Choosing a Metamodel of a Simulation Model for Uncertainty Quantification

Tiago M. de Carvalho, Joost van Rosmalen, Harold B. Wolff, Hendrik Koffijberg, and Veerle M. H. Coupé

Background. Metamodeling may substantially reduce the computational expense of individual-level state transition simulation models (IL-STM) for calibration, uncertainty quantification, and health policy evaluation. However, because of the lack of guidance and readily available computer code, metamodels are still not widely used in health economics and public health. In this study, we provide guidance on how to choose a metamodel for uncertainty quantification.

Methods. We built a simulation study to evaluate the prediction accuracy and computational expense of metamodels for uncertainty quantification using life-years gained (LYG) by treatment as the IL-STM outcome. We analyzed how metamodel accuracy changes with the characteristics of the simulation model using a linear model (LM), Gaussian process regression (GP), generalized additive models (GAMs), and artificial neural networks (ANNs). Finally, we tested these metamodels in a case study consisting of a probabilistic analysis of a lung cancer IL-STM.

Results. In a scenario with low uncertainty in model parameters (i.e., small confidence interval), sufficient numbers of simulated life histories, and simulation model runs, commonly used metamodels (LM, ANNs, GAMs, and GP) have similar, good accuracy, with errors smaller than 1% for predicting LYG. With a higher level of uncertainty in model parameters, the prediction accuracy of GP and ANN is superior to LM. In the case study, we found that in the worst case, the best metamodel had an error of about 2.1%.

Conclusion. To obtain good prediction accuracy, in an efficient way, we recommend starting with LM, and if the resulting accuracy is insufficient, we recommend trying ANNs and eventually also GP regression.

Keywords
cost-effectiveness analysis, metamodels/emulators, probabilistic sensitivity analyses, simulation models, uncertainty quantification

Date received: July 7, 2020; accepted: April 5, 2021

Decision models are routinely used in health economics and public health for the purpose of evaluating the harms and benefits of competing health interventions. The results of these models may be used to inform clinical guidelines, optimize health care resources, or guide public health policies. An individual-level state transition simulation model (IL-STM) is a type of decision model in which individual life histories are simulated through multiple health states. These models are flexible enough to be used for a wide range of policy evaluations in complex decision problems. However, reflecting individual characteristics comes at a cost of increased model complexity, resulting in a high number of model parameters. In addition, it may be necessary to simulate thousands or even millions of life histories to minimize stochastic uncertainty and obtain stable model outcomes.

Several tasks could then easily become computationally expensive, namely, 1) calibration, due to a large number of required IL-STM runs; 2) policy evaluation, if many treatment regimens or screening policies are evaluated; and 3) uncertainty quantification, in the form of

Corresponding Author
Tiago M. de Carvalho, Department of Epidemiology and Biostatistics, Amsterdam UMC, Location VUMC, Amsterdam, the Netherlands; (tm.decarvalho@amsterdamumc.nl).
a probabilistic analysis (PA)\(^1\) (also known as probabilistic sensitivity analyses) or value-of-information analyses (VOI).\(^4\) For instance, Rutter et al.\(^5\) performed 180,000 model runs (2 million life histories) to calibrate their model parameters using a Markov chain Monte Carlo algorithm, van Hees et al.\(^6\) performed 19,200 model runs (10 million life histories) for policy analyses of colorectal cancer screening, and VOI may require an analyst to run the model more than 100,000 times.\(^6\)

This has led to the development of metamodels or emulators, regression models for the relationship between the IL-STM parameters and outcomes, which are computationally inexpensive to train and run.\(^7,8\) The main purpose of metamodeling is to substantially decrease the amount of computation time needed to perform calculations with the IL-STM by replacing it with a fast approximation (metamodel), which requires a small number of IL-STM runs to train. Previous studies in health economics claimed that metamodels could reduce the computational expense of the analyses between 85%\(^9\) and more than 99%,\(^10\) and this has been named as a “key area for further research work” in 2018 by the Second Panel on Cost-effectiveness in Health and Medicine.\(^11\) Emulators have been applied for multiple tasks within decision modeling in health economics, including model calibration, PA, VOI, inference of parameter influence, and resource optimization\(^9,12–14\); however, they are still not routinely used.\(^8\)

The routine use of emulators is hindered by a lack of guidance and training on how to perform emulation in an efficient way.\(^11\) A key step of metamodeling is the choice of statistical model. Common choices include linear regression and Gaussian process regression (GP). GP is widely used in the engineering literature, as it can handle multiple shapes of smooth curves. On the other hand, GP has the limitation\(^15\) that if the number of parameters is relatively large (>15), the model fit will become relatively slow. These facts motivated the use of alternative nonlinear statistical models, including generalized additive models (GAMs) and artificial neural networks (ANNs).\(^12,16\) However, there is not yet any formal comparison between these statistical models, and hence, researchers may adopt a suboptimal type of metamodel, compromising the runtime and/or accuracy of their analyses.

The goal of this study is to provide guidance on choosing an accurate metamodel with a fast computation time for uncertainty quantification using R. In particular, we focus on PA, which reflects the uncertainty in IL-STM outcomes caused by uncertainty in IL-STM parameters by repeatedly drawing parameter values from relevant sampling distributions with Monte Carlo simulation and generating an empirical distribution for the IL-STM outcomes.

