A Novel Approach to Building Surrogate Models of Stochastic Simulations

by

Samira Mohammadi

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 6, 2022

Keywords: surrogate models, high fidelity simulations, uncertainty propagation, optimization

Approved by

Selen Cremaschi, Chair, B. Redd & Susan W. Redd Professor of Chemical Engineering
Peter He, Associate Professor of Chemical Engineering
Elizabeth Lipke, Mary and John H. Sanders Professor of Chemical Engineering
Aleksandr Vinel, Associate Professor of Industrial and Systems Engineering

Abstract

High-fidelity simulations represent the processes and phenomena in a detailed manner. However, they are generally computationally expensive to evaluate for optimization and sensitivity analysis applications. Additionally, some high-fidelity simulations are stochastic due to different uncertainty sources. The source of uncertainty in simulations can be divided into two groups of intrinsic and extrinsic uncertainty: in the first case, the uncertainty is due to uncertain parameters or model form, and in the latter one, the simulation has uncertain inputs. The uncertainty of the output can be quantified and analyzed through different methods.

Uncertainty propagation (UP) is one of the popular methods to propagate the uncertainty of the uncertain inputs (extrinsic uncertainty) to the output. Most UP methods require multiple simulation runs, which is not favorable with high-fidelity simulations. Different UP methods were compared to each other in terms of their efficiencies to estimate the first four statistical moments of the output in this study. The metric used to assess the performance is the minimum number of simulation runs required to reach a certain confidence level for the moment estimates. The methods considered include Monte-Carlo simulation, numerical integration, and expansion-based methods. The results reveal that, despite their accuracy, numerical integration methods' performance deteriorates quickly with increases in the number of uncertain inputs. The Monte-Carlo simulation methods converge to the moments' *true* values with the minimum number of model evaluations if model characteristics are not considered or known.

One popular method to handle the intrinsic uncertainty due to uncertain parameters is utilizing surrogate models representing high-fidelity stochastic simulations. The surrogate models approximate the high fidelity simulation with cheaper to evaluate functions. The existing surrogate modeling techniques are mainly designed for deterministic systems, and only a few approaches

are available for stochastic simulations. My study introduces a new method, called PARIN (PARameter as INput), to efficiently construct accurate surrogate models of high-fidelity stochastic simulations. PARIN is compared to three existing approaches in terms of accuracy and efficiency: fixing the uncertain parameters at a preselected value (Fixed parameter), training multiple surrogate models for a selected set of uncertain parameter values (Parameter set), and stochastic kinging. The results reveal that PARIN generally has a lower normalized root mean square error in predicting the mean and standard deviation of the simulation outputs. The output distribution predicted by PARIN has the lowest Wasserstein distance (W_d) from the actual output distribution compared to the other approaches. However, both metrics for PARIN estimates deteriorate for simulations with a significantly large number of input variables in low computational-budget cases.

Acknowledgements

First and foremost, I would like to thank my advisor Dr. Selen Cremaschi for her unlimited and persistent support and guidance. Her intelligence, patience, and knowledge of different areas have always inspired me to become a better and stronger researcher. I have learned a lot along the way from her in both personal and professional level.

I would like to give my special thanks to Dr. Peter He, Dr. Elizabeth Lipke, and Aleksandr Vinel for serving in my committee and providing me guidance on making my research better. I am grateful for having had Dr. Gregory Purdy as my university reader. I would also like to thank Dr. Mario Eden, the chair of chemical engineering department at Auburn University for being an incredible person and providing unlimited help throughout my graduate school years.

This journey was not possible without help of my colleagues in Cremaschi group and Auburn University. I would like to give my special thanks to Dr. David Young for his continuous support throughout all these years both technically and emotionally, he has been a real academic brother to me. I am grateful for having Dr. Amirhossein Hejri, Afsaneh Radmand, Dr. Zahra Karimi, Navid Etebarialamdari, Dr. Mariam Khachatriyan, and Kritika Malhotra as my emotional support circle during past 6 years, not only by being great friends but also becoming my family during the years being far from my own family.

I would like to thank my partner Shahryar Afzal, for believing in me no matter what and helping me to get through tough moments with love and patience . Finally yet importantly, I would like to dedicate my deepest gratitude to my family, my father and mother, Ali Mohammadi and Marziyeh Javadimarand, for their unconditional love since the day I was born till now. My siblings,

Sima and Amir, who never gave up on helping me and pushing me forward and being there for me in any time of need.

Table of Contents

List of Tables

List of Figures

To my middle school science teacher, Ms. Mashhadizadeh, who cultivated the curiosity in me.

To my beloved sister, Sima, who never stopped supporting and believing in me.

To all the women in science, especially immigrants, who paved the way for me and helped me see it through.

Chapter 1 – **Introduction**

With recent advances in computational capacities and capabilities, high-fidelity simulations have become a popular and powerful method for decision-making and optimization in many engineering fields, e.g., chemical engineering (Al et al., 2020; Burnak et al., 2019). High-fidelity simulations can be developed to model a variety of processes, including multi-phase flow modeling in different equipment and applications like pipelines or wellbores (Alizadehdakhel et al., 2010; Livescu et al., 2010), erosion predictions in pipelines with different flow regimes (Zahedi et al., 2017), heat transfer in pipelines or heat exchangers (Gupta et al., 2010), and many other applications. Most of these simulation models consist of Monte Carlo simulations and a number of differential equations (partial or ordinary) that should be solved, generally numerically, using a large number of discretization. Therefore, high-fidelity simulations are generally computationally expensive to evaluate. In many cases, employing high-fidelity simulations for applications like optimization or sensitivity analysis of the systems, where a high number of simulation runs are needed (Liu et al., 2016; Peherstorfer et al., 2017), requires computational resources beyond available. High-fidelity simulations may have uncertain inputs, which is an extrinsic uncertainty, uncertain parameters, and uncertain model form, which are intrinsic (Ankenman et al., 2008). The estimation of the outputs and propagation of uncertainty for high-fidelity simulations with uncertainty is computationally expensive.

One source of uncertainty in high-fidelity simulations is uncertain inputs, i.e., extrinsic uncertainty. There are different methods to propagate input uncertainty to outputs. Although many studies have carried out comparative analyses of different uncertainty propagation methods, none considered the efficiencies of these methods in terms of the required number of simulation runs for accurate propagation of uncertainty. This dissertation fills this gap by comparing six

commonly-used uncertainty propagation methods in terms of their efficiencies to predict the four statistical moments of the simulation outputs and deriving guidelines for selecting the appropriate uncertainty propagation methods based on the simulation model characteristics.

The computational burden of high-fidelity stochastic simulations in sensitivity analysis, optimization studies, or uncertainty propagation can be reduced using surrogate models representing the simulations (Quirante et al., 2015). Surrogate models are reduced-order models of the simulations, which map inputs to the output(s) (Quirante et al., 2015). They lead to computationally cheaper to run models (Jiang et al., 2020). Various modeling techniques have been developed and implemented to build surrogate models. (e.g., (L. E. O. Breiman, 2001; Friedman, 1991; Haleem et al., 2013; Williams and Rasmussen, 2006)). They are developed for deterministic systems and can be used for approximation of the stochastic simulations only with uncertain inputs, i.e., extrinsic uncertainty (Staum, 2009). In such cases, a deterministic surrogate model is trained to represent the simulation output, and then the uncertainty of the surrogate model output(s) due to uncertain inputs is estimated using uncertainty propagation methods (Kim, 2016). However, the existing modeling techniques fail to accurately map the input-output pairs for simulations with intrinsic uncertainty.

Few studies investigated the problem of building surrogate models for high-fidelity simulations with intrinsic uncertainty. Stochastic kriging (Ankenman et al., 2008), using a subset of uncertain parameter realizations (Hüllen et al., 2019), and fixing uncertain parameter values (Hüllen et al., 2019) are the approaches that have addressed this problem. However, each has limitations, making these approaches a weak candidate for accurate estimation of the output for high-fidelity stochastic simulations and their uncertainty. This dissertation addresses this gap by

introducing a framework to accurately and efficiently estimate the output of high-fidelity stochastic simulations and the uncertainty of this estimate.

1.1 Objectives

This dissertation:

(1) Explores and compares the accuracy of different uncertainty propagation methods based on the required number of simulation runs to estimate the first four statistical moments of the output.

(2) Implements and compares different machine learning techniques for building surrogate models of experimental applications.

(3) Develops a new approach for efficiently building accurate surrogate models of high-fidelity stochastic simulations.

1.2 Organization

The dissertation is organized as follows. Chapter 2 presents background information about three main areas of the study. Section 2.1 provides the literature review on uncertainty propagation (UP) methods and gives details of six common UP methods. Section 2.2 goes over machine learning applications and techniques implemented in this research study. The literature review on the construction of surrogate models for stochastic simulations is included in Section 2.3. Chapter 3 describes work on the assessment of six different uncertainty propagation methods in estimating the four statistical moments of outputs from simulations with uncertain inputs and discusses the resulting guidelines from this study. In Chapter 4, the workflow and results for two ML technique applications for classification and regression are discussed. Section 4.1 discusses and shows results on classifying cardiomyocyte content differentiated from human-induced pluripotent stem cells. Another application of ML techniques is included in Section 4.2, where regression models are

constructed for predicting the size and polydispersity prediction of produced nanoparticles. Chapter 5 explains PARIN, the approach developed to estimate outputs of simulations with uncertain parameters. It also contains the computational experiments and results for comparison of PARIN to three existing approaches for building surrogate models of stochastic simulations. Finally, Chapter 6 goes over the conclusions and future directions of this dissertation.

Chapter 2 – **Literature Review**

2.1 Uncertainty Propagation

With advances in computing systems and improved computational power, simulation models have become popular methods for assisting with decision-making in chemical process design and operation. Many uncertainties present in the simulation models, e.g., in the model inputs and/or parameters, model formulations, and numerical calculations, cause their outputs to be uncertain. In recent years, the uncertainty due to numerical calculations has reduced significantly with advanced computational power (Hüllen et al., 2019). Therefore, the primary sources of uncertainty in simulation outputs are uncertain inputs, uncertain parameters, and model form uncertainty.

Model uncertainty is studied and characterized using the uncertainty quantification (UQ) methods. The UQ methods are also used to reduce uncertainties in the systems to generate reliable output values and increase confidence in the models (Miller et al., 2014). Important steps of UQ are 1) identification of uncertainty sources, 2) characterization of the sources, 3) uncertainty propagation (UP), and 4) analyzing the uncertainties (Gel et al., 2013). Uncertainty propagation investigates the contribution of uncertain sources to the final uncertainty of the model. When only extrinsic uncertainty is considered, the UP methods propagate the uncertainty of the inputs (X) to the model outputs ($Y=g(X)$) of the model $g(.)$ (Lee and Chen, 2009). For propagating extrinsic uncertainty to outputs, UP methods first require selecting the appropriate statistical representation for the uncertain input variables. Next, the UP is carried out to make statistical inferences regarding the outputs. Statistical inferences regarding the uncertain outputs are generally carried out through estimating three main statistical concepts: the probability density function of the outputs, statistical moments of the outputs, and the probability of a certain outcome, such as failure, based on output

distribution (Yang et al., 2017). Estimation of the four statistical moments was the focus of this study.

There are many challenges in UQ and UP, such as discontinuous response surfaces, selection of significant uncertain parameters for models with high dimensionality, highly complex physical/simulation models, and computational cost associated with UP. There are many UP methods in the literature addressing parts of these challenges. (Groen et al., 2014; Luo and Yang, 2017; Wang and Sheen, 2015).

Uncertainty propagation methods are divided into two groups, intrusive and non-intrusive methods. In intrusive methods, the model formulation is needed and modified to propagate input uncertainty. The models are treated as black boxes for non-intrusive methods. Lee and Chen (2009) categorized the non-intrusive UP methods into five groups, 1) simulation-based methods, e.g., Monte Carlo (MC) simulations, 2) local expansion based methods, 3) most probable point-based methods, 4) functional expansion–based methods, e.g., polynomial chaos expansion (PCE), and 5) numerical integration-based methods. It has been established that the moment estimates obtained using local expansion-based UP methods are significantly different from the true values for models with high nonlinearities (Jia et al., 2019; Lee and Chen, 2009). Most probable point-based UP methods are typically used for reliability applications and do not provide accurate estimates of higher statistical moments (Arakere et al., 2010; Padulo et al., 2007). In addition to these five categories, response-surface-based methods have been used in recent years, where the models of interest are represented through surrogate models (Murcia et al., 2018; Sofi et al., 2020; Tripathy et al., 2016). The response-surface-based methods encompass the fourth category, which is functional expansion-based methods.

Most UP methods require the evaluation of complex simulation models and many model runs (Liu and Gupta, 2007). Carrying out UP for complex or high fidelity models that are computationally expensive to evaluate could be prohibitive for achieving accurate results with some UP methods (Rajabi, 2019). Hence, selecting the appropriate UP method is crucial for efficient and accurate UP. The following section gives a brief explanation of several different UP methods which were used in this study.

2.1.1 Uncertainty Propagation Methods

2.1.1.1 Monte Carlo Simulation-based Methods

For Monte Carlo simulation (MCS) based methods, the method of moments (Hansen, 1982) is used to estimate the statistical moments. The i^{th} moment (μ_i) is calculated by Eq. 1,

$$\mu_i = E\left[g(X)^i\right] \approx \frac{1}{m}\sum_{j=1}^{m} g(X_j)^i \tag{1}$$

where $g(X_j)$ is the model value at j^{th} sample point X_j from input distribution(s), and m is the number of sample points. Three methods, LHS (McKay et al., 1979), Sobol sequences (Sobol', 1967), and Halton series (Halton, 1960), are implemented as sampling techniques for determining uncertain input space sample locations and calculating the corresponding model outputs for propagating input uncertainty. These sampling methods are space-filling techniques (Crombecq et al., 2011), which evenly spread the sample points throughout the input space. Moreover, Sobol and Halton series are sequential sampling methods (Hou et al., 2019), allowing previous sample points and model evaluations to be reutilized if additional sample points are collected.

2.1.1.1 Latin Hypercube Sampling (LHS)

In the LHS method (McKay et al., 1979), the range for each uncertain variable is divided into m bins with equal probability, where m is the number of required sample points. Then, a sample

is randomly selected from each bin for each uncertain variable. Next, the samples of different uncertain variables are randomly matched and result in $m \times n$ sample matrix, where n is the number of uncertain inputs. The initial binning enables LHS to cover the space of each uncertain parameter better than MCS with random sampling. However, the random matching of the samples from different dimensions could cause clusters in the design space and may lead to a poor space-filling attribute. Latin Hypercube sampling is not a sequential sampling method because the binning of the uncertain parameters is a function of the number of sample points, and it changes as the value of m is modified (Fahmi and Cremaschi, 2016).

2.1.1.2 Sobol Sequences

Sobol sequences are low-discrepancy pseudo-random series (Sobol', 1967). These sequences are constructed to generate samples as uniformly as possible over the sampling space (Saltelli et al., 2010). Every new sample point is generated based on the location of the existing points, which helps with avoiding clusters and gaps (Burhenne et al., 2011). According to Joe and Kuo (Joe and Kuo, 2008), for the generation of the j^{th} component of Sobol samples, a primitive polynomial of order s_j, must be selected (Eq. 2).

$$x^{s_j} + a_{1,j}x^{s_j-1} + a_{2,j}x^{s_j-2} + \cdots + a_{s_j-1,j}x + 1 \quad (2)$$

The coefficients $a_{1,j}, a_{2,j}, \ldots, a_{s_j-1,j}$ in Eq. 2 are binary values. The j^{th}component of the β^{th} point in Sobol series, $x_{\beta,j}$, is calculated by Eq. 3,

$$x_{i,j} = \beta_1 v_{1,j} \oplus \beta_2 v_{2,j} \oplus \cdots \quad (3)$$

where β_k is the k^{th} digit from the right when β is written in binary $\beta = (\ldots \beta_3\beta_2\beta_1)2$ and $\oplus$ is the bit-by-bit exclusive-or operator. The parameters $v_{k,j}$ are called the direction numbers (Eq. 4), where $q_{k,j}$ are positive integers calculated in Eq. 5.

$$v_{k,j} = \frac{q_{k,j}}{2k} \tag{4}$$

$$q_{k,j} = 2a_{1,j}q_{k-1,j} \oplus \ 2^2 a_{2,j}q_{k-2,j} \oplus \ldots \oplus \ q_{k-s_j,j} \tag{5}$$

2.1.1.3 Halton Series

Halton sequences (Halton, 1960) are low-discrepancy series used for sampling. Given n number of uncertain inputs, the i^{th} sample point obtained using Halton series is given by Eq. 6,

$$\left(\Phi_{p(1)}(i-1), \Phi_{p(2)}(i-1), \ldots, \Phi_{p(n)}(i-1)\right) \tag{6}$$

where $p(\mathrm{n})$ is a selected arbitrary prime number, which is subject to $p(1) < p(2) < \cdots < p(n-1)$. The variable $\Phi_p(i)$ is defined in Eq. 7 (Wong et al., 2005).

$$\Phi_p(i) = \frac{t_0}{p^1} + \frac{t_1}{p^2} + \frac{t_2}{p^3} + \cdots + \frac{t_r}{p^{r+1}} \tag{7}$$

In Eq. 7, t_r is an integer in $[0, p-1]$ that follows Eq. 8, which shows the expansion of integer i with the maximum order of r, where r is any positive integer value, via prime base p (Wong et al., 2005).

$$i = a_0 + a_1 p + a_2 p^2 + \cdots + a_r p^r \tag{8}$$

2.1.1.2 Numerical Integration Methods

The output moments of a model with uncertain inputs are defined in Eq. 9, which gives the integral for the i^{th} moment, μ_i (Grimmett and Stirzaker, 2001). This integral can be computed using an appropriate numerical integration method.

$$\mu_i = \int_{-\infty}^{\infty} g(X)^i f(X) dX \tag{9}$$

In Eq. 9, $f(X)$ is the joint density function of the uncertain input variables, X, for the model $g(X)$. Here, we utilize FFNI, UDR, and SG, which are popular numerical integration methods.

2.1.2.1 Full Factorial Numerical Integration

Full factorial numerical integration (FFNI) employs a weighted sum of the model output values at specified input values (Duffy et al., 1998). For estimating the statistical moments of model output with uncertain inputs, the model is evaluated at $m = \delta^n$ specific input values, which can also be referred to as sample locations, where n is the number of uncertain inputs and δ is the number of nodes from each of these inputs. The weights (w_{k_j}), and sample locations (x_{k_j}) for the kth point of jth dimension are determined by implementing Gauss-Hermite, Gauss-Legendre, or Gauss-Laguerre quadrature based on the distribution of each uncertain input variable (Abramowitz et al., 1988). Then, the desired moments are calculated by Eq. 10.

$$\mu_i = E\left(g(X)^i\right) = \sum_{k_1=1}^{\delta} w_{k_1} \dots \sum_{k_n=1}^{\delta} w_{k_n} \times [g(x_{k_1}, \dots, x_{k_n})]^i \tag{10}$$

2.1.2.2 Univariate Dimension Reduction (UDR)

In the UDR method, the multivariate function, $g(X)$, is approximated to $\hat{g}(X)$ using the summation of several univariate functions (Rahman and Xu, 2004). Each of these univariate functions, $g_j(X_j)$, are similar to the original one, but each one is only a function of one variable with the remaining variables set to their mean values ($M_{\boldsymbol{X}_{j'}}$) (Eq. 11). Then, the original model output is approximated by the additive decomposition of the model (Eq. 12),

$$g_j(X_j) = g\left(X_j, X_{j'} = M_{\boldsymbol{X}_{j'}}\right) \quad \forall j, j' \in \{1,2, \dots, n\}, \; j \neq j' \tag{11}$$

$$g(X) \approx \hat{g}(X) = \sum_{j=1}^{n} g_j(X_j) - (n-1)g(M_X) \tag{12}$$

where $g(M_X)$ is the model output value with all uncertain input variables set at their mean values. The moments of the output function ($g(X)$) are calculated using the estimated model outputs, $\hat{g}(X)$, and the quadrature formula (Eq. 13), similar to the FFNI method. Employing univariate quadrature formula with m nodes, the number of model evaluations is equal to $m = \delta n + 1$, where n univariate models are calculated at δ different nodes, and one extra model evaluation is performed with all variables set to their mean values.

$$\mu_i = E\left(g(X)^i\right) \approx E\left(\hat{g}(X)^i\right) = E\left(\left\{\sum_{j=1}^{n} g_j(x_j) - (n-1)\, g(M_x)\right\}^i\right) \tag{13}$$

2.1.2.3 Sparse Grid Numerical Integration

Sparse Grid (SG) (Smolyak, 1963) is a numerical integration method that uses quadrature formulas, similar to FFNI, for estimating integrals (such as Eq. 9 and Eq. 10). The sample points and the weights are determined using Eq. 14 and Eq. 15, respectively (Xiong et al., 2010).

$$\vec{U}_n^k = \bigcup_{k+1 \le |i| \le k+n} U_1^{i_1} \otimes U_1^{i_2} \ldots \otimes U_1^{i_n} \tag{14}$$

$$w_l = (-1)^{k+n-|i|} \binom{n-1}{k+n-|i|} \left(w_{j_1}^{i_1} \ldots w_{j_n}^{i_n}\right) \tag{15}$$

where $\vec{U}_n^k$, is a $P_s \times n$ array of all sample points with an accuracy of k, n denotes the number of dimensions, and P_s is the number of resulting possible sample points given $k+1 \le |i| \le k+d$. The operation $\otimes$ corresponds to the tensor product of arrays. The variables $U_1^{i_1}, \ldots, U_1^{i_n}$ are one-dimensional quadrature points for each dimension, and i_α is the number of nodes in dimension α. Variable $|i|$ is the summation of the multi-indices ($|i| = i_1 + \cdots + i_n$). In Eq. 15, w_l is the weight

for the l^{th} sample point $\overrightarrow{X_l} = \left[X_{j_1}^{i_1}, \dots, X_{j_n}^{i_n}\right] \in \vec{U}_n^k$, where $j_\alpha \in \{1, \dots, i_\alpha\}$. The parameter $w_{j_{i_1}}^{i_1}$ is the weight for the sample sets of one-dimensional quadrature. The i^{th} moment (μ_i) is calculated using Eq. 16.

$$\mu_i = E\left(g(X)^i\right) = \sum_{l=1}^{P_s} w_l \; g(\overrightarrow{X_l})^i \tag{16}$$

2.1.1.3 Functional Expansion–based Methods

In functional expansion-based methods, the model output is approximated by a polynomial function. This approximate simpler model is used in conjunction with UP methods to estimate the statistical moments. In this study, PCE is selected as a representative functional expansion–based method.

2.1.3.1 Polynomial Chaos Expansion (PCE)

Polynomial chaos expansion (Wiener, 1938) approximates the model output using orthogonal polynomials. It projects the output variable as a function of random variables with a specific distribution based on orthogonal stochastic polynomials (Anthony, 2013; Crestaux et al., 2009). The statistical moments of the output are calculated using the projected polynomial expansion. The general form of the PCE of a random variable, $u(\theta)$, can be written as Eq. 17 (Crestaux et al., 2009).

$$u(\theta) = c_0\Gamma_0 + \sum_{i_1=1}^{\infty} c_{i_1}\,\Gamma_1\left(\xi_{i_1}(\theta)\right) + \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{i_1} c_{i_1 i_2}\Gamma_2\left(\xi_{i_1}(\theta), \xi_{i_2}(\theta)\right) + \cdots \tag{17}$$

In Eq. 17, Γ_p are the orthogonal polynomials of order p. Different orthogonal polynomials are used depending on the distribution of the random variable θ. For example, Hermite and Legendre are the polynomial basis for normal and uniform distributions, respectively. $\xi_i(\theta)$ is the standard

variable, e.g., standard normal variable if the distribution for input(s) is normal, and c_i's are deterministic coefficients. Eq. 17 can be approximated using Eq. 18,

$$u(\theta) = u(\xi_1, \xi_2, \dots, \xi_n) \approx \sum_{j=0}^{\infty} b_j \psi_j(\xi(\theta)) \tag{18}$$

$$y = g(\xi) \approx \sum_{j=0}^{\frac{(p+n)!}{p!n!}-1} b_j \psi_j(\xi) \tag{19}$$

$$b_i = \frac{E[y\psi_j(\xi)]}{E[\psi_j^2(\xi)]} \tag{20}$$

where ψ_j is the j^{th} component of the orthogonal polynomials. In Eq. 18, b_j correspond to $c_{i_1 i_2 \dots i_p}$'s and are calculated based on Eq. 20. The output (y) for the models with multiple input variables, $g(\xi)$, can be approximated using n-dimensional PCE with an order of p (Eq. 19). For inputs not distributed normally, either specific polynomials are used, or the transformation of the variables to the standard normal variables is carried out.

2.1.2 Uncertainty Propagation Comparison Studies

Multiple studies compared the performance of different UP techniques in terms of accuracy and efficiency to guide selecting an appropriate UP method. Several of these studies compare simulation-based methods to other categories of UP methods. For instance, Klavetter et al. (2012) compared perturbation, Taylor series expansion, and Monte Carlo methods in propagating the uncertainty in slug length and liquid entrainment in gas core to the outputs of a multiphase flow model. The results stated that Taylor series expansion overestimated variance for most outputs, and the other two methods yielded comparable estimates. However, the perturbation method may

not provide reasonable uncertainty estimates for models that are not monotonically increasing or decreasing, and it does not provide confidence levels (Klavetter et al., 2012).

Several studies investigated simulation-based methods versus functional expansion-based methods. Safta et al. (2017) and Hunt et al. (2015) compared the accuracy and efficiency of MC simulations, PCE, and Quasi-Monte Carlo (QMC) simulation methods. Both studies concluded that the PCE required fewer model evaluations to converge to the true value of the output mean for the test functions. Aleti et al. (2018) studied the efficiency of MC simulation and PCE methods based on the number of sample points used to estimate the output distribution accurately. The results revealed that the PCE was 90% more efficient than MC methods in terms of the number of numerical calculations.

Jia et al. (2019) evaluated the performance of MC simulation and the numerical integration approaches, including sparse grid numerical integration (SG), univariate dimension reduction (UDR), and extended sparse grid methods. They concluded that the SG methods were the most efficient in estimating the first four moments of the output requiring the fewest model evaluations. Allen and Camberos (2009) compared simulation-based methods to response surface approaches to estimate the output probability density function and calculate the probability of failure, defined as the probability of an event that the output value exceeds a specific critical level, using the probability density function. They employed two models as case studies, one with high nonlinearity and one with high dimension. The results were evaluated based on the number of required model evaluations to predict the desired uncertainty metrics. They concluded that response-surface methods, especially polynomial chaos expansion, accurately estimated the probability of failure with the lowest number of samples compared to other methods. One other

conclusion was that accurate output distribution estimates required many samples from the uncertain input space and many model evaluations.

Some studies only considered different simulation-based methods and compared their performances. Burhenne et al. (2011) and Hou et al. (2019) studied MC and QMC methods by employing different sampling techniques. The performance was assessed based on accuracy and efficiency in estimating the output mean for a set of test functions in both studies and standard deviation in Hou et al. (2019). The results suggested that QMC methods are efficient and outperform MC methods in most cases.

Other studies investigated the difference in the performance of other UP method categories. Padulo et al. (2007) employed local expansion and most probable point-based approaches in their study. First- and third-order Taylor series expansion and Sigma point methods were used to estimate output uncertainty for four test functions. Sigma point methods provided better estimates of the output mean and standard deviation for input distributions with high variance. Sigma point methods do not require derivatives, which gives them a computational advantage over Taylor series expansion for functions with expensive derivative calculations. Rajabi (2019) and Tardioli et al. (2016) investigated different response surface-based methods. Rajabi (2019) compared PCE to Gaussian Process Emulation (GPE). The study suggested that although GPE had lower normalized Root Mean Square Error (nRMSE) in estimating the response surface, PCE estimated output mean, standard deviation, and probability density function tails with higher accuracy. In addition, PCE tended to have lower statistical dispersion with noisier input probability distributions. Tardioli et al. (2016) compared PCE, Tchebycheff expansions with sparse grids, kriging (Gaussian process modeling), and high dimensional model representation (HDMR) methods. The performance was evaluated based on the methods' ability to represent the response

surface of the test models at different sample sizes using RMSE as the metric. Tchebycheff expansion was concluded to be efficient due to its use of sparse grids and required a lower number of sample points to get to the desired accuracy. The performance of PCE was observed to be inconsistent, and HDMR provided very close results to the Tchebycheff expansion method requiring a lower number of samples to converge to the desired accuracy for all test models. Compared to the other methods, kriging required a high number of model evaluations and had a higher RMSE for all the case studies.

Two papers compared more than two main categories of UP methods. Lee and Chen (2009) and Fahmi and Cremaschi (2016) included MC, Full Factorial Numerical Integration (FFNI), UDR, and PCE methods in a comparative analysis. Fahmi and Cremaschi (2016) also studied different sampling schemas of random, Halton sequences, and Latin Hypercube sampling (LHS) for both MC and PCE. The number of function evaluations used for the analysis was fixed in both studies. The methods were compared in terms of their ability to estimate the four statistical moments of the model outputs. Lee and Chen (2009) concluded that the performance of the methods depended on the model characteristics, such as nonlinearity and uncertain variable interactions. The results from Fahmi and Cremaschi (2016) revealed that simulation-based methods were more sensitive to existing nonlinearities in the test functions than other methods.

The UP method comparisons carried out in the literature demonstrate the importance of the UP method selection for efficiently propagating the extrinsic uncertainty for obtaining accurate estimates of the output uncertainty. They also allude to the correlation between the UP method's accuracy and efficiency and the model characteristics the method is applied to. None of the existing literature compares UP methods based on their efficiency.

2.2 Machine Learning Applications and Techniques

Different feature selection and machine learning techniques were used and compared in two different applications to build data-driven models based on the data gathered from experimental systems. The applications included the classification of cardiac differentiation outcomes for hydrogel-encapsulated human-induced pluripotent stem cells (hiPSCs) and the prediction of poly lactic-co-glycolic acid (PLGA) nanoparticle sizes synthesized using the emulsion solvent evaporation method. For each application, specific feature selection and modeling techniques were used to build either classifiers or regression models to predict the output. This section introduces these techniques.

2.2.1 Feature Selection Methods

Reducing the initial feature sets to smaller ones to eliminate the features with redundant information and keeping the most relative ones is called feature selection (Blum and Langley, 1997). It has been shown that the accuracy of classification models depends on the input features set (Chen et al., 2020). Furthermore, the limited number of data makes feature selection essential to training an accurate model (Chen and Wasikowski, 2008). The generalization of the performance of the trained models is poorer when features are irrelevant or redundant (Remeseiro and Bolon-Canedo, 2019). The feature selection methods considered included filter (Hall and Smith, 1999), embedded (Jović et al., 2015), wrapper methods (Kohavi and John, 1997), and principal component analysis (PCA) (Hotelling, 1933).

2.2.1.1 Filter Methods

Filter methods are generally employed as a preprocessing step to remove the highly correlated input features from the initial feature set. The Pearson correlation coefficient (Soper et al., 1917) can be used to identify strongly correlated features and remove these features, thereby

reducing the multicollinearity of features in the feature set. The Pearson correlation coefficient measures the strength of the linear relationship between two variables. The coefficient value ranges from -1 to 1, with a value of 1 indicating a perfect positive linear correlation, a value of 0 indicating no linear correlation, and a value of -1 indicating a perfect negative linear correlation (Soper et al., 1917).

2.2.1.2 Embedded Methods

Some machine learning techniques have built-in feature selection methods, such as automatic relevance determination in artificial neural networks (MacKay, 1994) and Gaussian process models (Williams and Rasmussen, 2006). These built-in feature selection methods are referred to as embedded methods (L. Breiman, 2001; Williams and Rasmussen, 1996). Simplifying a model by selecting only the most relevant input features makes the model more interpretable and can provide time and resource savings by reducing the amount of data that must be collected. Relevant input features are highly predictive of the output variable for a model (Paananen et al., 2019). Among all the classification techniques employed in this study, Random Forests (RF) and Gaussian Process (GP) based classification techniques have built-in feature selection methods, which calculate the relative importance of the features for predicting the classes.

2.2.1.3 Wrapper Methods

In wrapper methods, different candidate subsets of the features are used to build the models, and the feature set yielding the best-performing model is selected as the final input feature set (Das and Das, 2001; Kohavi and John, 1997). Different strategies exist to generate the candidate subsets. Among them, Forward Selection (FS) (Kittler, 1978), Backward Elimination (BE) (Kittler, 1978), and Bidirectional(BD) Search (Pudil et al., 1994) are some of the popular ones and are considered in this dissertation (Wah et al., 2018). In FS, the feature set is initially empty, and

features are progressively added to the set based on their improvements in the model performance (Almany et al., 2003; Wah et al., 2018). In BE, the initial feature set contains all the features, and the features that least impact the model performance are progressively eliminated from the set (Almany et al., 2003; Wah et al., 2018). In both FS and BE methods, the algorithms used to add or eliminate the features are greedy and may yield sub-optimal feature sets at termination. In the Bidirectional Search, the algorithm is flexible and allows dynamic addition and elimination of the features until a specific stopping criterion is met (Wah et al., 2018).

2.2.1.4 Principal Component Analysis

Principal component analysis (PCA) is a dimensionality reduction technique used to reduce the number of features in large datasets by orthogonally transforming a set of possibly correlated features into a set of linearly uncorrelated principal components (PCs). The PCs are linear combinations of the original features, transformed so that each PC points in the direction of the greatest variance in the data (Hotelling, 1933). The PCs are assigned in ordinal format, such that the first PC contains the highest percentage of the original data variance, and the last PC contains the least. The number of features, or dimensions, is reduced by only using the first few PCs while retaining as much of the variance in the original dataset as possible.

2.2.2 Surrogate Modeling Techniques

Six common surrogate modeling techniques were considered for this work. These techniques include Multivariate Adaptive Regression Splines (MARS), single hidden layer feed-forward Artificial Neural Networks (ANN), Extreme Learning Machines (ELM), Random Forests (RF), Gaussian Process (GP), and Support Vector Machine (SVM). Depending on the project, some of these methods were used for classification or regression.

2.2.2.1 Random Forest

Random forest (RF) (L. Breiman, 2001) models are collections of decision trees making predictions of the desired output. In classification models, the output is predicted by the majority vote of the decision trees in the forest. For the regression case, the output is the average of the predictions from all the trees in the forest. Each tree is constructed independently and depends on a random vector sampled from the input data, with all trees in the forest having the same distribution (L. Breiman, 2001). The RF models have an embedded feature selection algorithm that ranks features in order of importance by calculating how much each feature decreases the impurity (number of incorrect classifications or mean error for regression) at each decision node in the decision trees.

2.2.2.2 Gaussian Process

Gaussian process models are generalizations of the Gaussian probability distribution governed by prior covariance. The prior covariance is specified by a kernel function, which is a measure of similarity between data points that the model has not seen before and the data used to construct, or train, the model (Rasmussen Christopher K. I. Williams., 2005). The GP models in this work use a radial basis function as the kernel. Radial basis functions, $\varphi(r)$, take the form

$$\varphi(r) = e^{-\left(\frac{r^2}{2l^2}\right)} \tag{21}$$

where

$$r = \|x - x'\| \tag{22}$$

In Eqs. (21) and (22), r is the Euclidean distance between data points x and x', and l is a length scale vector determined during estimation of the GP model parameters. Feature selection for GP

models is performed using the values of the length scale vector at each input dimension, where input dimensions with smaller length scale values have larger relevance in the model.

2.2.2.3 Support Vector Machines

Support vector machines (SVM) are machine learning models that transform input data into n-dimensional space. This transformation to a higher dimensional space is performed using a kernel function (Drucker et al., 2002). The kernel function linear separates the data by constructing a set of hyperplanes where the distance from the hyperplanes to the nearest data point on each side of the plane is maximized. On the other hand, for the SVMs for regression (SVRs), the n-dimensional hyperplanes are built in a way that the total distance between the points out of the tolerance margin of the hyperplanes and them is minimized. This study has implemented a radial basis kernel function for classification and regression.

2.2.2.4 Artificial Neural Networks and Extreme Learning Machines

Artificial neural networks (ANNs) (Haykin, 2009) are a modeling technique inspired by how the human brain functions. An ANN consists of input, hidden, and output layers, each containing a number of neurons. A bias is associated with each neuron, and neurons are connected via the layer weights in the network. A nonlinear activation function determines whether a neuron fires or not. In the training of ANN models, the weights and biases are tuned to minimize the error of the predictions. Two types of ANNs are considered in this study. The first one is a feedforward network with one input, one hidden, and one output layer with a hyperbolic tangent as the activation function. The second type is called Extreme Learning Machines (ELMs) (Huang et al., 2006). In ELMs, the weights for all the layers are assigned randomly except for the last hidden layer whose layers are estimated using the training data.

2.2.2.5 Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines (MARS) (Friedman, 1991) is a nonparametric machine learning method based on the linear summation of the basis functions. Basis functions in MARS models are either constants, a hinge function, or the product of two or more hinge functions. In the first step of training MARS models, the basis functions are initialized at the mean value of the outputs, and the model then over fits the training data by adding more basis functions. Next, by implementing a backward pruning pass that uses a generalized cross-validation criterion, the terms with minimal effects on the predictions are eliminated from the model.

2.3 Surrogate Modeling Methods and Techniques for Stochastic Simulations

The three main existing methods to construct surrogate models for stochastic simulations are 1) fixing the value of the uncertain parameter(s) (Fixed) (Hüllen et al., 2019), 2) using several realizations of the uncertain parameter values (PSet) (Hüllen et al., 2019), and 3) Stochastic Kriging (SK) (Ankenman et al., 2008). A brief explanation of each of these methods is included in the next sections. The first method uses a fixed value, usually, the mean value, to construct a deterministic surrogate model. In this method, the uncertainty information is lost, and the output uncertainty is not captured. A set of values is sampled from the uncertain parameter(s) space in the second method, and a surrogate model is trained for each sample value. Several models need to be built and used for predictions using this method, which raises the possibility of added uncertainty based on the data availability. The third method, SK, is based on regular kriging, where the uncertainty is incorporated into the covariance function used in the model. In SK, the surrogate model type is limited to kriging, and there is no option for employing different modeling techniques. However, it has been shown that the best ML technique for building a surrogate model is a function of the data characteristics, which is dependent on the underlying phenomena the

simulation represents (Williams and Cremaschi, 2021). Hence, each of these existing approaches come with limitations of their own for building accurate surrogate models of stochastic simulations efficiently.

2.3.1 Existing Surrogate Modeling Approaches for Stochastic Simulations

2.3.1.1 Fixed Parameter Value (Fixed)

In the Fixed approach, the uncertain parameters are fixed at a nominal value, which is normally the mean value of the parameters. The stochastic simulation ($g(X;K)$) is converted to a deterministic one ($\acute{g}(X)$) by fixing the value of the uncertain parameters ($K = K_{mean}$) (Hüllen et al., 2019). In this method, the uncertainty of the parameters is ignored and not considered in the modeling and prediction of the output (Y). The surrogate model ($h(X)$) of the deterministic simulation, $\acute{g}(X)$, can be built with any ML technique in the absence of uncertainty (Eq. 23). The output ($\hat{Y}$), without uncertainty information, is predicted using the trained surrogate model.

$$Y = g(X;K) \cong \acute{g}(X) \approx \hat{Y} = h(X) \tag{23}$$

2.3.1.2 Subset of Uncertain Parameter Realizations (PSet)

In the PSet approach, multiple observations of uncertain parameters are generated and used to build surrogate models of the stochastic simulation ($g(X;K)$) (Hüllen et al., 2019). The uncertain parameter space is sampled. The uncertain parameter values are fixed to the sampled ones, which results in a set of deterministic simulations, each one corresponding to one set of uncertain parameter realizations. The result is m deterministic simulations ($\acute{g_1}(X), \dots, \acute{g_m}(X)$) given m different realizations of uncertain parameter $K \in \{K_1, \dots, K_m\}$. Using ML techniques, m number of surrogate models ($h_i(X)$) are trained to represent each of the deterministic simulations,

$\acute{g}_\iota(X)$. The mean output value and its standard deviation are estimated as the average value of the trained surrogate models' outputs, Eq. 24, and their standard deviation, Eq. 25, respectively.

$$Y = g(X;K) \approx \bar{\hat{Y}} = E[h(X)] \tag{24}$$

$$\sigma \approx \hat{\sigma} = \sqrt{E\left[\left(h(X) - \bar{\hat{Y}}\right)^2\right]} \tag{25}$$

2.3.1.3 Stochastic Kriging (SK)

Stochastic kriging (SK) was proposed based on the classic kriging method to account for the intrinsic uncertainty of the system itself (Ankenman et al., 2008). It is a tool for evaluating the expected value of the response surface in each design point (Wang and Chen, 2016). Stochastic kriging is preferred over regular kriging because it not only considers the uncertainty occurring due to the input sampling but also the uncertainty corresponding to the stochasticity of the system and/or simulation through the replications information at each training point. The value of the output (Y) on replication j is evaluated using Eq. 26,

$$\boldsymbol{Y_j} = \boldsymbol{f(x)^T\beta} + \boldsymbol{M(X)} + \boldsymbol{\varepsilon_j(x)} \tag{26}$$

where $f(.)$ is the basis function, which is assumed to be one for many applications, β are the unknown parameters, M is a Gaussian random field with a mean of zero, and ε_j is the mean-zero random field realized in the j^{th} replication due to system error.

The unbiased prediction of the output value at any point x is calculated via Eq. 27.

$$\widehat{\boldsymbol{Y}} = \boldsymbol{\beta_0} + \sum_{\boldsymbol{M}} (\boldsymbol{x},.)^T[\sum_{\boldsymbol{M}} + \sum_{\boldsymbol{\varepsilon}}]^{-1} (\bar{\boldsymbol{Y}} - \boldsymbol{\beta_0 1_k}) \tag{27}$$

In Eq. 27, $\sum_M$ is the covariance matrix across all sample points, $\sum_M (x,.)$ is the covariance vector consisting of the covariance of the point x and other points, $\sum_\varepsilon$ is the diagonal matrix of the

covariance of the intrinsic uncertainty, and β_0 is the unknown parameter estimated by the maximum likelihood.

