Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

AN APPLICATION OF AUTOMATED MACHINE LEARNING WITHIN A DATA FARMING PROCESS

Lynne Serré
Maude Amyot-Bourgeois

Defence Research and Development Canada
Department of National Defence
60 Moodie Drive
Ottawa, ON K1A 0K2, CANADA

ABSTRACT

Data farming is a simulation-based methodology used within the defense community to analyze complex systems and provide insights to decision makers. It can produce very large, multi-dimensional data sets that require sophisticated analysis tools, such as metamodeling. Advances in explainable artificial intelligence have expanded the types of metamodels that can be considered; however, constructing a well-fitting machine learning metamodel involves many tasks that can become time consuming for an analyst. Automated machine learning (autoML) can save an analyst time by automating metamodel training, tuning and testing. Using outputs of an agent-based simulation of a military ground-based air defense scenario, we compared the performance of metamodels trained using autoML and different experimental designs. We found that autoML can reasonably automate the construction of metamodels and adds robustness to the analysis by considering multiple types of metamodels; however, the type and size of experimental design can significantly impact metamodel performance.

1 INTRODUCTION

Decision makers within the defense community often need to understand complex systems involving large sets of uncertain factors (Horne et al. 2018). Real-world experiments of these complex systems are not always possible; for instance, in a procurement process, it may be cost-prohibitive to acquire all options for testing purposes. In other situations, the decisions may pertain to systems not yet developed or fielded. Simulation models can help analysts and decision makers develop a basic understanding of a system, discover robust options, and compare possible outcomes of those options (Kleijnen et al. 2005).

Data farming is a methodology developed within the defense community that aims to improve the understanding of the many possibilities facing decision makers by running large-scale, efficiently-designed simulation experiments (Horne et al. 2018). It is a collaborative and iterative process consisting of five building blocks: rapid scenario prototyping, model development, design of experiments, high performance computing, and analysis and visualization; details on the data farming methodology can be found in Horne et al. (2014) while a recent overview of current data farming capabilities can be found in Sanchez (2020). The data farming process can generate large volumes of multi-dimensional data that require sophisticated analytical techniques in order to highlight useful information, extract conclusions and support decision-making (Horne et al. 2014). Often, multiple techniques are needed to fully exploit the data (Horne et al. 2014; Sanchez 2020), including the construction of metamodels. As defined in Kleijnen and Sargent (2000), a "metamodel is an approximation of the input/output (I/O) transformation that is implied by the simulation model." Metamodels are helpful because they can promote understanding (Sanchez 2020); for instance, the

978-1-6654-7661-4/22/$31.00 ©2022 Crown

functional form of a metamodel (e.g., a low-order polynomial model) can provide insights into how a simulation output changes as the simulation inputs change (e.g., in a linear or non-linear manner). Metamodels can also help assess which simulation inputs are key drivers for a simulation output (Sanchez 2020).

Common examples of metamodels in past military applications of data farming include polynomial regression models, logistic regression models, and decision trees (Kleijnen et al. 2005; Lucas et al. 2007; Kallfass and Schlaak 2012; Sanchez and Wan 2015; Hill et al. 2019; Kesler et al. 2019). These types of models are often described as interpretable or white-box models: it is possible to study their internal mapping of the I/O relations, which can then be used to infer knowledge about the modeled system (Feldkamp et al. 2020; Feldkamp 2021). Many machine learning models, such as deep learning or ensemble models, are often described as black-box models. They have been shown to achieve higher prediction accuracy than white-box models, creating a trade-off between interpretability and accuracy (Lundberg and Lee 2017). In response to this trade-off, the field of explainable artificial intelligence (XAI) has emerged, leading to the development of methods aimed at making black-box models transparent (Feldkamp 2021). XAI comprises a broad range of methods, some examples include permutation feature importance and SHapley Additive exPlanations (SHAP) (Feldkamp 2021). Recent military applications of these methods within a data farming context can be found in Amyot-Bourgeois et al. (2021) and Serré et al. (2021). Feldkamp (2021) proposed a workflow for incorporating XAI methods into the output analysis of the data farming process.

The application of machine learning models in combination with XAI methods, as noted in Feldkamp (2021), opens up a whole new range of techniques that can be applied to build and interpret metamodels of farmed data. Building machine learning models involves many tasks, such as choosing a type of model or family of models, tuning model hyperparameters, and evaluating model performance. This can become a time consuming process, especially within an iterative process like data farming where a series of data sets may be generated as the experiment grows and evolves. Automated machine learning (autoML) refers to tools that automate some or all machine learning tasks with the goal of making the practice of machine learning more systematic and more efficient (Ghahramani 2019). Open source examples of autoML tools include Auto-Sklearn (Feurer et al. 2021), H2O AutoML (LeDell and Poirier 2020) and the Tree-based Pipeline Optimization Tool (Le et al. 2020).

AutoML provides an opportunity to further expand the set of tools available for output analysis and can play a key role in enabling XAI within data farming. However, when building metamodels within a data farming process, consideration must also be given to the experimental design. This has been described as a chicken-and-egg problem (Kleijnen et al. 2005): the type of metamodels considered depend on the experimental design and vice versa. While guidance on an appropriate choice of experimental design for white-box metamodels is available in the literature, it appears less guidance is available for black-box metamodels. Therefore, the objective of our paper is two-fold. Firstly, it seeks to increase the efficiency of metamodel construction through autoML. Secondly, it contributes to guidance on the choice of experimental design for black-box metamodels by undertaking a multi-model, multi-design comparison that also considers the trade-off between repetition and coverage in an experimental design.

