From Explainable AI to Explainable Simulation: Using Machine Learning and XAI to understand System Robustness

Niclas Feldkamp
Technische Universität Ilmenau
niclas.feldkamp@tu-ilmenau.de

Steffen Strassburger
Technische Universität Ilmenau
steffen.strassburger@tu-ilmenau.de

ABSTRACT

Evaluating robustness is an important goal in simulation-based analysis. Robustness is achieved when the controllable factors of a system are adjusted in such a way that any possible variance in uncontrollable factors (noise) has minimal impact on the variance of the desired output. The optimization of system robustness using simulation is a dedicated and well-established research direction. However, once a simulation model is available, there is a lot of potential to learn more about the inherent relationships in the system, especially regarding its robustness. Data farming offers the possibility to explore large design spaces using smart experiment design, high performance computing, automated analysis, and interactive visualization. Sophisticated machine learning methods excel at recognizing and modelling the relation between large amounts of simulation input and output data. However, investigating and analyzing this modelled relationship can be very difficult, since most modern machine learning methods like neural networks or random forests are opaque black boxes. Explainable Artificial Intelligence (XAI) can help to peak into this black box, helping us to explore and learn about relations between simulation input and output. In this paper, we introduce a concept for using Data Farming, machine learning and XAI to investigate and understand system robustness of a given simulation model.

CCS CONCEPTS

• **Computing methodologies** → Modeling and simulation; Machine learning; • ;

KEYWORDS

machine learning, deep learning, robustness optimization, simulation, explainable AI, XAI

ACM Reference Format:
Niclas Feldkamp and Steffen Strassburger. 2023. From Explainable AI to Explainable Simulation: Using Machine Learning and XAI to understand System Robustness. In *ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS '23), June 21–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3573900.3591114

SIGSIM-PADS '23, June 21–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0030-9/23/06.
https://doi.org/10.1145/3573900.3591114

1 INTRODUCTION

Modeling and simulation are well-established methods for the analysis of systems. Various objectives can be targeted when simulation experiments are conducted and subsequently analyzed. One of those goals is the evaluation of robustness. Robustness is particularly important for example for production and logistic systems, but the basic concept is applicable to any other system as well. Robustness is achieved when the controllable factors of a system are set such that any variance in the uncontrollable factors (noise) has minimal effect on the variance of the desired output [39]. For calculating the robustness of a system, the quality-loss-formulas originating from the commonly known Taguchi method can be used [45]. Taguchi came from a background of quality engineering and management, and he found that it is more cost-efficient to reduce the variance in the process instead of optimizing for pure maximum performance [31]. Optimizing system robustness using simulation is a well-established research direction, for example using metamodeling [30, 35] or even artificial intelligence [5]. However, once a simulation model is available, there is great potential to learn more about the inherent relationships of the system, particularly with respect to its robustness [8]. Data farming offers the opportunity to explore large design spaces using smart experiment design, high-performance computing, automated analysis, and interactive visualization. This can be used to discover surprises in the system by uncovering aspects that may have been previously hidden or unknown. Learning more about the general behavior of the model can facilitate decision making [7, 21, 37]. With data farming approach, we can use the generated bulk of simulation data to train metamodels using sophisticated machine learning methods. Those in turn are ideally suited for mapping even the most complex relationships between input and output data [41]. However, using those machine learning models to actually explore and analyze this modeled relationship in order to learn more about the underlying system can be very difficult, as most modern machine learning methods such as neural networks or random forests are opaque black boxes [7]. Explainable AI (XAI) can help peek into this black box and explore and learn the relationships between simulation input and output [3]. In this paper, we present an approach for using data farming, machine learning, and XAI to explore, explain and understand the system robustness of a given simulation model. This includes for example identifying robust configurations, identifying critical combinations of noise factor values, and investigating which factor values and combinations contribute to robustness and which do not. The remainder of this paper is structured as follows: In Section 2, we give an overview of system robustness and robustness analysis, including a brief introduction into Taguchi's quality-loss-formulas. This is then followed by a brief introduction into data farming and the use of machine learning for the analysis of large quantities of

Table 1: Types of loss functions for different targets

Type of loss function	Formula
Nominal-the-best	$\eta = 10\ log_{10}(\frac{\mu^2}{\sigma^2})$
Smaller-the-better	$\eta = -10\ log_{10}(\frac{1}{n}\sum_{j=1}^{n} y_j^2)$
Larger-the-better	$\eta = -10\ log_{10}(\frac{1}{n}\sum_{j=1}^{n} \frac{1}{y_j^2})$

simulation data, as well as explainable artificial intelligence. We then present our concept for using data farming, machine learning and XAI for robustness analysis in Section 3, followed by an exemplary case study in Section 4. In Section 5 we give some concluding remarks and a discussion of possible future work.

2 RELATED WORK

2.1 System Robustness and Simulation-based Robustness Optimization

Robustness refers to setting the controllable factors of a system in such a way that the variance from uncontrollable noise has a minimal effect on a given output [39]. Variations due to noise can originate from a variety of sources. For example, fluctuations in customer demand can lead to fluctuations in the mix of orders that are being processed in the system, which in turn influences the system performance. This effect can dramatically increase in a supply chain, commonly known as the bullwhip effect [18].

A very popular method for measuring robustness is the Taguchi method. Genichi Taguchi [45] developed a methodology for evaluating decision alternatives not only on the basis of their outcome value, but also the variability around that outcome against the noise. Taguchi developed formulas to calculate the loss in quality that results from deviation from a desired value. Taguchi considers the loss as a quadratic function that has zero value exactly when the measured value of y is equal to the desired target outcome τ [31]:

$$Q(y) = k(y-\tau)^2$$

The parameter k is a constant corresponding to the application, which specifies the magnitude of the loss. Since the deviations from the target value are observed for different noise factor configurations in the course of robustness optimization, an expected value of the loss can be calculated for each configuration in the following form [36],

$$Q(y) = k\left[(\mu-\tau)^2 + \sigma^2\right]$$

where μ and $\sigma 2$ indicate the mean and variance of the measured output variable for a tested configuration of controllable factors. Thus, the loss takes into account not only the mean of the measured output variable but also its variance. However, using a loss function in this form the deviation of the mean value from the target value is often too dominant and the sensitivity to noise variables, expressed by the variance, is not considered enough. Therefore, this is taken into account by performing a logarithmic transformation into the so-called signal-to-noise ratio (S/N-ratio) [31]. This ratio can directly be used as a measure of robustness, also called the loss function. Besides a loss function that aims to minimize against a distinct target value τ (nominal-the-best), there is also a function available for minimizing the target value (τ =0, smaller-the-better) and the reciprocal thereof for maximizing the target value (larger-the-better) [31]. The goal of the robustness optimization is to minimize the loss by maximizing η, which is the value of the S/N-ratio over j different settings of noise and n different settings of controllable factors. In Table 1, we summarized the three types of loss functions and their respective formulas according to [31].

