclude that:

- Morris is one of the most robust GSA methods that performs well even with a high number of dimensions and a small sample size.
- Density-based methods such as DELTA perform well when the sample size is sufficiently large, and have the advantage of not depending on a sampling scheme or model.

Given the promising performance of the Morris method based on our comparisons, we select it to present a GSA use case study for genomic prediction. The high dimensionality, discrete nature of the factors, and small amount of training data make it a challenging problem to address with GSA. Morris' sensitivity analysis is performed for all factors, for groups of factors of equal cardinality, and for unequal groups defined by splitting the factors according to their correlation. Although further developments and more in-depth analyses are possible (e.g., optimized techniques for generating trajectories, nested sensitivity analyses in promising groups, dynamic sensitivity analyses by reducing group ranges), our preliminary study shows good agreement between model results and actual ground truth data.

For future research, new methods could be incorporated into the experiments and a methodology to test accuracy for more complex functions could be explored. This work was limited to the comparison of global SA methods, while there are also many local SA methods that would require different experimental setups for comparison.

REFERENCES

[1] A. Saltelli, M. Ratto, S. Tarantola, and F. Campolongo, "Sensitivity analysis for chemical models," *Chem. Rev.*, vol. 105, no. 7, pp. 2811–2828, May 2005.
[2] Z. Zi, "Sensitivity analysis approaches applied to systems biology models," *IET Syst. Biol.*, vol. 5, no. 6, pp. 336–346, 2011.
[3] J. Chen, Q. Shi, and W. Zhang, "Structural path and sensitivity analysis of the CO_2 emissions in the construction industry," *Environ. Impact Assessment Rev.*, vol. 92, Jan. 2022, Art. no. 106679.
[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Francisco, CA, USA, Aug. 2016, pp. 1135–1144.
[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf
[6] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*.
[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
[8] B. Iooss and P. Lemaître, "A review on global sensitivity analysis methods," in *Uncertainty Management in Simulation-Optimization of Complex Systems*. Boston, MA, USA: Springer, 2015, pp. 101–122.
[9] K. Cheng, Z. Lu, C. Ling, and S. Zhou, "Surrogate-assisted global sensitivity analysis: An overview," *Struct. Multidisciplinary Optim.*, vol. 61, no. 3, pp. 1187–1213, 2020.
[10] W. Becker, "Metafunctions for benchmarking in sensitivity analysis," *Rel. Eng. Syst. Saf.*, vol. 204, Dec. 2020, Art. no. 107189. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832020306906
[11] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tusar, and D. Brockhoff, "COCO: A platform for comparing continuous optimizers in a black-box setting," *Optim. Methods Softw.*, vol. 36, pp. 114–144, Jan. 2021.
[12] S. Andrea, R. Marco, A. Terry, C. Francesca, C. Jessica, G. Debora, S. Michaela, and T. Stefano, *Global Sensitivity Analysis: The Primer*, 1st ed. Hoboken, NJ, USA: Wiley, 2008.
[13] M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Math. Comput. Simul.*, vol. 55, nos. 1–3, pp. 271–280, 2001.
[14] R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek, and J. H. Schaibly, "Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I theory," *J. Chem. Phys.*, vol. 59, no. 8, pp. 3873–3878, 1973.
[15] A. Saltelli, S. Tarantola, and K. P.-S. Chan, "A quantitative model-independent method for global sensitivity analysis of model output," *Technometrics*, vol. 41, no. 1, pp. 39–56, 1999.
[16] C. Xu and G. Gertner, "Understanding and comparisons of different sampling approaches for the Fourier amplitudes sensitivity test (FAST)," *Comput. Statist. Data Anal.*, vol. 55, no. 1, pp. 184–198, Jan. 2011.
[17] S. Tarantola, D. Gatelli, and T. A. Mara, "Random balance designs for the estimation of first order global sensitivity indices," *Rel. Eng. Syst. Saf.*, vol. 91, no. 6, pp. 717–727, 2006.
[18] M. D. Morris, "Factorial sampling plans for preliminary computational experiments," *Technometrics*, vol. 33, no. 2, pp. 161–174, 1991.
[19] F. Campolongo, J. Cariboni, and A. Saltelli, "An effective screening design for sensitivity analysis of large models," *Environ. Modell. Softw.*, vol. 22, no. 10, pp. 1509–1518, 2007.
[20] I. Sobol and S. Kucherenko, "Derivative based global sensitivity measures," *Proc.-Social Behav. Sci.*, vol. 2, no. 6, pp. 7745–7746, 2010.