We first survey previously used metamodels in health economics and public health and corresponding R packages. We then evaluate each R package with respect to the computational speed to fit the metamodel and prediction accuracy for PA in a simulation study. We investigate the role of IL-STM characteristics, related to the level of parameter uncertainty (size of the 95% confidence interval or uncertainty interval), number of model runs, number of simulated life histories, low/high rate of health state transitions, and Markov/semi-Markov assumption, on the choice of the most accurate model. Finally, we apply the metamodels in a case study concerning a PA for a cost-effectiveness analysis comparing 2 treatments for stage I non–small-cell lung cancer.\(^17\)

**Methods**

**Metamodeling Steps for Uncertainty Quantification**

In Supplementary Figure S1, we show the steps of metamodeling for uncertainty quantification.

**Training data requirements.** Before starting the analysis, there are at least 2 prerequisites related to minimizing the amount of first-order uncertainty (stochastic uncertainty) to ensure a reasonable metamodel prediction accuracy: 1) the number of simulated life histories by the IL-STM should be large enough, to obtain stable outcomes with low simulation error, and 2) in each simulation run, common random numbers should be used, that is, each simulation run should use the same seed number for the random generator. Each parameter included in the analyses should have a corresponding sampling distribution.\(^9\) To reduce the computational expense associated with generating the training data or fitting the metamodel, the researcher could use prior knowledge about the model behavior and/or variable selection techniques to select the model parameters with the strongest effect on the model outcomes and leave out of the metamodel the parameters with little to no impact.\(^18,19\)
Examples of variable selection techniques include, among others, running a linear regression and selecting only the variables with a $P$ value below a certain threshold, least absolute shrinkage, and selection operator regression or computing variable importance measures after fitting a random forest model.

**Generate training data.** Let $y$ be a vector of $S$ simulation model runs denoting a particular model outcome. Each element, $y_s$, is the result of running the IL-STM given model parameter values $x_{sk}, s = 1, \ldots, S; k = 1, \ldots, P$, where $P$ is the number of IL-STM parameters. Let $X$ be the matrix with elements $x_{sk}$. The first step of the metamodeling process is to choose matrix $X$ (i.e., the values of the IL-STM parameters where the IL-STM should be run). For this, it is common practice to use Latin hypercube sampling (LHS).8 We first define the region of the parameter space where we are interested in applying the metamodel. Because this is a PA, the upper and lower limits of the region of interest could be based on the 95% confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter. Alternatively, we could choose a wider interval (0.1th and 99.9th confidence interval of each model parameter.

This is implemented by the command `maximinlhs` from the R package `lhs`,20 which generates the samples in $[0, 1]^p$ scale, and the function `qunif` from base R, which rescales the values of the model inputs back to its original scale.

**Choice of regression model for the emulator.** The relationship between simulator input parameters and outcomes may be characterized by a high degree of nonlinearity and possibly multiple interactions and correlations between model parameters. However, for a PA, we are usually interested in only a small subset of the whole parameter space, based on the distribution of each model parameter. If the associated confidence intervals are relatively small and/or if the relationship between model parameters and outcomes is relatively simple, then even linear regression could be a good approximation to the effect of model inputs on the outcomes. Alternatively, we may obtain a better fit by using nonlinear models. For instance, GP regression is a flexible model that makes metamodel predictions by interpolating between design points, GAMs allow for nonlinear relationships by including smooth functions of model parameters, and ANNs allow for arbitrary complex nonlinear relationships.

In Table 1, we review previously used statistical models as metamodels in health economics. Our search strategy consisted of combining the results of a February 2018 review on metamodeling in which one of the authors participated (H.K.)8 and our own literature search on PubMed to capture articles published between 2018 and January 2020 and/or studies that were missed in the review. We found 5 studies9,21–24 of 300 PubMed search results (search took place in March 2020; see Supplementary Figure S2) in addition to the 13 studies found in Degeling et al.8 The most common choices in health economics are linear regression (LM) and Gaussian GP, with a squared exponential covariance matrix.8 Alternative choices include GAMs16,25 ANNs,12,14 GP using Matern and rational quadratic covariance matrix,22 and symbolic regression.26 Although symbolic regression is valuable because it does not assume any prior model structure, it is relatively more difficult to implement and therefore was excluded from this study (Table 1).

"Classical" models for emulation

**Linear regression and GP.** A simple metamodel could be built by assuming that the model outcome of interest is well approximated by a linear function of the model parameters,

$$y = \beta_0 + \sum_{k=1}^{P} x_k \beta_k + \epsilon,$$

where, $\epsilon \sim N(0, \sigma^2 I)$, and $I$ is the identity matrix.

