Dear Dr. Easterbrook and Reviewers:

Thank you for your initial consideration of our manuscript (gmd-2019-17, "What do we do with model simulation crashes? Recommendations for global sensitivity analysis of earth and environmental systems models"). We appreciate the detailed suggestions for improvement and opportunity to submit a revised version. In response to the reviewers' comments, we have made major revisions to the manuscript and addressed all the comments as described in this rebuttal document. These are detailed in a point-by-point response to each comment below; reviewer comments are in *italicized*, *blue text* and our response is in normal, black text.

The reviewers' comments have substantially improved the manuscript in a number of ways. Major changes have been made to the revised manuscript, as listed below:

1) The title of paper has been modified in the revised manuscript as suggested by Reviewer 2.
2) In response to the comments raised by Reviewer 1, another sensitivity analysis method (i.e., a variance-based method) has been applied in order to demonstrate the utility of the proposed crash handling approach.
3) A new figure has been added into the results section to further compare the RBF and single NN techniques in terms of approximating the response surface.
4) New papers have been cited in the revised manuscript to demonstrate the motivation of our work and its merit over previous studies.

Other major/minor modifications have also been made to the manuscript to address the review comments, which have been listed below their associated review comments in the following. All these revisions have enhanced the contribution of our study to the literature by providing an efficient and effective approach to cope with failed simulations when performing global sensitivity analysis. We believe this is a valuable contribution to the Earth and Environmental Systems Modelling community and will be of great interest to Geoscientific Model Development readers.

Thank you for your consideration, and we hope to hear from you soon,

Razi Sheikholeslami et al.
razi.sheikholeslami@usask.ca

Response to Reviewers' Comments

This document contains copies of all the comments of Editor and Reviewers (in *italicized, blue text*) and our subsequent efforts to address them (in normal, black text).

Reviewer 1

With great interest I have read and reviewed the manuscript "What do we do with model simulation crashes? Recommendations for global sensitivity analysis of earth and environmental systems models" by Sheikholeslami et al. In general, the paper presents a novel and interesting approach to deal with the issue of model crashes when applying global sensitivity analysis. Although this new idea looks promising, additional investigations and explanations are necessary before this paper can be published. In the next sections general, major and minor comments and suggestions are provided that should allow the authors to improve their manuscript.

Response: We are very thankful to the reviewer for the time and effort spent on reviewing our paper. The comments and suggestions were constructive and helped us improve the quality of our manuscript.

General

To improve the validity of this novel idea and allow it to be applied in a more general context, more investigation is required by:

- Applying this for different SA techniques (e.g. a variance-based technique, as one of the proposed approaches might influence the variance of the output) (in particular on the HBV example with different ratios of the number of crashes).

Response: Thanks for this important comment. As suggested by the reviewer, we have tested the proposed crash handling approach using a variance based GSA technique and compared the results for the first case study in the revised manuscript (see **Figure 8** and **Appendix B**). However, note that since our proposed strategy is GSA-method-free and does not depend on the utilized GSA algorithm, the results have not changed, i.e., the RBF and single NN techniques still outperformed the median substitution in terms of closeness to the true GSA results and robustness when crashes happened at different locations of the parameter space.

- Applying the k-NN technique instead or next to the simple NN, as the former seems to be more powerful.

Response: We agree that the *k*-NN technique is an effective regression approach compared to the single NN. Considering our main goal in this study, i.e., finding simple yet effective strategies for handling simulation failures, we have adopted the single NN technique because it is a simple, persimmons method. Of course, more complex metamodeling options are available, including *k*-NN. The objective was not to provide a comprehensive comparison though. We have added the following paragraph in **Section 2.2.2** to address this comment:

> In this study, we choose to use the single NN technique with Euclidean distance measure. We do so because the single NN technique is very parsimonious and simple to understand and implement. To substitute the crashed simulations, the single-NN algorithm reads through whole dataset to find the nearest neighbour and then imputes the missing value with the model response of that nearest neighbour. It is noteworthy that some authors have asserted that covariances among Y variables are preserved in the NN-based techniques when using small k values (Hudak et al., 2008; McRoberts et al., 2002; Tomppo et al., 2002). But, McRoberts (2009) showed that the variance and covariance of the Y variables tend to be preserved for k = 1 but not for k > 1 (McRoberts, 2009). In general,

compared to the single NN-technique, the k-NN technique may provide a better fit to data but at the expense of being more complex and requiring a careful (and subjective) selection of the kernel functions and variable k. As a more complex technique, we suggest directly using a model emulation technique as described in the section below.

- Applying a convergence analysis on the SA results. It appears to me that the proposed approach only slows down the convergence of the results (so an evolution of the SA statistics for both the simulations without crashes and the simulations with suggested crashes should be performed. Possible approaches can be found in (Sarrazin et al., 2016) or (Nossent et al., 2011)).

Response: Thanks for providing these useful references. We think there is no point in this comparison, as when a model crashes the classic convergence analysis does not make sense because the GSA algorithm cannot be finished. In other words, since the crash handling strategies are applied after all runs are completed and without any need to repeat any experiment, these strategies should not matter for convergence. To emphasize on the importance of the convergence analysis for the GSA, the following paragraph has been added into the discussion section (**Section 5.2**)

It is important to note that the sample size in GSA studies should not only be determined based on the available computational budget but also considerations of GSA stability and convergence. Therefore, it is of vital importance to monitor and evaluate the convergence rate of the GSA algorithms. Strategies introduced by Nossent et al. (2011), Sarrazin et al. (2016), and more recently by Sheikholeslami et al. (2019) enable users to diagnose the convergence behaviour of the GSA algorithms.

- Applying different sampling techniques (as the density of the samples might have an influence on the results) (e.g. on p13, L33 one could argue that this statement should be supported by applying different strategies next to "STAR").

Response: Thanks for pointing out this comment. As mentioned in **Section 5.2**, regardless of the chosen method for handling simulation crash problem in GSA, it is advisable to spend some time up front to find an optimal sample set before submitting it for evaluation to the computationally expensive models. Therefore, we used an advanced sampling strategy called Progressive Latin Hypercube Sampling (PLHS) to ensure sufficient coverage of the parameter space (see **Section 3.3**). In other words, the STAR-VARS algorithm employs the PLHS strategy to locate star centers in the first phase of sampling. In addition, for the variance-based GSA, we applied Sobol quasi-random sequences combined with skipping, leaping, and scrambling operations to generate the base sample points.

- Adding information on the computation time of the different step.

Response: To address this comment, the following sentences have been added to **Section 3.3** of the revised manuscript:

The entire set of 100,000 function evaluations of the MESH model would take more than 6 months if we used a single CPU core. However, we used the University of Saskatchewan's high-performance computing system to run the GSA experiment in parallel on 160 cores. Therefore, completing all model runs required approximately 32 hours. For the MESH model, using an Intel® Core™ i7 CPU 4790 3.6GHz desktop PC, the RBF technique took only 65 seconds to substitute 3,084 crashed runs, while the single NN technique required about 97 seconds to complete the task.

Major

** p2, L27: In most cases, the samples for GSA are independent. This is important in the interpretation of your proposed strategy, so you should clearly mention this in the text.*

Response: Thanks to Reviewer 1 for this important comment. As we mentioned in the introduction section, in case of model crashes, re-running the entire experiment is inevitable when using a GSA technique that utilizes a sampling strategy with a particular structure. To address this comment, we have revised **Section 1.2** as follows:

> 3. Ignoring the crashed runs in GSA may only be seen relevant when using purely random (and independent) samples (i.e., Monte Carlo method). In such cases, if the model crashes at a given parameter set, one may simply exclude that parameter set or generate another random parameter set (at the expense of increased computational cost) that results in a successful simulation.
>
> 4. Some efficient sampling techniques follow specific spatial arrangements; examples include the variance-based GSA proposed by Saltelli et al. (2010) or STAR-VARS of Razavi and Gupta (2016b). In GSA enabled with such structured sampling techniques, we cannot ignore crashed simulations because excluding sample points associated with simulation crashes will distort the structure of the sample set, causing inaccurate estimation of sensitivity indices. As a result, the user may have to re-do a part or the entire experiment depending on the GSA implementation.

** p4, L22: In many cases of GSA, it is not necessary to re-run the entire experiment, but just a limited number of runs. This is important to put this into perspective.*

Response: Agreed. Please see our response to previous comment.

** p10, L11: What about parameter "CO"? It is influential, but you don't talk about that one.*

Response: Thanks for pointing this out. To address this comment, we have edited the text as follows:

> Moreover, Fig. 4 and 6 show that when crashes were substituted using the RBF technique, the STAR-VARS algorithm estimated the sensitivity indices of the most important parameters (FRAC, FC, C0) (Fig. 4) and less important parameters (LP, ETF, beta, K2) (Fig. 6) with high degrees of accuracy and robustness.

** p12, L18: I have the impression that you went into detail too much on these causes of crashes of this model, whereas your main focus should be on the SA. The part starting on p13, L10 can be maintained as it is an interesting addition to this topic.*

Response: As suggested by the reviewer, the length of this section has been reduced in the revised manuscript.

** p13, L26: Is this valid for both single NN and k-NN? Specify this.*

Response: Yes, the poorly sampled parameter space can also influence the performance of the *k*-NN technique. In regions of the parameter space where the sample points are sparsely distributed, distances to nearest neighbours can be high, leading to choosing physically incompatible neighbours. To clarify this point, we have modified the text as follows:

> For example, in the NN techniques (both single and *k*-NN) one major concern is that the sparseness of sample points may affect the quality of the results.

** p25, fig 7: Could you provide some additional figures of this type in annex? Although this is arbitrarily chosen, this would support the results.*

Response: We agree with the reviewer that reporting these figures might be of some value but adding these results would not change the discussions and conclusions already presented, and so we prefer not to add more figures to our already long paper having many figures. To further investigate the performance of the

single NN and RBF techniques, we have added a new figure (the upper panel of the **Figure 9**) into the revised manuscript.

Minor

Response: Thank you very much for editing and proofreading our manuscript.

** p1, L28: Should it be "Dynamical Earth Systems Models" or "Dynamical Earth System Models"?*
Response: As suggested, we used "Dynamical Earth System Models".

** p2, L29: Remove either "that" or "how"*
Response: Fixed.
** p3, L21: Replace "is" by "are"*
Response: Typo fixed.
** p3, L24: Add "the" before "parameter space"*
Response: Fixed.
** p4, L24: Replace "the" by "a"*
Response: Fixed.
** p5, L21: It would be either "a computationally simple method" or "computationally simplest method"*
Response: Corrected.
** p6, L18: Add "a" before "response"*
Response: Fixed.
** p6, L20: Add a comma after "In the literature"*
Response: Fixed.
** p6, L22: Add "the" before "RBF"*
Response: Fixed.
** p8, L7: Add "a" before "highly"*
Response: Fixed.
** p8, L8: Add "a" before "minimum"*
Response: Fixed.
** p8, L15: Add "a" or "the" before "maximum"*
Response: Fixed.
** p8, L16: Add "the" before "output"*
Response: Fixed.
** p8, L16: Add "a" before "minimum"*
Response: Fixed.
** p8, L24 (and others): All superscript numbers seem to be written as normal numbers*
Response: Typo fixed.
** p8, L26: "of which" should go before "10"*
Response: Corrected as suggested.
** p8, L29: The last sentence seems to have an odd structure*
Response: It has been modified.
** p9, L20: Add "the" before "STAR-VARS*
Response: Fixed.
** p9, L25: Add "the" before "GSA"*
Response: Fixed.
** p9: L30: I would suggest to move "when there are no crashes" between the brackets on the previous line ("after 9100 function evaluations").*
Response: Corrected as suggested.
** p10, L7: Add "the" before "parameter space"*

Response: Fixed.
** p10, L7: Remove the "s" from "ratios"*
Response: Typo fixed.
** p10, L10: Reformulate this sentence*

Response: We have modified p10, L10 and now it reads:

> Moreover, Fig. 4 and 6 show that when crashes were substituted using the RBF technique, the STAR-VARS algorithm estimated the sensitivity indices of the most important parameters (FRAC, FC, C0) (Fig. 4) and less important parameters (LP, ETF, beta, K2) (Fig. 6) with high degrees of accuracy and robustness.

** p11, L1: Add "the" before "Four" and before "water"*
Response: Fixed.
** p11, L10: Replace "with" by "between these"*
Response: Corrected.
** p11, L12: Add "a" before "vegetation"*
Response: Fixed.
** p11, L13: Add "the" before "soil"*
Response: Fixed.
** p11, L21: Reformulate "As shown"*

Response: It has been edited as follows:

> The STAR-VARS algorithm identified these parameters as weakly influential (very low IVARS-50 values) using the proposed crash handling techniques. However, the associated sensitivity indices obtained by the RBF imputation method are about two orders of magnitude larger for the parameters in the left panel (Fig.13 (a, c)) and about four orders of magnitude larger for the parameters in the right panel (Fig. 13 (b, d)) compared to those obtained by the single NN and median substitution methods.

** p11, L23: Add an "s" to "order"*
Response: Fixed.
** p11, L29: Remove the "7"*
Response: Fixed.
** p13, L30: Replace "depends" by "depending"*
Response: Typo fixed.
** p14, L1: Which feature? Reformulate this sentence*
Response: Corrected.
** p14, L6: Replace "are" by "should be"*
Response: Corrected.
** p14, L7: Add an "s" to "problem"*
Response: Typo fixed.
** p14, L25: Remove the "s" from "involves"*
Response: Typo fixed.
** p14, L31: "The efficiency of our proposed simulation based strategies was shown: : :"*
Response: Fixed.

** p14, L30: This is a very long, complex sentence.*

Response: This sentence has been reformulated in the revised manuscript and now it reads:

The high efficiency of our proposed substitution-based approach is of prominent importance, particularly when dealing with GSA of the computationally expensive models mainly because our proposed approach does not need repeating the entire experiment.

** p15, L11: "causing" instead of "casing"*
Response: Typo fixed.
** p15, L18: "understanding" instead of "understand"*
Response: Typo fixed.
** p23 caption: Remove "C0" from the list of "moderately influential parameters"*
Response: Corrected.

Reviewer 2

The authors argue to substitute data of failed simulation members in large ensemble simulations conducted for global parametric sensitivity analysis of dynamical earth system models. It is common for the models to crash for certain parameter value combinations that are randomly sampled from multidimensional parameter space using standard automated techniques. Using case studies, the authors show that it may be better to fill in the data from the failed experiments with data substitution techniques rather than the general practice of ignoring those experiments completely. The paper is generally well written and motivated. I point out my concerns below.

We greatly appreciate Reviewer 2 for reviewing the manuscript and providing positive evaluations.

1. The authors motivate the study well (Section 1.2). However, the authors state that the automated sampling method that they use - STAR-VARS breaks down if there are failed simulations for certain parameter combinations (Section 2.3). They do not provide a good reasoning for that, which I think is warranted. Are there other sampling methods that would not be sensitive to failed simulations? Why use STAR-VARS? Is the data substitution strategy only designed because of the limitation of STAR-VARS?

Thanks for this comment. As we mentioned in **Section 1.2**, those GSA techniques that use a sampling strategy with a specific structure will fail if the simulation model crashes at certain parameter configurations such as the widely-used variance-based method of Saltelli et al. (2010). To further explain, we have revised **Section 1.2** as follows:

> 3. Ignoring the crashed runs in GSA may only be seen relevant when using purely random (and independent) samples (i.e., Monte Carlo method). In such cases, if the model crashes at a given parameter set, one may simply exclude that parameter set or generate another random parameter set (at the expense of increased computational cost) that results in a successful simulation.
>
> 4. Some efficient sampling techniques follow specific spatial arrangements; examples include the variance-based GSA proposed by Saltelli et al. (2010) or STAR-VARS of Razavi and Gupta (2016b). In GSA enabled with such structured sampling techniques, we cannot ignore crashed simulations because excluding sample points associated with simulation crashes will distort the structure of the sample set, causing inaccurate estimation of sensitivity indices. As a result, the user may have to re-do a part or the entire experiment depending on the GSA implementation.

2. The impact of three data substitution techniques are compared. However, the first two methods are overly simplistic, and one can argue that they would yield poorer results a-priori - for example, the median is definitely not a good approximation for parameter combinations that are in the distribution tails, which may be more likely to crash. I do not see why the authors chose to present the results from those methods as one of their main results. It is fine to include them, but I think it would have been more useful to include results from different surrogate models, e.g. kriging, neural networks etc., which may be better as models for data substitution.

We certainly agree with reviewer that the median substitution is a very simple (perhaps naive) approach compared to other methods such as RBF. Nevertheless, we have adopted these methods considering our main goal in this study, which was finding simple and effective strategies for handling simulation failures. To improve the explanation, we have added the following statement to **Section 2.2.1**:

> In sampling-based optimization, one may assign a very poor objective function value (e.g., a very large objective function in the minimization case) to a crashed solution, similar to the big M method for handling optimization constraints (Camm et al., 1990). Our first strategy in the GSA context adopts such an approach. However, since replacing crashes with a big value can magnify the effect of the crashed runs in GSA, instead we suggest choosing a measure of central tendency such as

mean or median to minimize the impact of the implausible parameter configurations on the GSA results. Perhaps replacing each simulation crash with some "central" value is the easiest and a computationally simple method for imputation. Depending on the distribution of the model response variables Y, the central value can be median or mean. If the distribution of the model responses is not highly skewed, imputing the crashes with the mean of the non-missing values may work. However, if the distribution exhibits skewness, then the median may be a better replacement because the mean is sensitive to the outliers.

Regarding the application of other surrogate models (e.g., kriging, etc.) as we mentioned in **Section 2.2.3** depending on the complexity and dimensionality of the response surface, other types of metamodels can be incorporated into the proposed framework. We did not intend to compare the performance of different metamodelling techniques in this study, so we only applied the well-known RBF technique. Of course, this could be a potential direction for future research. To address this comment, we have added the following sentence to the conclusion section of the revised manuscript:

Finally, another possible future direction is to apply and test other types of emulation techniques such as kriging and support vector machine in handling model crashes.

3. The authors appear to consider simulation failure as numerical artefacts. It could well be that parameter combinations are unphysical resulting in genuine crashes. Substituting data for these model crashes would result in unrealistic sensitivity. Likewise, unrealistic parameter combinations could also result in successful runs without crashes distorting the sensitivity analysis. It will be good if the authors could discuss this. The authors discuss this partly in section 5.1 for MESH model while exploring the reasons of simulation failure, but do not seem to relate it to their substitution strategy which is their main point.