Chapter 3 – **Assessment of Uncertainty Propagation Methods**

In this chapter, seven non-intrusive UP methods are compared based on their ability to estimate the first four statistical moments of the outputs of models ($Y = g(X)$) with uncertain inputs (Figure 1). The estimated mean, standard deviation, skewness, and kurtosis are the extracted information of the output uncertainty. The methods considered are Monte Carlo simulation using 1) Sobol sequences (Sobol', 1967), 2) Halton series (Halton, 1960), and 3) LHS (McKay et al., 1979), FFNI (Duffy et al., 1998), UDR (Rahman and Xu, 2004), SG (Smolyak, 1963), and PCE (Ghanem and Spanos, 1991). An extensive set of test functions were employed to study the effects of 1) nonlinearities, 2) the number of uncertain inputs, and 3) different input uncertainty distributions for establishing guidelines for selecting efficient UP methods. The efficiency of the methods was evaluated using the minimum number of model evaluations required by each method to converge to a preset gap around the true value of the first four statistical moments. Finally, the guidelines were utilized to determine the most appropriate UP method for two case studies. The following sections include the computational experiment results and discussions of this study.

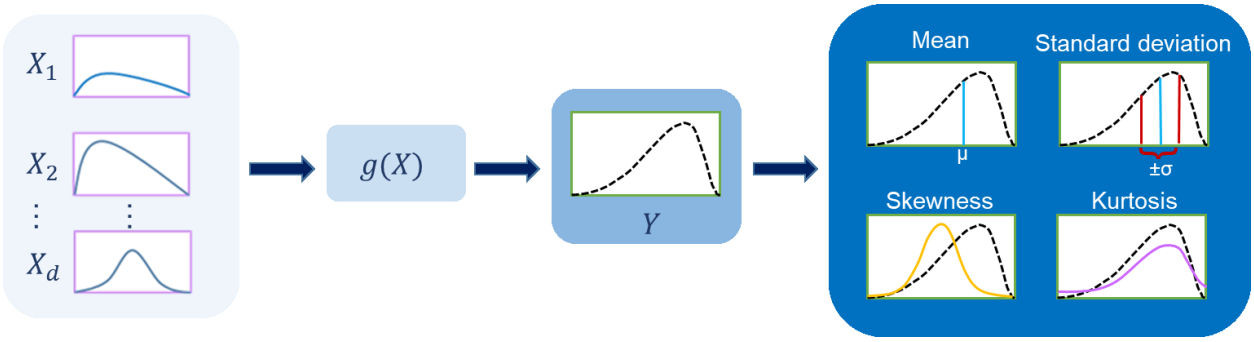


Figure 1. Propagation of uncertain inputs to the simulation output via four statistical moments.

3.1 Computational Experiments

For computational experiments, all UP methods are implemented for propagating input uncertainty to the outputs of a set of test functions with known analytical forms. Numerous test functions and input distributions are considered in the experiments for studying the impacts of functional forms, the number of uncertain inputs, and distributions on the performance of the UP methods. The test function names, their formulas, and the input distributions are summarized in Appendix 1. The uncertainty propagation is carried out by calculating the first four statistical moments of the function outputs. The computational experiments start with three function evaluations and terminate at 1×10^6 function calls. At each increment, four moments are estimated using all applicable UP methods.

In the PCE method, the polynomials are truncated at the order of p. In this study, four different values of p, $p = \{2,3,4,5\}$, are considered to observe the impact of the polynomial truncation order. Also, we use three methods, Sobol sequence (PCE-S), Halton series (PCE-H), and FFNI (PCE-F), for estimating the numerator of Eq. 20.

The quality is defined as the minimum number of function evaluations required for a statistical moment to reach and remain within the desired error gap. To insure that the estimated value stays within the 5% error gap, the search for minimum number of function calls started from the highest value then decreased gradually until the point where the estimation violated the 5% error gap of the true moment value. The error gap is a band with a width equal to a pre-determined error percentage of the '*true*' moment value. Four different error gaps of 2%, 5%, 10%, and 20% were considered. The results for 10% and 20% error gaps were not different from the 5% error gap trends. The variability was high for the 2% error gap, and in many cases, most of the UP methods could not converge to the specified error gap of the test functions. Hence, the 5% error gap results

are presented and discussed in this dissertation. The '*true*' values of the moments are obtained using Monte Carlo simulation with 5×10^6 function evaluations. Experiments are implemented in Python 3.6. The package Chaospy (Feinberg and Langtangen, 2015) is utilized for generating the Sobol series samples and incorporating PCE, respectively.

Four different cases are considered for assessing the performance of the UP methods. In each case, the effects of a specific factor are studied. The functions implemented in each case are included in Table A1.3 in Appendix 1.

3.1.1 Impact of Nonlinearity

The performance of UP methods for various nonlinear functions is studied using two groups of functions. The first group contains twenty different nonlinear functions with one uniformly distributed input. The source of nonlinearity stems from exponential and trigonometric functions and the absolute value operator. The second group contains one-dimensional power functions, in which the input is uniformly distributed. The UP methods are evaluated for power functions with exponent values ranging from one to five to investigate the accuracy and efficiency of their estimates for increasing nonlinearity in a model.

3.1.2 Impact of the Number of Uncertain Inputs

The effects of the number of uncertain inputs on the UP methods are studied by changing the dimensions of G (Surjanovic and Bingham, 2013) and Ackley (Surjanovic and Bingham, 2013) functions. It is assumed that all inputs are uncertain and uniformly distributed. The number of inputs is increased from one to five for the G function and to eleven for the Ackley function.

3.1.3 Impact of the Uncertain Input Distribution

The impact of input distributions is investigated via two test sets. In the first set, UP methods are implemented and evaluated for different one-dimensional functions with both uniform and lognormal distributions. The second set includes the Ackley function with varying input dimensions (from one to eleven) and different distributions, uniform, normal, and lognormal.

3.1.4 General Performance

As the final case, we consider all the 95 different test functions with various properties to assess the general performance of the UP methods. The results are studied to observe and deduct general trends for each UP method.

3.1.5 Application of the UP Methods to Borehole and Steel Column Models

The **Borehole** and **Steel Column** models (Surjanovic and Bingham, 2013), Eq. 28 and Eqs. 29-31, respectively, are used as problems to test the trends and the resulting recommendations generated using the computational experiments. The **Borehole** model has eight input variables and calculates the flow rate in a borehole given its specifications. Table 1 lists the inputs with their distributions. The **Steel Column** model is nine-dimensional, and the input distributions are listed in Table 2. This model evaluates the reliability of a steel column based on the limit state function shown in Eq. 29, Eq. 30, and Eq. 31, which is a criterion of failure.

$$g(x) = \frac{2\pi T_u (H_u - H_l)}{\ln\left({}^{r}/_{r_w}\right)\left(1 + \frac{2LT_u}{\ln\left({}^{r}/_{r_w}\right) r_w^2 K_w}\right) + \frac{T_u}{T_l}} \tag{28}$$

$$g(x) = F_s - P\left[\frac{1}{2BD} + \frac{F_0 E_b}{BDH(E_b - P)}\right] \tag{29}$$

$$P = P_1 + P_2 + P_3 \tag{30}$$

$$E_b = \frac{\pi^2 EBDH^2}{2L^2} \tag{31}$$

Table 1. Distribution of uncertain inputs for Borehole function

Variable	Distribution
radius of borehole (m)	r_w ~ N (μ=0.10, σ=0.0161812)
radius of influence (m)	r ~ Lognormal (μ=7.71, σ=1.0056)
transmissivity of upper aquifer (m^2/yr)	T_u ~ Uniform [63070, 115600]
potentiometric head of upper aquifer (m)	H_u ~ Uniform [990, 1110]
transmissivity of lower aquifer (m^2/yr)	T_l ~ Uniform [63.1, 116]
potentiometric head of lower aquifer (m)	H_l ~ Uniform [700, 820]
length of borehole (m)	L ~ Uniform [1120, 1680]
hydraulic conductivity of borehole (m/yr)	K_w ~ Uniform [9855, 12045]

Table 2. Distribution of uncertain inputs for Steel function

Variable	Distribution
yield stress (MPa)	F_s ~ Lognormal(mean=400, standard deviation=35)
deadweight load (N)	P_1 ~ N(μ=500000, σ=50000)
variable load (N)	P_2 ~ Gumbel(mean=600000, standard deviation=90000)
variable load (N)	P_3 ~ Gumbel(mean=600000, standard deviation=90000)
flange breadth (mm)	B ~ Lognormal(mean=300, standard deviation=3)
flange thickness (mm)	D ~ Lognormal(mean=20, standard deviation=2)
profile height (mm)	H ~ Lognormal(mean=300, standard deviation=5)
initial deflection (mm)	F_0 ~ N(μ=30, σ=10)
Young's modulus (MPa)	E ~ Weibull(mean=210000, standard deviation=4200)

3.2 Results and Discussion

3.2.1 Impact of Nonlinearity on the Performance of Uncertainty Propagation Methods

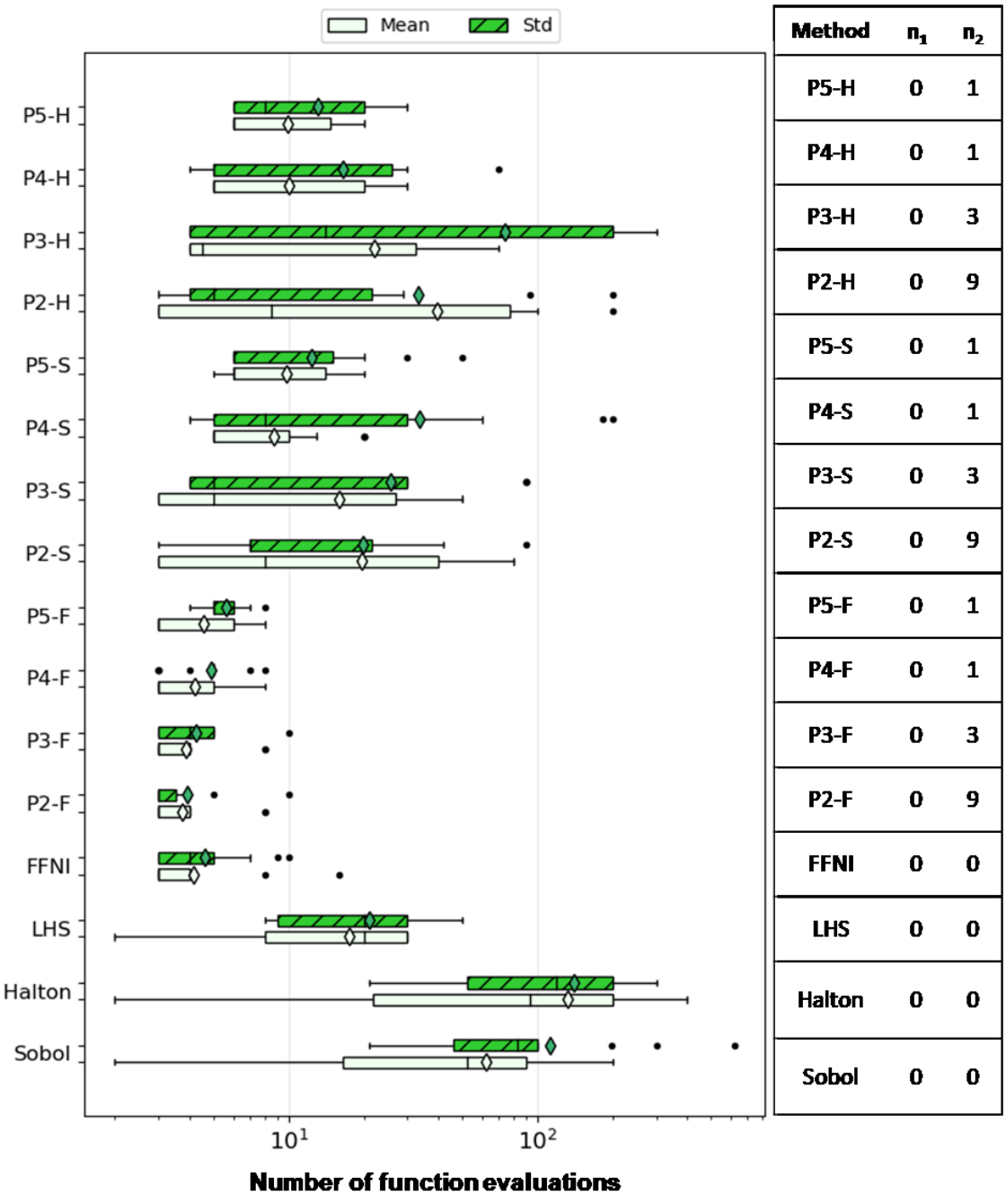
Figures 2 and 3 include boxplots for the minimum number of function evaluations required to converge to a 5% error gap for each of the first four moments for the first group of nonlinearity test functions. In the graphs, P(i)-F stands for the i^{th} order PCE where the integral was estimated using FFNI, P(i)-S using Sobol, and P(i)-H using Halton. The variables, n_1, n_2, n_3, and n_4, are the number of functions for which the method did not yield results within the 5% error gap of the *true* mean, standard deviation, skewness, and kurtosis values, respectively, with one million function evaluations. The plots do not depict results for UDR and SG because all test functions are one-dimensional, and for these functions, the UDR and SG revert to FFNI.

Figure 2 reveals that FFNI and P(i)-F required the lowest number of function evaluations, on average, to yield estimates within the 5% error envelope of the *true* mean and standard deviation. Their interquartile ranges and range of whisker values were the smallest for both mean and standard deviation estimations in comparison to all the methods (Figure 2). The outlier values for FFNI and P(i)-F were lower than the average outlier values of other UP methods. The use of MCS-based methods in PCEs resulted in PCE requiring, on average, a higher number of function evaluations than FFNI and P(i)-F to yield mean and standard deviation estimates within the 5% error gap. For PCE using Sobol (P(i)-S) and Halton (P(i)-H) sampling, the average number of function evaluations required and the interquartile range decreased with an increase in the PCE order. This behavior suggests that the higher-order polynomials aided in representing the nonlinearity in the test functions. However, PCEs with the lower orders, specially $p = 2$, did not yield estimates within the 5% error gap of the *true* standard deviation for a large number of the test functions. The number of test functions for which the standard deviation estimate was within

the 5% error gap increased as the order of the PCEs grew. The MCS-based methods, Sobol and Halton sampling, and LHS needed more function evaluations to estimate the mean and standard deviation (Figure 2). Although the whiskers range was largest for MCS-based methods, the interquartile range was smaller than lower-order P(i)-S and P(i)-H suggesting a peaked distribution for the minimum number of function evaluations in comparison to the PCE methods with approximately same whisker range but larger interquartile ranges.

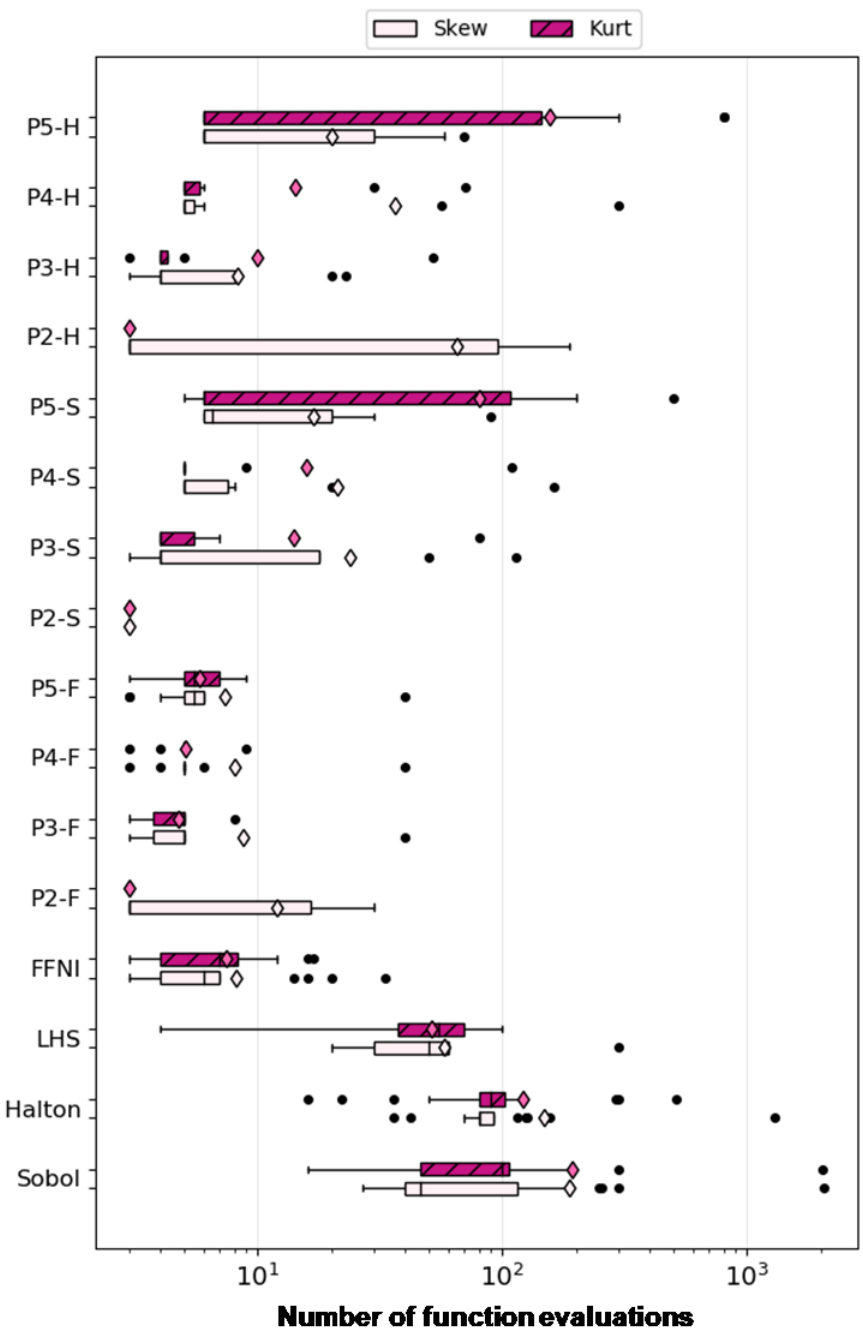
FFNI required the fewest number of function evaluations to estimate skewness and kurtosis (Figure 3) within the 5% error gap of the *true* values for all the functions. Both interquartile and whiskers ranges were smaller than other methods. Higher-order PCE where FFNI is used to calculate the integral (P(i)-F) estimated these two moments with a low number of function calls. However, they did not converge to a 5% error gap for all of the functions. The two lower-order P(i)-Fs did not converge to the error gap for more than 60% of the functions. Sampling using Sobol and Halton sequences and LHS yielded skewness and kurtosis estimates within the 5% error gap for all test functions; however, they required more function calls than FFNI and higher-order P(i)-F on average. The interquartile range of MC-based methods was significantly larger than that obtained by FFNI but smaller than what higher-order P(i)-S and P(i)-H methods yielded. The skewness and kurtosis estimates of PCEs were still not within the 5% error gap at the maximum allowed function evaluations for most functions, especially when lower-order PCEs were used (Figure 3). However, the number of functions where the estimates were not within 5% decreased as the order increased, suggesting that higher-order polynomials were necessary for estimating higher-order moments of the outputs for Sobol and Halton-based PCEs. On the other hand, this is not true for PCEs based on FFNI, since FFNI is very efficient in the prediction of the moments for one-dimensional models, which leads to quick convergence to the desired error gap even with

lower order polynomials. The number of outliers in estimating skewness and kurtosis is larger than the number observed when estimating the mean and standard deviation for all the methods.



Method	n_1	n_2
P5-H	0	1
P4-H	0	1
P3-H	0	3
P2-H	0	9
P5-S	0	1
P4-S	0	1
P3-S	0	3
P2-S	0	9
P5-F	0	1
P4-F	0	1
P3-F	0	3
P2-F	0	9
FFNI	0	0
LHS	0	0
Halton	0	0
Sobol	0	0

Figure 2. The minimum number of function evaluations for estimating mean and standard deviation within 5% of their *true* values for the test functions to study the impact of nonlinearity. n_1 and n_2 are the numbers of functions for which the method did not yield an estimate within a 5% gap of the *true* values for the mean (Mean) and standard deviation (Std), respectively.



Method	n_3	n_4
P5-H	4	4
P4-H	9	10
P3-H	12	12
P2-H	17	18
P5-S	4	1
P4-S	9	10
P3-S	12	12
P2-S	18	18
P5-F	4	4
P4-F	9	8
P3-F	12	12
P2-F	17	18
FFNI	0	0
LHS	0	0
Halton	0	0
Sobol	0	0

Figure 3. The minimum number of function evaluations for estimating skewness and kurtosis within 5% of their *true* values for test functions to study the impact of nonlinearity. n3 and n4 are the numbers of functions for which the method did not yield an estimate within a 5% gap of the *true* values for skewness (Skew) and kurtosis (Kurt), respectively.

Figure 4 summarizes the results for the second group of test functions for evaluating the effect of nonlinearity, where different exponent values are used in the power function. Each figure demonstrates the minimum number of required function evaluations for each UP method to converge to a 5% error gap around the moment for Power functions with different exponent values. Based on Figure 4, MCS-based methods required a higher number of function evaluations to generate accurate estimates of both mean and standard deviation compared to the other methods. Furthermore, the number increased as the value of the exponent rose. The number of function evaluations required by FFNI and PCE where FFNI is used to estimate the integral (P(i)-F) did not change significantly for estimating the mean when the exponent value increased, and the change was the lowest for estimating standard deviation compared to the other methods. PCEs with second-order polynomials with integral estimates carried out using either Sobol or Halton sampling methods, P2-S and P2-H, needed more function calls than the higher-order PCEs for estimating mean and standard deviation of the output for power function with the large exponent values.

Monte Carlo simulation-based methods required more function evaluations to estimate skewness and kurtosis with increases in the exponent (Figure 4). The required function evaluations were the lowest, in general, for LHS for estimating skewness and kurtosis among the three sampling schemas. FFNI needed a significantly lower number of function evaluations than the MCS-based methods for accurate estimation of the skewness and kurtosis for the power function with all considered values of exponents. Furthermore, the change in the number of function calls as the power value rose was minimal compared to MCS-based methods. All PCEs with orders of two and three did not yield third- and fourth-moment estimates within the 5% error gap for the

power function with an exponent of five due to high nonlinearity. As a result, higher orders of PCEs must be used when predicting higher moments.

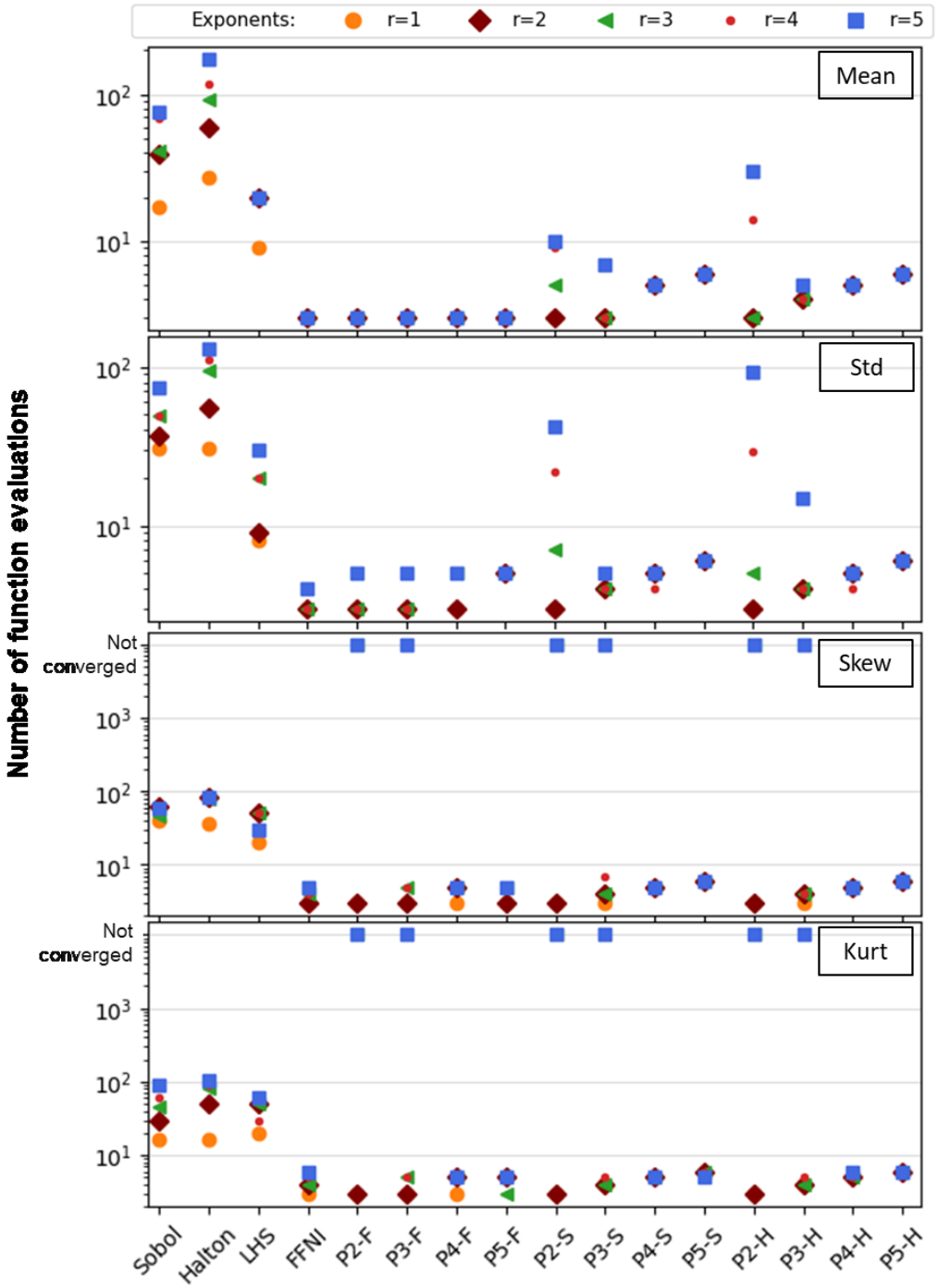


Figure 4. The minimum number of function evaluations for estimating mean (Mean), standard deviation (Std) skewness (Skew), and kurtosis (Kurt) within 5% of their *true* values for the power function with different exponents.

3.2.2 Impact of the Number of Uncertain Inputs on the Performance of Uncertainty Propagation Methods

Figures 5 and 6 plot the minimum number of function evaluations needed by each method to achieve mean, standard deviation, skewness, and kurtosis estimates within the 5% error gap for G and Ackley functions. These plots show the impact of increasing the number of uncertain inputs on the performance of the UP methods. The plots reveal that the minimum number of required function calls to estimate the first four moments increases for all UP methods as the number of uncertain inputs increases. The increase is about an order of magnitude or more function evaluations for each input increment, and it is more significant for FFNI, SG, and P(i)-F for estimating all moments. This result is not surprising because the number of samples required increases exponentially with the number of uncertain inputs for these methods. The impact can be seen clearly for FFNI for the number of inputs above four in both functions. According to the plots (Figures 5 and 6), either the number of function calls to estimate all the moments using FFNI is above 10^5 or FFNI did not converge to the error gap for dimensions larger than four. However, SG converged to a 5% error gap for all moments of the G function and for the mean of the Ackley function for all dimensions within the maximum number of allowed function calls.

The UDR converged to the 5% error gap only up to five and three dimensions for the Ackley function (Figure 6) in estimating the mean and standard deviation, respectively, and did not yield accurate estimates of the other three moments for any of the dimensions. The results for high dimensional functions, especially estimation of higher moments, agree with the expected performance from UDR. The accuracy of the linear combination of univariate functions in representing the test function drop quickly for high dimensions because a higher number of relations between the variables is overlooked by this method. In addition, higher moments contain

strong nonlinearity, which is not well captured by univariate functions. Consequently, moment estimates obtained using these approximations did not converge to the error gap within the allowed number of function evaluations. The UDR is not applicable for the G function (and hence is not included in Figure 5) because the univariate functions are equal to zero when G function is approximated as the linear combination of the univariate functions (these formulas are given in Appendix 1).

Although the MCS-based methods required higher function evaluations, especially for lower dimensions, to estimate the moments, the number of function calls varied less with changes in the number of uncertain inputs compared to other methods. The results demonstrated the MCS-based methods as reliable approaches for estimating the moments as they yielded estimates of all four moments within the error envelope for all the functions with different dimensions. The PCEs where the integrals were approximated using Sobol and Halton sampling methods, in general, converged to the 5% gap of the *true* mean with a lower number of function evaluations compared to numerical integration methods and the PCEs associated with them. This is because the number of samples does not increase exponentially using the low-discrepancy series, unlike the quadrature-based methods. Furthermore, the rate of increase in required evaluations was slow as the dimension increased. The plots also reveal that, as a general trend, the number of required function evaluations increases as the polynomial order increases for PCEs (e.g., mean plot in Figure 6). According to Eq. 19, as the order and dimension of the polynomials for PCEs increase, the number of coefficients, b_i, increase, and based on Eq. 20, a larger number of samples provides more accurate estimates of these coefficients. The number of function evaluations required by PCEs is dependent on two factors, the order of the polynomial and the number of uncertain inputs of the model. Figure 6 illustrates that P2-S and P2-H did not yield estimates within the gap for any number of uncertain

inputs for the Ackley function; however, the higher-order polynomials estimated the skewness and kurtosis within the 5% error gap. These observations suggest that lower-order polynomials were not representing the nonlinearity of functions with higher-order moments accurately. For the G function, the MCS-based PCEs did not converge to the desired gap of the standard deviation with a higher number of uncertain inputs and skewness and kurtoses with almost none of the dimensions (Figure 5). We think the main reason for these results is the high nonlinearity in the G function due to several interaction terms of different uncertain inputs and absolute value function, and added nonlinearity for calculating the second to fourth moments of the output, which is making it difficult for the PCEs to capture the response surface of the G functions accurately.

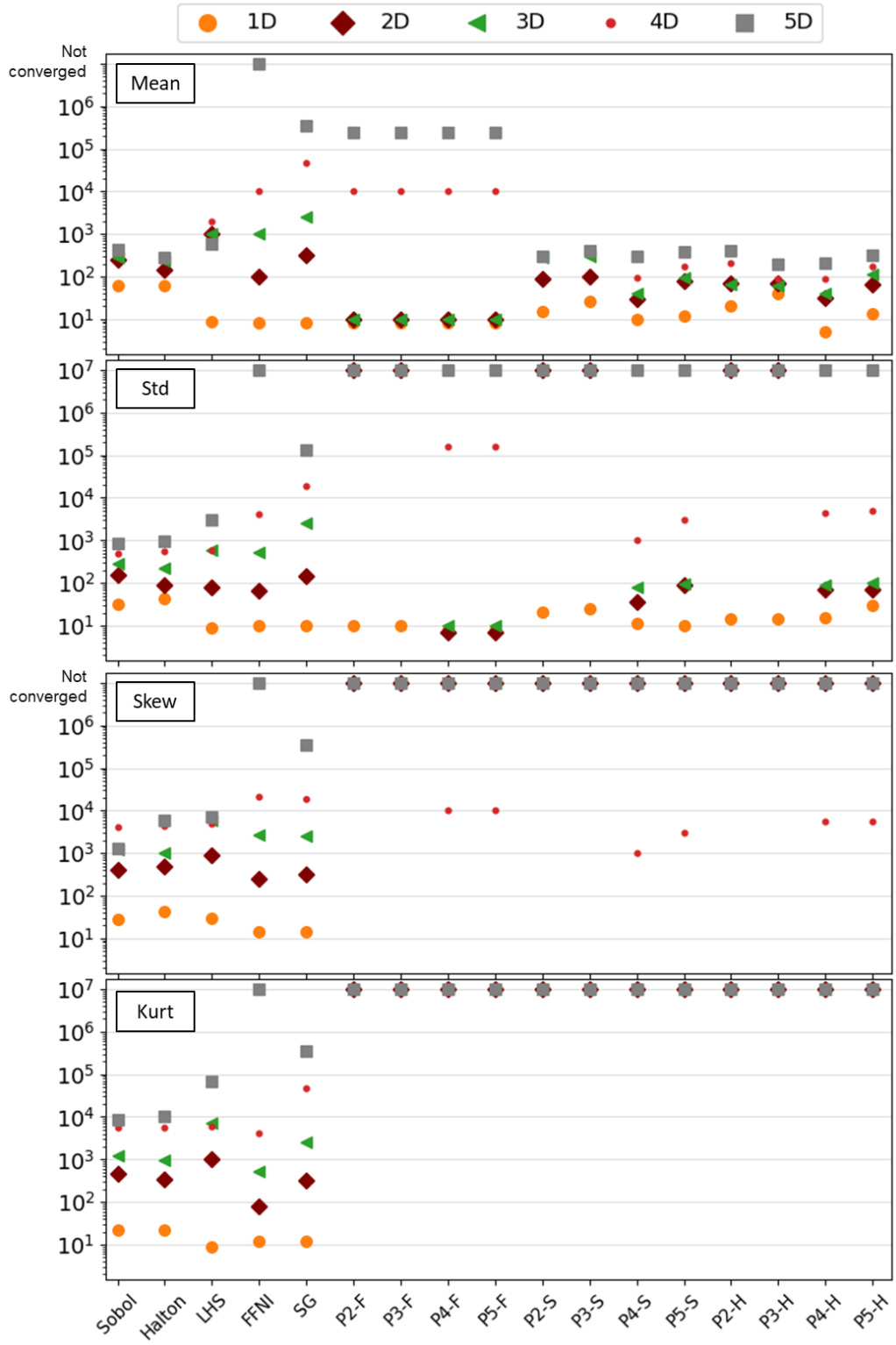


Figure 5. The minimum number of function evaluations for estimating mean (Mean) and standard deviation (Std), skewness (Skew), and kurtosis (Kurt) within 5% of their true values for the case with the impact of dimensionality in G functions.

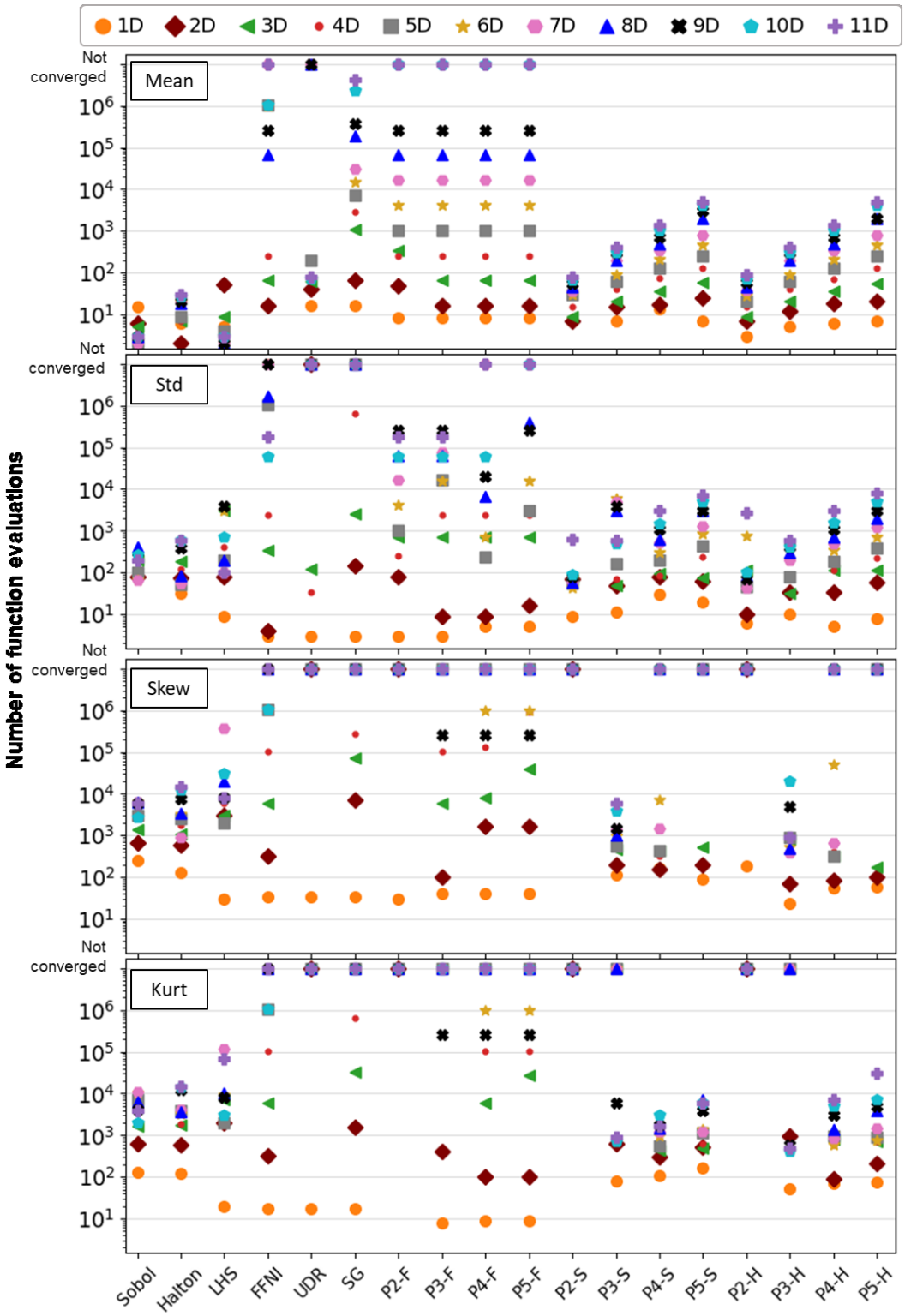


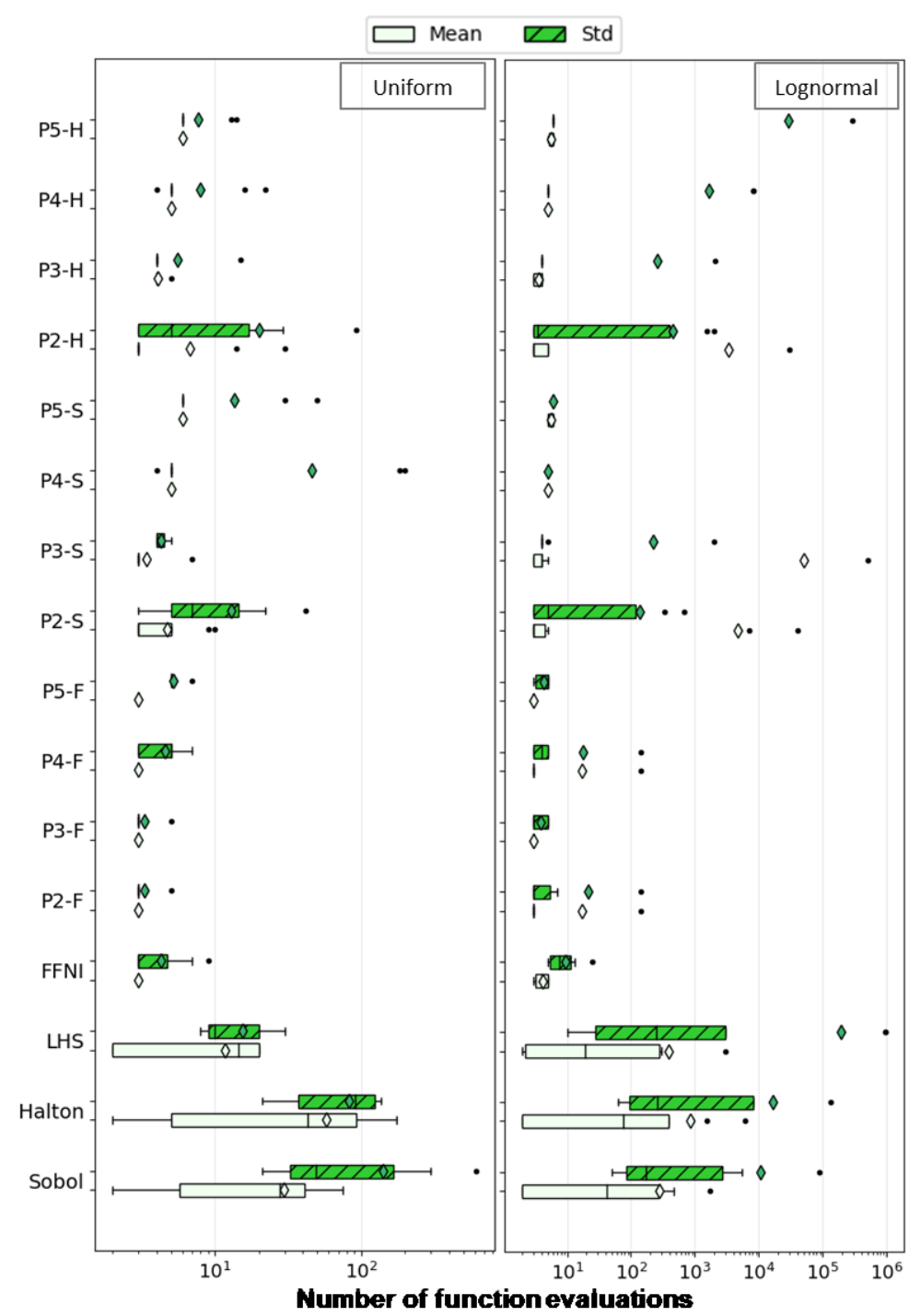
Figure 6. The minimum number of function evaluations for estimating mean (Mean) and standard deviation (Std), skewness (Skew), and kurtosis (Kurt) within 5% of their true values for the case with the impact of dimensionality in Ackley functions.