However, the linear model could be a poor approximation of the relationship between $y$ and $X$ because of the possible nonlinearities in the simulation model. We could consider instead the following,

$$y = \beta_0 + \sum_{k=1}^{P} f_k(x_k) \beta_k + \epsilon(X),$$

where $\epsilon(X)$ is a Gaussian process, with $E(\epsilon(X)) = 0$, and covariance $V(x_r, x_s) = \sigma^2 C(x_r, x_s)$.30 In this study, $f_k(.)$ is just the identity function ($f_k(x_k) = x_k$ for all $k$); however, other choices are possible. There are several possible forms for $C(.)$.31 The most popular is the squared exponential covariance function, which has $r,s$ entries,

$$C_{rs}(x_r, x_s) = \exp \left( -\sum_{k=1}^{P} \frac{(x_{rk} - x_{sk})^2}{\phi_k^2} \right).$$

The key parameter here is vector $\phi$, which models how fast the correlation decays as the distance between the pair $(x_r, x_s)$ increases. In this parametrization, lower estimated values of $\phi_k$ ($< 1$) mean that the simulation model
outcome is highly sensitive to changes in the \( k \)-th model parameter. Parameters \( \beta, \sigma^2 \) and \( \phi \) are estimated iteratively via maximum likelihood.

The R packages used in this study contain different implementations of GP: \texttt{kernlab},\(^{32}\) which allows for a choice between multiple covariance matrices but estimates only a single correlation length parameter, \texttt{tgp}\(^{33}\) adopts a Bayesian approach and \texttt{GPfit}\(^{34}\) includes the estimation of a nugget term in \( C(\cdot) \) for numerical stability.

**Alternative choices for emulator/metamodel**

**Generalized additive model.** Another way to extend the linear regression would be to consider a GAM,

\[
y = \beta_0 + \sum_{k=1}^{p} f_k(x_k) + \epsilon.
\]

where, \( f_k(x_k) = \sum_{j=1}^{J} \beta_{kj} B_{kj}(x_k) \), the \( B_{kj}(\cdot) \) are known basis functions and \( J \) is the degree of the spline. In this study, all R package implementations use \( b \)-splines as the basis functions\(^{35-37}\); however, other choices are also possible\(^{35,38}\).

**Artificial neural networks.** ANN is a nonlinear regression model suitable to model highly complex nonlinear relationships. Each ANN consists of the combination of neurons, layers, and an activation function\(^{39}\). In this study, we consider only ANNs with a single hidden layer \( f^{(1)}(\cdot) \),

\[
y = b_0 + w^{T} f^{(1)}(x, W, b_1) + \epsilon,
\]

where \( W \) is a \( mn \) by \( P \) weight matrix, \( x \) is a \( P1 \) vector, \( w \) and \( b_1 \) are \( mn \) by 1 vectors, and \( mn \) is the number of neurons included in the network. Two commonly used

---

### Table 1  Review of Statistical Models Used as a Metamodel in Health Economics and Corresponding R Packages\(^{a,b}\)

<table>
<thead>
<tr>
<th>Metamodels</th>
<th>Studies</th>
<th>Objectives</th>
<th>R Package</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear model</td>
<td>Jalal et al. (2013)(^{27})</td>
<td>Perform a sensitivity analysis</td>
<td>\texttt{lm}</td>
<td>Classic squared exponential covariance matrix; includes a nugget term.</td>
</tr>
<tr>
<td>Gaussian process regression</td>
<td>Stevenson et al. (2004)(^{10})</td>
<td>Probabilistic analysis, multiobjective calibration</td>
<td>\texttt{GPfit}</td>
<td>Isotropic (single correlation length parameter), multiple covariance matrices available.</td>
</tr>
<tr>
<td></td>
<td>de Carvalho et al. (2019)(^{9})</td>
<td></td>
<td>\texttt{kernlab}</td>
<td>Estimation via Monte Carlo Markov Chain sampling</td>
</tr>
<tr>
<td></td>
<td>Sai et al. (2019)(^{22})</td>
<td></td>
<td>\texttt{tgp}</td>
<td>Estimation via iteratively reweighted least squares</td>
</tr>
<tr>
<td>Generalized additive models</td>
<td>Jalal and Alarid-Escudero (2018)(^{16})</td>
<td>Calculate expected value of sampling information</td>
<td>\texttt{gam}</td>
<td>Estimation via penalized maximum likelihood</td>
</tr>
<tr>
<td></td>
<td>Strong et al. (2015)(^{25})</td>
<td></td>
<td>\texttt{mgcv}</td>
<td>Estimation via gradient boosting</td>
</tr>
<tr>
<td>Artificial neural networks</td>
<td>Kilmer et al. (1997)(^{24})</td>
<td>Emergency department simulation, optimization of resource allocation</td>
<td>\texttt{mboost}</td>
<td>Single-layer neural networks</td>
</tr>
<tr>
<td></td>
<td>Yousefi et al. (2018)(^{14})</td>
<td></td>
<td>\texttt{nnet}</td>
<td>Neural networks with multiple layers</td>
</tr>
<tr>
<td>Symbolic regression(^{b})</td>
<td>Willem et al. (2014)(^{26})</td>
<td>Iterative active learning and uncertainty quantification</td>
<td>\texttt{neuralnet}</td>
<td>Symbolic regression with genetic programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>\texttt{rGP}</td>
<td>Symbolic regression with grammar evolution(^{28})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>\texttt{gramEvol}</td>
<td></td>
</tr>
</tbody>
</table>

\(^{a}\)This review is based on Degeling et al.\(^{8}\) Some studies used multiple statistical models as a metamodel in health economics (linear model, neural networks, and Gaussian process regression) including Cevik et al. (2015),\(^{12}\) Tappenden et al. (2004),\(^{45}\) and Alam and Briggs (2019).\(^{21}\) In Tappenden et al.,\(^{45}\) the metamodels were used for value-of-information analyses. In Cevik et al.,\(^{12}\) the metamodels were used for calibration. In the study by Alam and Briggs,\(^{21}\) the metamodels were used for sensitivity analyses.

