Probabilistic Evaluation of Critical Scenarios with Adaptive Monte Carlo Simulations Using the Software Tool SUSA

Jan Soedingrekso[a], Tanja Eraerds[a], Martina Kloos[a], Jörg Peschke[a], and Josef Scheuer[a]
[a] GRS gGmbH, Garching, Germany, jan.soedingrekso@grs.de

Abstract: The uncertainties of an accident analysis can be addressed by performing Monte Carlo simulations within the so-called best-estimate plus uncertainty (BEPU) approach. By varying the uncertain input parameters and running the respective simulations of a deterministic code, tolerance intervals of the safety relevant simulation result can be calculated using, for instance, the software tool for uncertainty and sensitivity analysis, SUSA. However, the analysis of critical combinations of parameters resulting in rare undesired events requires a large number of simulations to accurately describe the underlying parameter regions and to quantify the probabilities of the undesired events. By incorporating adaptive sampling methods in the Monte Carlo simulation, these rare scenarios can be evaluated probabilistically with reasonable computational effort. Three adaptive sampling methods have been implemented in SUSA to determine parameter regions leading to rare critical events and to estimate the corresponding probabilities. The first approach applies a support vector regression metamodel in the frame of a subset simulation. The second approach combines a genetic adaptive sampling algorithm with an ensemble of classification algorithms, and the third approach uses an adaptive Gaussian process. This contribution presents two of the adaptive sampling approaches implemented in the GRS software tool SUSA and their application to a loss of coolant accident (LOCA) scenario.

1. Introduction

To analyze the influence of uncertain input parameters on accident events Monte Carlo (MC) simulation is usually used. However, in classical MC simulation, scenarios with an undesired event (e.g., damage or failure event) typically occur only in connection with parameter combinations that define only a small portion of the total parameter space and, consequently, do not often get sampled. Therefore, a large number of simulation runs are required to determine the parameter regions leading to an undesired event and to estimate the probability of occurrence of the parameter region, i.e., the probability of the undesired event. Due to the complexity of the modeled accidents in a system such as a nuclear power plant (NPP), these simulation runs are often very time consuming. Therefore, using classical MC simulation to evaluate a targeted parameter region with low probability of occurrence is computationally infeasible. Adaptive sampling approaches can be used to increase runtime performance in this context.

In the first part of this paper, an overview of the workflow of the adaptive sampling methods implemented in the software tool SUSA (Software for Uncertainty and Sensitivity Analyses) [1] is given and one of the implemented procedures is described in more detail. In the second part, the results of an application of two of the adaptive sampling procedures in SUSA are described. The exemplary event sequence studied is a loss of coolant accident (LOCA), where a peak cladding temperature (PCT) above 1200 °C defines the undesired scenario.

2. Adaptive Sampling in SUSA

Adaptive sampling algorithms are an approach to efficiently increase the relative number of samples in interesting parts of the admissible parameter space. Their goal is to reduce the number of simulation runs required while providing a high level of accuracy for estimating the probabilities of undesired scenarios. It should be mentioned that the generated simulation dataset can only be used for analyses

interested in the specific region of the parameter space leading to a particular undesired event. The interest of other analyses may be in other regions of the parameter space leading to other events and would require additional simulation runs. Therefore, adaptive sampling approaches do not always reduce the number of simulation runs required, especially when multiple analyses, all interested in different parameter regions, can use a common MC data set. However, for a single analysis or multiple analyses interested in the same region of parameter space, adaptive sampling of parameters can reduce the number of simulation runs by several orders of magnitude.

The requirements for the implemented adaptive sampling approaches are the same as for classical sampling: a set of uncertain input parameters together with their probability distributions, and a deterministic simulation program calculating the system behavior depending on the input parameters. In addition, an algorithm is required to evaluate the usefulness of input parameter combinations for the analysis so that useful combinations are sampled more frequently. The classification of an undesired event may relate to a single output quantity of the simulation, e.g., the PCT with the critical threshold PCT > 1200 °C. In general, however, it can also refer to multiple quantities, which can for instance be calculated by applying an algorithm that computes, for example, a score that maps the multivariate dependence to a one-dimensional variable. For simplicity and to better understand the concept, only a single simulation output variable is considered in this paper to decide if an undesired event occurs or not. Furthermore, it is assumed that the undesired event is defined by a region either at the lower or upper boundary of the range of the considered output quantity and that the corresponding values can be ranked according to their distance from the targeted undesired event. In the following, when referring to a selection of samples near the target range, this also includes samples that are already within the target region.

The general idea of the iterative approach of adaptive sampling can be described in the following steps.

1. The initial step is to create a training dataset by randomly sampling the uncertain input parameters according to their probability distributions and to run the simulations with these samples. Due to the long duration of the simulation runs, only a small set of samples, e.g., 20 – 50, should be created for efficiency reasons. Since this initial step is not an integral part of an adaptive sampling algorithm, the initial training dataset can alternatively be taken from a previous uncertainty analysis if the uncertain parameters are the same.
2. The training dataset created is used to train a single or multiple metamodels, i.e., machine learning algorithms, to predict the simulation result for the considered output quantity.
3. A large set of input parameter values is randomly sampled according to the probability distributions, e.g., $10^4 – 10^5$ samples, depending on the applied adaptive sampling algorithm, but instead of running the simulations, the trained metamodel(s) are applied to this sample to predict the simulation results.
4. The predictions of the metamodel(s) are used to identify candidates of parameter combinations that are best suited to be added to the training dataset to improve the predictions of the metamodel(s), especially in the vicinity of the targeted parameter region. For these candidates the results are calculated by the actual simulation code; therefore, only a few samples should be selected, e.g., 5 – 8 candidates.
5. Depending on predefined termination criteria, the algorithm either terminates and provides the estimated probability of the targeted region, i.e., probability of the undesired scenario or the algorithm is repeated and returns to step 2, now with the enhanced training dataset.