2 BACKGROUND

Simulation metamodeling is a process that involves choosing an experimental design, the type and form of metamodel, and a validation strategy to assess the metamodel (Meckesheimer et al. 2002). Kleijnen and Sargent (2000) break down this process into ten steps and emphasize the importance of identifying the goal of the metamodel as part of the process. In this paper, our focus is on building predictive metamodels that can be used to identify which simulation inputs are key drivers for a simulation output.

2.1 Experimental Designs

In a synthetic environment, experimentalists have more, or sometimes perfect, control over the parameters defined as inputs to a simulation model. A subset of these parameters may be identified as variables of interests, or factors, likely to influence the outcome of the simulation. These factors can take on multiple values, a single combination of which is called a design point (DP), and a set of DPs is called the design of experiment (DoE) (Keijnen et al. 2005). Devising efficient DoEs is an important element of a good investigation in a synthetic environment. Key considerations for DoEs are discussed in Kleijnen et al. (2005) and Sanchez (2020), examples include an acceptable trade-off between the number of DPs and the simulation run time, a low correlation between factors to facilitate the identification of each factor's individual contribution to the outcome, good space-filling properties, and a simple generation method that accommodates different types of factors (e.g., categorical, discrete, continuous). A DoE should also assist in answering the objectives of the experiment, be it initial exploration, identifying the relevant factors, or optimizing the outcome, as a few examples. Short descriptions of some common DoEs are provided below; additional designs and a more in-depth discussion can be found in Kleijnen et al. (2005).

A full factorial (FF) design is a simple DoE where each factor is divided into a number of possible values, called levels, and each possible combination of levels forms a DP. For example, a two-level FF design assigns a minimum and maximum value for each of the N factors, giving a total possible number of combinations and DPs of 2^N. Building on this example, a three-level FF design adds a center point to the possible factor levels, which increases the number of DPs to 3^N. Thus, the number of DPs grows very quickly. Fractional factorial designs can be used to reduce the number of DPs, but at the expense of hiding some features of the response such as possible interaction effects (NIST/SEMATECH 2013a).

A random sampling (RS) design is generated by randomly selecting a subset of the possible values for each factor while taking into account the continuous or discrete nature of the variable. Using this technique, there is a risk of having an unbalanced design with clusters and empty regions within the design space, as mentioned in Pereda et al. (2017). To minimize this risk, one possible solution is to divide the range of values for each factor into sub-intervals or strata and to randomly sample from within each strata; this is known as stratified random sampling in survey methodology.

A central composite (CC) design separates the range of each factor into five levels, allowing one to observe more complex responses in the output such as the main effects, interactions and quadratic effects, as noted in Sanchez and Wan (2015). Whereas a FF design with five levels would generate a high number of DPs, the CC design generates a smaller and smarter data set by combining a two-level factorial design with center points and "star points" (representing extreme values), greatly reducing the number of DPs needed in the process. Additional details can be found in Sanchez and Wan (2015) and NIST/SEMATECH (2013a); Alam et al. (2004) gives a detailed description of a CC design with four factors.

Latin hypercube (LH), as defined in Sanchez and Wan (2015), is a DoE where the factors are gridded equally into a number of levels n. Organizing the set of DPs into a design matrix where the rows are DPs and the columns are factors, a LH design permutes all possible n levels for each column such that each possible factor value appears only once. Building on the LH design, nearly orthogonal Latin hypercube (NOLH) is a DoE that can achieve a degree of space-filling similar to the finer grids of FF designs with much fewer DPs, as pointed out in Sanchez and Wan (2015). The objectives of using a NOLH design in complex simulations, as stated in Cioppa and Lucas (2007), include the following: the ability to handle a large number of factors sampled in an almost uncorrelated sequence (i.e., in the design matrix, the columns are nearly orthogonal), the ability to extract complex models from the output while maintaining a fixed number of DPs, and to obtain a good space-filling design (i.e., where the DPs are dispersed across the full range of the experimental region, minimizing clusters and empty spaces).

Four of the above-mentioned common DoEs – FF, RS, CC and NOLH – will be considered in our study as they are relatively simple to generate (either directly or through software) but have different space-filling properties. Figure 1 illustrates the different space-filling properties of the four chosen designs by projecting each design onto a single pair of factors where the total number of DPs is similar.

2.2 Curse of Dimensionality

As the number of factors increases, it becomes more evident that using an efficient DoE is a necessary step for conducting the experiment: by selecting an inefficient design for a scenario exploring dozens of factors, the number of DPs can quickly increase to billions, and the associated total simulation run time can increase to years or decades. This has been referred to as the "curse of dimensionality" in Sanchez (2020). Another aspect of stochastic simulations that affects the simulation run time is the number of replications per DP. For stochastic simulations, a large enough sample size of replicated runs is necessary to obtain a valid distribution of outcomes, but as the number of replications increases, so does the simulation run time.

The question "how big is big enough" has been asked more broadly in the field of statistics such as in Lenth (2001) and NIST/SEMATECH (2013b). More specifically, in the field of experimental simulation, the trade-off between the number of DPs and the number of replications has been studied in terms of its impact on metamodel precision, examples include Santos and Santos (2009) and MacDonald and Gunn (2012). Together, these papers show that the impact of this trade-off on the performance of the metamodel depends on the type and form of metamodel. Given this dependency, in our analysis, we will consider two different numbers of replications (20 and 100) to examine the impact on metamodel precision.

