Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

DATA FARMING OUTPUT ANALYSIS USING EXPLAINABLE AI

Niclas Feldkamp

Group for Information Technology in Production and Logistics
Technische Universität Ilmenau
Max-Planck-Ring 12
Ilmenau, 98693, GERMANY

ABSTRACT

Data Farming combines large-scale simulation experiments with high performance computing and sophisticated big data analysis methods. The portfolio of analysis methods for those large amounts of simulation data still yields potential to further development, and new methods emerge frequently. Especially the application of machine learning and artificial intelligence is difficult, since a lot of those methods are very good at approximating data for prediction, but less at actually revealing their underlying model of rules. To overcome the lack of comprehensibility of such black-box algorithms, a discipline called explainable artificial intelligence (XAI) has gained a lot of traction and has become very popular recently. This paper shows how to extend the portfolio of Data Farming output analysis methods using XAI.

1 INTRODUCTION

For complex simulation models, conducting a traditional simulation study usually aims at meeting a specific, predetermined goal, like performing a scenario-based analysis or even simulation-based optimization. This can still leave a lot of room for actually understanding the behavior of the model in terms of relation between factors and outputs (Feldkamp et al. 2015; Painter et al. 2006). Sometimes the discovery of new and interesting relations, that previously were unknown and that are outside of a priorly defined simulation project scope, can actually improve decision-making. In this context, the method of Data Farming was established (Horne and Meyer 2005). Data Farming refers to the method of growing data from your simulation model by using large-scale experimental design and high performance computing for massively parallelized experiments in order to focus on a more complete coverage of possible system responses (Horne and Schwierz 2008). Data Farming problems falls into a category of problem solving that Lempert et al. define as long term policy analysis, where large ensembles of scenarios need to be considered (Lempert et al. 2003). Data Farming research has also always been addressing the application of modern data analytics methods to handle those large amounts of simulation output data and to provide reasonable and adequate insights (Lucas et al. 2015; Sanchez 2014). The concept of knowledge discovery in simulation data (KDS) was developed to take a deep dive into the analysis side of Data Farming, providing a detailed process model for applying data mining methods alongside suitable visualization and interactions methods onto large-scale farmed simulation data (Feldkamp et al. 2020). This is contributing to the idea of simulation modeling as a hybrid and connecting link between methods from different research disciplines (Mustafee and Powell 2018), mutually creating greater value for the analyst (Tolk et al. 2021). The concept is able to create knowledge that is decision-supportive, yet can discover surprise in terms of uncovering hidden relations in the system that the analyst did not think about before. One partial aspect of the KDS process model is to use model-building algorithms that can approximate relations between simulation input and output data. Rules from these models can then be used to infer knowledge about the system (Feldkamp et al. 2015). However, only white-box models such as frequent pattern mining, decision trees or Bayesian Networks are applicable. White-box models are easy to interpret, since they reveal their internal mapping

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

of the input/output relations (Feldkamp et al. 2020). In black-box models, this internal logic remains hidden, so that no derivation of generalizable rules is possible. However, some black-box models, with artificial neural networks being the prime example, are excellent at approximating even complex relations (Morocho-Cayamcela et al. 2019), which in turn would be very useful for simulations models with lots of factors and complex response surfaces. The objective in this paper therefore is extending the portfolio of Data Farming output analysis methods while also contributing to narrowing the gap between simulation and AI research.

The lack of explainability and transparency of black-box algorithms is more and more becoming a problem in general, for instance in security critical applications, or if legal issues are affected, such as credit accommodation and loan default predictions (Dosilovic et al. 2018; Guidotti et al. 2019). To overcome the lack of explainability of such algorithms, a discipline called explainable artificial intelligence (XAI) has gained a lot of traction and has become very popular recently (Adadi and Berrada 2018; Barredo Arrieta et al. 2020). The goal of this research is to peek inside the black-box of those algorithms in order to make their decisions reconstructable and comprehensible. In this paper, we propose a workflow on how XAI methods can be used for Data Farming output analysis. The remainder of this papers is structured as follows: In section two, we give an overview of the related work, namely the current state of XAI methods, as well as a brief overview of Data Farming and Knowledge Discovery in Simulation Data. In Section 3, we show which XAI methods are applicable for the specific demands and requirements of interpreting massive amounts simulations data. This is then followed by some practical demonstrations using an example scenario in Section 4, and then concluded by final remarks and a discussion of possible future work in Section 5.

2 RELATED WORK

2.1 Data Farming and Knowledge Discovery in Simulation Data

Usually simulation studies aim at solving a clearly outlined goal. According to Law, not having such is even one of the greatest pitfalls (Law 2003). On the other hand, simulation experts traditionally aimed at minimizing computational effort when computing time and memory space have been rare and costly. Additionally, in real world situations, the simulation analyst might simply take an educated guess based on his experience about which factors might be influential in contributing to the goals of the simulation study. Kleijnen et al. refer to this as the trial-and-error approach to finding a good solution and argue that simulation analysts should spend more time in analyzing than in building the model (Kleijnen et al. 2005). Traditional, goal-based experimentation attempets to pinpoint an answer to very specific questions whereas the Data Farming method aims to cover the whole range of possible system behavior to allow a better understanding of possibilities and to discover all potential options (Horne and Schwierz 2008). The farming metaphor describes the maximization of data output in the most efficient way, resembling a farmer that cultivates his land in order to maximize his crop yield (Sanchez 2014). Data Farming combines large-scale simulation experiments with high performance computing and sophisticated big data analysis methods (Sanchez and Sanchez 2017). As an extension, the concept of knowledge discovery in simulation data was developed to take a deep dive into the analysis side of Data Farming by providing a process model and workflow for using data mining methods and suitable, interactive visualizations (Feldkamp et al. 2015). This is especially valuable for models with a large number of relevant outputs that exhibit a complex, multidimensional response surface, where outputs might even be contradictory or in a trade-off situation to each other and are therefore very difficult to mutually interpret. When outputs are congregated multidimensionally into groups of disparate system behavior, the relation between factors and outputs can then be investigated using algorithms that provide multidimensional pattern recognition, as shown in Figure 1 (Feldkamp et al. 2020). This makes the analysis and interpretation of large amounts of simulation data even from complex models much more manageable, as has been shown in various case studies (Lechler et al. 2021; Strassburger et al. 2018).