Taguchi's work on robustness analysis had a great impact and was very influential among statisticians [27]. A more in-depth review on the subject of Taguchi method and other robust design concepts can be found in Park et al. [29]. As already mentioned, both controllable and uncontrollable factors (noise) must be taken into account when calculating robustness. This is what makes robustness analysis so interesting from the perspective of simulation research. In principle, all factors can be controlled in a simulation model, but we can still make a distinction between factors that can be controlled in reality (also called decision factors) and those that cannot be controlled there [39]. Separate experiment plans need to be created for the control factors as well as for the noise factors. Each factor value combination of one respective experiment plan represents a configuration, i.e. a control factor configuration (also called system configuration) or noise factor configuration, respectively [39]. For the calculation of the robustness, we then need to simulate every noise factor configuration in combination with every system configuration [8]. This means that both plans need to be combined in a crossed experiment plan, so that a robustness measure for each row of the decision factor plan can be calculated against all noise factor combinations [8]. Obviously, the number of required experiments can get large very quickly because of the crossed experiment design. This is where aspects of data farming become very interesting, on the one hand because of its smart experiment design approaches, on the other hand for its capability of automated analysis of large quantities of simulation output data using data mining and machine learning, as discussed in the next section. In data farming based robustness analysis demonstrated in [8], we can specifically target the robustness in much broader way than traditional robustness analysis. That means that we explicitly include the whole range of factor value combinations, even those that could be considered as exotic, unrealistic or far from what is considered as a normal operation. That way we can explore when a system reaches a form of stable equilibrium, dependent on what we define as noise factors. We could even define the load of the system as noise to see how the system performs under a wide variety of

different loads and how we can make it robust against varying loads [8].

2.2 Data Farming and Knowledge Discovery in Simulation Data

In many simulation projects, a traditional simulation study usually attempts to achieve a specific, predefined objective, such as scenario-based analysis or even simulation-based optimization. This can still leave much room for actually understanding the behavior of the model in terms of global factor-to-ouput relationships [15, 28]. Sometimes discovering new and interesting relationships that were previously unknown and that fall outside the predefined scope of the simulation study design can improve decision-making. For this reason, the data farming approach has been developed [14, 15]. Data farming refers to the method of generating data from a simulation model using smart but still large-scale experimental designs, high-performance computing for massively parallelized experiments to create a nearly complete coverage of possible system responses, and an automated, machine-aided analysis [16, 38]. Data farming research has also always involved the application of advanced data analysis techniques in order to handle these large volumes of simulation outputs and derive relevant and adequate insights [21, 37]. As an extension, the concept of knowledge discovery in simulation data was developed to take a deep dive into the analysis part of data farming, providing a process model and workflow for the application of data mining and machine learning techniques, as well as suitable interactive visualizations [7]. This is particularly useful for models with a large number of relevant outcomes that have a complex, multidimensional response surface. It makes the analysis and interpretation of even large amounts of simulation data from complex models much more manageable, as has been demonstrated in various case studies, e.g. see [11, 17, 19, 44]. Simulation outputs can be aggregated into multidimensional groups representing different system behavior patterns by using pattern recognition methods such as clustering [6]. Then, the relationship between factors and outputs can be analyzed using supervised machine learning algorithms. Those algorithms can generate models that represent the relationship between simulation input and output data, from which in turn generalizable rules about the system can be extracted. In combination with human interpretation and reasoning, this can contribute to knowledge generation and facilitate decision making. When training a supervised algorithm, each simulation experiment acts as a learning record [7]. In other words, a classification problem needs to be solved. Extracting decision rules from the underlying classification model is only possible from white-box algorithms. White-box means that those algorithms provide their internal mapping of their x-y relation in an interpretable, human-readable form, like for example a decision tree classifier does. However, in robustness analysis, the relationship between control factors and the S/N-ratio can be notoriously complex and uneven, as shown exemplarily in Figure 1. Here we can see the relation of two control factors and the S/N-ratio of one simulation output from the even comparatively simple case study model that we use for demonstrating our concept later on in Section 4.

One can clearly see how complex the surface of robustness can get, even in a simple model. Those machine learning algorithms

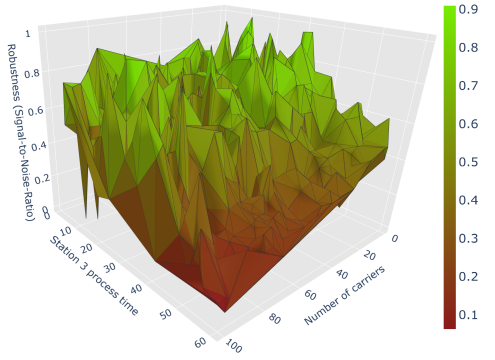

Figure 1: Complex surface of system robustness against two control factors

that are capable to model even the most complex, non-linear relationships are usually opaque black-boxes, for example ensemble methods like large random forests or artificial neural networks. There is a tradeoff between accuracy in terms of predictive power, and interpretability [26]. White-box, interpretable algorithms are simpler, easy to compute and easy to understand. On the other end of the spectrum, black-box algorithms are usually highly accurate and can model non-linear, non-smooth relationships, thereby requiring a lot of computation time and their internal decision-rules are difficult or impossible to comprehend [26]. In order to overcome the lack of interpretability for such black-box algorithms, XAI can help to make these rules and relationships visible and explain them in an understandable and comprehensible manner. Therefore, the application of XAI enables us to use even the most complex black-box algorithms for rule extraction and knowledge generation from our simulation data. A more detailed introduction into XAI is provided in the next section.

2.3 Explainable Artificial Intelligence

The transparency of decisions made by artificial intelligence and machine learning algorithms is becoming increasingly important, especially when people's daily lives are directly affected. A decision explained as "because the computer said so" is not acceptable and can even lead to legal problems if it is suspected of being unfair and discriminatory, for example in areas such as credit scoring [3, 13]. Some regulators, for example in the European Union, are even considering a "right to explanation" [12, 40]. However, it is not only consumers who are affected by the decisions made by machine learning. In fact, numerous stakeholders can benefit from the transparency of such algorithms, such as risk analysts, regulators, and developers [2]. On the other hand, black-box machine learning and artificial intelligence methods, such as artificial neural networks, are among the most powerful algorithms for regression and classification tasks. As already mentioned in the previous section, due to the complexity of these algorithms compared to their white-box counterparts, for example simple linear regression, there is a trade-off between performance and interpretability. For this reason, XAI has recently become a popular area of research with the goal of making black-box algorithms transparent. The term XAI

actually encompasses a very broad range of different methods, and so do efforts to catalog and categorize these methods [1, 32, 46].