3.2.3 Impact of the Input Distribution on the Performance of Uncertainty Propagation Methods

Figures 7 and 8 show the box plots of the minimum number of function evaluations to yield an accurate estimate of mean and standard deviation, and skewness and kurtosis, respectively, of the output for the one-dimensional test functions with input distributed uniformly and lognormally. The average number of function calls needed by each UP method is noticeably larger for the lognormal distribution than the uniform one, suggesting that input distribution is an important factor. Additionally, the interquartile ranges increased for lognormal distribution in estimating all the moments with at least one order of magnitude for all methods, and the increase was the maximum for the MCS-based methods. However, the average was located far from the interquartile range for the lognormal distribution case, suggesting the effect of the outliers on the final value of the average and indicating that not all the functions require considerably higher function evaluations to converge to the error gap. Different nonlinear one-dimensional functions were used in the first test group, and similar to the results of the first case study group in Section 3.2.1, the MCS-based methods were the ones with the highest variability in results for both input distributions with the largest interquartile and whiskers ranges. Based on Figures 7 and 8, the MCS-based methods, on average, required the highest number of function evaluations in both uniform and lognormal distributions to converge to a 5% error gap of all moments, and the values were higher for the latter distribution.

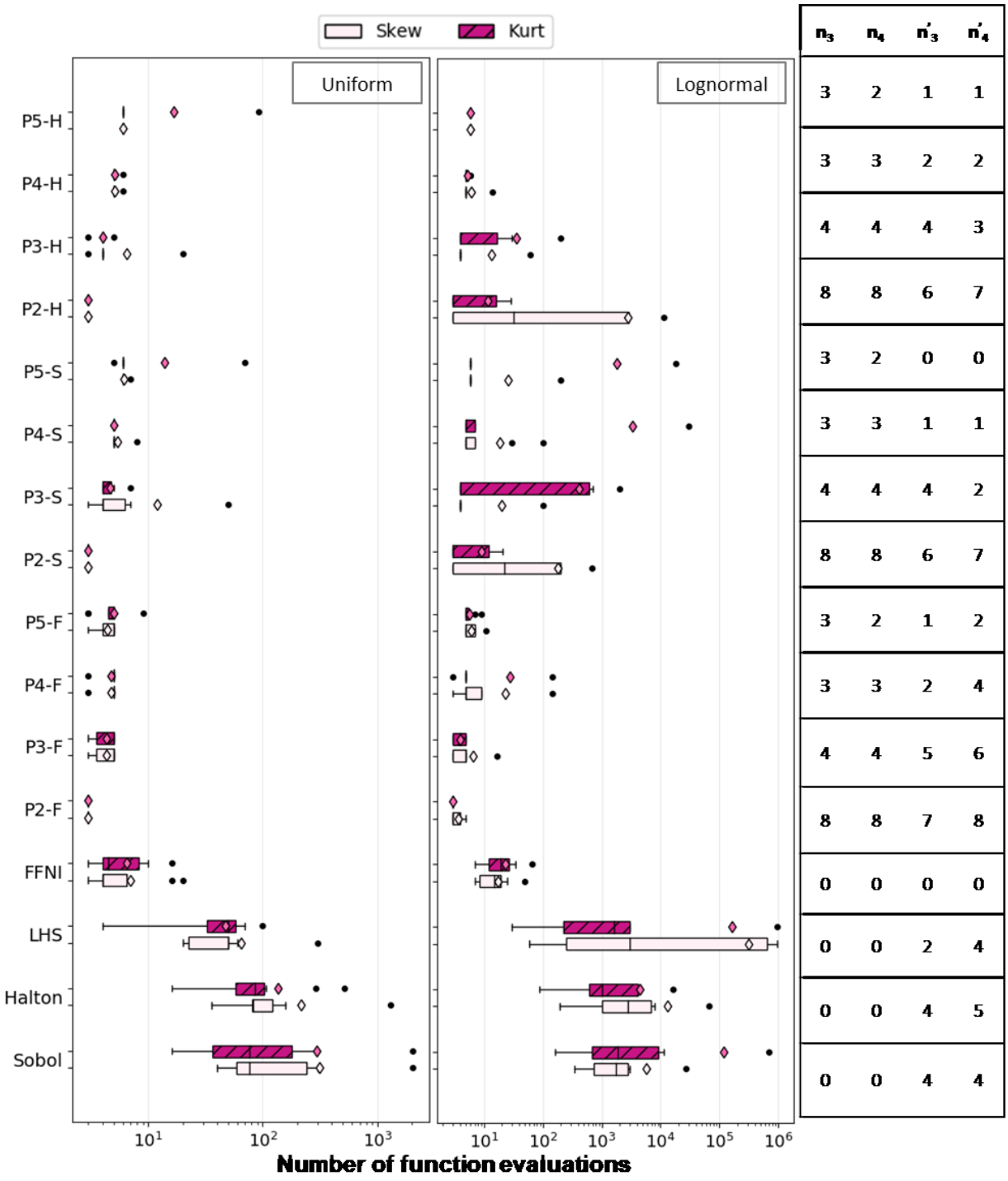
The FFNI had the minimum median and average number of function calls to estimate all moments within the desired error gap for both input distributions. The change in average required function calls between two distributions was the lowest among all the methods. Different quadratures are used for selecting the nodes in FFNI, and the impact from the distributions is

mitigated by the use of appropriate quadratures. The P(i)-F methods had comparable performance to the FFNI method in estimating the mean of the function outputs for both distributions. However, as the moment order increased, the number of function calls for the ones that converged to the error gap and the number of functions for which the P(i)-F methods failed to converge to a 5% error gap for the last three moments increased (Figure 7 and 8). There were no significant differences in the P(i)-F methods estimate quality between uniform or lognormal distributions, possibly due to the same reason for FFNI.



n_1	n_2	n_1'	n_2'
0	1	0	0
0	1	0	0
0	3	2	2
0	3	1	2
0	1	0	0
0	1	1	1
0	3	0	1
0	3	0	2
0	1	0	0
0	1	0	0
0	3	1	1
0	3	0	2
0	0	0	0
0	0	0	0
0	0	0	1
0	0	0	1

Figure 7. The minimum number of function evaluations for estimating mean (Mean) and standard deviation (Std) within 5% of their *true* values for case with impact of uniform distribution. n_1 and n_2 are the number of functions which did not converge to 5% gap of the *true* values for mean and standard deviation of uniform case study, respectively. n_1' and n_2' are the number of functions which did not converge to 5% gap of the *true* values for mean and standard deviation of lognormal case study, respectively.



n_3	n_4	n'_3	n'_4
3	2	1	1
3	3	2	2
4	4	4	3
8	8	6	7
3	2	0	0
3	3	1	1
4	4	4	2
8	8	6	7
3	2	1	2
3	3	2	4
4	4	5	6
8	8	7	8
0	0	0	0
0	0	2	4
0	0	4	5
0	0	4	4

Figure 8. The minimum number of function evaluations for estimating skewness and kurtosis within 5% of their *true* values for the case with impact of uniform distribution. n_3 and n_4 are the number of functions that did not converge to a 5% gap of the *true* values for mean and standard deviation, respectively.

Lower order PCEs with integral approximated using MCS with Sobol and Halton sampling did not yield moments within the error gap of the *true* values for a large number of functions. Higher-order PCEs were better in estimating mean and skewness, where the plots (Figures 7 and 8) do not demonstrate a considerable fluctuation in the number of function evaluations needed by these methods between two different input distributions. The average number of function evaluations required increased significantly for lognormal distribution in estimating the standard deviation and kurtosis. The trends and performance of the UP methods for both distributions agreed with those observed in Section 3.2.1, suggesting that those results can be extended to different distributions.

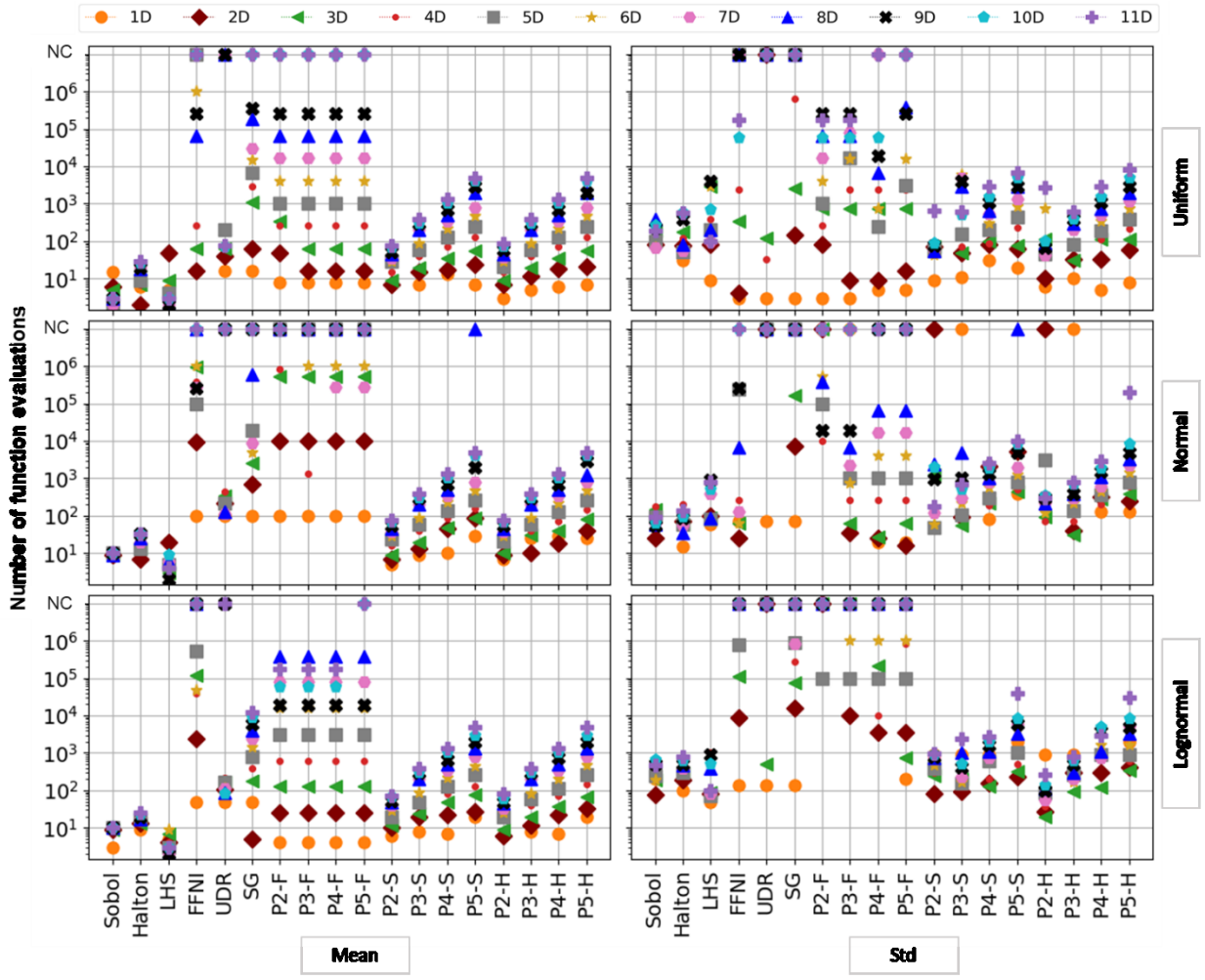


Figure 9. The minimum number of function evaluations for estimating mean (Mean), and standard deviation (Std) within 5% of their true values for the case with the impact of distributions in Ackley functions. NC (Not Converged) indicates the methods which were not able to converge to the desired gap within 106 function evaluations.

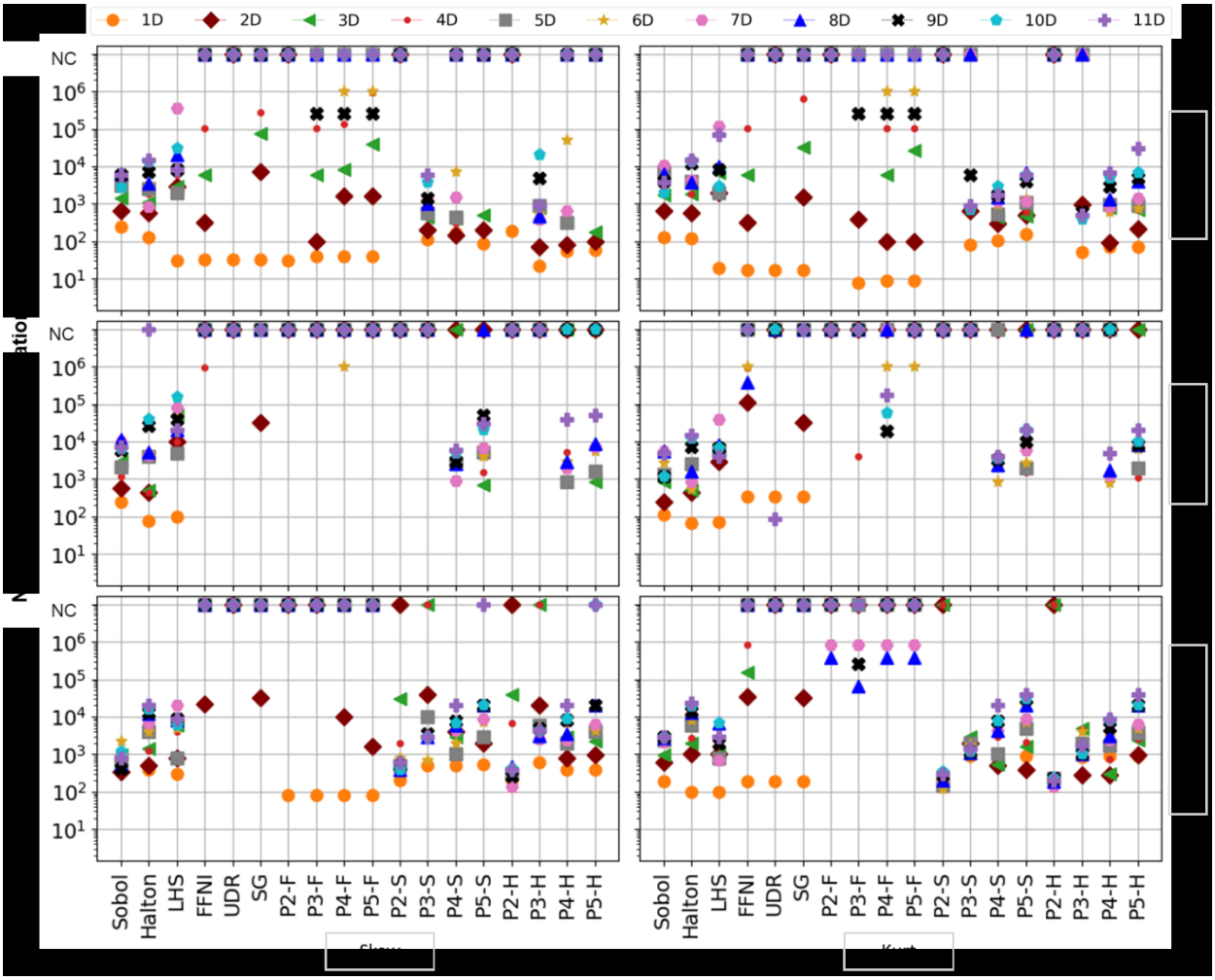


Figure 10. The minimum number of function evaluations for estimating skewness (Skew) and kurtosis (Kurt) within 5% of their true values for the case with the impact of distributions in Ackley functions. NC (Not Converged) indicates the methods which were not able to converge to the desired gap within 106 function evaluations.

In the second case study on the impact of input distributions, the minimum number of function calls to estimate the four moments for the Ackley function with three different distributions for the inputs are shown in Figures 9 and 10. According to the figures, FFNI demonstrates large changes in the number of function evaluations to estimate the output moments among different distributions for dimensions above two. The number of function calls does not change drastically for the UDR method in estimating the mean of the Ackley function. However, UDR was not able to converge to the error gap for most cases with higher moment orders. The

performance of the MCS-based methods varies less, i.e., the required number of function evaluations does not change significantly, in comparison to other methods. The PCEs based on Sobol and Halton sampling are not affected by the distribution of the input variables and require the same number of function evaluations for cases that converged to a 5% error gap of each moment.

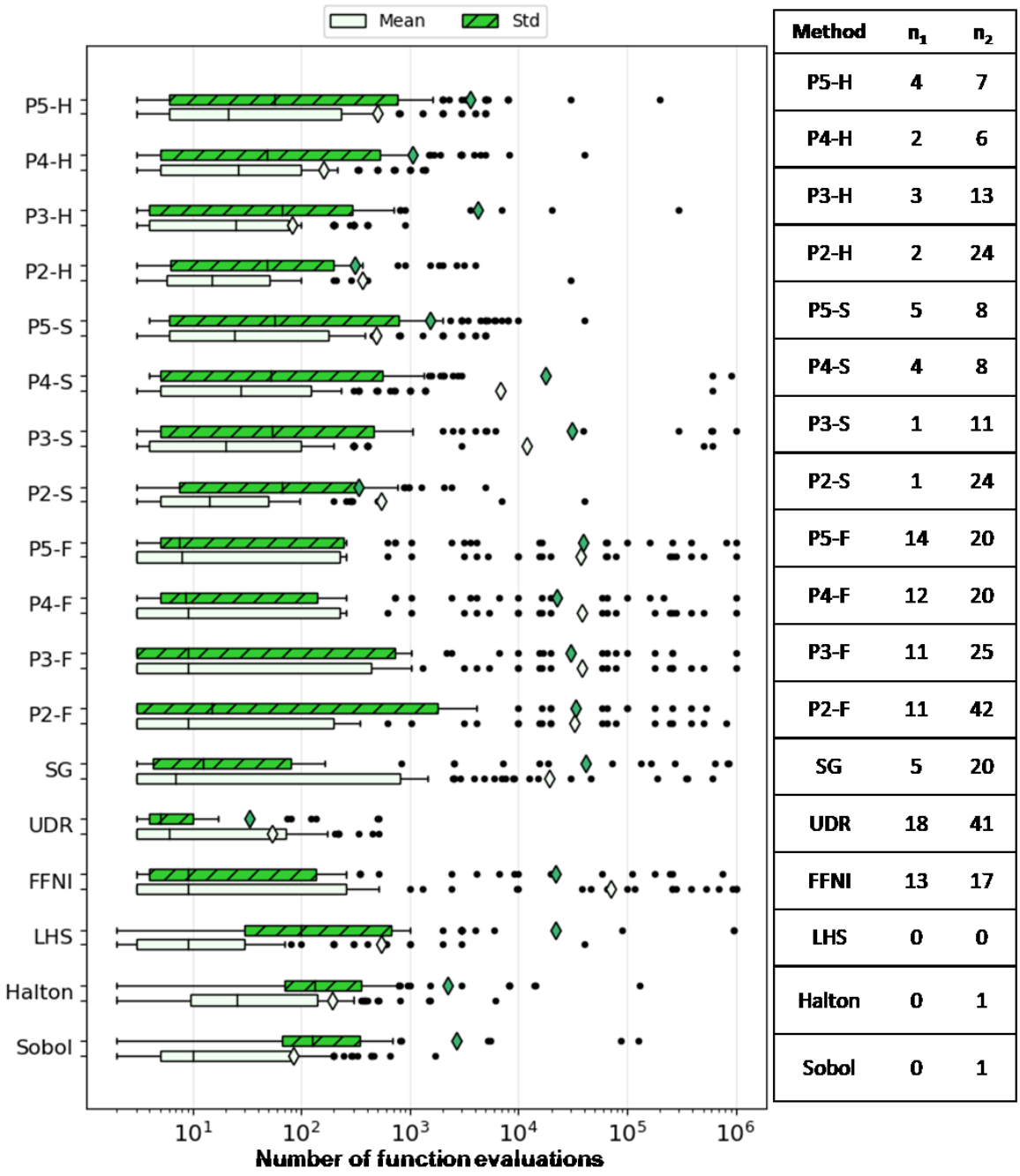
3.2.4 Comparison of the Performance of Uncertainty Propagation Methods for all Test Functions - Overall Performance Analysis

Figure 11 shows box plots of the minimum number of function evaluations required to estimate the mean and standard deviation within the 5% error gap of *true* moment values for all the test functions considered in this study. The results are important for problems where not all the characteristics of the models are known or given.

The plots in Figure 11 demonstrate that MSC-based methods outperformed the other methods in four aspects. First, they converged to the 5% error gap in estimating both mean and standard deviation for almost all the functions. Second, the median and the average number of required function evaluations are lower than the other UP methods. Third, the interquartile and whiskers ranges are the smallest in comparison to other methods converging to the error gap for the majority of the functions. Finally, the number of outliers is the lowest in comparison to any other method. The two prohibiting factors in estimating the moments, which are the dimensionality curse for numerical integration methods and the ability to represent the nonlinearity of the systems using PCEs, do not apply to MCS-based methods. Hence, they are more efficient in terms of the required number of function evaluations compared to all the other ones in this study.

Numerical integration methods and P(i)-F are the least efficient UP methods because the minimum number of required function evaluations to estimate the mean and standard deviation

using these methods is strongly affected by the number of uncertain inputs and grows quickly. There were many test functions with more than one uncertain input. As a result, FFNI, UDR, and P(i)-F could not yield accurate estimates of the first two moments for a larger number of the test functions, more than 10%, compared to other methods. However, among the numerical integration methods, SG converged to the error gap within the allowed number of function evaluations for a higher number of functions, and the medians of the number of function calls were less than the medians of MCS-based methods. As the order of the moments increased, the number of functions for which the estimation of standard deviation through SG was unsuccessful grew larger.



Method	n_1	n_2
P5-H	4	7
P4-H	2	6
P3-H	3	13
P2-H	2	24
P5-S	5	8
P4-S	4	8
P3-S	1	11
P2-S	1	24
P5-F	14	20
P4-F	12	20
P3-F	11	25
P2-F	11	42
SG	5	20
UDR	18	41
FFNI	13	17
LHS	0	0
Halton	0	1
Sobol	0	1

Figure 11. The minimum number of function evaluations for estimating mean and standard deviation within 5% of their *true* values for the case of general performance. n_1 and n_2 are the numbers of functions that did not converge to a 5% gap of the *true* values for the mean (Mean) and standard deviation (Std), respectively.

According to Figure 11, the function calls required by the P(i)-S and P(i)-H to converge to the desired error gap of mean does not have decreasing trend as the order of the polynomials increases. We assume this was due to the low order of the mean as the first moment, where the degree of nonlinearity is lower than in the other moments. On the other hand, the change in the number of functions that these methods did not converge to the error gap of standard deviation, n_2, is notable as the order of the polynomial rises in value. The P5-H and P5-S estimated the standard deviation within a 5% error gap for more than 90% of the test functions. This percentage is larger than 75%, the percentage of functions whose standard deviation estimates were within the error gap using the second-order PCEs. The mean number of function calls used for the first two moment approximations by P(i)-S and P(i)-H were lower than numerical integration methods and P(i)-Fs, but it was significantly larger than MCS-based methods. PCEs do not guarantee convergence as the number of function calls increases, whereas, with MCS-based models, the estimation converges to the desired value at some point if the number of function calls is large enough. If the PCE with chosen order does not represent the nonlinearity accurately, then increasing the number of function evaluations would give accurate estimates of the built PCE, which is not necessarily representative of the desired function. On the other hand, MCS-based methods, using space-filling or low discrepancy sampling methods, cover larger amounts of the input(s) space as the number of samples and consequently the number of function evaluations increase and yield more accurate estimates of the moments as a result of the higher number of data.

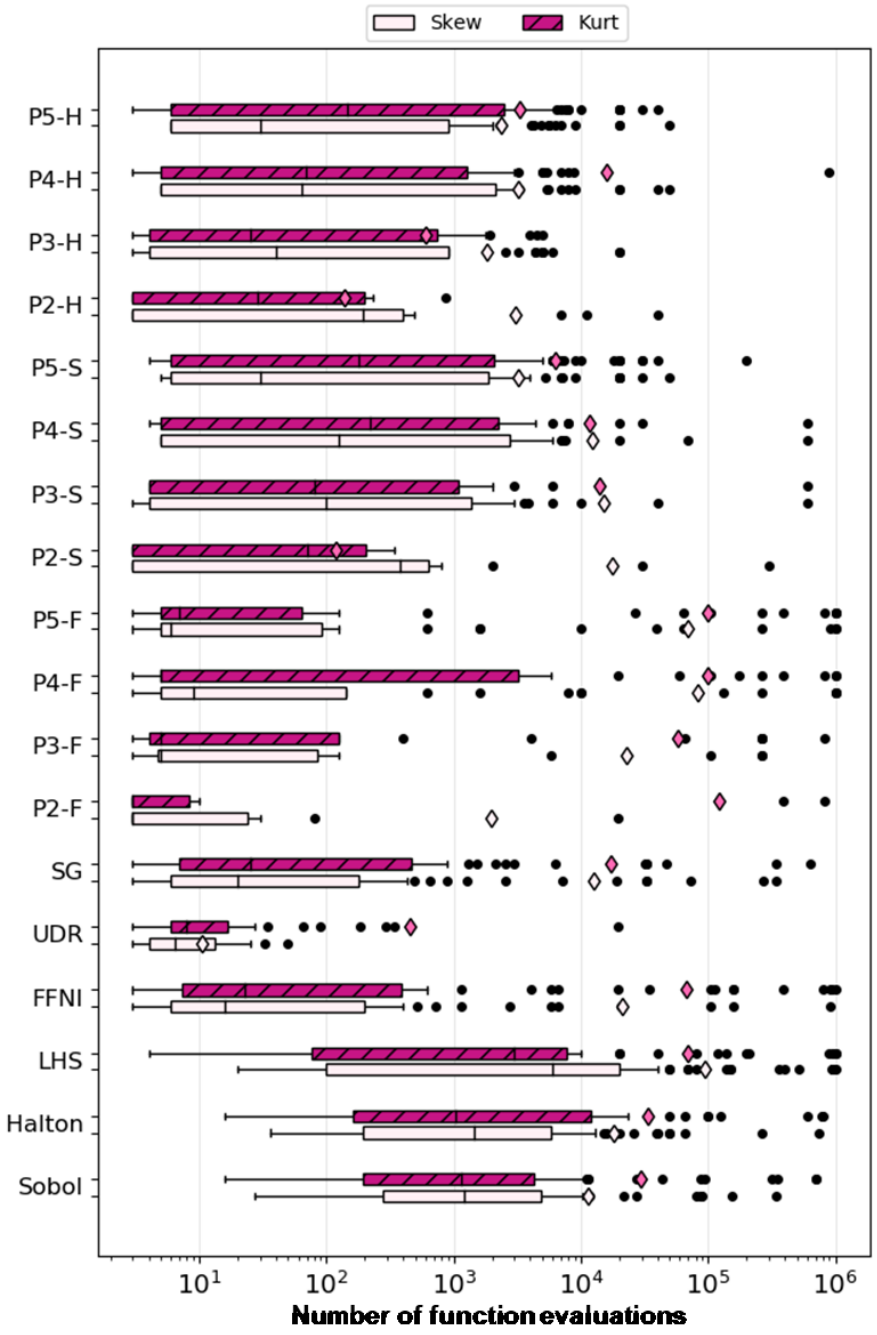
Figure 12 illustrates the box plots of minimum required function calls to estimate the skewness and kurtosis within a 5% error bound of the *true* values for all the 94 test functions used in this paper. MCS-based methods accurately estimated the third and fourth moments for more than 90% of the test functions. Thus, they were the most reliable methods for converging to the

desired error gap. FFNI and SG predicted skewness and kurtosis of a larger number of functions, more than 65% and 75%, respectively, in comparison to the UDR and P(i)-F, which was less than 50% for estimating both skewness and kurtosis. The median of the number of required function calls for the functions the numerical integrations methods were able to converge to the desired error gap was lower than the median for MCS-based methods, suggesting that if the numerical integrations are an appropriate method for the function characteristics, they are more likely to be efficient in estimating all moments accurately. Similar conclusions can be drawn for PCEs as well. For the cases in which PCEs converged to the 5% error gap of the *true* values of the moments, the mean and median of the number of the demanded function calls are less than the MCS-based methods. Hence, for the cases where the PCEs can approximate the nonlinearity of the model, the number of function evaluations is not large. Even though the performance of the PCEs was comparable to the MCS-based methods for estimating the mean, the performance significantly deteriorated for the higher moments, skewness and kurtosis, as can be seen with the higher values of n_3 and n_4 for PCEs versus MCS-based methods in Figure 12. The increase in the order of the polynomials improved the performance of the PCEs in converging to the error bound for a larger number of functions since the higher orders were able to represent the nonlinearity of the functions with better accuracy. The second-order PCEs had the highest number, more than 80%, of functions for which the convergence to the 5% error gap was not achieved. Therefore, based on Figure 12, they are not recommended to estimate the skewness and kurtosis.

3.2.5 Results for the Application of the Uncertainty Propagation Methods to Borehole and Steel Column Models

Based on all the previous case studies, it is expected that MCS-based methods to be reliable methods and converge to the 5% error gap of all moments for both Steel and Borehole models and

be one of the methods with a lower number of required model evaluations. Due to the high number of uncertain inputs for both models, SG among numerical integration methods is expected to have the best performance and require lower numbers of model calls if converged to the error gap. PCE-based methods are expected to have better or the same performance as MCS-based methods if they converge to the desired error gap of the *true* values of the moment, and the number of model evaluations should increase as the order of PCE gets larger.



Method	n_3	n_4
P5-H	31	20
P4-H	34	34
P3-H	49	50
P2-H	74	77
P5-S	28	20
P4-S	32	32
P3-S	49	49
P2-S	75	76
P5-F	46	47
P4-F	53	55
P3-F	66	65
P2-F	84	84
SG	31	27
UDR	54	48
FFNI	35	24
LHS	5	8
Halton	9	8
Sobol	10	9

Figure 12. The minimum number of function evaluations for estimating skewness and kurtosis within 5% of their true values for the case of general performance. n_3 and $n_4$are the numbers of functions that did not converge to a 5% gap of the true values for skewness (Skew) and kurtosis (Kurt), respectively.

The mean-standard deviation and skewness-kurtosis estimation for Steel and Borehole models are plotted in Figures 13 and 14, respectively. The figures show the minimum number of required model calls required to converge to a 5% error gap of the moments. MCS-based methods

converged to the desired error gap of the mean with the lowest number of function evaluations. However, lower-order (2 and 3) PCEs that employed Sobol and Halton sampling also needed relatively low function evaluations to reach 5% of the *true* values. The number of function evaluations increased as the order increased, and for the Steel model, the PCE with orders of 4 and 5 did not converge to the chosen error gap, which suggests that these polynomials were not able to represent the nonlinearity of the Steel model. SG had the best performance among all the numerical integration methods with the lowest number of model evaluations to estimate all the moments within the 5% error gap of their *true* values. FFNI and the PCEs associated with it required a higher number of function calls due to the relatively high number of uncertain inputs in the models.

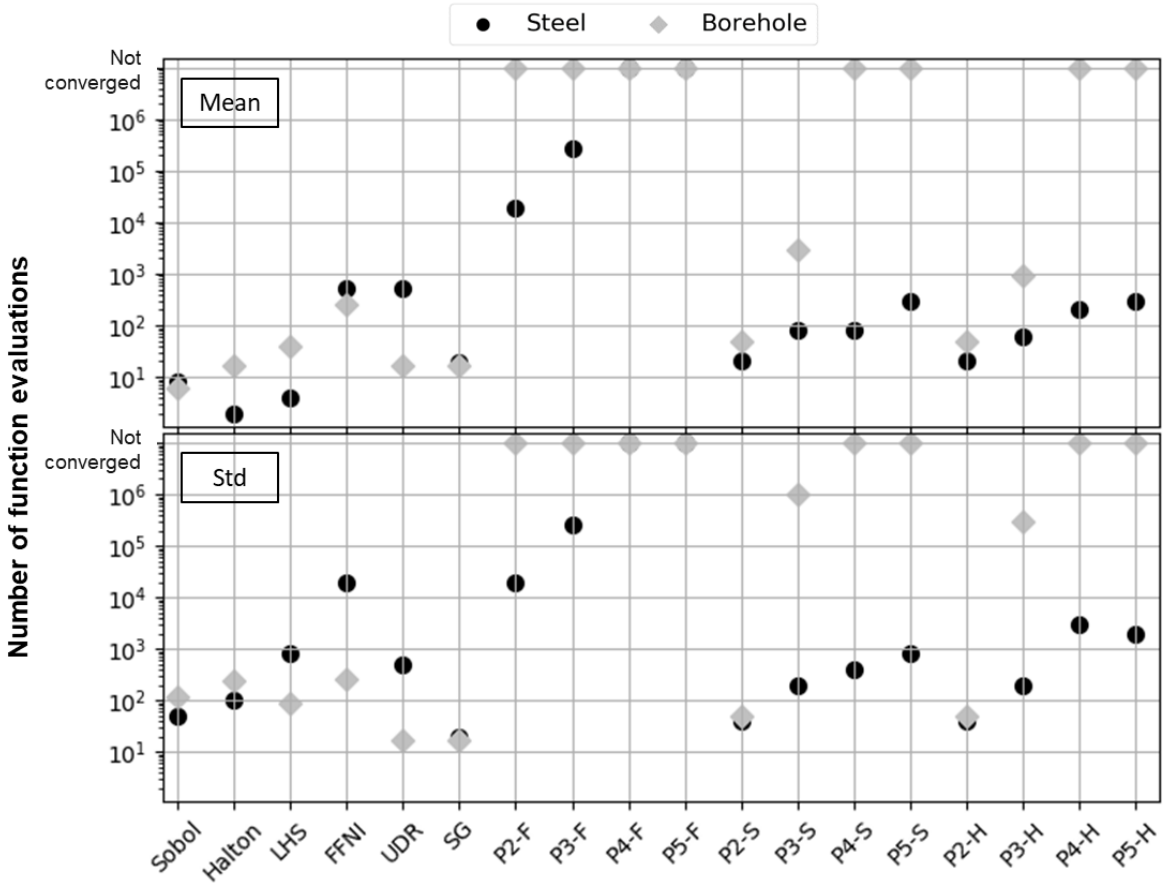


Figure 13. The minimum number of function evaluations for estimating mean and standard deviation within 5% of their true values for Steel and Borehole models.

For the Borehole model, none of the PCEs estimated the skewness and kurtosis within the desired error gap. The MCS-based methods and FFNI performed the best for this model, requiring the lowest number of function evaluations. On the other hand, PCEs converged to the desired gap with the number of function evaluations very close to the Sobol, Halton, and LHS for Steel model. As the order grew, there was an increasing trend for the required function evaluations of PCEs. The results for the Steel and Borehole models agree with the observations and conclusions drawn from the earlier case studies.

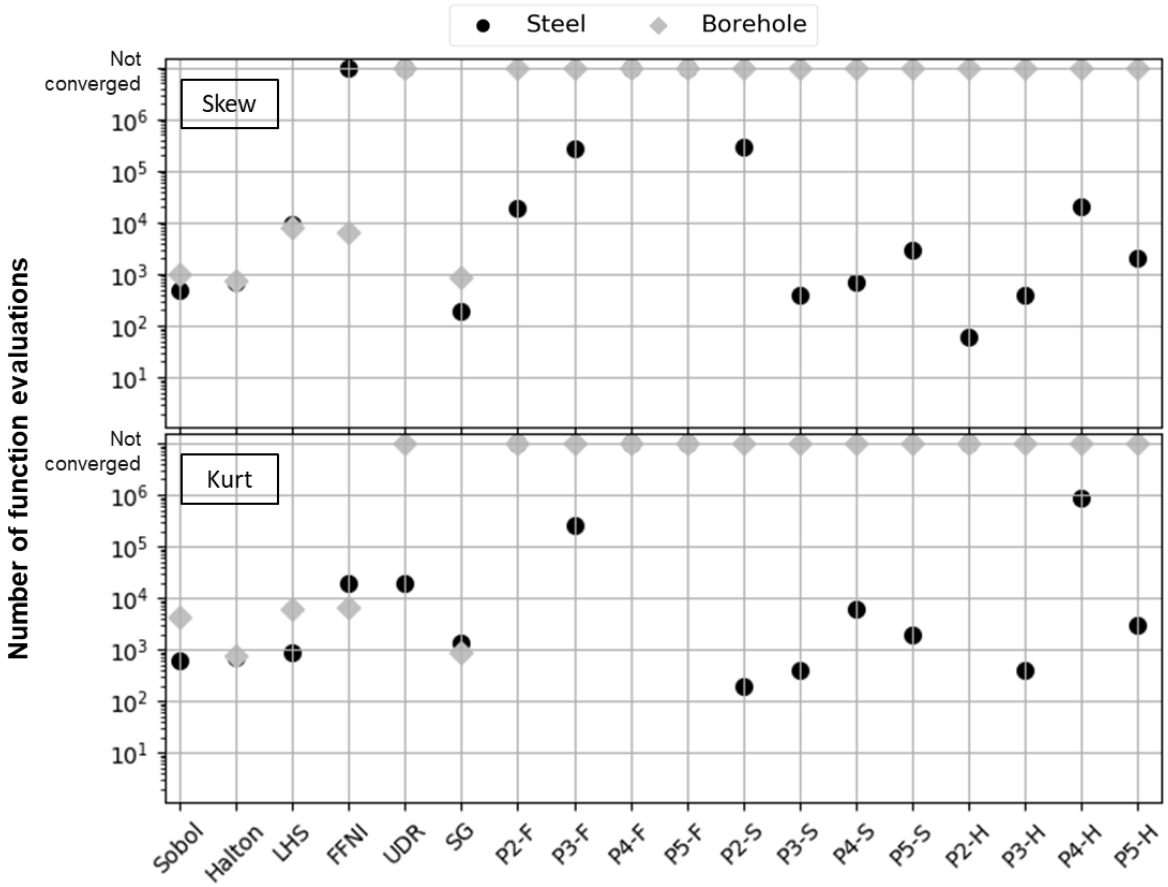


Figure 14. The minimum number of function evaluations for estimating skewness and kurtosis within 5% of their true values for Steel and Borehole models.

3.3 Conclusions

This chapter studied the performance of seven methods from three common groups of uncertainty propagation (UP) methods, including Monte Carlo simulation-based methods, numerical integration methods, and functional expansion-based methods, using computational experiments. The methods were the Monte-Carlo method using Sobol series, Halton series, and Latin Hypercube sampling (LHS), numerical integration methods of Full Factorial Numerical Integration (FFNI), Univariate Dimension Reduction (UDR), and Sparse Grids (SG), and Polynomial Chaos Expansion (PCE) as the function expansion-based method. The study evaluated the impact of model characteristics, such as the number of uncertain inputs, non-linearity of the

model, and uncertainty distribution type using 95 different test functions. The uncertainty propagation methods were compared based on the accuracy of the output estimates and the methods' efficiency in yielding these estimates. The accuracy was assessed using the first four statistical moments of the model output, and the efficiency was assessed using the minimum number of model calls required to reach and remain within the 5% error gap of the *true* values of these moments.

The efficiencies of the FFNI and PCEs that utilized FFNI strongly depended on the number of uncertain inputs. PCEs generally estimated the first two statistical moments accurately but did not converge to the desired error gap for skewness and kurtosis (third and fourth moments) for most test functions. The nonlinearity of the test functions considerably impacted the performance of the Monte-Carlo simulation-based method compared to the other methods, irrespective of the sampling technique used, and they required a higher number of model evaluations for complex functions. However, these methods were the most reliable UP methods and converged to the chosen error gap of all the moments for most (88%) test functions.

In light of our computational experiments, we constructed the following guidelines for selecting the UP method based on the characteristics of the model of interest: For models with less than five uncertain inputs, regardless of the nonlinearity, FFNI is generally the most efficient method to estimate the first four moments. SG is a better choice among numerical integration methods when the number of uncertain inputs is higher than five. As the number of uncertain inputs increases, for the models with high nonlinearities, such as models that contain high power values, logarithmic, and trigonometric functions with possibly interacting terms, higher-order PCEs can be used to estimate the mean and standard deviation. However, higher-order PCEs are not the most reliable methods for estimating higher-order moments, such as skewness and kurtosis,

because they did not converge to the error gap for most test functions. Although higher-order PCEs may capture the model nonlinearity, it must be noted that this increase in order results in a higher number of coefficients that need to be estimated. This, in turn, translates into a lower efficiency for the PCE methods. There is a trade-off between accuracy and efficiency with higher-order PCEs. Finally, Monte Carlo simulation-based methods are recommended for models with a high number of uncertain inputs and are reliable for estimating all four statistical moments of these models. The results from this study did not yield clear guidelines based on the distribution type, as the trends observed were not consistent for the test functions.

Chapter 4 – **Machine Learning Methods' Applications and Comparison**

4.1 Classification of Cardiomyocytes Content Differentiated from hiPSC in Hydrogel Capsules

Cardiovascular diseases have been and continue to be the leading cause of mortality worldwide due to the limited self-renewal capacity of adult human cardiomyocytes (CMs) after myocardial infarction (Kempf et al., 2016a). It has been estimated that billions of CMs per patient are needed to treat myocardial diseases effectively (Zweigerdt et al., 2011). The discovery of human-induced pluripotent stem cells (hiPSCs) in 2006 (Takahashi and Yamanaka, 2006) enables and facilitates generating *in vitro* cardiomyocytes (hiPSC-CMs). Current two-dimensional (2D) cell culture approaches result in highly reproducible differentiation of hiPSC to CMs; however, these 2D systems cannot recapitulate key physiological attributes of the complexity of the native myocardium. Furthermore, the large surface area required limits their applicability for scale-up for large-scale cell production. To overcome these 2D cell culture problems, tissue engineering offers three-dimensional (3D) cell culture approaches that can better mimic the *in vivo* conditions (Zhao et al., 2020) and have shown promise for the culture and differentiation of hiPSCs for scalable production of hiPSC-CMs (Kempf et al., 2014; Zhao et al., 2020). Although 3D cell culture suspension has the potential to be used for scaling up hiPSC-CM production, inconsistency within and between batches remains a challenge (Hamad et al., 2019). Therefore, the source of this variability in scalable, suspension-based production of hiPSC-CMs must be recognized and understood using reliable tools (Kropp et al., 2017).

Complicated systems and the high number of affecting variables make the development and optimization of biological processes challenging. In recent years, machine learning techniques

have been used as a new toolset to elucidate part of the complexities that exist in biological processes and identify critical variables in systems (Li et al., 2018; Xu and Jackson, 2019). Machine learning methods build algorithms to classify or predict a desired outcome based on the available data. Various machine learning techniques are used in bioprocesses development to reduce the complexity of the systems and help with identifying key variables. For example, Wu et al. (Wu et al., 2016) used support vector machines, K-nearest neighbor, and decision trees to predict the metabolic fluxes of heterotrophic bacteria, and Sciascio and Amicarelli (di Sciascio and Amicarelli, 2008) predicted the biomass in batch biotechnological processes using Gaussian process modeling.