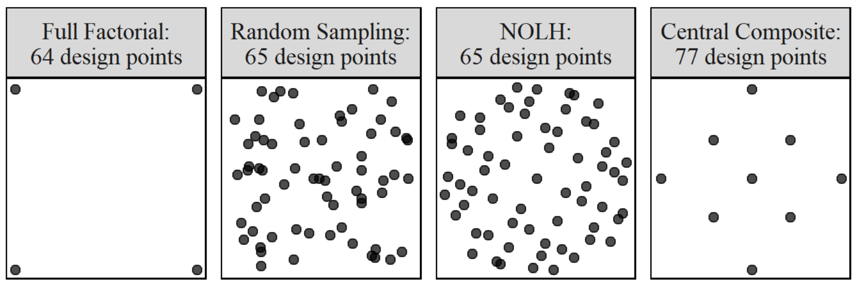

Figure 1: Projections of different types of experimental designs onto a single pair of factors.

2.3 Machine Learning Metamodels

De Reus et al. (2018) proposed several possibilities of how artificial intelligence techniques could be used to enhance the data farming process, which included the use of machine learning techniques to support metamodeling. Some examples of machine learning techniques that are generally viewed as black-box models include neural networks, random forests, gradient boosted trees, support vector machines, and nearest-neighbor methods; descriptions of these techniques can be found in Hastie et al. (2009).

Several examples of neural network simulation metamodels can be found in the literature. Using stochastic simulation models of manufacturing systems, Hurrion and Birgil (1999) showed that "neural network metamodels using a randomised experimental design produce more accurate and efficient metamodels than those produced by similar sized factorial designs with either regression or neural networks." Using a deterministic combat simulation model, Alam et al. (2004) studied the impact of different experimental designs on the predictive accuracy of a neural network metamodel. They found that a modified LH design, as compared to FF, RS and CC designs of the same size, produced the best performance. Using stochastic queuing and inventory system simulation models, MacDonald and Gunn (2012) examined the trade-off between a larger number of DPs with fewer replications and a smaller number of DPs with greater replications in the context of constructing neural network metamodels. In contrast to polynomial regression metamodels, their results suggest that the number of replications at each DP can be sacrificed in favor of good spatial coverage when training neural network metamodels.

De la Fuente and Smith (2017) conducted a literature review of the most applied types of simulation metamodels in the context of engineering problems. They concluded, based on their review, that support vector regression, neural networks and Gaussian processes are generally stable and reliable techniques while random forests and boosted trees are not commonly used in simulation metamodeling. Using a systems dynamic model of a hospital, they compared the performance of these five types of metamodels. They considered three evaluation criteria: fit quality, fitting time, and interpretability; models stronger in one criteria were generally found to be weaker in the others. Using a stochastic simulation model of a military operation, Amyot-Bourgeois et al. (2021) considered random forests and *k*-nearest neighbor as metamodels, which were found to have similar performance scores. Using a single-server simulation model, Feldkamp (2021) considered random forests and neural networks, which were also found to have similar performance scores. In Feldkamp (2021), as well as De la Fuente and Smith (2017) and Amyot-Bourgeois et al. (2021), only a single experimental design was considered.

From the above-mentioned studies, it is difficult to conclude which DoEs should be used for machine learning metamodels because the studies tend to compare a single metamodel with multiple designs or multiple metamodels with a single design. Our study considers multiple metamodels with multiple designs and therefore offers a more systematic look at the impact of the choice of experimental design on the performance of the metamodel.

2.4 Metamodel Validation

Kleijnen and Sargent (2000) define validation as the "substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model." In the context of simulation metamodels, validation should be considered with respect to both the system being modeled and the simulation model (Kleijnen and Sargent 2000); in this paper, we focus on the latter type of validation. In evaluating machine learning models, performance is often assessed on an independent data set (i.e., test set) not used to build the model (i.e., training set). It is also common to split the training set for the purposes of model selection (e.g., hyperparameter tuning), which can be done through K-fold cross-validation. In K-fold cross-validation, the data set is divided into roughly K equal-sized subsets. Each subset is used as a test (or validation) set while the model is trained on the other $K-1$ subsets. The process is then repeated K times and the prediction error estimates are combined; details on this method as well as a more in-depth discussion on machine learning model assessment can be found in Hastie et al. (2009, Chapter 7). These methods mirror those that have been proposed for validating metamodels with respect to the simulation model. For instance, Kleijnen and Sargent (2000) discuss the use of test sets while Meckesheimer et al. (2002) propose the use of cross-validation. In the comparative studies summarized in the previous section, Hurrion and Birgil (1999), Alam et al. (2004) and Amyot-Bourgeois et al. (2021) used independent test sets. The validation strategy used in this study is closest to Hurrion and Birgil (1999): an independent, randomly chosen set of 10,000 unreplicated DPs was used as a test set. De la Fuente and Smith (2017) used cross-validation for hyperparameter optimization; this strategy is also used in our study to tune the machine learning metamodels.

3 METHOD

3.1 Simulation Model

The scenario investigated here is the point defense of a tactical ground-based air defense (GBAD) system against airborne threats. The air defense scenario is implemented in a synthetic, agent-based environment called Map-Aware Non-uniform Automata (MANA) that was developed by the New Zealand Defence Technology Agency (Anderson 2013). It has been used as the modeling environment in past military applications of data farming and found to have "good traits" for model development within a data farming process (Horne et al. 2014). Further, we have developed Python scripts that automate the generation of MANA input files for each DP and facilitate running MANA in batch mode on a high-performance computing system (Serré et al. 2021). Both the GBAD capability and the airborne threats modeled do not correspond to specific existing systems, instead they were kept generic. This is to allow the exploration of a wide range of possible values for the parameters defining the two entities. The parameters selected as factors of interest are shown in Table 1. The output of interest, sometimes called a measure of effectiveness (MOE), records whether all incoming threats were killed. This MOE is known as raid negation and takes on a value of one if all airborne threats were successfully intercepted by the GBAD system and zero otherwise.

Table 1: Factors of interest parameterized in the simulation model of the point defense of a tactical ground-based air defense (GBAD) system against airborne threats.