\(^{b}\)While 2 R packages were found for this model, 1 is removed from the R package main repository (last checked April 2020; \texttt{rgp}) and the other (\texttt{gramEvol}) would require using an additional R package to fit a proper metamodel.\(^{28}\)
activation functions \( f^{(1)}(.) \) are the rectified linear unit and logistic (also called sigmoid) functions,\(^{40}\)
\[
y_s = b_0 + w' \max(0, W x_s + b_1) + \epsilon_s, \quad y_s = b_0 + w' \sigma(W x_s + b_1) + \epsilon_s,
\]
\[\sigma(z) = \frac{1}{1 + \exp(z)} \text{ and } 0 \text{ is a } m \times 1 \text{ vector of zeroes.} \]

In this study, both R packages \texttt{nnet}\(^{40}\) and \texttt{neuralnet}\(^{41}\) use a logistic activation function.

**Metamodel validation.** Before using the metamodel, the analyst should verify whether it produces an accurate prediction of the IL-STM outcome. One could validate the metamodel by using a training-test set approach, in which the decision model is run for an additional set of parameter values \( x_i \) not included in the training data, with the resulting outcomes \( y_i \), then used to evaluate the prediction error of the metamodel. However, if the decision model is relatively slow to run, it is likely more computationally efficient to use cross-validation.

This method requires only a single training data set and requires less IL-STM runs, as the metamodels are fitted multiple times to subsets of the training data and the remaining data are used as the test set. The drawback of this method, as compared with the training-test set approach, is that the prediction error will be slightly overestimated, because we are not fitting the metamodel to the full training data.

**Simulation Study: Prediction Accuracy and Computational Speed of Statistical Models and Their Corresponding R Packages**

**Aims.** We evaluate the accuracy of metamodels when the goal is to carry out a PA, given different IL-STM characteristics. We also evaluate the computation time of each R package used, to prevent usage of inefficient packages (i.e., with a computation time much longer than other packages with similar or higher accuracy; Table 1).

**Data-generating process: IL-STM.** We build a simple continuous-time microsimulation multistate model with 5 health states and 10 parameters in total (see Supplementary Table S1). The health states correspond to 1, healthy; 2, mild disease; 3, severe disease; 4, disease-specific death; 5, other-cause death. The main outcome is life-years gained due to treatment. This is obtained by running the IL-STM with and without an effect of treatment. The simulation model is governed by a matrix of transition hazards \( Q \).

**Evaluation of metamodel accuracy**

**Metamodeling scenarios.** We evaluate how metamodel accuracy and choice may change under different commonly observed scenarios in health economic modeling. In Table 2, we describe the different scenarios. The exact details of the simulation settings are shown in Supplementary Table S2. We examine the following metamodeling characteristics: 1) level of uncertainty in model parameters: for the transition hazards, this is based on sample size of the individual-level data sets \( N \) used for parameter estimation, for the treatment effects the level of uncertainty is randomly sampled with higher uncertainty bounds for smaller \( N \); life histories simulated \( M \) during training; number of simulation training runs \( S \), and b) for a fixed total number of simulation model runs \( M \times S \), the effect of the choice of \( M \) and \( S \). We also examine the effects of the following simulation model characteristics: 3) disease-specific mortality under treatment as model outcome; 4) parameter values, namely, high versus low transition rates; 5) high parameter uncertainty or little information about the parameter values;
3) Model outcomes. We evaluate the metamodel accuracy as in the base case, using the disease-specific training data by running the simulation model with and \( S \) parameter during PA. Inside the upper and lower bounds for the distribution of each existing literature. These confidence intervals give the extracting parameter values and distributions from the sampled (Supplementary Table S2). This is analogous to LHS, and their confidence intervals are randomly sampled (Supplementary Table S2). This is analogous to a data set from a clinical trial. We use an individual-level data set of observed disease progression once to generate each model parameter is allowed to vary during PA.

4) High and low transition rates. Same parameter values for each iteration \((d = 1, \ldots, D)\). In the first scenario (ParamV1), \( \beta_{1,12} = \beta_{1,23} = -3.5\) (long duration, few transitions). In the second scenario (ParamV2), \( \beta_{1,12} = \beta_{1,23} = -1\) (short duration, many transitions). See Supplementary Table S1, “ParamV1” and “ParamV2” rows, for details.

5) High parameter uncertainty. This scenario is analogous to a situation in which we need to calibrate the IL-STM parameters. Parameter values are sampled as usual; however, parameter uncertainty is not based on an estimated confidence interval but given by the min/max rows of Supplementary Table S1. For instance, the confidence interval for \( \beta_{1,12} \) ranges between 0 and \(-3.5\). Then, setting all covariates equal to zero, the average duration to state 2 ranges between 1 and 33 years.