Cell production is an intense bioprocess with high variability, and machine learning methods have shown remarkable performance in predicting the outputs for these systems. Mehrian et al. (Mehrian et al., 2020) predicted the population doubling time of cells in human mesenchymal stromal cell expansion using random forest models, which had significantly better performance than theoretical estimates. Cunha et al. (Cunha et al., 2019) assessed the differentiation and proliferation of stem cells using artificial neural networks, and Schmidt-heck et al. (Schmidt-Heck et al., 2005) implemented support vector machines and random forest models to predict the performance of a bioreactor producing human liver cells.

Producing CMs is a multi-factor process with complicated interactions, making experimental testing time-consuming and expensive. Hence, different applications of machine learning methods can be found in the field of hiPSC-CMs to account for the complexities in the system and provide models and tools to be used for different applications of hiPSC-CMs. Models like the random forest, k-nearest neighbors, and decision trees were implemented to analyze drug effects on CMs and to build classification tools (Heylman et al., 2015). Lee et al. (Lee et al., 2017) used support

vector machines to predict the impacts of drugs with different components on cardiac functionality. They constructed a multiclass model, based on the influence on cardiac functionality, capable of classifying a library of drugs with different components that were not included in the training data. Orita et al. (Orita et al., 2020) classified the contractility of hiPSC-CMs using support vector machines as a metric to assess the functional quality of the cultured hiPCS-CMs. Naïve Bayes, support vector machines, and k-nearest neighbors were used to classify the diseased and abnormal CMs (Hwang et al., 2020; Teles et al., 2021). Williams et al. (Williams et al., 2020) predicted the outcome for producing CMs, which were differentiated from hiPSCs in a bioreactor, using different models, such as random forest and Gaussian process modeling. All of these studies demonstrated the promising results of using machine learning techniques to classify, predict, and select features for constructing accurate predictive models for 3D CM production systems and their potential to be used in the scale-up analysis.

A single-step cell handling approach has previously been established for hiPSC encapsulation and direct differentiation in a 3D engineered tissue microenvironment. The approach employs a novel and cost-effective microfluidic system (Finklea et al., 2021; Seeto et al., 2019) to create microtissues appropriate for scaling up the production of hiPSC-CMs. Creating cardiac microspheres with >75% CMs content by encapsulation of hiPSC within PEG-fibrinogen (PF) has been reported by Finklea et al. (Finklea et al., 2021). Furthermore, this microfluidic platform is highly flexible and can rapidly fabricate hiPSC-laden microspheroids with different sizes (diameter), shapes (axial ratios), and cell densities in PF (Tian and Lipke, 2020). Understanding the impact of these tunable parameters along with different differentiation protocols and protein concentration in PF, as controllable features, and leveraging the use of the microfluidic PF encapsulation system to adjust them, has the potential to advance reliable and reproducible

scalable, suspension culture-based engineered cardiac tissue production. The ability to predict the final CM content is a crucial step for increasing the efficiency of hiPSC-CM production due to the expensive and time-consuming nature of the CM differentiation process and the challenges of downstream processing to remove non-CMs. The most influential experimental variables, individually or in combination, are not well understood for this microfluidic system. Additional insight is essential to improve the experimental conditions and ultimately increase the consistency and efficiency of hiPSC to CM differentiation at larger scales.

In this study, machine learning techniques are used for two main goals. The first goal is identifying the most influential variables for classifying the CM content. The second one is to classify the CM content on a specific day, i.e., day 10, of differentiation. Identification of the predictive parameters and an accurate classification model can be used to optimize the process by interrupting the processes leading to failure, consequently saving money and time. In addition, the informative parameters can lead to further modifications in the system for higher CM content. The experiments for CM production were carried out by Dr. Lipke's group using the microfluidic system for hiPSC encapsulation on day -3 (three days before the start of differentiation, day 0) and collecting and analyzing the CM content in microspheroids on day 10 of differentiation (First and second rows in Figure 15). The information from the experiments was used for the computational analysis. In addition to the experimental features (variables), several other features were generated using feature engineering techniques to include more information in the input feature set for the model training step (third row in Figure 15). Using different feature selection methods, subsets of the initial input sets were chosen to construct classifier models (fourth row in Figure 15). Although different studies reported different hiPSC-CM percentages for successful CM differentiation in 3D platforms (Branco et al., 2019; Dahlmann et al., 2013; Fonoudi et al., 2015; Williams et al., 2020),

in the context of this study, the yield of 65% hiPSC-CM content or higher on differentiation day 10 is considered *sufficient* based on the recent experience in the hiPSC microspheroid encapsulation and cardiac differentiation (Finklea et al., 2021). In contrast, less than 65% CM content is defined as *insufficient* class. The classifier models are built using three different machine learning techniques and compared in terms of their performance (the last row in Figure 15). Ultimately, the features used in constructing the best-performing model based on the comparison metrics were selected as the most influential features of the system.

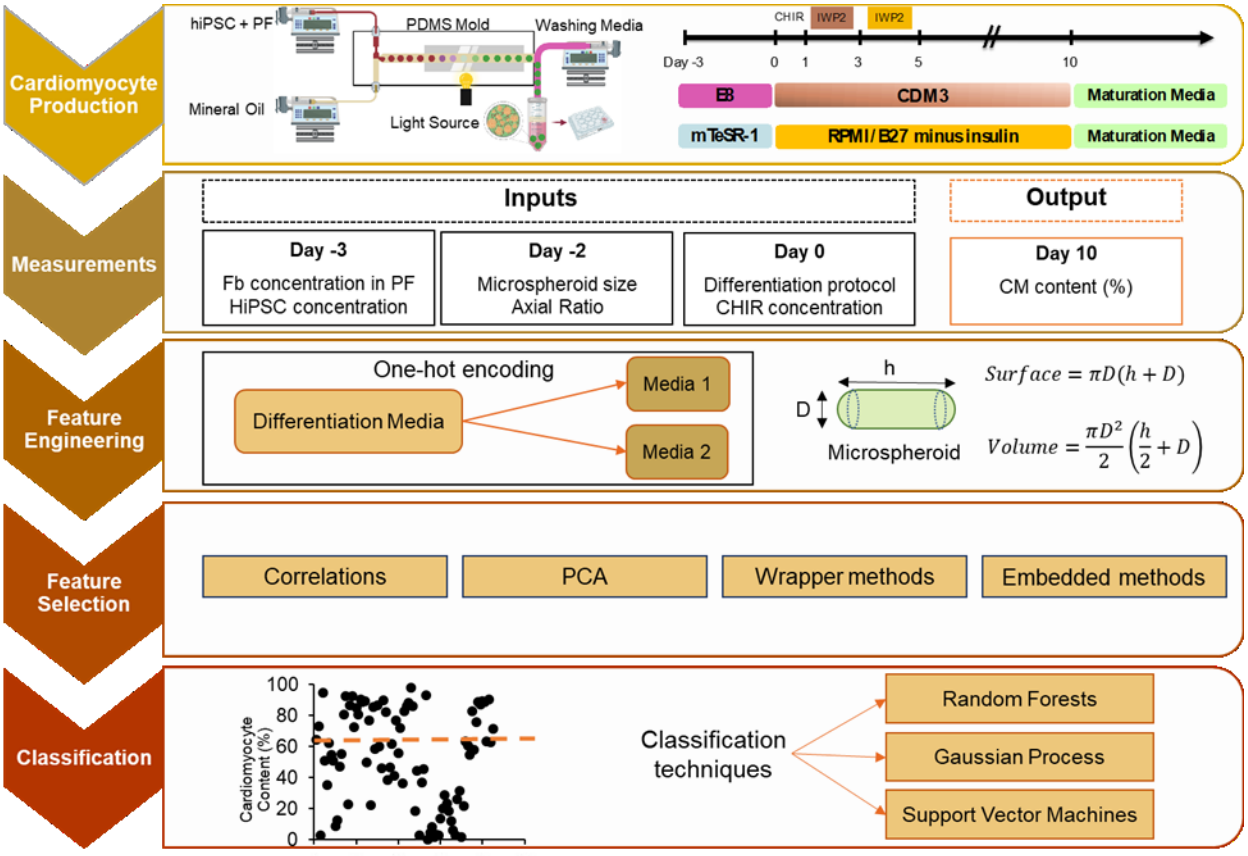


Figure 15. Schematic of the process to generate experimental data for creating a data-driven model of predicting CM content from cell encapsulation experiments. Human-induced pluripotent stem cells (hiPSCs) were resuspended in PF precursor solution including the photo initiator at a concentration of 30, 40, 50, and 60 million cells mL^{-1} of PF. The precursor solution and mineral oil were infused into the top and bottom inlets of the PDMS mold, respectively. With breaking the surface tension of the precursor solution, the microspheroids of hydrogel were created and crosslinked in the outlet by using visible light. Microspheroids were collected at the end of the PDMS mold and

removed from the oil phase and cultured in two different stem cell media for expansion for 3 days. Cardiac differentiation was initiated on day 0 using two different differentiation protocols as shown in the timeline. Experimental features which were measured and intentionally changed were fibrinogen concentration in PF and hiPSCs concentration in PF on day -3, microspheroid size and shape which were measured on day -2, and CHIR concentration and using to different differentiation protocol on day 0. The output feature was CM content in each batch of encapsulation which was measured by flow cytometry data. The only categorical input feature associated with differentiation media and was encoded into numerical variables (Media 1, Media 2) using one-hot encoding. Features like surface and volume of the micrpspheroids were derived from the input features to have indicators of their geometry of them. As the first potential set of inputs were ready subsets of them were selected by different feature selection methods: Correlations, Principal Component Analysis (PCA), Wrapper methods, amd Embedded methods. As the final step, using all the subsets selected by the methods classification models were trained and the performances of them were compared. The machine learning techniques used for the modeling were Random Forest, Gaussian Process, and Support Vector Machines.

4.1.1 Computational Methods and Theory

4.1.1.1 Feature Engineering

The data used for computational studies were obtained from 85 batches of experiments by our collaborators, Dr. Lipke's group. Each of these batches was considered one data point for constructing the models. The first potential set of input features were the tunable variables that described the experimental conditions of the differentiation process. This set included the number of cells being encapsulated, the post-freeze passage number, type of media, CHIR concentration, cell concentration, fibrinogen concentration in PF, microspheroid size, and the axial ratio of the microspheroid. Based on experimental observations, nine additional derived features were added to the primary set to capture the impact of the microspheroid geometry on differentiation. These additional features include surface (S) and volume (V) of the microspheroids, the ratio of microspheroid surface to volume (S/V), ratio of CHIR molecule concentration to microsphere surface, volume and their ratio, and inverse of these ratios. The feature set included one categorical variable, media type, which was represented using one-hot encoding (Potdar et al., 2017) in the models. In one-hot encoding, the categorical feature with m different observations is transformed

into m binary variables indicating each observation (Potdar et al., 2017). Using one-hot encoding, the variables Media 1 and Media 2, which are binary variables, substituted the media type variable in the feature set. Consequently, the initial input feature set for computational studies included 16 attributes.

4.1.1.2 Feature Selection Methods

Different feature selection methods (Guyon and Elisseeff, 2003) were utilized to identify the input features with a significant impact on the classification of the CM content (Figure 16). It has been shown that the accuracy of classification models depends on the input features set (Chen et al., 2020; Chen and Wasikowski, 2008). Furthermore, the limited number of data makes feature selection essential to training an accurate model (Remeseiro and Bolon-Canedo, 2019; Wang et al., 2016). The generalization of the performance of the trained models is poorer when the features are irrelevant or redundant (Salcedo-Sanz et al., 2014). Classification models were trained using the selected features by each of the methods. By comparing the performance of the classification models, the features with the strongest influence on the classification of the CM content on day 10 were determined. The feature selection methods considered included filter (Hall and Smith, 1999), embedded (Jović et al., 2015), wrapper methods (Kohavi and John, 1997), and principal component analysis (PCA) (Hotelling, 1933) (Figure 16). The description of the feature selection methods is included in Section 2.2.1 of Chapter 2.

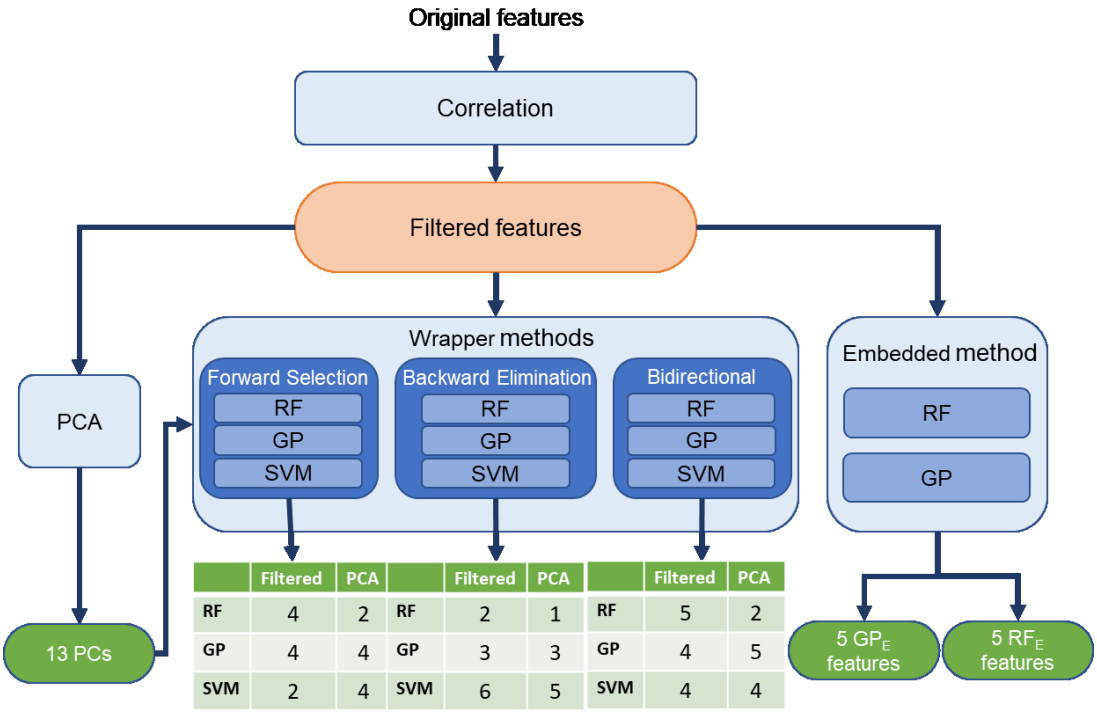


Figure 16. Diagram of feature selection methods used for selection of most significant features. The first step was using correlation to filter the initial set of features. The Principal Component Analysis (PCA), Wrapper Methods, and Embedded methods were implemented. The modeling techniques used were Random Forest (RF), Gaussian Process (GP), and Support Vector Machines (SVM).

4.1.1.3 Evaluation Metrics

The classification model performance was evaluated and compared using four metrics: accuracy, recall, precision, and the Mathews correlation coefficient (MCC). The accuracy measures the proportion of correct model classifications. The precision is the probability, given the model predicts the *"sufficient"* class (CM content greater than or equal to 65%), that the differentiation actually resulted or will result in a CM content equal to or above 65% on day 10. The recall is the ratio of all the correct *"sufficient"* class predictions to all of the *"sufficient"* class observations in the data set or the percentage of the *"sufficient"* class differentiations that the model captures successfully. Accuracy, precision, and recall have values between zero and one, with one indicating a perfect score for these metrics (Sokolova and Lapalme, 2009). The MCC is

a correlation coefficient between the actual classifications of the differentiation experiments and the classification predicted by the model (Matthews, 1975). The MCC ranges from -1 to 1. A value of zero for the MCC indicates no correlation between the data classifications and the model predictions, meaning that the prediction with the model is no better than randomly assigning classifications. A value of -1 indicates a perfect negative correlation, with all of the predicted classes being the opposite of the actual classes, and a value of one indicates a perfect positive correlation with all predicted classes being correct.

4.1.2 Feature Selection Results

The initial feature set included the number of cells being encapsulated, the post-freeze passage number, type of media, CHIR concentration, cell concentration, fibrinogen concentration in PF, microspheroid size, and the axial ratio of the microspheroid. This feature set was developed by our collaborators based on previous studies on 3D hiPSC-CM differentiation (Chang et al., 2020; Finklea et al., 2021; Halloin et al., 2019; Zhao et al., 2019) like CHIR concentration, and the experimental observations during hiPSC encapsulation and differentiation. For instance, the initial stem cell density for encapsulation was observed to impact CM differentiation outcome. The other initial features, such as S/V/CHIR, CHIR/S, CHIR/S/V, and S/CHIR, were selected based on the experimental observation of our collaborators during differentiation from more than 100 batches of hiPSC-microspheroid cardiac differentiations. The features from the initial set were first filtered using the Pearson (linear) correlation. For sets of features with a correlation coefficient higher than 0.8, only one feature was kept in the set, and the rest were eliminated (Figure 17). The resulting feature set was named filtered feature set and used as the base feature group for the remaining feature selection tasks. The features in the filtered feature set are the number of cells, the post-freeze passage number, type of media, CHIR concentration, cell concentration, fibrinogen

concentration in PF, microspheroid size, axial ratio, S/V/CHIR, CHIR/S, CHIR/S/V, and S/CHIR. The classification models, including RF, GP, and SVM, were built using the filtered feature set, and the performance metrics were evaluated for each of the three classifiers.

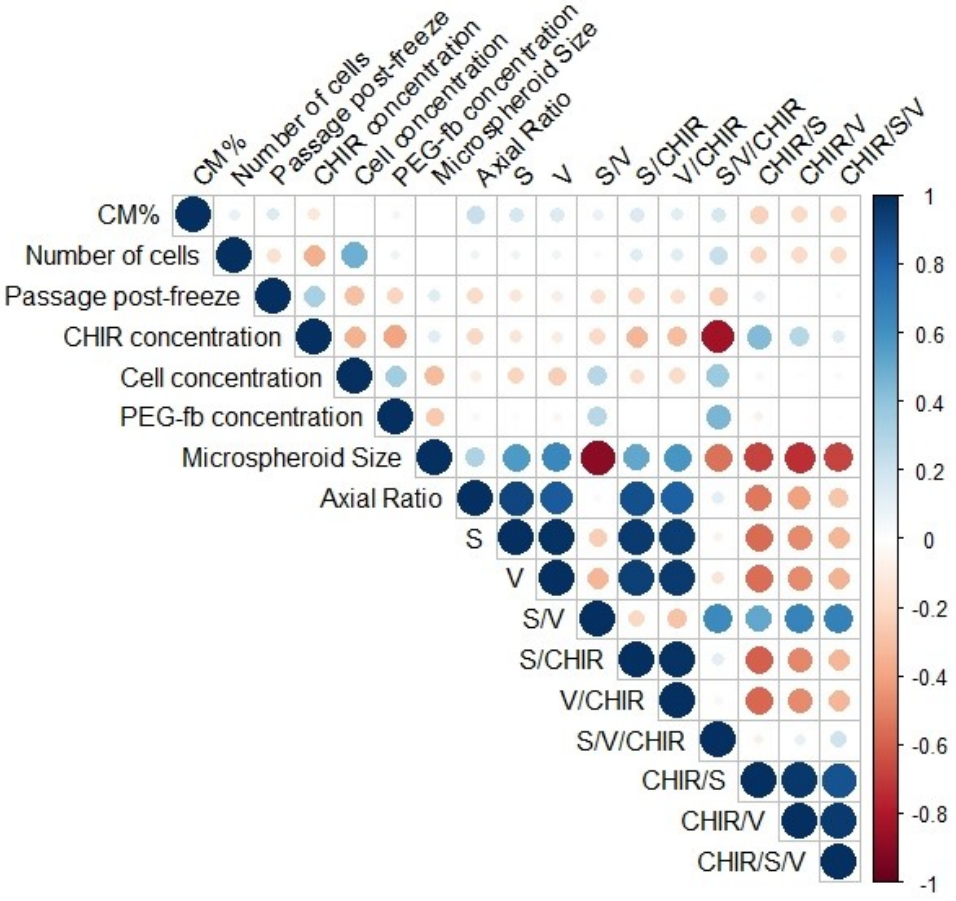


Figure 17. Heatmap of the Pearson correlation values between the input features and output variable. The larger and darker the circles, the higher the absolute value of the correlation value. The colors blue and red correspond to positive and negative values, respectively.

PCA was used to find the orthogonal principal components (PCs) explaining the variance in input data. Using the filtered feature set, 13 Principal Components (PCs) were constructed. The details about 13 PCs with their ratio of input variance description are included in Appendix 2. The first six PCs of the 13 explained above 90% of the filtered feature set variance. The first six PCs were utilized to train RF_{PCA}, GP_{PCA}, and SVM_{PCA} models for classifying CM content.

Table 3. Relative feature importance from the random forest (RF) and Gaussian process (GP) models embedded functions for Filtered features.

RF		GP	
Features	**Importance**	**Features**	**Importance**
S/CHIR	0.13	Cell concentration	2.00×10^{3}
S/V/CHIR	0.13	Media1	9.30×10^{2}
CHIR/S	0.12	CHIR/S/V	8.51×10^{2}
Microspheroid Size	0.11	Axial Ratio	3.03×10^{0}
Number of cells	0.10	Media2	3.21×10^{-1}
Passage post-freeze	0.08	Passage post-freeze	1.43×10^{-9}
Axial Ratio	0.07	S/CHIR	1.21×10^{-9}
Fibrinogen concentration in PF	0.07	CHIR concentration	1.05×10^{-9}
CHIR concentration	0.06	Number of cells	7.86×10^{-10}
Media 2	0.05	Microspheroid Size	7.68×10^{-10}
Media 1	0.05	S/V/CHIR	5.36×10^{-10}
Cell concentration	0.03	Fibrinogen concentration in PF	5.34×10^{-10}
CHIR/S/V	0.00	CHIR/S	3.04×10^{-10}

Table 4. Relative feature importance from the random forest (RF) and Gaussian process (GP) models embedded functions for PCs.

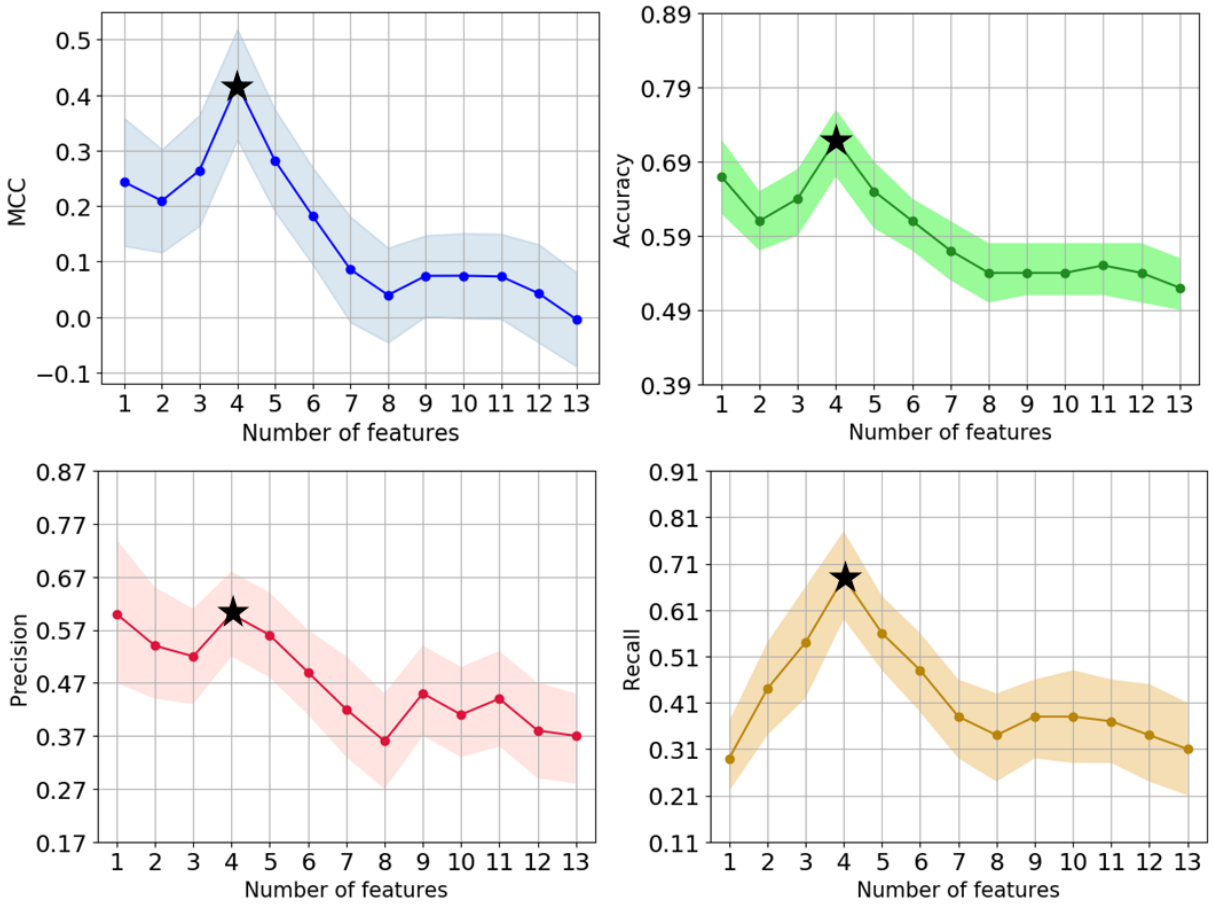
RF		GP	
Features	**Importance**	**Features**	**Importance**
PC1	0.16	PC4	1.80×10^{4}
PC8	0.11	PC3	1.67×10^{4}
PC6	0.09	PC7	1.30×10^{4}
PC4	0.08	PC12	9.81×10^{2}
PC7	0.08	PC13	1.31×10^{-1}
PC2	0.08	PC2	3.54×10^{-9}
PC12	0.08	PC1	2.62×10^{-9}
PC5	0.08	PC8	3.40×10^{-9}
PC9	0.06	PC10	1.97×10^{-9}
PC10	0.06	PC11	1.69×10^{-9}
PC11	0.06	PC6	1.62×10^{-9}
PC3	0.06	PC9	1.25×10^{-9}
PC13	0.00	PC5	7.97×10^{-10}

Built-in feature selection functions of RF and GP were also investigated for feature selection. The RF and GP classifiers were constructed using features selected by their embedded functions. Each embedded function calculates relative feature importance for building the corresponding classifier. The calculated relative importance of features for each classifier model is given in Table 3 and Table 4 for the filtered feature set and PCs, respectively. The first two columns in Table 3 and Table 4 of these two tables correspond to RF classifier and the last two to the GP classifier. The features are sorted based on their importance for each classification model.

Based on Table 3, the first five features in the GP method had the highest importance in comparison to the others, with higher than six orders of magnitude difference in their importance value. Thus, cell concentration, media type (Media 1 and Media 2), CHIR/S/V, and axial ratio were used to build a GP classifier. The feature importances for PCs in Table 4 for the GP method, given in the first five rows of the table, PC4, PC3, PC7, PC12, and PC13, are significantly larger than the rest of the PCs. Unlike GP, the feature importance calculated using the RF model did have a significant change in the values to choose the most important ones for features or PCs. As a result, to choose the features, first, the RF model using the most important feature was constructed. Then, based on feature importance order, one feature at a time was added to the input set, and a new model was constructed up to the point that all the features were included in the feature set. The feature set with the highest values of performance metrics was chosen as the features selected by RF. The first four features from the RF column in Table 3 were the most important features identified by RF-based feature selection. The RF classier was constructed using these four features (S/CHIR, S/V/CHIR, CHIR/V, and microspheroid size). No common features were selected by GP and RF models, suggesting that none of the features were informative to be selected regardless

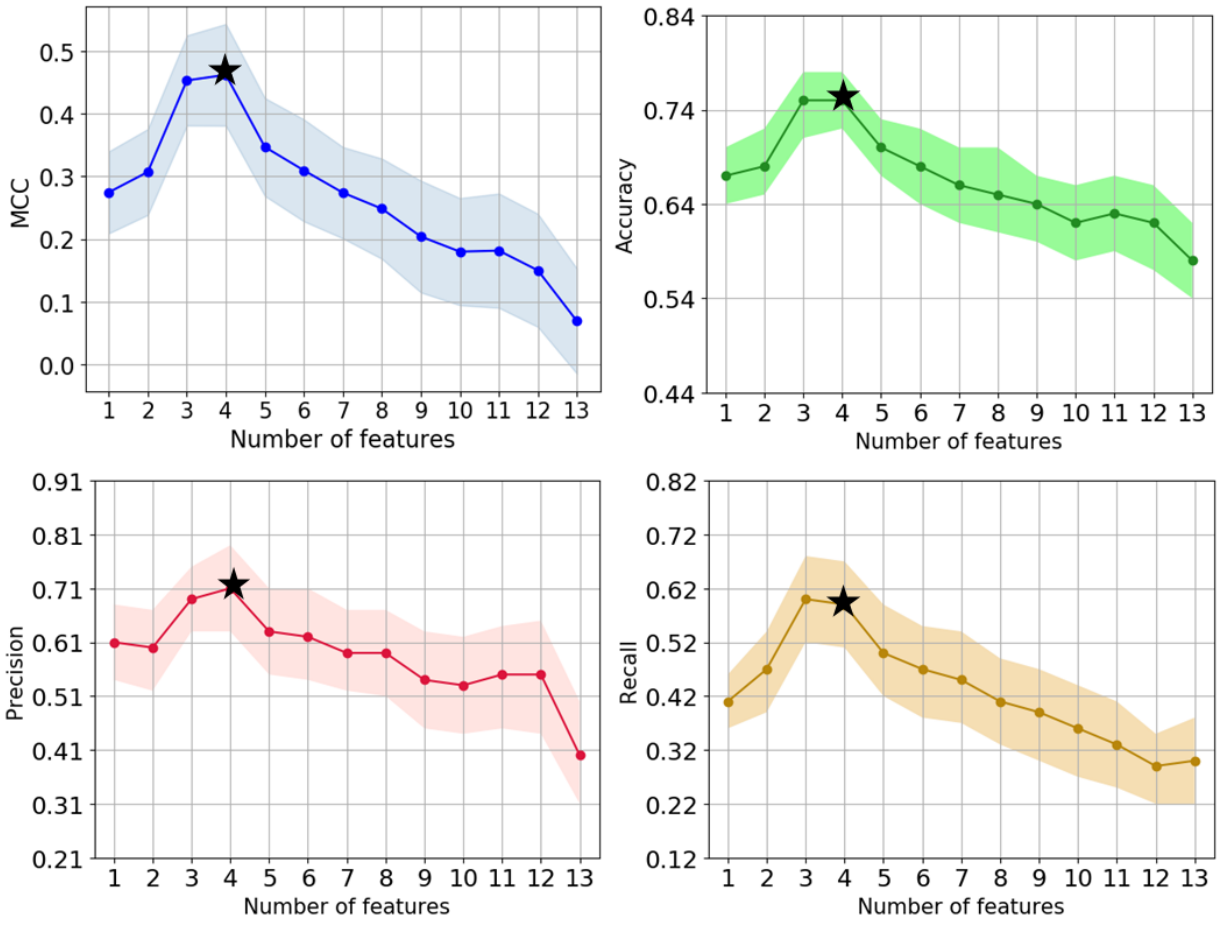
of the modeling technique. The first three PCs in Table 2 (PC1, PC6, and PC8) were selected using the embedded feature selection method of RF and used for building the RF classifier.

The last feature selection technique was wrapper methods. All the three classification models, RF, GP, and SVM, were trained with FS, BE, and BD methods for finding the best feature combination exclusive to each model. The wrapper methods were implemented for both the filtered feature set and PCs. Figure 18 and Figure 19 contain plots of how the MCC, accuracy, precision, and recall values for the GP classifier changed with the number of features selected by the FS method from the filtered feature set and PCA, respectively. The plots illustrate the performance metric values for constructed models with different combinations of the features. The lines in Figures 18 and 19 are used to track the changes in metric values shown with markers, and the shaded area demonstrates the 95% confidence interval calculated based on 30 iterations of Monte Carlo cross-validation. The tables below the plots list the features in the order they were selected. According to Figures 18 and 19, the values of all four metrics show increasing trends as the number of selected features increases to four features. After four features, the values of MCC, accuracy, precision, and recall exhibit a descending trend. The number of features corresponding to the highest comparison metrics is selected as the optimal feature set in each case. The rest of the figures associated with the performance of three different wrapper methods with other modeling techniques are included in Appendix 2.



	Feature		Feature
1	PEG-fb concentration	8	Media 1
2	S/V/CHIR	9	CHIR/S
3	CHIR concentration	10	Media 2
4	Passage post-freeze	11	Axial Ratio
5	S/CHIR	12	Cell concentration
6	Microspheroid Size	13	Number of cells
7	CHIR/S/V		

Figure 18. Matthew's correlation coefficient (MCC), and associated Accuracy, Precision, and Recall plots to it for the forward selection (FS) algorithm with Gaussian Process (GP) classifier on filtered features. The table shows the order each of the features was added until all the features were selected. The black stars show the best case with the highest metric value.



1	2	3	4	5	6	7	8	9	10	11	12	13
PC8	PC5	PC4	PC13	PC11	PC2	PC12	PC10	PC7	PC6	PC9	PC3	PC1

Figure 19. Matthew's correlation coefficient (MCC), and associated Accuracy, Precision, and Recall plots to it for forward selection (FS) algorithm with Gaussian Process (GP) classifier on Principal Components (PCs). The table shows the order each of the features was added until all the features were selected.

The results for selected features using each method and modeling technique are included in Table 5 and Table 6 for filtered features and PCs, respectively. In Tables 5 and 6, the first column is the list of the features/PCs used in the study and the evaluation metrics. The rest of the columns are associated with selected features and results of the modeling techniques using embedded methods and three different wrapper methods, which had the highest evaluation metrics values. The features selected in each case are marked, and the accuracy, recall, precision, and MCC

associated with models built using the selected features are calculated using Monte Carlo cross-validation with 30 iterations.

Table 5. Features selected from the filtered feature set using different wrapper methods and embedded functions.

	RF_{FS}	RF_{BE}	RF_{BD}	SVM_{FS}	SVM_{BE}	SVM_{BD}	GP_{FS}	GP_{BE}	GP_{BD}	RF_{E}	GP_{E}
Number of cells					✓						
Passage post-freeze				✓	✓	✓	✓	✓	✓		
CHIR concentration					✓	✓	✓				
Cell concentration			✓								✓
Fibrinogen concentration in PF		✓	✓	✓	✓	✓	✓				
Microspheroid Size					✓					✓	
Axial Ratio											✓
CHIR/S										✓	
CHIR/S/V	✓							✓	✓		
S/CHIR	✓	✓								✓	
S/V/CHIR						✓	✓			✓	
Media1	✓		✓		✓			✓	✓		✓
Media2	✓										✓
Accuracy	0.67	0.66	0.69	0.69	0.66	0.71	**0.72**	0.63	0.63	0.65	0.52
Recall	0.55	0.55	0.65	0.33	0.59	0.61	**0.68**	0.55	0.55	0.52	0.48
Precision	0.59	0.56	0.57	0.68	0.58	0.64	**0.6**	0.57	0.57	0.55	0.40
MCC	0.31	0.3	0.36	0.3	0.32	0.41	**0.42**	0.28	0.28	0.26	0.02

According to Tables 5 and 6, several features were frequently selected by the feature selection methods. These features include post-freeze passage number, fibrinogen concentration in PF, media type, CHIR concentration, and the ratio of CHIR concentration to the surface/volume.

Based on Table 6, PC8 and PC13, which are strongly correlated with post-freeze passage number and media type, respectively, were the most frequently chosen among all the PCs. Based on previous studies hiPSCs effectiveness highly relies on their phenotype stability during long-term passaging (Kempf et al., 2016b; Volpato and Webber, 2020), and the passage number of iPSC impacts differentiation outcomes. Furthermore, other important parameters such as CHIR concentration (Kempf et al., 2016b) and media type (Lian et al., 2013) affected differentiation yield in the 2D monolayer differentiation system. Interestingly, for the 3D hiPSC-CM differentiation used in this study, these parameters (features) have been selected by wrapper methods. Besides these parameters, the PF fibrinogen concentration has been selected with different methods. Fibrinogen, as a well-defined extracellular matrix (ECM) molecule, has been used to provide a biological site in PF for cell adhesion (Fuoco et al., 2012). The PF percent composition impacts the resulting PF hydrogel stiffness (Almany et al., 2003); our collaborators hypothesize that PF matrix stiffness plays an important role in hiPSC-CM differentiation outcome. Based on our collaborators' experimental observations, they also hypothesize that microspheroid shape or CHIR concentration to surface area and CHIR concentration to microspheroid volume ratios may affect differentiation.

Based on Tables 5 and 6, GP models trained with features selected using the FS method (GP_{FS}) had the highest accuracy and MCC for both the filtered feature set and PCs, and the values for recall and precision were higher than most of the classifiers. Post-freeze passage number, fibrinogen concentration in PF, CHIR concentration, and S/V/CHIR were the selected features by the GP_{FS} method from the filtered feature set. Among all PCs, GP_{FS} chose PC8, PC4, PC5, and PC13. These PCs are associated with post-freeze passage number, media type, Fibrinogen concentration in PF, CHIRS/S/V, axial ratio, and cell density. In both cases, most of the selected

features overlapped with the most frequently selected ones, except for axial ratio and cell density. According to the literature, cell density and confluency influence the CM content in 2D environments (Balafkan et al., 2020). The axial ratio, defined as a ratio between the microspheroid longitudinal and transverse axes, accounts for the shape of the microspheroids.

Table 6. Features selected from PCs using different wrapper methods and embedded functions.

	RF_{FS}	RF_{BE}	RF_{BD}	SVM_{FS}	SVM_{BE}	SVM_{BD}	GP_{FS}	GP_{BE}	GP_{BD}	RF_{E}	GP_{E}
PC1											✓
PC2				✓		✓					
PC3					✓					✓	
PC4							✓		✓	✓	
PC5							✓	✓	✓		
PC6											✓
PC7				✓	✓	✓				✓	
PC8	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
PC9				✓		✓					
PC10											
PC11											
PC12					✓					✓	
PC13	✓				✓		✓	✓	✓	✓	
Accuracy	0.67	0.67	0.67	0.73	0.7	0.73	**0.75**	0.69	0.69	0.51	0.66
Recall	0.57	0.57	0.57	0.64	0.5	0.64	**0.59**	0.44	0.44	0.36	0.52
Precision	0.59	0.59	0.59	0.66	0.65	0.66	**0.71**	0.62	0.62	0.34	0.55
MCC	0.31	0.31	0.31	0.44	0.35	0.44	**0.46**	0.31	0.31	-0.03	0.27

4.1.3 Classification Results

Classification models were constructed using the filtered feature set and PCs. Table 7 summarizes the accuracy, recall, precision, and MCC values of all trained classification models. The metrics were calculated using 30 iterations of Monte Carlo cross-validation. The first column of Table 7 shows the machine learning techniques used to construct the classifiers, and the next four columns demonstrate the performance metrics for the corresponding classifier models.

The first three rows of Table 7 show the results for three modeling techniques using the filtered feature set as their inputs. The accuracy and precision of the three methods are very close, and the values are around 60% and 50% on average, respectively. However, the GP model has a lower recall value (27%) and consequently a lower MCC value compared to the other two methods. The RF classifiers performed better with all metrics higher than the other two.

Rows four through six are the results of classification performance for three models of GP, RF, and SVM using PCs as their input features. All methods yielded models with accuracies close to 60%. The recall and precision were around 30% and 45% on average, respectively. The highest MCC using PCs as inputs of the three models was 0.16 with the SVM_{PCA} model, and the lowest was 0.06 for RF_{PCA}.

The seventh and eighth rows of Table 7 show the classification results with features selected using embedded methods for RF and GP models. The average accuracy and recall for these two cases were 60% and 47%. The MCC value for RF_{EB}, 0.26, was two folds of the GP_{EB} method.

The last two rows contain the results for GP classifier with features selected from filtered set and PCs, respectively, using wrapper methods. Based on Table 5 and Table 6, GP models with features chosen using the FS method for the filtered feature set and PCs had the highest values of performance metrics. Therefore, GP-FS and GP-FS_{PCA} were selected as the representative models to discuss the results of the features selected using wrapper methods. The accuracy was above 70% for both GP-FS and GP-FS_{PCA} methods marking the maximum value achieved in comparison to all other methods. The other three metrics were also the highest, with average values of 65%, 56%, and 0.44 for recall, precision, and MCC, respectively.

4.1.3.1 Classification Models Trained Using Principal Components

Six PCs, which accounted for 90% of the variance in the filtered feature set, were used as input features for training RF, GP, and SVM classification models. Rows corresponding to GP_{PCA}, RF_{PCA}, and SVM $_{PCA}$ in Table 7 summarize the performance metrics for these models. The results reveal that the performance of the RF classifiers deteriorates as PCs are used as the inputs and as can be seen in the decreasing accuracy, recall, precision, and MCC values by 8%, 19%, 7%, and 0.14, respectively. However, the performance of the GP and SVM models did not change significantly, as the changes in all metric values were less than 6%. The variance in the output was not accurately captured by the PCs accounting for 90% of the variance in input, suggesting the possibility of features not captured in the current set also impacting the differentiation process.

Table 7. Classification models' performance using all feature selection methods.

Model	Accuracy	Recall	Precision	MCC
GP	0.59	0.40	0.51	0.14
RF	0.62	0.48	0.52	0.20
SVM	0.58	0.45	0.45	0.12
GP $_{PCA}$	0.57	0.37	0.48	0.12
RF $_{PCA}$	0.56	0.29	0.45	0.06
SVM $_{PCA}$	0.60	0.40	0.52	0.16
RF_E	0.65	0.49	0.57	0.26
GP_E	0.55	0.46	0.31	0.12
GP-FS	0.72	0.68	0.60	0.42
GP_FS $_{PCA}$	0.75	0.59	0.71	0.46

4.1.3.2 Classification Models Trained Using Features Selected by Embedded Methods

GP and RF models come with built-in feature selection functions. We used these functions to select features and train classifier models using the selected features. The first four and five most important features identified with built-in functions (Tables 3 and 4) were used for training RF and GP models, respectively. The seventh and eighth rows of Table 7 show the accuracy, recall, precision, and MCC values of the RF and GP classifiers trained using the features selected by their corresponding built-in selection functions. The results show a small improvement in RF performance metrics when the classifier is trained using the five most important features compared to the classifier trained using the filtered feature set with 1% to 6% increases in the evaluation metrics. However, the metrics for the GP model decreased, and the maximum decrease was in precision from 51% to 31%, suggesting that the eliminated features helped predict the classes, although they had lower importance.