This general concept may vary slightly for a particular adaptive sampling algorithm, but the general approach remains the same. In SUSA, three adaptive sampling methods have been implemented to increase the sampling efficiency for accident analyses that aim to derive an estimate of the probability for rare undesired scenarios They are described in detail in [2]. For a better understanding of the adaptive sampling procedure, the GASA-PRECLAS algorithm is discussed here. The other two approaches, one based on Gauss processes, with a metamodel consisting of Gaussian Kernels, and the other based on a combination of subset sampling and support vector regression (SuSSVR), have already been discussed in [3], [4] respectively.

2.1. The GASA-PRECLAS-Algorithm

The GASA-PRECLAS algorithm (Genetic Adaptive Sampling Algorithm – Probability Estimation using an Ensemble of Classification Algorithms) combines a genetic algorithm and an ensemble of classification algorithms with a Bayesian approach. The algorithm was developed to achieve the following three goals:

- Generality: The algorithm should be applicable on different kind of complex problems, especially on multivariate non-linear functions.
- Usability: The metamodels should have a robust structure with only a few hyper-parameters, that need to be adjusted.
- Efficiency: The dataset to train the metamodels should be created with small effort, i.e., with as few calculations as possible.

In principle, these goals also apply to the other two adaptive sampling approaches since all algorithms aim to obtain a stable estimate of the probability of the undesired event. As the name implies, the GASA-PRECLAS algorithm is subdivided into two phases. First, the GASA algorithm is used to create a dataset with a minimum number of samples within the targeted parameter region leading to the undesired event. In a second phase, this dataset is optimized by the PRECLAS algorithm to produce a stable probability estimate for the undesired event. The reason for this separation is that the classifiers in the second phase require training data that contain at least some parameter samples that lead to an undesired event. Otherwise, the data would not allow any differentiation. After the metamodels contain at least some samples of both classes (inside and outside the targeted region), the second algorithm is applied, which has the main goal of precisely estimating the probability distribution for an undesired event. The workflow of both algorithms is illustrated in Figure 1 and described in detail below.

