Article:
Degeling, K., IJzerman, M.J., Lavieri, M.S. et al. (2 more authors) (2020) Introduction to metamodeling for reducing computational burden of advanced analyses with health economic models : a structured overview of metamodeling methods in a 6-step application process. Medical Decision Making, 40 (3). pp. 348-363. ISSN 0272-989X

https://doi.org/10.1177/0272989X20912233

Degeling, K., IJzerman, M. J., Lavieri, M. S., Strong, M., & Koffijberg, H. (2020). Introduction to Metamodeling for Reducing Computational Burden of Advanced Analyses with Health Economic Models: A Structured Overview of Metamodeling Methods in a 6-Step Application Process. Medical Decision Making, 40(3), 348–363. Copyright © 2020 The Author(s) DOI: 10.1177/0272989X20912233

Title

An introduction to metamodeling for reducing computational burden of advanced analyses with health economic models: a structured overview of metamodeling methods in a six-step application process

Running heading

Metamodeling methods for health economics

Authors

K. Degeling, PhD, M.J. IJzerman, PhD, M.S. Lavieri, PhD, M. Strong, PhD, H. Koffijberg, PhD

Koen Degeling, PhD, Cancer Health Services Research Unit, School of Population and Global Health, Faculty of Medicine, Dentistry and Health Sciences, University of Melbourne, Melbourne, Australia, and Health Technology and Services Research Department, Faculty of Behavioural, Management and Social Sciences, Technical Medical Centre, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands. Email: koen.degeling@unimelb.edu.au

Maarten J. IJzerman, Cancer Health Services Research Unit, School of Population and Global Health, Faculty of Medicine, Dentistry and Health Sciences, University of Melbourne, Melbourne, Australia, Victorian Comprehensive Cancer Centre, Melbourne, Australia, and Health Technology and Services Research Department, Faculty of Behavioural, Management and Social Sciences, Technical Medical Centre, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands. Email: m.j.ijzerman@utwente.nl

Mariel S. Lavieri, PhD, Industrial and Operations Engineering, University of Michigan, 1891 IOE Building, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, United States. Email: lavieri@umich.edu

Mark Strong, PhD, School of Health and Related Research (ScHARR), University of Sheffield, 30 Regent Street, Sheffield S1 4DA, United Kingdom. E-mail: m.strong@sheffield.ac.uk

Hendrik Koffijberg, PhD, Health Technology and Services Research Department, Faculty of Behavioural, Management and Social Sciences, Technical Medical Centre, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands. Email: h.koffijberg@utwente.nl

Corresponding author: Hendrik Koffijberg, PhD

Abstract

Metamodels can be used to reduce the computational burden associated with computationally demanding analyses of simulation models, though applications within health economics are still scarce. Besides a lack of awareness of their potential within health economics, the absence of guidance on the conceivably complex and time-consuming process of developing and validating metamodels may contribute to their limited uptake. To address these issues, this paper introduces metamodeling to the wider health economic audience and presents a process for applying metamodeling in this context, including suitable methods and directions for their selection and use. General (i.e., non-health economic specific) metamodeling literature, clinical prediction modeling literature, and a previously published literature review were exploited to consolidate a process and to identify candidate metamodeling methods. Methods were considered applicable to health economics if they are able to account for mixed (i.e., continuous and discrete) input parameters and continuous outcomes. Six steps were identified as relevant for applying metamodeling methods within health economics, i.e. 1) the identification of a suitable metamodeling technique, 2) simulation of datasets according to a design of experiments, 3) fitting of the metamodel, 4) assessment of metamodel performance, 5) conduct the required analysis using the metamodel, and 6) verification of the results. Different methods are discussed to support each step, including their characteristics, directions for use, key references, and relevant R and Python packages. To address challenges regarding metamodeling methods selection, a first guide was developed towards using metamodels to reduce the computational burden of analyses of health economic models. This guidance may increase applications of metamodeling in health economics, enabling increased use of state-of-the-art analyses, e.g. value of information analysis, with computationally burdensome simulation models.

1. Introduction

Decision analytic models are valuable tools to inform health policy decisions by estimating the health and economic impact of healthcare technologies. When decision analytic models take the form of simulation models, and particularly if they incorporate patient-level heterogeneity and stochasticity, the computational power of standard desktop computers may be insufficient to perform computationally demanding analysis within feasible time horizons (1-3). Although it is typically feasible to perform traditional analyses, such as probabilistic analysis to reflect parameter uncertainty (4), performing more advanced analyses, such as value of information analysis (5), may not be possible within a feasible timeframe unless simulations are executed in parallel using high performance computing clusters. Similarly, if we wish to optimize some specific model outcome, for example to identify a screening or treatment strategy that maximizes patient outcomes subject to some set of constraints, we may find that this is infeasible using only desktop computing resources (6).

Performing these more advanced analyses may be computationally challenging, because they can require a large number of model evaluations (i.e., simulation runs). For example, suppose a discrete event simulation model has been developed to estimate the health economic impact of a novel cancer drug compared to an existing drug. Now assume that running this simulation model with 10,000 hypothetical patients for each of the two treatment strategies is sufficient to obtain stable outcomes over model runs and takes approximately 1 minute. If an expected value of perfect parameter information analysis is to be performed for only 1 group of parameters using an inner probabilistic analysis simulation loop of 5,000 runs and outer simulation loop of 2,500 runs, 12.5 million simulation runs would be required in total. Even if it only requires 1 minute to perform a simulation run, performing this analysis using a brute force approach on a desktop computer with 8 central processing unit (CPU) cores working in parallel would take more than 1000 days.

Metamodeling methods can be applied to reduce the computational burden of computationally demanding analyses with simulation models (7, 8). A metamodel, also known as surrogate model or

emulator, in general can be thought of as a function that approximates an outcome (i.e., response variable or dependent variable) of a simulator (i.e., the original simulation model) based on input that would otherwise have been provided to that simulator (9). Metamodels are typically defined over the same (constrained) input parameter range as the corresponding simulator, as caution is needed when extrapolating input parameter values beyond their simulator range. Since metamodels are computationally cheap to evaluate, requiring only a fraction of the time that it takes to evaluate the simulator, they can be used as substitute for the simulator to substantially reduce the analysis runtime. In the example above, and as illustrated in Figure 1, metamodels can be used to replace a health economic simulation model. Although this will still require 12.5 million evaluations to be performed, this can be done in very limited time. For example, if a metamodel would require approximately 0,1 second to evaluate, performing the analysis using the metamodel would take 2 instead of over 1000 days. However, metamodels themselves take time to build and validate (10, 11), though this will not take 1000 days.

Figure 2, which will be discussed in detail throughout this manuscript, includes an overview of how metamodels are developed. First, a set of experiments is to be defined. An experiment refers to the generation of a single sample of values of the model input parameters (so, if there are k input parameters, a vector of length k), which is different to the use of the word "experiment" in the context of clinical studies. For health economic models, these input parameters may be probabilities, costs or utilities, for example. Second, the set of experiments is to be evaluated from the simulator to obtain a training dataset that contains the experiments and their corresponding model outcomes, such as mean or incremental costs and quality-adjusted life years (QALYs). Finally, metamodels are fitted to the training dataset to approximate the relationship between simulator inputs and outcomes. Different metamodeling techniques can be used to approximate this relationship, each of which makes different assumptions about the functional form of the relationship between the inputs and outcomes of the simulator. Although the extent to which fitted metamodels can be interpreted varies, this is not of primary interest when using metamodels to reduce computational burden, because the main aim is to approximate simulators outcomes accurately and not to

make inferences between inputs and outcomes. Most techniques approximate a single model outcome, requiring multiple metamodels to approximate multiple simulator outputs. Hence, one metamodel can be used to approximate the net health benefit at a given willingness to pay, but two metamodels would be required to approximate costs and QALYs separately (Figure 1). After developing a metamodel, it needs to be validated by assessing its accuracy in approximating simulator outcomes, which is done based on a testing dataset containing experiments and outcomes that should be similar but different from those included in the training dataset.

Metamodeling methods are used widely across different fields of science and engineering, for example to optimize designs of coronary stents (12), high speed trains (13), and groundwater remediation (14), as well as to estimate future water temperatures (15). In health economics, de Carvalho et al. recently demonstrated that metamodel can be used to perform probabilistic analysis, which was not possible in a feasible timeframe using their original model (16). A previous literature review only identified 13 additional applications of metamodeling methods in health economics, mostly aiming to perform value of information analysis and applying various, relatively basic metamodeling methods compared to those used in other fields of research, suggesting the field of metamodeling within health economics to be in its infancy (3). An important reason for the limited uptake of metamodeling methods within health economics may be that most health economic models and applied analyses have, until recently, been relatively simple and could often be performed within acceptable time frames. Other potential reasons include a lack of awareness of the potential of metamodeling methods to reduce runtime, and a lack of guidance on how to apply these methods in a health economic context, which would explain the diversity in methods applied.

To increase awareness of the potential for applying metamodels within health economics, and provide guidance for doing so, this study introduces the concepts of metamodeling to the wider health economic audience, and presents a comprehensive, structured overview of metamodeling methods deemed suitable for use in a health economic context. Points of consideration for selecting and applying metamodeling methods are discussed, including directions specific to health economics.