Thanks for this very good comment. The following paragraph has been added to **Section 5.1** of the revised manuscript to improve the discussion:

We conclude this section by highlighting a point that should receive careful attention when applying the substitution-based methods in handling model crashes. In addition to the numerical artefacts in simulation models, some combinations of parameter values, which may not be physically justified, can also lead to simulation failures. As a result, there is risks that substituting data for these crashed runs contaminate the assessment of parameter importance. Preventing this type of risks requires knowledge about the reasonable parameter ranges in DESMs. This type of crashes can be significantly reduced by selecting plausible ranges of parameters based on physical knowledge or information of the problem (a process referred to as "parameter space refinement" (see e.g., Li et al., 2019; Williamson et al., 2013)). However, DESMs often consist of many interacting, uncertain parameters, and therefore very little may be known a priori about the implausible regions of the parameter space.

4. The title reads as if something useful can be done with simulations that crashed. But, the strategy of the paper is to actually substitute the failed simulations. The authors should think about revising the title so that its not too misleading.

We greatly appreciate the reviewer for this valuable suggestion. The title has been modified in the revised manuscript as:

What should we do when a model crashes? Recommendations for global sensitivity analysis of earth and environmental systems models

What ~~do~~ should we do ~~with model simulation~~ when a model crashes? Recommendations for global sensitivity analysis of earth and environmental systems models

Razi Sheikholeslami[1,2], Saman Razavi[1,2,3], Amin Haghnegahdar[1,2]

[1]School of Environment and Sustainability, University of Saskatchewan, Saskatoon, Canada
[2]Global Institute for Water Security, University of Saskatchewan, Saskatoon, Canada
[3]Department of Civil, Geological, and Environmental Engineering, University of Saskatchewan, Saskatoon, Canada

Correspondence to: Razi Sheikholeslami (razi.sheikholeslami@usask.ca)

Abstract. Complex, software- intensive, technically advanced, and computationally demanding models, presumably with ever-growing realism and fidelity, have been widely used to simulate and predict the dynamics of the Earth and environmental systems. The parameter-induced simulation crash (failure) problem is typical across most of these models despite considerable efforts that modellers have directed at model development and implementation over the last few decades. A simulation failure mainly occurs due to the violation of the numerical stability conditions, non-robust numerical implementations, or errors in programming. However, the existing sampling-based analysis techniques such as global sensitivity analysis (GSA) methods, which require running these models under many configurations of parameter values, are ill equipped to effectively deal with model failures. To tackle this problem, we propose a new approach that allows users to cope with failed designs (samples) ~~during the~~when performing GSA~~, without knowing where they took place and~~ without re-running the entire experiment. This approach deems model crashes as missing data and uses strategies such as median substitution, single nearest neighbour, or response surface modelling to fill in for model crashes. We test the proposed approach on a 10-parameter HBV-SASK rainfall-runoff model and a 111-parameter MESH land surface-hydrology model. Our results show that response surface modelling is a superior strategy, out of the data filling strategies tested, and can comply with the dimensionality of the model, sample size, and the ratio of the number of failures to the sample size. Further, we conduct a "failure analysis" and discuss some possible causes of the MESH model failure that can be used for future model improvement.

1 Introduction

1.1 Background and motivation

Since the start of the digital revolution and subsequent increase in computers' processing power, the advancement of information technology has led to significant development of the modern software programs for Dynamical Earth System Models (DESMs). The current-generation DESMs typically span upwards of several thousand lines of code and require huge

amounts of data and computer memory. The flip side of the growing complexity of the DESMs is that running these models will pose many types of software development and implementation issues such as simulation crashes/failures. The simulation crash problem happens mainly due to violation of the numerical stability conditions needed in DESMs. Certain combinations of model parameter values, improper integration time step, inconsistent grid resolution, or lack of iterative convergence as well as model thresholds and sharp discontinuities in model response surfaces, all associated with imperfect parameterizations, can cause numerical artefacts and stop DESMs from properly functioning.

When model crashes occur, the accomplishment of automated sampling-based model analyses such as sensitivity analysis, uncertainty analysis, and optimization becomes challenging. These analyses are often carried out by running DESMs for a large number of parameter configurations randomly sampled from a domain (parameter space) (see, e.g., Raj et al., 2018; Williamson et al., 2017; Metzger et al., 2016; Safa et al., 2015). In such situations, for example, the model's solver may break down because of the implausible combinations of parameters ("unlucky parameter set" as termed by Kavetski et al., (2006)), failing to complete the simulation. ~~It is also possible that a model may~~It is also possible that a model will be stable against perturbation of ~~one~~ a single parameter, while it may crash when several parameters are perturbed simultaneously. "Failure analysis" is a process that is performed to determine the causes that have led to such crashes while running DESMs. Before achieving a conclusion on the most important causes of crashes, it is necessary to check the software code of the DESMs and confirm if it is error-free~~;~~ (e.g., if a proper numerical scheme has been adopted and correctly coded in the software). This often requires investigating both the software documentation and a series of nested modules. However, the existence of numerous nested programming modules in ~~a~~ typical DESMs can make the identification and removal of all software defects so tedious. In addition, as argued by Clark and Kavetski (2010), the numerical solution schemes implemented in DESMs are sometimes not presented in detail. This is one important reason why detecting the causes of simulation crashes in DESMs is usually troublesome. For example, Singh and Frevert (2002) and Burnash (1995) described the governing equations of their models without explaining the numerical solvers that were implemented in their codes.

Importantly, the impact of simulation crashes on the validity of global sensitivity analysis (GSA) results has often been overlooked in the literature, where simulation crashes ~~are~~ have been commonly classified as ignorable (see section 1.2). As such, a surprisingly limited number of studies have reported simulation crashes (examples related to uncertainty analysis include Annan et al., 2005; Edwards and Marsh, 2005; Lucas et al., 2013). This is despite the fact that these crashes can be very computationally costly for the GSA algorithms because they can waste the rest of the model runs, prevent completion of GSA, or inevitably introduce ambiguity into the inferences drawn from GSA. For example, Kavetski and Clark (2010) demonstrated ~~that~~ how numerical artefacts ~~can~~could contaminate the assessment of parameter sensitivities ~~in six hydrological models~~. Therefore, it is important to devise solutions that minimize the effect of crashes on GSA ~~results~~. In the next subsection, we critically review the very few strategies for handling simulation crashes that have been proposed in the literature and identify their shortcomings.

1.2 Existing approaches to handling simulation crashes in DESMs

We have identified, as outlined below, four types of approaches in the modelling community to handle simulation crashes~~, as outlined below~~. The first two are perhaps the most common approaches (based on our personal communications with several modellers); however, we could not identify any publication that formally reports their application:

1. After the occurrence of a crash, modellers commonly adopt a conservative strategy to address this problem by altering/reducing the feasible ranges of parameters and re-starting the experiment in a hope to prevent recurrence of the crashes in the new analyses.
2. Instead of GSA that runs many configurations of parameter values, analysts may choose to employ local methods such as local sensitivity analysis (LSA) through running the model only near the known plausible parameter configurations.
3. Some modellers may adopt an ignorance-based approach by using only a set of ''good'' (or behavioural) outcomes/responses in sampling-based analyses and ignoring unreasonable (or non-behavioural) outcomes such as simulation crashes. This can be done in conjunction with ~~via~~ defining a performance metric to ~~determine~~ choose which simulations to ~~should be~~ exclude~~d~~ from the analysis (see, e.g., Pappenberger et al., 2008; Kelleher et al., 2013).
4. The most rigorous approach seems to be a non-substitution approach that tries to predict whether or not a set of parameter values will lead to a simulation crash. Webster et al. (2004), Edwards et al. (2011), Lucas et al. (2013), Paja et al. (2016), and Treglown (2018) are among few studies that aimed at developing statistical methods to predict if a given combination of parameters can cause a failure. For example, Lucas et al. (2013) adopted a machine learning method to estimate the probability of crash occurrence as a function of model parameters. They further applied this approach to investigate the impact of various model parameters on simulation failures. A similar approach is based on model pre-emption strategies, where the simulation performance is monitored while the model is running and the model run is terminated early if it is predicted that the simulation will not be informative (Razavi et al. 2010; Asadzadeh et al., 2014).

The above approaches have some major limitations in handling simulation crashes in the GSA context, because:

1. Locating regions of the parameter space responsible for crashes (i.e., "implausible regions") is difficult and requires analysing the behaviour of the DESMs throughout the often high-dimensional parameter space. Implausible regions usually have irregular, discontinuous, and complex shapes, and thus are too effortful to identify. Additionally, altering/reducing the parameter space by excluding the implausible regions changes the original problem at hand.
2. It is well known that local methods (e.g., LSA) can provide inadequate assessments that can often be misleading (see e.g., Saltelli and Annoni, 2010, Razavi and Gupta, 2015).
3. ~~When applying a sampling-based technique that uses an ad-hoc sampling strategy with particular spatial structure (e.g., the variance-based GSA proposed by Saltelli et al. (2010) or STAR-VARS of Razavi and Gupta (2016b)), ignorance-~~

based procedures become impractical. In this case, excluding sample points associated with simulation crashes will distort the structure of the sample set, causing the failure of the entire GSA experiment. As a result, a new sample set (or a succession of sample sets) must be generated to resume the experiment, leading to a waste of previous model runs. Ignoring the crashed runs in GSA may only be seen relevant when using purely random (and independent) samples (i.e., Monte Carlo method). In such cases, if the model crashes at a given parameter set, one may simply exclude that parameter set or generate another random parameter set (at the expense of increased computational cost) that results in a successful simulation.

4. Some efficient -sampling techniques follow specific spatial arrangements; examples include the variance-based GSA proposed by Saltelli et al. (2010) or STAR-VARS of Razavi and Gupta (2016b). In GSA enabled with such structured sampling techniques, we cannot ignore crashed simulations because excluding sample points associated with simulation crashes will distort the structure of the sample set, causing inaccurate estimation of sensitivity indices. As a result, the user may have to re-do a part or the entire experiment depending on the GSA implementation.
5. Implementation of the non-substitution procedures necessitates significant prior efforts to identify a number of model crashes based on which a statistical model can be built, so as to predict and avoid simulation failures in the subsequent model runs. Such procedures can easily become infeasible in high-dimensional models, as ~~then~~ they would require an extremely large sample size to ensure an adequate coverage of the parameter space for characterizing implausible regions and building a reliable statistical model. These strategies can be more challenging when a model is computationally intensive. For example, to determine which parameters or combinations of parameters in a 16-dimensional climate model were predictors of failure, Edwards et al. (2011) used 1,000 evaluations (training samples) for constructing a statistical model to identify parameter configurations with high probability of failure in the next 1,087 evaluations (2,087 model runs in total). As pointed out by Edwards et al. (2011), although 2,087 evaluations might impose high computational burdens, a much larger sample size spreading out over the parameter space is required to guarantee reasonable exploration of the 16-dimensional space.

These shortcomings and gaps motivated our investigation to develop effective and efficient crash handling strategies suitable for GSA of the DESMs, as introduced in section 2.

1.3 Scope and outline

The primary goal of this study is to identify and test practical "substitution" strategies to handle the parameter-induced crash problem in GSA of the DESMs. Here, we treat model crashes as missing data and investigate the effectiveness of three efficient strategies to replace them using available information rather than discarding them. Our approach allows the user to cope with failed simulations in GSA without knowing where they will take place and without re-running the entire experiment. The overall procedure can be used in conjunction with any GSA technique. In this paper, we assess the performance of the proposed

substitution approach on two hydrological models, by coupling it with ~~the~~ a variogram-based GSA technique (VARS; Razavi and Gupta (2016a,b)).

The rest of the paper is structured as follows. We begin in the next section by introducing our proposed solution methodology for dealing with simulation crashes. In section 3, two real-world hydrological modelling case studies are presented. Next, in section 4, we evaluate the performance of the proposed methods across these real-world problems. The discussion is presented in section 5, before drawing conclusions and summarizing major findings in section 6.

2 Methodology

2.1 Problem statement

We denote the output of each model run (realization) $y(\boldsymbol{X})$, which corresponds to a d-dimensional input vector $\boldsymbol{X} = \{x_1, x_2, \dots, x_d\}$, where x_i $(i = 1,2, \dots, d)$ is a factor that may be perturbed for the purpose of GSA (e.g., model parameters, initial conditions, or boundary conditions). Running a GSA algorithm usually requires generating n realizations of a simulation model using an experimental design $\boldsymbol{X}^s = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_n\}^{\mathrm{T}}$ forming a $n \times d$ sample matrix. Then, the model responses will form an output space as $\boldsymbol{Y} = \{y(\boldsymbol{X}_1), y(\boldsymbol{X}_2), \dots, y(\boldsymbol{X}_n)\}^{\mathrm{T}}$. Here, we deem simulation crashes as missing data and consider the model mapping of $\boldsymbol{X}^s \rightarrow \boldsymbol{Y}$ as an incomplete data matrix. For a given $\boldsymbol{Y} \in \Re^{1\times n}$ with missing values, let the vector $\boldsymbol{Y}_a$ consist of the n_a locations in the input space for which, in the given $\boldsymbol{Y}$, the model responses are available, and let the vector $\boldsymbol{Y}_m$ consist of the remaining n_m locations ($n_m = n - n_a$) for which, in the given $\boldsymbol{Y}$, the model responses are missing due to simulation crashes. For convenience of expression and computation, we use the "NAN_j" symbol to represent the jth missing value in vector $\boldsymbol{Y}$. The main goal now is to develop and test data recovery methods that can be used to substitute model crashes $\boldsymbol{Y}_m$ using available information (i.e., $\boldsymbol{Y}_a$ and $\boldsymbol{X}^s$).

2.2 Proposed strategy for ~~crash~~ handling model crashes in GSA

We propose and test three techniques adopted from the "incomplete data analysis" for missing data replacement– the process known as imputation (Little and Rubin, 1987). ~~We use imputation techniques to fill in missing values~~ Our techniques do not account for the mechanisms leading to crashes because identifying such mechanisms can be very challenging (Liu and Gopalakrishnan, 2017). Therefore, only the non-missing responses and the associated sample points are included in our analysis to infill model crashes for GSA, as described in the next sub-sections.

2.2.1 Median substitution

In sampling-based optimization, one may assign a very poor objective function value (e.g., a very large objective function in the minimization case) to a crashed solution, similar to the big M method for handling optimization constraints (Camm et al., 1990). Our first strategy in the GSA context adopts such an approach. However, since replacing crashes with a big value can

magnify the effect of the crashed runs in GSA, instead we suggest choosing a measure of central tendency such as mean or median to minimize the impact of the implausible parameter configurations on the GSA results. ~~Perhaps replacing each simulation crash with some "central" value is the easiest and a computationally simple method for imputation. Depending on the distribution of the model response variables $\boldsymbol{Y}$, the central value can be median or mean. For example,if~~ If the distribution of the model responses is not highly skewed, imputing the crashes with the mean of the non-missing values may work. However, if the distribution exhibits skewness, then the median may be a better replacement because the mean is sensitive to the outliers. Therefore, we use the median substitution technique for the experiments reported in this paper. In general, this strategy treats each model response as a realization of a random function and ignores the covariance structure of the model responses~~, and thus considers the mean/median as a reasonable estimate for missing data~~. Also, ~~Although mean substitution preserves the mean of $\boldsymbol{Y}$,~~ a ~~major~~ shortcoming of this technique is that while it preserves the used measure of central tendency of $\boldsymbol{Y}$,~~, depending on the number of crashes,~~ it can distort other statistical ~~characteristics~~ properties of $\boldsymbol{Y}$, for example by ~~through~~ reducing its variance. ~~Thus, we use the median substitution technique in this paper.~~

2.2.2 Nearest neighbour substitution

The Nearest Neighbour (NN) technique (also known as hot deck imputation, see, e.g., Beretta and Santaniello~~,~~ (2016)) uses observations in the neighbourhood to fill in missing data. Let $\boldsymbol{X}_j \in \boldsymbol{X}^s$ be an input vector for which a simulation model fails to return an outcome. Basically, in the NN-based techniques, NAN_j is replaced by either a response value corresponding to a single nearest neighbour (single NN) or a weighted average of the response variables corresponding to k nearest neighbours (k-NN) where $k > 1$. The underlying rationale behind the NN-based techniques is that the sample points closer to $\boldsymbol{X}_j$ may provide better information for imputing NAN_j. In the k-NN techniques, weights are assigned based on the degree of similarity between $\boldsymbol{X}_j$ and the kth nearest neighbour $\boldsymbol{X}_k$, where $y(\boldsymbol{X}_k) \in \boldsymbol{Y}_a$, characterized through kernel functions (Tutz and Ramazan, 2015).

In this study, we choose to use the single NN technique with Euclidean distance measure. We do so because the single NN technique is very parsimonious and simple to understand and implement. To substitute the crashed simulations, the single-NN algorithm reads through whole dataset to find the nearest neighbour and then imputes the missing value with the model response of that nearest neighbour. It is noteworthy that some authors have asserted that covariances among $\boldsymbol{Y}$ variables are preserved in the NN-based techniques when using small k values (Hudak et al., 2008; McRoberts et al., 2002; Tomppo et al., 2002). But, McRoberts (2009) showed that the variance and covariance of the $\boldsymbol{Y}$ variables tend to be preserved for $k = 1$ but not for $k > 1$ (McRoberts, 2009). In general, compared to the single NN-technique, the k-NN technique may provide a better fit to data but at the expense of being more complex and requiring a careful (and subjective) selection of the kernel functions and variable k. As a more complex technique, we suggest directly using a model emulation technique as described in the section below.