The taxonomy of methods according to [43] is divided among four categories: Model-agnostic vs. model-specific methods, global vs. local explanation, the timing of application and the form of presentation. The first category is concerned with the flexibility of the XAI methods' applicability [1, 2]. Model-agnostic methods are usually independent from the type of the underlying algorithm, because their explanation is based solely on the relationship between input and output in consideration of the model's prediction. Model-specific methods, on the other hand, are specifically tailored to a particular algorithm, for example artificial neural networks used for image classification [23]. The next category is global versus local explanation. Global explanation methods aim to explain tendencies on a more general level, for example by evaluating global feature importance. Local explanation methods on the other hand are used to explain individual samples predicted by the machine learning algorithm [20, 24]. Furthermore, XAI-methods can be distinguished by means of their timing of application. Pre-model methods are applied on the data directly, before the model is even trained [43], like for example methods of dimensionality reduction [25]. In- and post-model application is a matter of whether the XAI algorithm is directly integrated into the model or is applied to it afterwards [43]. In the last category, we distinguish XAI methods according to the form of presentation of the explanation, like for example text or tabular-based models, or visualization methods that directly visualize relations. This could be for example in the form of activation maps for the layers of a neural network [23, 43]. The most commonly used and frequently cited packages for XAI that are also publicly available are Local Interpretable Model-Agnostic Explanations (Lime) [33], Anchors High-Precision Model-Agnostic Explanations (Anchors) [34], and SHapley Additive exPlanations (SHAP) [22].

Only a few efforts have recently been made to use XAI for the purpose of facilitating simulation and especially data farming result analysis. However, the existing work is already very promising and yield a lot of potential for further development and future research, but none is focusing on system robustness. Feldkamp developed a basic workflow [4] according to which XAI approaches and methods can be used in the context of data farming output analysis: As a rule of thumb, only model-agnostic methods should be applied in order to remain independent from the type of machine learning algorithm. The workflow basically is top down with increasing level of detail, but also increasing computational cost. For example, the complete set of simulation result data can be investigated using global explanation methods like permutation feature importance. This does not explain distinct relations from factor values to output values, but can get insight about the influence of factors on the general variability of the simulation output. When drilling down further into the data by filtering on a distinct subset of output values, this subset can then be seen as a distinct class, so that its relation to the factor values can be explained using local explanation methods for classification problems like SHAP or Anchors. This can then be further drilled down to the explanation of even single experiments, where the outcome of single simulation output can be explained using local explanation methods that support the explanation for regression prediction models (that maps factors to a numeric output) [4]. A summary of the possible applications for XAI for data farming output analysis is given in Table 2.

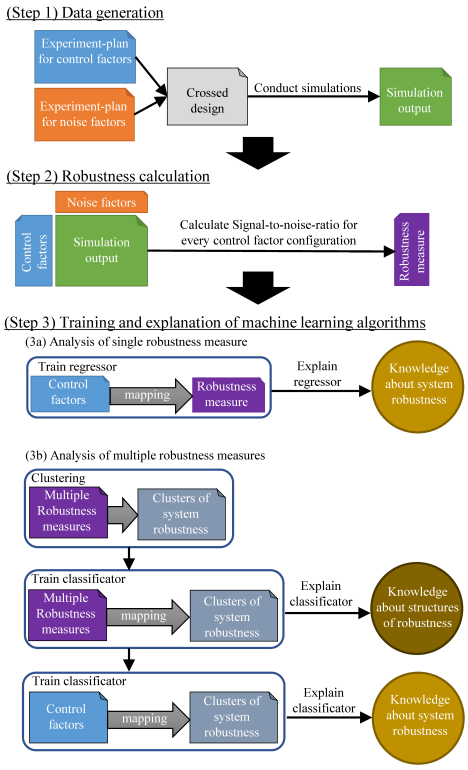

Figure 2: Concept for Robustness Analysis Using Data Farming and XAI

Feldkamp et al. applied this workflow in a real world case study conducted in the context of automotive manufacturing [9]. Serré et al. applied XAI-methods for explaining metamodels generated within a data farming process [41], and also presented case studies in the context of defense applications [42]. In the next section, we present our concept for using XAI for robustness analysis.

3 CONCEPT FOR ROBUSTNESS ANALYSIS USING DATA FARMING AND XAI

In this section, we explain the concept for using machine learning and XAI to analyze and better understand system robustness. This includes for example identifying robust configurations, identifying critical combinations of noise factor values, and investigating which factor values and combinations contribute to robustness and which do not. This differs from approaches of optimizing robustness, where the overall most robust configuration can be found, but we learn nothing about the underlying rules and relations regarding why the system is robust. Figure 2 shows the general workflow of this concept, which we will walk through exemplarily in the next section using a case study. The coloring shows which data from a sub step is processed in the subsequent steps.

The first step is data generation. In accordance with the basic idea of data farming, the basis for this is to generate a very large

Table 2: Applications of XAI for data farming output analysis [4]

Application	XAI approach	XAI method
Investigating important factors in the global dataset	Global explanations	Permutation feature importance
Investigating important factors and relations for specific filters, explanation for specific filters on the dataset	Global explanations, local explanation	Permutation feature importance, explanation for classification
Explanation for a specific experiment	Local explanation	Explanation for regression

amount of simulation data by combining all - or at least a very large number of - possible factor combinations, which allow a comprehensive insight into the model and its behavior. This is based on smart experiment design on the one hand and on the other hand on high performance computing and parallelization, as explained in Section 2.2. However, it must be taken into account that the number of experiments that need to be conducted is leveraged even more by crossing control and noise factor plans. In the second step, the signal-to-noise ratio for the considered output is calculated for each configuration of the control factors over all configurations of the noise factors, according to the formulas given in Section 2.1. I.e., for each configuration of control factors a robustness value is calculated. At the end of this step a set of factor value combinations x is mapped to the robustness value y. That means robustness always applies to one output variable under consideration. If we want to investigate the robustness for more than one output variable, this step must be repeated for each of the considered outputs. I.e., the set of factor combinations x is matched by a set of robustness values for different outputs. In the third step, both machine learning and XAI come into play. Here we must distinguish whether only a single robustness measure is of interest or whether the robustness of several outputs is to be investigated simultaneously. In the first case (Figure 2, step 3a), a regressor is trained that maps the relationship between the control factors and the corresponding robustness measure. Preference should be given to regression algorithms that are able to establish complex, non-linear relationships, such as random forests or artificial neural networks. The trained regressor can then be analyzed with XAI by explaining selected control factor configurations. For example, we can take the configuration with the highest and lowest measured robustness value and let XAI explain, why the robustness is high or low respectively. This can be used to derive information about the system robustness.