6) Semi-Markov. Durations between health states are generated based on a Gompertz distribution instead of an exponential distribution. See Supplementary Table S1 for the model parameter range.

**Table 2** Modeling Scenarios Used to Evaluate Metamodel Accuracy

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base case</td>
<td>Data set used to estimate the simulation model parameters has ( N = 5000 ) individuals. To build the training data set, we perform ( S = 100 ) simulation model runs, each simulating ( M = 10,000 ) life histories. All steps are repeated ( D = 50 ) times. The main model outcome is life-years gained due to treatment (per person). See Supplementary Table S2 for basic simulation settings.</td>
</tr>
<tr>
<td>1) Metamodeling characteristics</td>
<td>Certain characteristics of the metamodeling situation may affect accuracy. We changed ( N ) from 5000 to 500 and 100, ( S ) from 100 to 50, and ( M ) from 10,000 to 1000.</td>
</tr>
<tr>
<td>2) Choosing ( M ) and ( S )</td>
<td>Instead of simulating ( M = 10,000 ) life histories and ( S = 100 ) model runs, with the same computational budget for simulation, we could do instead ( M = 2000 ) life histories and ( S = 500 ) model runs.</td>
</tr>
<tr>
<td>3) Model outcomes</td>
<td>We evaluate the metamodel accuracy as in the base case, using the disease-specific mortality rate under treatment as model outcome.</td>
</tr>
<tr>
<td>4) High and low transition rates</td>
<td>Same parameter values for each iteration ((d = 1, \ldots, D)). In the first scenario (ParamV1), ( \beta_{1,12} = \beta_{1,23} = -3.5) (long duration, few transitions). In the second scenario (ParamV2), ( \beta_{1,12} = \beta_{1,23} = -1) (short duration, many transitions). See Supplementary Table S1, “ParamV1” and “ParamV2” rows, for details.</td>
</tr>
<tr>
<td>5) High parameter uncertainty</td>
<td>This scenario is analogous to a situation in which we need to calibrate the IL-STM parameters. Parameter values are sampled as usual; however, parameter uncertainty is not based on an estimated confidence interval but given by the min/max rows of Supplementary Table S1. For instance, the confidence interval for ( \beta_{1,12} ) ranges between 0 and (-3.5). Then, setting all covariates equal to zero, the average duration to state 2 ranges between 1 and 33 years.</td>
</tr>
<tr>
<td>6) Semi-Markov</td>
<td>Durations between health states are generated based on a Gompertz distribution instead of an exponential distribution. See Supplementary Table S1 for the model parameter range.</td>
</tr>
</tbody>
</table>

IL-STM, individual-level state transition simulation model.

*See Supplementary Table S1 for the range of simulated model parameters and Supplementary Table S2 for the basic simulation settings.*

6) semi-Markov (Gompertz distribution), instead of Markov (exponential distribution) assumption for the transition hazards.

**Simulation procedure.** The simulation study has 2 phases. First, we generate the uncertainty range in which each model parameter is allowed to vary during PA. Then, we use this range to simulate training data to fit and validate the metamodels (Figure 1). The algorithm for the simulation is described in Box 1.

The first step is to sample microsimulation model parameter values (\( \theta_d \)). Then, we run the microsimulation model described in the Methods section once to generate an individual-level data set of observed disease progression analogous to a data set from a clinical trial. We use this data set to estimate \( \beta_d \) and extract their confidence intervals. The treatment effects \( \lambda_d \) are sampled based on LHS, and their confidence intervals are randomly sampled (Supplementary Table S2). This is analogous to extracting parameter values and distributions from the existing literature. These confidence intervals give the upper and lower bounds for the distribution of each parameter during PA. Inside the \( S \) loop, we generate training data by running the simulation model with and without treatment effect. Inside the \( cvfolds \) loop, we perform cross-validation by leaving a part of the training data (\( cv \)) out of the estimation and use it to compute the prediction error.

**Statistical models/R packages.** The R packages to be tested are shown in Table 1 and correspond to previously used metamodels in the health economic/public health literature. See Supplementary Table S3 for hyperparameter details. The main source for the R packages used is the caret package.\(^46\) We included between 2 and 3 packages per statistical model, with a goal of including, as far as possible, distinct implementations of each statistical model. This list is not meant to be exhaustive and, in particular for ANNs, there are additional packages included in caret that could also have been used. On the other hand, we included 3 models not previously used in health economics, which are popular in machine-learning literature\(^38\): boosting, random forests with decision trees, and Bayesian networks. If the expected time per model fit of a single cross-validation iteration is higher than 300 s, we exclude the package from the particular scenario.

**Performance measures.** We evaluate the model accuracy by computing the root mean square error (RMSE)
divided by the range of the model outcome and percentage absolute error (PAE) over 50 data sets. We also show the mean absolute error and the mean square error. For each cross-validation iteration, we left 10% of the data out of the training sample in each iteration.

Evaluation of metamodel computation time. We fit the metamodels to a simulated single training data set sample \((y, X)\), including 5 parameters \((S = 50, 100\) simulation runs) and 10 parameters \((S = 100, 200)\). The computational expense of fitting a metamodel to training data is measured using the R package `microbenchmark`. We perform 5 repetitions and report the average fitting time.