2.2.3 Model emulation-based substitution

Model emulation is a strategy that develops statistical, cheap-to-run surrogates of response surfaces of complex, often computationally intensive models (Razavi et al., 2012a). Here we develop an emulator $\hat{y}(.)$, which is a statistical approximation of the simulation model based on a response surface modelling concept. This strategy consists in finding an approximate/surrogate model with low computational cost that fits the non-missing response values $\boldsymbol{Y}_a$ to predict the fill-in values for the missing responses $\boldsymbol{Y}_m$. In the literature, various types of response surface surrogates exist and have been extensively discussed (see, e.g., Razavi et al., 2012a). Examples are polynomial regression, radial basis functions (RBF), neural networks, kriging, support vector machines, and regression splines. Here, we employ the RBF approximation as a well-established surrogate model. It has been shown that RBF can provide an accurate emulation for high-dimensional problems (Jin et al., 2001; Herrera et al., 2011), particularly when the computational budget is limited (Razavi et al., 2012b). An RBF model as a weighted summation of n_a basis functions (and a polynomial or constant value) can approximate t~~T~~he predictive response $\hat{y}(\boldsymbol{X})$ at a sample point $\boldsymbol{X}$ ~~can be approximated by an RBF model as a weighted summation of n_a basis functions (and a polynomial or constant value)~~ as follows:

$$\hat{y}(\boldsymbol{X}) = \sum_{i=1}^{n_a} \omega_i f(\|\boldsymbol{X} - \boldsymbol{X}_i\|) = \boldsymbol{f}(\boldsymbol{X})\boldsymbol{\omega} \tag{1}$$

where $\boldsymbol{f} = \{f_1, f_2, \dots, f_{n_a}\}$ is the vector of the basis functions, ω_i is the *i*th component of the radial basis coefficient vector $\boldsymbol{\omega} = \{\omega_1, \omega_2, \dots, \omega_{n_a}\}^{\mathrm{T}}$, and $\|\boldsymbol{X} - \boldsymbol{X}_i\|$ is the Euclidian distance between two sample points.

There are various choices for the basis function, such as Gaussian, thin-plate spline, multi-quadric, and inverse multi-quadric (Jones, 2001). In the present study, we utilize the well-known Gaussian kernel function for RBF:

$$f(\|\boldsymbol{X} - \boldsymbol{X}_i\|) = exp\left(\frac{\|\boldsymbol{X} - \boldsymbol{X}_i\|^2}{c_i^2}\right) \tag{2}$$

where c_i is the shape parameter which determines the spread of the *i*th kernel function f_i.

After choosing the form of the basis function, the coefficient vector $\boldsymbol{\omega}$ can be obtained by enforcing the accurate interpolation condition, i.e.,

$$\begin{bmatrix} y(\boldsymbol{X}_1) \\ y(\boldsymbol{X}_1) \\ \vdots \\ y(\boldsymbol{X}_{n_a}) \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n_a} \\ f_{21} & f_{22} & \dots & f_{2n_a} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_a1} & f_{n_a2} & \dots & f_{n_an_a} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_{n_a} \end{bmatrix} \tag{3}$$

where $f_{uv} = f(\|\boldsymbol{X}_u - \boldsymbol{X}_v\|)$. In a matrix form, **Eq. (3)** can be simply rewritten as $\boldsymbol{Y}_a = \boldsymbol{F}\boldsymbol{\omega}$. This equation has a unique solution $\boldsymbol{\omega} = \boldsymbol{F}^{-1}\boldsymbol{Y}_a$ if and only if all the sample points are different from each other. Therefore, the fill-in values for remaining n_m locations, for which the model responses are missing due to simulation crashes, can be approximated by:

$$\hat{y}(\boldsymbol{X}_j) = \boldsymbol{f}(\boldsymbol{X}_j)\boldsymbol{F}^{-1}\boldsymbol{Y}_a \quad (j = 1,2,\dots,n_m) \tag{4}$$

To reduce the computational cost and avoid overfitting when building RBF, for each failed simulation at $\boldsymbol{X}_j$ one can choose k non-missing nearest neighbours of that missing value (here we arbitrarily set k = 100). Then, a function approximation can be built using these k sample points to approximate that missing value, i.e., in **Eq. (3)**, we set n_a to 100. Moreover, the shape parameter c in the Gaussian kernel function, which is an important factor in the accuracy of the RBF, can be determined using an optimization approach. We use the Nelder-Mead simplex direct search optimization algorithm (Lagarias et al., 1998) to find an optimal value for c by minimizing the RBF fitting error (for more details see Forrester and Keane (2009) and Kitayama and Yamazaki (2011)).

Note that in general depending on the complexity and dimensionality of the model response surfaces, other types of emulations can be incorporated into our proposed framework. However, for the crash handling problem, it is beneficial to utilize the function approximation techniques that exactly pass through the all sample points (i.e., the response surface surrogates categorized as "Exact Emulators" in Razavi et al. (2012a)) such as kriging and RBF. This is mainly because ~~that~~ most of the DESMs are deterministic, and therefore generate identical outputs/responses given the same set of input factors. In other words, an exact emulator at any successful sample point $\boldsymbol{X}_k$ (not crashed) reflects our knowledge about the true value of the model's output at that point, i.e., it returns $\hat{y}(\boldsymbol{X}_k)$ without any error. ~~Thus, exact emulators can be appropriate surrogates to for adequately characterize characterizing the shape of the response surfaces in deterministic DESMs for handling simulation crashes.~~

2.3 The utilized GSA frameworks

We illustrate the incorporation of the proposed crash handling methodology into a variogram-based GSA approach called Variogram Analysis of Response Surfaces (VARS; Razavi and Gupta (2016a)), and a variance-based GSA approach adopted from Saltelli et al. (2008). The VARS framework has successfully been applied to several real-world problems of varying dimensionality and complexity (Sheikholeslami et al., 2017; Yassin et al., 2017; Krogh et al., 2017; Leroux and Pomeroy, 2019; etc.). VARS is a general GSA framework that utilizes directional variograms and covariograms to quantify the full spectrum of sensitivity-related information, thereby providing a comprehensive set of the sensitivity measures called IVARS (Integrated Variogram Across a Range of Scales) at a range of different "perturbation scales" (Haghnegahdar and Razavi, 2017). Here, we use IVARS-50, referred to as "total-variogram effect", as a comprehensive sensitivity measure since it contains sensitivity analysis information across a full range of perturbation scales.

We utilize the STAR-VARS implementation of the VARS framework (Razavi and Gupta, 2016b). STAR-VARS is a highly efficient and statistically robust algorithm that provides stable results with a minimal number of model runs compared with other GSA techniques, and thus is suitable for high-dimensional problems (Razavi and Gupta, 2016b). This algorithm employs a star-based sampling scheme, which consists of two steps: (1) randomly selecting star centres in the parameter space, and (2) using a structured sampling technique to identify sample points revolved around the star centres. Due to the structured nature

of the generated samples in STAR-VARS, ignorance-based procedures (see section 1.2) cannot be useful in dealing with simulation crashes because deleting sample points associated with crashed simulations will demolish the structure of the entire sample set. Moreover, to achieve a well-designed computer experiment and sequentially locate star centres in the parameter space, we use the Progressive Latin Hypercube Sampling (PLHS) algorithm. It has been shown that PLHS can grasp the maximum amount of information from the output space with a minimum sample size, while outperforming traditional sampling algorithms (for more details see Sheikholeslami and Razavi, (2017)).

For the variance-based GSA, we calculate the total-effect index (Sobol-TO), which accounts for the impact of any individual parameter and its interaction with all other parameters, according to the widely used algorithm proposed by Saltelli et al. (2008). This algorithm follows a specific arrangement of randomly generated samples to calculate the sensitivity indices as follows: first, an $n \times 2d$ matrix of independent random numbers is generated (hereafter called "base sample"). Next, by splitting the base sample in half, two new sample matrices, $\boldsymbol{X}^A$ and $\boldsymbol{X}^B$, are built (each of size $n \times d$). Then, to calculate the ith sensitivity index TO_i, an additional sample matrix of size $n \times d$, $\boldsymbol{X}^{Ci}$ $(i = 1,2, \dots, d)$, is constructed by recombining the columns of $\boldsymbol{X}^A$ and $\boldsymbol{X}^B$ such that $\boldsymbol{X}^{Ci}$ contains the columns of $\boldsymbol{X}^B$ except the ith column which is taken from $\boldsymbol{X}^A$. To build the base sample, we use the Sobol quasi-random sequences. Furthermore, to achieve maximum space-filling properties and to maximize uniformity in the parameter space, for the given sample size, the skip, leap, and scramble operations are applied (for more details see Estrada 2017).

3 Case studies

3.1 A conceptual rainfall-runoff model

As an illustrative example, we applied the HBV-SASK conceptual hydrologic model to assess the performance of the proposed crash handling strategies. HBV-SASK is based on Hydrologiska Byråns Vattenbalansavdelning model (Lindström et al., 1997) and was developed by the second author for educational purposes (see Razavi et al., 2019; Gupta and Razavi, 2018). Here, we used HBV-SASK to simulate daily streamflows in the Oldman river basin in Western Canada (Fig. 1) with a watershed area of 1434.73 km^2. Historical data is available for periods 1979-2008, from which we estimate average annual precipitation to be 611 mm, and average annual streamflow to be 11.7 m^3/s with a runoff ratio of approximately 0.42. HBV-SASK has 12 parameters, of which 10 ~~of which~~ are perturbed in this study (Table 1).

3.2 A land surface-hydrology model

In the second case study, we demonstrate the utility of the imputation-based methods in crash handling via their application to the GSA of a high-dimensional and much more complex problem. We used the Modélisation Environmentale– Surface et Hydrologie (MESH; Pietroniro et al., (2007)) which is a semi-distributed, highly-parameterized land surface-hydrology modelling framework developed by Environment and Climate Change Canada (ECCC) mainly for large-scale watershed

modelling with consideration of cold region processes in Canada. MESH combines the vertical energy and water balance of the Canadian Land Surface Scheme (CLASS, Verseghy, 1991; Verseghy et al., 1993) with the horizontal routing scheme of the WATFLOOD (Kouwen et al., 1993). We encountered a series of simulation failures while assessing the impact of uncertainties in 111 model parameters (see Table A1 in Appendix A) on simulated daily streamflows in Nottawasaga river basin, Ontario, Canada (Fig. 3). For this case study, the drainage basin of nearly 2700 km^2 was discretized into 20 grid cells with a spatial resolution of 0.1667 degrees (~15 km). The dominant land cover in the area is cropland followed by deciduous forest and grassland. The dominant soil type in the area is sand followed by silt and clay loam (for more details see Haghnegahdar et al., 2015).

3.3 Experimental setup

In the first case study, for STAR-VARS, we chose to sample 100 star centres (with a resolution of 0.1) from the feasible ranges of parameters (Table 1) using the PLHS algorithm, resulting in 9,100 evaluations of the HBV-SASK model. For the variance-based method, the base sample size was chosen to be 5,000, and thus the model was run 60,000 times. The larger base sample size was selected for the variance-based method to ensure the stability of the algorithm. The Nash-Sutcliffe efficiency criterion on streamflows (NS) was used as the model output for sensitivity analysis. After calculating the NS values, we performed a series of experiments each with a different assumed "ratio of failure" (from 1% to 20%), defined as the percentage of failed parameter sets to the total number of parameter sets. In each experiment, we randomly selected a number of sampled points based on the associated ratio of failure and considered them as simulation failures. Then, we evaluated the performance of the crash handling strategies ~~to~~ in ~~replace~~ replacing simulation failures during GSA of the HBV-SASK model and compared the results with the case when there are no failures. In addition, we accounted for the randomness in the comparisons by carrying out 50 replicates of each experiment with different random seeds. This allowed us to see a range of possible performances for each strategy and to assess their robustness when crashes occurred at different locations in the parameter space.

In the second case study having 111 parameters, we only tested STAR-VARS with~~,~~ 100 star centres randomly generated using the PLHS algorithm (with a resolution of 0.1), resulting in ~~a total of 100,000~~100,000 MESH runs. The NS performance metric was used to measure daily model streamflow performance, calculated for a period of three years (October 2003-September 2007) following a one-year model warmup period. Due to various physical and/or numerical constraints inside MESH (or more precisely in CLASS), some combinations of the 111 parameters caused model crashes. Here, approximately 3% of our simulations failed (3,084 out of 100,000 runs). We applied the proposed crash handling strategies to infill the missing model outcomes in the GSA of the MESH model. The entire set of 100,000 function evaluations of the MESH model would take more than 6 months if we used a single standard CPU core. However, we used the University of Saskatchewan's high-performance computing system to run the GSA experiment in parallel on 160 cores. Therefore, completing all model runs required approximately 32 hours. For this case study, using an Intel® Core™ i7 CPU 4790 3.6GHz desktop PC, the RBF technique took only 65 seconds to substitute 3,084 crashed runs, while the single NN technique required about 97 seconds to complete the task.

4. Numerical results

4.1 Results for the HBV-SASK model

According to both of the IVARS-50 and Sobol-TO sensitivity indices, the parameters of the HBV-SASK (when there were no model crashes) were ranked as follows from the most important to the least important one: {*FRAC*, *FC*, *C0*, *TT*, *alpha*, *K1*, *LP*, *ETF*, *beta*, *K2*}. We assume these rankings and respective sensitivity indices as the "true" values. Based on the dendrogram (Fig. 3) generated by the factor grouping algorithm introduced by Sheikholeslami et al., (2019), we categorized these parameters into three groups with respect to their importance, i.e., {*FRAC*, *FC*, *C0*} are the strongly influential parameters, {*TT*, *alpha*, *K1*} are moderately influential parameters, and {*LP*, *ETF*, *beta*, *K2*} are weakly influential parameters.

Fig. 4, 5, and 6 show the cumulative distribution functions (CDFs) for the 50 independent estimates of IVARS-50, obtained when 1%, 3%, 5%, 8%, 10%, 12%, 15%, and 20% of model runs were deemed to be simulation failures. Overall, the RBF and single NN techniques outperformed the median substitution in terms of closeness to the true GSA results and robustness when crashes happened at different locations of the parameter space.

As can be seen, by increasing the ~~ration~~ ratio of failure, the performance of the crash handling strategies, particularly the median substitution became progressively worse. Note that the median substitution technique resulted in a significant bias manifested through over-estimation of the sensitivity indices for all the parameters. Moreover, Fig. 4 and 6 show that when crashes were substituted using the RBF technique, the STAR-VARS algorithm estimated the sensitivity indices of the most important parameters ~~({~~{*FRAC*, *FC*, *C0*}~~)~~ } (Fig. 4) and less important parameters ~~({~~{*LP*, *ETF*, *beta*, *K2*}~~)~~ } (Fig. 6) with high degrees of accuracy and robustness. However, for the moderately influential parameters {*TT*, *alpha*, *K1*} in Fig. 5, its performance degraded (i.e., the CDFs are wider in Fig. 5). The respective results using the variance-based algorithm are presented in Fig. B1, B2, and B3 for strongly influential, moderately influential, and weakly influential parameters, respectively (see Appendix B). Because our proposed approach for crash handling is GSA-method-free, we observed a similar performance when using the variance-based algorithm. In other words, the RBF effectively handled the crashes and produced reasonable sensitivity analysis results compared to the NN and median substitution techniques.

More importantly, as the number of crashes increases, rankings of the parameters in terms of their importance may change. Fig. 7 and 8 show the number of times out of 50 independent runs that the rankings of the parameters were equal to the "true" ranking for the STAR-VARS and variance-based GSA algorithms. In all 50 runs, regardless of the number of model crashes, the rankings obtained by the STAR-VARS using the RBF technique were equal to the "true" ranking, indicating a ~~is an indication of~~ high degree of robustness in terms of parameter ranking. The performance of the single NN slightly decreased when the crash percentage were more than 15%, while the STAR-VARS algorithm wrongly determined the rankings in more than 50% percent of the replicates using median substitution technique (see Fig. 7c and d). This highlights that the rankings can be estimated much more accurately than the sensitivity indices in the presence of simulation crashes. In addition, it can be seen that while the RBF-based strategy performed perfectly in this example, the performance of the single NN technique was comparably well (Fig. 7). However, for the variance-based technique, only the rankings of the most important parameters were

equal to the "true" ranking, regardless of the number of model crashes and the utilized crash handling strategy (Fig. 8). Moreover, the performance reduction of the single NN technique was higher when the variance-based method was employed. In fact, the variance-based algorithm wrongly estimated the rankings in more than 30% percent of the replicates using the single NN technique when the ratio of failure was 15% (Fig. 8c) and 20% (Fig. 8d).

Finally, Fig. 9 presents the performance of the single NN (Fig. 9a and c) and RBF (Fig. 9b and d) strategies in approximating the fill-in values for the missing responses when 5% (upper panel) and 20% (bottom panel) of the HBV-SASK simulations were deemed failures. As shown, the RBF outperformed single NN technique in terms of closeness to the true NS values. For example, having 20% of the model runs failed, the linear regression has an R^2 value of 0.834 when single NN was used, while the RBF strategy achieved a linear regression with an R^2 value of 0.996. In fact, the results of the RBF strategy are almost unbiased, as the linear regression plotted on Fig. 9b and d is very close to the ideal (1:1) line.

4.2 Results for the MESH model

We demonstrate the GSA results of the MESH model by categorizing the 111 parameters of the model into three groups as shown in Fig. 10 (for more details on grouping see Sheikholeslami et al. (2019)). Fig. 11-13 present the sensitivity analysis results obtained by the STAR-VARS algorithm for the MESH model, when we applied different crash handling strategies. These groups are labelled according to their importance, i.e., Group 1 (Fig. 11) contains the strongly influential parameters, while parameters in Group 2 (Fig. 12) are moderately influential, and Group 3 (Fig. 13) is the group of weakly influential parameters.

The f~~F~~our most influential parameters in Group 1 are *SDEPC* and *DRNC* ("C" stands for crops), controlling the water storage and water movement in the soil, *WFR22* (river channel routing), and *ZSN*L (snow cover fraction). As shown in Fig. 11 (upper panel), the sensitivity indices associated with these parameters are almost similar regardless of the employed crash handling technique. As discussed in our failure analysis (see Section 5.1), we also identified three of these parameters (i.e., *SDEPC*, *DRNC*, and *ZSNL*) responsible for some of the model crashes. In other words, ~~the~~ those parameters ~~which~~ that strongly contribute to the variability of the MESH model output can also be convicted of model crashes. To enhance future development and application of the MESH model, more efforts should be directed at better understanding the functioning of these parameters and their effects acting individually or in combination with other parameters over their entire range of variations.

For the other 15 influential parameters in Group 1 (Fig. 11, bottom panel), there is general agreement between these three crash handling techniques about the sensitivity indices calculated by the STAR-VARS except for the parameter *ROOTC*, which defines the annual maximum rooting depth of a vegetation category. The RBF and median substitution methods give more importance to *ROOTC* compared to the single NN technique. It is noteworthy that the oversaturation of the soil layer, which can cause many model runs to fail, is subject to the interaction between *ROOTC* and *SDEPC*.