The second case, where we aim to understand the robustness of multiple outputs simultaneously, is slightly more complicated and consists of several sub steps (Figure 2, step 3b). Here, we first apply a clustering algorithm to the multidimensional robustness values. Using clustering algorithms like k-means, we are able to uncover structures within multiple dimension. The clustering algorithm groups according to similarity. Thus, groups are formed so that such control factor configurations in the same cluster are very similar with respect to robustness of multiple outputs, and at the same time very dissimilar to control factor configurations outside of their own cluster. While the clustering algorithm helps to group the robustness data, it can be difficult to subsequently understand, characterize, and qualitatively evaluate the composition of the clusters manually, because we have a lot of data with each data point having multiple dimensions. It is therefore helpful to first train a machine learning algorithm that initially only maps the relationship between the robustness dimensions and the corresponding clusters. In contrast to the regressor in step 3a, which maps the direct relationship between control factor configurations and a metric robustness value, a classifier is now required, since the clusters each represent discrete classes. With the subsequent use of XAI, the clusters can now be explained by explaining their centroids, i.e., the respective midpoints of the clusters.

With this knowledge about the inherent structures of robustness, the final relationship between the control factor configurations and those structures represented by the clusters can be established by training another classifier. This maps accordingly from the control factor configurations to the clusters. If this model is in turn explained by XAI, finally knowledge about the system robustness can be obtained, or more precisely, about the relationship between the control factor configurations and the resulting structures and distributions of the system robustness. In the next section, this workflow will illustrated using a short case study as an example.

4 CASE STUDY

4.1 Simulation Model, Experiment Design, and Data Generation

In this chapter, we will demonstrate the workflow of the concept presented in the previous section using a straightforward case study as an example. The simulation model was adapted from [5], where we used this model in order to optimize robustness in terms of finding the most robust configuration. In this work, we added more decision factors for the experiment design in order to explore the inherent relation between factors and robustness and learn what factors and factor values contribute to robustness and what might even be counter-productive for robustness by using our proposed method. The model was implemented using Siemens Plant Simulation. A screenshot of the model is shown in Figure 3.

The simulation model represents a small assembly line. In this model, three different job types are loaded onto carriers that are then transported via a conveyor system. Some jobs are processed on station 1, that can be scaled up to 5 slots, other job types are processed on station 2. At the end of the line, there is a quality inspection (station 3), that decides whether or not a processed job needs rework, which is the case for a fixed proportion of jobs. Jobs that need rework take the conveyor back to the main queue,

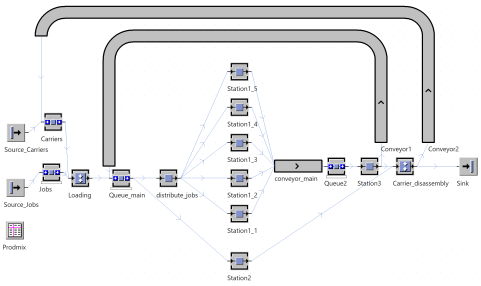

Figure 3: Screenshot of the simulation model implemented in Plant Simulation

otherwise they are unloaded from their carrier and leave the system. The mixture of jobs can vary, and arriving jobs are kept in a buffer until they are cleared to get mounted on a free carrier. Some stochastic effects arise through machine reliability and the proportion of jobs that fail the quality assurance and are rescheduled for manufacturing. Therefore, individual experiments with the same factor configurations need to be replicated multiple times to get a meaningful average for corresponding outputs.

The goal here is to make the line robust against variations in the product mixture. Therefore, the proportions of job types that are summed up to the total product mix can be varied and are considered the uncontrollable noise in the model, because we make the assumption that it is not possible to control which types are ordered by the customer or any upstream production system that feeds into this system. Regarding the experiment plan, we used a nearly orthogonal nearly balanced mixed design template (NOB-Mixed-Design) that offers 512 design points [48], which is a renowned experiment design for data farming projects [47]. Table 3 shows a breakdown of all control factors and their respective factor limits.

For the experiment design for the noise configurations, we used a simplex lattice design, which is the full-factorial equivalent in a mixture design problem. With a grid size of 10, this resulted in 66 design points. For the crossed design needed for the robustness calculation, this finally resulted in 512 x 66 = 33792 simulation experiments.

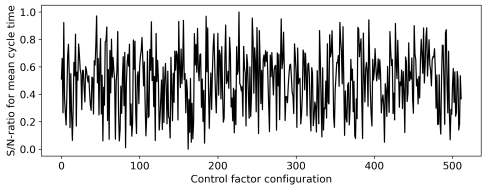

Figure 4: S/N-ratio for the output mean cycle time over all control factor configurations

4.2 Application and Results of Robustness Explanation

4.2.1 Calculation of Robustness for Cycle Time. After all experiments are conducted, we first have to calculate the robustness measures via signal-to-noise ratio. In the first approach, we only look at one single output, and we chose the mean cycle time of jobs for this purpose. If we assume that this output should be as low as possible while also being robust, we apply the smaller-the-better formula according to Taguchis robustness formulas (see Table 1). Figure 4 shows the distribution of the robustness over all 512 control factor configurations. For easier comparison, the values have been transformed on a scale from 0 to 1, where 0 represents the least robust and 1 represents the most robust configuration.

4.2.2 Explanation of the Relation between Factors and Robustness of Cycle Time. Following the workflow outlined in the previous section, we can now train a regression model to map the relation between control factors and the robustness directly. For this purpose, we used a boosted decision tree regressor using adaptive boosting with 100 estimators and maximum depth of 10000. This algorithm is very good at approximating complex input/output-relations, but opaque in terms of traceability. The R-squared performance for the total dataset was at 99.99%. Note that in this approach, an overfitting is actually desired, since we don't what to use the regressor to predict the y of unknown x, but to map the relation between x and y as exact as possible in order to gain knowledge out of it. Therefore, in the next step, we used XAI to explain the mapped relation. For this purpose, we used SHAP-values [22] using the respective python package (see https://github.com/slundberg/shap). Figure 5 shows the explanation of the most robust control factor configuration (S/N-ratio = 1).