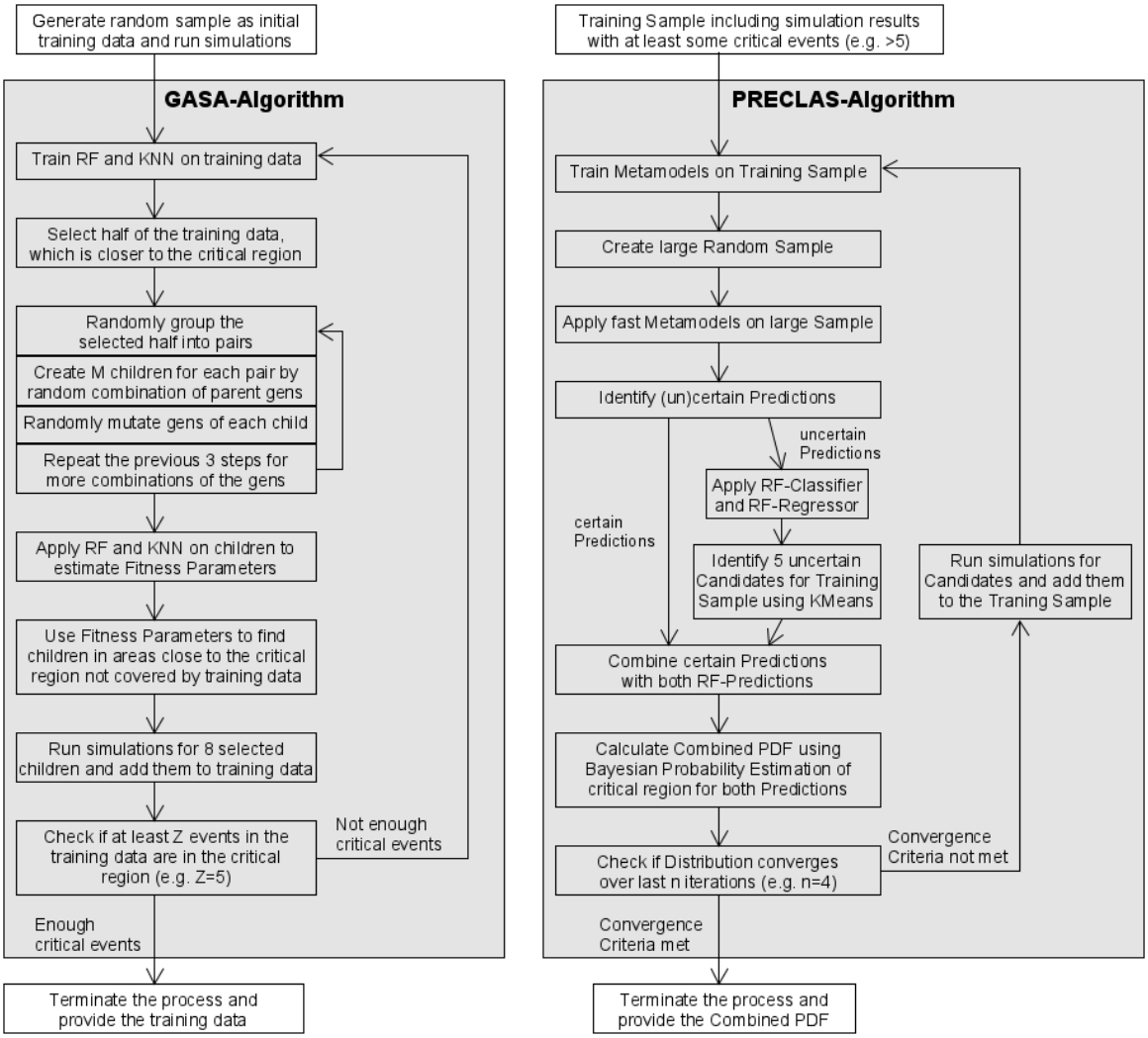
2.2. The GASA Algorithm

The Genetic Adaptive Sampling Algorithm (GASA) integrates a genetic algorithm into an adaptive sampling approach and aims to generate a training dataset biased towards the targeted region with a limited number of simulation runs. The general idea of the genetic algorithm is to successively create new generations of individuals (observations) by selecting promising candidates as parents, recombining the parents´' characteristics or genes for the new generation, and mutating the children's genes. An overview over genetic algorithms can be found in [5].

In the GASA algorithm, a gene is equivalent to a sampled parameter value. A chromosome describes a combination of genes, i.e., a combination of values of all uncertainty parameters. For both, the GASA and the PRECLAS algorithm, not the parameter values themselves are stored in the genes or sampled parameter values, but rather the corresponding values of the cumulative distribution functions of their probability distributions. This has the advantage for machine learning algorithms that all values are normalized between 0 and 1. The conversion of the cumulative distribution function values to the actual parameter values is calculated each time before the simulation run. A chromosome or sampled combination of parameter values is then used either to run a single simulation or for the predictions of the metamodels. The one-dimensional simulation result can be represented by the function $G(x)$ where x represents the parameter vector used as input for the simulation.

The GASA algorithm starts with a pre-generated sample. This is either a stored dataset, e.g., from a previous uncertainty analysis, or an MC sample of small size, e.g., 20 samples, based on the probability distributions of the uncertain parameters. For each vector of this initial sample x, the corresponding function $G(x)$ is calculated by running the simulation code. The training dataset D, consisting of the initial sample of vectors x and their corresponding function values $G(x)$, is used to create a new generation of children and train metamodels to predict the function values $G(x)$ for a new large sample (population) of parameter vectors. The metamodels used in the GASA algorithm are a Random Forest (RF) Regressor, and a K-Nearest Neighbors (KNN) Classifier. The generation of the children is independent of the training of the metamodels, which can in principle be computed in parallel.

Figure 1: Flowchart of the Adaptive Sampling Procedure of the Two Parts of the GASA-PRECLAS Algorithm (RF = Random Forest algorithm, KNN = K-Nearest Neighbors algorithm, PDF = Probability Density Function).



First, an even number of parents is selected from the training dataset D that is allowed to create new children. Only half of the samples in the dataset D whose function values $G(x)$ are closer to the threshold defining the undesired event, are selected as parents. The pool of parents is randomly grouped into pairs. Each pair creates M children, where for each child, the genes are randomly selected by one of the two parent genes. Each gene of a child x_i is mutated by a value sampled from a uniform distribution between $x_{min} = x_i - x_i(1 - x_i)$ and $x_{max} = x_i + x_i(1 - x_i)$. To increase the variety of the genes for the generated children, the parents are again randomly combined to pairs for generating further children. The number of children produced by each pair as well as the number of repetitions for combining pairs and producing children can be adjusted. For the examples presented, 25 children are generated by each pair and the random combination of parents is repeated once.

The trained metamodels, i.e., the RF and KNN algorithms, are then applied to the children to predict their function values $G_{RF}(x)$ and $G_{KNN}(x)$. In addition, the absolute difference between the two predictions $\delta y = |G_{RF}(x) - G_{KNN}(x)|$ is calculated as well as the distance between the genes of each child and its nearest neighbor using the Manhattan metric $\delta q = \Sigma|x_i - x_{i,KNN}|/N$. The predictions $G_{RF}(x)$ and $G_{KNN}(x)$ as well as δy and δq are considered as fitness parameters, which are used to select the children to be most beneficial to add to the dataset D.

The selection procedure of the children is divided into four steps, each selecting two candidates.