Fig. 12 illustrates the sensitivity indices for the moderately influential parameters (i.e., Group 2). For all these 78 parameters, the sensitivity analysis results were highly dependent on the chosen crash handling strategy. As can be seen, the sensitivity

indices associated with the median substitution and RBF techniques are higher than those obtained by the single NN technique (this difference is considerable for the parameters in the upper and lower subplots than those in the middle subplot).

Finally, the results of the sensitivity analysis for the weakly or non-influential (Group 3) parameters of the MESH model are plotted in Fig. 13. ~~Although t~~The STAR-VARS algorithm identified these parameters as weakly influential (very low IVARS-50 values) using the proposed crash handling techniques. However, the associated sensitivity indices obtained by the RBF imputation method are ~~about~~ two orders of magnitude larger for the parameters in the left panel (Fig. 13 (a, c)) and about four orders of magnitude larger for the parameters in the right panel (Fig. 13 (b, d)) compared to those obtained by the single NN and median substitution methods.

It is important to note that in high-dimensional DESMs, when the number of parameters is very large, the estimation of sensitivity indices is likely not robust to sampling variability. On the other hand, parameter ranking (order of relative sensitivity) is often more robust to sampling variability and converges more quickly than factor sensitivity indices (see e.g., Vanrolleghem et al., 2015; Razavi and Gupta, 2016b; Sheikholeslami et al., 2019). To investigate how different crash handling strategies can affect the ranking of the model parameters in terms of their importance, Fig. 14 compares the rankings obtained by the RBF, single NN, and median substitution techniques.

As shown in Fig. 14a, the single NN and median substitution techniques resulted in almost similar parameter rankings for the strongly influential (Group 1) and weakly influential (Group 3) parameters, while for moderately influential parameters (Group 2) the rankings are significantly different. Meanwhile, the RBF and median substitution techniques yielded very distinctive rankings except for the strongly influential parameters (Fig. 14b). Furthermore, Fig. 14c indicates that the single NN and RBF methods ~~give~~ provided similar rankings for the influential parameters.

A closer examination, however, reveals that rankings can be very contradictory for some of the parameters~~,~~ when using different crash handling strategies (see Fig. 14d-f). For example, consider the soil moisture suction coefficient for crops (*PSGAC*) which is used in calculation of the stomatal resistance in the evapotranspiration process of the MESH (for more details see Fisher et al., 1981; Choudhury and Idso 1985; Verseghy, 2012). As can be seen, according to the RBF method, *PSGAC* is one of the weakly influential parameters (ranked 5th) (note that a ranking of 1 means the least influential, while ranking of 111 means the most influential parameter), while using the single NN it is determined to be one of the moderately influential parameters (ranked 43rd). In contrast, it is one of the strongly influential parameters based on the median substitution (ranked 83rd). However, in a comprehensive study of the MESH model using various model configurations and different hydroclimatic regions in Eastern and Western Canada, Haghnegahdar et al. (2017) found that *PSGAC* is one of the least influential parameters considering three model performance criteria with respect to high flows, low flows, and total flow volume of the daily hydrograph. As another example, consider *ZPLS7* (maximum water ponding depth for snow-covered areas) and *ZPLG7* (maximum water ponding depth for snow-free areas) which are used in surface runoff algorithm of the MESH (i.e., PDMROF). The single NN and median substitution methods both ranked *ZPLS7* as second and *ZPLG7* as third least influential parameters, whereas the RBF ranked them as 61 and 45 (i.e., moderately influential) which is in accordance with the results reported by Haghnegahdar et al. (2017).

5. Discussion

5.1 Potential causes of failure in the MESH

Our further investigations of the MESH model revealed at least two possible causes responsible for many of the simulation failures, i.e., the threshold behaviour of some parameters and oversaturation of the soil layers. For example, the threshold behaviour of the *ZSNL* (the snow depth threshold below which snow coverage is considered less than 100%) might cause many model crashes. When *ZSNL* was relatively large, it resulted in calculation of overly thick snow columns inside the model, violating the energy balance constraints and triggering a simulation abort. This situation became more severe when the calculated snow depth was larger than the maximum vegetation height(s). Fig. 15 (left column) shows the scatterplots of the *ZSNL* values sampled from the feasible ranges for all model simulations used for GSA of the MESH, with failed designs marked by red dots.

Furthermore, from our analysis we found that the oversaturation of the soil layer may happen especially at lower values of the soil permeable depth (*SDEP*) and also when it becomes less than the maximum vegetation rooting depth (*ROOT*). The situation is more severe when the soil drainage index (*DRN*) is reduced. These interactions can collectively cause a thinner soil column for water storage and movement that now has a lower chance for transpiration and drainage, thereby resulting in over accumulation of the water beyond the physical limits set for the soil in the model. Fig. 15 (right column) displays the pairwise scatterplots of the *SDEP*, *ROOT*, and *DRN*. To avoid model crashes, it is necessary to ensure that *SDEP* and *ROOT* values are not unrealistically low and that their values and/or their ranges are assigned as accurately as possible using the available data.

As can be seen from Fig. 15, very high values of parameters *DRNC* and *SDEPC* can also cause simulation crashes, while these crashes were happened at lower values of *ZSNL7*. Note that from these 2-dimensional projections of the 111-dimensional parameter space of the MESH no general conclusions can be drawn. This even becomes more complicated when noticing some isolated crashes in regions where most of the simulations were successful. Furthermore, as shown in Fig. 15, there are considerable overlaps between successful simulations and crashed ones in the feasible ranges of parameters. For example, there are many crashed simulations when *DRNC* was sampled from [3.5-4], at the same time a high density of successful simulations can also be observed in the same range. This indicates that locating regions of parameter space responsible for crashes is difficult, if not impossible, and necessitates analysing the MESH's response surface throughout a high-dimensional parameter space.

5.2 The role of sampling strategies in handling model crashes

Due to the extremely large parameter space ($\boldsymbol{X}$) of high-dimensional DESMs, it may require many properly distributed sample points ($\boldsymbol{X}_s$) to generate/explore a full spectrum of model behaviors such as simulation crashes, discontinuities, stable regions, optima, etc. Together with the computationally intensive nature of DESMs, this issue can make both non-substitution procedures and imputation-based methods (those proposed in the present study) very costly in dealing with crashes, if not

impractical. It is important to note that the sample size in GSA studies should not only be determined based on the available computational budget but also considerations of GSA stability and convergence. Therefore, it is of vital importance to monitor and evaluate the convergence rate of the GSA algorithms. Strategies introduced by Nossent et al. (2011), Sarrazin et al. (2016), and more recently by Sheikholeslami et al. (2019) enable users to diagnose the convergence behaviour of the GSA algorithms.

Because the non-substitution procedures rely on constructing a statistical model based on observed crashes to predict and avoid them in the follow-up experiments, they need a good coverage of the domain to attain a reliable statistical model. This issue also challenges the use of imputation-based methods. For example, in the NN techniques (both single and *k*-NN) one major concern is that the sparseness of sample points may affect the quality of the results. In regions of the parameter space where the sample points are sparsely distributed, distances to nearest neighbours can be relatively large, leading to choosing physically incompatible neighbours. Moreover, in response surface modelling-based techniques, building an accurate and robust function approximation ~~is~~ directly ~~depends~~ depend~~s~~ing on the utilized sampling strategy and how dense mappings between parameter and output spaces are (see, e.g., Jin et al., 2001; Mullur and Messac, 2006; Zhaou and Xue, 2010).

A crucial consideration in the use of any sampling strategy is the exploration ability of that strategy (i.e., space-filling~~ness~~ ability), which significantly influences the effectiveness of the utilized crash handling approach. When having this feature enabled (i.e., exploration), the non-substitution procedures can reliably identify implausible regions in the entire parameter space, meaning that the sample set is not confined to only a limited number of regions. Furthermore, it can notably improve the predictive accuracy of the response surface modelling-based methods (Crombecq et al., 2011). Exploration requires sample points to be evenly spread across the entire parameter space to ensure that all regions of the domain are equally explored, and thus sample points should be located almost equally apart. This feature rectifies the problems relating to the distances between sample points when using NN techniques since in space-filling designs these distances ~~are~~ should be as evenly distributed as possible.

Given this, regardless of the chosen method for solving simulation crash problem in GSA, it is advisable to spend some time up front to find an optimal sample set before submitting it for evaluation to the computationally expensive DESMs. It is, therefore, necessary to prudently use improved sampling algorithms such as Progressive Latin Hypercube Sampling (PLHS; Sheikholeslami and Razavi (2017)), K-extended Latin Hypercubes (k-extended LHCs; Williamson (2015)), or Sequential Exploratory Experimental Design (SEED; Lin (2004)). Generally, these sampling techniques optimize some characteristics of the sample points such as sample size, space-filling properties, projective properties, etc.

We conclude this section by highlighting a point that should receive careful attention when applying the substitution-based methods in handling model crashes. In addition to the numerical artefacts in simulation models, some combinations of parameter values, which may not be physically justified, can also lead to simulation failures. As a result, there is risks that substituting data for these crashed runs contaminate the assessment of parameter importance. Preventing this type of risks requires knowledge about the reasonable parameter ranges in DESMs. This type of crashes can be significantly reduced by selecting plausible ranges of parameters based on physical knowledge or information of the problem (a process referred to as "parameter space refinement" (see e.g., Li et al., 2019; Williamson et al., 2013)). However, DESMs often consist of many

interacting, uncertain parameters, and therefore very little may be known a priori about the implausible regions of the parameter space.

6 Conclusion

Understanding the complex physical processes in Earth and environmental systems and predicting their future behaviours rely routinely on high-dimensional, computationally expensive models. These models are often involved in the processes of model calibration, and/or uncertainty and sensitivity analysis. If a simulation failure/crash occurs at any of these processes~~stages~~, these models will stop functioning, and thus need user intervention. Generally, there are many reasons for failure of a simulation in models, including the use of inconsistent integration time steps or grid resolutions, lack of convergence, and inadequate threshold behaviours in models. Determining whether these "defects" exist in the utilized numerical schemes or they are programming bugs can only be done through analysing a high-dimensional parameter space and characterizing implausible regions responsible for crashes. This imposes a heavier computational burden on analysts. More importantly, every "crashed" simulation can be very demanding in terms of computational cost for global sensitivity analysis (GSA) algorithms because they can prevent the completion of the analysis and introduce ambiguity into the GSA results.

These challenges motivated us to implement missing data imputation-based strategies for handling simulation crashes in the GSA context. These strategies involve substituting plausible values for the failed simulations in the absence of a priori knowledge regarding the nature of the failures. Here, our focus was to find simple yet computationally frugal techniques to palliate the effect of model crashes on the GSA of Dynamical Earth System Models (DESMs). Thus, we utilized three techniques, including median substitution, single nearest neighbour, and emulation-based substitution (here we used radial basis functions as a surrogate model) to fill in a value for the failed simulations using available information and other non-missing model responses. The high efficiency of our proposed substitution-based approach is of prominent importance, particularly when dealing with GSA of the computationally expensive models mainly because our proposed approach does not need repeating the entire experiment.

We compared the performance of our approach in GSA of two modelling case studies in Canada, including a 10-parameter HBV-SASK conceptual hydrologic model and a 111-parameter MESH land surface-hydrology model. Our analyses revealed that:

- Overall, the emulation-based substitution can effectively handle the simulation crashes and produce promising sensitivity analysis results compared to the single nearest neighbour and median substitution techniques.
- As expected, the performance of the proposed methods deteriorates as the ratio of failures increases. The rate of degradation ~~is dependent~~depends on the number of model parameters (dimensionality of the parameter space).

- We observed in our experiments that the utilized crash handling strategy (i.e., median substitution, single NN, and RBF) has minimum influence on the rankings of the strongly and weakly influential parameters identified by the GSA algorithms, while for the moderately influential parameters, different strategies yielded different rankings.

Furthermore, we conducted a failure analysis for the second case study (MESH model) and identified some parameters that seem to be frequently causing model failures. Such analyses are helpful and much needed to improve the fidelity and numerical stability of the DESMs and may constitute a promising avenue of research. In doing so, applying other advanced methods (see e.g., Lucas et al. (2013)) can be beneficial to diagnose existing defects of the complex models.

Future work should include extending the proposed crash handling approach to time-varying sensitivity analysis of the DESMs because a comprehensive GSA requires a full consideration of the dynamical nature of the models. Our proposed approach can be integrated with any time-varying sensitivity analysis algorithm, for example, with the recently developed Generalized Global Sensitivity Matrix (GGSM) method (Gupta and Razavi, 2018; Razavi and Gupta, 2019). This further helps understanding the temporal variation of the parameter importance and model behaviour. Finally, another possible future direction is to apply and test other types of emulation techniques such as kriging and support vector machine in handling model crashes.

Code Availability.

The MATLAB codes for the proposed crash handling approach and the HBV-SASK model are included in the VARS-TOOL software package, which is a comprehensive, multi-algorithm toolbox for sensitivity and uncertainty analysis (Razavi et al., 2019). VARS-TOOL is freely available for non-commercial use and can be downloaded from http://vars-tool.com/. The most recent version of the MESH model can be downloaded from https://wiki.usask.ca/display/MESH/Releases. Additional data and information are available upon request from the authors.

Appendix A: Parameters of the MESH model

Parameters of the MESH model and their corresponding groups are listed in Table A. The description of parameters and their feasible ranges can be found in Haghnegahdar et al. (2017).

Appendix B: Performance of the crash handling strategies in sensitivity analysis of the HBV-SASK model using the variance-based algorithm

Author contributions.

All authors contributed to conceiving the ideas of the study. RS and SR designed the method and experiments. The simulations for the first case study were carried out by RS. AH developed the second case study and performed the MESH simulations. RS developed the MATLAB codes for the proposed crash handling approach and conducted all the experiments. RS wrote the manuscript with contributions from SR and AH. All authors contributed to the interpretation of the results, structuring and editing of the paper at all stages.

Competing interests.

The authors declare that they have no conflict of interest.

References

Annan, J. D., Hargreaves, J.C., Edwards, N.R., and Marsh, R.: Parameter estimation in an intermediate complexity earth system model using an ensemble Kalman filter, Ocean. Model., 8, 135–154, https://doi.org/10.1016/j.ocemod.2003.12.004, 2005.

Asadzadeh, M., Razavi, S., Tolson, B. A., and Fay, D.: Pre-emption strategies for efficient multi-objective optimization: Application to the development of Lake Superior regulation plan, Environ. Modell. Softw., 54, 128–141, https://doi.org/10.1016/j.envsoft.2014.01.005, 2014.

Beretta, L., and Santaniello, A.: Nearest neighbor imputation algorithms: a critical evaluation, BMC. Med. Inform. Decis., 16(3), 74, https://doi.org/10.1186/s12911-016-0318-z, 2016.

Burnash, R. J. C.: The NWS River forecast system-catchment modeling, in: Computer Models of Watershed Hydrology, edited by Singh, V. P., Water Resources Publication, Highlands Ranch, Colorado, USA, 311–366, 1995.

Choudhury, B. J., and Idso, S. B.: An empirical model for stomatal resistance of field-grown wheat, Agr. Forest. Meteorol., 36(1), 65–82, https://doi.org/10.1016/0168-1923(85)90066-8, 1985.

Camm, J.D., Raturi, A.S. and Tsubakitani, S.: Cutting big M down to size, Interfaces, 20(5), 61–66, https://doi.org/10.1287/inte.20.5.61,1990.

Clark, M. P., and Kavetski, D.: Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes, Water. Resour. Res., 46(10). https://doi.org/10.1029/2009WR008894, 2010.

Crombecq, K., Laermans, E., and Dhaene, T.: Efficient space-filling and non-collapsing sequential design strategies for simulation-based modelling, Eur. J. Oper. Res., 214(3), 683–696. https://doi.org/10.1016/j.ejor.2011.05.032, 2011.

Edwards, N. R., and Marsh, R.: Uncertainties due to transport-parameter sensitivity in an efficient 3-D ocean-climate model, Clim. Dynam., 24(4), 415–433. https://doi.org/10.1007/s00382-004-0508-8, 2005.

Edwards, N. R., Cameron, D., and Rougier, J.: Precalibrating an intermediate complexity climate model, Clim. Dynam., 37(7-8), 1469–1482. https://doi.org/10.1007/s00382-010-0921-0, 2011.

Estrada, E.: Quasirandom geometric networks from low-discrepancy sequences. Phys. Rev. E., 96, 022314. https://doi.org/10.1103/PhysRevE.96.022314, 2017.

Fisher, M. J., Charles-Edwards, D. A., and Ludlow, M. M.: An analysis of the effects of repeated short-term soil water deficits on stomatal conductance to carbon dioxide and leaf photosynthesis by the legume Macroptilium atropurpureum cv. Siratro, Func. Plant. Biol., 8(3), 347–357. https://doi.org/10.1071/PP9810347, 1981.

Forrester, A. I., Keane, A. J.: Recent advances in surrogate-based optimization, Prog. Aerosp. Sciences., 45(1-3), 50–79. https://doi.org/10.1016/j.paerosci.2008.11.001, 2009.

Gupta, H. V., and Razavi, S.: Revisiting the basis of sensitivity analysis for dynamical Earth system models, Water. Resour. Res., 54, 8692–8717. https://doi.org/10.1029/2018WR022668, 2018.

Haghnegahdar, A., and Razavi, S.: Insights into sensitivity analysis of earth and environmental systems models: On the impact of parameter perturbation scale, Environ. Modell. Softw., 95, 115–131. https://doi.org/10.1016/j.envsoft.2017.03.031, 2017.

Haghnegahdar, A., Razavi, S., Yassin, F., and Wheater, H., Multicriteria sensitivity analysis as a diagnostic tool for understanding model behaviour and characterizing model uncertainty, Hydrol. Process., 31(25), 4462–4476., https://doi.org/10.1002/hyp.11358, 2017.

Haghnegahdar, A., Tolson, B. A., Craig, J. R., and Paya, K.T.: Assessing the performance of a semi-distributed hydrological model under various watershed discretization schemes, Hydrol. Process., 29(18), 4018–4031. https://doi.org/10.1002/hyp.10550, 2015.

Herrera, L. J., Pomares, H., Rojas, I., Guillén, A., Rubio, G., and Urquiza, J.: Global and local modelling in RBF networks, Neurocomputing, 74(16), 2594–2602. https://doi.org/10.1016/j.neucom.2011.03.027, 2011.