Table 3: Control factors and factor limits

Factor name	Description	Margins
S1NumberOfSlots	Number of parallel slots for station 1	1-5 (discrete)
NumberOfCarriers	Number of work piece carriers for the conveyor system.	1-100 (discrete)
Queue1Capacity	Maximum capacity of queue 1	5-100 (discrete)
Queue2Capacity	Maximum capacity of queue 2	5-100 (discrete)
Conveyor1Speed	Speed of conveyor 1 in meters per second	1-4 (continuous)
Conveyor2Speed	Speed of conveyor 2 in meters per second	1-4 (continuous)
ConveyorMSpeed	Speed of conveyor main in meters per second	1-4 (continuous)
S3ProcTime	Process time of station 3 in seconds	1-60 (continuous)

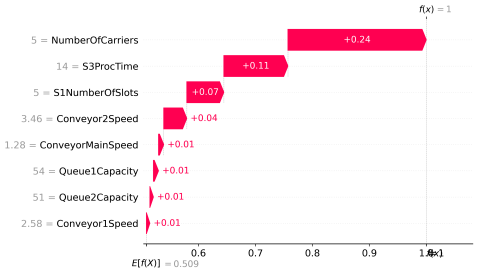

Figure 5: Explanation of regression model for the simulation experiment that exhibits the highest robustness using SHAP

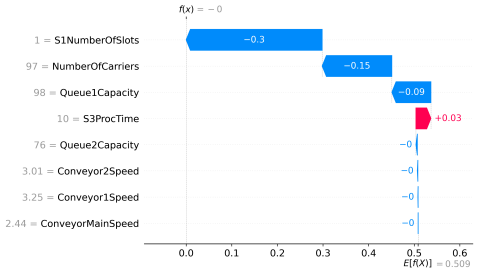

Figure 6: Explanation of regression model for the simulation experiment that exhibits the lowest robustness using SHAP

The result of the SHAP-value calculation is broadly comparable to the coefficients of linear regression model, but the relationship between x and y is not of a linear nature. This means that we can track each factors contribution (which can be positive or negative) to the overall prediction, comparable to what the coefficients of a linear regression model would do the target value. We can see, that the trained regressor predicts a value of 1 for this control factor configuration (f(x)=1), while the expected mean over all samples is 0.509 (E[f(x)]=0.509). The plot shows the contribution of every factor and its corresponding value for the explained sample with respect to the predicted value. Starting from expected mean, these contributions show how the explained sample is dragged away from this mean value towards the actual predicted value. The conclusion from Figure 5 that we can learn therefore is, that the most import contribution to the robustness of the cycle time of jobs is having five carriers. The fact that we have five carriers in the system adds 0.24 to the prediction of the perfect robustness value of 1 for this factor configuration, followed by the process time of 14 for station 3, which adds 0.11 to the robustness and five slots for station 1, which adds 0.07 to the robustness of the cycle time. A lower number of carriers will probably result in jobs waiting for transport, but a higher number will presumably clog the system, as we can see in Figure 6. Here, the explanation of the control factor configuration with the lowest robustness regarding cycle time is shown.

The predicted robustness is zero, and we can see how various factors drags the robustness away from the expected mean towards zero. For example, the fact that we have only one slot for station 1 decreases the predicted robustness by 0.3, followed by a very high number of 97 carriers in system. So we can conclude here that a high

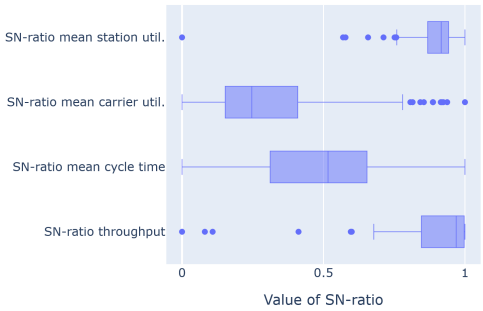

Figure 7: Value distribution for multiple robustness dimensions

number of carriers may clog the system in combination with only one available slot for station 1. This probably results in a bottleneck that probably contributes most to a very unstable, non-robust cycle time. In fact, the contribution of having only one slot for station 1 to the prediction of a poor robustness is over four times as high as the contribution of having five slots to a prediction of perfect robustness. Interestingly, in this control factor configuration, the process time for station 3 was 10 seconds, which actually increases the robustness by 0.03, but with a contribution of only 0.03 this is probably negligible.

4.2.3 Explanation of the Relation between Factors and Robustness of multiple Outputs. As explained in the previous section, this approach can also be used to investigate the robustness for multiple dimensions. In this particular example, we want to investigate the robustness of four outputs, namely throughput, mean cycle time, mean carrier utilization and mean station utilization. Except for mean cycle time, we assume for all other outputs that these should be robust and as high as possible, therefore Taguchis larger-the-better formula is applied. Figure 7 shows the distribution for the corresponding S/N-ratios over all control factor combinations.

We can see that the distribution of values is very irregular and uneven. For mean station utilization and throughput, the mean expected robustness is high, which means that most control factor configurations are robust anyway, however there are some outliers with very poor robustness. For the mean carrier utilization, it is the opposite, while mean cycle time is almost evenly distributed. Using a correlation matrix shown in Figure 8, we can see that the relationship of the four dimensions to each other is also very heterogeneous.

We can see that some dimensions are correlated slightly positively, while others correlated negatively, and some do not correlate at all. The strongest correlation is between the S/N-ratios of mean cycle time and throughput, which is still only 0.56. This shows that, even in this comparatively simple simulation model, the global, multi-dimensional space of robustness is in fact very uneven und not linear. To uncover possible structures in this space, we can use a clustering algorithm, as we have outlined in the previous section. To find the best possible structuring of data, we tested two different clustering algorithms, namely k-means and Gaussian mixture model clustering (GMM), each with different numbers of clusters.

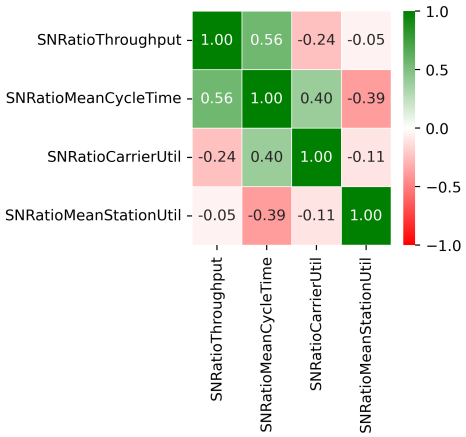

Figure 8: Correlation matrix for multiple robustness dimensions

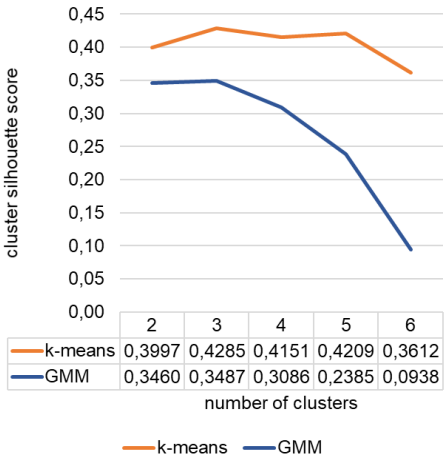

Figure 9: Cluster size evaluation using silhouette coefficient

To evaluate the goodness of the clustering, we calculated the silhouette score, where a higher number indicates a better clustering, as shown in Figure 9.