2. Identification of Metamodeling Methods

Metamodeling methods (and the steps to be taken when applying them) were identified by a scoping literature search that was performed by KD. This involved online searches, searches in Scopus and PubMed, and cross-referencing. Several publications that provide information on steps taken when applying metamodeling methods in health economics, identified in a recent review (3), were used as a starting point (17-20). Method-specific information, other candidate metamodeling methods, and potentially relevant process steps were identified by iterative searches on methods and process steps introduced in these publications and by cross-referencing. For example, if the impact of different experimental designs on metamodel performance was discussed in a paper found from a search on a specific metamodeling technique (i.e., structure of the metamodel), additional searches on these designs of experiments were performed to identify further information on these experimental designs and other designs of experiments. The iterative search process terminated when additionally found literature did not result in further inclusion of methods, i.e. until theoretical saturation was reached.

Metamodeling methods were only included if they are considered appropriate for use in health economics and have been commonly used in other fields of research, in line with the objective of the study. Methods were considered applicable to health economics if they are able to account for mixed (i.e., continuous and discrete) input parameters and continuous outcomes (i.e., response variables). Typical continuous input parameters of health economic models are, for example, costs and utilities, whereas the number of hospital days after a surgical intervention can be included as a discrete parameter. Similarly, typical continuous outcomes of interest are the net health or monetary benefit, total cost, and QALYs. Relevant steps to be taken when applying metamodeling methods in health economics were not prespecified, but extracted from the literature as described above, and structured in a process. Metamodeling methods and their characteristics were described according to this process, and presented in a table or graphs when appropriate. Additionally, examples of packages available to implement methods in R Statistical

Software (21) and Python (22) were identified via an online search and introduced along with the corresponding methods. These two software environments were selected, because they can be used to develop both the health economic simulation model and metamodel in one script, and are commonly used by academics, though other software environments like SAS, Stata, and C++ can also be used to develop metamodels.

3. A Process for Metamodeling in Health Economics

A six-step process for metamodeling in health economics was consolidated, covering methods from selecting suitable metamodeling techniques up to validating metamodel outputs against simulator outputs (Figure 2). A validated health economic simulation model (i.e., simulator) that is considered appropriate to perform the analysis of interest is a prerequisite, because while metamodels can theoretically be as accurate as their corresponding simulators, they cannot compensate for inaccuracies or bias in these simulators. Here, the analysis refers to what is to be analyzed using the original health economic model, but is considered infeasible due to the associated computational burden. Depending on the analysis to be performed, the sixth step is facultative, as will be discussed. As for any type of modeling study, the process of metamodeling is iterative, since new insights may question prior decisions. Next, each process step will be described, including an overview of corresponding methods. An illustration of how this process would be applied to perform value of information analysis is presented in Appendix A.

Step 1: identifying candidate metamodeling techniques

Identification of theoretically suitable metamodeling techniques is based on study characteristics, including the analysis to be performed, type of input parameters (continuous, discrete, or mixed (i.e., both continuous and discrete)), number of input parameters, and type of outcome (continuous or discrete). As discussed previously, the focus here is on techniques capable of handling mixed input parameters and continuous

outcomes. In the presence of time or budget constraints for metamodel development, and when multiple techniques are considered appropriate for use, modelers can start by selecting and applying one of these techniques and only select and apply another technique if the resulting metamodel does not yield acceptable performance (see Step 4).

Tappenden et al. (18) identified five metamodeling techniques for application in value of information analysis: linear regression, response surface methodology, multivariate adaptive regression splines, Gaussian processes, and neural networks. These techniques are complemented with symbolic regression, which was also identified from the review (19), and generalized additive models, which have been used previously for performing value of information analysis (23, 24). In Table 1, an overview of techniques and their characteristics is provided. For each metamodeling technique, this overview includes the typically required number of experiments (which we have defined as low: $n < 500$, or high: $n \geq 500$), number of input parameters it allows (which we have defined as low: $n < 20$, or high: $n \geq 20$), interpretability of the resulting metamodel structure (which we have classified as low: not or barely possible to understand relations between inputs and outputs, moderate: input-output relations can be understood to some extent, or high: input-output relations can be understood), and the description of any R and Python packages available to apply the technique. Regarding the interpretability of the metamodels' structures, this is typically not of primary interest when using metamodeling for reducing computational burden, as accurate and fast approximation of simulator outcomes is the main goal.

Simple linear regression is a statistical modeling technique well known in health economics and, theoretically, suitable for metamodeling. It assumes a linear relationship between independent variables (i.e., input parameters) and the dependent variable (i.e., outcome of interest) and is linear in the regression model parameters (25). These models can easily be fitted to datasets of all sizes, including datasets with large numbers of experiments and input parameters, while allowing for both continuous and categorical input parameters. Although fitting linear regression models and interpreting their structure can be considered relatively easy, they are unlikely to be useful as metamodels of health economic simulation

models, as the latter typically induce complex and non-linear parameter interactions. More advanced techniques, allowing for more flexible model structures, are often better suited to represent such simulation models.

Response surface methodology is also linear in the regression model parameters, but does not assume a linear input-output relationship, and fits polynomial regression models to predict responses, i.e. outcomes (10, 26, 27). Both continuous and categorical input parameters can be considered in response surface models, and datasets including large numbers of experiments and input parameters can be used. However, high non-linearity will require higher order polynomials, which will require larger numbers of experiments and, hence, it will require larger up-front simulator runtime. Although polynomial models are more difficult to interpret compared to linear models, if desired, general trends on model parameter influence can still be extracted from their model structures.

Symbolic regression uses genetic programming to construct a mathematical expression from elementary operators (e.g., '+' and '×') and elementary functions (e.g., 'log') accurately describing the relation between input parameters and the outcome of interest, without making any priori assumption about this relationship (28, 29). Fitting an accurate symbolic regression model may take substantial time, due to a potentially large number of candidate metamodels, i.e. large solution space. However, symbolic regression is capable of handling large datasets including a large number of mixed input parameters. Symbolic regression models can be difficult to interpret unless the final expression is relatively simple or is simplified.

Multivariate adaptive regression splines were developed to model input-outcome relations that may not be constant across input space (10, 30-32). Regression spline modeling divides the outcome domain into intervals, and then estimates an equation, typically a low-order polynomial, for each interval. Different types of splines can be distinguished, based on how the number of intervals and level of smoothness are defined. Fitting multivariate adaptive regression splines includes an automated input parameter importance analysis (see Step 2). Although capable of handling large datasets of mixed input parameters, regression

splines are prone to overfitting. In contrast to the previously discussed metamodeling techniques, the interpretability of multivariate adaptive regression splines is limited.

Generalized additive models assume that the dependent variable is a smooth, but unknown, function of the independent variables (33, 34). This unknown underlying smooth function is usually represented using splines, with the cubic spline as a common choice. In its simplest case, a univariate cubic spline represents an arbitrary smooth single-input function as a series of short cubic polynomials joined piecewise such that the function is twice-differentiable at the ''knots'' (i.e., join points). The same spline can also be represented as the weighted sum of a series of predetermined ''basis functions'' that extend over the whole range of the function input. Simple univariate cubic splines have natural extensions to higher dimensions and to a metamodeling framework, where the spline parameters (i.e., the basis function weights) are estimated from noisy data. Generalized additive models can handle large datasets and high numbers of input parameters, but their structure is difficult to interpret.

Gaussian process regression is a nonparametric regression method also known as Kriging (10, 35). Gaussian processes use information on neighbor experiments for new predictions, while directly providing information on the uncertainty in these predictions. This is unique for metamodeling techniques, since other techniques require additional effort to obtain information on prediction uncertainty. Although Gaussian processes are capable of considering mixed input parameters (36), Treed Gaussian processes have been developed specifically for this type of data (37). The interpretability of Gaussian processes is low. A disadvantage of Gaussian processes is that computational burden, both in terms of fitting and predicting, increases dramatically with increasing numbers of experiments and parameters, limiting their applicability. Hence, Gaussian processes are often well suitable for optimization problems, which are typically defined by limited numbers of decision parameters. Furthermore, input parameter importance analysis can be performed to reduce the number of parameters (see Step 2) and, thereby, computational burden.

Neural networks are non-parametric models that are commonly found in machine learning applications. These models exist of networks of nodes (called neurons) and layers, which learn about

relationships between inputs, either continuous or categorical, and outputs, typically using large datasets (10, 31, 38). Although neural networks are commonly used for classification, they are also able to predict continuous outcomes (39). Since no assumption regarding simulator structure is made, neural networks may well represent complex, i.e. non-linear, health economic models. Developing large neural networks typically requires large numbers of experiments, which may pose challenges regarding obtaining sufficient simulator samples. Similar to multivariate adaptive regression splines, generalized additive models, and Gaussian processes, neural networks are "black boxes" that are hard to interpret.

In conclusion, as illustrated in the selection flowchart (Figure 3), Gaussian processes are particularly useful when obtaining sufficient simulator samples to apply the other techniques is infeasible. Response surface methodology, symbolic regression, multivariate adaptive regression splines, generalized additive models, and neural networks are typically useful when sufficient samples can be obtained from the simulator, i.e. original health economic model. If metamodel interpretation is important, response surface methodology and symbolic regression can be used to develop metamodels that may be interpretable to some extent.