Factors of interest		Range of values
GBAD-related factors	System (single shot) kill probability System delay between two engagements System (engagement/kill) range System ammunition load	0.1 to 0.99 1 to 10 seconds 0.25 to 45 kilometers 30 to 90 ammunitions
Threat-related factors	Number of threats Threat speed	1 to 30 threats 200 to 800 meters/second

3.2 Experimental Designs

Table 2 provides a summary of the selected designs (FF, CC, NOLH, and RS) and their respective number of DPs. For every design, each DP was replicated 100 times to account for the stochastic variation in the simulation model. For the FF design, the number of DPs is determined by the number of factors and the number of factor levels. In this study, there are six factors and we considered two, three or four levels per factor. For the CC design, the number of DPs is determined only by the number of factors; therefore, only a single size of CC design was considered. The DPs were generated following the details in NIST/SEMATECH (2013c) for an inscribed CC design. For the NOLH design, the number of DPs considered was determined based on the worksheets available from the SEED Center for Data Farming at the Naval Postgraduate School (Sanchez 2011). Lastly, the RS design is perhaps the most flexible in terms of the number of DPs. DPs were randomly sampled from the intervals shown in Table 1 while accounting for the desired number of decimals (set by MANA input requirements). As noted in Section 2.1, while there is a risk of the RS design having clusters and empty spaces, it is included here as a benchmark against the other designs of the same size. However, only a single RS design was generated for each number of DPs in Table 2. This is a potential limitation of the study.

Table 2: Experimental designs considered and their respective number of design points.

Random Sampling (RS)	Full Factorial (FF)	Central Composite (CC)	Near-Orthogonal Latin Hypercube (NOLH)
17	-	-	17
65	64 (2 levels)	-	65
77	-	77	-
129	-	-	129
257	-	-	257
729	729 (3 levels)	-	725
4096	4096 (4 levels)	-	-

3.3 Metamodel Training using AutoML

Based on the findings of a benchmarking study of five autoML tools by Ebadi et al. (2019), this study uses an open source tool called H2O that has programming interfaces in several languages, including R and Python. H2O AutoML simplifies the "training and tuning of machine learning models by offering a single function to replace a process that would typically require many lines of code" (LeDell and Poirier, 2020). This function has only three required parameters: the response column, the training data, and a stopping strategy. In this study, the response column is the MOE, raid negation, which is a binary variable. The different sizes of experimental designs are used as training data. To explore the trade-off between a greater number of DPs with fewer replications and a smaller number of DPs with greater replications, each data set of 100 replications was randomly divided into five data sets each with 20 replications; this resulted in a total of 96 training sets. A maximum of 20 models was specified as the stopping strategy. In addition to the required parameters, two optional parameters were also specified. Firstly, the same random seed was specified for all data sets. However, as noted in the H2O documentation, setting a random seed does not guarantee reproducibility for deep neural nets, which are not reproducible by default for performance reasons (H2O.ai 2022). Secondly, stacked ensemble models were excluded because feature importance methods are not available in H2O for these types of models. Therefore, the following models were trained:

- A fixed grid of Generalized Linear Models (GLMs),
- A Default Random Forest (DRF),
- Five pre-specified Gradient Boosting Machines (GBMs),
- A near-default deep neural net,
- Extremely Randomized Trees (XRT),
- A random grid of GBMs, and
- A random grid of deep neural nets.

Output from the H2O AutoML algorithm includes a leaderboard that ranks all models trained in the process using five-fold cross-validated model performance by default. For a binary classification problem, the default metric used for ranking is AUC – the Area Under the receiver operating characteristics (ROC) Curve. Based on the AUC, the best model from each algorithm family was selected. The applicable algorithm families are: deep learning, DRF (includes XRT), GLM and GBM. For a binary classification problem, the GLM algorithm trains a logistic regression model with regularization. This study used version 3.36.0.1 of the R H2O Package for the Windows platform. Additional details on H2O can be found in the H2O documentation (H2O.ai 2022); an overview of the H2O AutoML algorithm can be found in LeDell and Poirier (2020).

3.4 Metamodel Validation

To assess the metamodels trained using different experimental designs, their predictive performance was evaluated using a random sample of 10,000 DPs. In the test set, 36% of the observations belong to Class 1 (raid negation) and 64% to Class 0 (no raid negation). Due to this class imbalance, the mean per class accuracy and the area under the precision recall curve were used to measure predictive performance in addition to the AUC. Details on these measures can be found in the H2O documentation (H2O.ai 2022).

4 ANALYSIS AND DISCUSSION

Figure 2 summarizes the algorithms that produced the best performing model in each family for all 96 training sets, 16 of which had 100 replications and 80 had 20 replications. For deep neural nets, the random grid search tended to produce better performing models than the near-default model. For GBMs, the pre-specified models tended to produce better performing models than the random grid search, especially for the data sets with 100 replications. For the random forest family, just over half of the models were XRTs. For logistic regression, only a fixed grid is considered by the H2O AutoML algorithm.

While three performance measures were considered, they all showed the same overall trends; therefore, only a single performance measure is presented. Figure 3 compares the predictive performance of the models using the mean per class accuracy on an independent test set of 10,000 random DPs. This plot highlights several trends. Firstly, models trained using a FF design tended to have the weakest performance scores overall, and were consistently outperformed by designs of the same size, including RS designs. This latter finding is consistent with the findings of Hurrion and Birgil (1999) for neural networks. However, some caution is needed as only a single set of random DPs was considered in each case. Secondly, for most models, greater gains in performance tended to be observed when the number of DPs increased than when the number of replications increased. This is consistent with the findings of MacDonald and Gunn (2012) for neural networks, again suggesting that increased spatial coverage can be favored over increased replications. A notable exception is logistic regression where the mean per class accuracy remained relatively stable as the number of DPs or replications increased for the NOLH design and all but the smallest RS design. This finding mirrors those for polynomial regression metamodels in Santos and Santos (2009), which found that there was no significant difference in the precision of the fitted metamodels between designs with more DPs, but fewer replications, and their high-replication, fewer DPs, counterparts.