Hudak, A.T., Crookston, N.L., Evans, J.S., Hall, D.E. and Falkowski, M.J.: Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data, Remote. Sens. Environ., 112(5), 2232–2245. https://doi.org/10.1016/j.rse.2007.10.009, 2008.

Jin, R., Chen, W., and Simpson, T. W.: Comparative studies of metamodelling techniques under multiple modelling criteria, Struct. Multidiscip. O., 23(1), 1–13. https://doi.org/10.1007/s00158-001-0160-4, 2001.

Jones, D. R.: A taxonomy of global optimization methods based on response surfaces, J. Global. Optim., 21(4), 345–383. https://doi.org/10.1023/A:1012771025575, 2001.

Kavetski, D., and Clark, M. P.: Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction, Water. Resour. Res., 46(10). https://doi.org/10.1029/2009WR008896, 2010.

Kavetski, D., Kuczera, G., and Franks, S. W.: Calibration of conceptual hydrological models revisited: 1. Overcoming numerical artefacts, J. Hydrol., 320(1-2), 173–186. https://doi.org/10.1016/j.jhydrol.2005.07.012, 2006.

Kelleher, C., Wagener, T., McGlynn, B., Ward, A. S., Gooseff, M. N., and Payn, R. A.: Identifiability of transient storage model parameters along a mountain stream, Water. Resour. Res., 49(9), 5290–5306. https://doi.org/10.1002/wrcr.20413, 2013.

Kitayama, S., and Yamazaki, K.: Simple estimate of the width in Gaussian kernel with adaptive scaling technique, Appl. Soft. Comp., 11(8), 4726–4737. https://doi.org/10.1016/j.asoc.2011.07.011, 2011.

Kouwen, N., Soulis, E. D., Pietroniro, A., Donald, J., and Harrington, R. A.: Grouped response units for distributed hydrologic modelling, J. Water. Res. Plan. Man., 119(3), 289–305. https://doi.org/10.1061/(ASCE)0733-9496(1993)119:3(289), 1993.

Krogh, S. A., Pomeroy, J. W., and Marsh, P.: Diagnosis of the hydrology of a small Arctic basin at the tundra-taiga transition using a physically based hydrological model, J. Hydrol., 550, 685–703. https://doi.org/10.1016/j.jhydrol.2017.05.042, 2017.

Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., Convergence properties of the Nelder–Mead simplex method in low dimensions, SIAM. J. Optimiz., 9 (1), 112–147. https://doi.org/10.1137/S1052623496303470, 1998.

Leroux, N. R., and Pomeroy, J. W.: Simulation of capillary overshoot in snow combining trapping of the wetting phase with a non-equilibrium Richards equation model, Water. Resour. Res., 54, https://doi.org/10.1029/2018WR022969, 2019.

Li, S., Rupp, D.E., Hawkins, L., Mote, P.W., McNeall, D., Sparrow, S.N., Wallom, D.C., Betts, R.A. and Wettstein, J.J.,: Reducing climate model biases by exploring parameter space with large ensembles of climate model simulations and statistical emulation. Geosci. Model. Dev., 12(7), 3017–3043. https://doi.org/10.5194/gmd-12-3017-2019, 2019.

Lin, Y.: An Efficient Robust Concept Exploration Method and Sequential Exploratory Experimental Design, Ph.D. thesis, Georgia Institute of Technology, USA, 2004.

Lindström, G., Johansson, B., Persson, M., Gardelin, M., and Bergström, S.: Development and test of the distributed HBV-96 hydrological model, J. Hydrol., 201(1-4), 272–288, 1997.

Little, R. J. A., and Rubin, D. B.: Statistical Analysis with Missing Data, John Wiley & Sons, New York, USA, 1987.

Liu, Y., and Gopalakrishnan, V.: An overview and evaluation of recent machine learning imputation methods using cardiac imaging data, Data, 2(1), 8. https://doi.org/10.3390/data2010008, 2017.

Lucas, D. D., Klein, R., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D., and Zhang, Y.: Failure analysis of parameter-induced simulation crashes in climate models, Geosci. Model. Dev., 6(4), 1157–1171. https://doi.org/10.5194/gmd-6-1157-2013, 2013.

McRoberts, R. E.: Diagnostic tools for nearest neighbors techniques when used with satellite imagery, Remote. Sens. Environ., 113(3), 489–499. https://doi.org/10.1016/j.rse.2008.06.015, 2009.

McRoberts, R.E., Nelson, M.D. and Wendt, D.G.: Stratified estimation of forest area using satellite imagery, inventory data, and the k-Nearest Neighbors technique, Remote. Sens. Environ., 82(2-3), 457–468. https://doi.org/10.1016/S0034-4257(02)00064-0, 2002.

Metzger, C., Nilsson, M. B., Peichl, M., and Jansson, P. E.: Parameter interactions and sensitivity analysis for modelling carbon heat and water fluxes in a natural peatland, using CoupModel v, Geosci. Model. Dev., 9(12), 4313–4338. https://doi.org/10.5194/gmd-9-4313-2016, 2016.

Mullur, A. A., and Messac, A.: Metamodeling using extended radial basis functions: a comparative approach, Eng. Comput., 21(3), 203–217. https://doi.org/10.1007/s00366-005-0005-7, 2006.

Paja, M., Wrzesien, M., Niemiec, R., and Rudnicki, W. R.: Application of all-relevant feature selection for the failure analysis of parameter-induced simulation crashes in climate models, Geosci. Model. Dev., 9(3), 1065–1072. https://doi.org/10.5194/gmd-9-1065-2016, 2016.

Pappenberger, F., Beven, K. J., Ratto, M., and Matgen, P.: Multi-method global sensitivity analysis of flood inundation models, Adv. Water. Resour., 31(1), 1–14. https://doi.org/10.1016/j.advwatres.2007.04.009, 2008.

Pietroniro, A., Fortin, V., Kouwen, N., Neal, C., Turcotte, R., Davison, B., Verseghy, D., Soulis, E. D., Caldwell, R., Evora, N., and Pellerin, P.: Development of the MESH modelling system for hydrological ensemble forecasting of the Laurentian Great Lakes at the regional scale, Hydrol. Earth. Syst. Sc., 11(4), 1279–1294. https://doi.org/10.5194/hess-11-1279-2007, 2007.

Raj, R., Tol, C. V. D., Hamm, N. A. S., and Stein, A.: Bayesian integration of flux tower data into a process-based simulator for quantifying uncertainty in simulated output, Geosci. Model. Dev., 11(1), 83–101. https://doi.org/10.5194/gmd-11-83-2018, 2018.

Razavi, S., and Gupta, H. V.: A multi-method generalized global sensitivity matrix approach to accounting for the dynamical nature of Earth and environmental systems models, Environ. Modell. Softw., In press, https://doi.org/10.1016/j.envsoft.2018.12.002, 2019.

Razavi, S., and Gupta, H. V.: What do we mean by sensitivity analysis? The need for comprehensive characterization of "global" sensitivity in Earth and Environmental systems models, Water. Resour. Res., 51(5), 3070–3092. https://doi.org/10.1002/2014WR016527, 2015.

Razavi, S., and Gupta, H. V.: A new framework for comprehensive, robust, and efficient global sensitivity analysis: 1. Theory, Water. Resour. Res., 52, 423–439. https://doi.org/10.1002/2015WR017558, 2016a.

Razavi, S., and Gupta, H. V.: A new framework for comprehensive, robust, and efficient global sensitivity analysis: 2. Application, Water. Resour. Res., 52, 440–455. https://doi.org/10.1002/2015WR017559, 2016b.

Razavi, S., Sheikholeslami, R., Gupta, H. V., and Haghnegahdar, A.: VARS-TOOL: A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis. Environ. Modell. Softw., 112, 95–107. https://doi.org/10.1016/j.envsoft.2018.10.005, 2019.

Razavi, S., Tolson, B. A., Burn, D. H.: Review of surrogate modeling in water resources, Water. Res. Res., 48(7), W07401, https://doi.org/10.1029/2011WR011527, 2012a.

Razavi, S., Tolson, B. A., Burn, D. H., Numerical assessment of metamodelling strategies in computationally intensive optimization, Environ. Modell. Softw., 34, 67–86. https://doi.org/10.1016/j.envsoft.2011.09.010, 2012b.

Razavi, S., Tolson, B. A., Matott, L. S., Thomson, N. R., MacLean, A., and Seglenieks, F. R.: Reducing the computational cost of automatic calibration through model pre-emption, Water. Resour. Res., 46, W11523, https://doi.org/10.1029/2009WR008957, 2010.

Safta, C., Ricciuto, D. M., Sargsyan, K., Debusschere, B., Najm, H. N., Williams, M., and Thornton, P. E., Global sensitivity analysis, probabilistic calibration, and predictive assessment for the data assimilation linked ecosystem carbon model, Geosci. Model. Dev., 8, 1899–1918. https://doi.org/10.5194/gmd-8-1899-2015, 2015.

Saltelli, A., and Annoni, P.: How to avoid a perfunctory sensitivity analysis, Environ. Modell. Softw., 25(12), 1508–1517. https://doi.org/10.1016/j.envsoft.2010.04.012, 2010.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., and Tarantola, S., Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index, Comput. Phys. Commun., 181(2), 259–270. https://doi.org/10.1016/j.cpc.2009.09.018, 2010.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global Sensitivity Analysis. The Primer. Wiley. 2008.

Sheikholeslami, R., and Razavi, S.: Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models, Environ. Modell. Softw., 93, 109–126. https://doi.org/10.1016/j.envsoft.2017.03.010, 2017.

Sheikholeslami, R., Razavi, S., Gupta, H. V., Becker, W., and Haghnegahdar, A.: Global sensitivity analysis for high-dimensional problems: how to objectively group factors and measure robustness and convergence while reducing computational cost, Environ. Modell. Softw., 111, 282–299. https://doi.org/10.1016/j.envsoft.2018.09.002, 2019.

Sheikholeslami, R., Yassin, F., Lindenschmidt, K. E., and Razavi, S.: Improved understanding of river ice processes using global sensitivity analysis approaches, J. Hydrol. Eng., 22(11), p.04017048. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001574, 2017.

Singh, V. P., Frevert, D. K.: Mathematical Models of Small Watershed Hydrology and Applications, 950 pp., Water Resources Publication, Highlands Ranch, Colorado, USA, 2002.

Tomppo, E., Nilsson, M., Rosengren, M., Aalto, P. and Kennedy, P.: Simultaneous use of Landsat-TM and IRS-1C WiFS data in estimating large area tree stem volume and aboveground biomass, Remote. Sens. Environ., 82(1), 156–171. https://doi.org/10.1016/S0034-4257(02)00031-7, 2002.

Tutz, G., and Ramzan, S.: Improved methods for the imputation of missing data by nearest neighbor methods, Comput. Stat. Data. An., 90, 84–99. https://doi.org/10.1016/j.csda.2015.04.009, 2015.

Vanrolleghem, P. A., Mannina, G., Cosenza, A., and Neumann, M. B.: Global sensitivity analysis for urban water quality modelling: Terminology, convergence and comparison of different methods, J. Hydrol., 522, 339–352. https://doi.org/10.1016/j.jhydrol.2014.12.056, 2015.

Verseghy, D. L.: CLASS—A Canadian land surface scheme for GCMs, I. Soil model, Int. J. Climatol., 11(2), 111–133, https://doi.org/10.1002/joc.3370110202, 1991.

Verseghy, D. L., McFarlane, N. A,. and Lazare, M.: CLASS— A Canadian land surface scheme for GCMs, II. Vegetation model and coupled runs, Int. J. Climatol., 13(4), 347–370, https://doi.org/10.1002/joc.3370130402, 1993.

Verseghy, D.: CLASS – the Canadian Land Surface Scheme (Version 3.6), Technical Documentation, Science and Technology Branch, Environment and Climate Change Canada, Toronto, Tech. Rep., 179 pp. 2012.

Webster, M., Scott, J., Sokolov, A. and Stone, P.: Estimating probability distributions from complex models with bifurcations: The case of ocean circulation collapse, J. Environ. Syst., 31, 1–21, https://doi.org/10.2190/A518-W844-4193-4202, 2004.

Williamson, D., Goldstein, M., Allison, L., Blaker, A., Challenor, P., Jackson, L., and Yamazaki, K.: History matching for exploring and reducing climate model parameter space using observations and a large perturbed physics ensemble, Clim. Dynam., 41, 1703–1729, https://doi.org/10.1007/s00382-013-1896-4, 2013.

Williamson, D.: Exploratory ensemble designs for environmental models using k-extended Latin Hypercubes. Environmetrics, 26(4), 268–283. https://doi.org/10.1002/env.2335, 2015.

Williamson, D. B., Blaker, A. T., and Sinha, B.: Tuning without over-tuning: parametric uncertainty quantification for the NEMO ocean model, Geosci. Model. Dev., 10(4), 1789–1816. https://doi.org/10.5194/gmd-10-1789-2017, 2017.

Yassin, F., Razavi, S., Wheater, H., Sapriza-Azuri, G., Davison, B., and Pietroniro, A.: Enhanced identification of a hydrologic model using streamflow and satellite water storage data: a multi-criteria sensitivity analysis and optimization approach, Hydrol. Process., 31, 3320–3333. https://doi.org/10.1002/hyp.11267, 2017.

Zhao, D., and Xue, D., A comparative study of metamodeling methods considering sample quality merits. Struct. Multidiscip. O., 42(6), 923–938. https://doi.org/10.1007/s00158-010-0529-3, 2010.

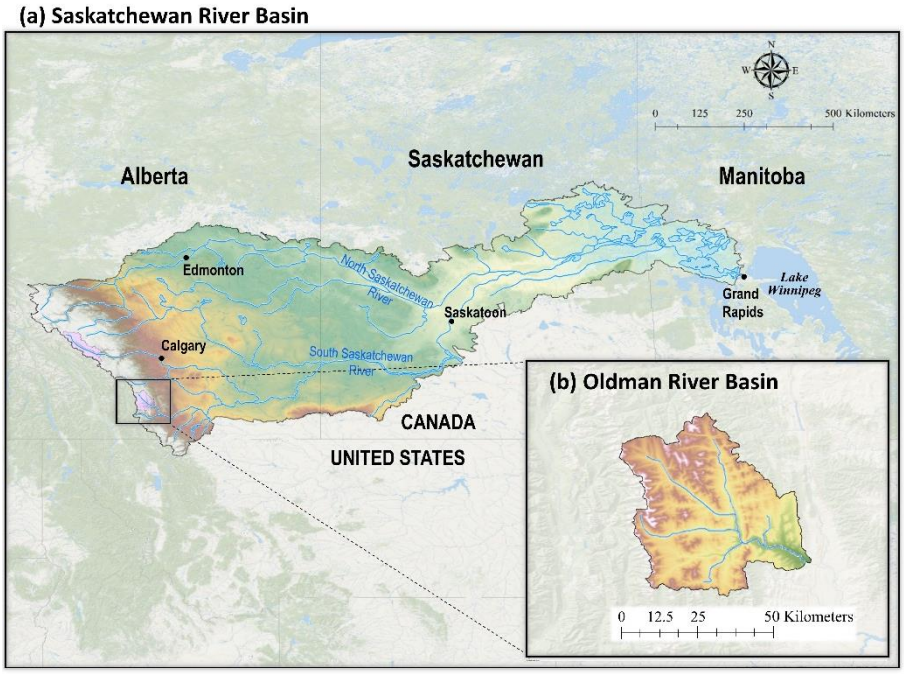

Figure 1: Oldman river basin (a) located in the Rocky Mountains in Alberta, Canada, flows into the Saskatchewan River Basin (b).

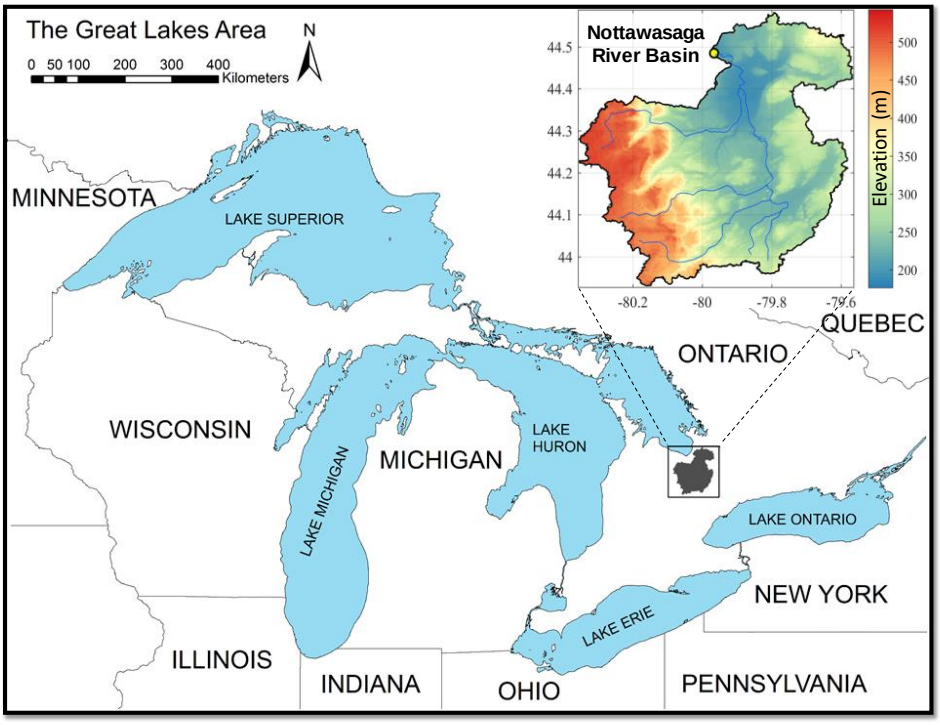

Figure 2: Nottawasaga river basin in in Southern Ontario, Canada.