4.1.3.4 Classification Models Trained Using Features Selected by Wrapper Methods

Three wrapper methods, FS, BE, and BD Search, were used to select features and train classifiers using RF, GP, and SVM methods for the filtered feature set and PCs. Classifiers trained using GP using the FS method had the highest accuracy and MCC for both inputs (Figures 18 and 19). Figures 18 and 19 show the trends of change in values of four metrics, MCC, accuracy, precision, and recall, for trained GP models, as one feature is selected at a time from the filtered feature set and PCs, respectively, and added to the training input set. The cases identified as the best-performing model based on their highest metric values were marked with a star in Figures 18 and 19.

The GP model with FS selected four features, fibrinogen concentration in PF, S/V/CHIR, CHIR concentration, and post-freeze passage number from the filtered feature set as the best input

set. The GP classifier trained using the selected four features had an accuracy of 72%, recall of 68%, precision of 60%, and MCC of 0.42. All metric values were significantly larger than all the previously trained models. The GP classifier with PCs employed four components - PC8, PC5, PC4, and PC13 (Figure 19) - as inputs. The largest absolute values of the coefficients of different features for the selected PCs were associated with post-freeze passage number, media type, fibrinogen concentration in PF, CHIR/S/V, axial ratio, and cell concentration. The accuracy of this model was 75%, with a recall of 59%, precision of 71%, and MCC of 0.48. Best classifiers trained using both filtered features and PCs included post-freeze passage number, fibrinogen concentration in PF, and the ratio between the CHIR concentration and microspheroid surface to volume ratio as the predictors of the output. Experiments have shown that the post-freeze passage number and CHIR concentration impact hiPSC differentiation to CM (Kempf et al., 2016b; Liu et al., 2014; Volpato and Webber, 2020). Our experimental collaborators also hypothesize that the PF fibrinogen concentration plays an important role in the outcome of hiPSC-CM differentiation outcome. PF composition impacts hydrogel stiffness (Almany et al., 2003); it has been suggested that physical parameters such as mechanical stiffness may influence hiPSC capacity for self-renewal, stem cell fate, and differentiation to CMs (Kerscher et al., 2016). Dynamic substrate stiffening, mimicking the changes that heart tissue undergoes during development, has been previously shown to drive the maturation of precardiac cells and CMs (Teng and Engler, 2019; Young and Engler, 2011). Furthermore, substrate mechanics have been shown to impact mesoderm induction when cells are cultured on 2D surfaces (Park et al., 2011). When encapsulating and differentiating hiPSCs within a 3D microenvironment, having an initial PF stiffness that more closely mimics the native embryonic microenvironment may be a critical factor in the outcome of hiPSC-CM differentiation (Kerscher et al., 2016).

The results from GP-FS_{PCA} showed the highest metric values, except for recall which was the second-highest, compared to all the methods considered in this study. GP-FS_{PCA} model had a lower recall than the GP-FS method, indicating that the latter classifier correctly predicted a higher number of positive class points. On the other hand, the precision value was higher for GP-FS_{PCA} than GP-FS, suggesting that the probability of a predicted positive outcome to be a real positive class was higher with GP-FS_{PCA}. The four PCs selected by GP-FS_{PCA}, PC8, PC5, PC4, and PC13, explained only 18% of the variance in the input data. The classification results suggest that the variance in the output may not be strongly related to the variance in the input data for the experimental data used in the study since the weakest descriptors of the input data variance were selected as the most influential predictors for the classifier model.

In summary, the best classification results were generated by the features selected by the FS using GP modeling, where the PCs were inputs to the model. These results indicate the importance of correctly selecting the predictors, i.e., features, as the right combination of the predictors led to classifier models with higher evaluation metric values. The best classifier model had MCC of 0.46, suggesting that it can predict classes of the new points accurately 46% of the time better than a random classifier, accuracy of 0.75, which means it can classify any point correctly 75% of time, recall and precision of 0.59 and 0.71, respectively, suggesting there is 59% chance that a point with sufficient class would be classified correctly and 71% probability the points predicted as sufficient are truly from that class. Additionally, the variance in the output was revealed not to be strongly correlated to the variance of the input data based on the results from classification with PCs. Based on this outcome, the results from classification performances with different feature selection methods, and results from the previous work in CM content prediction in a bioreactor platform (Williams et al., 2020), we hypothesize that other biological features,

specifically the ones associated with the differentiation process during the protocol timeline (dissolved oxygen in media, PH, cell concentration, transcriptomics/gene expression, etc.), may have additional key information necessary to predict the output more accurately and describe the variance existing in the CM content on day 10 of differentiation.

4.1.4 Conclusions

In this chapter, the goal was to classify CM content on day 10 of differentiation and to identify the most significant features and conditions which lead to higher CM content in one cell-handling step differentiation of hiPSCs to CMs using encapsulated stem cells in PF hydrogel. Machine learning techniques were used to engineer and select features and build classification models to classify CM content on the 10^{th} day of differentiation into two classes, "*sufficient*" and "*insufficient"*. The initial feature set was constructed by combining experimentally measured variables and engineered features for 85 differentiation experiments. Filter, embedded, and wrapper methods, and principal component analysis were employed to find the best combination/subset of the filtered features for accurately classifying the CM content. The classifier models were constructed using random forests, Gaussian process classifier, and support vector machines. Accuracy, recall, precision, and Mathew's correlation coefficient (MCC) were the metrics for comparing the performance of the classifiers.

The highest metric values of 75%, 59%, 71%, and 0.46 for accuracy, recall, precision, and MCC were achieved using principal components (PCs) selected by wrapper methods to build a Gaussian process classifier. The PCs used in the best-performing model explained 18% of the input data variance, suggesting that the variance of the input set did not fully explain the variance in the output data. Based on previous work in using machine learning techniques to predict CM content in a bioreactor platform in our group (Williams et al., 2020), we suggest including more biological

features and indicators corresponding to cell growth and differentiation, such as dissolved oxygen, pH, and cell density during the time course of differentiation, in experimental data can improve the prediction accuracy.

This study provides helpful information on the important controllable parameters affecting the differentiation process by identifying variables that are strong predictors of the CM content on the 10^{th} day of differentiation. This information is valuable in two main ways. First, the selected features are computationally important for accurately predicting CM content. Second, they provide potential leads for further exploration of this system experimentally to explain the observed outcomes. Additionally, the classification models built in this study were the first steps toward predicting CM content on the desired day of differentiation in hiPSC-CM differentiation in hydrogel microspheres.

4.2 Prediction of the Size for PLGA-based Nano-particles

Polymeric nanoparticles (e.g., those based on aliphatic polymers such as poly-lactic-co-glycolic acid, PLGA) have drawn significant attention for their potential use in biomedical applications (Fredenberg et al., 2011; Makadia and Siegel, 2011). This interest is due to their favorable properties, such as biodegradability, biocompatibility, and tunable degradation rate, for the controlled release of encapsulated therapeutics. Despite the tremendous effort to develop these particles, clinical translation has been slow, with no polymeric nanoparticles currently approved by the U.S. Food & Drug Administration for clinical application (Anselmo and Mitragotri, 2019; Operti et al., 2021). It has been suggested that a major obstacle to overcome is the development of processes for the consistent and controlled production of particles with well-defined properties (Wang et al., 2021). Control of nanoparticle properties (e.g., particle size distribution) is important

for obtaining formulations with predictable performance and safety profiles – keys to approval (Berkland et al., 2004; Elsabahy and Wooley, 2012; Sahin et al., 2017).

Various methods for synthesizing polymeric nanoparticles have been explored to control average particle size and the polydispersity index (PDI) – a measure of size distribution. These include emulsion-based methods, nanoprecipitation, salting out methods, or dialysis (Crucho and Barros, 2017). The emulsion solvent evaporation (ESE) method is the most commonly studied of these methods. This method offers good control of synthesis parameters and high encapsulation efficiencies and yields highly uniform polymeric nanoparticles.

Several studies have evaluated the effect of process parameters on the average size and PDI of particles produced by the ESE method. In one study, Hernández-Giottonini et al. (Hernández-Giottonini et al., 2020) evaluated the impact of PLGA concentration, polyvinyl alcohol (PVA) concentration, organic solvent (volume) fraction, sonication amplitude, and the mixing speed used during the evaporation step. Their findings indicated that PLGA concentration within the range of 5–15 mg/mL (0.5–1.5% w/v) did not significantly change the nanoparticle size or PDI. This result contradicted other published findings by Song et al. (Song et al., 2008) that reported a positive correlation between size and PLGA concentration; it was suggested that variation in properties of PLGA used could be a contributing factor to this discrepancy. A positive correlation was also observed in a study reported by Halayqa et al. (Halayqa and Domańska, 2014), who saw an increase in average nanoparticle diameter as PLGA concentration increased from 0.8% to 1.3% or 1.6% for two different formulations. Both studies also observed that increasing PVA concentration was associated with a slight decrease in particle size and PDI. This effect, where the concentrations of both PLGA and PVA components have contrasting effects on final nanoparticle size, was hypothesized to occur because they change the interfacial surface tension and the bulk phase

viscosity of the emulsion droplets, which in turn, change the droplet coalescing stability (Shekar et al., 2011). Particle size was also observed to increase with a larger organic-phase volume fraction, which is expected, as total surface energy is a function of surface area. It should also be noted that PDI increased significantly as the organic volume fraction was increased to 0.500. Hernández-Giottonini et al. (Hernández-Giottonini et al., 2020) also observed that increasing the mixing speed yielded smaller particles, with no change to PDI, and increased sonication energy during emulsion formation also decreased size. An excellent summary of other attempts to control PLGA nanoparticle size is provided by Rao et al. (Rao and Geckeler, 2011). However, a rigorous assessment of the synthesis parameters and their potential use for controlling particle size and polydispersity is missing in these studies. Additionally, none of these studies provide predictive models and the importance of different synthesis parameters for accurately estimating the size and PDI of produced particles.

This work investigates the feasibility of employing machine learning techniques to determine which process parameters for the ESE method would enable the control of particle size and PDI. PLGA nanoparticles were synthesized via ESE using a wide range of parameter values. A subset of data from the synthesis experiments was used to train models to predict the particle size and PDI of PLGA nanoparticles with synthesis parameters as inputs. The remaining data set was kept as the testing set. We considered Gaussian process regression, random forest, support vector regression, multivariate adaptive regression splines, and artificial neural networks as candidate machine learning techniques for building the models. We evaluated the models using cross-validation root mean square error (RMSE) and mean absolute error (MAE) and selected the technique that yielded the lowest RMSE and MAE. Feature selection and model sensitivity was used to determine the most significant control parameters. Furthermore, an optimization problem

was constructed to predict synthesis parameters to yield a target average particle size. While explicitly focused on the synthesis of PLGA nanoparticles via ESE, we expect this work to provide a general framework that could be applied to other synthesis methods and various polymers.

4.2.1 Size and Size Distribution Width Characterization (PDI)

The experimental data were collected by our collaborators, Dr. Allan's group in the Department of Chemical Engineering at Auburn University. The average hydrodynamic size and size distribution (PDI) of nanoparticles were measured by Dynamic Light Scattering (DLS) using a Malvern Zetasizer ZS (Malvern, UK). Nanoparticle suspensions were prepared at concentrations of approximately 0.2 mg/mL in deionized water, and readings were taken at 25°C. A refractive index of 1.44 was used for PLGA. Results were the average and standard deviation of the three measurements for each data point. The width of the size distribution curve approximates three standard deviations (3σ) from the Z-average size (3σ) and can be correlated with the PDI by the relationship in Eq. 32 (Nobbmann, 2015).

$$\boldsymbol{PDI} = \left(\frac{\boldsymbol{\sigma}}{\boldsymbol{Z_{avg}}}\right)^{\mathbf{2}} \tag{32}$$

4.2.2 Computational Theory and Methods

Several computational models were utilized to evaluate the effect of process parameters on the average hydrodynamic size (S) and PDI of synthesized nanoparticles. Models used for data analysis included a power-law model and six different machine learning techniques – random forests (RFs), Gaussian process regression (GP), artificial neural networks (ANNs), extreme learning machines (ELMs), support vector regression (SVRs), and Multivariate adaptive regression splines (MARS). The accuracy of the models was evaluated using RMSE and mean absolute error (MAE). The model with the lowest RMSE and MAE was employed to formulate an

optimization problem for determining the synthesis parameter values for obtaining any specific desired PLGA nanoparticle size. The inputs for the models are specific energy (E_v) (J/mL), the moles of PVA in each described step of the process (flask (N_f^{PVA}), emulsion (N_e^{PVA}), wash (N_w^{PVA})), the total PVA moles used (N_t^{PVA}), PLGA moles (N^{PLGA}), and the ratio of moles of PVA to PLGA ($R_N = \frac{N_t^{PVA}}{N^{PLGA}}$).

Experimental data were divided into training (118 points), and test (20 points) sets. Data points in the test set were selected randomly from the 20 equal width bins across the range of sizes obtained experimentally. Monte-Carlo cross-validation with 30 replications and a 20/80 split was implemented for tuning model parameters during training.

To analyze and compare the uncertainty of the selected model to observed uncertainty from experiments, three new experimental conditions, i.e., synthesis parameter values, were identified, and PLGA nanoparticles were synthesized via ESE at these parameter values. The three new points were selected based on the ranges and trends observed in the existing data. The modeling studies revealed that size decreases exponentially with increasing energy. Based on this observation, the logarithm of the existing energy data range was divided into three equal segments, and one existing experimental condition was randomly selected from each segment for uncertainty analysis. At each condition, ten repetition experiments were conducted. The standard deviation of the replicate size measurements at each condition was compared to the standard deviation of the model predictions. The uncertainty estimates of the experiments were compared to the uncertainty evaluated based on the model with the lowest error metrics.

4.2.2.1 Performance Metrics Used for Model Evaluation

Root mean square error (RMSE) gives the mean value over the squared errors in the predictions (Eq. 33). When comparing models, the model with the lower RMSE value is preferred as its predictions, on average, are closer to the observations. The RMSE for n data points with $\boldsymbol{y}$ as the actual output value and $\hat{\boldsymbol{y}}$ as predicted value is calculated using Eq. 33.

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}}\left(\boldsymbol{y_j} - \hat{\boldsymbol{y}}_{\boldsymbol{j}}\right)^{\mathbf{2}}} \quad \text{Eq. (33)}$$

Mean absolute error (MAE) gives the average absolute error values of the predictions (Eq. 34). Similar to RMSE, lower values of MAE are preferred. MAE corresponds to the average distance of the predictions from the observed values. In contrast, RMSE puts higher weights on large error values (distances) because the squared absolute error values are used to calculate it. Thus, while MAE is a good measure of the average error of the predicted values, RMSE accounts for large prediction errors.

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{n}}\sum_{\boldsymbol{j=1}}^{\boldsymbol{n}}\left|\boldsymbol{y_j} - \hat{\boldsymbol{y}}_{\boldsymbol{j}}\right| \quad \text{Eq. (34)}$$

4.2.3 Size and PDI Regression Models Results

According to the initial screenings and plots of size (S) versus the input variables, the energy feature (E_v) showed the highest impact on the size values (Figure 20). The trend suggested a power-law relationship between energy and size. A power-law model fitted to data in Figure 20 (Eq. 35) was considered the baseline model for comparing the machine learning models. Interestingly, this power-law model suggests that the minimum size that can be achieved via sonication using specific energy (E_v) as the only controlling feature would be 149.15 nm.

However, in reality, this "minimum" size is specific to the process parameter ranges used and does not consider changes in the centrifugation collection process that were not analyzed in this study. For instance, centrifuge time and force could be increased, possibly leading to a smaller average diameter due to collecting any smaller particles that were formed. However, these features were deemed different from the emulsion formation process and thus were not considered in this work. Also, changing solvents or surfactants/co-surfactants could lead to smaller average sizes.

$$S = 1309.3\, E_v^{-0.575} + 149.15 \tag{35}$$

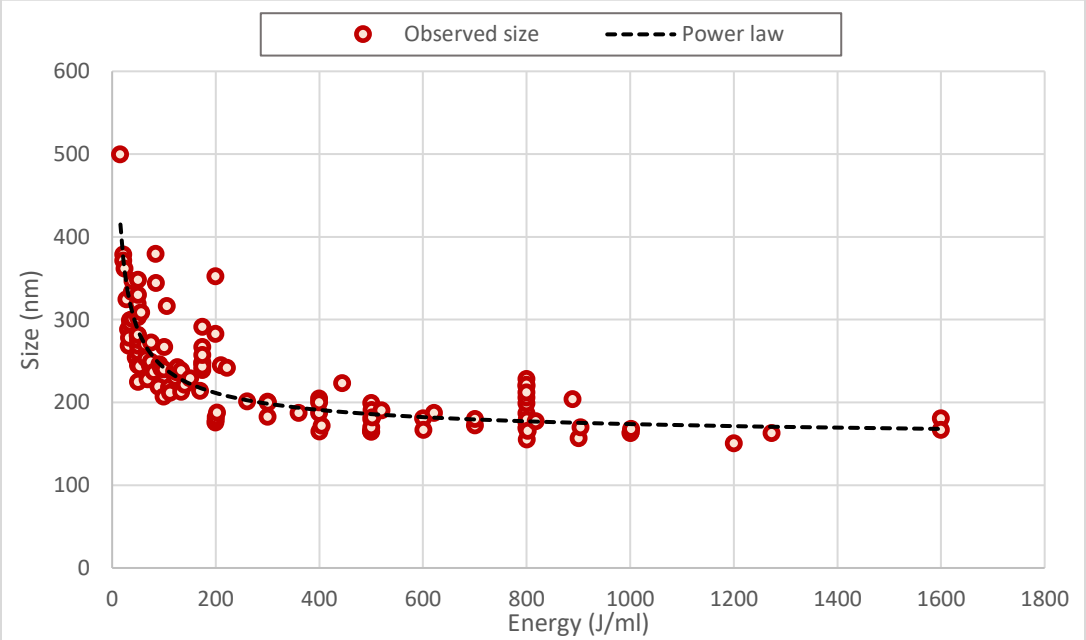


Figure 20. Observed average size and predicted size from the power-law model using energy as the independent variable

4.2.3.1 Performance of Trained Regression Models for Nanoparticle Average Size Prediction

The six machine learning models, including GP, ANN, ELM, MARS, SVR, and RF, were trained to predict the size of the nanoparticles. The comparison of the trained and power-law models' performances based on RMSE and MAE is given in Table 8. The random forest model is the best performing machine learning technique for the predicting size according to

training and test sets with the lowest RMSE and MAE values. The RF model predictions also have lower errors than the baseline model (Eq. 35). The other techniques generally overfit the data, yielding lower errors for the training set but higher for the test set.

Table 8. Performance of different models for predicting the average size of nanoparticles.

Model	RMSE-train	MAE-train	RMSE-test	MAE-test
Power-law	33.41	24.27	37.74	27.06
GP	34.30	23.39	38.17	27.64
ANN	21.85	12.71	32.32	24.45
ELM	28.18	19.34	34.49	24.49
MARS	20.17	12.18	30.78	19.28
SVR	27.72	18.51	34.88	24.08
RF	26.62	21.06	22.19	17.51

Figure 21 includes a plot of the experimentally observed sizes versus the predicted sizes calculated using the trained RF model for the test points. The uncertainty of the RF model predictions was evaluated for each test data point based on the standard deviation of the predictions from all the trees in the forest (L. Breiman, 2001). The RF model prediction uncertainty is represented with the vertical gray error bars in Figure 21. The experimental uncertainty was estimated for three points via repeated experiments, and it is represented with the orange horizontal error bars in Figure 21. The error bars in Figure 21 correspond to one standard deviation around the mean sizes for both experimental observations and model predictions. The standard deviation for sizes above 250 nm is high for both experimental measurements and model predictions. However, the standard deviations are remarkably small for sizes below 200 nm. We hypothesize that the higher error in size prediction around 250-350 nm in the data is due to the high uncertainty of the observed values in this range.

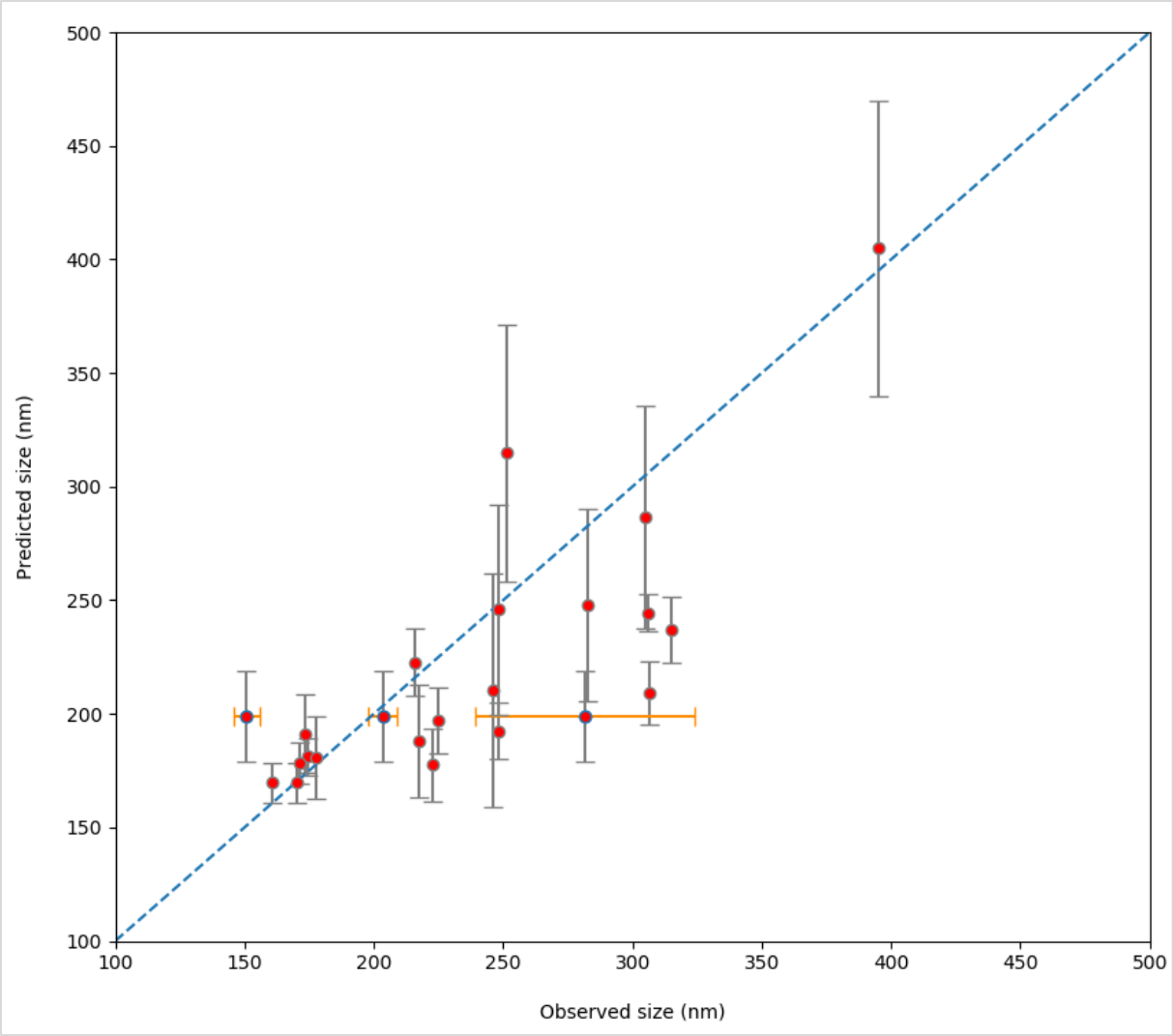


Figure 21. Predicted sizes with the RF model versus the observed size values for the test points. The horizontal and vertical error bars correspond to one standard deviation from the mean value for experimental and model size values, respectively.

4.2.3.2 PDI Model Performances

The first six rows of Table 9 list the performance of models trained to predict PDI. The last row of the table shows results for a linear model built to predict the PDI, which is discussed in the next paragraph. The performance comparison is based on the RMSE and MAE on the test set. The models trained using RF and GPR have the lowest MAE and RMSE, respectively, compared with other machine learning techniques.

Table 9. Performance of different models for predicting PDI.

Model	RMSE-train	MAE-train	RMSE-test	MAE-test
GP	0.038	0.026	0.041	0.033
ANN	0.033	0.021	0.056	0.043
ELM	0.041	0.029	0.056	0.043
MARS	0.036	0.023	0.051	0.036
SVR	0.041	0.025	0.047	0.033
RF	0.017	0.013	0.042	0.030
Linear	0.035	0.028	0.041	0.032

PDI is defined as the square of the ratio of standard deviation to the average size (Eq. 32). We investigated the relationship between the size and standard deviation for our experimental measurements. Figure 22 shows a linear relationship between size and standard deviation, especially for sizes greater than 200 nm. A model that utilizes the average standard deviation value ($\bar{\sigma}$) for sizes below 200 nm and linear regression between standard deviation (σ) and size above 200 nm was developed to predict standard deviation given size (Eq. 36). The R^2, RMSE, and MAE of this model for training and test data sets are summarized in Table 10.

$$\sigma = \begin{cases} \bar{\sigma} = 29.02, & Size < 200 \\ 0.73\ size - 106.67, & otherwise \end{cases} \tag{36}$$

We also developed a simple model (model labeled Linear in Table 9) that employs Eq. 36 to predict PDI. For sizes above 200 nm, the Linear model estimates standard deviation using Eq. 36 and calculates PDI using Eq. (32). For sizes below 200 nm, the average PDI value of those measurements is used as the prediction. We hypothesize that the PDI is roughly constant for average sizes below 200 nm due to reaching the effective limit on how energy is dispersed

throughout the emulsion. As energy is added to the emulsion, it is distributed to the emulsion droplets near the sonication tip first. This initial energy increase decreases the average diameter the most near the sonication tip, with the diameter increasing as one moves away spherically from the tip region. As more energy is added, the emulsion droplets near the tip reach the diameter plateau (leveling of diameter seen, for example, in Figure 21) first. The droplets further away reach this plateau later, and any further energy has very little change in any droplets' diameter and little change in PDI. In fact, in this regime, PDI should fluctuate based on deviations in droplet interfaces and coalescing effects. This would result in a linear decrease in PDI until a point, as seen in Figure 22.

The RMSE and MAE of these predictions are calculated and included in the last row of Table 9. The RMSE and MAE values are comparable and, in some cases, lower than those of ANN, ELM, MARS, and SVR models. The Linear model requires size as input to predict PDI. Therefore, utilization of this model would require using a model for predicting size first, which would increase the uncertainty of the PDI predictions of the Linear model. None of the other models in Table 9 require size as input and suffer from this additional prediction uncertainty.

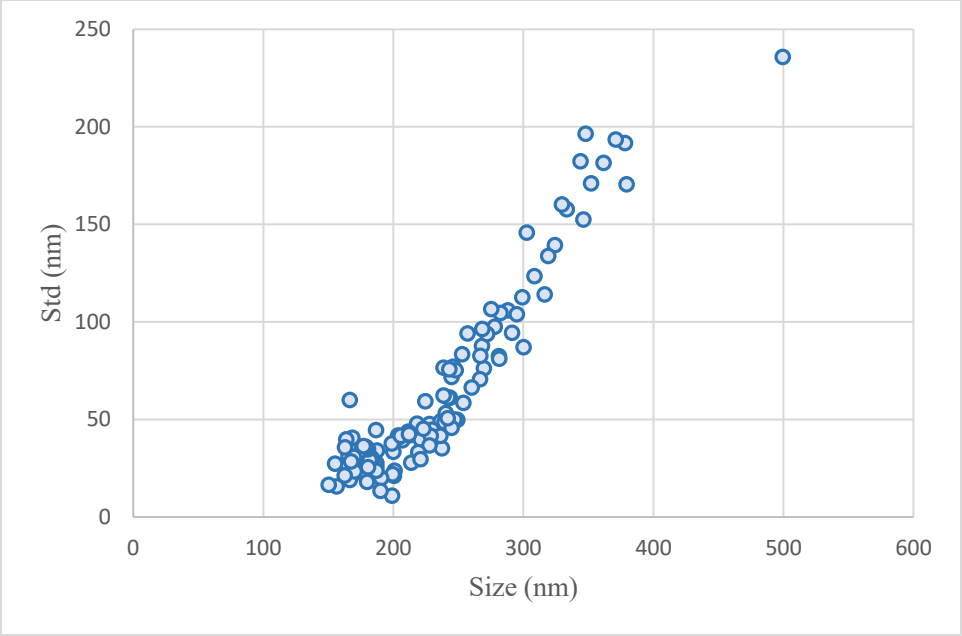


Figure 22. Standard deviation (Std) versus the observed average size of the nanoparticles.

Table 10. Performance of the linear regression model of standard deviation for sizes larger than 200 nm.

	Train	**Test**
R^2	0.88	0.71
RMSE	15.89	20.81
MAE	12.91	15.55

4.2.3.3 Feature Importance Calculated Using RF Model

In the RF models, each feature (i.e., input) may have a different impact on predicting the output. The RF modeling technique has an embedded function that quantifies these impacts via a metric called relative importance (L. Breiman, 2001). Table 11 shows the relative importance of each feature in predicting the size and PDI values for the trained RF models compared to other features. Energy (E_v) was specified as the most important feature contributing around 90% (importance of 0.897 in Table 11) for size prediction and 74%

(importance of 0.744 in Table 11) for PDI. The molar ratio of the PVA to PLGA was the second most important feature in both models.

Table 11. Feature importance for RF models predicting the size and PDI

Size		PDI	
Feature	**Importance**	**Feature**	**Importance**
$\boldsymbol{E_v}$	0.897	$\boldsymbol{E_v}$	0.744
$\boldsymbol{R_N}$	0.041	$\boldsymbol{R_N}$	0.130
$\boldsymbol{N_e^{PVA}}$	0.021	$\boldsymbol{N^{PLGA}}$	0.070
$\boldsymbol{N^{PLGA}}$	0.020	$\boldsymbol{N_e^{PVA}}$	0.025
$\boldsymbol{N_t^{PVA}}$	0.013	$\boldsymbol{N_t^{PVA}}$	0.014
$\boldsymbol{N_f^{PVA}}$	0.004	$\boldsymbol{N_f^{PVA}}$	0.009
$\boldsymbol{N_w^{PVA}}$	0.003	$\boldsymbol{N_w^{PVA}}$	0.008

The importance of the specific energy and molar ratio features is significant when considering the thermodynamic base of emulsion formation. The work, W, needed to reduce the size of the droplets can be represented by Eq. 37, where γ is the droplet interfacial tension, ΔA is the change in total surface area, and is a function of the number of droplets and their average size (radii) in Eq. 38.

$$W = \gamma \Delta A \tag{37}$$

$$\Delta A = n_{droplets,i}(4\pi r_i^2) - n_{droplets,f}\left(4\pi r_f^2\right) \tag{38}$$

The specific energy applies for direct work on the droplet to increase the number of droplets and decrease the average radius. However, this work does not correspond to the work term in Eq. 37 because there is some temperature change within the emulsion and the sonicator tip itself.

Furthermore, the energy reported by the sonicator is not the energy put directly into the emulsion; it is the energy used by the sonicator equipment to maintain desired tip vibrational amplitude.

The molar ratio affects the interfacial tension of the emulsion because as it increases, the PVA molecules available to coat the surface of each droplet also increase, which in turn decreases the free surface tension of each droplet. However, this observation also suggests that there is a limit to the effect of this ratio, as once each droplet surface is fully coated, any additional PVA molecule would not change free surface tension. Combined, the significance of both of these features means they can be used to control average diameter and PDI in other systems as well. Their significance on the final size and PDI should remain the same for other solvents, surfactants, or co-surfactants though the parameter values would differ from the ranges investigated in this work.

4.2.4 Optimization Results

4.2.4.1 Optimization Model to Estimate Synthesis Parameter Values for Achieving a Specified Size

Knowing the values for synthesis parameters leading to a desired average size is critical for efficient and reliable production of the nanoparticles. The values for these parameters can be estimated using accurate models predicting the average size of the produced nanoparticles. Hence, an optimization model is developed to determine the synthesis parameter values to obtain a given nanoparticle size. The optimization model in Eqs. 39 – 43 can utilize any regression model, $f(x)$, that can predict size given synthesis parameters, x, where $x = \left(E_v, N_f^{PVA}, N_e^{PVA}, N_w^{PVA}, N_t^{PVA}, N^{PLGA}, R_N\right)$. The objective function of the model minimizes the difference between the desired size (D) and the predicted size ($f(x)$).

$$\min_{x} |f(x) - D| \tag{39}$$

s.t.

$$N_f^{PVA} + N_e^{PVA} + N_w^{PVA} = N_t^{PVA} \tag{40}$$

$$N_t^{PVA} = N^{PLGA}.R_N \tag{41}$$

$$N_e^{PVA} \leq N_w^{PVA} \tag{42}$$

$$N_e^{PVA} \leq N_f^{PVA} \tag{43}$$

We employed the RF model, which yielded the lowest RMSE and MAE, to predict size in Eq. 39. The resulting mixed integer nonlinear programming model (MINLP) was solved using Gurobi 9.1 for desired sizes of 200, 250, and 300 nm. Table 12 summarizes the synthesis parameter values for each size. Our experimental collaborators conducted synthesis experiments at these parameter values and measured the resulting nanoparticle sizes, which are given in the last column of Table 12. The experimental results showed reasonable agreement between the desired size (first column in Table 12) and observed values. However, the experimental observation for the desired size of 250 nm has a larger error than nRMSE of the RF model. This larger error agrees with the high uncertainty observed for this range of sizes (Figure 21). This result suggests that there may be room for improving the predictive models for better agreement between the experimental observations and model predictions. The potential solutions may be adding more descriptive variables, which were not captured in the

first round of the experiments, increasing the number of data, and conducting a larger number of repetitions to decrease the uncertainty of the observed values.

Table 12. Synthesis parameter values obtained through optimization using the RF model and the corresponding experimental measurements

D (nm)	E_v (J/ml)	N_f^{PVA}	N_e^{PVA}	N_w^{PVA}	N_t^{PVA}	N^{PLGA}	R_n	Experiment results (nm)
200	230.36	3.02×10^{-6}	2.63×10^{-6}	2.52×10^{-7}	5.88×10^{-6}	2.99×10^{-8}	196.87	216.9
250	90.10	2.81×10^{-7}	2.93×10^{-6}	1.07×10^{-7}	3.32×10^{-6}	6.99×10^{-8}	47.01	224.6
300	31.45	3.58×10^{-7}	1.60×10^{-7}	2.59×10^{-7}	7.77×10^{-7}	1.62×10^{-7}	4.78	282.7

4.2.5 Conclusions

This work integrates the experimental data and machine learning techniques to build the best data-driven model for predicting the size and polydispersity of PLGA nanoparticles produced by the emulsion solvent evaporation method, using the most common solvent DCM and surfactant, PVA. While previous work has investigated the various synthesis factors for this highly popular method and polymer, none has produced a model of this process and the effect on size and polydispersity. This work applied machine learning and optimization to produce a predictive model for the first time.

Six different machine learning techniques were used to find the best-performing model for predicting the size and PDI of PLGA nanoparticles. Gaussian process regression, random forests, multivariate adaptive regression splines, artificial neural networks, extreme learning machines, and support vector regression were the modeling techniques investigated for constructing the models. The results revealed that random forest models predicted the size and PDI best with the lowest RMSE (22.19 and 0.041, respectively) and MAE (17.51 and 0.032, respectively). The

feature importance studies using the random forest model revealed the specific energy (total sonication energy divided by the total emulsion volume) and PVA to PLGA molar ratio as the key variables for the predictions. These features fit with the understanding of emulsion formation and stability. The energy needed to form emulsion droplets of a certain size from an initial bulk fluid is directly related to the work needed to change the total surface area of the emulsion, which is itself directly related to the energy from the vibration of the sonicator tip. In addition, the ability of the emulsion droplets to not coalesce is related to the viscosity of the bulk aqueous liquid and the interfacial tension with the organic phase, which can be captured by the molar ratio between the molecules of the two phases (PVA to PLGA). The random forest model was used in an optimization problem for estimating the synthesis parameter values, which would result in the desired size, where the error for two of the three points between the observed value and predicted size were within the uncertainty of the RF model predictions.

The best model (RF) yielded an RMSE of around 22 nm for predicting the average size of the nanoparticles. In many applications, this level of control would be acceptable. However, improvements to the model could reduce this error. For applying machine learning techniques, a larger data set with larger ranges of all synthesis parameters of interest would yield more accurate models. Additionally, procedures could be used to reduce any potential error or variability in parameters that may unknowingly exist, such as using polymers (PVA or PLGA) with a more defined molecular weight instead of using average molecular weights for calculations. Another improvement could be in feature optimization. For instance, high amplitude powers (>60%) were generally avoided during synthesis due to the high heat that high power settings distributed into the emulsion, which, if not properly controlled, could change the emulsion formation and thus the final nanoparticle size. An ice bath was used to avoid the problem, but this or other restraints could

be incorporated into the machine learning data prior to training so that it more completely captures the process. However, the main purpose of this work was to investigate using machine learning algorithms on common and popular process data and the ability to understand and control it for successful future nanoparticle modification and clinical translation. We believe this work shows a novel approach in this aspect and lays out a procedure for future research.

Consequently, the model from this work or an improved one in the future could be used for adaptive experimental campaigns and possibly scale-up designs. This initial work lays the groundwork for how this analysis technique can be used on other nanoparticle synthesis methods and materials.

Chapter 5 –**Surrogate Models of Stochastic Simulations and Surrogate-based Optimization**

5.1 Building Surrogate Models of Stochastic Simulations

This chapter proposes a new framework to construct surrogate models of stochastic simulations with uncertain parameters. The framework, PARIN (PARameter as INput-variable), aims to alleviate the limitations of the existing approaches. PARIN enables the use of any existing ML technique for building surrogate models of stochastic simulations, and it estimates the output uncertainty. PARIN considers the uncertain parameters of the stochastic simulations as input variables and converts the stochastic formulation of the simulations to a deterministic one. In PARIN, any ML technique can be used to approximate the new deterministic simulation. Finally, the uncertainty of the parameters is propagated through the surrogate model to the outputs using uncertainty propagation methods. We implemented six ML techniques to assess the performance of PARIN in terms of accuracy of output predictions using the output mean, standard deviation, distribution, and efficiency. The ML techniques include Gaussian process regression (GP), random forests (RF), support vector regression (SVR), artificial neural networks (ANN), extreme learning machines (ELM), and multivariate adaptive regression splines (MARS). Ultimately, PARIN's performance is compared to three existing approaches of fixed parameter value (Fixed), a subset of realization of uncertain parameters (PSet), and stochastic kriging (SK). The computational studies employed a diverse number of models in the form of test functions. The details of the PARIN, computational experiments and results are included in the following sections.

5.1.1 Proposed method: PARameter as INput-variable (PARIN)

In this study, we proposed a new approach for building surrogate models of stochastic simulations. The new approach is called PARIN, which stands for PARameter as INput-variable, and consists of three main steps (Figure 23). First, the uncertain parameters of the system are added to the input variables. Hence, the intrinsic uncertainty is converted to an extrinsic one. With consideration of the uncertain parameters as input variables of the simulation model, the stochastic formulation of the simulation becomes a deterministic one. In the second step, a surrogate model is built to represent the outputs of the deterministic simulation. Finally, in the last step, the surrogate model is converted back into a stochastic one by separating the uncertain parameters from the input variables and propagating their uncertainty to the output using uncertainty propagation methods. PARIN approach conserves the uncertainty information of each uncertain parameter through the modeling step, which can be used to study the impact of each uncertain parameter on the simulation output and to carry out sensitivity analysis. The structure change of the formulation from stochastic to deterministic provides the possibility of implementing various types of ML techniques for surrogate model construction.

The general workflow of PARIN is given in Figure 23. Assume $Y = g(X; K)$ is a high-fidelity stochastic simulation model with uncertain parameter(s) K, where Y is the stochastic output, X is the input vector with d_1 components, and K is the vector of system uncertain parameters with the dimension of d_2. Using PARIN, the stochastic simulation, $Y = g(X; K)$, is converted to a deterministic one, $Y' = \acute{g}(X')$, by considering the uncertain parameters as additional inputs to the system (Figure 23). In the new simulation model ($\acute{g}(X')$), X' is a d-dimensional vector of inputs with $d = d_1 + d_2$, which contains all inputs and the uncertain parameters of the stochastic simulation ($g(X; K)$). We assume that the distributions of the

uncertain parameters (K) are known, and the parameters of the distributions are constant. Any surrogate modeling technique can be utilized to train a model representing the generated deterministic simulation, $g'(X') \approx h'(X')$, where $h'(X')$ is the trained surrogate model of the deterministic simulation (Figure 23).

Different sampling schemes with various sampling budgets could be used to generate the training input/output data sets. For instance, the original input variables and uncertain parameters, which are assumed as input variables now, can be sampled separately. Then, the full factorial design of these two sample sets, random combination, or a similar method could be used to generate the final input sample sets. Another method can be to sample the combined space of the original input variables and uncertain parameters, which are now assumed as additional input variables of the model. Depending on the sampling method and combination of the samples, the budget allocation for the number of samples could be different. For instance, consider a total budget of 100 function calls for a model with one input variable and one uncertain parameter. The function call budget could be utilized by the full factorial design with 10 levels for the input variable and uncertain parameter. On the other hand, the budget could be utilized by randomly generating 100 samples from the combined input variable and uncertain parameter space. We use low-discrepancy sampling methods to sample the X' input space and produce the training data set. Once the training data are ready, the surrogate model $h'(X')$ is generated (Figure 23). The uncertain parameters K are nested in variable X', hence, the surrogate model representing the stochastic simulation is now shown with $\hat{Y} = h\,(X; K)$ in Figure 23.