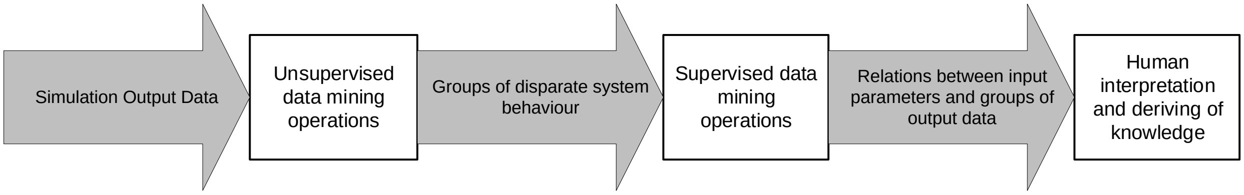

Simulation Output Data
Unsupervised data mining operations
Groups of disparate system behaviour
Supervised data mining operations
Relations between input parameters and groups of output data
Human interpretation and deriving of knowledge

Figure 1: KDS process for analysis of output data (Feldkamp et al. 2020).

More precisely, finding patterns in uncategorized data is known as unsupervised data mining. Subsequently, the relation between patterns of output data and corresponding factors can be analyzed using supervised data mining algorithms, when the simulation data was previously categorized into different groups of system behavior. Supervised learning algorithms can create models that represent the relations between simulation input and output data, from which in turn we can derive rules that can be contribute to knowledge creation through human interpretation. For a supervised algorithm, each simulation experiment acts as a training record (Feldkamp et al. 2020). In other words, a classification problem needs to be solved. Extracting the underlying decision rules for the classification from white-box algorithms is relatively easy, as those are explicitly visible. XAI, as described in the next section, enables the extraction from back-box algorithms.

2.2 Explainable Artificial Intelligence

Black-box methods of machine learning and artificial intelligence, such as artificial neural networks, are among the most powerful algorithms for tasks of prediction and classification. Due to the complexity of those algorithms in contrast to their white-box counterparts, such as decision trees and linear regression, there is a trade-of between performance and readability (Dosilovic et al. 2018), as shown in Figure 2.

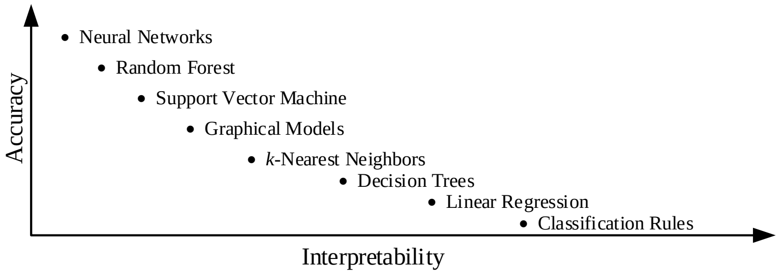


Figure 2: Performance vs. readability trade-off (Morocho-Cayamcela et al. 2019).

Transparency of decisions made by artificial intelligence and machine learning algorithms is becoming more and more important when they directly affect people’s lives. A decision explained with “because the computer said so” is not sufficient anymore and can consequently even cause legal trouble when being under suspicion of unfairness and discrimination, e.g., in areas like recruiting or credit granting (Dosilovic et al. 2018; Guidotti et al. 2019). Some lawmakers even think about a “right to an explanation” (Edwards and Veale 2017). Therefore, XAI has become a very popular research field recently in trying to make black-box algorithms transparent. Since there is no unified definition, the term XAI actually comprises a very broad scale of heterogeneous methods. Various efforts to categorize those methods exists (Barredo Arrieta et al. 2020; Ras et al. 2018; Tjoa and Guan 2020). One of the most straightforward categories for distinction is between global and local explanation. Global explanation methods aim to explain the general model, usually by evaluating the importance of individual factors and factor values. Local explanations on the other hand try to explain why an individual prediction was made. Local explanations are usually more detailed in their explanation, but in turn are only looking at the individual prediction that is currently evaluated. Some methods try to provide a more global viewpoint by summing up a large number of local predictions, but these methods tend to be extremely computation-intense (Lin et al. 2020; Molnar 2019). Another distinction between those methods is whether they are model-agnostic or model-specific. Model-agnostic methods are mostly independent of the underlying type of algorithm. That means their explanation is solely based on the relation between input and output. Model specific methods, on the other hand, are specifically tailored for an algorithm, such as artificial neural networks or random forests. They directly hook into the internals of these algorithms in order to provide the explanations. For example, this is used for explanation of image classification using neural networks, where model-specific algorithms can transparently explain which layer of the network was activated and is responsible for the detection of certain patterns that result in the complete image (Barredo Arrieta et al. 2020; Belle and Papantonis 2020). Most methods come with some visual component for the illustration of the actual results, while some methods offer text-based explanation in a natural language. Numerous methods have been developed for a specific use case defined by its underlying data type, like images, sequence data, or the explanation of predictions for text classifiers (Adadi and Berrada 2018). Feature relevance measures, that simply calculated the importance of each input to the overall outcome, are the most straight-forward and commonly used methods in the context of XAI. Examples are partial dependence and individual conditional expectation plots (Goldstein et al. 2015), and permutation feature importance (Altmann et al. 2010). Counterfactual explanations offer a contrary approach by explaining how the input needs to change in order to achieve a desired output compared to the actual output (Stepin et al. 2020). More individualized XAI methods and algorithms often come in their own software package, that, besides the actual XAI algorithms, usually include methods for data transformation and visualization. New software tools and packages are developed with an impressive velocity, but the most commonly used and frequently cited are Local Interpretable Model-agnostic Explanations (Lime) (Ribeiro et al. 2016), Anchors High-Precision Model-Agnostic Explanations (Anchors) (Ribeiro et al. 2018), and SHapley Additive exPlanations (SHAP) (Lundberg and Lee 2017). Table 1 provides a summary of these methods.