Step 2: simulating datasets

Simulating data from the simulator is crucial in metamodeling studies, as metamodel performance is highly dependent on the data used for fitting (9). Modelers control the number and definition of experiments used for fitting metamodels, which is fundamentally different from prediction modeling studies, for which data is typically observed from clinical studies or registries (25). Furthermore, challenges regarding handling missing data, reversed causality, omitted variables, and measurement error are not applicable to metamodeling. There are five key aspects to simulating datasets for metamodeling: 1) the number of datasets, 2) parameter ranges, 3) design of experiments, 4) number of experiments, and 5) analysis used for obtaining simulator outcomes. As explained previously, an experiment refers to the generation of a single

sample of model input parameter values in a metamodeling context. Hence, the number of experiments does not refer to a number of (hypothetical) patients, but to the number of sets of model input parameter values for which the simulation model is evaluated to create datasets for metamodel fitting and validation.

As in prediction modeling, two distinct datasets are preferred for metamodeling studies: one for fitting (i.e., training or development dataset) and one for validation (i.e., testing or validation dataset). In prediction modeling studies, validation datasets would typically be obtained by isolating a proportion of the data from a single cohort for internal validation, or by gathering additional data from another 'plausibly related' cohort for external validation (25). In metamodeling, however, it is preferable to obtain two separate datasets from the simulator, each having a prespecified design with comprehensive coverage. Obtaining one large dataset and separating it in two datasets for training and validation, may compromise the coverage of these datasets: either dataset may lack the structure and properties induced by the design of experiments used to generate the single large dataset. By obtaining two separate datasets, their structure and properties according to the design of experiments used will be maintained, as will be discussed.

The range of values that is to be covered in the datasets requires careful consideration for each input parameter separately. Although metamodels are theoretically capable of extrapolating beyond the parameter ranges covered by the dataset on which they were fitted, such extrapolations are not preferable. The ranges that need to be covered are determined by the ranges of interest in the analysis that is to be performed using the metamodel. For example, if a metamodel is developed to optimize a cancer screening strategy, the ranges that are considered feasible in the optimization should be the same as those in the datasets used for fitting and validating the used metamodel(s). If the screening interval in years is a parameter of interest and any value between 1 and 10 is considered feasible, the parameter range for this parameter in the training and testing dataset should also be from 1 to 10.

Design of experiments methods determine how sets of samples of parameter values are selected, to be evaluated from the simulator in order to obtain datasets for fitting and validation (40). The objective of these methods is to cover parameter spaces and parameter interactions as effectively and efficiently as

possible, i.e. with the least number of experiments. Failing to represent the full parameter spaces and parameter interactions will decrease metamodel performance. Most common designs of experiments are so called single-pass methods that first define a complete set of experiments, which are subsequently all evaluated using the simulator (11). Commonly used designs are random designs, full factorial designs, and Latin Hypercube designs.

Random designs, also known as Monte Carlo Sampling methods, obtain n sets of experiments by generating n draws from the joint probability distribution for the input parameters (40). These designs require a large number of experiments to sufficiently cover the parameter space. Input distributions may be designed to cover a pre-specified range with equal probability (i.e. uniform distributions), or may represent judgements about the true unknown value of some population quantity, for example using Gamma distributions for parameters with a positive range (4). When a random design is used, one large dataset can be separated in two datasets for fitting and validation, while maintaining its random properties.

Full factorial designs fully enumerate possible combinations of discrete parameter values (11). More specifically, for n values of k parameters, a full factorial design represents all n^k combinations of these parameter values. Although full factorial designs are able to cover the full parameter space and interactions, the number of experiments exponentially increases with the number of parameters and they are, therefore, often infeasible to use. Fractional factorial designs have been introduced to address challenges regarding high numbers of experiments when using factorial designs, and consist of subsets of full factorial designs (41).

Latin Hypercube designs have been used often for designing computer experiments, as they efficiently cover the full parameter space (40, 42, 43). In its simplest form, Latin Hypercube samples represent random combinations of values for each parameter, which are equally spaced between their minimum and maximum value for each parameter. More often, Latin Hypercube samples represent random combinations of random values from equally sized bins that cover the parameters' domains. Over the years,

more advanced versions have been developed, such as the maximin Latin Hypercube design (44), which maximizes the minimum distance between design points, and orthogonal Latin Hypercube designs (45).

Figure 4 illustrates how random, full factorial, and maximin Latin Hypercube designs may define nine experiments for two continuous parameters $Test\ Cost$ and $Consultation\ Cost$. Although simulators and metamodels in practice will have more than two parameters, this figure clearly demonstrates differences between the designs. It shows that parameter spaces are most effectively covered by maximin Latin Hypercube sampling, as the corresponding experiments are properly distributed over all bins of the parameter ranges. Conversely, the full factorial design covers some bins multiple times and others not at all. The randomly sampled experiments also cover some bins multiple times and others not at all, though which those are is determined by chance. From this figure, it can also be seen why simply isolating a proportion of experiments from the dataset for model validation is not appropriate, and a separate dataset needs to be simulated when a non-random design is used. Isolating a (random) proportion from a dataset generated according to a full factorial or Latin Hypercube design, will result in a training dataset that no longer covers the full parameter space consistently. The remaining experiments will no longer cover all bins of the parameter domain in a Latin Hypercube design, or all parameter value combinations in a full factorial design.

In general, Latin Hypercube designs are preferable for both training and testing datasets, especially when only a limited number of experiments can be evaluated from the simulator in the available time. Optimized Latin Hypercube designs can easily be generated in most software environments, for example using the `lhs` package in R (46) or `pyDOE` package in Python. However, these designs are challenging to apply when constraints on combinations of parameters are applicable. Although some work has been done on conditioned Latin Hypercube designs, accounting for inequality constraints (47), this might not enable all constraints to be accounted for. In such situations, factorial designs can be used if the resulting number of experiments is considered feasible. However, when using factorial designs for continuous variables, a finite set of discrete values within the continuous parameter range needs to be defined, which may result in

an infeasible number of experiments to cover those parameters' ranges at the desired level of detail. If using a factorial design is considered infeasible, random designs allow constraints to be accounted for easily. However, random designs are likely less efficient, which may result in suboptimal solution space coverage and, consequently, lower metamodel performance, especially when a limited number of experiments can be evaluated from the simulator.

How many experiments are required, i.e. how large n should be, heavily depends on the desired metamodel accuracy, which will be discussed in Step 4. Additionally, the design of experiments method used, and how well the metamodeling techniques match the unknown relation between inputs and outputs, influence the number of experiments required (48, 49). A general rule of thumb is to start with $n = 10 \times d$, where d refers to the number of input parameters (49, 50). After evaluating model performance for the initial set of experiments (see Step 4), n may be increased until the desired level of overall accuracy is achieved (see Appendix A for an example). Alternatively, adaptive sampling methods may be applied to improve accuracy in local regions of the parameter space (51), but these methods are outside the scope of this study (see Discussion section). If the desired model accuracy cannot be achieved with a feasible number of experiments, importance analysis methods may be applied to reduce the number of input parameters d, by analyzing which parameters are most important in terms of predicting the simulator outcomes (18, 52, 53). Only including the most parameters might result in less complex metamodel structures, and if redundant input parameters can be removed, metamodel accuracy may improve as overfitting is reduced.

Whether a deterministic or probabilistic analysis needs to be performed to evaluate experiments from the simulator, depends on the analysis to be performed with the metamodel. In a deterministic analysis, the simulator is evaluated once for the expected values of the input parameters (4). In a probabilistic analysis, the simulator is evaluated numerous times, typically thousands of times, based on parameter values sampled from distributions that reflect the uncertainty in the parameter values (i.e., second-order uncertainty). If a model is non-linear, which most health economic models are, health economic outcomes from a deterministic analysis are not equal to those of a probabilistic analysis (54). If metamodels are being

used to perform model probabilistic analysis or value of information, simulator outcomes based on a deterministic analysis should be used. If the aim is to perform calibration or optimization, simulator outcomes based on probabilistic analyses are preferred, because these are the values expected to be observed in reality given the current information. However, performing a probabilistic analysis for each experiment might not be feasible because of the required simulator runtime. In that case, outcomes from a deterministic analysis may be used to approximate the outcomes of a probabilistic analysis, though this should be clearly noted as a limitation when reporting the results.

The stability of outcome estimates is another important aspect. If stochastic uncertainty, also referred to as uncertainty on patient level or first-order uncertainty, is reflected in a patient-level simulation model, sufficient hypothetical patients need to be simulated to obtain stable outcomes. Similarly, regardless of whether first-order uncertainty is reflected, sufficient probabilistic analysis runs need to be performed to obtain stable point-estimates. When insufficient hypothetical patients are simulated, or probabilistic analysis runs performed, the subsequent noise in the data used for fitting metamodels may have a pernicious effect on metamodel performance. Outcomes can be considered stable, if outcomes obtained from simulations with different random numbers, but with the same input parameter values, are sufficiently similar. What defines "sufficiently similar", differs over case studies and should be discussed with all relevant stakeholders, e.g. care providers, decision makers, and modelers. Obtaining stable outcomes may require a substantial number of patients to be simulated, or simulation runs to be performed, and may not be feasible in practice. However, to reduce the number of patients to be simulated, or number of runs to be performed to obtain stable outcomes, variance reduction techniques may be applied, such as using common random numbers when comparing strategies (55, 56).