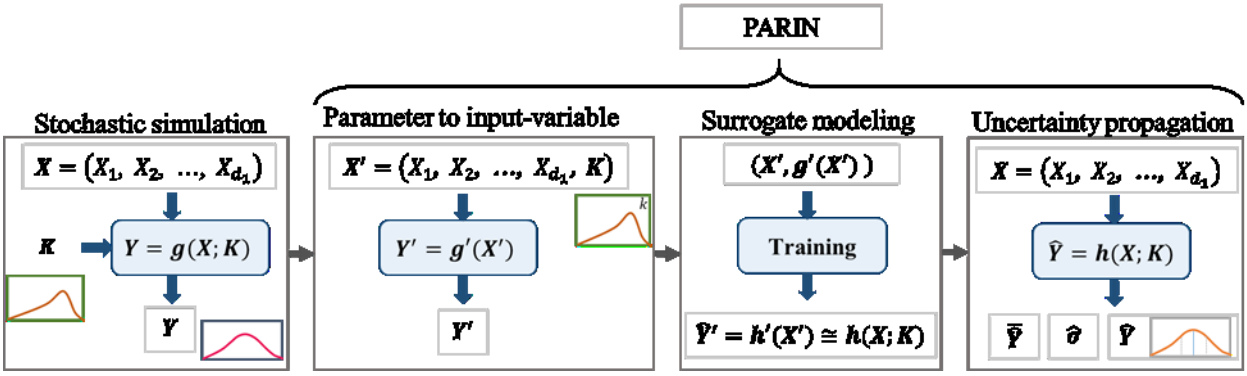


Figure 23. Transformation of the stochastic model ($g(X;K)$) to a deterministic model ($g'(X')$) by extracting uncertain simulation parameters and using them as additional inputs to the simulation. The deterministic simulation is represented by a surrogate model ($h'(X')$). Finally, the output distribution ($\hat{Y}$), mean ($\bar{\hat{Y}}$), and standard deviation ($\hat{\sigma}$) are estimated using uncertainty propagation methods.

The output mean ($\bar{\hat{Y}}$), standard deviation ($\hat{\sigma}$), and distribution is estimated by propagating the uncertainty of the uncertain parameters K through the surrogate model $h\,(X;K)$ (Figure 23). Although PARIN can utilize any uncertainty propagation method, in this dissertation, we implemented Monte Carlo simulation method with Halton sampling to estimate the predicted mean output value and its standard deviation. The predicted mean output value ($\bar{\hat{Y}}_i$) and standard deviation ($\hat{\sigma}_i$) for test point i is calculated using 1000 repetitions at each point via Eq. 44 and Eq. 45, respectively, where $\hat{Y}_{ij}$ corresponds to j^{th} repetition output of test point i.

$$\bar{\hat{Y}}_i = \frac{\sum_{j=1}^{10^3} \hat{Y}_{ij}}{10^3} \tag{44}$$

$$\hat{\sigma}_i = \sqrt{\frac{\sum_{j=1}^{10^3} \left(\hat{Y}_{ij} - \hat{Y}_i\right)^2}{10^3}} \tag{45}$$

For example, consider one dimensional Ackley function given in Eq. 46 with $x \in [-32.77, 32.77]$, and assume that the uncertain parameter is $c \sim N(20, 2)$. Using PARIN, the formulation of Ackley is changed to Eq. 47, where x_2 corresponds to the uncertain parameter c, which is now considered an additional variable (First step of PARIN, Figure 23). Then, 1000

samples of $X' = (x_1, x_2)$ are generated using Sobol sampling. The output for each sample is calculated through Eq. 47. Then, using these input/output data, the surrogate model, $h(X')$, is trained (Second step of PARIN, Figure 23). The formulation is converted back to $h(x; c)$, and using Monte-Carlo simulation-based methods, the uncertainty of parameter c for each given test point is propagated (Final step of PARIN, Figure 23). Halton sampling with 1000 samples is used to sample c parameter space for each given test point of X^*, and the mean and standard deviation of each test point is calculated using Eqs. 44 and 45. Figure 24 includes three plots. The first plot, Function in Figure 24, shows the output means and standard deviations for 50 test points for Ackley function (Eq. 46). The other two plots, GP and MARS in Figure 24, show these values predicted by PARIN using two ML modeling techniques of GP and MARS, as examples. In Figure 24,the markers represent the output mean, and the error bars correspond to one standard deviation around the mean for each test point in each plot. According to Figure 24, GP and MARS models accurately estimate the mean and standard deviation of different test points for the Ackley function, and the estimations of the GP model are closer to the true output mean and standard deviation.

$$Y = f(x) = -c\, exp(-0.2|x|) - exp(\cos(2\pi x)) + c + \exp(1) \tag{46}$$

$$Y' = f(x_1, x_2) = -x_2\, exp(-0.2|x_1|) - exp(\cos(2\pi x_1)) + x_2 + \exp(1) \tag{47}$$

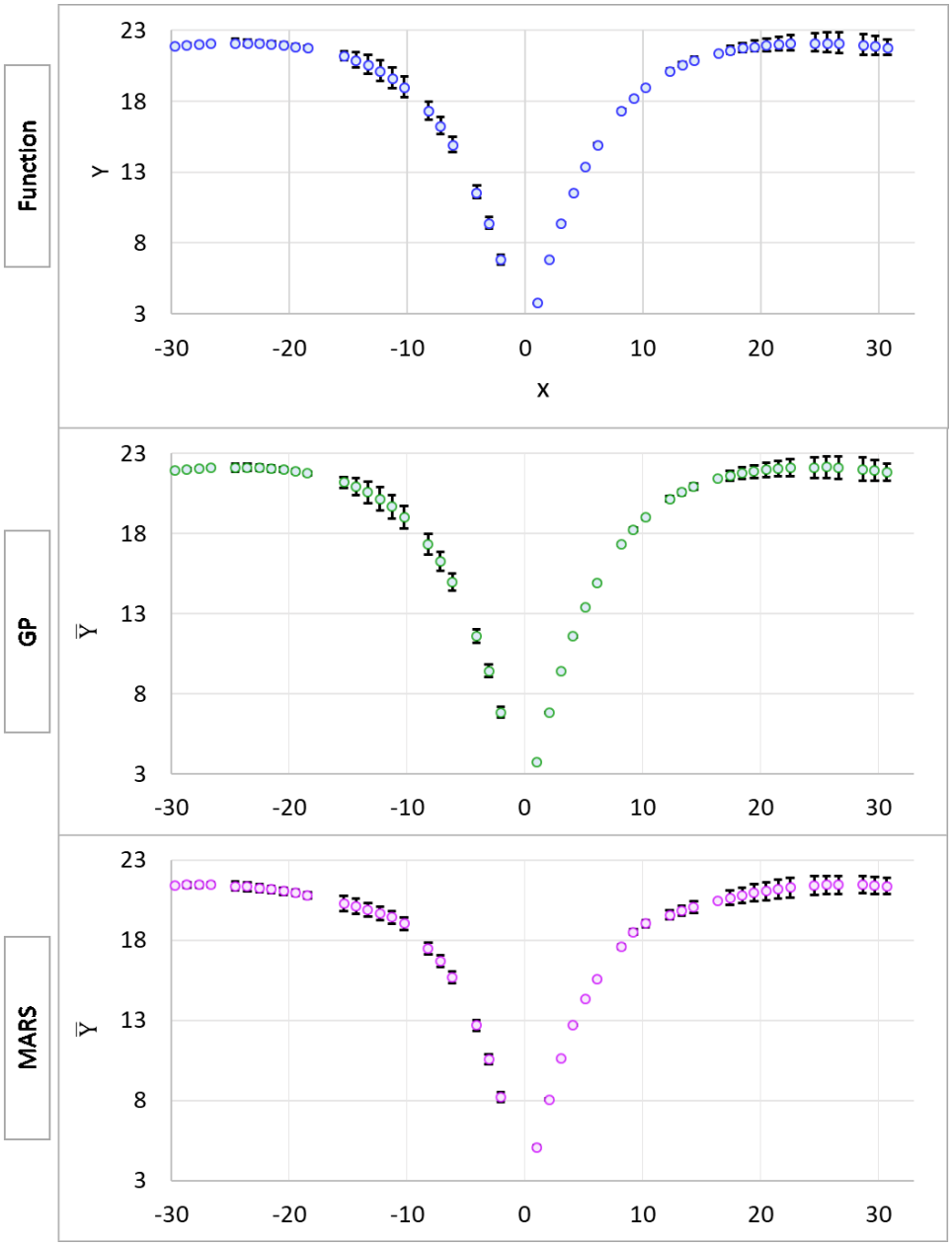


Figure 24. Output mean values for Ackley function using the original function (Eq. 46) and PARIN using two modeling techniques of Gaussian process (GP) regression and multivariate adaptive regression splines (MARS). The error bars show one standard deviation around the output mean values.

5.1.2 Computational Experiments

Two case studies were designed to investigate the impact of simulation input dimensions and the number of uncertain parameters on the accuracy and efficiency of PARIN compared to Fixed, PSet, and SK. The first case study included nine test functions with an adjustable number

of inputs, referred to as model dimensions, from the Virtual Library of Simulation Experiments Optimization Test Suite (Surjanovic and Bingham, 2013). These test functions were utilized to study the impact of model dimension on the performance of PARIN, Fixed, PSet, and SK in estimating the output of the stochastic simulations and the uncertainty associated with them. The details of input ranges and parameters of uncertainty distributions of the test functions are included in Table A3.1 of Appendix 3. The number of inputs/dimensions (D) was increased from one to thirty-two geometrically for each test function, $D \in \{1, 2, 4, 8, 16, 32\}$. The computational budget, b_D, available to train the surrogate models were defined using the number of model runs for generating the training data. Four different budgets, $b \in B_D = \{100, 250, 500, 1000\}$, were considered to assess the efficiency of the approaches.

The second study investigated the influence of the number of uncertain parameters,n_K, in the stochastic simulations on the performance of the four approaches. The study employed fourteen test functions from the Virtual Library of Simulation Experiments Optimization Test Suite (Surjanovic and Bingham, 2013) with $n_K \in \{1, 2, 3, 4\}$. Four different budget values were used for the second case study, as well, where $b \in B_K = \{120, 360, 600, 1080\}$.

Figure 25 illustrates the steps for the computational experiments. For each approach, data generation was the first step of the process. Sobol series and Halton sequences were used for sampling inputs and uncertain parameters to generate the training and testing data sets. Uncertain parameters were assumed to be distributed normally, and the parameters of the distributions for each test function are included in Table A3.2 of Appendix 3. For the Fixed method, the values of the uncertain parameters were set to their mean values, and the sampling for training data was conducted only for input variables of each test function. For PSet and SK methods, the sampling was done for inputs (X) and parameters (K), separately, then the simulation runs were conducted

using a final sample set which was the Cartesian product of the two sample sets. The number of uncertain parameter samples of each function was set to $m_k = 10 \times n_k$ for PSet and SK approaches. Therefore, for a given budget, b, the number of function input samples, m_X, was calculated via Eq. 48. In the last framework, PARIN, the uncertain parameters are considered additional input variables. The new input variable (X') space was sampled and used for output value generation from the simulation runs.

$$m_X = \frac{b}{m_k} \tag{48}$$

The training data was used to build the surrogate models in the second step of the computational experiments. For all approaches except the SK method, which only uses kriging, six ML modeling techniques including Random forest (RF) (L. Breiman, 2001), Gaussian process (GP) (Williams and Rasmussen, 2006), support vector regression (SVR) (Drucker et al., 2002), multivariate adaptive regression splines (MARS) (Friedman, 1991), artificial neural networks (ANN) (Haykin, 2009), and extreme learning machines (Huang et al., 2006), were trained using Scikit-learn package 0.23.2 (Pedregosa et al., 2011) in python 3.7 using Auburn University HPC resources. A brief explanation of each of the ML modeling techniques is included in Appendix 3. The hyper-parameters of the surrogate models were tuned utilizing five-fold cross-validation (Wong, 2015). Using Halton series N number of samples from input variables were selected as the test points. In the next step (Figure 25), the output values for the test points were predicted using all the trained surrogate models for each approach. The PSet, SK, and PARIN approaches return the mean predicted output value, its standard deviation, and the estimated output distribution for a given test point. Only the "mean" value is obtained for a given test point using the Fixed approach. The mean predicted output value for the PSet is the mean value of all the trained models' predictions. The standard deviation is the standard deviation of these predictions, and it can be

viewed as a measure of the prediction's uncertainty. SK model returns a normal distribution for the output with a known mean and standard deviation. The mean and standard deviation for PARIN is calculated through Monte-Carlo simulation-based uncertainty propagation of the uncertain parameters through the final trained model.

The last step (Figure 25) was the comparison of different approaches to each other in terms of their accuracy and efficiency. The comparison was conducted using two metrics explained in Section 5.1.2.1. Because Fixed, PSet, and PARIN methods allow using different ML techniques, the comparisons also assessed the impact of ML techniques.

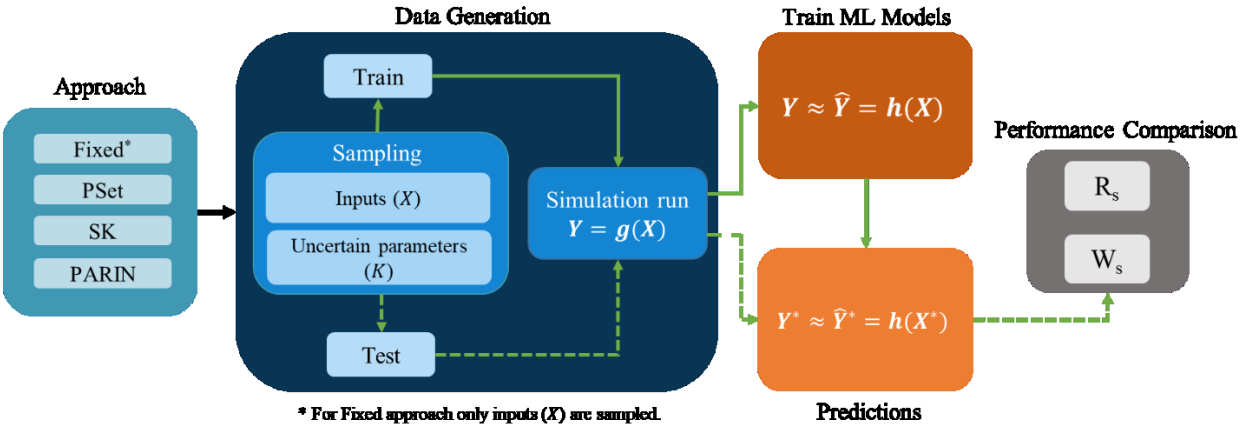


Figure 25. The framework for computational experiments. In the first step, training and test data sets are generated for each of the four approaches, including 1) Fixed: fixing the uncertain parameter value, 2) PSet: subset of realizations of the uncertain parameter, 3) SK: stochastic kriging, and 4) PARIN: the proposed approach. Then, the simulation ($Y = g(X)$) is represented by trained Machine learning (ML) models ($\hat{Y} = h(X)$). Then, the output ($\hat{Y}^*$) for each test point X^* is calculated using the trained model. Finally, two comparison metrics of rank-score (R_s) and Wasserstein score (W_s) are evaluated to conduct the comparison among different approaches.

5.1.2.1 Comparison Metrics

Four surrogate modeling approaches, Fixed, PSet, SK, and PARIN, were compared based on the accuracy and efficiency of the methods. Accuracy was evaluated using normalized root mean square error (nRMSE) and Wasserstein distance (Villani, 2009). Efficiency was evaluated

based on the accuracy of the models on different fixed computational budgets, defined as the number of simulation runs available for training the surrogate models.

5.1.2.1.1 Normalized Root Mean Square Error (nRMSE)

The normalized root mean square error (nRMSE) is used for the evaluation of the predictions for both outputs mean and standard deviation. The nRMSE between the two predicted outputs from each model, which are mean ($\hat{\bar{Y}}_i$) and standard deviation ($\hat{\sigma}_i$) for i^{th} test point, and their true values (Y_i, σ_i) is calculated using Eq. 49 and 50, where subscripts max and min correspond to the maximum and minimum of these outputs in the training data, respectively. Since six ML models were trained, for three of the approaches, Fixed, PSet, and PARIN, a new score, R_s, based on the nRMSE, was defined to compare different approaches to each other based on the best performing ML modeling technique in each approach.

$$nRMSE_Y = \frac{\sqrt{\frac{\sum_{i=1}^{N}(Y_i - \hat{\bar{Y}}_i)^2}{N}}}{Y_{max} - Y_{min}} \tag{49}$$

$$nRMSE_\sigma = \frac{\sqrt{\frac{\sum_{i=1}^{N}(\sigma_i - \hat{\sigma}_i)^2}{N}}}{\sigma_{max} - \sigma_{min}} \tag{50}$$

Calculation of rank score (R_s) consists of four main steps. In the first step, the ML technique with the lowest nRMSE for each approach is selected for each test function, insuring that each approach is represented by the most accurate ML technique for that test function. In the next step, the four approaches are sorted based on their nRMSE for each function. The score S for each approach is obtained from Table 13 based on the sorted errors for the given function. Steps one through three are carried out for all the test functions. Finally, the rank score for k^{th} approach (R_s^k) is calculated using Eq. 51, where n_f corresponds to the number of test functions. The values

of R_s is between 0 and 1. The higher the score, the more often the nRMSE was the lowest for that approach, i.e., Fixed, PSet, PARIN, SK methods.

$$R_s^k = \frac{\sum_{j=1}^{n_f} S_j^k}{4n_f} \tag{51}$$

Table 13. The Score (S) associated with the rank of each approach.

Rank	Score (S)
1st	4
2nd	3
3rd	2
4th	1
No information	0

5.1.2.1.2 Wasserstein Distance

Wasserstein distance (Villani, 2009) is a similarity metric between the distribution of two variables and calculates the distance between two distributions (Sun et al., 2018). The distance, W_d, between distributions of variables F and G with cumulative probability density functions U and V, respectively, is calculated using Eq. 52. In this study, the Wasserstein score, W_s^i, for approach i is defined to as the percentage of the test functions for which the estimated output distribution using approach i has the lowest Wasserstein distance.

$$W_d(F, G) = inf\|U - V\| \tag{52}$$

5.1.3 Results and Discussion

The first step for comparison of the accuracies from different approaches was the calculation of the rank score (R_s) discussed in Section 5.1.2.1.1 Six different ML modeling techniques were used to build surrogate models for Fixed, PSet, and PARIN approaches. The nRMSE was the metric to choose the best ML modeling technique with the lowest error value for each test function to represent the trained model for each of these three approaches. Figure 26 shows the nRMSE values from each modeling technique to predict the mean and standard deviation of the Griewank function as an example, for a given number of input dimensions with a budget equal to 10^3. The results show that the ML modeling techniques with the lowest nRMSE for each dimension were not the same for all the considered number of input dimensions. The ML techniques with the lowest nRMSE were selected as the representative model for each of the Fixed, Pset, and PARIN approached, hence the best performing models from each approach were compared to each other in terms of their accuracy.

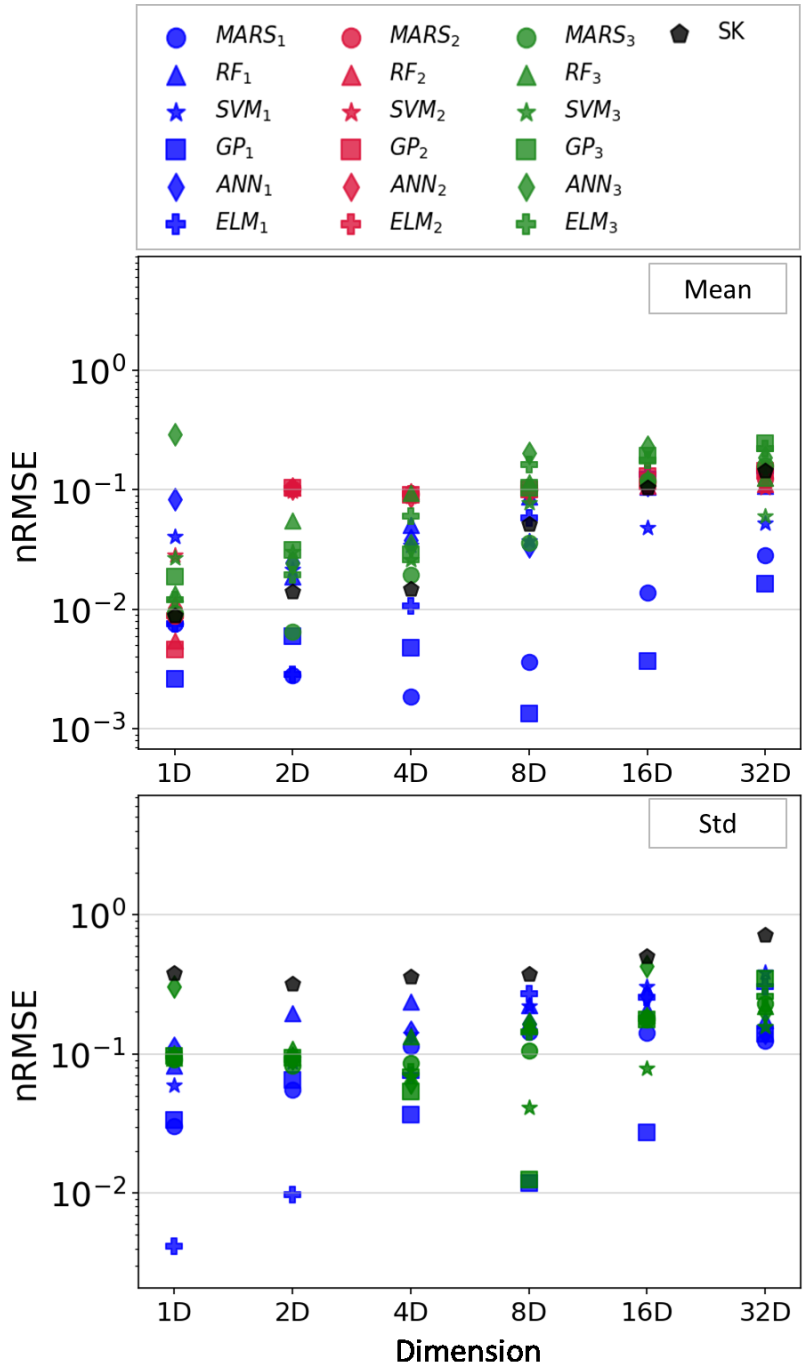


Figure 26. The normalized root mean square error (nRMSE) from each modeling technique in the prediction of mean and standard deviation (Std) of Griewank function with the computational budget of 1000. The subscripts 1, 2, and 3 correspond to PARIN, Fixed, and PSet approaches, respectively.

5.1.3.1 Results on the Impact of Input Dimension

In Figure 27, the accuracy of the four different approaches of Fixed, PSet, PARIN, and SK in estimating the mean and standard deviation of test functions with different numbers of input variables, i.e., dimensions, are compared. The comparison is based on the rank score, R_s, with maximum budget value ($b = 1000$). According to Figure 27, PARIN has the highest score for estimating both the mean and standard deviation of the output for all considered dimensions of test functions compared to the other approaches. SK had the highest deterioration of performance in estimation of the mean because the R_s value decreases significantly as the number of inputs increases. Fixed and PSet approaches have similar scores in predicting the mean for dimensions larger than two. The results suggest that the PARIN approach with the right ML modeling technique has the highest accuracy in predicting the mean and standard deviation given a large budget for simulation runs regardless of the number of dimensions.

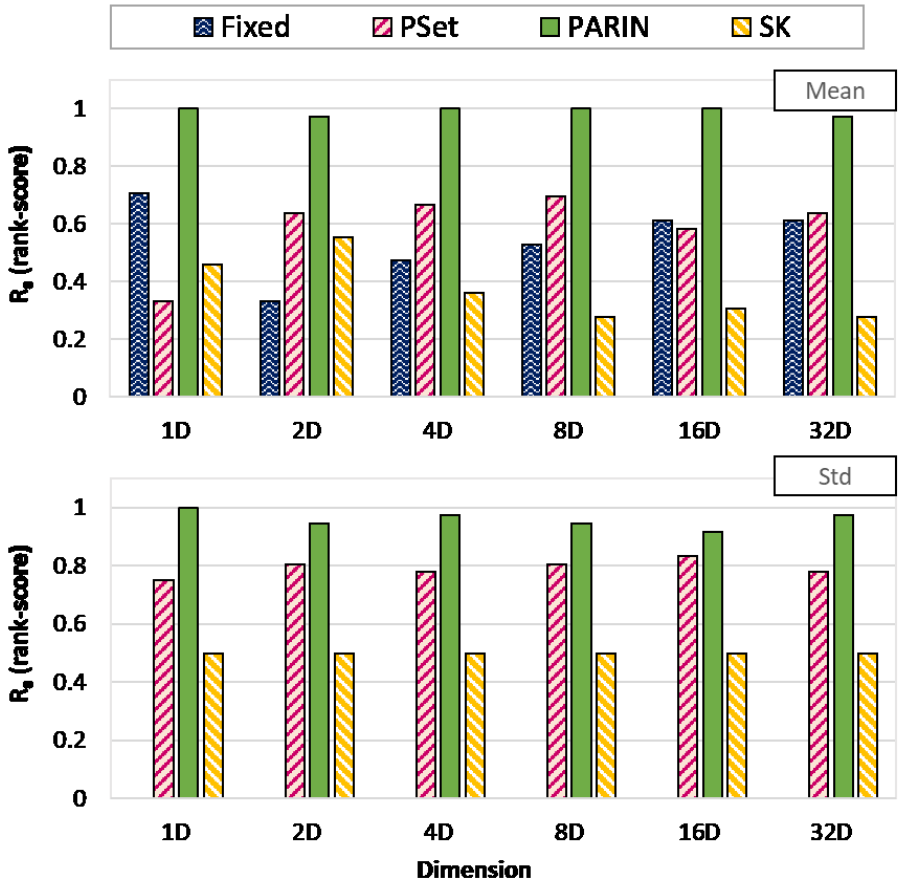


Figure 27. The rank-score plot of mean and standard deviation (Std) estimations for fixed-parameter (Fixed), parameter-set (PSet), PARIN, and stochastic kriging (SK) approaches given different input dimensions for a simulation run budget of 1000.

Figure 28 shows the rank score of Fixed, PSet, PARIN, and SK approaches in predicting the mean and standard deviation, first and second column, respectively, with different budget values and input dimensions. Based on the plot, PARIN yields the highest rank score in predicting the mean and standard deviation of the outputs except for the low budget cases of high dimensional functions, e.g., 16D and 32D. As the available budget for the number of simulation runs decreases, the accuracy of PARIN estimations deteriorates for high dimensional test functions. The high number of input dimensions and low available resources for simulation runs limit PARIN's ability to explore and cover enough uncertainty space to capture its impact. Based on these results, for

functions with high dimensions and low computational budget, PSet has the highest rank score values in predicting the mean and standard deviation of the function output. The SK approach yields the lowest, i.e., worst, score for predicting the mean and standard deviation given all budget values.

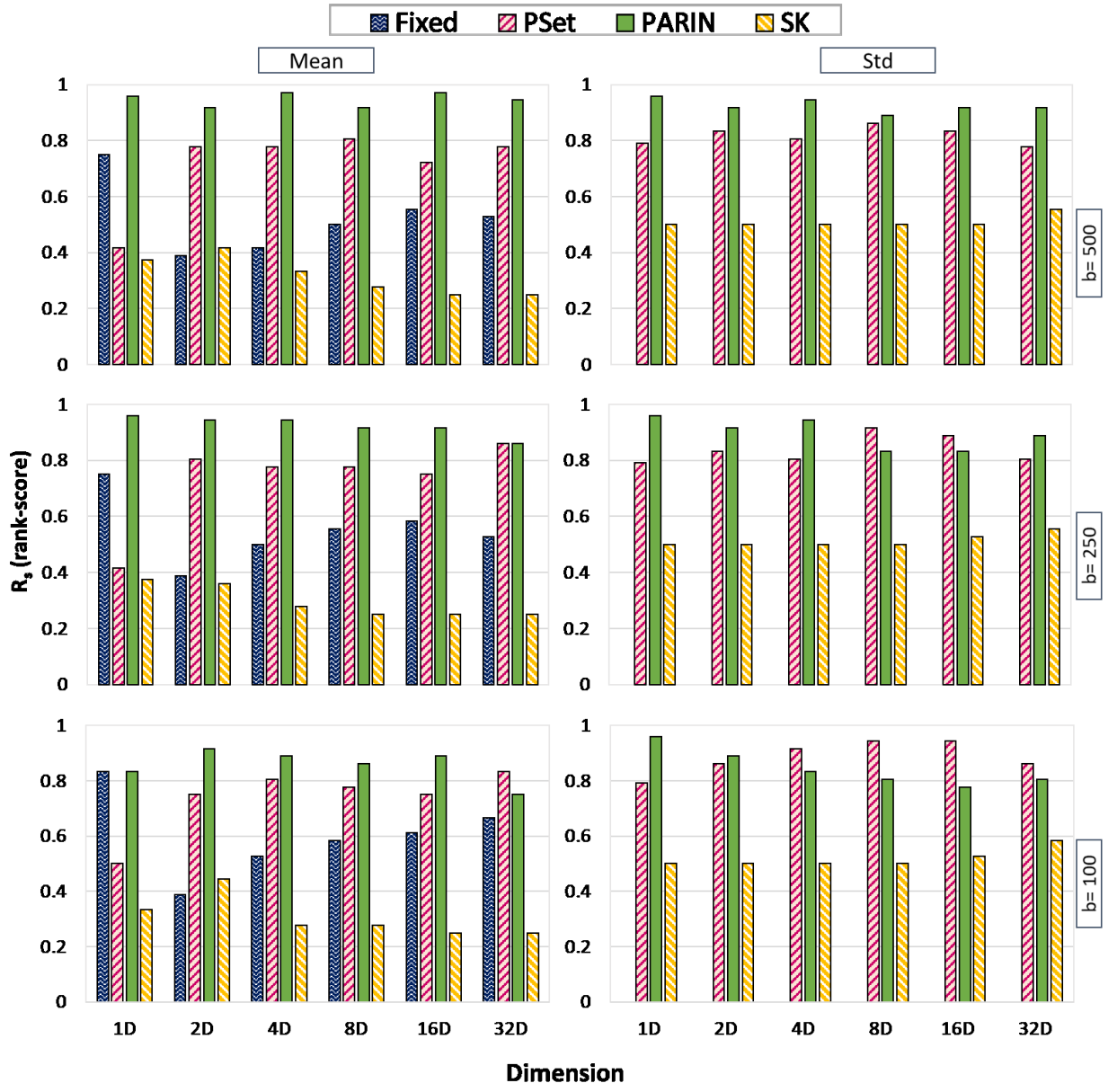


Figure 28. Rank score values for mean and standard deviation (Std) estimations for fixed-parameter (Fixed), parameter-set (PSet), PARIN, and stochastic kriging (SK) approaches given different input dimensions and budget values ($b \in B_D$).

The accuracy of the predicted output distribution was another metric for comparing all four approaches. Figure 29 illustrates the Wasserstein score (W_s^i), introduced in Section 5.1.2.1.2, for PSet, PARIN, and SK approaches for four different budget values. The Fixed method does not provide distribution estimates thus, W_s^i is not calculated for it. Figure 29 reveals that PARIN had the lowest Wasserstein score for all the test functions at the highest budget value ($b = 1000$). However, the accuracy of the estimated distribution by PARIN decreases with decreases in the available budget. The Wasserstein score for PARIN decreases drastically for low budget cases, $b = 100$ or $b = 250$, as the dimension of the test function grows, leading to higher distance values in comparison to PSet approach estimations. The distribution predicted by the SK method has the largest Wasserstein distance (W_d) for all the dimension and budget values, resulting in the Wasserstein score (W_s^i) value of zero.

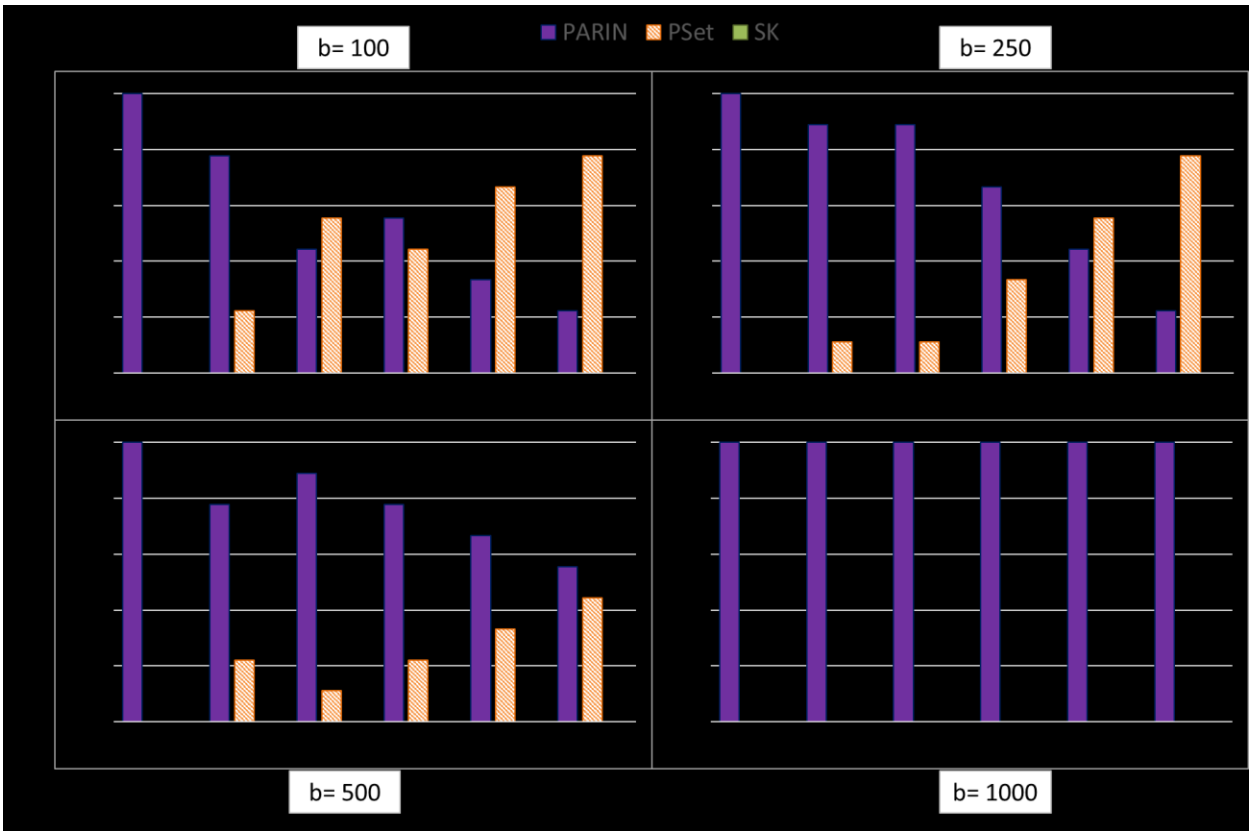


Figure 29. Wasserstein score for PARIN, parameter-set (PSet), and stochastic kriging (SK) for functions with different dimension values given four different budget (b) values.

5.1.3.2 Results on the Impact of the Number of Uncertain Parameters

Figure 30 shows the rank score results as the number of uncertain parameters increased from one to four for estimating the mean and standard deviation of two-dimensional test functions using Fixed, PSet, PARIN, and SK approaches at the maximum budget ($b = 1080$). Based on Figure 30, PARIN had the highest rank score among all approaches in estimating the mean for functions with less than three uncertain parameters. With the higher number of uncertain parameters, PARIN is the second-best method for estimating the mean after the Fixed approach based on the R_s value. On the other hand, PARIN consistently yields the highest rank score in the estimation of the standard deviation value for all the functions with a various number of uncertain

parameters. After PARIN, PSet is the next recommended approach for the prediction of the output standard deviation according to the R_s values from the plot. SK has the lowest rank score values among all four approaches for estimating mean and standard deviation.

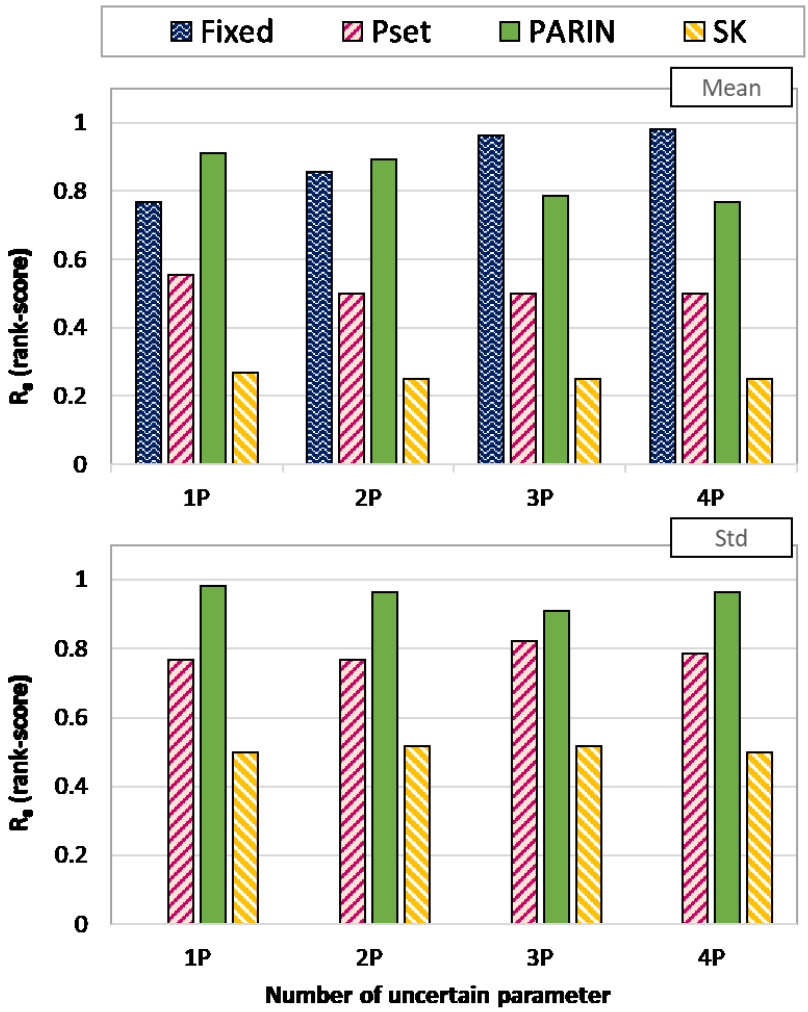


Figure 30. Rank score plot of mean and standard deviation (Std) estimations for fixed-parameter (Fixed), parameter-set (PSet), PARIN, and stochastic kriging (SK) approaches for two-dimensional functions with different numbers of uncertain parameters.

Figure 31 shows the results for the estimation of the mean and standard deviation of the two-dimensional test functions with various numbers of uncertain parameters at different budget values. According to the plot, the Fixed approach is the best method with the highest rank score values for estimation of the mean values for functions with different uncertain parameters in cases

with low available budgets, and PARIN is the next recommended method with R_s values larger than PSet and SK. The highest rank score for prediction of the standard deviation with all the budget values belongs to PARIN for all the test functions regardless of the number of uncertain parameters.

The Wasserstein score for each approach was calculated for the test functions in this case study with a different number of uncertain parameters. PARIN had the highest Wasserstein score among all the approaches with different values of the computational budgets. This result showed that the output distribution estimated by PARIN had the lowest Wasserstein distance and consequently the predicted distribution had higher similarity to the original output distribution in comparison to the other approaches. The Wasserstein distances from all the approaches using different ML techniques for the test functions are included in Appendix 3.

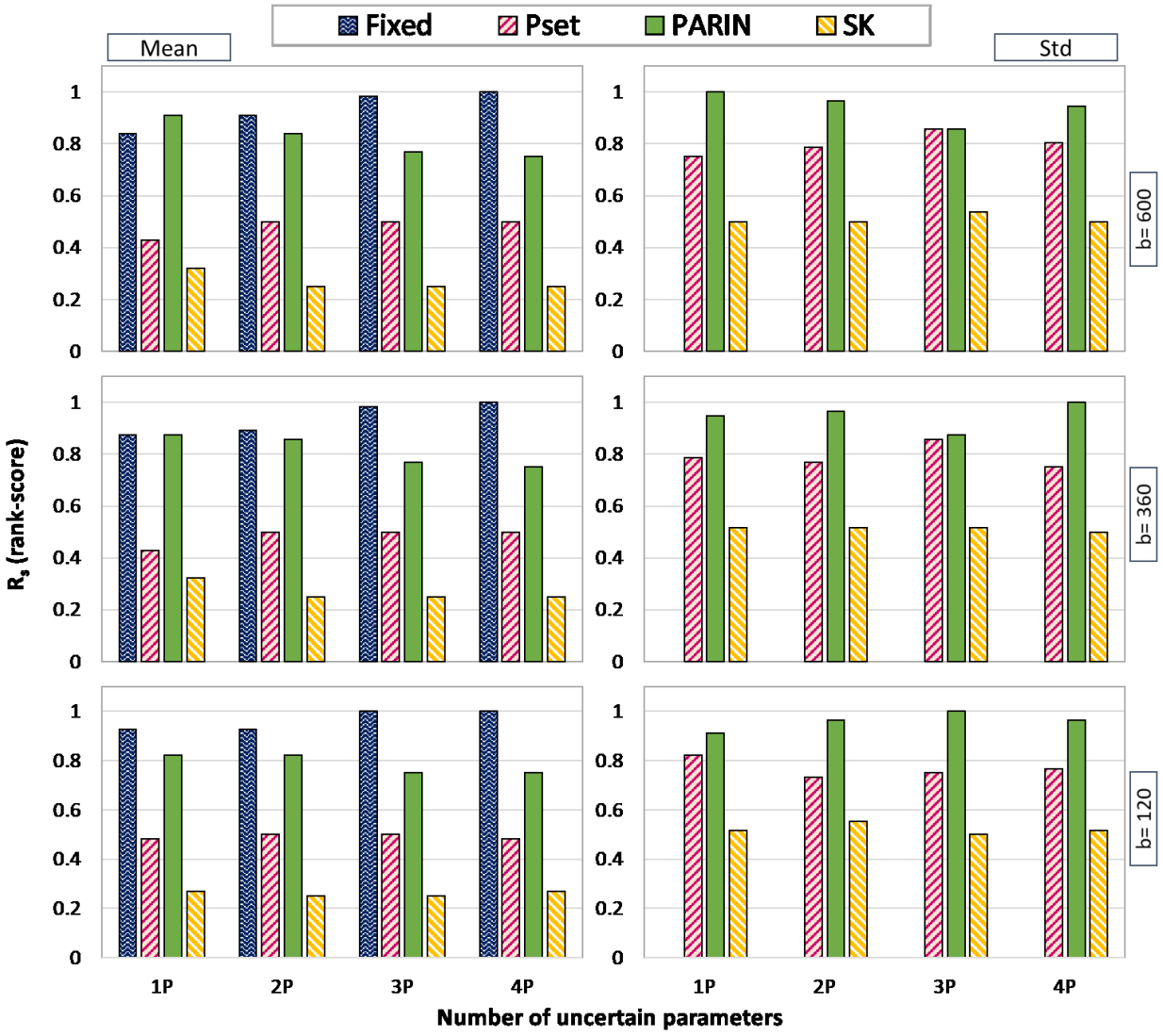


Figure 31. The rank-score plot of mean and standard deviation (Std) estimations for fixed-parameter (Fixed), parameter-set (PSet), PARIN, and stochastic kriging (SK) approaches given a different number of uncertain parameters and budget values (b).