Table 1: Overview of popular XAI-packages.

Name	Scope	Methodology	Data	Presentation	Principle
Lime	Local	Rule extraction by simplification	Tabular data, text classification, image classification	Text-based, visual	Approximation of a simplified, interpretable model around the local target prediction
Anchors	Local	Rule extraction	Tabular data, text classification, image classification	Text-based	Construction of high-precision rules around the local target prediction
SHAP	Global and local	Feature relevance	Tabular data, sequence data, text classification, image classification	Visual	Game-theoretic approach of calculating Shapley values for optimal credit allocation

3 USING XAI FOR DATA FARMING

3.1 Preliminary Considerations

Before an XAI-method can be applied, obviously a black-box-model to be explained is needed. While not being the main focus of this paper, we need to discuss some basic and preliminary considerations to keep in mind when using black-box-algorithms for approximation of simulation data and subsequent analysis of Data Farming output. As already stated in Section 2.1, the KDS workflow groups multiple simulation model outputs into groups of disparate system behavior (or similar system behavior within one group, respectively) using unsupervised methods like clustering algorithms. If we then map the corresponding factors to these clusters, we basically have a classification problem, but not in terms of predicting the class of new, unknown inputs, but rather in terms of understanding which input values relates to which class in order to gain insight into the system behavior. While the original KDS concept was limited to white-box classifiers, using XAI methods opens up the opportunity for using theoretically any available black-box classifier (see Section 2.2, Figure 2). In this paper, we focused on using artificial neural networks and random forest in our subsequent case study. Training those classifiers has to be a bit different compared to traditional classification tasks though. Typically, various measures against overfitting are put into place, like splitting training and test sets, cross-validating, and other regularization techniques. The idea behind this is to prevent the model from memorizing and approximating the training data in such a way that it is not able to accurately predict new and unknown cases anymore (Hawkins 2004). However, the use-case for Data Farming is a completely different one. The model is not utilized to make predictions but rather to learn from its internal input/output mapping and approximation of rules. Because we already have a large and mostly complete representation of the response surface (within the given input value limits, though), an overfitting of the model does not cause the usual problems. Instead, a training phase as thoroughly as possible is desired, so that the existing data can be fitted as smoothly as possible. Furthermore, in stochastic models, a trade of between number of experiments in total and number of replications per experiment emerges, if we consider computational resources limited and fixed. MacDonald and Gunn showed that a more dense input space should be favored over the statistical accuracy of each individual point when training neural networks for metamodeling purposes, so this rule of thumb should be adapted to this approach for Data Farming as well (MacDonald and Gunn 2012).

3.2 Workflow for XAI Application

According to the KDS workflow, simulation outputs are clustered based on their output behavior. Therefore, the most interesting use case for Data Farming analysis is to use machine learning algorithms for classification of factors to a cluster allocation. Then, XAI methods can be leveraged for the subsequent investigation of this classification. In this section, we propose a workflow for the latter. The most challenging problem here is that we typically have to deal with massive amounts of data in a Data Farming study. Those studies consists of large-scale simulation experiments that are usually stored in some tabular form, like simple comma separated values (csv). The largeness of data does not necessarily apply to the sheer file size, but certainly in regards to the amount of records. Unfortunately, some XAI-algorithms struggle with scaling up to large data, and among the most effort-intensive ones are those that provide the most detailed insight into the black-box (Lin et al. 2020). In traditional XAI applications like image classification, this is combated by drawing a reasonable sample from the total dataset. This is in turn can be problematic in a Data Farming study because the dataset usually is created using a sophisticated experiment design method, where every single data point is important in its contribution to the overall picture. Therefore, a trade-off between level of detail and computational effort arises. The most straight forward way is to apply more lightweight algorithms first and then to increase the level of detail gradually by applying more computation-intensive methods until the limits of accepted computational cost is reached. The outline of this workflow is shown in Figure 3.

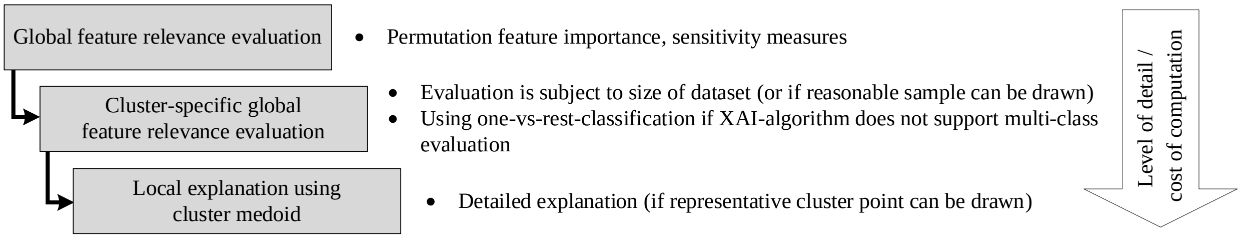

Global feature relevance evaluation
• Permutation feature importance, sensitivity measures
Cluster-specific global feature relevance evaluation
• Evaluation is subject to size of dataset (or if reasonable sample can be drawn)
• Using one-vs-rest-classification if XAI-algorithm does not support multi-class evaluation
Local explanation using cluster medoid
• Detailed explanation (if representative cluster point can be drawn)
Level of detail / cost of computation

Figure 3: Workflow for application of XAI methods for Data Farming analysis.