Step 3: fitting metamodels

After evaluating an initial set of experiments from the simulator, this training dataset can be used to fit selected metamodeling techniques. Steps involved in fitting processes differ between techniques, as well as

any settings to be provided. We refer to the corresponding literature and software documentation to learn about steps to be taken and settings to be provided (see Step 1 and Table 1). As a basic example, some metamodeling techniques or software packages require input parameters to be rescaled. Fitting meta-models is an iterative process, in which settings may be adapted, or more experiments may be evaluated from the simulator, after assessing model performance (see Step 4).

Step 4: assessing metamodel performance

Assessing performance of fitted metamodels is essential to further improve that performance, by iteratively improving (extending) the design of the training dataset used, or adapting the settings for fitting these models. Additionally, an initially selected metamodeling technique may be deemed inappropriate if performance does not reach an acceptable level, resulting in exclusion of this technique from the list of potential candidates (Step 1). Performance can be assessed using the testing dataset evaluated from the simulator in Step 2. Since metamodels of health economics models will typically predict continuous scale outcomes, of main interest is to quantify how close predictions are to actual simulator outcomes. Assessing accuracy and comparing different metamodels can be done graphically and using quantitative performance criteria. A validation plot, with predicted values on the x-axis and observed values from the simulator on the y-axis, is fundamental in assessing model performance, and presents information on systematic trends, as well as general performance (see Appendix A for an example). Several quantitative performance criteria are available, including mean or maximum values of the absolute error, absolute relative error, squared error, which may all be normalized using the sample range or standard deviation, and summarized by their mean or maximum values, and R^2 (31, 49, 57, 58).

It is important to be aware of performance criteria characteristics when selecting one, or several, to compare metamodels or to assess whether model performance is acceptable. For example, compared to mean absolute errors, mean squared errors places more weight on outliers. Additionally, compared to

squared errors, setting a desired level of accuracy is more straightforward for absolute (relative) errors, as these can be set by answering questions such as "what is the maximum mean deviation in predicted life-years the metamodel is allowed to have compared to the simulator outcomes?" What performance can be considered acceptable for deciding to apply a metamodel for performing analyses, differs over case studies and should be based on input from all stakeholders. For example, when the point-estimate for the incremental QALYs is 0.18 QALYs, an absolute error of 0.01 QALY may be considered appropriate by stakeholders. Since different performance criteria and definitions of acceptable performance may yield alternative conclusions, these should be decided upon prior to metamodel development.

Step 5: applying metamodel

Once a metamodel has been developed and validated, it can be used to perform analyses that could not be performed in a feasible time period with the original health economic model. Previous applications of metamodels in health economics include value of information analysis, model calibration, optimization, probabilistic analysis, and obtaining stable outcomes over multiple runs with the same input values (3). Additionally, metamodels can be used in online tools for which limited computer resources are available. For example, see de Carvalho et al. for a demonstration of metamodels used to probabilistic analysis (16), or Appendix A for an illustration of the presented six-step process for performing value of information analysis. Another example may be to use metamodels for evaluating a large set of (thousands of) screening strategies, for example to identify the starting age, screening interval, and number of screening rounds that optimize health and economic outcomes.

Step 6: verifying results (optional)

If metamodels are used for optimization purposes, it is recommended to re-evaluate a certain number of best strategies identified by the metamodel using the original health economic model, to assess whether

their outcome and ordering meaningfully differ. By providing these results, decision makers are better informed about the expected impact of choosing a good but not optimal candidate strategy for implementation, which may be favored over the optimal strategy for practical reasons. For other types of analyses, such as probabilistic or value of information analysis, additional verification will not add to the validation of Step 4, because re-evaluating a number of strategies using the simulator will yield approximately the same error values as those obtained in Step 4. Although this will also be the case for several best-performing strategies when optimization is performed, knowing the true outcomes and ordering of the strategies according the simulator is informative, whereas knowing the true outcome for a specific probabilistic analysis run is not of any value.

4. Discussion

This study provides an introduction to metamodeling methods that can be used to reduce the computational burden of advanced analyses with health economic models, and addresses challenges regarding the selection and application of these methods. Similar to ordinary statistical regression modeling, different methods are available with their own advantages, disadvantages, and underlying assumptions, which are discussed and directions for selecting and implementing these methods are provided. Selected methods are structured in a comprehensive six-step process that can be followed to assure essential modeling steps are covered, as it includes all relevant design choices. Additionally, the process discussed can be used as a structure to effectively and efficiently communicate metamodeling studies, to increase modeling transparency and reproducibility.

Given that tools and packages are available to generate experiments according to specific designs and to fit different types of metamodels, for example in R and Python, applying metamodeling methods is feasible for health economic analysts. Currently available software and results from this study enable analysts to perform computationally demanding analyses with their models, such as value of information

analysis, model calibration, and optimization. Benefits of developing metamodels are relevant to analyses using patient-level simulation methods, such as microsimulation state-transition modeling and discrete event simulation, but also to cohort models used to perform analyses that require a large number of model evaluations.

Applying metamodeling methods can reduce computational burden, but this usually comes at the price of introducing additional uncertainty in the model outputs. Consequently, checking whether underlying assumptions are met and checking metamodel performance, are crucial to success and essential to build confidence in the metamodel. Since modelers typically have access to the original health economic model, validation of the metamodel is often not a problem, though likely to be more demanding in terms of effort compared to developing the metamodel itself. The starting point for building any metamodel, however, should be a realistic and validated health economic model, since metamodels can theoretically be as accurate as their corresponding simulators, but will not compensate for inaccuracies in these simulators. Moreover, when metamodels are used for optimization, the strategies considered, and possibly identified as optimal, may not be supported by (the data underlying) the simulator. Caution is required when such extrapolation is (automatically) performed, and such optimization results should only serve to initiate discussion on the appropriateness and validity of the simulator and the data supporting it. In addition, application of metamodeling methods requires communication of metamodeling design choices made in publications, for which space typically already is limited. Hence, metamodeling studies may be published separately from their simulator to assure the metamodeling process can be appropriately described. Furthermore, there is a 'sweet spot' for metamodeling: sufficient experiments need to be evaluated using the simulator to develop an accurate metamodel, but evaluating all experiments of interest should not be feasible.

Several technical challenges regarding the application of metamodeling methods in health economics remain. Simulators in health economics may include complex behavior, such as rigid cutoffs due to clinical decision rules, which may be complex for metamodeling techniques to capture. Additionally,

(combinations of) model input parameters may be subject to constraints, which are difficult to incorporate in efficient designs of experiments, such as Latin Hypercube sampling. If sufficient samples can be evaluated from the simulator to use random or full factorial designs in which constraints can be accounted for more easily, however, this might not be an issue. Alternatively, more advanced adaptive sampling strategies may need to be applied.

Not all metamodeling methods are directly suitable for application in health economics and, hence, have been discussed. However, it is important to note that some techniques, such as those that can be used for categorical outcomes (i.e., classification), could theoretically be applied in health economics after discretizing continuous outcomes. Such an approach has been taken previously by using a binary outcome to reflect whether one treatment was preferred over another in a logistic regression model (53). Similarly, examples of packages for R and Python were discussed, whereas additional packages are likely to be available and other software environments can also be used to develop metamodels, such as Stata, SAS, and C++. Additionally, alternative performance criteria for metamodel validation can be found, or may be developed, based on study-specific needs. With regard to sampling methods, only single-pass methods have been discussed, whereas iterative methods, also known as adaptive sampling or active learning methods, also exist (59, 60). Iterative methods use an initial dataset for fitting an initial metamodel, which is subsequently used in an iterative process to identify additional experiments to be added to the dataset, to update the initial metamodel and check the updated metamodel performance, until this performance is according to a pre-defined threshold (51). The additional experiments are sampled in the area in which performance needs to be improved. Although iterative methods are more efficient compared to single pass methods, they are substantially more complicated to implement and require simulators to be available in the same software environment used for generating experiments and fitting the metamodel. Nevertheless, these methods may be useful if insufficient experiments according to a single-pass design can be obtained to develop an accurate metamodel. Also, alternative designs of experiments are available, such as D-optimal designs, which are efficient and can account for constraints, but for which an linear or quadratic model

simulator model structure should be known (61), or so called Sobol sequences, which may be more efficient compared to Latin Hypercube designs for low dimensions (i.e., number of input parameters) problems (62, 63).

Future metamodeling applications should further illustrate the potential and use of these research methods, and identify common challenges. Once the field of metamodeling in health economics has evolved, good research practices (i.e., consensus guidance) can be identified to further improve the quality of metamodeling studies.

References

1. Brennan A, Chick SE, Davies R. A taxonomy of model structures for economic evaluation of health technologies. Health economics. 2006;15(12):1295-310.

2. Karnon J, Haji Ali Afzali H. When to use discrete event simulation (DES) for the economic evaluation of health technologies? A review and critique of the costs and benefits of DES. PharmacoEconomics. 2014;32(6):547-58.

3. Degeling K, IJzerman MJ, Koffijberg H. A scoping review of metamodeling applications and opportunities for advanced health economic analyses. Expert Review of Pharmacoeconomics & Outcomes Research. 2018;19(2):1-7.