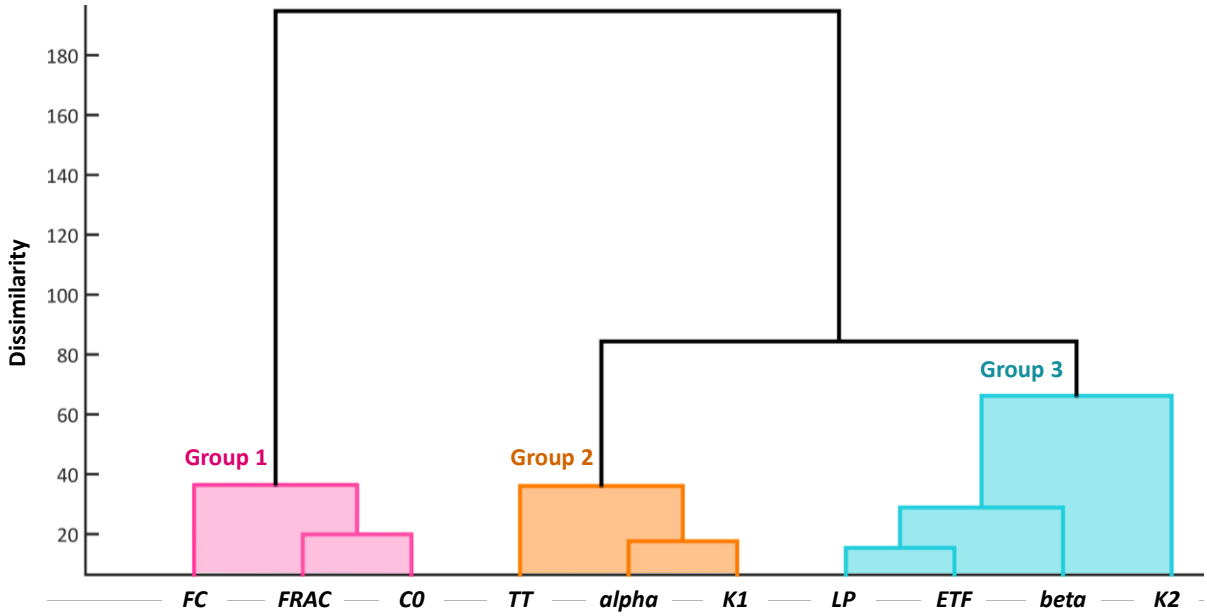

Figure 3: Grouping of the 10 parameters of the HBV-SASK model when applied on the Oldman River Basin. The parameters are sorted from the most influential (to the left) to the least influential (to the right).

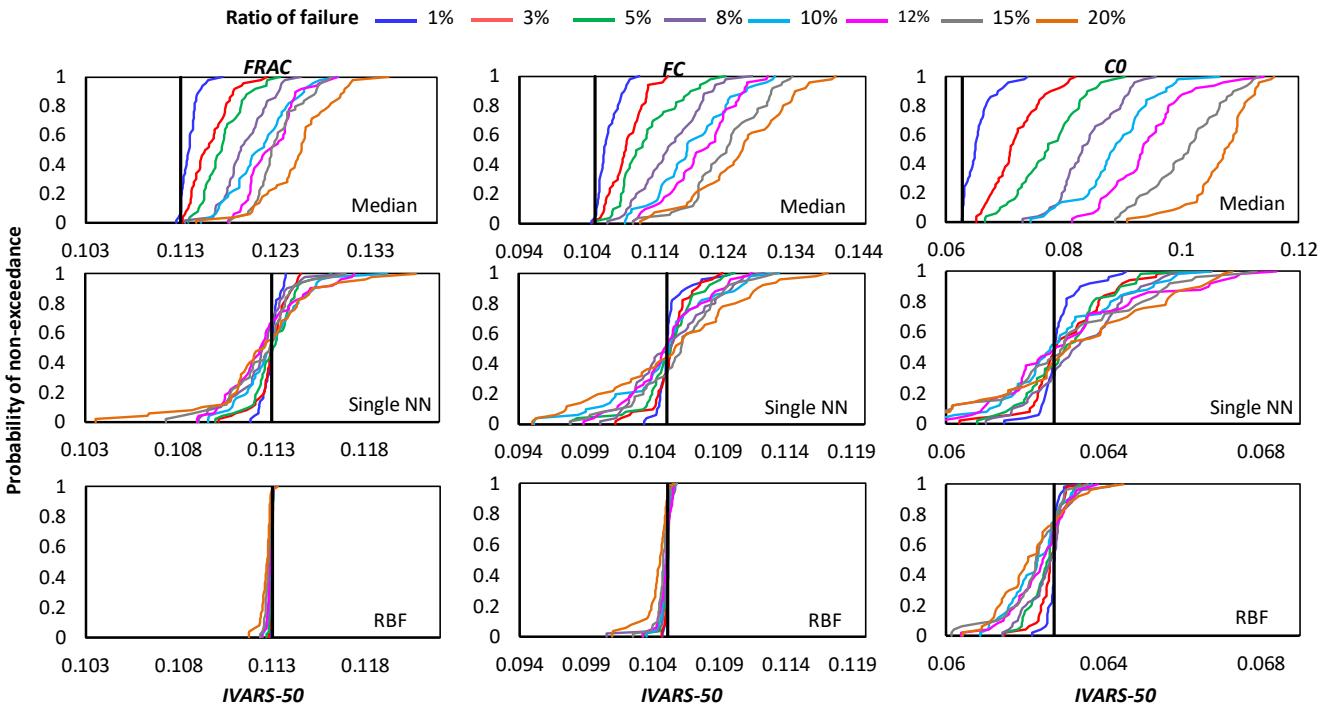

Figure 4: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the STAR-VARS algorithm for different ratios of failures. The CDFs of the sensitivity indices for strongly influential parameters {*FRAC*, *FC*, *C0*} are compared in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

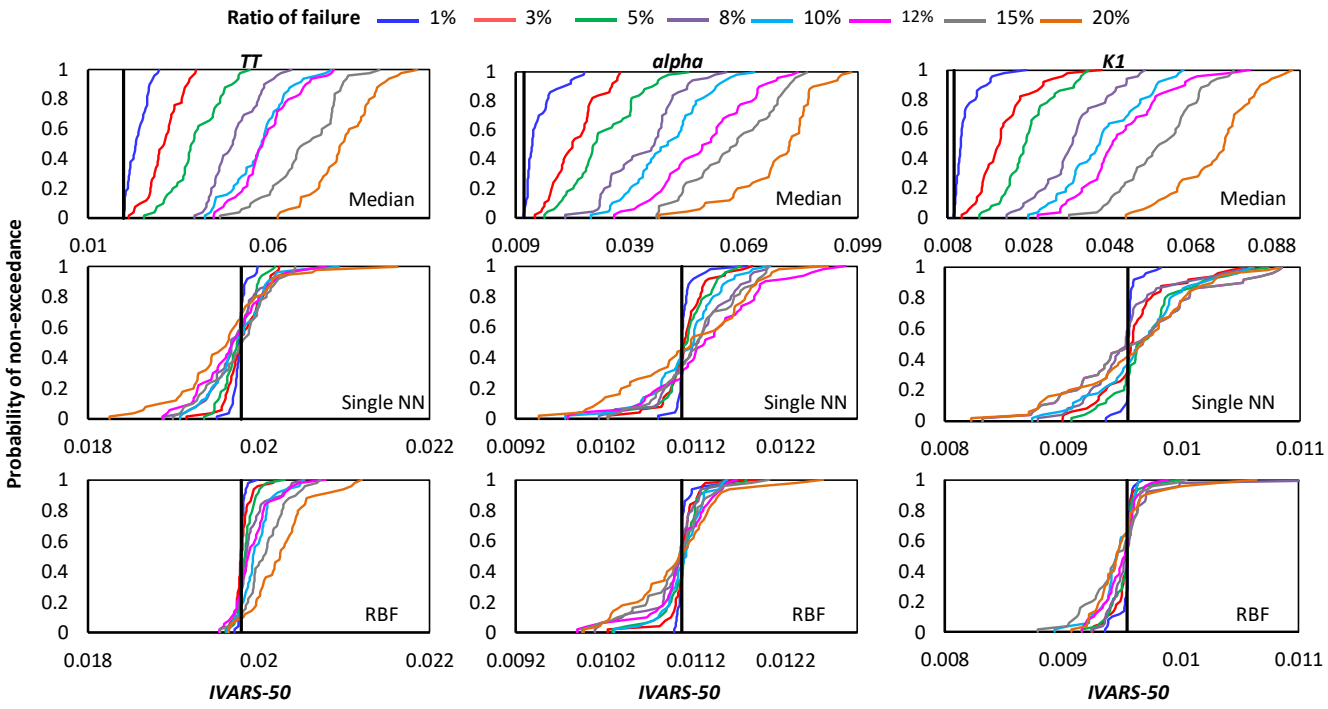

Figure 5: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the STAR-VARS algorithm for different ratios of failures. The CDFs of the sensitivity indices for moderately influential parameters {*TT*, *alpha*, *K1*} are compared in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

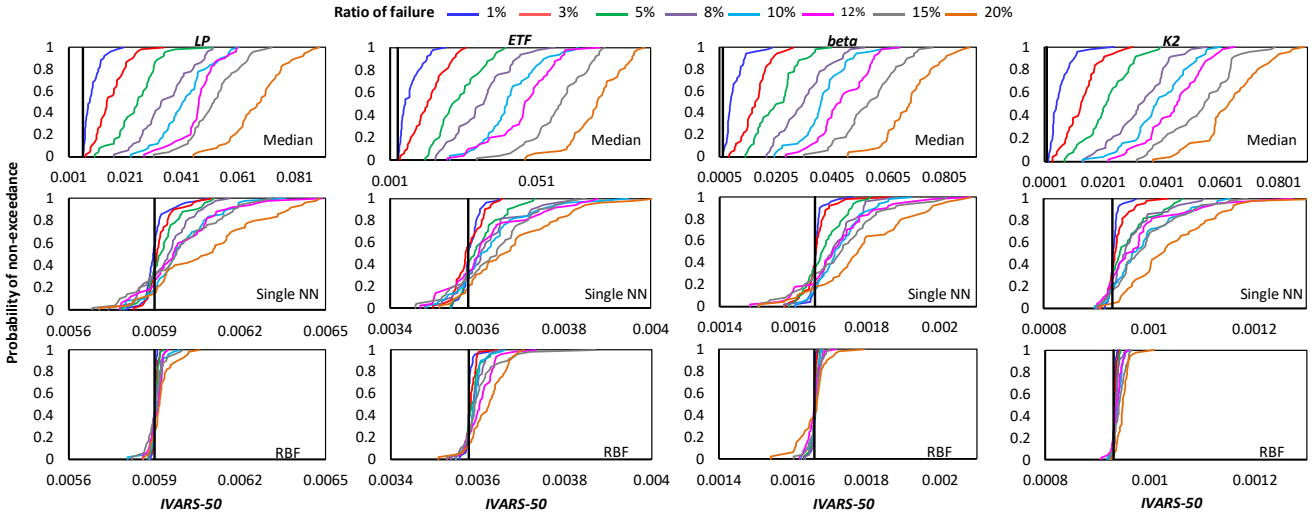

Figure 6: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the STAR-VARS algorithm for different ratios of failures. The CDFs of the sensitivity indices for weakly influential parameters (*LP, ETF, beta, K2*) are shown in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

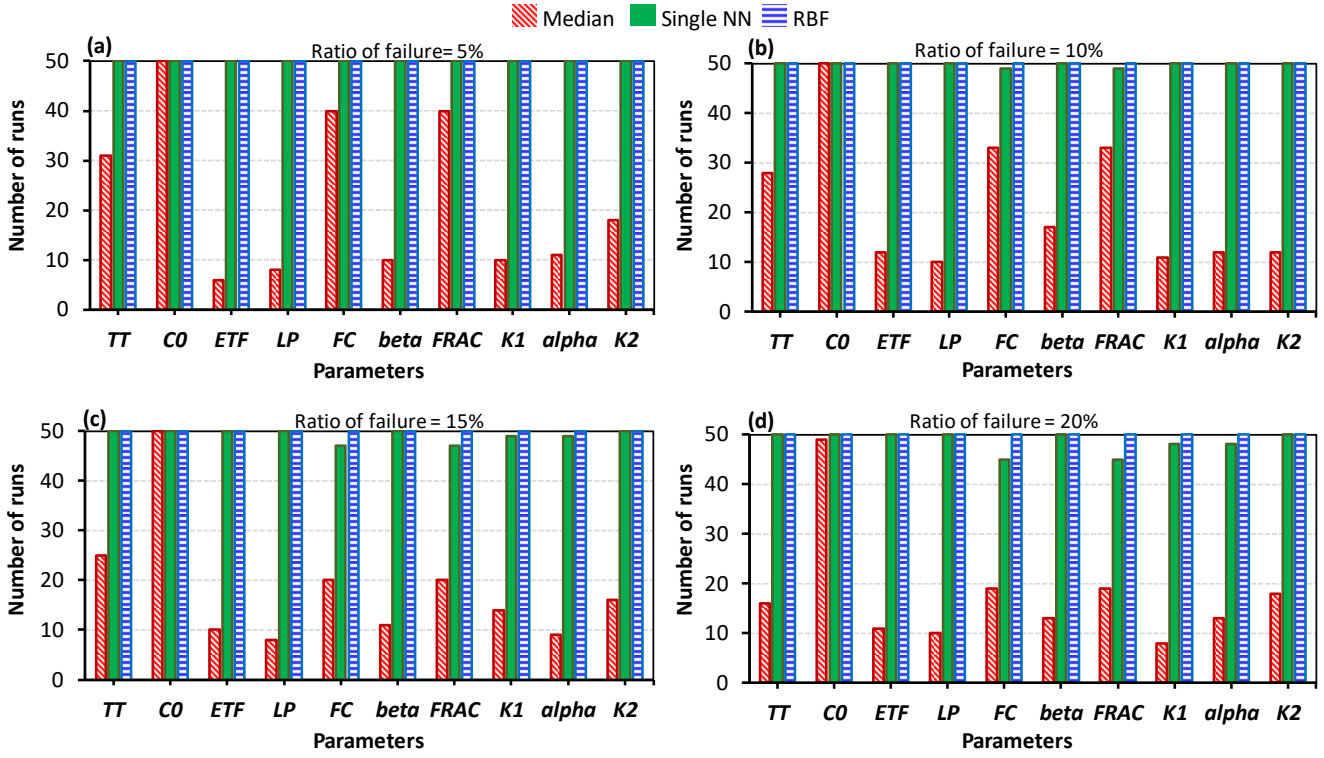

Figure 7: Comparison of the crash handling strategies in estimating the parameter rankings for HBV-SASK model using the STAR-VARS algorithm when the ratio of failure was (a) 5%, (b) 10%, (c) 15%, and (d) 20%. The *y*-axis in each subplot shows the number of times out of 50 replicates that the rankings of the parameters were equal to the true ranking.

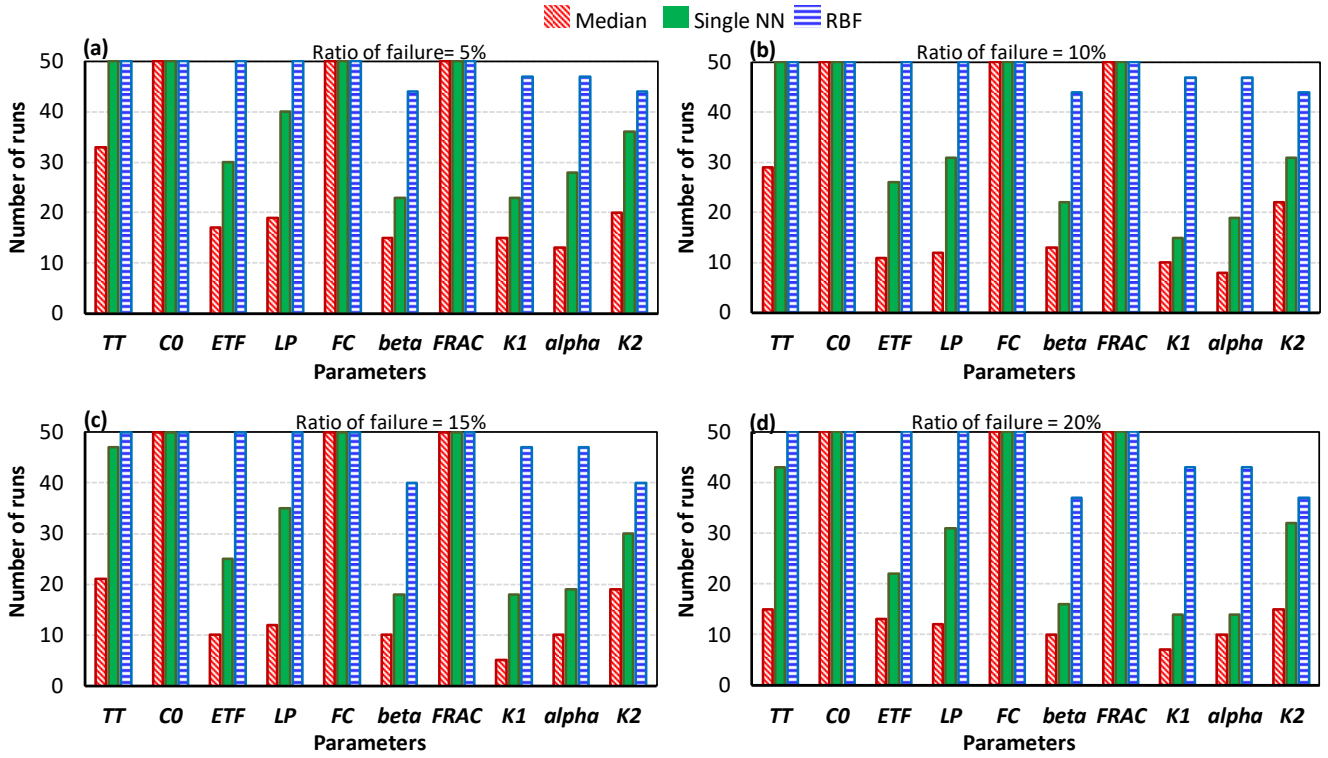

Figure 8: Comparison of the crash handling strategies in estimating the parameter rankings for HBV-SASK model using the variance-based algorithm when the ratio of failure was (a) 5%, (b) 10%, (c) 15%, and (d) 20%. The *y*-axis in each subplot shows the number of times out of 50 replicates that the rankings of the parameters were equal to the true ranking.