Given the results of the silhouette score calculation, a clustering using k-means with three clusters is to be preferred. Figure 10 shows a matrix plot of the clustering in the four-dimensional robustness data.

As we can see from Figure 10, the distribution of data in those clusters can be cumbersome to manually review and understand, with clusters even overlapping to some extend in some dimensions. Following our proposed workflow, we can therefore train a classifier that maps the relation between the robustness values and their respective clusters, which can then be explained using XAI afterwards. For this purpose, we used a random forest classifier with 1000 estimators, achieving an R-squared performance of 100%. In the next step, we used the Lime python package (see https://github.com/marcotcr/lime) to explain each of the cluster's

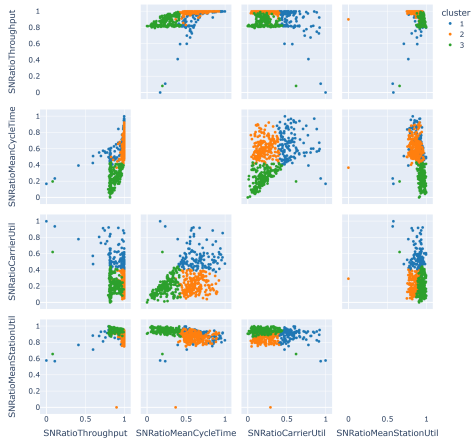

Figure 10: Distribution of clusters in their respective robustness dimensions

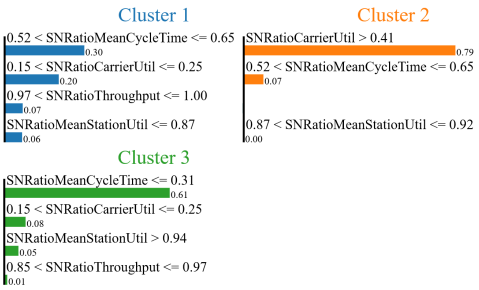

Figure 11: Explanation of clusters using Lime

center to get a textual and easy-to-understand explanation of each cluster composition. This is shown in Figure 11.

The lime explanation shows the contribution of each feature value to the overall probability of the prediction to respective cluster. Using this information, we can rename the clusters with more meaningful cluster names, as shown in Table 4 below.

Now we have identified the composition of the clusters and meaningfully condensed the multidimensional space of robustness. We can therefore conclude how robustness distributes in the system over the four dimensions that we selected. In fact, there is no scenario where we can make all four outputs (throughput, cycle time of jobs, utilization of carriers, and utilization of stations) very robust at the same time. If want a robust throughput, we can combine this either with making the cycle time robust, or with making station utilization robust. This is probably due to the fact that if we want the output of jobs to be robust and stable in terms of quantity and cycle time, we need a higher number of slots for station 1. This in turn will let the mean utilization of the stations fluctuate when product mixtures occur through noise factors that do not put enough load on the system in order to adequately utilize all slots of station 1. In fact, the configurations that have the highest robustness for throughput will also exhibit a high variance in station utilization. Following the workflow that we proposed in

Table 4: Renaming of clusters

Cluster	Renamed clusters
Cluster 1	"throughput and cycle time will be robust"
Cluster 2	"highest robustness in throughput, but carrier and station util. will have high variance"
Cluster 3	"throughput and station util. will be robust, but cycle time and carrier util. will have high variance"

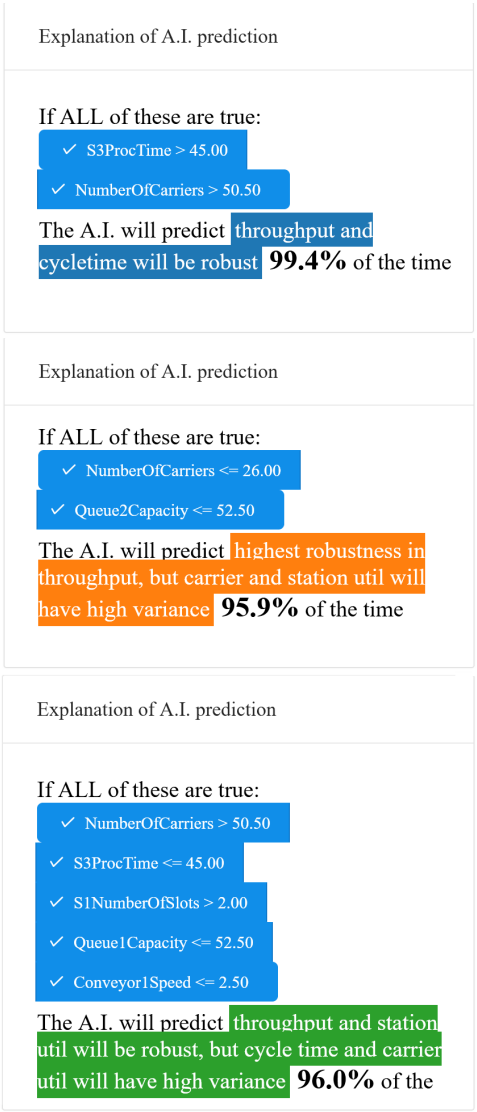

Figure 12: Explanation of classification model using Anchors

the previous section, we can now train a final classifier that maps the relation between factors and those clusters. For this purpose, we again used a large random forest of classification trees. This relation can then again be explained using XAI. Figure 12 shows the result of the explanation using the Anchors python package (see https://github.com/marcotcr/anchor).

Here, we took the control factor configurations whose S/N-ratios were most similar to the respective cluster center for each of the three clusters. The Anchors-XAI-method focuses on the feature values, in our case the values of the control factor configurations, and the corresponding prediction probability of the classifier. Because we renamed our cluster with meaningful information in the previous step, we can now conclude meaningful and condensed information about the control factors and their relation to the global system robustness represented by the three clusters. Because we learned from the previous step that we actually cannot achieve a perfect robustness for all four dimensions at the same time, we basically have to prioritize accordingly to desired requirements, and the results in Figure 12 then tell us how to get there. Interestingly, some of the required factor value combinations are quite different from what we expected in advance. For example if we want to achieve a good robust throughput and cycle time of jobs, we need process time of station 3 to be 45s and at least 51 carriers. However, we already learned in Section 4.2.2 that in order to achieve the maximum possible robustness for cycle time, we actually need a much smaller number of carriers. This in turn seems not be possible in combination with any other output dimension, where we actually need a higher number of carriers. We can achieve the highest robustness in throughput at the cost of a high variance in mean station and carrier utilization. On the other hand, we can achieve a robust throughput and mean station utilization, but at the expense of a high variance in mean cycle time and mean carrier utilization, assuming we have enough slots for station 1 and if we are willing to increase the number of carriers to over 51. However, this figure is only one illustrative example. Actually we can use several points from the clusters and have them explained. This way, which each explanation we can iteratively learn more about the relation of the factor values and their combination to the respective clusters in order to gain more your knowledge about the system.