- The first two candidates are identified by pre-selecting those five children closest to the targeted parameter region according to the prediction of the Random Forest $G_{RF}(x)$. Out of these 5 children, the child with the highest value of δy and the child with the highest value of δq are selected. If the second child is the same as the first one, the child with the second highest value of δq is chosen.
- The same procedure is done for the next two candidates, except that the pre-selection now considers the sum of the prediction of the Random Forest and the KNN $G_{RF}(x) + G_{KNN}$. The selection of two candidates regarding the highest values of δy and δq or those next to the highest values, if the child has already been selected, is the same as in the first step.
- In the third step, the pre-selection considers the ten children with the highest values of δy. Out of these children, the two children for which the predicted function values are closest to the threshold defining the undesired event according to $G_{RF}(x)$ and $G_{RF}(x) + G_{KNN}$ and which have not yet been selected as candidates are added to the candidates.
- The last selection step is similar to the third one, except that the pre-selection considers the highest values according to δq instead of δy. The selection procedure of the two candidates is the same as in the third step.

This procedure always identifies eight children as the most promising candidates to be added to the training dataset D. The idea behind this is that, on the one hand, the candidates should be close to or inside the targeted parameter region and, on the other hand, to increase the variability of the training sample in order to identify further regions which are not yet covered by the training data. In case of a small training data, many children share the same nearest neighbor and would get the same value of the KNN prediction. Therefore, a combination of the Random Forest and the KNN is used here.

For the eight candidates identified, the corresponding function values are calculated by running the simulations and added to the training dataset D. As long as the number of samples inside the targeted parameter region is below a number specified by the user, e.g., at least five samples, the algorithm continues to the step, where the metamodels are trained and the parents are selected, now using the enhanced training dataset D. If a sufficient number of samples are inside the targeted parameter region, the GASA algorithm terminates and the PRECLAS algorithm continues using the training dataset D.

2.3. The PRECLAS Algorithm

The **Pr**obability Estimation using an **E**nsemble of **Clas**sification Algorithms (PRECLAS) builds on a pre-generated sample, e.g., provided by the GASA algorithm, which already includes a sufficient number of samples in the targeted region, e.g., five samples. In contrary to the GASA algorithm, the aim is not the number of samples in the targeted region. The PRECLAS algorithm aims at a stable and converging estimation of the probability distribution for an undesired event.

In the first step of the PRECLAS algorithm, the given training dataset D is used to train several machine learning algorithms creating metamodels for distinguishing between critical (undesired) and uncritical events. Multiple machine learning algorithms are used to compensate uncertainties of a single metamodel and identify certain and uncertain predictions. The following metamodels or machine learning algorithms are trained:

- Decision Tree Classifier,
- Decision Tree Regressor,
- Gaussian Naive Bayes Classifier,
- Non-Parametric Naive Bayes Classifier,
- K-Nearest Neighbors Classifier,
- Random Forest Classifier,
- Random Forest Regressor.

Except of the non-parametric naive Bayes algorithm, all used machine learning algorithms can be found in the python library scikit-learn [6] with documentation of their functionality. Since the Gaussian Naive Bayes assumes a normal distribution of the input parameters, an additional non-parametric Naive Bayesian approach was developed using an underlying histogram distribution created empirically out of the training data (cf. [2]).

While the classifiers are trained to distinguish between critical and uncritical events, the regressors are trained to predict the value of an output quantity of the actual simulation code. The predicted values of the regressors are then used to classify an event being critical or uncritical to get further independent predictions that are used to increase the certainty on the classification.

After the training of the machine learning algorithms, a random sample of large size is created, e.g., consisting of 100,000 samples. The size of the population should be large enough to cover a large fraction of the parameter space, while not being too large that the runtime of the prediction of the metamodels would last too long.

Since the Random Forest algorithms are ensemble learners, they take significantly more time to predict events, especially for a large population. For that reason, only all fast-running metamodels, i.e., all learners except the Random Forest algorithms, are applied to the created large sample in a first step. The following rules are applied:

- Disagreement: If the classifications of the fast algorithms provide the same results, it is assumed that the predictions of the RF algorithms would provide the same results as the fast algorithms.
- Similarity: If the mean Manhattan-distance of a parameter vector of the population to the nearest parameter vector of the training data is below a threshold (e.g., 5 E-03), the parameter vector of the large sample is assumed to be the same as the nearest parameter vector of the training data.
- Uncertainty: If the average of the prediction probability whether the sample belongs to the targeted region is between 0.4 and 0.6 and therefore close to a simple guess, the classification is assumed as uncertain. The average of the prediction probability is calculated by the mean of the probabilities of two Naive Bayes algorithms.

In a second evaluation step, the RF-algorithms are applied to those parameter vectors of the population for which the fast metamodels have been assessed as uncertain or too similar to existing event sin the training dataset. The selection of new candidates for the training data D starts with a pre-selection according to the following criteria:

- Disagreement: The classifications differ between the RF Classifier, the RF Regressor, and the KNN.
- Uncertainty: The average of the prediction probability of the two Naive Bayes and the RF Classifier is between 0.4 and 0.6.

Out of the parameter vectors of the population which meet one of the two criteria (disagreement or uncertainty) new candidates for the training dataset are selected. A cluster analysis is applied which separates the data into five different clusters. The K-Means algorithm is used, which is also implemented in the scikit-learn library [6]. The number of clusters is adjustable and should met the goal of high entropy between the clusters and low entropy inside the clusters. For each generated cluster, that candidate is selected, for which the predicted function values approximate best to the threshold defining the undesired scenario Therefore, the number of clusters defines the number of events added to the training sample in each iteration of the PRECLAS algorithm.