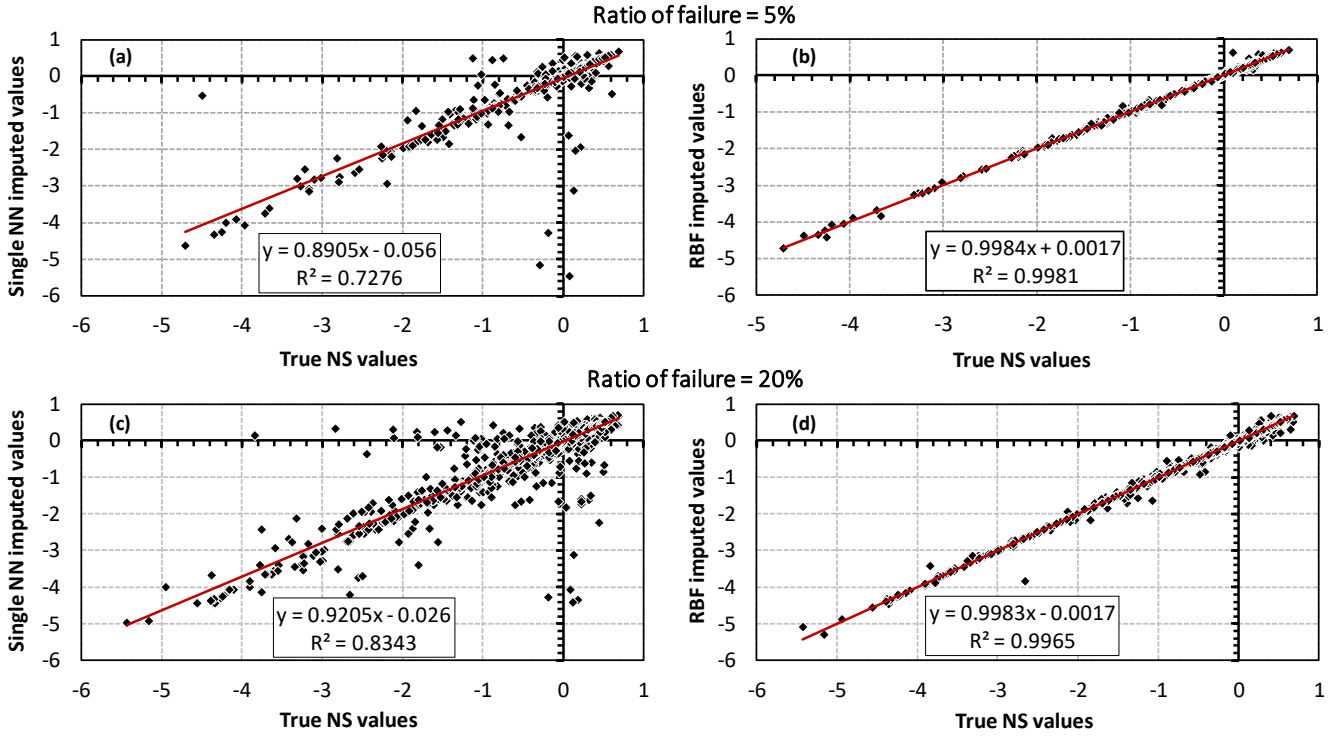

Figure 9: Scatter plots of the true NS values versus the imputed NS values when the ratio of failure was 5% (top panel) and 20% (bottom panel) for the HBV-SASK model. The accuracy of the crash handling strategies is demonstrated in subplots ((a) a) and (c) for the single NN method and in subplots ((b)b) and (d) for the RBF method. These results belong to one arbitrarily chosen replicate out of 50 independent runs.

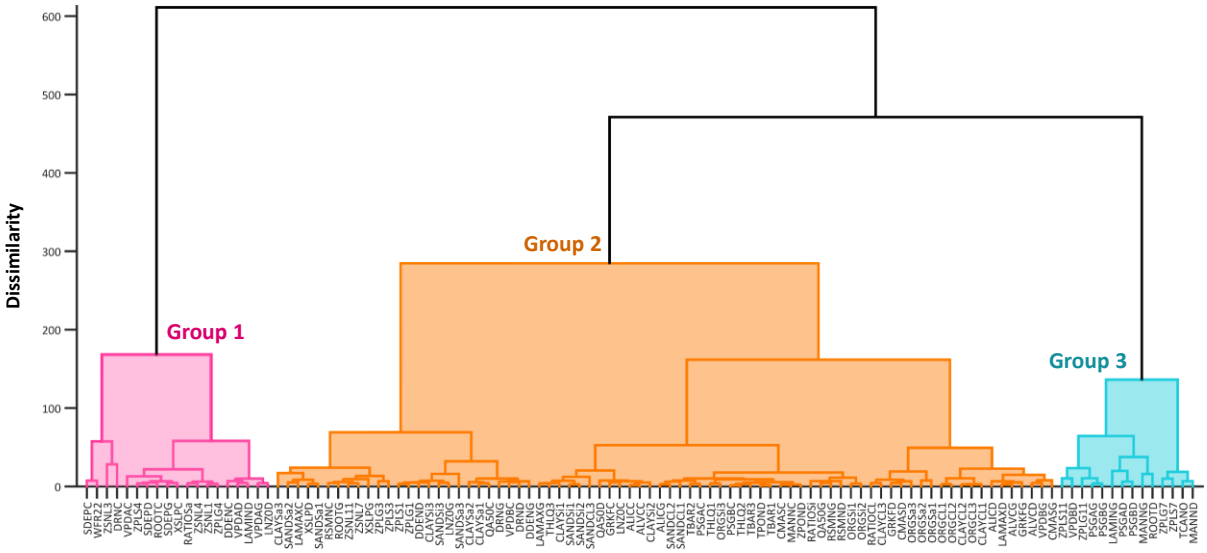

Figure 10: Grouping of the 111 parameters of the MESH model. The parameters are sorted from the most influential (to the left) to the least influential (to the right). This grouping is based on the results of the RBF method.

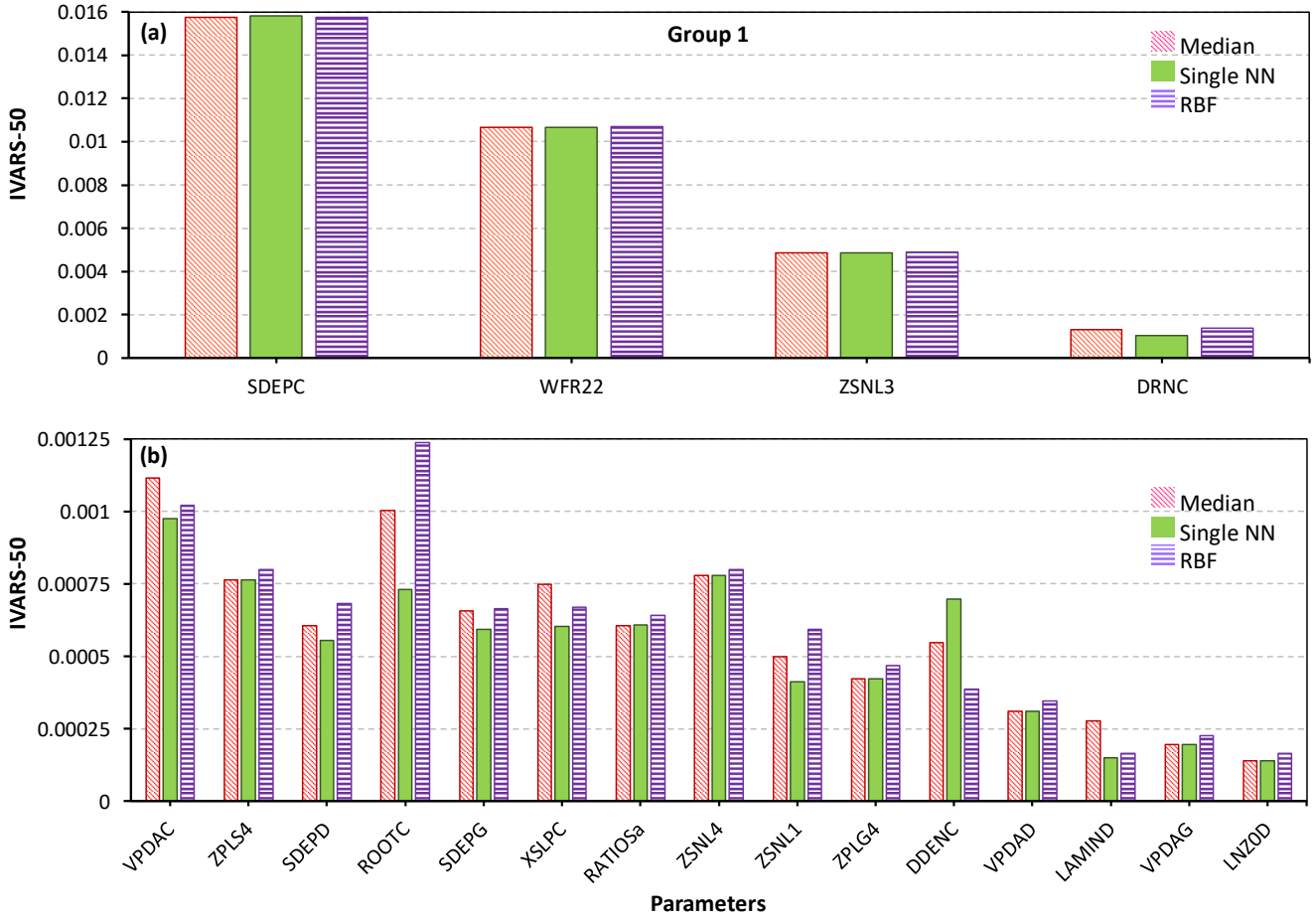

Figure 11: Sensitivity analysis results of the MESH model using different crash handling strategies for the most influential parameters. To better illustrate the results, the highly influential parameters in Group 1 (see Figure 10) are separately shown in two subplots (a) and (b).

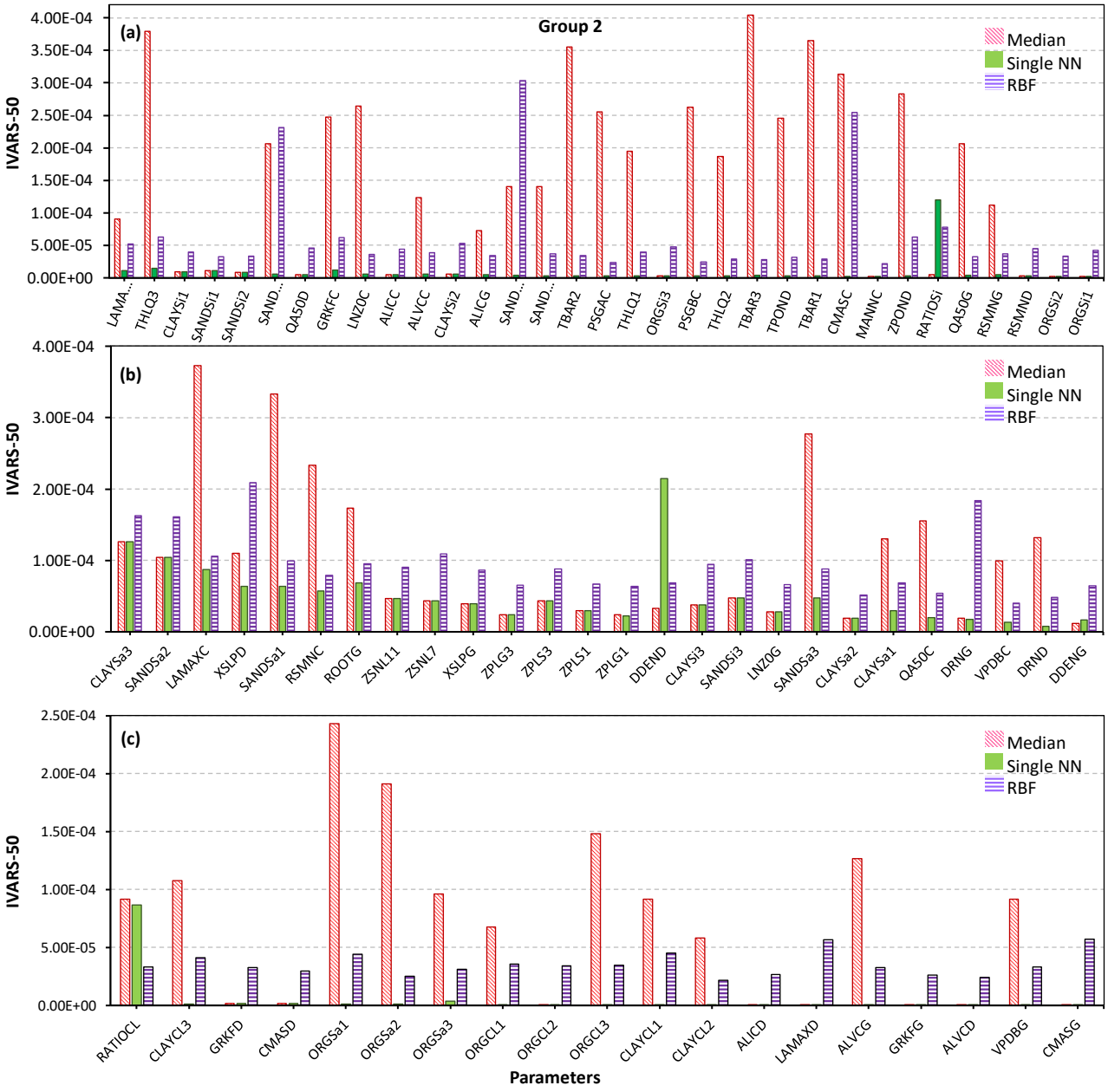

Figure 12: Sensitivity analysis results of the MESH model for moderately influential parameters using different crash handling strategies. To better illustrate the results, the moderately influential parameters in Group 2 (see Figure 10) are separately shown in three subplots (a), (b), and (c).

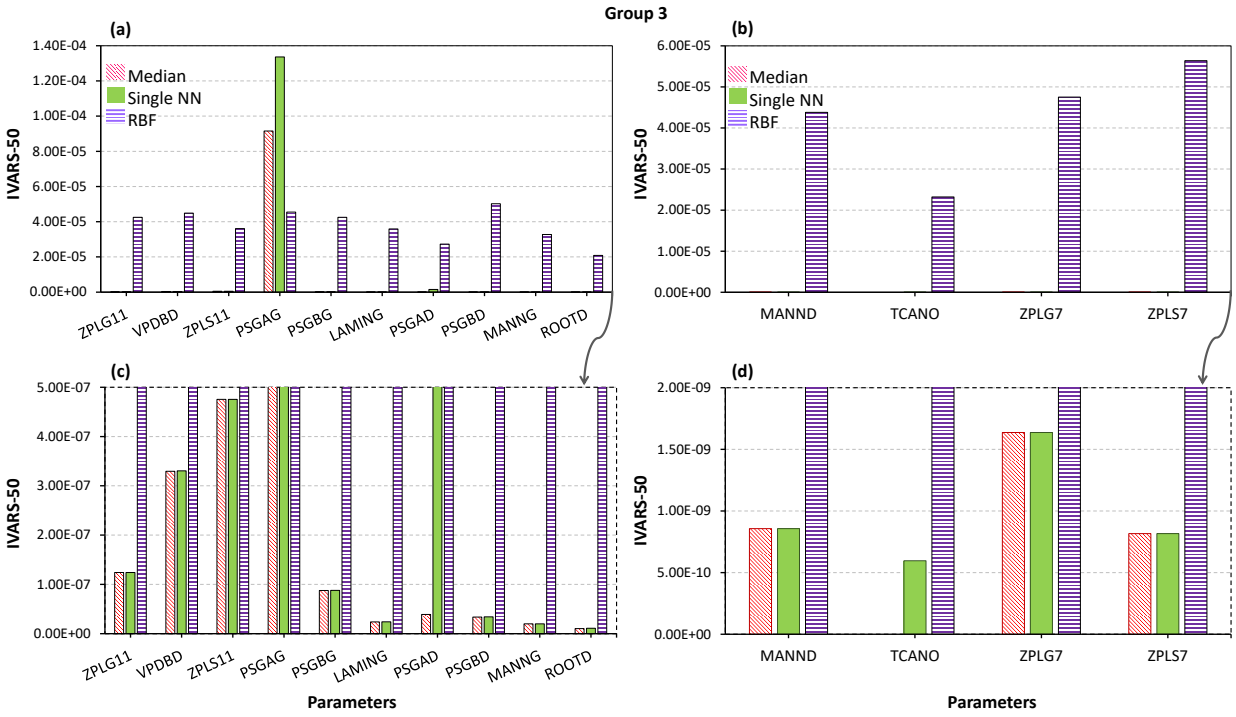

Figure 13: Sensitivity analysis results of the MESH model using different crash handling strategies for weakly/non- influential parameters in Group 3 (see Figure 10). The bottom panel (c and d) shows a zoom-in of the top subplots for very small values on the vertical axis.