4.2.4 Summary of Findings for the Case Study using the proposed Concept. In summary, we learned some insight about the behavior of the system in terms of robustness using the proposed method. Those can then be used for decision-making and improving robustness of the system. When considering the single selected output cycle time of jobs, we could conclude that a carrier number of five is by far the most important factor for making the cycle time of jobs robust against variations in the product mixture. On the other hand, while having five slots for Station 1 contributes to the robustness a little bit, having only one slot contributes strongly to a very poor robustness when combined with having a large number of carriers that might clog the system. Furthermore, this is only true when considering cycle time and ignoring the robustness of other outputs. Therefore, we also carried out an investigation of robustness over multiple output dimensions, namely, throughput,

cycle time of jobs, utilization of carriers, and utilization of stations. First, XAI helped us to structure and understand the simulation output data. By separating the output data into three clusters and letting XAI explain their components, we concluded that there is no scenario where we can achieve high robustness in all four output dimensions simultaneously. In fact, we rather have to prioritize the robustness of some outputs in favor of the robustness of others. Finally, we used XAI to explain a model that mapped the factor values to these clusters. Through this explanation, we could conclude what factor settings can be used in order to achieve system robustness according to one if the clusters.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a concept for using data farming, machine learning and explainable AI to investigate the system robustness. The applicability of the concept was demonstrated in an exemplary case study. In general, the application of black box machine learning algorithms and the subsequent removal of the opaqueness of those algorithms through XAI provides a whole new range of possibilities for simulation output analysis, especially when large quantities of simulation data are given and an automated yet a qualitatively appealing analysis is needed. Furthermore, this work contributes in closing the gap between simulation and AI/machine learning research communities. However, in future research, this workflow could even further be automated by presenting the user the most useful visualizations at any given stage of the process, thereby maximizing the users capability for visual reasoning [10]. Furthermore, the small scale of the simulation model used in the presented case study was chosen in order to illustrate the concept. This simulation model obviously does not actually require a data farming study to be understood thoroughly, but rather was used to demonstrate that insights can be found using the presented method. The applicability of this concept should therefore be put to test using more complex simulation models and large-scale experimentation from real world projects in future research. For instance, further research is needed for solutions when the data volumes for large-scale simulation models and large-scale experiment plans grow also very large, as some XAI methods can sometimes be very computationally intensive.

REFERENCES

[1] Alejandro Barredo-Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58, 82–115. DOI: https://doi.org/10.1016/j.inffus.2019.12.012.

[2] Vaishak Belle and Ioannis Papantonis. 2021. Principles and Practice of Explainable Machine Learning. *Front. Big Data* 4, 688969. DOI: https://doi.org/10.3389/fdata.2021.688969.

[3] Filip K. Dosilovic, Mario Brcic, and Nikica Hlupic. 2018. Explainable Artificial Intelligence: A Survey. In *41st International Convention on Information and Communication Technology, Electronics and Microelectronics*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 210–215. DOI: https://doi.org/10.23919/MIPRO.2018.8400040.

[4] Niclas Feldkamp. 2021. Data Farming Output Analysis Using Explainable AI. In *Proceedings of the 2021 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

[5] Niclas Feldkamp, Soeren Bergmann, Florian Conrad, and Steffen Strassburger. 2022. A Method Using Generative Adversarial Networks for Robustness Optimization. *ACM Trans. Model. Comput. Simul.* 32, 2, 1–22. DOI: https://doi.org/10.1145/3503511.

[6] Niclas Feldkamp, Soeren Bergmann, and Steffen Strassburger. 2015. Knowledge Discovery in Manufacturing Simulations. In *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. SIGSIM PADS '15. ACM Press, New York, New York, 3–12. DOI: https://doi.org/10.1145/2769458.2769468.

[7] Niclas Feldkamp, Soeren Bergmann, and Steffen Strassburger. 2020. Knowledge Discovery in Simulation Data. *ACM Trans. Model. Comput. Simul.* 30, 4, 1–25. DOI: https://doi.org/10.1145/3391299.

[8] Niclas Feldkamp, Soeren Bergmann, Steffen Strassburger, and Thomas Schulze. 2017. Knowledge Discovery and Robustness Analysis in Manufacturing Simulations. In *Proceedings of the 2017 Winter Simulation Conference*. IEEE Inc.

[9] Niclas Feldkamp, Jonas Genath, and Steffen Strassburger. 2022. Explainable AI For Data Farming Output Analysis: A Use Case for Knowledge Generation Through Black-Box Classifiers. In *Proceedings of the 2022 Winter Simulation Conference*. IEEE, 1152–1163.

[10] Jonas Genath. 2021. Automation within the Process of Knowledge Discovery in Simulation Data [Poster]. In *Proceedings of the 2021 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

[11] Jonas Genath, Sören Bergmann, Niclas Feldkamp, Sven Spieckermann, and Stephan Stauber. 2022. Development of an Integrated Solution for Data Farming and Knowledge Discovery in Simulation Data. *SNE* 32, 3, 121–126. DOI: https://doi.org/10.11128/sne.32.tn.10611.

[12] Bryce Goodman and Seth Flaxman. 2017. European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AIMag* 38, 3, 50–57. DOI: https://doi.org/10.1609/aimag.v38i3.2741.

[13] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5, 1–42. DOI: https://doi.org/10.1145/3236009.

[14] G. E. Horne and T. Meyer. 2010. Data farming and defense applications. In *MODSIM World Conference and Expo*. Langley Research Center, Hampton, VA, 74–82.

[15] Gary E. Horne and Ted E. Meyer. 2005. Data Farming: Discovering Surprise. In *Proceedings of the 2005 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 1082–1087.

[16] Gary E. Horne and Klaus-Peter Schwierz. 2008. Data Farming Around The World Overview. In *Proceedings of the 2008 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 1442–1447. DOI: https://doi.org/10.1109/WSC.2008.4736222.

[17] Joachim Hunker, Anne Antonia Scheidler, Markus Rabe, and Hendrik van der Valk. 2021. A New Data Farming Procedure Model for a Farming for Mining Method in Logistics Networks. In *Proceedings of the 2021 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

[18] Florian Klug. 2013. The internal bullwhip effect in car manufacturing. *International Journal of Production Research* 51, 1, 303–322. DOI: https://doi.org/10.1080/00207543.2012.677551.

[19] Tobias Lechler, Martin Sjarov, and Jörg Franke. 2021. Data Farming in Production Systems - A Review on Potentials, Challenges and Exemplary Applications. *Procedia CIRP* 96, 230–235. DOI: https://doi.org/10.1016/j.procir.2021.01.156.