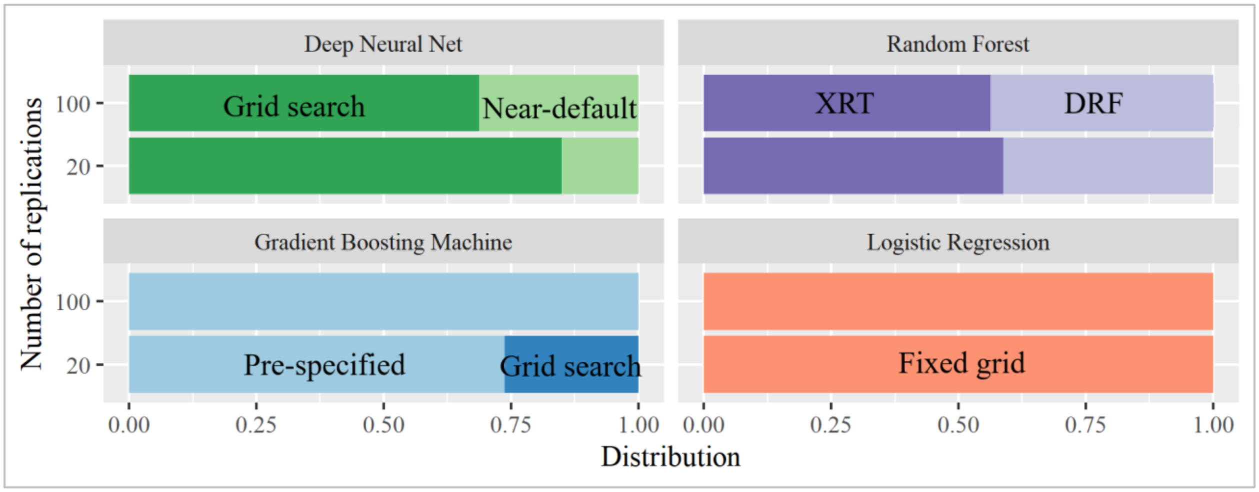

Figure 2: Distribution of algorithms that produced the best performing model in each of the four families of models across all training sets, which had either 20 or 100 replications.

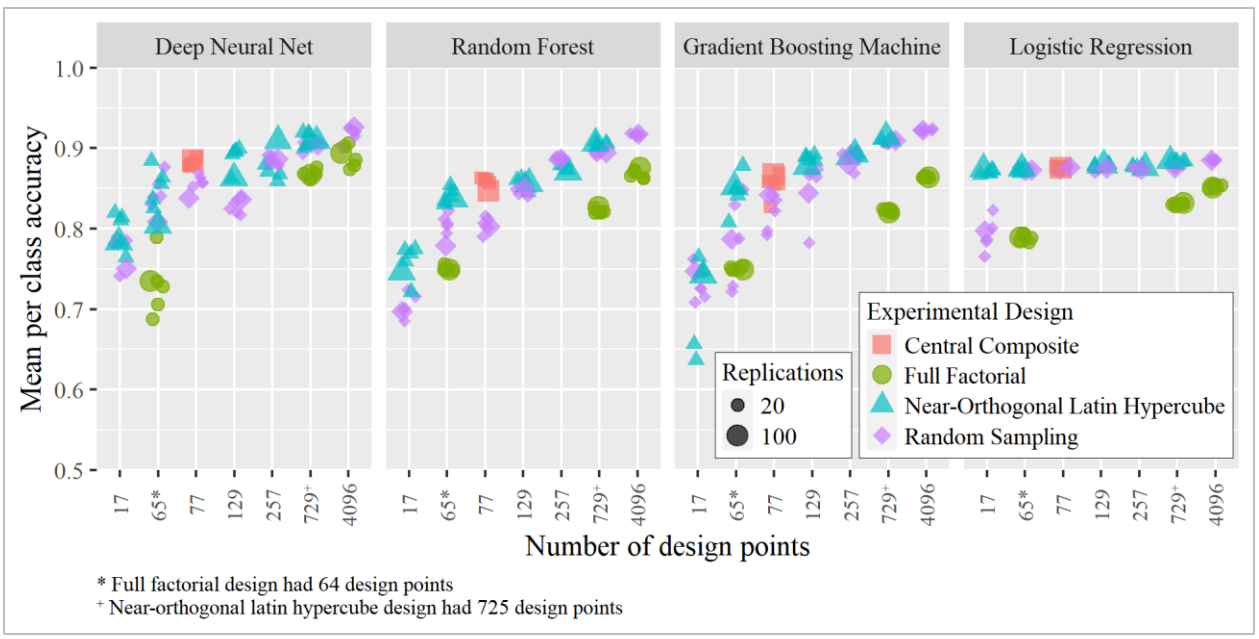

Figure 3: Comparison of the performance of the binary classifiers trained using automated machine learning on data sets generated with different experimental designs, numbers of design points, and replications.