The starting point are easy-to-compute relevance measures, like permutation feature importance. These measures asses the overall importance of an factor to the output of the model. Since these measures are interpreted regardless of the actual classification, they only provide a coarse overview. The next step is to extend the analysis into the actual classifications (that are factor to cluster allocations) on a global level. As described in Section 2.2, global explanation means the evaluation of the general model (in contrast to explaining individual predictions). With these methods, we can draw conclusions regarding the differences between the individual clusters, in terms of which factor values contribute to which cluster allocation. It is worth noting that many XAI methods only support explanations for binary classifications. However, using the KDS-workflow, we usually group the data into multiple clusters therefore creating a multi-class classification problem. This problem can be bypassed by transforming the analysis into a one-vs-rest-classification. This means we compare each cluster of interest against the rest of the data. The downside of this approach is that a separately trained classification-model and a subsequent XAI-evaluation for each one of those comparisons is needed. The final step is using local explanation methods, that usually provide the most detailed insight. In a Data Farming study, looking at individual experiments usually provides only limited value. Nevertheless, we can use the cluster medoid as a representative for the total cluster. The medoid of a cluster is the nearest existing point to the theoretical center of the cluster (centroid), where the distance to every point of the cluster's boundary is the same. However, this only makes sense with partitioning cluster algorithms like k-means that are based on a defined center, and when the response surface is dense and therefore the actual medoid is not too far away from the calculated cluster center. In the next section, we demonstrate the application of this workflow using a case study scenario.

4 USE CASE

4.1 Simulation Model

The first use case is a simple simulation model representing a modified single-server-example. Figure 4 shows a screenshot of the model which has been implemented in Siemens Plant Simulation.

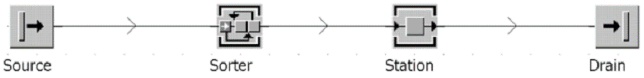


Figure 4: Screenshot of the modified single server model.

The adjustable input factors of the model include the deterministic interarrival time of jobs (*ArrivalTime*), the maximum capacity of the sorter (*SorterCap*), the sorting strategy, and the mixture of product types that are processed on the station (*PropA*, *PropB*, *PropC*). The product mixture determines the probability of a job being one of three different product types when entering the system, each having different values for setup and process time. The sorting strategies that the sorter can implement are: First in first out (*FIFO*), shortest processing time (*SPT*), minimum slack time (*SLACK*), a weighted combination of SPT and earliest due date (*SPTEDD*), and sorting according to current station setup state (*SetupOptimal*). The number of experiments that have been conducted in a corresponding Data Farming study sums up to 1.05 million. The most relevant outputs in this model according to their magnitude of variance are throughput, station utilization and proportion of setup time. A k-means clustering with 5 clusters on these outputs provided the best data separation and is shown in Figure 5 as a parallel coordinates plot, where each line represents one simulation experiment. We can see that cluster 5 (orange) collects the best performing experiments (having high throughput, high utilization and low proportion of setup processes), and cluster 1 (blue) consists of mostly bad performing experiments.

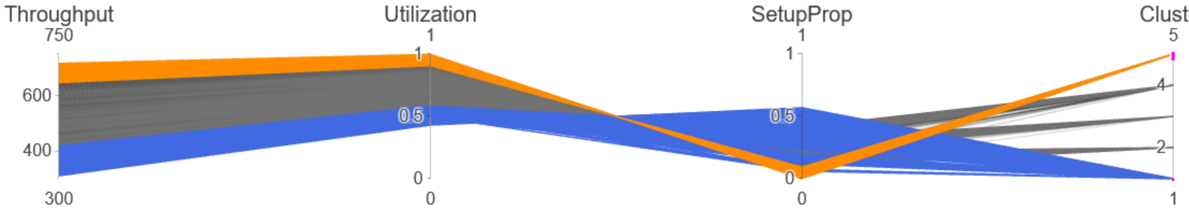


Figure 5: Parallel coordinates visualization of the clustering based on three outputs. Each vertical axis represents one standardized output.

4.2 Results

We applied our presented XAI-approach for investigating the relation between the factors of the simulation model and the corresponding cluster allocation. For this purpose, we used a multilayer neural network and a random forest classification. For the neural network, we used an implementation with two hidden layers, a 10 node input layer and a 5 node output layer using softmax activation. For one-vs-rest-classifications, a one node output layer using sigmoid activation was used. For the random forest classification, a forest using 500 trees was implemented. No notable difference in accuracy and results was perceived between those algorithms, so the results presented here are based on the neural network. For the first step involving cluster-independent feature relevance evaluation, we used the output layer for multiclass classification with 5 neurons. Feature relevance using permutation feature importance is shown in Figure 6.

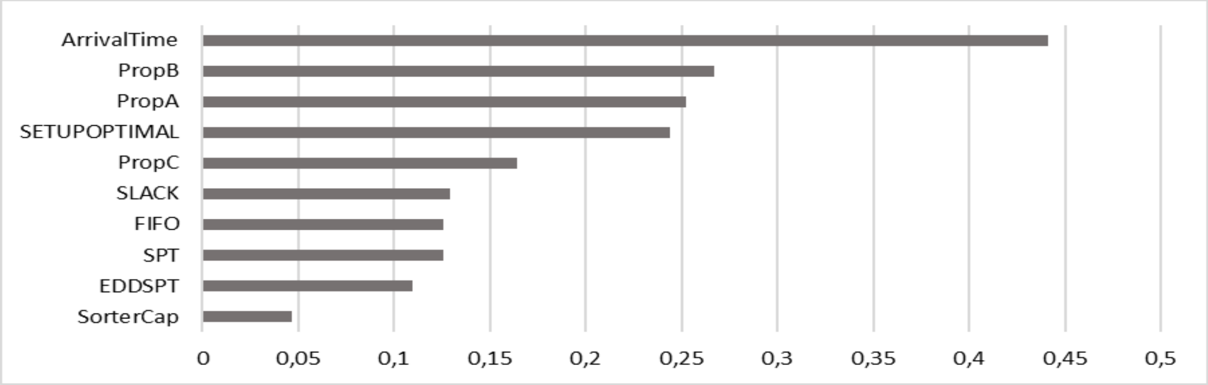


Figure 6: Results of the feature relevance evaluation using permutation feature importance.