4. Briggs AH, Weinstein MC, Fenwick EAL, Karnon J, Sculpher MJ, Paltiel AD. Model Parameter Estimation and Uncertainty Analysis: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force Working Group–6. Medical Decision Making. 2012;32(5):722-32.

5. Heath A, Manolopoulou I, Baio G. A Review of Methods for Analysis of the Expected Value of Information. Medical Decision Making. 2017;37(7):747-58.

6. Crown W, Buyukkaramikli N, Thokala P, Morton A, Sir MY, Marshall DA, et al. Constrained Optimization Methods in Health Services Research - An Introduction: Report 1 of the ISPOR Optimization Methods Emerging Good Practices Task Force. Value in Health. 2017;20(3):310-9.

7. Karnon J, Stahl J, Brennan A, Caro JJ, Mar J, Möller J. Modeling using Discrete Event Simulation: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force-4. Medical Decision Making. 2012;32(5):701-11.

8. Law AM. Simulation Modeling and Analysis. Singapore: McGraw-Hill Higher Education, 2007.

9. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and Analysis of Computer Experiments. Statist Sci. 1989;4(4):409-23.

10. Barton RR. Simulation Metamodels. Proceedings of the 1998 Winter Simulation Conference. 1998:167-76.

11. Simpson TW, Poplinski JD, Koch PN, Allen JK. Metamodels for Computer-based Engineering Design: Survey and recommendations. Engineering with Computers. 2001;17(2):129-50.

12. Li H, Gu J, Wang M, Zhao D, Li Z, Qiao A, et al. Multi-objective optimization of coronary stent using Kriging surrogate model. BioMedical Engineering OnLine. 2016;15(2):148.

13. Xu G, Liang X, Yao S, Chen D, Li Z. Multi-Objective Aerodynamic Optimization of the Streamlined Shape of High-Speed Trains Based on the Kriging Model. PloS one. 2017;12(1):e0170803.

14. Ouyang Q, Lu W, Lin J, Deng W, Cheng W. Conservative strategy-based ensemble surrogate model for optimal groundwater remediation design at DNAPLs-contaminated sites. Journal of Contaminant Hydrology. 2017;203:1-8.

15. Butcher JB, Zi T, Schmidt M, Johnson TE, Nover DM, Clark CM. Estimating future temperature maxima in lakes across the United States using a surrogate modeling approach. PloS one. 2017;12(11):e0183499.

16. de Carvalho TM, Heijnsdijk EAM, Coffeng L, de Koning HJ. Evaluating Parameter Uncertainty in a Simulation Model of Cancer Using Emulators. Med Decis Making. 2019;39(4):405-13.

17. Rojnik K, Naversnik K. Gaussian process metamodeling in Bayesian value of information analysis: a case of the complex health economic model for breast cancer screening. Value in health. 2008;11(2):240-50.

18. Tappenden P, Chilcott JB, Eggington S, Oakley J, McCabe C. Methods for expected value of information analysis in complex health economic models: developments on the health economics of interferon-beta and glatiramer acetate for multiple sclerosis. Health Technol Assess. 2004;8(27): 1-78.

19. Willem L, Stijven S, Vladislavleva E, Broeckhove J, Beutels P, Hens N. Active learning to understand infectious disease models and improve policy making. PLoS computational biology. 2014;10(4):e1003563.

20. Yousefi M, Yousefi M, Ferreira RPM, Kim JH, Fogliatto FS. Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments. Artificial intelligence in medicine. 2018;84:23-33.

21. R Core Team. R: a language and environment for statistical computing. R Foundation for Statistical Computing; 2018 [Available from: https://www.r-project.org/] [Accessed 5 Dec 2018].

22. Python Core Team. Python: a dynamic, open source programming language. Python Software Foundation; 2019 [Available from: https://www.python.org/] [Accessed 25 July 2019].

23. Strong M, Oakley JE, Brennan A. Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample: A Nonparametric Regression Approach. Medical Decision Making. 2014;34(3):311-26.

24. Strong M, Oakley JE, Brennan A, Breeze P. Estimating the Expected Value of Sample Information Using the Probabilistic Sensitivity Analysis Sample: A Fast, Nonparametric Regression-Based Method. Medical Decision Making. 2015;35(5):570-83.

25. Steyerberg EW. Clinical Prediction Models: A Practical Approach to Development, Validation, and Updating. New York: Springer; 2009.

26. Carson Y, Maria A. Simulation Optimization: Methods and Applications. Proceedings of the 1997 Winter Simulation Conference. 1997:118-26.

27. Myers RH, Montgomery DC, Anderson-Cook CM. Response Surface Methodology: Process and Product Optimization Using Designed Experiments. Hoboken: Wiley, 2016.

28. Gaucel S, Keijzer M, Lutton E, Tonda A. Learning Dynamical Systems Using Standard Symbolic Regression. Berlin: Springer Berlin Heidelberg, 2014.

29. Koza JR. Genetic Programming: On the Programming of Computers by Means of Natural Selection. Cambridge: MIT Press, 1992.

30. Knafl GJ, Ding K. Adaptive Regression for Modeling Nonlinear Relationships. Basel: Springer International Publishing, 2016.

31. Li YF, Ng SH, Xie M, Goh TN. A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems. Applied Soft Computing. 2010;10(4):1257-73.

32. Madan J, Ades AE, Price M, Maitland K, Jemutai J, Revill P, et al. Strategies for efficient computation of the expected value of partial perfect information. Med Decis Making. 2014;34(3):327-42.

33. Hastie T, Tibshirani R. Generalized Additive Models. Statist Sci. 1986;1(3):297-310.

34. Wood S. Generalized Additive Models: An Introduction with R. Boca Ralton: Taylor & Francis; 2006.

35. O'Hagan A, Kingman JFC. Curve Fitting and Optimal Design for Prediction. Journal of the Royal Statistical Society Series B (Methodological). 1978;40(1):1-42.

36. Swiler LP, Hough PD, Qian P, Xu X, Storlie C, Lee H. Surrogate Models for Mixed Discrete-Continuous Variables. Cham: Springer International Publishing, 2014:181-202.

37. Gramacy RB, Lee HKH. Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. Journal of the American Statistical Association. 2008;103(483):1119-30.

38. Dreyfus G. Neural Networks: Methodology and Applications. Berlin: Springer Berlin Heidelberg, 2005.

39. Hurrion RD. Using a Neural Network to Enhance the Decision Making Quality of a Visual Interactive Simulation Model. The Journal of the Operational Research Society. 1992;43(4):333-41.

40. Garud SS, Karimi IA, Kraft M. Design of computer experiments: A review. Computers & Chemical Engineering. 2017;106:71-95.

41. Box GEP. Statistics for Experimenters: Design, Innovation, and Discovery. Hoboken: Wiley, 2005.

42. Husslage BGM, Rennen G, van Dam ER, den Hertog D. Space-filling Latin hypercube designs for computer experiments. Optimization and Engineering. 2011;12(4):611-30.

43. McKay MD, Beckman RJ, Conover WJ. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics. 1979;21(2):239-45.

44. Johnson ME, Moore LM, Ylvisaker D. Minimax and maximin distance designs. Journal of Statistical Planning and Inference. 1990;26(2):131-48.

45. Ye KQ. Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments. Journal of the American Statistical Association. 1998;93(444):1430-9.

46. Carnell R. lhs: Latin Hypercube Samples. 2018 [Available from: https://CRAN.R-project.org/package=lhs] [Accessed 5 Dec 2018].

47. Petelet M, Iooss B, Asserin O, Loredo A. Latin hypercube sampling with inequality constraints. AStA Advances in Statistical Analysis. 2010;94(4):325-39.

48. Levy S, Steinberg DM. Computer experiments: a review. AStA Advances in Statistical Analysis. 2010;94(4):311-24.

49. Loeppky JL, Sacks J, Welch WJ. Choosing the Sample Size of a Computer Experiment: A Practical Guide. Technometrics. 2009;51(4):366-76.

50. Jones DR, Schonlau M, Welch WJ. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization. 1998;13(4):455-92.

51. Crombecq K, Laermans E, Dhaene T. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. European Journal of Operational Research. 2011;214(3):683-96.

52. Jalal H, Dowd B, Sainfort F, Kuntz KM. Linear regression metamodeling as a tool to summarize and present simulation model results. Medical Decision Making. 2013;33(7):880-90.

53. Merz JF, Small MJ, Fischbeck PS. Measuring decision sensitivity: a combined Monte Carlo-logistic regression approach. Medical Decision Making. 1992;12(3):189-96.

54. Briggs AH, Claxton K, Sculpher MJ. Decision Modelling for Health Economic Evaluation. Oxford: Oxford University Press, 2006.

55. Kahn H, Marshall AW. Methods of Reducing Sample Size in Monte Carlo Computations. Journal of the Operations Research Society of America. 1953;1(5):263-78.

56. Murphy DR, Klein RW, Smolen LJ, Klein TM, Roberts SD. Using Common Random Numbers in Health Care Cost-Effectiveness Simulation Modeling. Health Services Research. 2013;48(4):1508-25.

57. Gano S, Kim H, Brown D. Comparison of Three Surrogate Modeling Techniques: Datascape, Kriging, and Second Order Regression. 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. American Institute of Aeronautics and Astronautics. 2006.