AutoML tools, like H2O, where the process of training and tuning machine learning models is done by a single function call, increase the efficiency of metamodel construction by reducing the amount of time an analyst spends coding. AutoML tools that also include XAI methods, such as variable importance measures, can introduce further efficiencies by allowing the analyst to explore the models further in the same environment with minimal additional coding. As an example, Figure 4 presents a variable importance heatmap based on the variable importance measures available in H2O. Note that H2O uses different variable importance calculations in their heatmaps for different types of models; further, the results are then standardized to a common scale to compare relative variable importance. Therefore, the heatmap should be viewed as an exploratory tool that shows how different models treated each variable. In Figure 4, when the number of DPs was small, some of the models considered one variable to be much more important than the others. These models also tended to have poorer performance. The deep neural net generally gave more equal weight to all variables; however, as the number of DPs increases, all models indicated that the threat speed and system ammunition load are the least important variables. Consistency across different types of metamodels adds robustness to the findings whereas inconsistencies may indicate a need for further analysis.

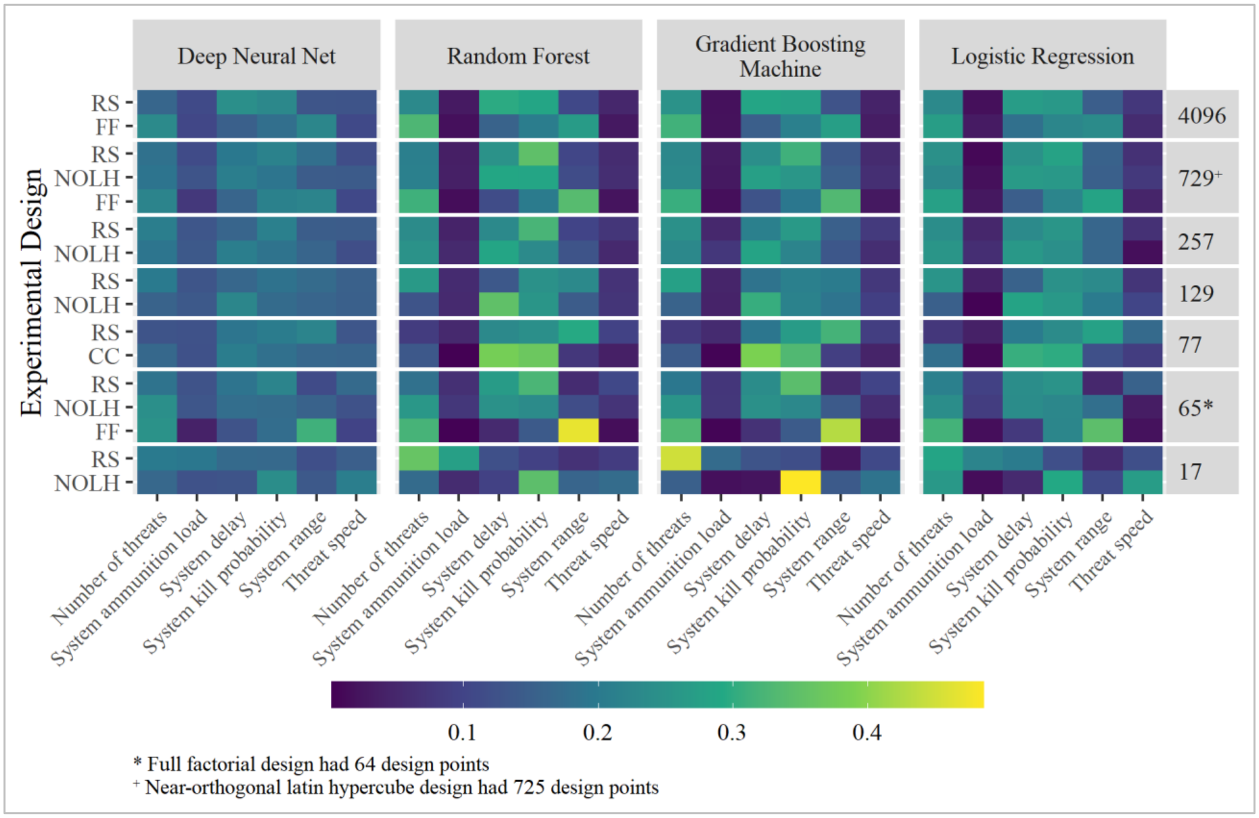

Figure 4: Standardized variable importance measures for the binary classifiers trained using automated machine learning on data sets generated with different experimental designs (left axis), numbers of design points (right axis), and 100 replications. Higher values indicate higher relative importance.

As a second example of XAI methods, Figure 5 presents the SHapley Additive Explanations (SHAP) summary plot generated using H2O for the best performing GBM model trained using the NOLH design with 725 DPs and 100 replications. Proposed by Lundberg and Lee (2017), SHAP values provide a model-agnostic approach for calculating variable importance. Each dot represents the SHAP value for a DP and is colored by the value of the corresponding individual feature with purple (darkest color) representing low values and yellow (lightest color) representing high values. In Figure 5, the variables are ordered from most important (system delay) to least important (system ammunition load) based on their global impact (i.e., the sum of the absolute values of the SHAP values), which agrees with the ordering in Figure 4 for this data set and model. For the system delay and the number of threats, the smooth gradation in coloring indicates a smooth increase in the model's output (odds of raid negation) as the value of these variables decreases (e.g., raid negation is more likely when the number of threats is smaller). The opposite trend is observed for the system kill probability and system range: the smooth gradation in coloring indicates a smooth increase in the model's output as the value of these variables increases (e.g., raid negation is more likely when the system kill probability is higher).