The results show that the most relevant factor for cluster allocation in general is the interarrival time, having almost double the importance of the runner-up, followed by the proportions of the product types and the setup optimal sorting strategy. In the next step, we investigate the feature importance per cluster. More specifically, we compare the feature importance of factors for the good performance cluster (left side) vs. the bad performance cluster (right side) using partial dependence plots (PD-plots), as shown in Figure 7. The subplots show the value of each corresponding factor on x-axis, while the partial dependence, representing the importance of that factor value for the specific cluster allocation, is shown on the y-axis of each subplot. Looking at the feature importance for the good performance cluster, we see overall low contribution values for individual factors compared to bad performance clusters. Therefore, we can assume that an allocation to the good performance cluster stems from a combination of factor values rather than a single dominating factor. The highest contribution comes from the factor interarrival time (*ArrivalTime*). We see that low values of interarrival time contribute to the good performance, and that this contribution drops drastically when above 100 seconds. We can see a contrasting behavior in the bad performance cluster for interarrival times greater than 200 seconds, but with a much higher magnitude of contribution (up to 1.0

compared to up to 0.15). We also see a mirrored contribution of feature importance for the setup optimal sorting strategy in contrast to the other sorting strategies in the good performance cluster.

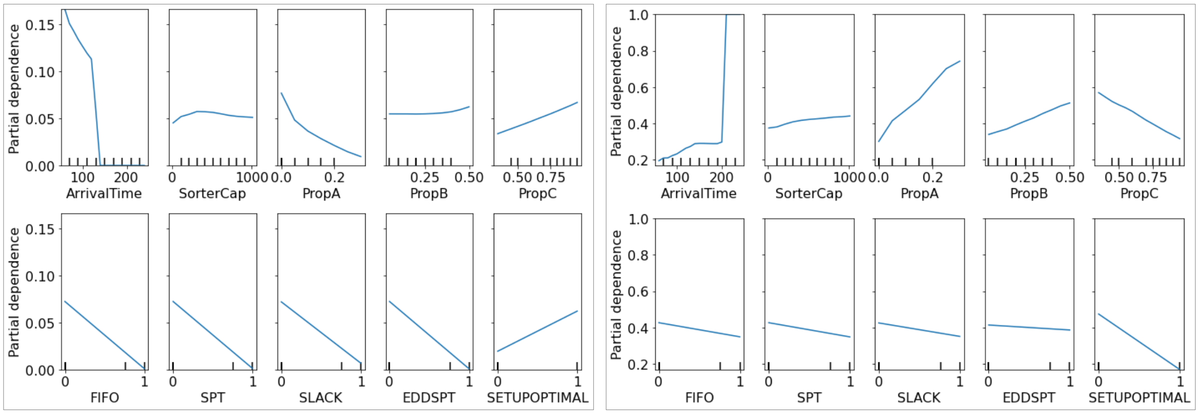


Figure 7: PD-plots for good performance cluster (left) vs. bad performance cluster (right).

That means that setup optimal sorting strategy contributes to a good performance cluster allocation, while simultaneously all other sorting strategies decrease the probability of an experiment being allocated to the good performance cluster. Therefore, a high frequency of jobs entering the system is a prerequisite for good system performance and on top of that, a setup optimal sorting strategy is needed in order to avoid bottlenecks, considering that Figure 7 also shows that the proportion of individual products in the mix seems to contribute to the system performance.

The next step was to use a more sophisticated explanation method than simple partial dependence plots. For demonstration purposes, we used the SHAP-package for computation of so-called SHAP-Values. The concept of SHAP-Values was adapted from game theory, where the Shapley-value quantifies the contribution of each player to the outcome of a game. The SHAP-Value for XAI picks up this idea by treating every factor of a black-box model as an individual player, thus calculating their contribution to the overall outcome of a prediction (representing the outcome of one game). This is done by aggregating the marginal contribution of each factor in every possible combination with the other factors present. This makes the calculation of SHAP-Values very costly in terms of computation time, because computation time grows exponentially with the number of factors (Lundberg et al. 2020; Lundberg and Lee 2017). In our example, as shown in Figure 8, we again resort to one-vs-rest classification, so an output of 1 indicates a classification of the corresponding cluster (good or bad performance), while an output of 0 indicates a classification towards not being in those clusters. Each line represents one classified simulation experiment.

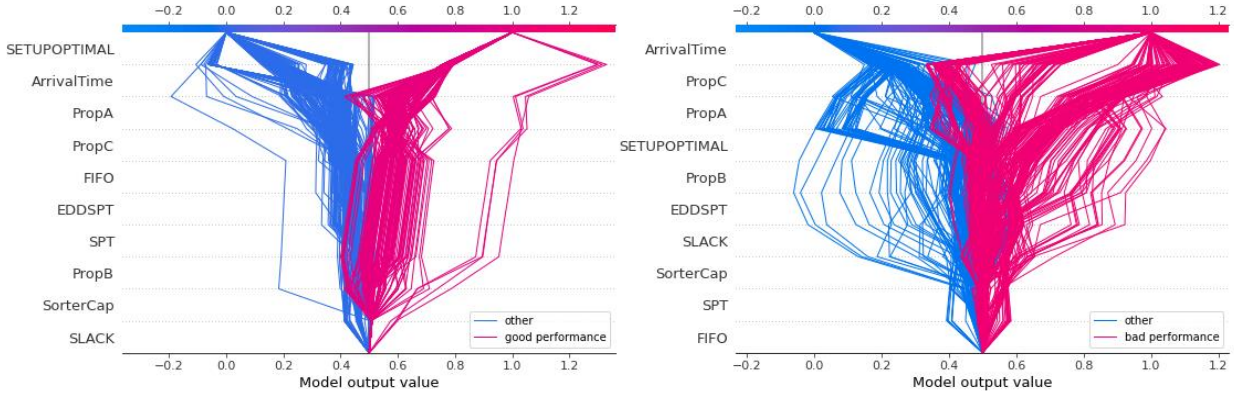


Figure 8: SHAP-Values plot for good performance cluster (left) and bad performance cluster (right).