58. Kleijnen JPC, Sargent R. A Methodology for Fitting and Validating Metamodels in Simulation. European Journal of Operational Research. 2000;120(1):14-29.59.

59. Alison C, V. SN, C. MD. Learning surrogate models for simulation-based optimization. AIChE Journal. 2014;60(6):2211-27.

60. Wilson ZT, Sahinidis NV. The ALAMO approach to machine learning. Computers & Chemical Engineering. 2017;106:785-95.

61. de Aguiar PF, Bourguignon B, Khots MS, Massart DL, Phan-Than-Luu R. D-optimal designs. Chemometrics and Intelligent Laboratory Systems. 1995;30(2):199-210.

62. Kucherenko S, Albrecht D, Saltelli A. Exploring multi-dimensional spaces: a Comparison of Latin Hypercube and Quasi Monte Carlo Sampling Techniques. 2015 [Available from: https://arxiv.org/abs/1505.02350] [Accessed 20 June 2018].

63. Soboĺ IM. Quasi-Monte Carlo methods. Progress in Nuclear Energy. 1990;24(1):55-61.

64. Lenth RV. Response-Surface Methods in R, Using rsm. Journal of Statistical Software. 2009;32(7):1-17.

65. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. Scikit-learn: Machine Learning in Python. 2011;12:2825-2830.

66. Flasch O, Mersmann O, Bartz-Beielstein T, Stork J, Zaefferer M. rgp: R genetic programming framework. 2014 [Available from: https://cran.r-project.org/src/contrib/Archive/rgp/] [Accessed 5 Dec 2018].

67. Fusting C. fastsr: fast symbolic regression powered by genetic programming. 2017 [Available from: https://pypi.org/project/fastsr/] [Accessed 25 July 2019].

68. Fortin F-A, De Rainville F-M, Gardner M-A, Parizeau M, Gagné C. DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research.2012;13:2171-2175.

69. Milborrow S. earth: Multivariate Adaptive Regression Splines. 2017 [Available from: https://CRAN.R-project.org/package=earth] [Accessed 5 Dec 2018].

70. Hastie T, Tibshirani R. mda: Mixture and Flexible Discriminant Analysis. 2017 [Available from: https://CRAN.R-project.org/package=mda] [Accessed 5 Dec 2018].

71. Rudy J. A Python implementation of Jerome Friedman's Multivariate Adaptive Regression Splines. 2013 [Available from: http://contrib.scikit-learn.org/py-earth/] [Accessed 25 July 2019].

72. Hastie T. gam: Generalized Additive Models. 2018 [Available from: https://CRAN.R-project.org/package=gam] [Accessed 5 Dec 2018].

73. Wood S. mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation. 2018 [Available from: https://cran.r-project.org/web/packages/mgcv/mgcv.pdf] [Accessed 5 Dec 2018].

74. Servén D., Brummitt C. (2018). pyGAM: Generalized Additive Models in Python. Zenodo. DOI: 10.5281/zenodo.1208723

75. MacDonald B, Ranjan P, Chipman H. {GPfit}: An {R} Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs. Journal of Statistical Software. 2015;64(12):1-23.

76. Gramacy RB. tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models. Journal of Statistical Software. 2007;19(9):1-46.

77. Gramacy RB, Taddy M. Categorical Inputs, Sensitivity Analysis, Optimization and Importance Tempering with tgp Version 2, an R Package for Treed Gaussian Process Models. Journal of Statistical Software. 2010;33(6):1-48.

78. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Pyhton. Journal of Machine Learning Research. 2011;12:2825−2830.

79. de G. Matthews AG, van der Wilk M, Nickson T, Fujii K, Boukouvalas A, León-Villagrá Pablo, et al. GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research. 2017;18(40):1-6.

80. Fritsch S, Guenther F. neuralnet: Training of Neural Networks. 2016 [Available from: https://CRAN.R-project.org/package=neuralnet] [Accessed 5 Dec 2018].

81. Venables WN, Ripley BD. Modern Applied Statistics with S. New York: Springer; 2002.

82. Chollet F, et al. Keras. 2015 [Available from: https://keras.io] [Accessed 25 July 2019].

83. Shevchuk Y. NeuPy: Neural Networks in Python. 2019 [Available from: http://neupy.com/pages/home.html] [Accessed 25 July 2019].

Figure 1. Illustration of how metamodels can be used in a health economic context to approximate the outcomes of the original health economic simulation model.

Figure 2. Process for developing, validating, and applying metamodeling methods in health economics.

Figure 3. Flowchart for the selection of appropriate metamodeling techniques for a specific case study.

Figure 4. Illustration of how a random uniform sample, full factorial design, and maximin Latin Hypercube sample may define nine experiments for two continuous parameters Test Cost and Consultation Cost.

FIGURE 1

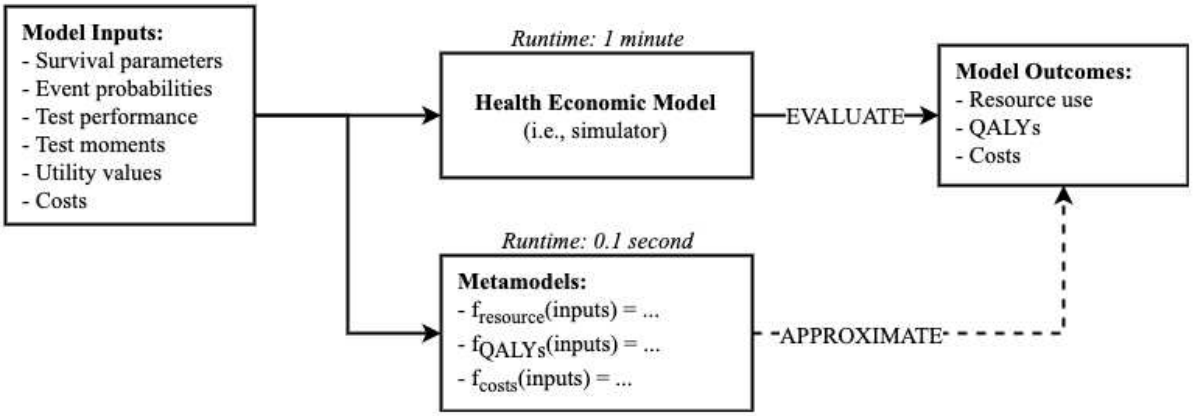
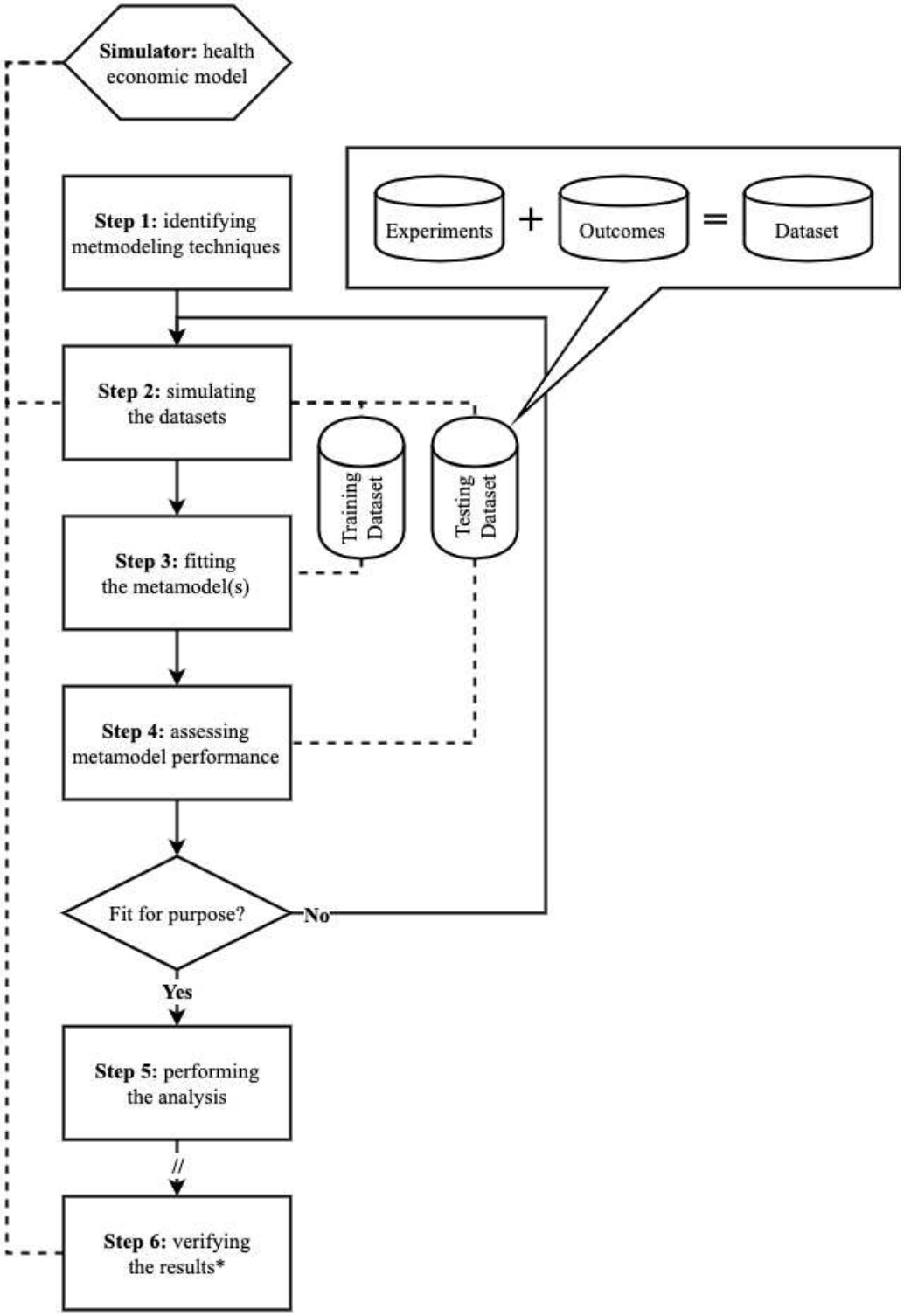


FIGURE 2



* Optional, depending on the analysis performed in Step 5.