For the probability estimation, the certain predictions of the fast metamodels are combined with the two predictions of the Random Forests for uncertain samples. A Bayesian approach is applied for both predictions to estimate the respective probability of the undesired event resulting in two Beta distributions. To combine the information of the two Beta distributions, the mean distribution is calculated describing the uncertainty about the probability estimation of the targeted region.

To retrieve a stable and converging probability distribution and further reduce the uncertainty introduced by the machine learning models the following criteria are used to decide, if the PRECLAS algorithm should terminate or if the algorithm should reiterate. At least four iterations are required to terminate the PRECLAS algorithm. Using the results of the last four iterations, the mean values of the probability distribution are calculated. The PRECLAS terminates, if

- the standard deviation of the last four means divided by the mean of the last four means is below 0.25, or if
- the difference between the maximum and the minimum of the last four means, relative to the minimum, is below 0.5.

Both thresholds can be adjusted, but the idea behind these criteria is to confirm a certain convergence of the probability distribution for the undesired event. However, decreasing these thresholds would increase the number of required simulation runs. The number of four loops used to calculate the criteria are also a trade-off between convergence and calculation effort.

In case of a reiteration, the function values are calculated running the actual simulation code for the selected candidates of the cluster analysis, which are added to the training dataset D. Then, the next iteration of the PRECLAS algorithm starts with the training of the metamodels using the updated training dataset.

3. Application to a Loss of Coolant Accident

The objective of the exemplary analysis for adaptive sampling is to localize the region of the input parameter space that leads to a peak cladding temperature (PCT) > 1200 °C during a loss of cooling accident (LOCA) and to determine its probability.

The simplified reference model of a NPP that is used in this work, is a pressurized water reactor (PWR) of 1425 MW_e. The accident is assumed to be initiated by a double ended guillotine break in the cold leg of the main coolant pipe. For the simulation of the accident, the simulation code ATHLET, version 3.2 [7] was used. In the underlying data set, the reference NPP was characterized by a 4-loop model. The reactor pressure vessel was modelled with seven thermal hydraulic core channels (five normally loaded core channels, one highly loaded core channel and one hot channel with the hot rod). For the neutron kinetics a point kinetics model was selected. The data used correspond to a core at the beginning of the cycle. The initial boron concentration is 1150 ppm. The opening of the leak in the cold leg (Loop 10) was initiated at t = 600 s. The performance profile of a conservative deterministic analysis was considered, and the highest rod length power was about 460 W/cm (16 x 16 fuel elements).

35 input parameters identified in a previous analysis [8] and used for the ATHLET simulation were considered as uncertain. To determine the range in which the PCT value varies as a function of randomly selected values for the input parameters, a classical uncertainty analysis was first performed using SUSA. A subsequent sensitivity analysis was used to determine the parameters with the largest influence on the PCT variation.

3.1. Classical Uncertainty and Sensitivity Analysis

For the uncertainty analysis, a total of 60 different samples for the uncertain parameters were randomly selected and corresponding simulation runs were performed with ATHLET. The interesting result from these simulation runs is the temporal PCT curve in Figure 2 and the corresponding maximum value from this curve. The empirical distribution function for the maximum value of the PCT is shown in Figure 3. An upper (95 %, 95 %) tolerance limit (according to Wilks' theorem [9]) of 1203 °C is obtained, which indicates that at least 95 % of the PCT maximum values are below 1203 °C with a statistical confidence of at least 95 %. Main reason for the relatively high PCT values is the conservative performance profile considered for the application.

Figure 2: Peak Cladding Temperature (PCT) Development over Time of 60 simulation runs.