The individual SHAP-values for the factors can be interpreted relative to the base effect, comparable to how we would interpret coefficients relative to the intercept in a linear regression model. The base effect represents the average prediction in a model without any factors, which would be 0.5 in our case. The figure shows a comparison of the classification explanation of samples between the good performance cluster (on the left side) and the bad performance cluster (on the right side). The factors are ordered according to their importance for the classification from top to bottom, and the individual value show their contribution to the classification output. Since this method is very computationally intensive, only a small random subsample of data could be considered, therefore results have to be taken with a little grain of salt. However, since the response surface is very dense (due to large experiment design) compared to model simplicity, results can still be considered as reasonable. This sample of ~1000 experiments already took 24h worth of computation time on a high-end consumer-grade machine. We can see that while for the bad performance cluster allocation, interarrival time indeed is the most contributing factor. For the good performance it is actually the setup optimal sorting strategy. This makes sense since we already assumed that a high frequency of arriving jobs alone may lead to bottlenecking which in turn decreases system performance, instead of increasing it. So regarding the contribution towards good performance, interarrival time and setup optimal sorting have mutually the highest contribution values. There are even some outliers that exhibit an extremely high contribution of those two factors compared to the majority of predictions, which is because they get dragged to the right due to high contributions of the factors *PropA* and *PropC*, which are the factors that determines the individual proportion of product types A and C in the mix. We could already see the importance of *PropA* in Figure 7, that indicated a higher importance of the *PropA* factor towards values near 0, and vice-versa for *PropC*. For those outlier predictions on the right side in the good performance plot, we can also determine a high contribution of the sorting strategies earliest due date (*EDDSPT*) and shortest processing time (*SPT*). This is presumably due to the one-hot-encoding of the factor sorting strategy, that turns one categorical factor into multiple binary factors. That means having the setup optimal sorting strategy in place automatically boosts the importance of not having any of the other sorting strategies active, or at least not having the *EDDSPT* or *SPT* sorting strategies contributing against good system performance.

Another very useful plot from the SHAP-package is the so-called force plot. This plot shows the contribution of factors as a stacked surface for each prediction along the values of a selected factor. Figure 9 demonstrates this using the factor interarrival time. The top plot shows the predictions for the good performance cluster, the bottom plot shows the predictions for the bad performance cluster. Similar to the previous plot, red indicates a positive classification and vice versa for blue.

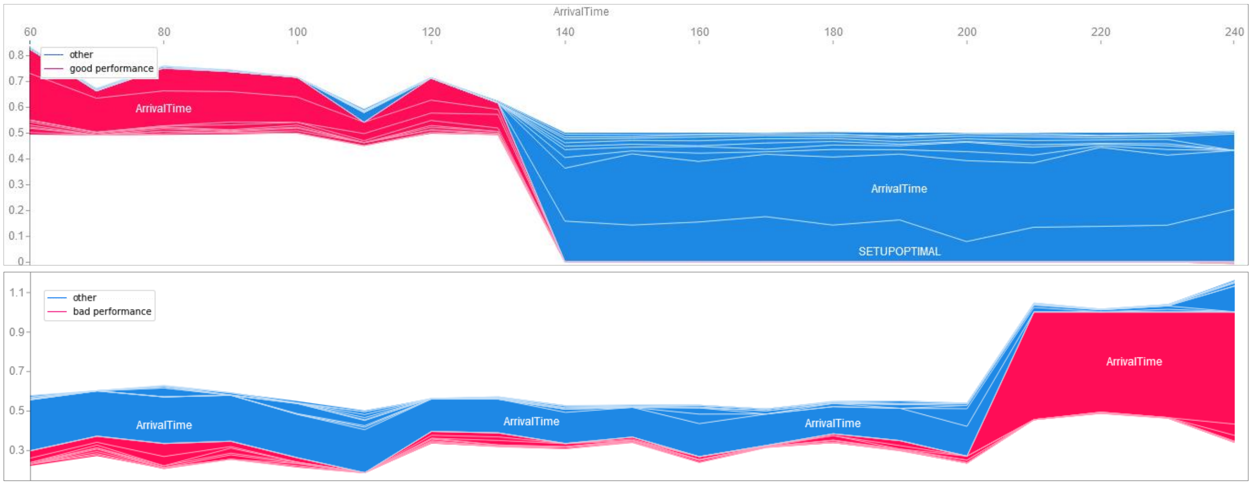
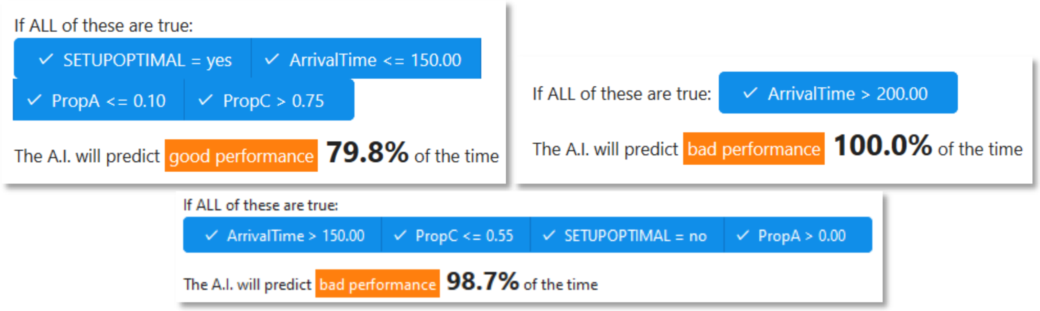


Figure 9: Force plots for analyzing contribution of factors along the values of interarrival time. Top shows predictions for the good performance cluster, bottom shows the predictions for the bad performance cluster.