[21] S. Kucherenko and B. Iooss, "Derivative based global sensitivity measures," in *Handbook Uncertainty Quantification*, D. H. R. Ghanem and H. Owhadi, Eds. Berlin, Germany: Springer, 2017.
[22] E. Plischke, E. Borgonovo, and C. L. Smith, "Global sensitivity measures from given data," *Eur. J. Oper. Res.*, vol. 226, no. 3, pp. 536–550, May 2013. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:ejores:v:226:y:2013:i:3:p:536-550
[23] F. Pianosi and T. Wagener, "A simple and efficient method for global sensitivity analysis based on cumulative distribution functions," *Environ. Model. Softw.*, vol. 67, pp. 1–11, May 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364815215000237
[24] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinform.*, vol. 9, no. 307, pp. 1–11, 2008.
[25] A. Antoniadis, S. Lambert-Lacroix, and J.-M. Poggi, "Random forests for global sensitivity analysis: A selective review," *Rel. Eng. Syst. Saf.*, vol. 206, Feb. 2021, Art. no. 107312.
[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[27] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable AI for trees," *Nature Mach. Intell.*, vol. 2, no. 1, pp. 2522–5839, 2020.
[28] M. Ali, L. Zhang, I. DeLacy, V. Arief, M. Dieters, W. H. Pfeiffer, J. Wang, and H. Li, "Modeling and simulation of recurrent phenotypic and genomic selections in plant breeding under the presence of epistasis," *Crop J.*, vol. 8, no. 5, pp. 866–877, Oct. 2020.
[29] J. Wang, H. Eagles, R. Trethowan, and M. Van Ginkel, "Using computer simulation of the selection process and known gene information to assist in parental selection in wheat quality breeding," *Austral. J. Agricult. Res.*, vol. 56, pp. 465–473, May 2005.
[30] B. L. Fridley and J. M. Biernacka, "Gene set analysis of SNP data: Benefits, challenges, and future directions," *Eur. J. Human Genet.*, vol. 19, no. 8, pp. 837–843, Aug. 2011.
[31] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Elementary Effects Method*. Hoboken, NJ, USA: Wiley, 2007, ch. 3, pp. 109–154. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470725184.ch3

BAS VAN STEIN (Member, IEEE) received the Ph.D. degree in computer science from the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands, in 2018.

From 2018 to 2021, he was a Postdoctoral Researcher with LIACS, Leiden University, where he is currently an Assistant Professor. His research interests include surrogate assisted optimization, surrogate assisted neural architecture search, and explainable AI techniques for industrial applications.

ELENA RAPONI received the Ph.D. degree in sciences and technology and mathematics from the University of Camerino, Italy, in May 2021.

She was a Postdoctoral Researcher with the Natural Computing Research Group, Leiden Institute of Advanced Computer Science (LIACS). She is currently a Postdoctoral Researcher with the Technical University of Munich (TUM) and the Chair of computational mechanics. Financed by a DAAD Prime Postdoctoral Fellowship, her postdoctoral position is joint between TUM and LIP6 Department, Sorbonne University, with hosts Fabian Duddeck and Carola Doerr, respectively. She has particular expertise in surrogate-based and high-dimensional (Bayesian) optimization in continuous domains. Her research interest includes development of analytical and numerical modeling techniques for the optimization of geometries and materials in structural mechanics.

ZAHRA SADEGHI received the Ph.D. degree in computer engineering majored in artificial intelligence and robotics from the Department of Electrical and Computer Engineering, University of Tehran. She has extensive experience of doing research and working in the field of artificial intelligence, machine learning, cognitive science, and computer vision in top universities and industry. She has published in peer-reviewed articles and book chapters. Her research interests include intersection of artificial intelligence and cognitive science. She was a recipient of prestigious awards such as the Marie Skłodowska-Curie Fellowship and the David Rumelhart Travel Award.

NIEK BOUMAN received the Ph.D. degree in applied mathematics from the Eindhoven University of Technology, The Netherlands, in 2013.

After obtaining his Ph.D., he joined as a Data Scientist with Agro-Biotech Company Keygene N.V. where he is also involved in the design and application of computational methods for accelerated crop improvement.

ROELAND C. H. J. VAN HAM received the Ph.D. degree in molecular evolutionary biology from Utrecht University, The Netherlands, in 1994.

Since 2011, he has been the Vice-President of bioinformatics and modeling with Agro-Biotech Company KeyGene N.V., where he leads the Research and Development Department in the development and application of computational methods for accelerated crop improvement. Since 2015, he combines his work at KeyGene with an appointment as a Professor in plant computational biology with Technical University Delft and as the Scientific Director of the TU Delft AgTech Institute. His research interest includes unraveling genotype-phenotype relationships in biological organisms.

THOMAS BÄCK (Fellow, IEEE) received the Diploma and Ph.D. degrees in computer science from the University of Dortmund, Germany, in 1990 and 1994, respectively.

Since 2002, he has been a Full Professor of computer science with the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. He is the author of the *Evolutionary Algorithms in Theory and Practice* (OUP, 1996) and the co-editor of the *Handbook of Evolutionary Computation* (CRC Press, 1997) and the *Handbook of Natural Computing* (Springer, 2012). His research interests include evolutionary computation, machine learning, and their real-world applications, especially in sustainable smart industry and health.

Prof. Bäck received awards and honors include membership in the Royal Netherlands Academy of Arts and Sciences (KNAW), in 2021; the IEEE Computational Intelligence Society Evolutionary Computation Pioneer Award, in 2015; the Fellow of the International Society of Genetic and Evolutionary Computation, in 2003; and the Best Ph.D. Thesis Award of the German Society of Computer Science (GI), in 1995.