All computations were run using an Intel® Xeon® CPU E5-2680 2.5GHz processor.

Case Study: PA for Cost-Effectiveness Analysis of 2 Treatments for Stage I Non–Small-Cell Lung Cancer

The case study is based on Wolff et al. (2020). This is a modeling study comparing the cost-effectiveness of 2 treatments for stage I non–small-cell lung cancer (NSCLC), stereotactic body radiation therapy (SBRT), and video-assisted thoracic surgery (VATS). This study contained a PA that required about 10,000 microsimulation model runs. This could be run in approximately 1 h, and in practice, we would not need to use a metamodel in this example. We use this case study to illustrate how to apply metamodeling to reduce the computation expense of uncertainty quantification by replacing most of the simulation model runs by the metamodel.

The main model outcomes are total discounted costs and total discounted quality-adjusted life-years (QALYs) gained by SBRT and VATS. A flowchart of the microsimulation model is shown in Supplementary Figure S3. For the PA, we include the same 22 model parameters included in the original analyses consisting of tumor growth parameters, probabilities for receiving treatment, excess mortality due to treatment, health utilities, and costs. Parameter values and their confidence intervals are given in Supplementary Table S4. The sample size for training is 10 times the number of included model parameters in the analyses \((S = 220)\). Some parameters were excluded for a specific model outcome if they were clearly not related (e.g., cost of SBRT for total VATS costs). We evaluate the prediction accuracy of the metamodel on the 10,000 microsimulation model runs using the same metamodels as in Table 1 and accuracy measures in the “Performance Measures” section.

Results

Accuracy of Different Statistical Models for Emulation

In Figure 2, we measure the effects of varying \(N, M,\) and \(S\) on metamodel accuracy. In Figure 3, we show alternative modeling scenarios including an alternative model outcome, semi-Markov assumption, and effect of different transition rates. Complete results, including all metamodels, additional measures of prediction error, estimated model parameters, and their confidence intervals, are shown in the online Supplement Appendix C (Supplementary Tables S5–S7). The main model outcome throughout this section (average life-years gained) ranges between 0.02 and 0.47 in the base-case scenario (Table 2; Figure 2). In all scenarios tested, most models had an RMSE divided
by range of the outcome below 0.05 and a PAE below 5%. The exceptions are the high parameter uncertainty scenario (Supplementary Figure S4), in which the prediction accuracy was poor, with a best average PAE of 40% and the best average RMSE close to 0.07, the $N = 100$ scenario (Figure 2), in which most models had a PAE or RMSE greater than 0.05, but in which some nonlinear models performed well (ANN and GP) and random forest (ranger) and Bayesian GP (tgp) models, which had an RMSE greater than 0.05 and PAE greater than 5% in most scenarios (see Supplementary Tables 6 and 7).

In the base-case scenario given in the upper left corner of Figure 2, commonly used metamodels in health economics (LM, GP, GAM, and ANNs) have a similar accuracy (RMSE $= 0.02$ and PAE $< 1\%$). If $M$ reduces from 10,000 to 1000 life histories, the RMSE for the best metamodels (LM, GAM, and ANNs) increases from 0.02 to 0.04 and the PAE increases from 0.7% to 1.2%. With $N = 500$ ($N = 100$), nonlinear models (ANNs and classic GP) have a higher accuracy than the linear model does. For the best nonlinear model (GP), the RMSE equals 0.01 (0.01) and the PAE equals 1.4% (21%). For the linear model, the RMSE equals 0.02 (0.04) and the PAE equals 2.7% (79%), respectively.

In Figure 3, no significant changes in the ranking of the metamodels were observed. In the upper row, we analyzed the effects of low and high transition rates. If $\beta_{1,12} = -3.5$ (lower rate of disease), the RMSE (PAE) of the best model is 0.02 (PAE = 1.4%), while for $\beta_{1,12} = -1$ (higher rate of disease), the RMSE is smaller than 0.01 (PAE = 0.3%).

**Computational Speed of R Packages to Fit a Metamodel**

In Supplementary Figure 5, we show the computational speed of each R package. Most metamodels can be fit a
Figure 2  Accuracy of different statistical models to emulate a simulation model with 10 parameters.\textsuperscript{abc}
(a) RMSE\_range denotes the average of the root mean squared error divided by the range of the simulated model outcome. This was computed over 50 data sets (\(D\)). The model outcome to be emulated is life-years gained due to treatment. \(M\) is the number of life histories simulated during simulation model training runs, \(N\) is the size of the data set used to estimate simulation model parameters and also affects the level of uncertainty in the treatment effect, \(S\) is the number of simulation model runs. (b) Each emulator is denoted by its R package name or R function. \texttt{lm} is the function for the linear regression model, \texttt{GPFit} is the R package for classic GP with squared exponential covariance, \texttt{kernlab} (here named \texttt{kernl}) is the R package for isotropic Gaussian process regression, \texttt{neuralnet} is the R package for artificial neural networks, \texttt{mgcv} is an R package for spline-based generalized additive models, and \texttt{gbm} is for boosting with decision trees. (c) See Supplementary Table S6 for other metamodels and error measures.
Figure 3  Accuracy of different statistical models: effect of parameter value (upper row), semi-Markov scenario, and disease-specific mortality with treatment (bottom row).abc