For the explanation of good performance in the top plot, we can clearly see that interarrival time is the dominating factor up to values of ~130 seconds. Above this threshold, most predictions get dragged towards zero by interarrival time in combination with the setup optimal sorting strategy (or lack thereof, respectively). In the bad performance cluster, interarrival time is the single most contributing factor along all the complete range of all values. Starting at 200s interarrival time, predictions get dragged drastically towards 1, so that most predictions in the value range are classified towards the bad performance cluster. However, some few samples are predicted red even in the low values of interarrival time, supporting the assumption that low interarrival time alone can still lead to bad system performance when bottlenecking.

Finally, we did a local prediction explanation using the medoids of the good and bad performance clusters, respectively, as shown in Figure 10 (top left and right). This was done using the Anchors-Package (Ribeiro et al. 2018).



If ALL of these are true:
✓ SETUPOPTIMAL = yes ✓ ArrivalTime <= 150.00 ✓ PropA <= 0.10 ✓ PropC > 0.75
The A.I. will predict good performance **79.8%** of the time

If ALL of these are true: ✓ ArrivalTime > 200.00
The A.I. will predict bad performance **100.0%** of the time

If ALL of these are true:
✓ ArrivalTime > 150.00 ✓ PropC <= 0.55 ✓ SETUPOPTIMAL = no ✓ PropA > 0.00
The A.I. will predict bad performance **98.7%** of the time

Figure 10: Local cluster medoid (top left and right) and boundary point (bottom) explanation using Anchors.

As already assumed, the prediction rule for the good performance is more complex than for the bad performance. Overall, the rules provided by this method are somewhat obvious findings considering a low complexity, single line system. Our goal however was to demonstrate the approach in general, being able to use black-box classifiers in combination with XAI-method in order to gain insight into the system. We only see the prediction of good and bad performance between values of inter arrival time for <= 150 seconds and > 200 seconds, respectively. This is due to the fact that we only look at the cluster center using local explanation techniques. Especially regarding the bad performance cluster, for simulation experiments that are located more towards the boundary of the cluster and exhibit inter arrival times smaller than 200 seconds, the derived rules are more complex. This is where for example the absence of the setup optimal sorting becomes relevant for the system performance. When picking another point away from the cluster medoid for local explanation with an interarrival time of 180s, we see this assumption confirmed, as shown in the bottom box of Figure 10. Note that this point belongs to the bad performance cluster, because it's performance still is closer to the bad performance centroid than to any of the other clusters. However, the underlying rule to get to this cluster allocation via factor values is different compared the cluster medoid. That means that when extracting rules using local explanation techniques, one should keep in mind that the produced rule is not necessarily exclusive for reaching the target classification

5 CONCLUSION AND FUTURE WORK

In this paper, we demonstrated how methods of explainable artificial intelligence can be used for the output analysis of Data Farming projects and presented a workflow for the application of those. By enabling the application of black-box classifiers, this opens up a whole new range of analysis methods and options and can even contribute to a way for an automated or assisted analysis of simulation data using AI in future work. Although options for finding hidden knowledge in a simple single-lane case study model is limited, we still were able to extract reasonable knowledge using this method. Utilizing modern, big data analytics methods have always been a research goal of Data Farming. This paper gets in line with this research by enabling the use of XAI methods, extending the portfolio of Data Farming output analysis methods while also contributing to narrowing the gap between simulation and AI research. Future work should investigate the potential of this approach in more complex case studies using large simulation models with more complex behavior and even larger response surfaces. In addition to that, the actual application of black-box algorithms should also be addressed in future work in more detail. In the presented case study, we narrowed our investigation to an artificial neural network and random forest. In our relatively simple simulation model, no difference in accuracy was perceived as expected. This should be extended in future work by investigating different types of black-box-classification algorithms, their configuration, hyper-parametrization and performance in association with different types of simulation models and simulation model complexity. Furthermore, we focused on a handful of the most frequently used XAI methods. For example, counterfactual explanation method were not considered, but application of those could be investigated in future work. XAI is a very popular topic, new methods emerge frequently, so that the portfolio of applicable methods can be continuously extended.

REFERENCES

Adadi, A., and M. Berrada. 2018. “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)”. *IEEE Access* 6:52138–52160.

Altmann, A., L. Toloşi, O. Sander, and T. Lengauer. 2010. “Permutation Importance: A Corrected Feature Importance Measure”. *Bioinformatics* 26(10):1340–1347.

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. 2020. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. *Information Fusion* 58:82–115.

Belle, V., and I. Papantonis. 2020. “Principles and Practice of Explainable Machine Learning”, arXiv preprint. http://arxiv.org/pdf/2009.11698v1.

Dosilovic, F. K., M. Brcic, and N. Hlupic. 2018. “Explainable Artificial Intelligence: A Survey”. In *41st International Convention on Information and Communication Technology, Electronics and Microelectronics,* edited by K. Skala, 210–215. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Edwards, L., and M. Veale. 2017. “Slave to the Algorithm? Why a Right to Explanation is Probably Not the Remedy You are Looking for”. *SSRN Electronic Journal* 16(1):18–84.

Feldkamp, N., S. Bergmann, and S. Strassburger. 2020. “Knowledge Discovery in Simulation Data”. *ACM Transactions on Modeling and Computer Simulation* 30(4):1–25.