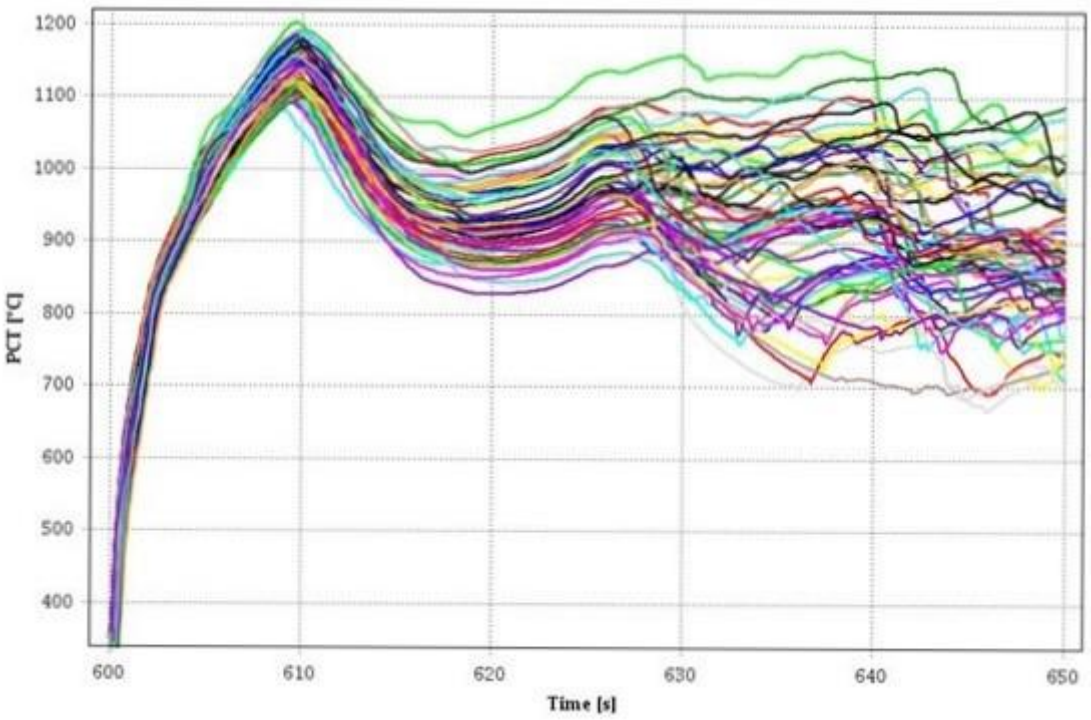
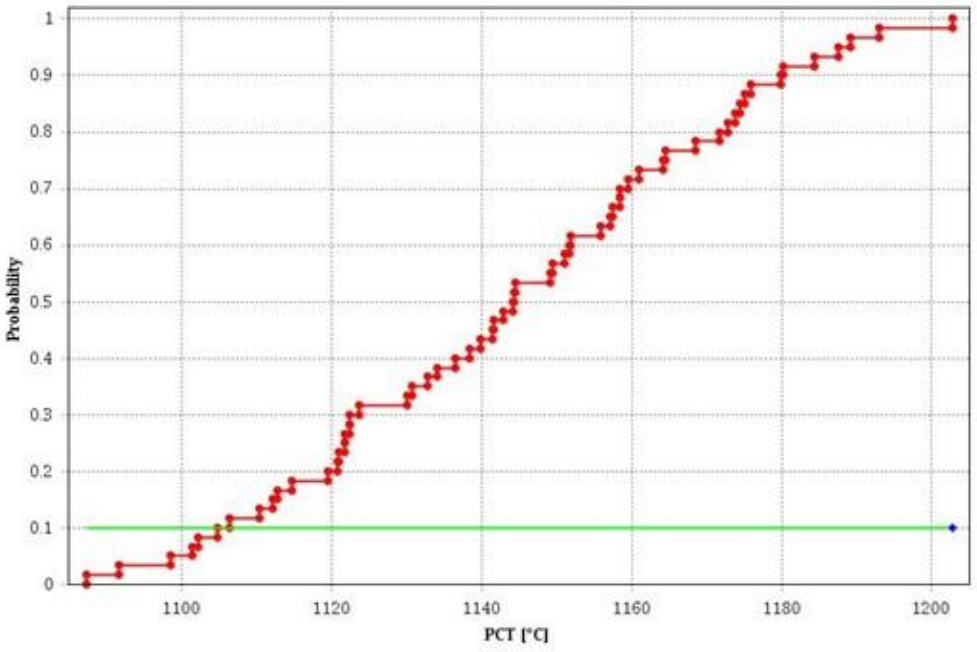


Figure 3: Empirical Cumulative Probability Distribution of the PCT with the upper (95 %, 95 %) tolerance limit using Wilks' theorem (blue diamond) for 60 simulation runs varied using classical Monte-Carlo sampling



The calculated value of 1203 °C for the upper (95 %, 95 %) tolerance limit suggests that the proportion of PCT values below the 1200 °C limit is lower than 95 %. Therefore, upper tolerance limits with smaller coverage probability were calculated. The calculation of the upper (92 %, 95 %) tolerance limit resulted in a value of 1193 °C. Accordingly, it can be assumed that at least 92 % of the possible PCT

values are below 1200 °C and thus at most 8 % above the limit value. Based on the result that one of in total 60 PCT maximum values exceeds 1200 °C, the 95 % confidence interval according to Pearson and Clopper [10] is given as [4.22 E-04 – 8.94 E-02] for the probability that the 1200 °C limit will be exceeded. A more precise estimate of this probability can be determined with practical computational effort using adaptive sampling methods. In the following paragraphs, first the application of the GASA-PRECLAS algorithm is outlined, focusing on the development of the sampling iterations. After that, the application of the SuSSVR method is described, focusing on the simulation results.

3.2. GASA-PRECLAS Application

The initial training dataset consisted of 50 random samples from the previous uncertainty analysis, where only one observation was inside the targeted region (PCT > 1200 °C). Therefore, the GASA algorithm was performed to increase the samples in the targeted region to at least five samples. After two iterations, the GASA algorithm terminated with a total number of five samples inside and 60 samples outside the targeted region. One simulation run was abnormally ended and yielded no results, which was excluded from the probability calculation. Using the generated training dataset of the GASA algorithm, the PRECLAS algorithm was applied, which terminated after nine iterations, meaning that the probability estimation meets the convergence criteria considering the previous iterations. In total, 103 ATHLET calculations were required for this estimate in addition to the 50 ATHLET calculations for the initial training dataset. The results of the iterations of the PRECLAS algorithm are shown in Table 1.