[20] Yi-Shan Lin, Wen-Chuan Lee, and Z. B. Celik. 2020. *What Do You See? Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors*, arxiv.org Preprint. http://arxiv.org/pdf/2009.10639v1.

[21] Thomas W. Lucas, W. D. Kelton, Paul J. Sánchez, Susan M. Sanchez, and Ben L. Anderson. 2015. Changing the paradigm: Simulation, now a method of first resort. *Naval Research Logistics* 62, 4, 293–303. DOI: https://doi.org/10.1002/nav.21628.

[22] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, Eds. Curran Associates, Inc, 4765–4774.

[23] Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. 2019. iNNvestigate Neural Networks! *Journal of Machine Learning Research* 20, 93, 1–8.

[24] Christoph Molnar. 2019. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable* (1st edition). Lulu Com, Zürich.

[25] Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. 2022. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models. In *xxAI - Beyond Explainable AI*, Andreas Holzinger, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller and Wojciech Samek, Eds. Lecture Notes in Computer Science. Springer International Publishing, Cham, 39–68.

[26] Manuel E. Morocho-Cayamcela, Haeyoung Lee, and Wansu Lim. 2019. Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions. *IEEE Access* 7, 137184–137206. DOI: https://doi.org/10.1109/ACCESS.2019.2942390.

[27] Vijayan N. Nair, Bovas Abraham, Jock MacKay, John A. Nelder, George Box, Madhav S. Phadke, Raghu N. Kacker, Jerome Sacks, William J. Welch, Thomas J. Lorenzen, Anne C. Shoemaker, Kwok L. Tsui, James M. Lucas, Shin Taguchi, Raymond H. Myers, G. G. Vining, and C. F. J. Wu. 1992. Taguchi's Parameter

Design. A Panel Discussion. *Technometrics* 34, 2, 127. DOI: https://doi.org/10.2307/1269231.

[28] Michael K. Painter, Madhav Erraguntla, Gary L. Hogg, and Brian Beachkofski. 2006. Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management. In *Proceedings of the 2006 Winter Simulation Conference. Monterey, CA, U.S.A., Dec 3-6, 2006*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey.

[29] Gyung-Jin Park, Tae-Hee Lee, Kwon H. Lee, and Kwang-Hyeon Hwang. 2006. Robust Design. An Overview. *AIAA Journal* 44, 1, 181–191. DOI: https://doi.org/10.2514/1.13639.

[30] Amir Parnianifard, A. S. Azfanizam, M.K.A. Ariffin, and M.I.S. Ismail. 2018. An overview on robust design hybrid metamodeling: Advanced methodology in process optimization under uncertainty. *10.5267/j.ijiec* 9, 1–32. DOI: https://doi.org/10.5267/j.ijiec.2017.5.003.

[31] Madhav S. Phadke. 1989. *Quality engineering using robust design*. Prentice Hall, Englewood Cliffs, N.J.

[32] Gabriëlle Ras, Marcel van Gerven, and Pim Haselager. 2018. Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, Hugo J. Escalante, Sergio Escalera, Isabelle Guyon, Xavier Baró, Yağmur Güçlütürk, Umut Güçlü and Marcel van Gerven, Eds. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham, 19–36. DOI: https://doi.org/10.1007/978-3-319-98131-4_2.

[33] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, 1135–1144. DOI: https://doi.org/10.1145/2939672.2939778.

[34] Marco T. Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1, 1527–1535. DOI: https://doi.org/10.1609/aaai.v32i1.11491.

[35] S. M. Sanchez. 2000. Robust Design: Seeking the Best of all Possible Worlds. In *Proceedings of the 2000 Winter Simulation Conference*. IEEE Inc, Piscataway, N.J., 69–76. DOI: https://doi.org/10.1109/WSC.2000.899700.

[36] Susan M. Sanchez. 1994. A Robust Design Tutorial. In *Proceedings of the 1994 Winter Simulation Conference*, 106–113. DOI: https://doi.org/10.1109/WSC.1994.717084.

[37] Susan M. Sanchez. 2014. Simulation Experiments: Better Data, Not Just Big Data. In *Proceedings of the 2014 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 805–816.

[38] Susan M. Sanchez. 2020. Data Farming: Methods for the Present, Opportunities for the Future. *ACM Trans. Model. Comput. Simul.* 30, 4, 1–30. DOI: https://doi.org/10.1145/3425398.

[39] Susan M. Sanchez and Paul J. Sanchez. 2020. Robustness Revisited: Simulation Optimization Viewed Through A Different Lens. In *Proceedings of the 2020 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 60–74. DOI: https://doi.org/10.1109/WSC48552.2020.9383880.

[40] Andrew D. Selbst and Julia Powles. 2017. Meaningful information and the right to explanation. *International Data Privacy Law* 7, 4, 233–242. DOI: https://doi.org/10.1093/idpl/ipx022.

[41] Lynne Serre and Maude Amyot-Bourgeois. 2022. An Application of Automated Machine Learning Within a Data Farming Process. In *Proceedings of the 2022 Winter Simulation Conference*. IEEE, 2013–2024. DOI: https://doi.org/10.1109/WSC57314.2022.10015513.

[42] Lynne Serre, Maude Amyot-Bourgeois, and Brittany Astles. 2021. Use of Shapley Additive Explanations in Interpreting Agent-Based Simulations of Military Operational Scenarios. In *2021 Annual Modeling and Simulation Conference (ANNSIM)*. IEEE, 1–12. DOI: https://doi.org/10.23919/ANNSIM52504.2021.9552151.

[43] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. 2020. Explainable Deep Learning Models in Medical Image Analysis. *Journal of imaging* 6, 6, 1–19. DOI: https://doi.org/10.3390/jimaging6060052.

[44] Steffen Strassburger, Sören Bergmann, Niclas Feldkamp, Kristina Sokoll, and Matthias Clausing. 2018. Data Farming Research Project with Audi and VW. In *2018 Plant Simulation Worldwide User Conference*.

[45] Genichi Taguchi. 1995. Quality engineering (Taguchi methods) for the development of electronic circuit technology. *IEEE Trans. Rel.* 44, 2, 225–229. DOI: https://doi.org/10.1109/24.387375.

[46] Erico Tjoa and Cuntai Guan. 2020. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. DOI: https://doi.org/10.1109/tnnls.2020.3027314.

[47] H. Vieira, Susan M. Sanchez, K. H. Kienitz, and M. C. N. Belderrain. 2013. Efficient, nearly orthogonal-and-balanced, mixed designs. An effective way to conduct trade-off analyses via simulation. *Journal of Simulation* 7, S4, 264–275. DOI: https://doi.org/10.1057/jos.2013.14.

[48] Hélcio Vieira. 2012. *NOB_Mixed_512DP_v2.xls design spreadsheet* (2012). Retrieved February 1, 2023 from http://harvest.nps.edu/.