FIGURE 3

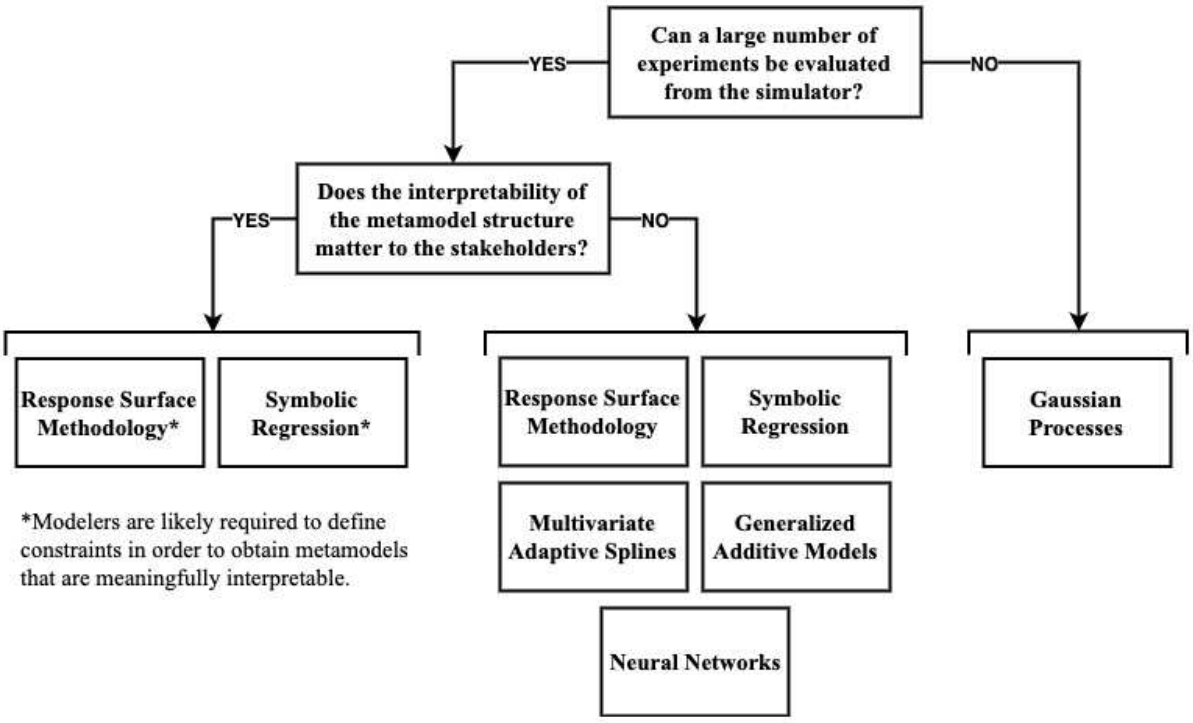

Can a large number of experiments be evaluated from the simulator?
YES
NO
Does the interpretability of the metamodel structure matter to the stakeholders?
YES
NO
Response Surface Methodology*
Symbolic Regression*
Response Surface Methodology
Symbolic Regression
Gaussian Processes
Multivariate Adaptive Splines
Generalized Additive Models
Neural Networks
*Modelers are likely required to define constraints in order to obtain metamodels that are meaningfully interpretable.

FIGURE 4

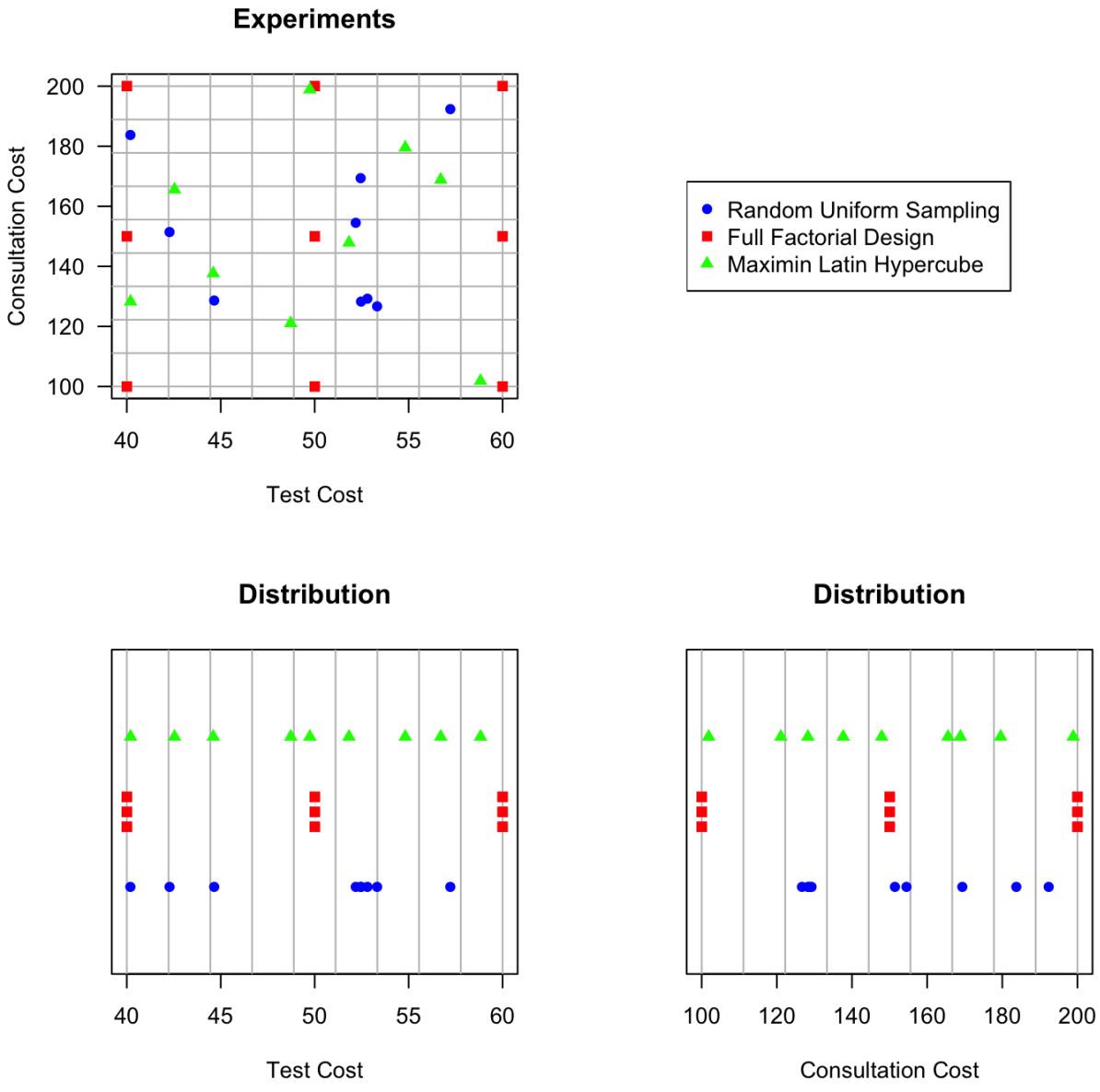

Experiments
Consultation Cost
200
180
160
140
120
100
40
45
50
55
60
Test Cost
Random Uniform Sampling
Full Factorial Design
Maximin Latin Hypercube
Distribution
40
45
50
55
60
Test Cost
Distribution
100
120
140
160
180
200
Consultation Cost

Table 1. Overview of candidate metamodeling techniques for application in health economics, which are all able to account for mixed input parameters and continuous outcomes.

Technique	Required Number of Experiments	Number of Inputs	Interpretability	R Package	Python Package	References
Response Surface Methodology	High	Large	Moderate	`rsm` (64)	`sklearn` (65)	(10, 26, 27)
Symbolic Regression	High	Large	Moderate	`RGP`* (66)	`fastsr` (67), `DEAP` (68)	(28, 29)
Multivariate Adaptive Regression Splines	High	Large	Low	`earth` (69), `mda` (70)	`py-earth` (71)	(10, 30, 31)
Generalized Additive Models	High	Large	Low	`gam` (72), `mgcv` (34, 73)	`pyGAM` (74)	(33, 34)
Gaussian Processes	Low	Low	Low	`GPfit` (75), `tgp` (76, 77)	`scikit-learn` (78), `GPflow` (79)	(10, 35)
Neural Networks	High	Large	Low	`Neuralnet` (80), `nnet` (81)	`keras` (82), `NeuPy` (83)	(10, 31, 38)

* No longer maintained by authors.