5.1.4 Conclusions

This study proposed a new approach to building accurate surrogate models of high-fidelity stochastic simulations. Six different machine learning techniques are used to train surrogate models. The proposed method, PARameter as Input-variable (PARIN), is compared to three existing methods, including a fixed parameter (Fixed), parameter realization sets (PSet), and stochastic kriging (SK) based on accuracy and efficiency metrics using several test functions. The

results suggest that PARIN has a lower error in estimations of the function average and standard deviation values in comparison to the other three methods for functions with a different number of input variables. However, the accuracy of PARIN decreases for the high dimensional functions with lower available computational budgets. The test functions with the number of uncertain parameters greater than one, PARIN had the minimum error compared to other methods for predicting both mean and standard deviation of the outputs even for the lower computational budget values. Moreover, the estimated output distribution by PARIN had the highest similarity to the original output distribution for all the functions based on the Wasserstein distance. The decrease in the computational budget causes a drop-down in similarity metric between the original and estimated distribution by PARIN for functions with very large dimensions.

5.2 Surrogate-based Optimization of High-fidelity Simulations

One major application of the surrogate models is optimizing processes that are represented with complicated models or high-fidelity simulations. When the models or simulations are stochastic, an efficient approach is needed to build their surrogate models. I proposed to use the PARIN approach to build the surrogate models for stochastic models in a surrogate-based optimization framework. For preliminary analysis, I implemented PARIN in two stochastic programming models that employ two test functions with uncertain parameters. Then, I compared the optimum identified using the model trained by PARIN to the optimums obtained when PSet and SK approaches were used to build the surrogate models.

5.2.1 Computational Experiments

Two-dimensional test functions of Three hump camel and Booth with four and three uncertain parameters, respectively, were used to study the application of PARIN for surrogate-

based optimization. The results were compared to the solutions from PSet and SK approaches as the surrogate modeling frameworks. The computational budget of 1080 simulation runs was considered for building the six different types of surrogate models. The ML model with minimum nRMSE was chosen to be the surrogate model for each approach used in the optimization model. Sample average approximation (Hannah, 2004) was implemented to calculate the expected value of the objective function. Monte-Carlo simulation method with Sobol sampling was used to generate the set of scenarios (K) for uncertain parameters of the test functions. The optimization problem is shown in in Eq. 53, with bound constraints on input variables shown in Eq. 54. The optimization was formulated using Pyomo (Bynum et al., 2021; Hart et al., 2011) and was solved using Antigone (Misener and Floudas, 2014).

$$\min_{x} \frac{\sum_{k \in K} h(x;k)}{|K|} \tag{53}$$

s.t.

$$x_i \in [x_{min}, x_{max}] \tag{54}$$

5.2.2 Optimization Results

Figure 32 shows the optimum values of optimization problems for the Three hump camel and Booth functions. The location of the optimum solution values from PARIN and PSet and the original function are included in the figures. SK kriging was not solved within the maximum allowable time because the solution time is a function of the number of training data points, and 1080 points resulted in a very long solution time. Based on Figure 32, the distance between the optimum decision variable values estimated by PARIN and the function itself was lower than PSet for both Booth and Three hump camel functions.

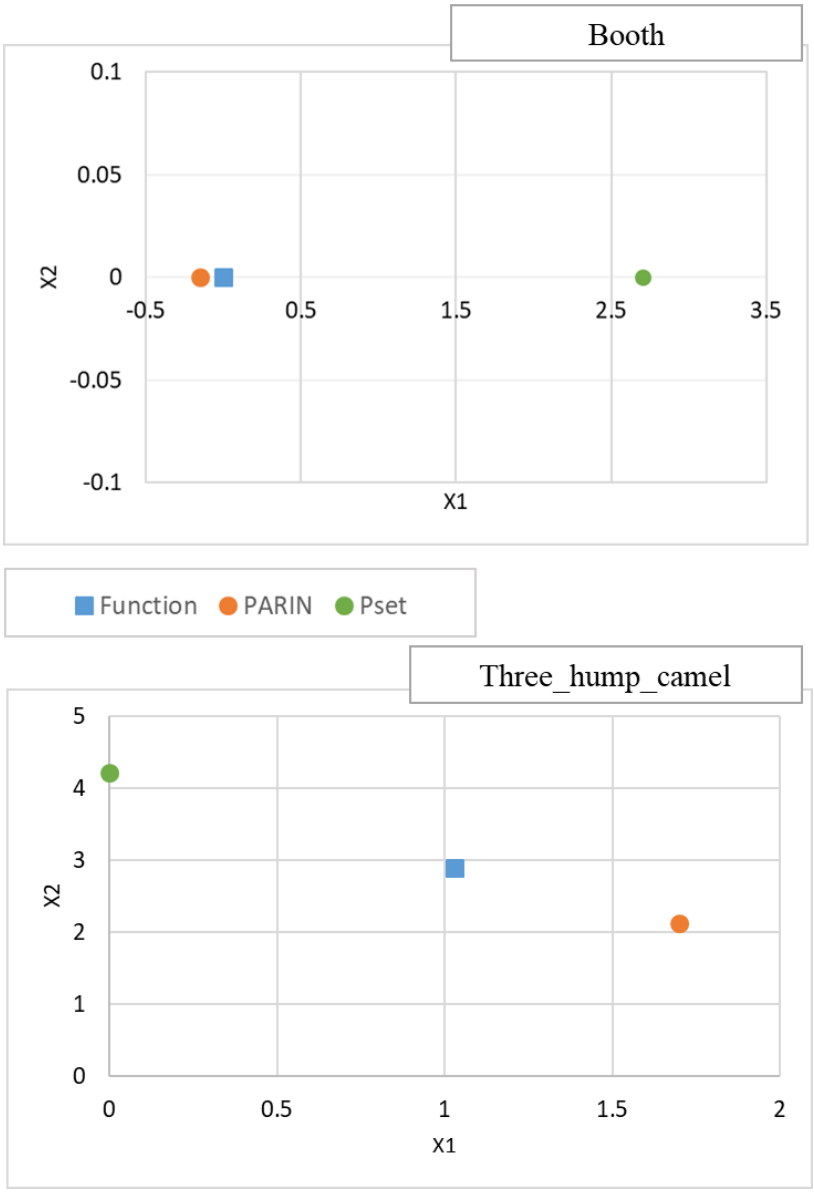


Figure 32. Optimum location solution from original function, PARIN, and PSet.

5.2.3 Conclusions

This study investigated the implementation of PARIN and the other two surrogate modeling approaches of PSet and SK for surrogate-based optimization problems. Two test functions were used as the case study for the comparison of the optimum solutions. The results showed that the optimum location found by using PARIN was significantly close to the original

functions' optima point, and SK had a large solution time and was not solved within the maximum allowable time in this computational experiment. These results were the first steps toward using PARIN for optimization and showed its potential for use in surrogate-based optimization. In the future, more intense and comprehensive experiments must be conducted to evaluate PARIN in surrogate-based optimization.

Chapter 6 – **Conclusions and Future Directions**

In this dissertation, I focused on handling high-fidelity stochastic simulations. First, I developed a guideline to select the most efficient uncertainty propagation methods for simulations with uncertain inputs. Then, I developed a new approach to building accurate surrogate models of high-fidelity stochastic simulations with uncertain parameters.

6.1 Assessment of Uncertainty Propagation Methods

Chapter 3 comprehensively investigates and compares the performance of several different uncertainty propagation (UP) methods for estimating the first four statistical moments of the outputs for simulation with uncertain inputs. The comparisons are based on the efficiency of each method. The efficiency is defined as the number of required simulation runs to achieve an estimate within a 5% error gap of the true values for each moment estimation. The UP methods considered include the Monte-Carlo method using Sobol series, Halton series, and Latin Hypercube sampling (LHS), Full Factorial Numerical Integration (FFNI), Univariate Dimension Reduction (UDR), Sparse Grids (SG), and Polynomial Chaos Expansion (PCE). The results of this comparison led to guidelines for selecting UP methods based on several simulation function characteristics.

The standard deviations of the uncertainty distributions used in this study were kept constant. Different standard deviations can be used in future analysis to investigate their impact on UP methods performances. In addition, the guidelines developed in this study can be used for two different applications: 1) building surrogate models of stochastic simulations and 2) scenario generation for optimization of stochastic simulations.

6.2 Machine Learning Applications and Comparison

Chapter 4 includes two machine learning (ML) applications in classification and regression for two experimental systems as black boxes. In the first study, a model is constructed to classify cardiomyocytes (CM) content, produced by differentiation of human-induced pluripotent stem cells (hiPSCs) on day 10 of differentiation, into two classes of '*sufficient'* and *'insufficient.'* The second study used ML techniques to build data-driven models to predict the size and polydispersity (PDI) of polymeric nanoparticles manufactured for drug delivery applications.

6.2.1 Classification of Hydrogel Encapsulated Cardiomyocytes Content

Section 6.2.1 describes my research with our collaborators to classify the CM content produced through differentiation of hydrogel encapsulated hiPSCs on day 10 of the differentiation process. The ability to accurately predict the sufficiency of CM content of a batch at an earlier time point is a significant step for efficient biomanufacturing of the CMs due to the high expense associated with the process. We used several ML techniques to build a classifier, including feature engineering, feature selection, and classification models. The feature selection methods considered were filter methods, embedded methods, principal component analysis, and wrapper methods. Random forests, Gaussian process classifiers, and support vector machines were used to build the classification models. The best model was a Gaussian process classifier using four principal components, selected by the forward selection method, with accuracy, recall, precision, and Mathew's correlation coefficient (MCC) of 0.75, 0.59, 0.71, and 0.46, respectively.

The features selected by the best classifier included post-freeze passage number, fibrinogen concentration in PEG-fibrinogen (PF), and the ratio between the CHIR concentration and microspheroid surface to volume ratio as the predictors of the classes, which were in agreement with observations from the literature and experiments. However, the four principal components

(PCs) only explained 18% of the input variance, suggesting that the output variance is not strongly correlated with the input variance for the current features. Modeling and classification results can be improved with additional features, specifically those capturing the changes through the differentiation process like dissolved oxygen in media, PH, and cell concentration, which is recommended for future studies.

6.2.2 Data-driven Model for Size Prediction of the PLGA Nano-particles

In Section 6.2.2, my research on predicting the size and polydispersity (PDI) for poly-lactic-co-glycolic acid (PLGA)-based nano-particles is discussed. PLGA nano-particles play a significant role in recent drug delivery methods, and control over the size and its uniformity is crucial for their reliable implementation. This study developed predictive models to predict the nanoparticle size and PDI using several different ML techniques. It utilized random forests, Gaussian process regression, support vector regression, multivariable adaptive regression splines, artificial neural networks, and extreme learning machines as modeling techniques. The results showed that the regression model trained using random forests was the most accurate, with root mean square error (RMSE) and mean absolute error (MAE) of 22.19 and 17.51 for predicting the size and RMSE and MAE of 0.041 and 0.032 for estimating the PDI of the particles. The feature importance analysis of the best model revealed that total sonication energy divided by the total emulsion volume and polyvinyl alcohol (PVA) to PLGA molar ratio were the most significant predictors for size. Considering new features in the analysis and training the models using data points covering wider ranges of the most important features are recommended for future studies. This work demonstrated promising performance from ML techniques to be implemented in these applications. In future directions, ML techniques are recommended to be utilized with other materials and processes to produce controllable and uniform size nanoparticles.

6.3 A Novel Approach for Building Surrogate Models of High-fidelity Stochastic Simulations

Chapter 5 presents my proposed novel approach for building surrogate models to accurately represent the outpus of high-fidelity simulation with uncertain parameters. High-fidelity simulations are computationally expensive to run. If they include uncertain parameters, their use in applications that require many simulation runs, such as sensitivity analysis, uncertainty quantification, and optimization, becomes computationally intractable. In this study, I proposed a new method to represent high-fidelity stochastic simulations with accurate cheap-to-evaluate surrogate models. The proposed method, PARameter as Input-variable (PARIN), is compared to three other existing methods: (1) setting the values of uncertain parameters to a fixed value (Fixed) and training a surrogate model, (2) selecting a subset of realizations of the uncertain parameters and building surrogate models corresponding to each of these realizations (PSet), and (3) stochastic kriging (SK). The comparison considered the accuracy and efficiency of the methods in predicting the mean and standard deviation and the similarity of the predicted output distribution to the original one using a set of test functions. The results revealed that PARIN was the most accurate approach with the lowest error in predicting mean and standard deviation for cases with varying input dimensions and uncertain parameters. However, PARIN's performance deteriorated for high dimensional functions for low computational budgets. The resulting output distributions from PARIN had the highest similarity to the original output distributions compared to PSet and SK. The similarity of the predicted distribution by PARIN dropped for functions with a large number of input dimensions as the computational budget value decreased.

In Section 5.2, PARIN is used for surrogate-based optimization, and the results are compared to PSet and SK approaches. The distance of optimum location estimated by PARIN

from the test function optimum was less than the PSet approach. The solution time for the optimization model with SK exceeded the maximum allowed time, suggesting how this method could not be a good candidate for surrogate-based optimization.

Future studies should investigate the impact of uncertain parameters with different distributions to evaluate the accuracy and robustness of PARIN. The spread of the data is another important factor in the performance of different approaches for building accurate surrogate models of simulations; hence considering different values of standard deviation for the uncertainty distribution should be investigated in the future. Moreover, more studies are required in terms of the evaluation of PARIN application in surrogate-based optimization. Additionally, PARIN could be used for applications like sensitivity analysis and uncertainty quantification in the future.

Chapter 7 – **References**

Abramowitz, M., Stegun, I.A., Romer, R.H., 1988. Handbook of mathematical functions with formulas, graphs, and mathematical tables.

Al, R., Behera, C.R., Gernaey, K. V., Sin, G., 2020. Stochastic simulation-based superstructure optimization framework for process synthesis and design under uncertainty. Comput. Chem. Eng. 143, 107118. https://doi.org/10.1016/j.compchemeng.2020.107118

Aleti, A., Trubiani, C., van Hoorn, A., Jamshidi, P., 2018. An efficient method for uncertainty propagation in robust software performance estimation. J. Syst. Softw. 138, 222–235. https://doi.org/10.1016/j.jss.2018.01.010

Alizadehdakhel, A., Rahimi, M., Alsairafi, A.A., 2010. CFD modeling of flow and heat transfer in a thermosyphon. Int. Commun. Heat Mass Transf. 37, 312–318. https://doi.org/10.1016/j.icheatmasstransfer.2009.09.002

Allen, M.S., Camberos, J.A., 2009. Comparison of uncertainty propagation/response surface techniques for two aeroelastic systems. Collect. Tech. Pap. - AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 1–19. https://doi.org/10.2514/6.2009-2269

Almany, L., Seliktar, D., Guyon, I., Elisseeff, A., Jović, A., Brkić, K., Bogunović, N., Das, S., Balafkan, N., Mostafavi, S., Schubert, M., Siller, R., Liang, K.X., Sullivan, G., Bindoff, L.A., Bonow, R.O., Mann, D.L., Zipes, D.P., Libby, P., Branco, M.A., Cotovio, J.P., Rodrigues, C.A. V, Vaz, S.H., Fernandes, T.G., Moreira, L.M., Cabral, J., Diogo, M.M., Burman, P., 2003. Transcriptomic analysis of 3D cardiac differentiation of human induced pluripotent stem cells reveals faster cardiomyocyte maturation compared to 2D culture. Sci. Rep. 10, 1–13.

Ankenman, B., Nelson, B.L., Staum, J., 2008. Stochastic kriging for simulation metamodeling. Proc. - Winter Simul. Conf. 362–370. https://doi.org/10.1109/WSC.2008.4736089

Anselmo, A.C., Mitragotri, S., 2019. Nanoparticles in the clinic: An update. Bioeng. Transl. Med. 4, 1–16. https://doi.org/10.1002/btm2.10143

Anthony, O., 2013. Polynomial Chaos : A Tutorial and Critique from a Statistician ' s Perspective 1–20.

Arakere, N.K., Pattabhiraman, S., Levesque, G., Kim, N.H., 2010. Uncertainty analysis for rolling contact fatigue failure probability of silicon nitride ball bearings. Int. J. Solids Struct. 47, 2543–2553. https://doi.org/10.1016/j.ijsolstr.2010.05.018

Balafkan, N., Mostafavi, S., Schubert, M., Siller, R., Liang, K.X., Sullivan, G., Bindoff, L.A., 2020. A method for differentiating human induced pluripotent stem cells toward functional cardiomyocytes in 96-well microplates. Sci. Rep. 10, 1–14.

Berkland, C., Kim, K. (Kevin), Pack, D.W., 2004. Precision Polymer Microparticles for Controlled-Release Drug Delivery. ACS Symp. Ser. 879, 197–213. https://doi.org/10.1021/bk-2004-0879.ch014

Blum, A.L., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artif. Intell. 97, 245–271. https://doi.org/https://doi.org/10.1016/S0004-3702(97)00063-5

Branco, M.A., Cotovio, J.P., Rodrigues, C.A. V, Vaz, S.H., Fernandes, T.G., Moreira, L.M., Cabral, J., Diogo, M.M., 2019. Transcriptomic analysis of 3D cardiac differentiation of human induced pluripotent stem cells reveals faster cardiomyocyte maturation compared to 2D culture. Sci. Rep. 9, 1–13.

Breiman, L., 2001. Random forests. Mach. Learn. 45, 5–32.

Breiman, L.E.O., 2001. Random Forests 5–32.

Burhenne, S., Jacob, D., Henze, G.P., 2011. Sampling based on sobol′ sequences for monte carlo techniques applied to building simulations. Proc. Build. Simul. 2011 12th Conf. Int. Build. Perform. Simul. Assoc. 1816–1823.

Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., 2019. Integrated process design, scheduling, and control using multiparametric programming. Comput. Chem. Eng. 125, 164–184. https://doi.org/10.1016/j.compchemeng.2019.03.004

Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Siirola, J.D., Watson, J.-P., Woodruff, D.L., 2021. Pyomo--optimization modeling in python, Third. ed. Springer Science \& Business Media.

Chang, S., Finklea, F., Williams, B., Hammons, H., Hodge, A., Scott, S., Lipke, E., 2020. Emulsion-based encapsulation of pluripotent stem cells in hydrogel microspheres for cardiac differentiation. Biotechnol. Prog. 36, e2986.

Chen, R.-C., Dewi, C., Huang, S.-W., Caraka, R.E., 2020. Selecting critical features for data classification based on machine learning methods. J. Big Data 7, 1–26.

Chen, X., Wasikowski, M., 2008. Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 124–132.

Crestaux, T., Le Maître, O., Martinez, J.M., 2009. Polynomial chaos expansion for sensitivity analysis. Reliab. Eng. Syst. Saf. 94, 1161–1172. https://doi.org/10.1016/j.ress.2008.10.008

Crombecq, K., Laermans, E., Dhaene, T., 2011. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. Eur. J. Oper. Res. 214, 683–696. https://doi.org/10.1016/j.ejor.2011.05.032

Crucho, C.I.C., Barros, M.T., 2017. Polymeric nanoparticles: A study on the preparation variables and characterization methods. Mater. Sci. Eng. C 80, 771–784. https://doi.org/10.1016/j.msec.2017.06.004

Cunha, A.B., Hou, J., Schuelke, C., 2019. Machine learning for stem cell differentiation and proliferation classification on electrical impedance spectroscopy. J. Electr. Bioimpedance 10, 124.

Dahlmann, J., Kensah, G., Kempf, H., Skvorc, D., Gawol, A., Elliott, D.A., Dräger, G., Zweigerdt, R., Martin, U., Gruh, I., 2013. The use of agarose microwells for scalable embryoid body formation and cardiac differentiation of human and murine pluripotent stem cells. Biomaterials 34, 2463–2471.

Das, S.K., Das, S.R.B.T.-I.C. on M.L., 2001. Wrappers and a boosting-based hybrid for feature selection.

di Sciascio, F., Amicarelli, A.N., 2008. Biomass estimation in batch biotechnological processes by Bayesian Gaussian process regression. Comput. Chem. Eng. 32, 3264–3273.

Drucker, H., Shahrary, B., Gibbon, D.C., 2002. Support vector machines: relevance feedback and information retrieval. Inf. Process. Manag. 38, 305–323.

Duffy, J., Liu, S., Moskowitz, H., Plante, R., Preckel, P. V., 1998. Assessing multivariate process/product yield via discrete point approximation. IIE Trans. (Institute Ind. Eng. 30,

535–543. https://doi.org/10.1080/07408179808966493

Elsabahy, M., Wooley, K.L., 2012. Design of polymeric nanoparticles for biomedical delivery applications. Chem. Soc. Rev. 41, 2707–2717. https://doi.org/10.1039/c2cs15327k

Fahmi, I., Cremaschi, S., 2016. Computational Experiments on Sampling Methods for Uncertainty Propagation and the Implications for Simulation-Based Optimization, in: Computer Aided Chemical Engineering. Elsevier, pp. 1779–1784.

Feinberg, J., Langtangen, H.P., 2015. Chaospy: An open source tool for designing methods of uncertainty quantification. J. Comput. Sci. 11, 46–57. https://doi.org/10.1016/j.jocs.2015.08.008

Finklea, F.B., Tian, Y., Kerscher, P., Seeto, W.J., Ellis, M.E., Lipke, E.A., 2021. Engineered cardiac tissue microsphere production through direct differentiation of hydrogel-encapsulated human pluripotent stem cells. Biomaterials 274, 120818.

Fonoudi, H., Ansari, H., Abbasalizadeh, S., Larijani, M.R., Kiani, S., Hashemizadeh, S., Zarchi, A.S., Bosman, A., Blue, G.M., Pahlavan, S., 2015. A universal and robust integrated platform for the scalable production of human cardiomyocytes from pluripotent stem cells. Stem Cells Transl. Med. 4, 1482–1494.

Fredenberg, S., Wahlgren, M., Reslow, M., Axelsson, A., 2011. The mechanisms of drug release in poly(lactic-co-glycolic acid)-based drug delivery systems - A review. Int. J. Pharm. 415, 34–52. https://doi.org/10.1016/j.ijpharm.2011.05.049

Friedman, J.H. (stanford U., 1991. Multivariate adaptive regression splines.

Fuoco, C., Salvatori, M.L., Biondo, A., Shapira-Schweitzer, K., Santoleri, S., Antonini, S.,

Bernardini, S., Tedesco, F.S., Cannata, S., Seliktar, D., 2012. Injectable polyethylene glycol-fibrinogen hydrogel adjuvant improves survival and differentiation of transplanted mesoangioblasts in acute and chronic skeletal-muscle degeneration. Skelet. Muscle 2, 1–14.

Gel, A., Garg, R., Tong, C., Shahnam, M., Guenther, C., 2013. Applying uncertainty quantification to multiphase flow computational fluid dynamics. Powder Technol. 242, 27–39. https://doi.org/10.1016/j.powtec.2013.01.045

Ghanem, R.G., Spanos, P.D., 1991. Spectral Stochastic Finite-Element Formulation for Reliability Analysis. J. Eng. Mech. 117, 2351–2372. https://doi.org/10.1061/(asce)0733-9399(1991)117:10(2351)

Grimmett, G., Stirzaker, D., 2001. Probability and random processes. Oxford university press.

Groen, E.A., Heijungs, R., Bokkers, E.A.M., de Boer, I.J.M., 2014. Methods for uncertainty propagation in life cycle assessment. Environ. Model. Softw. 62, 316–325. https://doi.org/10.1016/j.envsoft.2014.10.006

Gupta, R., Fletcher, D.F., Haynes, B.S., 2010. CFD modelling of flow and heat transfer in the Taylor flow regime. Chem. Eng. Sci. 65, 2094–2107. https://doi.org/10.1016/j.ces.2009.12.008

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Halayqa, M., Domańska, U., 2014. PLGA biodegradable nanoparticles containing perphenazine or chlorpromazine hydrochloride: Effect of formulation and release. Int. J. Mol. Sci. 15, 23909–23923. https://doi.org/10.3390/ijms151223909

Haleem, K., Gan, A., Lu, J., 2013. Using multivariate adaptive regression splines (MARS) to develop crash modification factors for urban freeway interchange influence areas. Accid. Anal. Prev. 55, 12–21. https://doi.org/10.1016/j.aap.2013.02.018

Hall, M.A., Smith, L.A., 1999. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper, in: FLAIRS Conference. pp. 235–239.

Halloin, C., Schwanke, K., Löbel, W., Franke, A., Szepes, M., Biswanath, S., Wunderlich, S., Merkert, S., Weber, N., Osten, F., 2019. Continuous WNT control enables advanced hPSC cardiac processing and prognostic surface marker identification in chemically defined suspension culture. Stem cell reports 13, 366–379.

Halton, J.H., 1960. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math. 2, 84–90. https://doi.org/10.1007/BF01386213

Hamad, S., Derichsweiler, D., Papadopoulos, S., Nguemo, F., Šarić, T., Sachinidis, A., Brockmeier, K., Hescheler, J., Boukens, B.J., Pfannkuche, K., 2019. Generation of human induced pluripotent stem cell-derived cardiomyocytes in 2D monolayer and scalable 3D suspension bioreactor cultures with reduced batch-to-batch variations. Theranostics 9, 7222.

Hannah, L.A., 2004. Stochastic Optimization. Encycl. Actuar. Sci. 1–20. https://doi.org/10.1002/9780470012505.tas034

Hansen, lars peter, 1982. Large Sample Properties of Generalized Method of Moments Estimators Author(s): Lars Peter Hansen Source: Econometrica 50, 1029–1054.

Hart, W.E., Watson, J.-P., Woodruff, D.L., 2011. Pyomo: modeling and solving mathematical programs in Python. Math. Program. Comput. 3, 219–260.

Haykin, S.O., 2009. Neural networks and learning machines, 3rd ed. Pearson Education Inc., Newjersey.

Hernández-Giottonini, K.Y., Rodríguez-Córdova, R.J., Gutiérrez-Valenzuela, C.A., Peñuñuri-Miranda, O., Zavala-Rivera, P., Guerrero-Germán, P., Lucero-Acuña, A., 2020. PLGA nanoparticle preparations by emulsification and nanoprecipitation techniques: Effects of formulation parameters. RSC Adv. 10, 4218–4231. https://doi.org/10.1039/c9ra10857b

Heylman, C., Datta, R., Sobrino, A., George, S., Gratton, E., 2015. Supervised machine learning for classification of the electrophysiological effects of chronotropic drugs on human induced pluripotent stem cell-derived cardiomyocytes. PLoS One 10, e0144572.

Hotelling, H., 1933. Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. 24, 417.

Hou, T., Nuyens, D., Roels, S., Janssen, H., 2019. Quasi-Monte Carlo based uncertainty analysis: Sampling efficiency and error estimation in engineering applications. Reliab. Eng. Syst. Saf. 191, 106549. https://doi.org/10.1016/j.ress.2019.106549

Huang, G.-B. Bin, Zhu, Q.-Y.Y., Siew, C.-K.K., Huang Ã, G.-B., Zhu, Q.-Y.Y., Siew, C.-K.K., Huang, G.-B. Bin, Zhu, Q.-Y.Y., Siew, C.-K.K., 2006. Extreme learning Mach. Neurocomputing 70, 489–501. https://doi.org/10.1016/j.neucom.2005.12.126

Hüllen, G., Zhai, J., Kim, S.H., Sinha, A., Realff, M.J., Boukouvala, F., 2019. Managing Uncertainty in Data-Driven Simulation-Based Optimization. Comput. Chem. Eng. 106519. https://doi.org/10.1016/j.compchemeng.2019.106519

Hunt, M., Haley, B., McLennan, M., Koslowski, M., Murthy, J., Strachan, A., 2015. PUQ: A code

for non-intrusive uncertainty propagation in computer simulations. Comput. Phys. Commun. 194, 97–107. https://doi.org/10.1016/j.cpc.2015.04.011

Hwang, H., Liu, R., Maxwell, J.T., Yang, J., Xu, C., 2020. Machine learning identifies abnormal Ca 2+ transients in human induced pluripotent stem cell-derived cardiomyocytes. Sci. Rep. 10, 1–10.

Jia, X.Y., Jiang, C., Fu, C.M., Ni, B.Y., Wang, C.S., Ping, M.H., 2019. Uncertainty propagation analysis by an extended sparse grid technique. Front. Mech. Eng. 14, 33–46. https://doi.org/10.1007/s11465-018-0514-x

Jiang, P., Zhou, Q., Shao, X., 2020. Surrogate model-based engineering design and optimization. Springer.

Joe, S., Kuo, F.Y., 2008. Notes on generating Sobol sequences Gray code implementation. English 2–4.

Jović, A., Brkić, K., Bogunović, N., 2015. A review of feature selection methods with applications, in: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 1200–1205.

Kempf, H., Andree, B., Zweigerdt, R., 2016a. Large-scale production of human pluripotent stem cell derived cardiomyocytes. Adv. Drug Deliv. Rev. 96, 18–30. https://doi.org/10.1016/j.addr.2015.11.016

Kempf, H., Olmer, R., Haase, A., Franke, A., Bolesani, E., Schwanke, K., Robles-Diaz, D., Coffee, M., Göhring, G., Dräger, G., 2016b. Bulk cell density and Wnt/TGFbeta signalling regulate mesendodermal patterning of human pluripotent stem cells. Nat. Commun. 7, 1–13.

Kempf, H., Olmer, R., Kropp, C., Rückert, M., Jara-Avaca, M., Robles-Diaz, D., Franke, A., Elliott, D.A., Wojciechowski, D., Fischer, M., 2014. Controlling expansion and cardiomyogenic differentiation of human pluripotent stem cells in scalable suspension culture. Stem cell reports 3, 1132–1146.

Kerscher, P., Kaczmarek, J.A., Head, S.E., Ellis, M.E., Seeto, W.J., Kim, J., Bhattacharya, S., Suppiramaniam, V., Lipke, E.A., 2016. Direct production of human cardiac tissues by pluripotent stem cell encapsulation in gelatin methacryloyl. ACS Biomater. Sci. Eng. 3, 1499–1509.

Kim, Y.J., 2016. Comparative study of surrogate models for uncertainty quantification of building energy model: Gaussian Process Emulator vs. Polynomial Chaos Expansion. Energy Build. 133, 46–58. https://doi.org/10.1016/j.enbuild.2016.09.032

Kittler, J., 1978. Feature set search algorithms. Pattern Recognit. signal Process.

Klavetter, K., Posluszny, D., Warr, J., Cremaschi, S., Sarica, C., Subramani, H.J., 2012. Uncertainty analysis of multiphase flow models: A comparison of three propagation approaches. BHR Gr. - 8th North Am. Conf. Multiph. Technol. 259–271.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97, 273–324.

Kropp, C., Massai, D., Zweigerdt, R., 2017. Progress and challenges in large-scale expansion of human pluripotent stem cells. Process Biochem. 59, 244–254.

Lee, E.K., Tran, D.D., Keung, W., Chan, P., Wong, G., Chan, C.W., Costa, K.D., Li, R.A., Khine, M., 2017. Machine learning of human pluripotent stem cell-derived engineered cardiac tissue contractility for automated drug classification. Stem cell reports 9, 1560–1572.

Lee, S.H., Chen, W., 2009. A comparative study of uncertainty propagation methods for black-box-type problems. Struct. Multidiscip. Optim. 37, 239–253. https://doi.org/10.1007/s00158-008-0234-7

Li, Y., Wu, F.-X., Ngom, A., 2018. A review on machine learning principles for multi-view biological data integration. Brief. Bioinform. 19, 325–340.

Lian, X., Zhang, J., Azarin, S.M., Zhu, K., Hazeltine, L.B., Bao, X., Hsiao, C., Kamp, T.J., Palecek, S.P., 2013. Directed cardiomyocyte differentiation from human pluripotent stem cells by modulating Wnt/β-catenin signaling under fully defined conditions. Nat. Protoc. 8, 162–175.

Liu, B., Koziel, S., Zhang, Q., 2016. A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems. J. Comput. Sci. 12, 28–37. https://doi.org/10.1016/j.jocs.2015.11.004

Liu, P., Kaplan, A., Yuan, B., Hanna, J.H., Lupski, J.R., Reiner, O., 2014. Passage number is a major contributor to genomic structural variations in mouse iPSCs. Stem Cells 32, 2657–2667.

Liu, Y., Gupta, H. V., 2007. Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework. Water Resour. Res. 43, 1–18. https://doi.org/10.1029/2006WR005756

Livescu, S., Durlofsky, L.J., Aziz, K., Ginestra, J.C., 2010. A fully-coupled thermal multiphase wellbore flow model for use in reservoir simulation. J. Pet. Sci. Eng. 71, 138–146. https://doi.org/10.1016/j.petrol.2009.11.022

Luo, Y. zhong, Yang, Z., 2017. A review of uncertainty propagation in orbital mechanics. Prog.

Aerosp. Sci. 89, 23–39. https://doi.org/10.1016/j.paerosci.2016.12.002

MacKay, D.J.C., 1994. Bayesian nonlinear modeling for the prediction competition. ASHRAE Trans. 100, 1053–1062.

Makadia, H.K., Siegel, S.J., 2011. Poly Lactic-co-Glycolic Acid (PLGA) as biodegradable controlled drug delivery carrier. Polymers (Basel). 3, 1377–1397. https://doi.org/10.3390/polym3031377

Matthews, B.W., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta (BBA)-Protein Struct. 405, 442–451.

McKay, M.D., Beckman, R.J., Conover, W.J., 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21, 239–245.

Mehrian, M., Lambrechts, T., Marechal, M., Luyten, F.P., Papantoniou, I., Geris, L., 2020. Predicting in vitro human mesenchymal stromal cell expansion based on individual donor characteristics using machine learning. Cytotherapy 22, 82–90.

Miller, D.C., Ng, B., Eslick, J., Tong, C., Chen, Y., 2014. Advanced Computational Tools for Optimization and Uncertainty Quantification of Carbon Capture Processes, in: Eden, M.R., Siirola, J.D., Towler, G.P. (Eds.), Proceedings of the 8th International Conference on Foundations of Computer-Aided Process Design, Computer Aided Chemical Engineering. Elsevier, pp. 202–211. https://doi.org/https://doi.org/10.1016/B978-0-444-63433-7.50021-3

Misener, R., Floudas, C.A., 2014. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. J. Glob. Optim. 59, 503–526.

https://doi.org/10.1007/s10898-014-0166-2

Murcia, J.P., Réthoré, P.E., Dimitrov, N., Natarajan, A., Sørensen, J.D., Graf, P., Kim, T., 2018. Uncertainty propagation through an aeroelastic wind turbine model using polynomial surrogates. Renew. Energy 119, 910–922. https://doi.org/10.1016/j.renene.2017.07.070

Nobbmann, U., 2015. PDI From an Individual Peak in DLS [WWW Document]. Malvern Panalytical.

Operti, M.C., Bernhardt, A., Grimm, S., Engel, A., Figdor, C.G., Tagit, O., 2021. PLGA-based nanomedicines manufacturing: Technologies overview and challenges in industrial scale-up. Int. J. Pharm. 605, 120807. https://doi.org/10.1016/j.ijpharm.2021.120807

Orita, K., Sawada, K., Matsumoto, N., Ikegaya, Y., 2020. Machine-learning-based quality control of contractility of cultured human-induced pluripotent stem-cell-derived cardiomyocytes. Biochem. Biophys. Res. Commun. 526, 751–755.

Paananen, T., Piironen, J., Andersen, M.R., Vehtari, A., 2019. Variable selection for Gaussian processes via sensitivity analysis of the posterior predictive distribution, in: The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, pp. 1743–1752.

Padulo, M., Campobasso, M.S., Guenov, M.D., 2007. Comparative analysis of uncertainty propagation methods for robust Engineering Design. Proc. ICED 2007, 16th Int. Conf. Eng. Des. DS 42, 1–12.

Park, J.S., Chu, J.S., Tsou, A.D., Diop, R., Tang, Z., Wang, A., Li, S., 2011. The effect of matrix stiffness on the differentiation of mesenchymal stem cells in response to TGF-β. Biomaterials 32, 3921–3930. https://doi.org/10.1016/j.biomaterials.2011.02.019

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Peherstorfer, B., Kramer, B., Willcox, K., 2017. Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models. J. Comput. Phys. 341, 61–75. https://doi.org/10.1016/j.jcp.2017.04.012

Potdar, K., S., T., D., C., 2017. A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers. Int. J. Comput. Appl. 175, 7–9. https://doi.org/10.5120/ijca2017915495

Pudil, P., Novovičová, J., Kittler, J., 1994. Floating search methods in feature selection. Pattern Recognit. Lett. 15, 1119–1125.

Quirante, N., Javaloyes, J., Ruiz-Femenia, R., Caballero, J.A., 2015. Optimization of chemical processes using surrogate models based on a kriging interpolation, in: Computer Aided Chemical Engineering. Elsevier, pp. 179–184.

Rahman, S., Xu, H., 2004. A univariate dimension-reduction method for multi-dimensional integration in stochastic mechanics. Probabilistic Eng. Mech. 19, 393–408. https://doi.org/10.1016/j.probengmech.2004.04.003

Rajabi, M.M., 2019. Review and comparison of two meta-model-based uncertainty propagation analysis methods in groundwater applications: polynomial chaos expansion and Gaussian process emulation. Stoch. Environ. Res. Risk Assess. 33, 607–631. https://doi.org/10.1007/s00477-018-1637-7

Rao, J.P., Geckeler, K.E., 2011. Polymer nanoparticles: Preparation techniques and size-control parameters. Prog. Polym. Sci. 36, 887–913. https://doi.org/10.1016/j.progpolymsci.2011.01.001

Rasmussen Christopher K. I. Williams., C.E., 2005. Gaussian Processes for Machine Learning. Adapt. Comput. Mach. Learn. Ser.

Remeseiro, B., Bolon-Canedo, V., 2019. A review of feature selection methods in medical applications. Comput. Biol. Med. 112, 25–29. https://doi.org/10.1016/j.compbiomed.2019.103375

Safta, C., Chen, R.L.Y., Najm, H.N., Pinar, A., Watson, J.P., 2017. Efficient Uncertainty Quantification in Stochastic Economic Dispatch. IEEE Trans. Power Syst. 32, 2535–2546. https://doi.org/10.1109/TPWRS.2016.2615334

Sahin, A., Esendagli, G., Yerlikaya, F., Caban-Toktas, S., Yoyen-Ermis, D., Horzum, U., Aktas, Y., Khan, M., Couvreur, P., Capan, Y., 2017. A small variation in average particle size of PLGA nanoparticles prepared by nanoprecipitation leads to considerable change in nanoparticles' characteristics and efficacy of intracellular delivery. Artif. cells, nanomedicine, Biotechnol. 45, 1657–1664. https://doi.org/10.1080/21691401.2016.1276924

Salcedo-Sanz, S., Pastor-Sánchez, A., Prieto, L., Blanco-Aguilera, A., García-Herrera, R., 2014. Feature selection in wind speed prediction systems based on a hybrid coral reefs optimization–Extreme learning machine approach. Energy Convers. Manag. 87, 10–18.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput. Phys. Commun. 181, 259–270. https://doi.org/10.1016/j.cpc.2009.09.018

Schmidt-Heck, W., Zeilinger, K., Pless, G., Gerlach, J.C., Pfaff, M., Guthke, R., 2005. Prediction of the performance of human liver cell bioreactors by donor organ data, in: International Symposium on Biological and Medical Data Analysis. Springer, pp. 109–119.

Seeto, W.J., Tian, Y., Pradhan, S., Kerscher, P., Lipke, E.A., 2019. Photocrosslinked Microspheres: Rapid Production of Cell-Laden Microspheres Using a Flexible Microfluidic Encapsulation Platform (Small 47/2019). Small 15, 1970254. https://doi.org/10.1002/smll.201970254

Shekar, M.C., Chary, B.B., Srinivas, S., Kumar, B.R., Mahendrakar, M.D., Varma, M.V.K., 2011. Effect of ultrasonication on stability of oil in water emulsions. Int. J. Drug Deliv. 3, 133–140. https://doi.org/10.5138/ijdd.2010.0975.0215.03063

Smolyak, S.A., 1963. Quadrature and interpolation formulas for tensor products of certain classes of functions, in: Doklady Akademii Nauk. pp. 1042–1045.

Sobol', I.M., 1967. On the distribution of points in a cube and the approximate evaluation of integrals. Zhurnal Vychislitel'noi Mat. i Mat. Fiz. 7, 784–802.