Table 1: PRECLAS Algorithm Results for the Training Dataset and Probability Distribution Parameters for the Targeted Region with the standard deviation σ and the difference between the maximum and the minimum δ of the last four means

Iteration	Number of Simulations with PCT		Mean	5 %	50 %	95 %	σ	δ
	> 1200	≤ 1200						
0	5	60	4.35 E-03	2.50 E-04	3.65 E-03	9.11 E-03		
1	6	69	4.26 E-03	4.30 E-04	3.51 E-03	8.91 E-03		
2	6	79	6.20 E-03	5.00 E-03	6.17 E-03	7.53 E-03		
3	8	87	5.95 E-03	2.38 E-03	5.53 E-03	1.01 E-02		
4	10	95	9.34 E-03	3.79 E-03	8.89 E-03	1.57 E-02	0.317	1.19
5	13	101	1.12 E-02	4.43 E-03	1.07 E-02	1.87 E-02	0.266	0.88
6	16	108	1.24 E-02	9.20 E-03	1.23 E-02	1.59 E-02	0.270	1.08
7	18	116	1.51 E-02	1.22 E-02	1.50 E-02	1.82 E-02	0.193	0.62
8	20	124	1.83 E-02	1.48 E-02	1.82 E-02	2.20 E-02	0.203	0.63
9	22	131	1.54 E-02	1.06 E-02	1.54 E-02	2.07 E-02	0.137	0.48

The final result of the PRECLAS algorithm provides a mean probability of $\widehat{P}_{PRECLAS}$ = 1.54 E-02 for the PCT exceeding 1200 °C with respect to the defined scenario. The 5 % and 95 % quantiles of the distribution are 1.06 E-02 and 2.07 E-02, respectively.

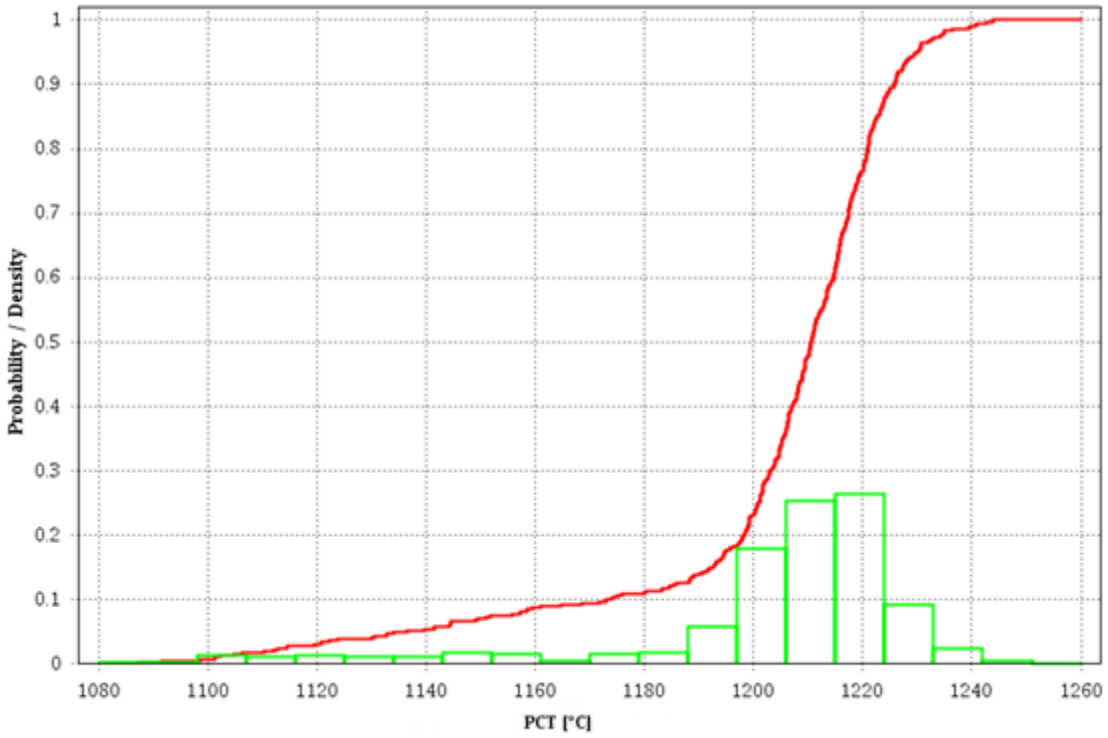
As stated above, one case occurred in the initial training dataset of 50 samples where the PCT threshold of 1200 °C was exceeded. Using a Bayesian approach with a non-informative prior distribution, a mean probability of $\widehat{P}_{MCS}$ = 2.94 E-02 for the PCT exceeding 1200 °C is calculated from this random sample. The 5 % and 95 % quantiles of the estimate are 3.53 E-03 and 7.55 E-02, respectively.

In order to be able assessing the quality of the estimation by the GASA-PRECLAS algorithm, it should first be noted that not only the mean probability but also the 5 % and 95 % quantiles of the PRECLAS estimation are in the 90 % range of the estimation obtained via the Bayesian estimation approach based on the initial training dataset, which indicates the validity of the estimation by the GASA-PRECLAS algorithm.

3.3. Subset-Sampling Support-Vector-Regression Application

In the SuSSVR approach, a machine learning algorithm is applied within a subset sampling. Here, a support vector regression is used [3], but other machine learning algorithms could have also been applied (cf. [11], [12]). The entire SuSSVR approach was automatically conducted resulting in a total of 414 simulation runs, including the initial 50 runs, before the termination criterion of the procedure was reached, i.e., until the trained SVR metamodel as well as probability estimator of the undesired event (PCT > 1200 °C) were judged to be robust enough. The calculated PCT values for all 414 ATHLET runs ranged from 1087.21 °C to 1243.95 °C. PCT values exceeding 1200 °C were calculated for 318 runs. The distribution of PCT values for the 414 runs is provided in Figure 4.