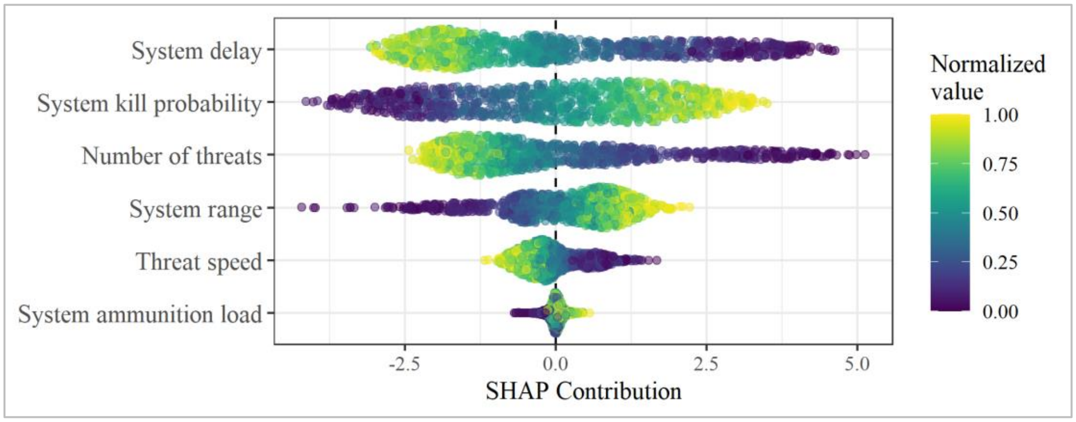

Figure 5: SHAP summary plot for the best performing GBM classifier trained using the NOLH design with 725 design points and 100 replications.

CONCLUSION

Recent research has demonstrated that XAI methods can be used within the data farming process to analyze the simulation output (Amyot-Bourgeois et al. 2021; Serré et al. 2021; Feldkamp 2021). However, before these methods can be applied, a well-trained machine learning model is needed. Our paper explored the use of autoML to increase the efficiency of metamodel construction while at the same time examining the relationship between the experimental design and precision of machine learning metamodels.

Four families of machine learning models were considered: deep neural nets, random forests (including XRTs), GBMs, and logistic regression with regularization. Classifiers trained using a FF design generally had weaker performance and were consistently outperformed by designs of the same size. For deep neural nets, random forests, and GBMs, greater gains in performance tended to occur when the number of DPs was increased rather than the number of replications. Together, these observations suggest that good space-filling properties are important for these types of classifiers. The performance of the logistic regression models was less impacted by the experimental design, number of DPs, and number of replications. It outperformed the other classifiers when the number of DPs was smaller. Variable importance heatmaps showed that once the number of DPs was large enough, the same set of four factors was identified as being more important by all families of models considered. This indicates some robustness in the findings and is a benefit of fitting several types of metamodels.

AutoML tools allow a wider set of metamodels to be trained and tested with less coding effort from an analyst. The results can be used as a starting point for additional model training and testing, and can also identify initial trends to investigate further. When XAI methods are integrated within autoML tools, additional model investigation can be done efficiently within the same environment. AutoML tools are part of a rapidly evolving field of study; as they continue to evolve, they can play a key role in building metamodels more efficiently and enabling XAI within a data farming process.

REFERENCES

Alam, F. M., K. R. McNaught, and T. J. Ringrose. 2004. "A Comparison of Experimental Designs in the Development of a Neural Network Simulation Metamodel". *Simulation Modelling Practice and Theory* 12(7-8):559–579.

Amyot-Bourgeois, M., L. Serré, and P. Dobias. 2021. "Use of Agent-Based Modeling and Data Farming for the Army Intelligence, Surveillance and Reconnaissance (ISR) Capability Assessment". In *Proceedings of the 14th NATO Operations Research and Analysis Conference: Emerging and Disruptive Technology*, STOPublisher.

Anderson, M. A. 2013. "Agent-Based Modelling in the New Zealand Defence Force". In *Proceedings of the 3rd International Defense and Homeland Security Simulation Workshop*, edited by A. Bruzzone, W. Buck, F. Longo, J. A. Sokolowski, and R. Sottilare, 61–66. ISBN 978-88-97999-21-8.

Cioppa, T. M., and T. W. Lucas. 2007. "Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes". *Technometrics* 49(1): 45-54.

De la Fuente, R. and R. Smith. 2017. "Metamodeling a System Dynamics Model: A Contemporary Comparison of Methods". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 1926–1937. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

De Reus, N., A. De Vos, B. Akesson, G. Horne, T. Kuhn, V. Sstritof, S. Seichter, and A. Zimmermann. 2018. "Data Farming Services in Support of Military Decision Making". In *Proceedings of the Specialist Meeting IST-160 on Big Data & Artificial Intelligence for Military Decision Making*, ISBN: 978-92-837-2181-9. STOPublisher.

Ebadi, A., Y. Gauthier, S. Tremblay, and P. Paul. 2019. "How Can Automated Machine Learning Help Business Data Science Teams?". In *Proceedings of the 18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, edited by M. A. Wani, T. M. Khoshgoftaar, D. Wang, H. Wang, and N. J. Seliya, 1186–1191. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Feldkamp, N. 2021. "Data Farming Output Analysis Using Explainable AI". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Feldkamp, N., S. Bergmann, and S. Strassburger. 2020. "Knowledge Discovery in Simulation Data". *ACM Transactions on Modeling and Computer Simulation* 30(24):1-25.

Feurer, M., K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter. 2021. "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning". https://arxiv.org/pdf/2007.04074.pdf, accessed 26 November 2021.

Ghahramani, Z. (2019). "Forward". In *Automated Machine Learning: Methods, Systems, Challenges*, edited by F. Hutter, L. Kotthoff, and J. Vanschoren. Cham, Switzerland: Springer Nature Switzerland AG.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

Hill, B., D. Vukcevic, T. Caelli, and A. Novak. 2019. "Insights into the Health of Defence Simulated Workforce Systems using Data Farming and Analytics". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 2491–2502. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Horne, G., B. Akesson, T. Meyer, S. Anderson, et al. 2014. "Data Farming in Support of NATO: Final Report of Task Group MSG-088". STO Technical Report, STO-TR-MSG-088.