Feldkamp, N., S. Bergmann, and S. Strassburger. 2015. “Knowledge Discovery in Manufacturing Simulations”. In *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation,* edited by S. J. E. Taylor, N. Mustafee, and Y.-J. Son, 3–12. New York, New York: ACM Press.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin. 2015. “Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation”. *Journal of Computational and Graphical Statistics* 24(1):44–65.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. 2019. “A Survey of Methods for Explaining Black Box Models”. *ACM Computing Surveys* 51(5):1–42.

Hawkins, D. M. 2004. “The Problem of Overfitting”. *Journal of chemical information and computer sciences* 44(1):1–12.

Horne, G. E., and T. E. Meyer. 2005. “Data Farming: Discovering Surprise”. In *Proceedings of the 2005 Winter Simulation Conference,* edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1082–1087. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Horne, G. E., and K.-P. Schwierz. 2008. "Data Farming Around The World Overview". In *Proceedings of the 2008 Winter Simulation Conference,* edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1442–1447. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kleijnen, J. P.C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. “State-of-the-Art Review: A User’s Guide to the Brave New World of Designing Simulation Experiments”. *INFORMS Journal on Computing* 17(3):263–289.

Law, A. M. 2003. “How to Conduct a Successful Simulation Study”. In *Proceedings of the 2003 Winter Simulation Conference,* edited by S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 66–70. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Lechler, T., M. Sjarov, and J. Franke. 2021. “Data Farming in Production Systems - A Review on Potentials, Challenges and Exemplary Applications”. *Procedia CIRP* 96:230–235.

Lempert, R. J., S. W. Popper, and S. C. Bankes. 2003. "Shaping the Next One Hundred Years: New Methods for Quantitative, Long-Term Policy Analysis". RAND Technical Report MR 1626 RPC, Santa Monica, CA, USA.

Lin, Y.-S., W.-C. 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.

Lucas, T. W., W. D. Kelton, P. J. Sánchez, S. M. Sanchez, and B. L. Anderson. 2015. "Changing the paradigm: Simulation, now a method of first resort". *Naval Research Logistics (NRL)* 62(4):293–303.

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

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. 2020. "From local explanations to global understanding with explainable AI for trees". *Nature Machine Intelligence* 2(1):2522–5839.

MacDonald, C., and E. A. Gunn. 2012. "Allocation of Simulation Effort For Neural Network Vs. Regression Metamodels". In *Proceedings of the 2012 Winter Simulation Conference (WSC 2012),* edited by C. Laroque, R. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Molnar, C. 2019. *Interpretable Machine Learning*: *A Guide for Making Black Box Models Explainable.* 1st ed. Zürich: Lulu Com.

Morocho-Cayamcela, M. E., H. Lee, and W. Lim. 2019. "Machine Learning for 5G/B5G Mobile and Wireless Communications: Potential, Limitations, and Future Directions". *IEEE Access* 7:137184–137206.

Mustafee, N., and J. H. Powell. 2018. "From Hybrid Simulation to Hybrid Systems Modelling". In *Proceedings of the 2018 Winter Simulation Conference,* edited by M. Rabe, A. A. Juan, N. Mustafee, and A. Skoogh, 1430–1439. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Painter, M. K., M. Erraguntla, G. L. Hogg, and B. Beachkofski. 2006. "Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optimized Aircraft Engine Fleet Management". In *Proceedings of the 2006 Winter Simulation Conference,* edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Ras, G., M. van Gerven, and P. Haselager. 2018. "Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges". In *Explainable and Interpretable Models in Computer Vision and Machine Learning,* edited by H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, 19–36. Cham: Springer International Publishing.

Ribeiro, M. T., S. Singh, and C. Guestrin. 2018. "Anchors: High-Precision Model-Agnostic Explanations". In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,* edited by S. Zilberstein, S. McIlraith, and K. Weinberger. Palo Alto, CA: AAAI Press.

Ribeiro, M. T., S. Singh, and C. Guestrin. 2016. ""Why Should I Trust You?"". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* edited by B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, and R. Rastogi, 1135–1144. New York, NY, USA: ACM.

Sanchez, S., and P. J. Sanchez. 2017. "Better Big Data via Data Farming Experiments". In *Advances in Modeling and Simulation*: *Seminal Research from 50 Years of Winter Simulation Conferences,* edited by A. Tolk, J. Fowler, G. Shao, and E. Yücesan, 159–179. 1st ed. Springer International Publishing.

Sanchez, S. M. 2014. "Simulation Experiments: Better Data, Not Just Big Data". In *Proceedings of the 2014 Winter Simulation Conference,* edited by A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 805–816. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Stepin, I., J. M. Alonso, A. Catala, and M. Pereira-Farina. 2021. "A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence". *IEEE Access* 9:11974–12001.

Strassburger, S., S. Bergmann, N. Feldkamp, K. Sokoll, and M. Clausing. 2018. "Data Farming Research Project with Audi and VW". In *2018 Plant Simulation Worldwide User Conference,* October 16th -18th, Stuttgart, Germany.

Tjoa, E., and C. Guan. 2020. "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". *IEEE Transactions on Neural Networks and Learning Systems*:1–21.

Tolk, A., A. Harper, and N. Mustafee. 2021. "Hybrid Models as Transdisciplinary Research Enablers". *European Journal of Operational Research* 291(3):1075–1090.

AUTHOR BIOGRAPHIES

NICLAS FELDKAMP is a research associate in the Group for Information Technology in Production and Logistics at the Ilmenau University of Technology. He received his Ph.D in business information systems from the Ilmenau University of Technology, and holds M.Sc. from the Ilmenau University of Technology and B.Sc. from the University of Cologne. His research interests include data science, business analytics, and industrial simulations. His email address is niclas.feldkamp@tu-ilmenau.de.