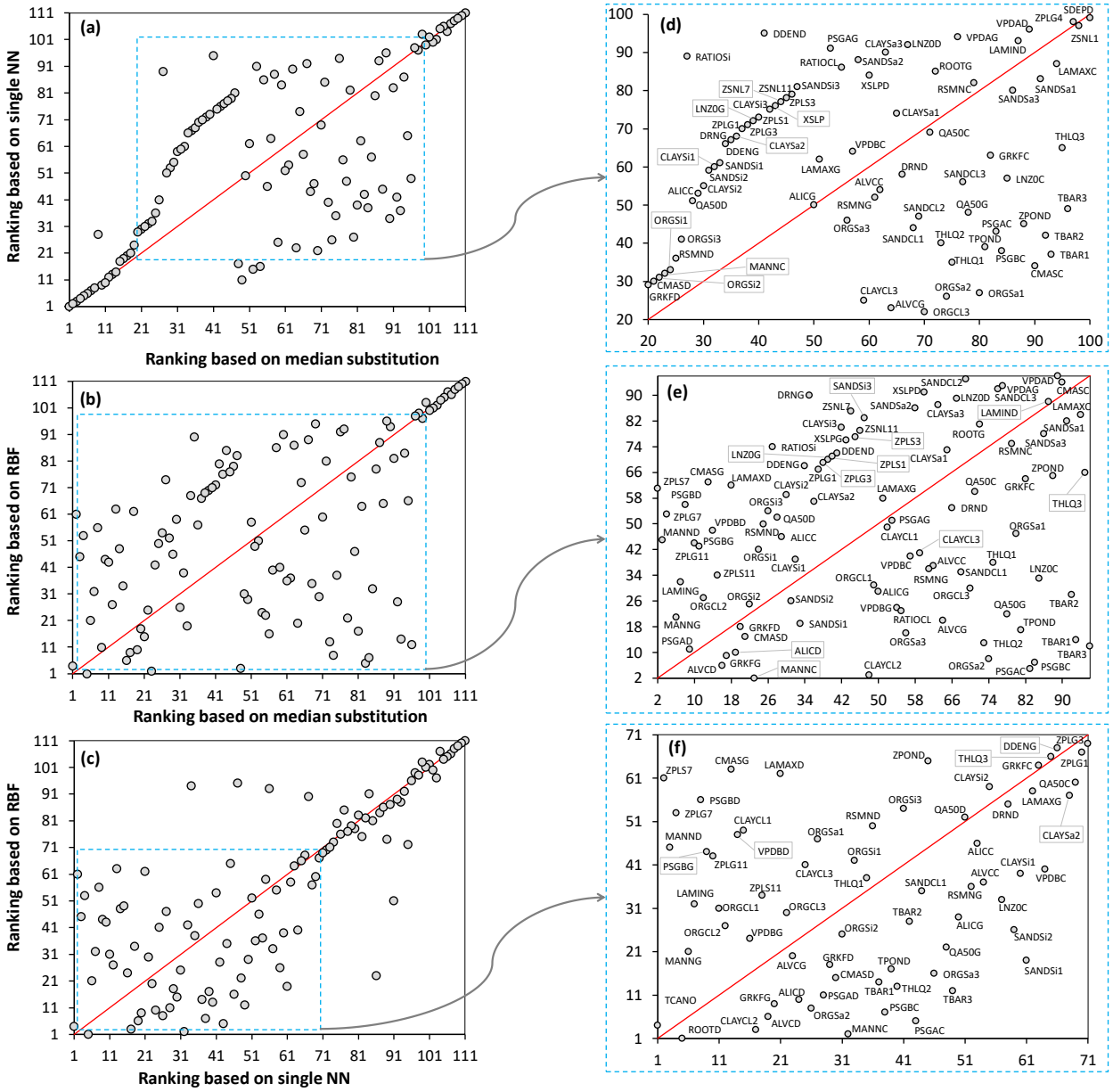

Figure: 14. Comparing rankings of the MESH model parameters obtained by different crash handling strategies using the STAR-VARS algorithm. Subplots (d), (e), and (f) (right column) show a zoom-in of the subplots (a), (b), and (c) (left column), respectively. The red line is the ideal (1:1) line. Note that a ranking of 1 represents the least influential and a ranking of 111 represents the most influential parameter.

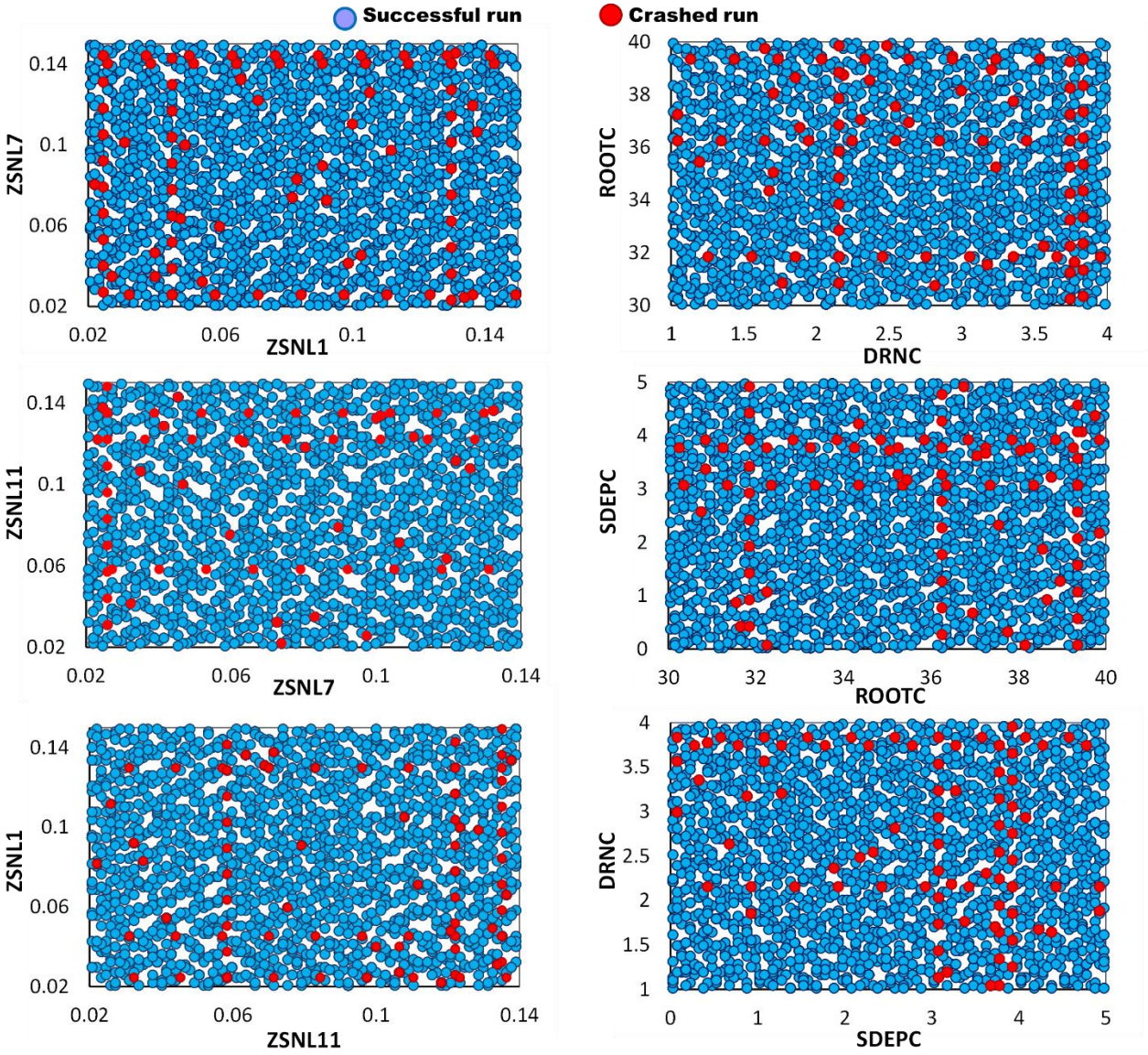

Figure 15: A 2-D projections of the MESH parameters for successful (blue dots) and crashed (red dots) simulations. Left column shows the threshold snow depth parameters *ZSNL* and right columns shows soil permeable depth (*SDEP*), maximum rooting depth (*ROOT*), and drainage index (*DRN*) for crop vegetation type (C).

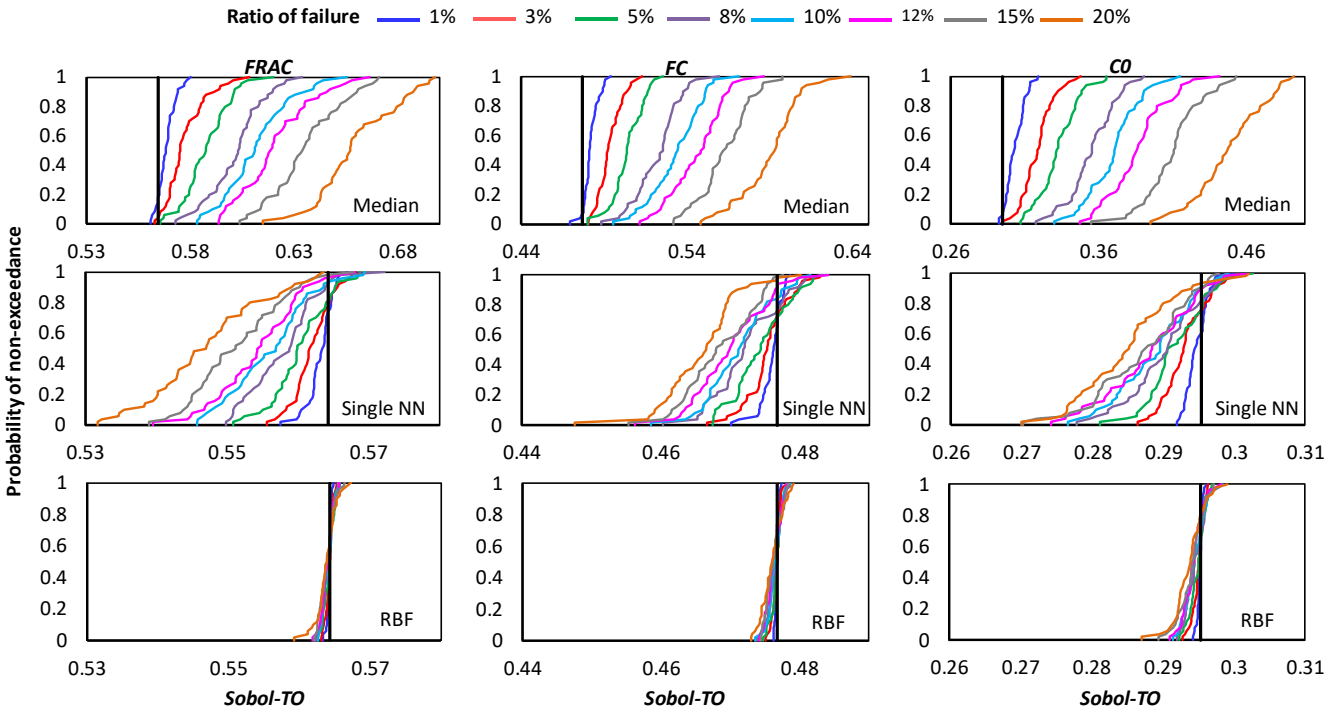

Figure B1: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the variance-based algorithm for different ratios of failures. The CDFs of the sensitivity indices for strongly influential parameters {*FRAC*, *FC*, *C0*} are compared in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

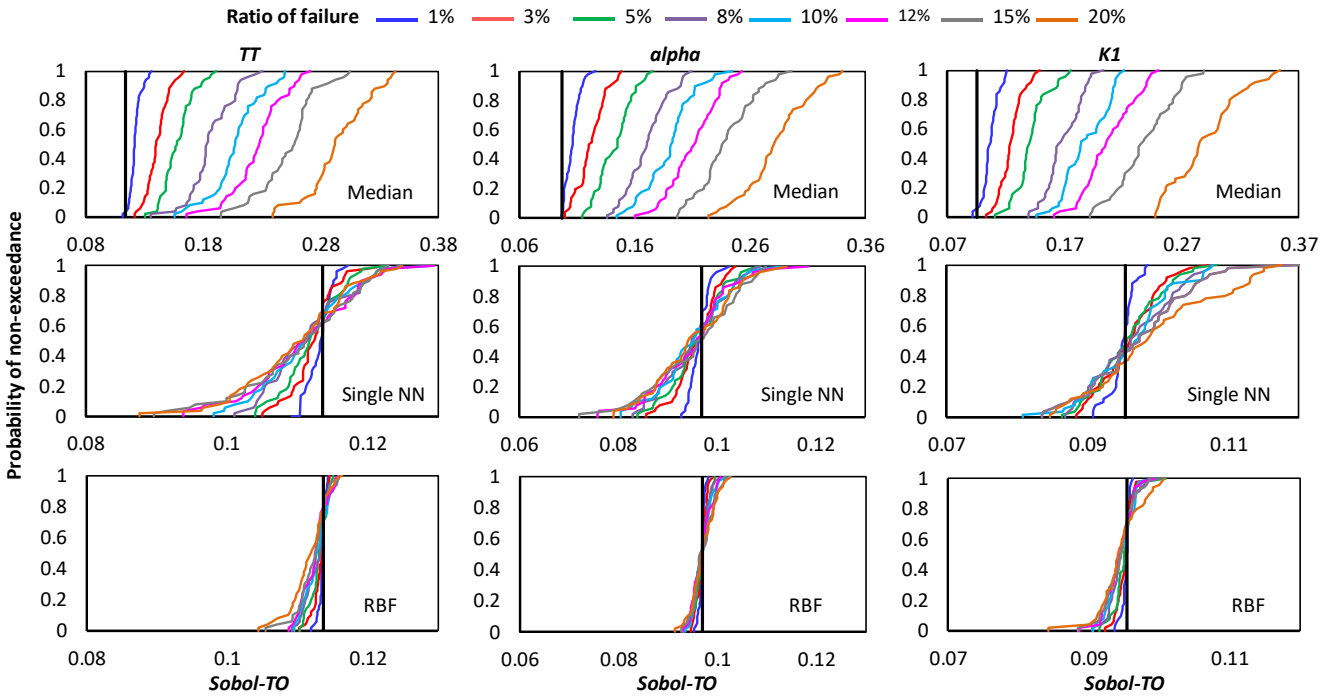

Figure B2: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the variance-based algorithm for different ratios of failures. The CDFs of the sensitivity indices for (b) moderately influential parameters {*TT*, *alpha*, *K1*} are compared in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

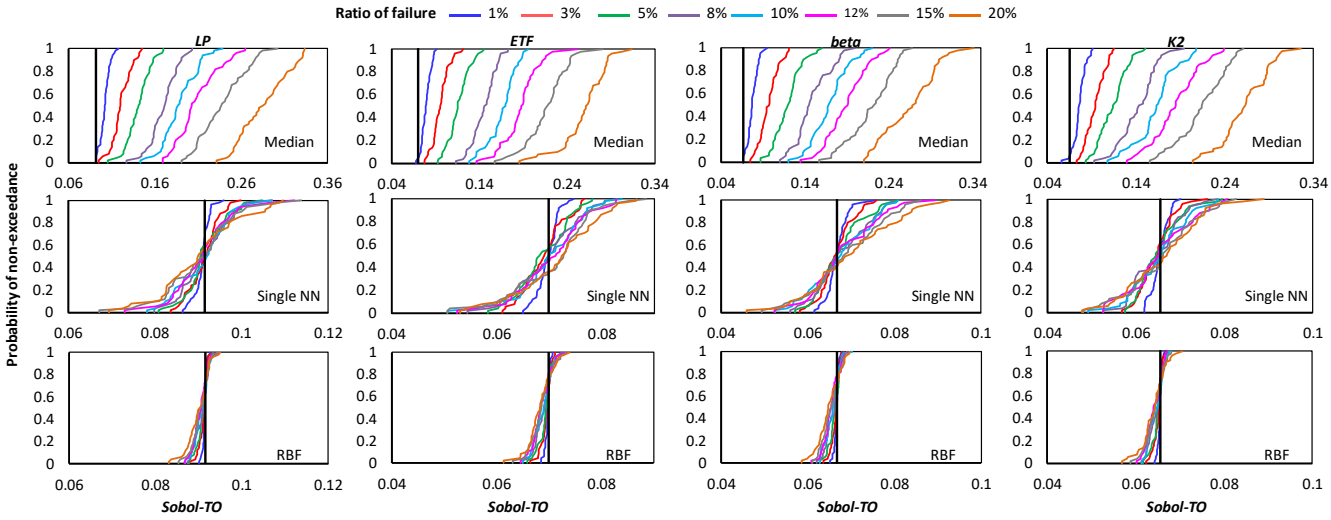

Figure B3: Comparison of the proposed crash handling strategies in sensitivity analysis of the HBV-SASK model using the variance-based algorithm for different ratios of failures. The CDFs of the sensitivity indices for weakly influential parameters (*LP*, *ETF*, *beta*, *K2*) are shown in this plot. The vertical line (solid black) on each subplot represents the corresponding ''true'' sensitivity index obtained when there were no failures.

Table 1. HBV-SASK model parameters and their feasible ranges, used in this study. For information on the full parameter set, refer to Razavi et al. (2019).

Parameter	Range	Description
TT	[-4,4]	Air temperature threshold in °C for melting/freezing and separating rain and snow
C0	[0,10]	Base melt factor, in mm/°C per day
ETF	[0,1]	Temperature anomaly correction in 1/°C of potential evapotranspiration
LP	[0,1]	Limit for PET as a multiplier to FC, i.e., soil moisture below which evaporation becomes supply limited
FC	[50,500]	Field capacity of soil, in mm. The maximum amount of water that the soil can retain
beta	[1,3]	Shape parameter (exponent) for soil release equation (unitless)
FRAC	[0.1,0.9]	Fraction of soil release entering fast reservoir (unitless)
K1	[0.05,1]	Fast reservoir coefficient, which determines what proportion of the storage is released per day (unitless)
alpha	[1,3]	Shape parameter (exponent) for fast reservoir equation (unitless)
K2	[0,0.05]	Slow reservoir coefficient which determines what proportion of the storage is released per day (unitless)

Table A1. Grouping of 111 MESH model parameters. These groups are numbered in order of importance.

Group number	Parameters
1	*SDEPC, WFR22, ZSNL3, DRNC, VPDAC, ZPLS4, SDEPD, ROOTC, SDEPG, XSLPC, RATIOs, ZSNL4, ZSNL1, ZPLG4, DDENC, VPDAD, LAMIND, VPDAG, LNZ0D*
2	*CLAYSa3, SANDSa2, LAMAXC, XSLPD, SANDSa1, RSMNC, ROOTG, ZSNL11, ZSNL7, XSLPG, ZPLG3, ZPLS3, ZPLS1, ZPLG1, DDEND, CLAYSi3, SANDSi3, LNZ0G, SANDSa3, CLAYSa2, CLAYSa1, QA50C, DRNG, VPDBC, DRND, DDENG, LAMAXG, THLQ3, CLAYSi1, SANDSi2, SANDCL3, QA50D, GRKFC, LNZ0C, ALICC, ALVCC, CLAYSi2, ALICG, SANDCL2, SANDCL1, TBAR2, PSGAC, THLQ1, ORGSi3, ORGSi1, PSGBC, THLQ2, TBAR3, TPOND, TBAR1, CMASC, MANNC, ZPOND, RATIOSi, QA50G, RSMNG, RSMND, ORGSi2, RATIOCL, CLAYCL3, GRKFD, CMASD, ORGSa3, ORGSa2, ORGSa1, ORGCL1, ORGCL2, CLAYCL2, ORGCL3, CLAYCL1, ALICD, LAMAXD, ALVCG, GRKFG, ALVCD, VPDBG, CMASG*
3	*ZPLS11, VPDBD, ZPLG11, PSGAG, PSGBG, LAMING, PSGAD, PSGBD, MANNG, ROOTD, ZPLG7, ZPLS7, TCANO, MANND*