(a) rmse.range denotes the average of the root mean squared error divided by the range of the simulated model outcome. This was computed over 50 data sets (D). The model outcome to be emulated is life-years gained due to treatment. M is the number of life histories simulated during the simulation model training runs, N is the size of the data set used to estimate simulation model parameters, and S is the number of simulation model runs. (b) Each emulator is denoted by its R package name or R function. lm is the function for the linear regression model, GPFit is the R package for classic GP with squared exponential covariance, kernlab is the R package for isotropic gaussian process regression, neuralnet is an R package for artificial neural networks, mgcv is an R package for spline-based generalized additive models, and gbm is for boosting with decision trees. (c) In the left corner of the upper row, \( \beta_{1,12} = -3.5 \) and \( \beta_{1,23} = -3.5 \), whereas in the right corner \( \beta_{1,12} = -1 \) and \( \beta_{1,23} = -1 \). All other parameters were fixed at the same values (see Supplementary Table S1). If one fixes all other covariates equal to zero, \( \beta_{1,12} = -3.5 \) corresponds to an average duration to state 2 of about 33 y, whereas \( \beta_{1,12} = -1 \) corresponds to an average duration of about 2.7 y. (d) In the semi-Markov scenario, the shape parameter for the Gompertz distribution is uniformly distributed between 0.1 and 0.5. The upper bound for variation is 30%. (e) See Supplementary Table S7 for other metamodels and error measures.
in less than 1 s. The exception is GP with both GP_fit and tgp packages. The maximum fitting times were observed with 10 parameters and 200 observations for training. Then, the tgp package takes about 73 s. The GP_fit package takes 1168 s.

Case Study: PA for Cost-Effectiveness Analyses of 2 Treatments for NSCLC

Figure 4 shows the accuracy of different metamodels for predicting 10,000 simulation IL-STM runs for SBRT costs, whereas in Supplementary Table S8, we show the results for all metamodels and model outcomes. For both SBRT and VATS costs metamodels, 13 parameters were included, whereas for SBRT and VATS QALYs gained, respectively, 18 and 17 parameters were included.

Most metamodels had an RMSE less than 0.01 for SBRT and VATS costs because of a relatively wide range of treatment costs. The best metamodels were GAM (R package mgcv, PAE 2.1%) for SBRT costs and ANN (R package nnet, PAE 1.0%) for VATS costs. For both outcomes, LM had good performance (SBRT costs, PAE = 2.0%, VATS costs, PAE = 2.6%). Some metamodels had difficulties emulating higher observed SBRT costs, showing a downward bias. Most metamodels for VATS and SBRT QALYs had an RMSE of about 0.05. The ranking of the metamodels was similar for both VATS and SBRT QALYs. ANNs (R package nnet) was the best model; however, the difference in accuracy as compared with GAMs, GP, and LM was relatively small (R packages, nnet: PAE = 2.1%, lm: PAE = 2.4%).

Discussion

In this study, we systematically evaluated the prediction accuracy of several statistical models for the purpose of uncertainty quantification. In general, prediction accuracy becomes better with increasing $N$ (lower parameter uncertainty), $M$ (more life histories simulated), and $S$ (more simulation runs) (Figure 2). Our findings indicate that the set of models with best accuracy include GAM, (frequentist) GP with squared exponential covariance, ANNs, and the LM. These are also the most commonly used models in metamodeling. Models for decision trees, Bayesian GP, and single-parameter GP (package kernlab) do not seem to provide good accuracy. The prediction accuracy of the Bayesian networks was similar to that of the linear model.

In the case study, a PA of an IL-STM for NSCLC was reanalyzed. The goal was to verify whether we could replace a majority of the 10,000 simulation model runs by a metamodel at a cost of a small prediction error. For all 4 model outcomes examined, the best metamodel had a reasonable prediction accuracy. In the worst case, the best metamodels had a PAE of 2.1%. In total, about 220 model runs were used to train the metamodels for the different model outcomes, which represents a reduction of more than 97% in the number of simulation model runs.

The 2 most commonly used metamodels are the LM and GP regression. LM provides a good performance in case the relationship between model parameters and outcomes is linear on the domain being considered (i.e., either this relationship is relatively simple or the confidence intervals are relatively small). It is much easier to understand, implement, and interpret than alternative models, especially compared with GP and ANNs. The computation time to fit a GP regression model grows exponentially with the number of IL-STM parameters and number of simulation runs. Modern machine-learning methods can easily handle dozens of model parameters in a fraction of the computation time. However, in this context of a small training sample and parameters, machine-learning methods could be at a disadvantage. Namely, models based on decision trees performed poorly, and Bayesian networks have a similar performance to the linear model. It also seems that the accuracy of GP is in some cases superior to that of ANN (Figure 2), and therefore, we should not completely discard classic GP regression if the number of model parameters is smaller than 15 and/or if we are not using cross-validation. GAMs are an interesting alternative to ANNs and GPs, especially if the number of parameters is large and the researcher is concerned about interpretability of the model parameters.