Figure 4: Empirical PCT Cumulative Probability Distribution (red) and Probability Density Function (green) for 414 Simulation Runs (50 runs using classical Monte-Carlo sampling and 364 varied using the SuSSVR approach)



The final robust SVR metamodel resulting from the SuSSVR procedure, was used to determine the occurrence probability for the targeted parameter region, i.e., the probability of the undesired event. The probability $P(D^{35})$ obtained from the subset sampling with the robust SVR metamodel, ranges from 8.16 E-03 to 9.36 E-03, and the associated variation coefficient is 0.022.

A probability estimate of this magnitude (i.e., of ~ 0.01) is obtained, e.g., if 100 runs are performed as part of a simple MC simulation and one of these runs has a PCT value higher than 1200 °C. In this case, however, the estimator is subject to a large error, e.g., the associated variation coefficient is 0.995. To obtain the very low variation coefficient of 0.022, a total of 204,546 runs would have to be performed. Nevertheless, the number of runs with 414 is relatively high for an adaptive sampling approach regarding a targeted region that occurs with a probability of ~ 0.01. The reason for the relatively high number of 414 computational runs is the high number of 35 uncertain parameters to be considered for fitting the SVR model. The more parameters (influencing factors) to be included, the more observations (parameter vectors and associated PCT values) are required until a robust regression model is built. This suggests the step to perform a sensitivity analysis before applying the SuSSVR procedure and to consider only the most important parameters in the adaptive sampling approach. For the application

case presented in [3] with six uncertain parameters, the SuSSVR approach was highly efficient. It required only 265 simulation runs to predict a probability ranging between 3.5 E-06 and 4.0 E-06.

It should be noted that, compared to the classical MC sampling, the SuSSVR method also provides an extensive sample of parameter combinations from the targeted parameter region, which allows to analyze this region more precisely and to specify essential characteristics. However, this is out of the scope of this publication.

4. Conclusions and Outlook

In this paper, the adaptive sampling methods implemented for the software tool SUSA have been presented. One of the methods, an approach combining a genetic algorithm with an ensemble of classification algorithms, was described in more detail. The results of the successful application of two of the methods to a LOCA scenario is presented. In the exemplary application case, a total of 35 uncertain parameters were considered. Therefore, a relatively high number of simulation runs was needed to provide a robust probability estimate of the considered undesired event defined by a peak cladding temperature exceeding the acceptance limit of 1200 °C.

Generally, the implemented adaptive sampling methods significantly reduce the number of simulation runs required for estimating probabilities of rare undesired events as demonstrated in the application examples considering a lower dimensional parameter space. In the future, all three adaptive sampling procedures in SUSA (including a Gaussian process approach) need to be compared and validated on various application examples of different complexity.

Acknowledgements

This activity has been carried out within the research and development project RS1599 initially granted by the German Federal Ministry for Economics and Energy (BMWi), now the German Federal Ministry of Economic Affairs and Climate Action (BMWK), is currently funded by the German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

References

[1] M. Kloos, et al. "*Main features of the tool SUSA 4.0 for uncertainty and sensitivity analyses*", Proceedings of the 25th European Safety and Reliability Conference (ESREL 2015), Zurich, Switzerland (2015).
[2] M. Kloos, et al. "*Weiterentwicklung des Analysewerkzeugs SUSA*", GRS-634, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Cologne, Germany, (2021).
[3] M. Kloos, et al. "*Adaptive Monte Carlo simulation for detecting critical regions in accident analyses*", Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference (ESREL 2020 – PSAM15), Venice, Italy, (2020).
[4] N. Berner, et al. "*Localizing cliff-edge effects in accident analyses via an adaptive Gauss Process sampling approach*", Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference (ESREL 2020 – PSAM15), Venice, Italy, (2020).
[5] D. Goldberg. "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley Longman Publishing, Boston, USA, (1989).
[6] F. Pedregosa, et al. "*Scikit-learn: Machine Learning in Python*", Journal of Machine Learning Research 12, pp. 2825-2830, (2011).
[7] A. Wielenberg, et al. "*Recent improvements in the system code package AC2 2019 for the safety analysis of nuclear reactors*", Nuclear Engineering and Design 354, 110211, (2019).
[8] W. Pointner, et al. "*Statistische LOCA-Analysen*", GRS-519, Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH, Cologne, Germany, (2018).

[9] S. Wilks "*Determination of sample sizes for setting tolerance limits*", Annals of Mathematical Statistics 12, pp. 91-96, (1941).
[10] C. Clopper and E. Pearson "*The use of confidence or fiducial limits illustrated in the case of the binomial*", Biometrika 26, pp. 404-413, (1934).
[11] X. Huang et al. "*Assessing small failure probabilities by AK–SS: An active learning method combining Kriging and Subset Simulation*", Structural Safety 59, pp. 86-95, (2016).
[12] N. Pedroni, et al. "*An Adaptive Metamodel-Based Subset Importance Sampling approach for the assessment of the functional failure probability of a thermal-hydraulic passive system*", Applied Mathematical Modelling 48, pp. 269-288, (2017).