Horne, G., S. Seichter, B. Akesson, M. De Reus, et al. 2018. "Developing Actionable Data Farming Decision Support for NATO: Final Report of MSG-124". STO Technical Report, TR-MSG-124.

H2O.ai. (2022). *Overview*. Accessed 18 March 2022.

Hurrion, R. D., and S. Birgil. 1999. "A Comparison of Factorial and Random Experimental Design Methods for the Development of Regression and Neural Network Simulation Metamodels". *Journal of the Operational Research Society* 50(10):1018–1033.

Kallfass, D., and T. Schlaak. 2012. "NATO MSG-088 Case Study Results to Demonstrate the Benefit of Using Data Farming for Military Decision Support". In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kesler, G., T. W. Lucas, and P. J. Sanchez. 2019. "A Data Farming Analysis of a Simulation of Armstrong's Stochastic Salvo Model". In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H. G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 2443–2452. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Kleijnen, J. P. C., and R. G. Sargent. 2000. "A Methodology for Fitting and Validating Metamodels in Simulation". *European Journal of Operational Research* 120(1):14–29.

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

Le, T. T., W. Fu, and J. H. Moore. 2020. "Scaling Tree-Based Automated Machine Learning to Biomedical Big Data with a Feature Set Selector". *Bioinformatics* 31(1):250–256.

LeDell, E., and S. Poirier. 2020. "H2O AutoML: Scalable Automatic Machine Learning". In *7th ICML Workshop on Automated Machine Learning*. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf, accessed 20 August 2021.

Lenth, R. V. 2001. "Some Practical Guidelines for Effective Sample Size Determination". *The American Statistician* 55(3):187-193.

Lucas, T. W., S . M Sanchez, F. Martinez, L. R. Sickinger, and J. W. Roginski. 2007. "Defense and Homeland Security Applications of Multi-Agent Simulations". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 138–149. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lundberg, S. M., and S.-I. Lee. 2017. "A Unified Approach to Interpreting Model Predictions". In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 1-10. Red Hook, New York: 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. Lee. 2020. "From Local Explanations to Global Understanding with Explainable AI for Trees". *Nature Machine Intelligence* 2(1):56–67.

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*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher. Piscataway, 2578- 2589.New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Meckesheimer, M., A. J. Booker, R. R. Barton, and T. W. Simpson. 2002. "Computationally Inexpensive Metamodel Assessment Strategies". *American Institute of Aeronautics and Astronautics Journal* 40(10):2053–2060.

NIST/SEMATECH. 2013a. "5.3. Choosing an Experimental Design". In *e-Handbook of Statistical Methods*. https://www.itl.nist.gov/div898/handbook/pri/section3/pri3.htm, accessed 21 March 2022.

NIST/SEMATECH. 2013b. "7.2.3.2. Sample Sizes Require". In *e-Handbook of Statistical Methods*. https://www.itl.nist.gov/div898/handbook/prc/section2/prc232.htm, accessed 21 March 2022.

NIST/SEMATECH. 2013c. "5.3.3.6.1. Central Composite Designs (CCD)". In *e-Handbook of Statistical Method.* https://www.itl.nist.gov/div898/handbook/pri/section3/pri3361.htm, accessed 21 January 2022.

Pereda, M., J. I. Santos, and J. M. Galan. 2017. "A Brief Introduction to the Use of Machine Learning Techniques in the Analysis of Agent-Based Models". In *Advances in Management Engineering*, edited by C. Hernández, 179–186. Cham, Switzerland: Springer International Publishing AG.

Sanchez, S. M. 2011. "NOLH designs spreadsheet". https://nps.edu/web/seed/software-downloads, accessed 2 February 2022.

Sanchez, S. M. 2020. "Data Farming: Methods for the Present, Opportunities for the Future". *ACM Transactions on Modeling and Computer Simulation* 30(4):1-30.

Sanchez, S. M., and H. Wan. 2015. "Work Smarter, Not Harder: A Tutorial on Designing and Conducting Simulation Experiments". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 1795–1809. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Santos, P. R., and M. I. Santos. 2009. "Design Experiments for the Construction of Simulation Metamodels". In *Proceedings of the 23rd European Conference on Modelling and Simulation*, edited by J. Otamendi, A. Bargiela, J. L. Montes, and L. M. D. Pedrera, 338–344. Nottingham, UK: European Council for Modelling and Simulation.

Serré, L., M. Amyot-Bourgeois, and B. Astles. 2021. "Use of Shapley Additive Explanations in Interpreting Agent-Based Simulations of Military Operational Scenarios". In *Proceedings of the 2021 Annual Modeling and Simulation Conference (ANNSIM'21)*, edited by C. R. Martin, M. J. Blas, and A. I. Psijas. 1-12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

LYNNE SERRÉ first joined Canada's Department of National Defence in 2013 as a defense scientist under Director General Military Personnel Research and Analysis where she specialized in military workforce modeling and analysis. In 2019, she joined Defence Research and Development Canada's Centre for Operational Research and Analysis, providing support to the Canadian Army headquarters in Ottawa. Since 2021, her research has been focused on topics related to air defense. She obtained her Master's degree in computational mathematics from the University of Waterloo, Canada. Her email address is lynne.serre@ecn.forces.gc.ca.

MAUDE AMYOT-BOURGEOIS is a junior defense scientist with Defence Research and Development Canada's Centre for Operational Research and Analysis. Since 2019, she has worked in collaboration with her colleagues from the Canadian Army Operational Research and Analysis Team on various combat simulation and wargaming studies. She obtained her Master's degree in physics from the University of Ottawa, Canada. Her email address is maude.amyot-bourgeois@ecn.forces.gc.ca.