Another factor to be considered when choosing a metamodel could be the number of hyperparameters used by each statistical method. Metamodels with a high number of hyperparameters (e.g., ANNs) may have a lower bias; however, this comes at a cost of a higher variance of the metamodel predictions. We may be more interested in methods with lower variance, if we use metamodeling for policy analyses, in which the confidence intervals of the metamodel predictions could also be relevant. Recent results in high-dimensional statistics suggest that under certain conditions, it is possible that overparametrized models (i.e., models with a $P$ significantly larger than $S$) could achieve a lower mean square error than a less complex model with $P < S$. Whether this could also be the case for metamodeling still requires further investigation.

We are not aware of any other study systematically evaluating metamodel prediction accuracy; however, 3 studies reported the prediction accuracy of 2 or 3
Figure 4 Observed (simulation model) and predicted (metamodels) total costs of SBRT.abc
(a) SBRT denotes stereotactic body radiation therapy. Costs are deflated to 2018 Euros. Discount rate for costs is 1.5%, based on Dutch guidelines. The average total discounted costs of SBRT during the probabilistic analysis (PA) using the individual-level state transition simulation models equal to 24,998 Euros. (b) Each emulator is denoted by its R package name or R function. lm is the function for the linear regression model, GPFit is the R package for classic GP with squared exponential covariance, nnet and neuralnet are R packages for artificial neural networks, mgcv is an R package for spline-based generalized additive models, mboost is an R package for boosting/spline–based generalized additive models, gbm is for boosting with decision trees, and ranger is an R package for random forests with decision trees. (c) See Supplementary Table S4 for included model parameters in the analyses and Supplementary Table S8 for additional metamodels and error measures.
metamodels, ANNs, GP, and LM. All studies reported a better accuracy of ANN and/or GP as compared with the LM. ANN was also superior to GP in one study. The exception is the metamodel for QALYs from Alam and Briggs, in which LM has roughly similar accuracy to the ANNs and GP.

The models that were found to have highest prediction accuracy are not necessarily the “optimal models.” Our goal was to find a set of models that results in a good accuracy, within a reasonable amount of time and without too much effort from the analyst. Therefore, we deliberately tested only a handful of possible hyperparameter combinations (a maximum of 6 per statistical model) to find the best models. For some models, such as ANNs or GAM boosting (R package: mboost), the number of possible specifications is large, and we cannot exclude the possibility that a model with a better accuracy could be found.

These results are applicable to a certain class of microsimulation models used to model chronic, nontransmissible diseases. The main assumption here is that small changes in input values will cause a small change in the output. The second assumption is that the region of interest for metamodeling is relatively small. If this is large, such as, for calibration, then the rule of thumb of using 10 times the number of parameters as the number of simulation model runs for training does not apply and may result in a poor approximation of the model outcome regardless of the metamodel used (Supplementary Figure S4). The main outcome measure of the simulation study is life-years gained, whereas the main outcome of interest in cost-effectiveness analyses is a ratio (cost per QALY), which could introduce additional complexity in the relationship between model parameters and outcomes compared with the simulation study and case study presented.

The generalizability of these results is also limited by the fact that we considered in the simulation study a single decision model with only 10 parameters, without any interactions or correlations. IL-STMUs used in public health could include between 50 and 100 parameters. However, in practice, it is unlikely that an analyst would consider including all model parameters in the metamodel, as this would require running the decision model thousands of times to generate training data, which could be computationally unfeasible. In a typical decision model, only a handful of model parameters are influential, whereas most parameters will have a limited effect on the outcome. The analyst could simplify the problem by using prior knowledge about the model to select the most influential model parameters or use variable selection techniques.

There are several possible statistical models that an analyst could use as a metamodel. For instance, caret R package has about 240 models available. However, it is sufficient to focus on a handful of models, LM, GAM, NN, and GP. Within this set, we recommend the following steps: 1) fit the linear model; 2) if the linear model gives sufficient accuracy stop, otherwise try ANNs; 3) if ANNs do not give sufficient accuracy and if the number of parameters is smaller than 15 and/or the number of cross-validation folds is not too large, try GP regression. The accuracy of the metamodel could still be poor, independently of the chosen metamodel, if we do not simulate enough life histories or if we apply the “10 times the number of parameters rule” to the size of the training data and parametric uncertainty is large.

To facilitate the use of metamodeling in the health economics and public health community, we share the code used to fit the metamodels at https://github.com/tmdecarvalho. This will provide a reference on how to implement different metamodels in R. Intermediate/advanced users could then adapt the script and “play” with the hyperparameters until they find the best specification for their needs or even add their own models and use these specifications as a benchmark.

Acknowledgments
We thank 4 anonymous reviewers, whose comments greatly improved this article.

ORCID iDs
Tiago M. de Carvalho https://orcid.org/0000-0001-8232-9210
Joost van Rosmalen https://orcid.org/0000-0002-9187-244X
Harold B. Wolff https://orcid.org/0000-0003-3417-6955

Supplemental Material
Supplementary material for this article is available on the Medical Decision Making website at http://journals.sagepub.com/home/mdm.

References
29. Belkin M, Hsu D, Ma S, and Mandal S. Reconciling modern machine-learning practice and the classical bias–variance