Appendix A: hypothetical illustration on performing value of information analysis

Consider the example used in the introduction of the manuscript in which a discrete event simulation model had been developed to estimate the health economic impact of a novel cancer drug compared to an existing drug. Performing one simulation run using this model required around 1 minute. An expected value of perfect parameter information (EVPPI) analysis was to be performed to answer the question "What is the value of collecting additional evidence on this single subgroup of model parameters?". Based on assessing the stability of modeling outcomes, the EVPPI was to be estimated by performing an inner probabilistic analysis simulation loop of 5,000 runs and outer simulation loop of 2,500 runs, resulting in a total of 12.5 million required simulation runs. To perform this analysis, a metamodel was developed to approximate the net monetary benefit (NMB) based on the subgroup of 10 model input parameters.

Step 1: identifying candidate metamodeling techniques

Despite the runtime of 1 minute, a relatively large number of experiments could be evaluated from the simulator. Hence, it was not necessary to use Gaussian Processes. As the only purpose of developing the metamodel was to perform the EVPPI analysis, the interpretability of the metamodel structure was not an issue and there was no specific reason to use response surface methodology or symbolic regression. Finally, because the modeler was familiar with Generalized Additive Models, this technique was selected (see Figure A1).

Step 2: simulating datasets

Because there were no specific constraints on (combinations of) model input parameters, a Latin Hypercube design was used for both the training dataset and testing dataset. Using the rule of thumb to start with $n = 10 \times d$ experiments, the initial training dataset was generated comprising 100 experiments (here, $d = 10$). To assess the accuracy of the metamodel in Step 4, a testing dataset of size $n = 20$ was generated. The net monetary benefit for all experiments in these two sets was evaluated with the simulator, i.e. discrete event simulation model. However, after fitting a metamodel in Step 3 based on 100 experiments, its performance was considered insufficient in Step 4. Therefore, the number of experiments was increased to 1000, which could still be evaluated with the simulator within a feasible timeframe, and the testing dataset size was increased to 200. This number of experiments resulted in a sufficiently accurate metamodel (Step 4).

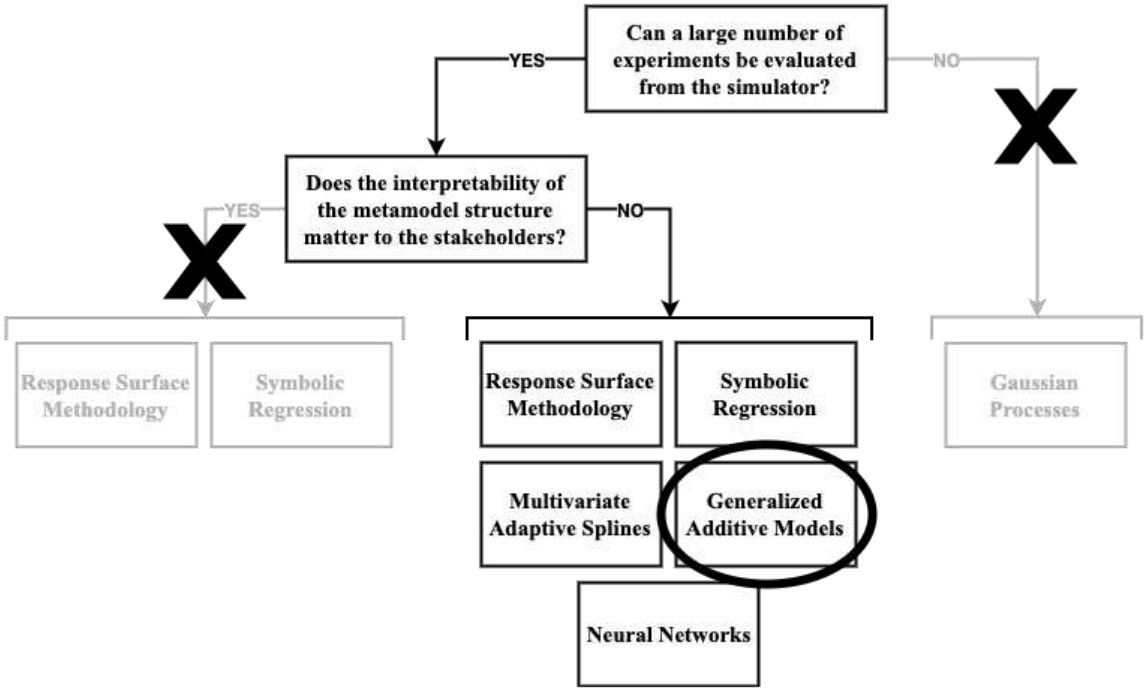


Figure A1. Illustration of the selection of the metamodeling technique for the hypothetical illustration.

Step 3: fitting metamodels

The metamodel that approximates the net monetary benefit was fitted in R using the gam function of the `gam` package, using all 10 model input parameters as input variables and the net monetary benefit as output variable.

Step 4: assessing metamodel performance

A validation plot and the mean absolute error and mean squared error were used to determine the accuracy with which the metamodel approximates the outcomes of the discrete event simulation (i.e., simulator). Figure A2 presents validation plots, including the two performance measures, for the metamodel that was fitted based on a training dataset of size 100 (i.e., $n = 100$ experiments) and the one fitted based on $n = 1000$ experiments. Clearly, the performance of the metamodel fitted based on 100 experiments was insufficient, as predictions on average were 13.67 below or above the values observed from the simulator (mean absolute error), which is more than 9% on average. The metamodel fitted to 1000 experiments, on the other hand, approximated the outcomes of the simulator very accurately and was considered appropriate to replace that simulator in the EVPPI (Step 5).

Step 5: applying metamodel

After the metamodel had been considered sufficiently accurate to substitute the discrete event simulation model, the EVPPI analysis was performed. All the 12.5 million simulations were performed in the same way as they would have been using the discrete event simulation model, but now the metamodel replaced the discrete event simulation model within the inner simulation loop to obtain the net monetary benefit estimates for specific combinations of input parameter values.

Step 6: verifying results (optional)

No final verification was necessary after performing the EVPPI analysis using the metamodel, as this would not have resulted in any additional insights than those obtained during the validation of the metamodel (Step 4). If the metamodel had been used to apply an optimization algorithm, the best performing sets of input parameters obtained from the optimization should be verified using the simulator, to check whether the rank of these scenarios meaningfully differed when evaluated using the simulator and, thereby, establish additional confidence in the optimization outcomes.

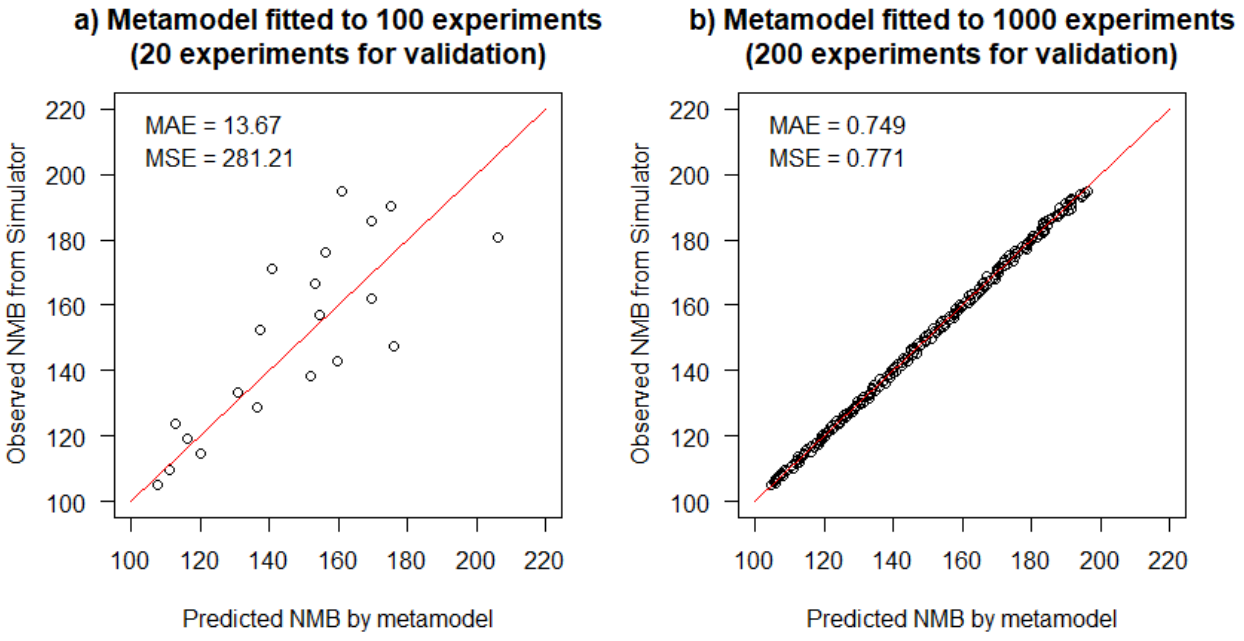


Figure A2. Validation plots for the net monetary benefit (NMB) as observed from the simulator and predicted by the metamodels fitted based on a) 100 experiments and b) 1000 experiments, including the mean absolute error (MAE) and mean squared error (MSE).