Sofi, A., Muscolino, G., Giunta, F., 2020. Propagation of uncertain structural properties described by imprecise Probability Density Functions via response surface method. Probabilistic Eng. Mech. 60, 103020. https://doi.org/10.1016/j.probengmech.2020.103020

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 45, 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Song, X., Zhao, Y., Hou, S., Xu, F., Zhao, R., He, J., Cai, Z., Li, Y., Chen, Q., 2008. Dual agents

loaded PLGA nanoparticles: Systematic study of particle size and drug entrapment efficiency. Eur. J. Pharm. Biopharm. 69, 445–453. https://doi.org/https://doi.org/10.1016/j.ejpb.2008.01.013

Soper, H.E., Young, A.W., Cave, B.M., Lee, A., Pearson, K., 1917. On the distribution of the correlation coefficient in small samples. Appendix II to the papers of" Student" and RA Fisher. Biometrika 11, 328–413.

Staum, J., 2009. Better Simulation Metamodeling: The why, what, and how of Stochastic Kriging 119–133.

Sun, C., Yan, H., Qiu, X., Huang, X., 2018. Gaussian Word Embedding with a Wasserstein Distance Loss.

Surjanovic, S., Bingham, D., 2013. Virtual Library of Simulation Experiments: Test Functions and Datasets.

Takahashi, K., Yamanaka, S., 2006. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. Cell 126, 663–676.

Tardioli, C., Kubicek, M., Vasile, M., Minisci, E., Riccardi, A., 2016. Comparison of non-intrusive approaches to uncertainty propagation in orbital mechanics. Adv. Astronaut. Sci. 156, 3979–3992.

Teles, D., Kim, Y., Ronaldson-Bouchard, K., Vunjak-Novakovic, G., 2021. Machine Learning Techniques to Classify Healthy and Diseased Cardiomyocytes by Contractility Profile. ACS Biomater. Sci. Eng.

Teng, E.L., Engler, A.J., 2019. Mechanical influences on cardiovascular differentiation and

disease modeling. Exp. Cell Res. 377, 103–108. https://doi.org/10.1016/j.yexcr.2019.02.019

Tian, Y., Lipke, E.A., 2020. Microfluidic Production of Cell-Laden Microspheroidal Hydrogels with Different Geometric Shapes. ACS Biomater. Sci. Eng. 6, 6435–6444.

Tripathy, R., Bilionis, I., Gonzalez, M., 2016. Gaussian processes with built-in dimensionality reduction: Applications to high-dimensional uncertainty propagation. J. Comput. Phys. 321, 191–223. https://doi.org/10.1016/j.jcp.2016.05.039

Villani, C., 2009. Optimal transport: old and new. Springer.

Volpato, V., Webber, C., 2020. Addressing variability in iPSC-derived models of human disease: guidelines to promote reproducibility. Dis. Model. Mech. 13, dmm042317.

Wah, Y.B., Ibrahim, N., Hamid, H.A., Abdul-Rahman, S., Fong, S., 2018. Feature Selection Methods: Case of Filter and Wrapper Approaches for Maximising Classification Accuracy. Pertanika J. Sci. Technol. 26.

Wang, H., Sheen, D.A., 2015. Combustion kinetic model uncertainty quantification, propagation and minimization. Prog. Energy Combust. Sci. 47, 1–31. https://doi.org/10.1016/j.pecs.2014.10.002

Wang, L., Wang, Y., Chang, Q., 2016. Feature selection methods for big data bioinformatics: a survey from the search perspective. Methods 111, 21–31.

Wang, W., Chen, X., 2016. The effects of estimation of heteroscedasticity on stochastic kriging, in: Proceedings of the 2016 Winter Simulation Conference. pp. 326–337.

Wang, Y., Qin, B., Xia, G., Choi, S.H., 2021. FDA's Poly (Lactic-Co-Glycolic Acid) Research Program and Regulatory Outcomes. AAPS J. 23, 1–7. https://doi.org/10.1208/s12248-021-

00611-y

Wiener, N., 1938. The homogeneous chaos. Am. J. Math. 60, 897–936.

Williams, B., Cremaschi, S., 2021. Selection of Surrogate Modeling Techniques for Surface Approximation and Surrogate-Based Optimization. Chem. Eng. Res. Des. https://doi.org/https://doi.org/10.1016/j.cherd.2021.03.028

Williams, B., Löbel, W., Finklea, F., Halloin, C., Ritzenhoff, K., Manstein, F., Mohammadi, S., Hashemi, M., Zweigerdt, R., Lipke, E., Cremaschi, S., 2020. Prediction of Human Induced Pluripotent Stem Cell Cardiac Differentiation Outcome by Multifactorial Process Modeling. Front. Bioeng. Biotechnol. 8. https://doi.org/10.3389/fbioe.2020.00851

Williams, C.K.I., Rasmussen, C.E., 2006. Gaussian processes for machine learning. MIT press Cambridge, MA.

Williams, C.K.I., Rasmussen, C.E., 1996. Gaussian Processes for Regression Advances in Neural Information Processing Systems 8. MIT Press 514–520.

Wong, T.-T., Luk, W.-S., Heng, P.-A., 2005. Sampling with Hammersley and Halton Points. Graph. Tools---The jgt Ed. Choice 255–270. https://doi.org/10.1201/b10628-32

Wong, T.T., 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognit. 48, 2839–2846. https://doi.org/10.1016/j.patcog.2015.03.009

Wu, S.G., Wang, Y., Jiang, W., Oyetunde, T., Yao, R., Zhang, X., Shimizu, K., Tang, Y.J., Bao, F.S., 2016. Rapid prediction of bacterial heterotrophic fluxomics using machine learning and constraint programming. PLoS Comput. Biol. 12, e1004838.

Xiong, F., Greene, S., Chen, W., Xiong, Y., Yang, S., 2010. A new sparse grid based method for uncertainty propagation. Struct. Multidiscip. Optim. 41, 335–349. https://doi.org/10.1007/s00158-009-0441-x

Xu, C., Jackson, S.A., 2019. Machine learning and complex biological data.

Yang, S., Xiong, F., Wang, F., 2017. Polynomial Chaos Expansion for Probabilistic Uncertainty Propagation. Uncertain. Quantif. Model Calibration. https://doi.org/10.5772/intechopen.68484

Young, J.L., Engler, A.J., 2011. Hydrogels with time-dependent material properties enhance cardiomyocyte differentiation in vitro. Biomaterials 32, 1002–1009. https://doi.org/10.1016/j.biomaterials.2010.10.020

Zahedi, P., Zhang, J., Arabnejad, H., McLaury, B.S., Shirazi, S.A., 2017. CFD simulation of multiphase flows and erosion predictions under annular flow and low liquid loading conditions. Wear 376–377, 1260–1270. https://doi.org/10.1016/j.wear.2017.01.111

Zhao, M., Tang, Y., Zhou, Y., Zhang, J., 2019. Deciphering role of Wnt signalling in cardiac mesoderm and cardiomyocyte differentiation from human iPSCs: Four-dimensional control of Wnt pathway for hiPSC-CMs differentiation. Sci. Rep. 9, 1–15.

Zhao, Y., Rafatian, N., Wang, E.Y., Wu, Q., Lai, B.F.L., Lu, R.X., Savoji, H., Radisic, M., 2020. Towards chamber specific heart-on-a-chip for drug testing applications. Adv. Drug Deliv. Rev. 165, 60–76.

Zweigerdt, R., Olmer, R., Singh, H., Haverich, A., Martin, U., 2011. Scalable expansion of human pluripotent stem cells in suspension culture. Nat. Protoc. 6, 689–700.

Chapter 8 – **Appendix 1**

Table A.1.1. List of all the functions used in the uncertainty propagation study

Function	**Formulation**	**Parameters**
Power	$y(x) = x^r$	r ∈ {1,2,3,4,5}
Hougen-Watson	$y(\mathbf{x}) = \dfrac{b_1 x_2 - \dfrac{x_3}{b_5}}{1 + b_2 x_1 + b_3 x_2 + b_4 x_3}$	$b_1 = 0.8291$ $b_2 = 0.6162$ $b_3 = 0.1829$ $b_4 = 0.0218$ $b_5 = 0.0624$
Kirby	$y(\mathbf{x}) = \dfrac{b_1 + b_2 x_1 + b_3 x_1^{\,2}}{1 + b_4 x_1 + b_5 x_1^{\,2}}$	$b_1 = 0.9756$ $b_2 = 0.2811$ $b_3 = 0.9424$ $b_4 = 0.0772$ $b_5 = 0.8096$
Hahn	$y(\mathbf{x}) = \dfrac{b_1 + b_2 x_1 + b_3 x_1^{\,2} + b_4 x_1^{\,3}}{1 + b_5 x_1 + b_6 x_1^{\,2} + b_7 x_1^{\,3}}$	$b_1 = 0.8125$ $b_2 = 0.6340$ $b_3 = 0.9300$ $b_4 = 0.2442$ $b_5 = 0.1146$ $b_6 = 0.6329$ $b_7 = 0.2985$
MGH-09	$y(\mathbf{x}) = \dfrac{b_1 (x_1^{\,2} + b_2 x_1)}{x_1^{\,2} + b_3 x_1 + b_4}$	$b_1 = 0.5379$ $b_2 = 0.9537$ $b_3 = 0.1418$ $b_4 = 0.2027$

Table A1.1. List of all the functions used in the uncertainty propagation study (cont'd)

Function	Formulation	Parameters
CPC-X-4	$y(\mathbf{x}) = \frac{b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4}{1 + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4}$	$b_0 = 0.8223$ $b_1 = 0.6608$ $b_2 = 0.7110$ $b_3 = 0.1908$ $b_4 = 0.5105$ $a_1 = 0.8455$ $a_2 = 0.2007$ $a_3 = 0.6728$ $a_4 = 0.0437$
CPC-X-7	$y(\mathbf{x}) = \frac{b_1 + b_2x_1 + b_3x_2 + b_4x_1x_2}{1 + b_5x_1 + b_6x_2 + b_7x_1x_2}$	$b_1 = 0.5630$ $b_2 = 0.0513$ $b_3 = 0.4498$ $b_4 = 0.7311$ $b_5 = 0.3501$ $b_6 = 0.1530$ $b_7 = 0.8833$
CPC-X-1	$y(\mathbf{x}) = \frac{1}{b_1 + b_2x_1^{\,b_3}} + b_4x_1^{\,b_5}$	$b_1 = 0.9856$ $b_2 = 0.6216$ $b_3 = 1$ $b_4 = 0.9602$ $b_5 = 3$
CPC-X-5	$y(\mathbf{x}) = b_1 + b_2x_1^{\,b_3} + b_4x_2^{\,b_5} + b_6x_1^{\,b_7}x_2^{\,b_8}$	$b_1 = 0.3228$ $b_2 = 0.4029$ $b_3 = 2$ $b_4 = 2574$ $b_5 = 3$ $b_6 = 0.1821$ $b_7 = 3$ $b_8 = 3$

Table A1.1. List of all the functions used in the uncertainty propagation study (cont'd)

Function	Formulation	Parameters
CPC-X-6	$y(\mathbf{x}) = b_1 + b_2 x_1^{b_3} + b_4 x_1^{b_5} + b_6 x_1^{b_7}$	$b_1 = 0.9109$ $b_2 = 0.7658$ $b_3 = 1$ $b_4 = 0.5906$ $b_5 = 2$ $b_6 = 0.6277$ $b_7 = 3$
CPC-X-8	$y(\mathbf{x}) = \frac{b_1}{(b_2 + x_1)(1 + b_3 x_2)(x_3 - b_4)^2} + b_5 x_3^{b_6}$	$b_1 = 0.$ $b_2 = 0.$ $b_3 = 0.$ $b_4 = 0.$ $b_5 = 0.$ $b_6 =$
CPC-X-9	$y(\mathbf{x}) = b_1 \exp(b_2 \mid x_1 + b_3 \mid^{b_4})$	$b_1 = 0.4143$ $b_2 = 0.9133$ $b_3 = 0.8557$ $b_4 = 0.2248$
Exponential	$y(\boldsymbol{x}) = \exp(x)$	
Linear Exponential	$y(\boldsymbol{x}) = \exp(x) + x^r$	r ∈ {1,2,3,4,5}
Nonlinear Exponential	$y(\boldsymbol{x}) = x\ exp(x)$	
Ackley	$y(\boldsymbol{x}) = -a\ exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$	$a = 20$ $b = 0.2$ $c = 2\pi$
Borehole	$y(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{ln\left(r/r_w\right)\left(1 + \frac{2LT_u}{ln\left(r/r_w\right) r_w^2 K_w} + \frac{T_u}{T_l}\right)}$	

Table A1.1. List of all the functions used in the uncertainty propagation study (cont'd)

Function	Formulation	Parameters
Short column	$y(\boldsymbol{x}) = 1 - \frac{4M}{bh^2Y} - \frac{P^2}{b^2h^2Y^2}$	$b = 5$ $h = 15$
Steel column	$y(\boldsymbol{x}) = F_s - P\left[\frac{1}{2BD} + \frac{F_0 E_b}{BDH(E_b - P)}\right], where$ $P = P_1 + P_2 + P_3$ $E_b = \frac{\pi^2 EBDH^2}{2L^2}$	
G	$y(\boldsymbol{x}) = \prod_{i=1}^{d} \frac{\lvert 4x_i - 2\rvert + a_i}{1 + a_i}$	$a_i = \frac{i-2}{2}$ $i \in \{1, \ldots, d\}$
Ishigami	$y(\boldsymbol{x}) = \sin(x_1) + a\sin^2(x_2) + bx_3^4 \sin(x_1)$	$a = 7$ $b = 0.1$
Oakley & O'Hagan 1D	$y(\boldsymbol{x}) = 5 + x + \cos(x)$	
Oakley & O'Hagan 2D	$y(\boldsymbol{x}) = 5 + x_1 + x_2 + 2\cos(x_1) + 2\sin(x_1)$	
Webster	$y(\boldsymbol{x}) = A^2 + B^3$	
Lognormal ration	$y(\boldsymbol{x}) = \frac{x_1}{x_2}$	

Table A1.2. Distribution of inputs for each of the functions

Function	Input distributions	Function	Input distributions
Power	U (0, 10)	CPC-X-8	U (0, 10)
	N (5, 3)		N (5, 3)
	L (1.5, 0.37)		L (1.5, 0.37)
Hougen-Watson	U (0, 10)	CPC-X-9	U (0, 10)
	L (1.5, 0.37)		N (5, 3)
Kirby	U (0, 10)		L (1.5, 0.37)
	L (1.5, 0.37)	Exponential	U (0, 10)
Hahn	U (0, 10)	Linear Exponential	U (0, 10)
	L (1.5, 0.37)	Nonlinear Exponential	U (0, 10)
MGH-09	U (0, 10)	Ackley	U (0, 10)
	L (1.5, 0.37)		N (5, 3)
CPC-X-4	U (0, 10)		L (1.5, 0.37)
	L (1.5, 0.37)	G	U (0, 1)
CPC-X-7	U (0, 10)	Ishigami	U ($-\pi$, π)
	L (1.5, 0.37)	Oakley & O'Hagan 1D	N (0, 4)
CPC-X-1	U (0, 10)	Oakley & O'Hagan 2D	U (-0.01, 0.01)
CPC-X-5	U (0, 10)	Webster	$x_1 \sim U(1,10)$ $x_2 \sim N(2,1)$
	N (5, 3)	Lognormal ration	L (1, 0.5), Correlation coeff. = 0.3
CPC-X-6	U (0, 10)		
	N (5, 3)		
	L (1.5, 0.37)		

Table A1.3. List of functions used in each case study

Function	Case1		Case2	Case3		Case 5	Case 6
	1.1	1.2		3.1	3.2		
Power	✓	✓		✓			✓
Hougen-Watson							✓
Kirby	✓			✓			✓
Hahn	✓			✓			✓
MGH-09	✓			✓			✓
CPC-X-4							✓
CPC-X-7							✓
CPC-X-1	✓						✓
CPC-X-5							✓
CPC-X-6	✓			✓			✓
CPC-X-8							✓
CPC-X-9	✓			✓			✓
Exponential	✓						✓
Linear Exponential	✓						✓
Nonlinear Exponential	✓						✓
Ackley	✓		✓	✓	✓		✓
Borehole						✓	✓
Short column							✓
Steel column						✓	✓
G	✓		✓				✓
Ishigami							✓
Oakley & O'Hagan 1D							✓
Oakley & O'Hagan 2D							✓
Webster							✓
Lognormal ration							✓

Chapter 9 – **Appendix 2**

9.1 Principal Component Analysis Results

Table A2.1. The ratios of the variance explained by each PC

PC1	PC2	PC3	PC4	PC5	PC6
0.322	0.272	0.116	0.095	0.064	0.045
PC7	**PC8**	**PC9**	**PC10**	**PC11**	**PC12**	**PC13**
0.035	0.022	0.019	0.005	0.003	0.001	0.0000

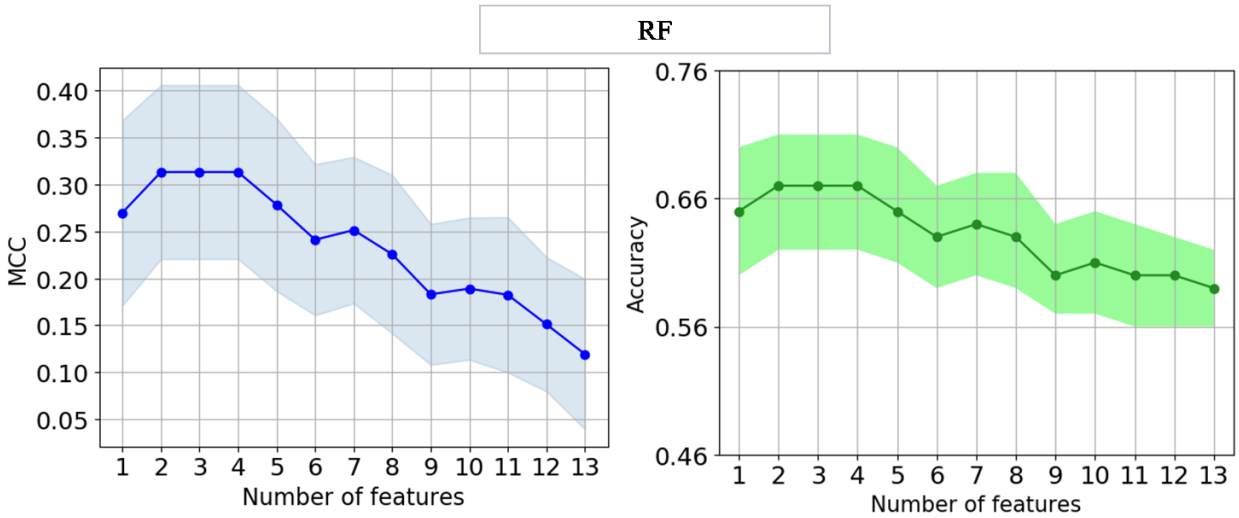
Table A2.2. The coefficients of each variable in every PC

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Number of cells	0.1	-0.16	-0.65	-0.12	-0.02	0.35	-0.38	-0.31	0.41	0.01	0.01	0.01	0
Passage post-freeze	-0.27	-0.03	-0.39	-0.35	-0.25	-0.5	-0.29	0.48	-0.16	0.03	0.05	-0.02	0
CHIR concentration	-0.32	0.33	0.07	-0.13	0.24	0.25	0.06	0.25	0.31	0.26	0.64	0.08	0
Cell concentration	-0.07	-0.23	-0.53	0.16	0.47	0.1	0.48	0.14	-0.39	0.04	0.03	0.01	0
PEG-fb concentration	-0.18	-0.06	0.05	0.66	0.4	-0.21	-0.57	0.07	0.03	0.01	0.01	0	0
Microspheroid Size	0.22	0.41	-0.11	0.13	-0.19	0.24	-0.23	-0.06	-0.51	0.29	0.15	-0.49	0
Axial Ratio	0.3	0.23	0.02	-0.32	0.55	-0.22	0	0.13	0.29	0.13	-0.33	-0.43	0
S/V/CHIR	0.2	-0.43	0.05	0	-0.02	-0.45	0.12	-0.35	0.07	0.2	0.55	-0.3	0
CHIR/S	-0.39	-0.23	0.16	-0.23	0.13	0.25	-0.12	-0.04	-0.12	-0.58	0.1	-0.52	0
CHIR/S/V	-0.23	-0.32	0.25	-0.38	0.22	0.15	-0.26	-0.21	-0.31	0.54	-0.18	0.2	0
S/CHIR	0.37	0.22	0	-0.27	0.31	-0.13	-0.23	-0.18	-0.32	-0.41	0.31	0.42	0
Media1	0.36	-0.32	0.12	0.01	-0.03	0.22	-0.12	0.43	0.01	0	0.09	0.01	-0.71
Media2	-0.36	0.32	-0.12	-0.01	0.03	-0.22	0.12	-0.43	-0.01	0	-0.09	-0.01	-0.71

9.2 Wrapper Method Feature Selection Results

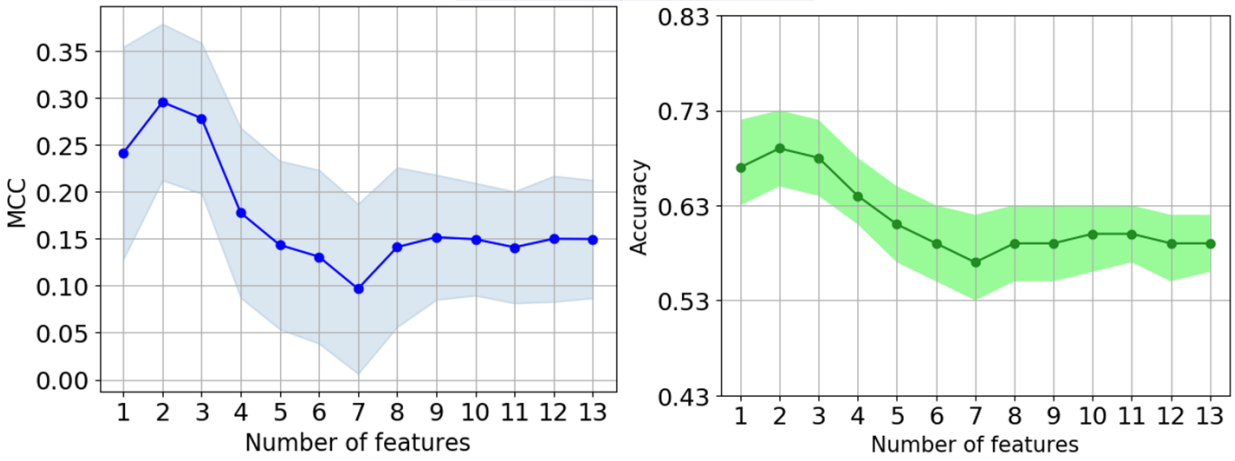
9.2.1 Forward Selection

The following graphs show the changes in Matthew's Correlation Coefficient (MCC) and accuracy trends of two modeling techniques of RF, GP and SVM as the feature numbers grow by selection and addition of one feature at a time. The table beneath the plots are showing the order of features being selected going forward.



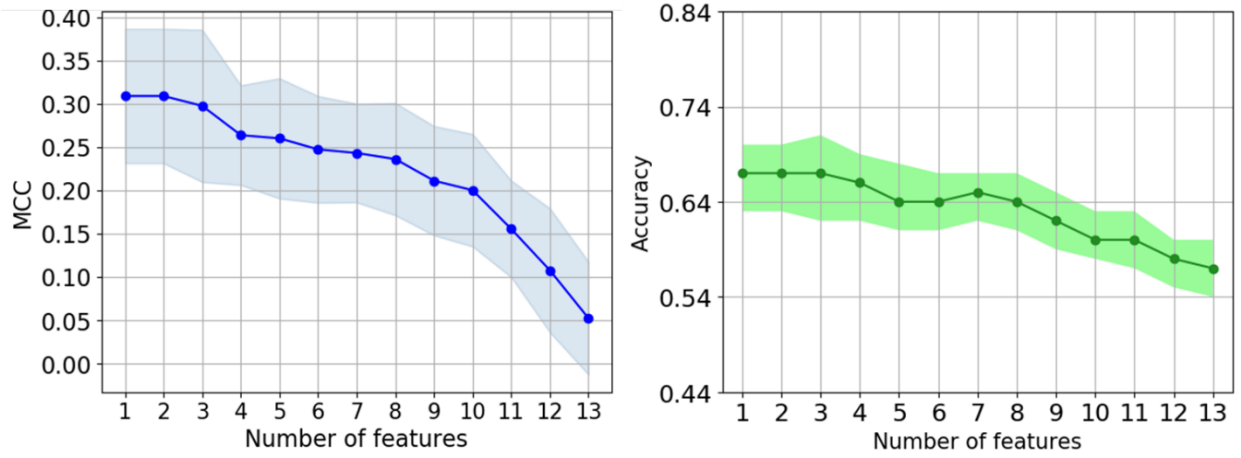
1	S/CHIR	8	CHIR concentration
2	Media1	9	Microspheroid Size
3	CHIR/S/V	10	Axial Ratio
4	Media2	11	Number of cells
5	CHIR/S	12	Passage post-freeze
6	PEG-fb concentration	13	S/V/CHIR
7	Cell concentration		

SVM

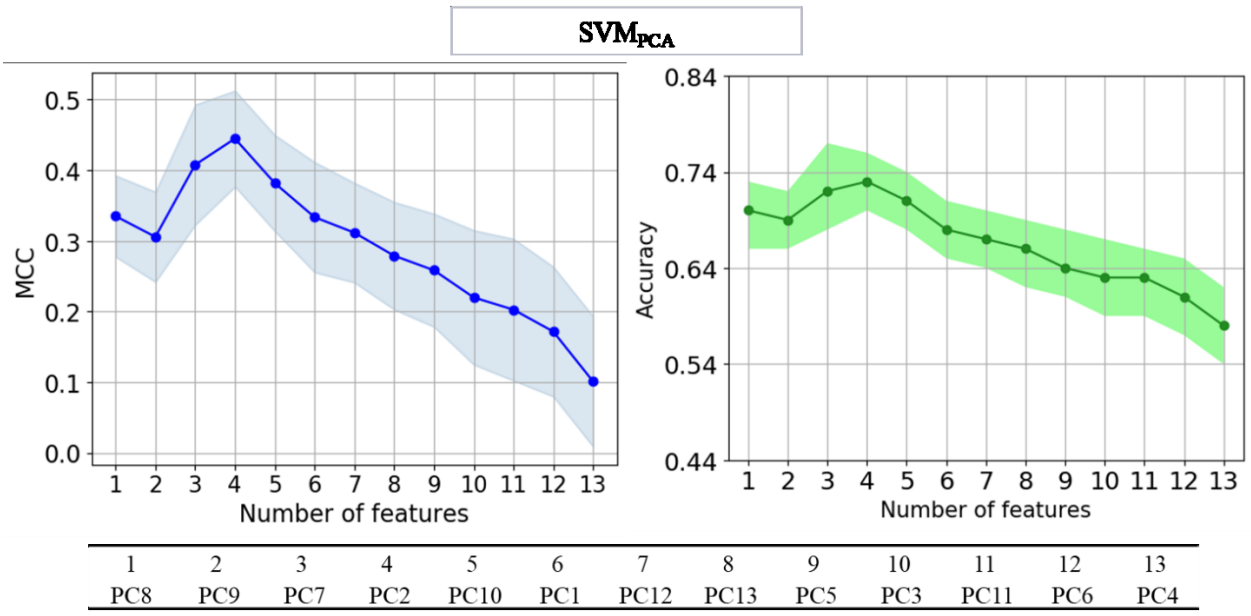


1	PEG-fb concentration	8	Media1
2	Passage post-freeze	9	Microspheroid Size
3	S/V/CHIR	10	CHIR/S/V
4	CHIR concentration	11	Axial Ratio
5	Number of cells	12	Cell concentration
6	Media2	13	S/CHIR
7	CHIR/S		

RF_{PCA}



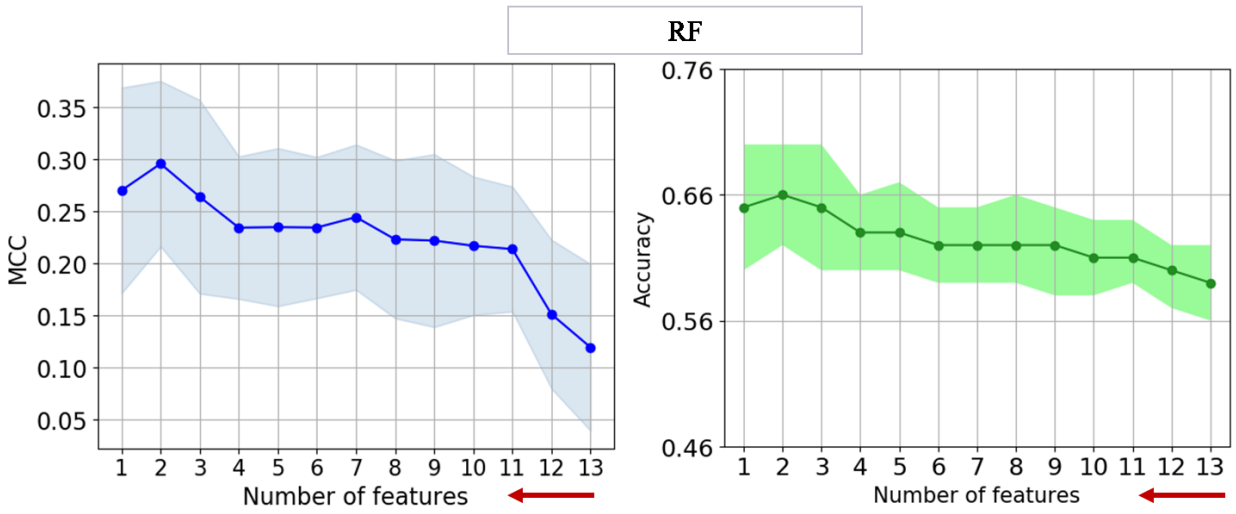
1	2	3	4	5	6	7	8	9	10	11	12	13
PC8	PC13	PC9	PC3	PC1	PC12	PC6	PC2	PC11	PC10	PC7	PC4	PC5



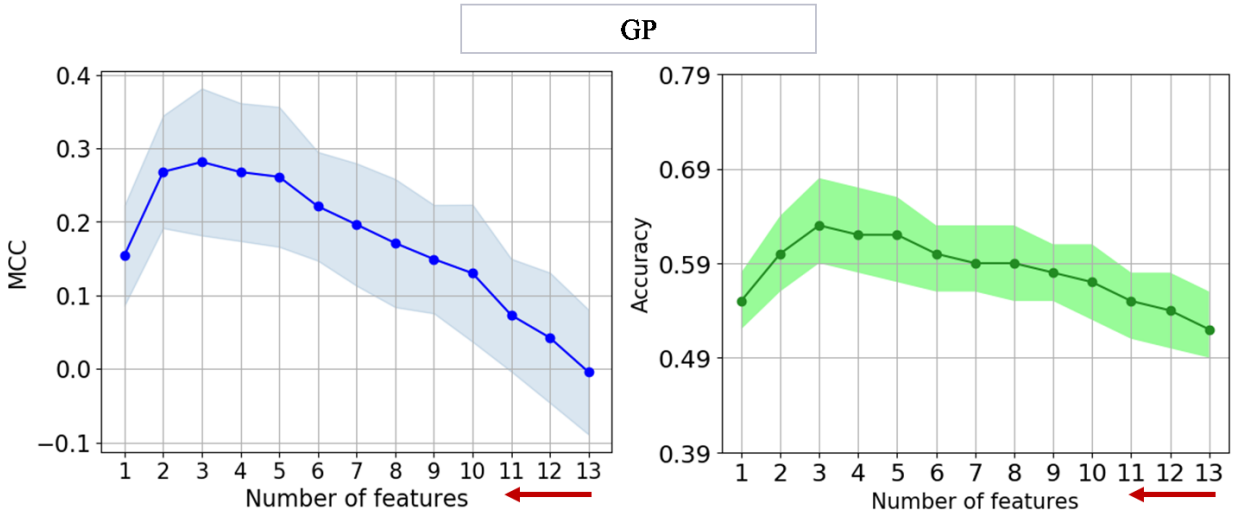
1	2	3	4	5	6	7	8	9	10	11	12	13
PC8	PC9	PC7	PC2	PC10	PC1	PC12	PC13	PC5	PC3	PC11	PC6	PC4

9.2.2 Backward Elimination

The following plots show the Backward Elimination search for feature selection procedure using wrapper methods. The graphs show the change in Matthew's Correlation Coefficient (MCC) and accuracy values of the different models as one feature at a time is eliminated from the input feature set. The red arrows show the initial point and the direction of the change in number of features. The results using RF, GP, and SVM modeling techniques. The tables below the graphs show the order which the features were eliminated from the feature set starting from index 13.

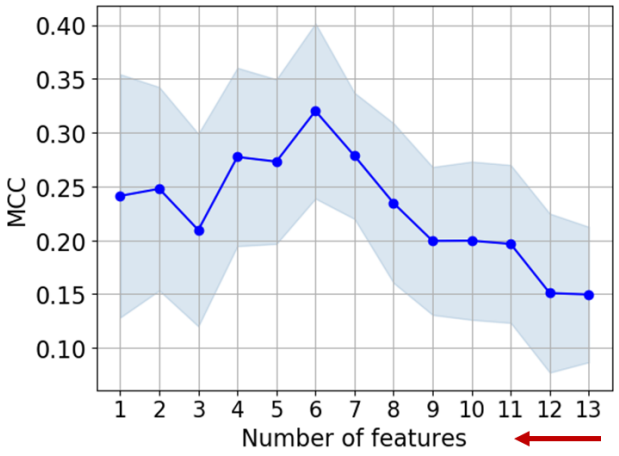


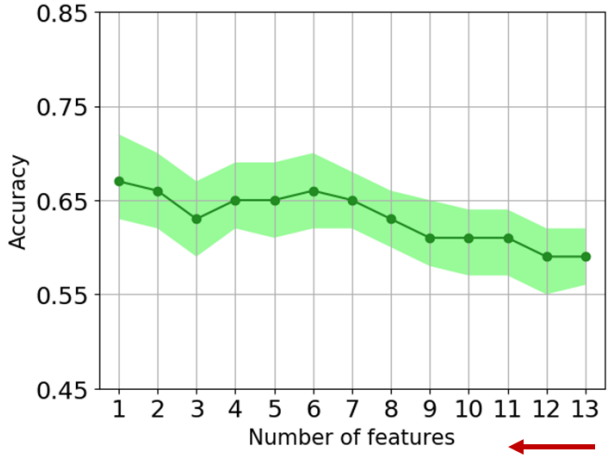
1	S/CHIR	8	Axial Ratio
2	PEG-fb concentration	9	Media1
3	CHIR concentration	10	Passage post-freeze
4	Number of cells	11	Media2
5	CHIR/S	12	Cell concentration
6	Microspheroid Size	13	S/V/CHIR
7	CHIR/S/V		



1	Cell concentration	8	Media2
2	PEG-fb concentration	9	CHIR concentration
3	S/CHIR	10	Number of cells
4	Microspheroid Size	11	CHIR/S/V
5	CHIR/S	12	Media1
6	Axial Ratio	13	Passage post-freeze
7	S/V/CHIR		

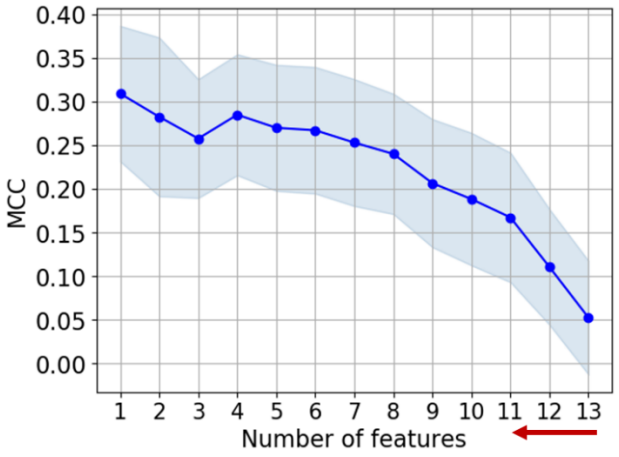
SVM

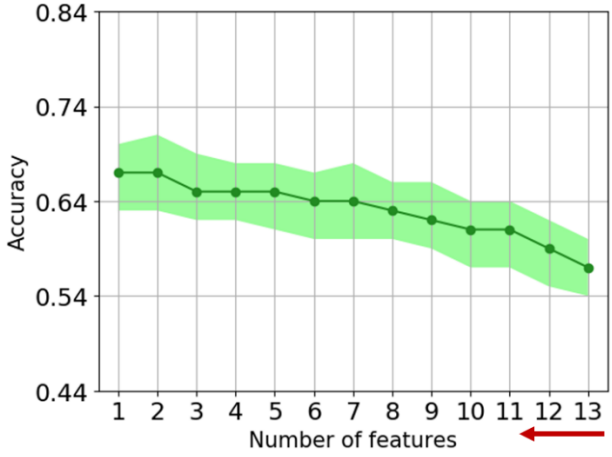




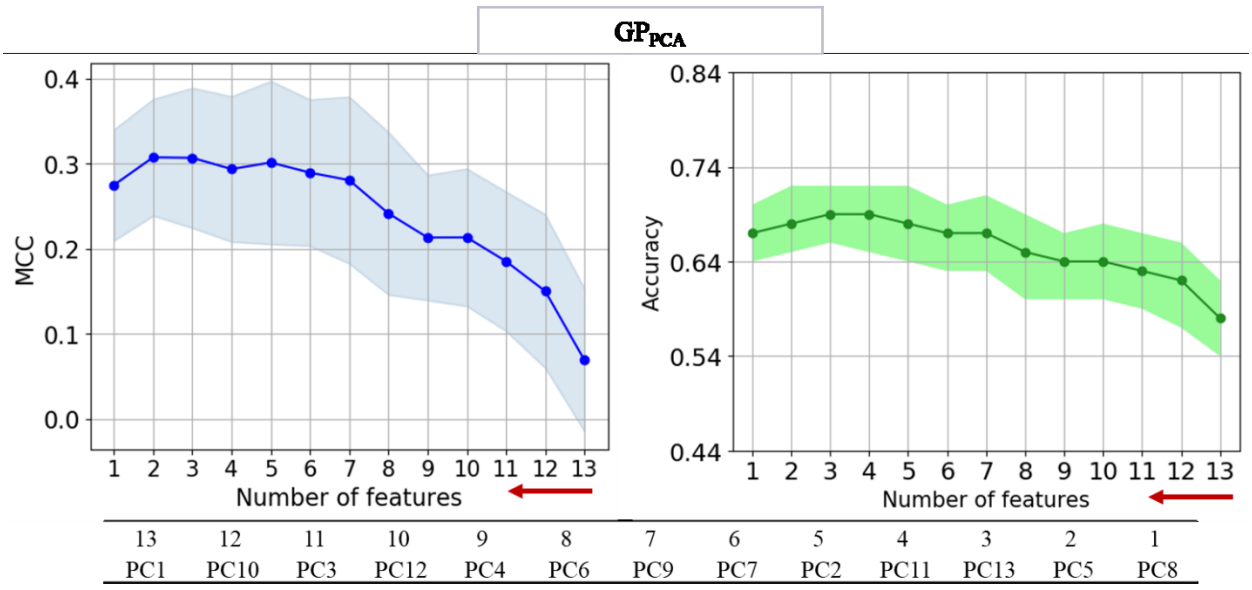
1	PEG-fb concentration	8	Cell concentration
2	CHIR concentration	9	CHIR/S
3	Passage post-freeze	10	S/CHIR
4	Number of cells	11	Axial Ratio
5	Media2	12	S/V/CHIR
6	Microspheroid Size	13	Media1
7	CHIR/S/V		

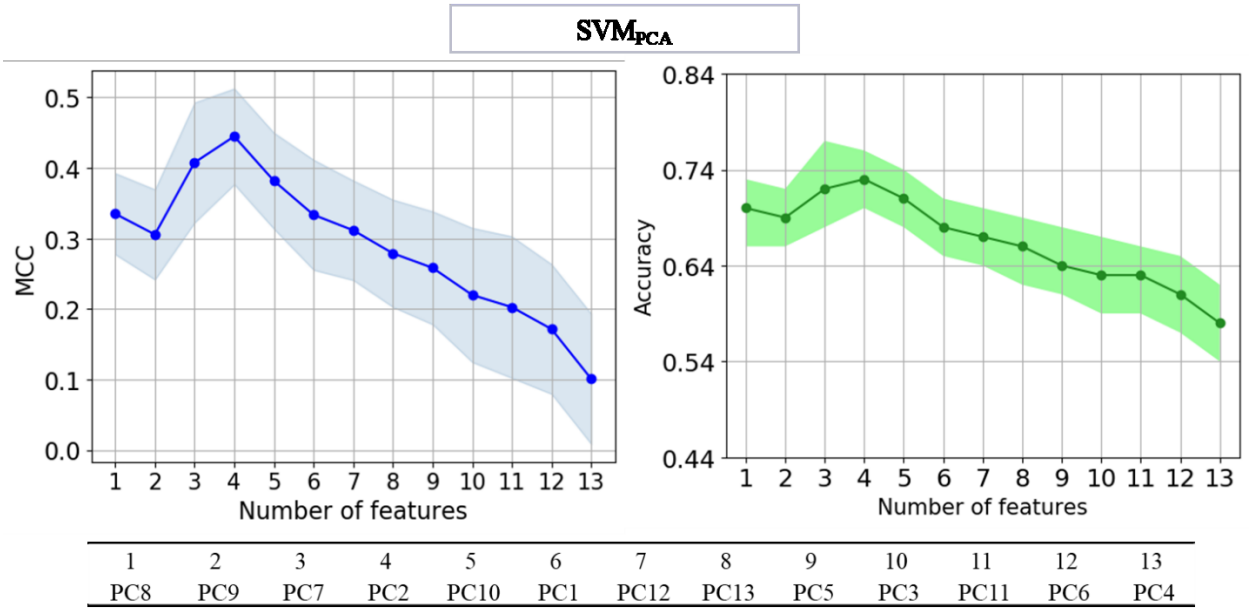
RF_{PCA}





13	12	11	10	9	8	7	6	5	4	3	2	1
PC2	PC7	PC3	PC11	PC10	PC4	PC5	PC13	PC6	PC12	PC1	PC9	PC8

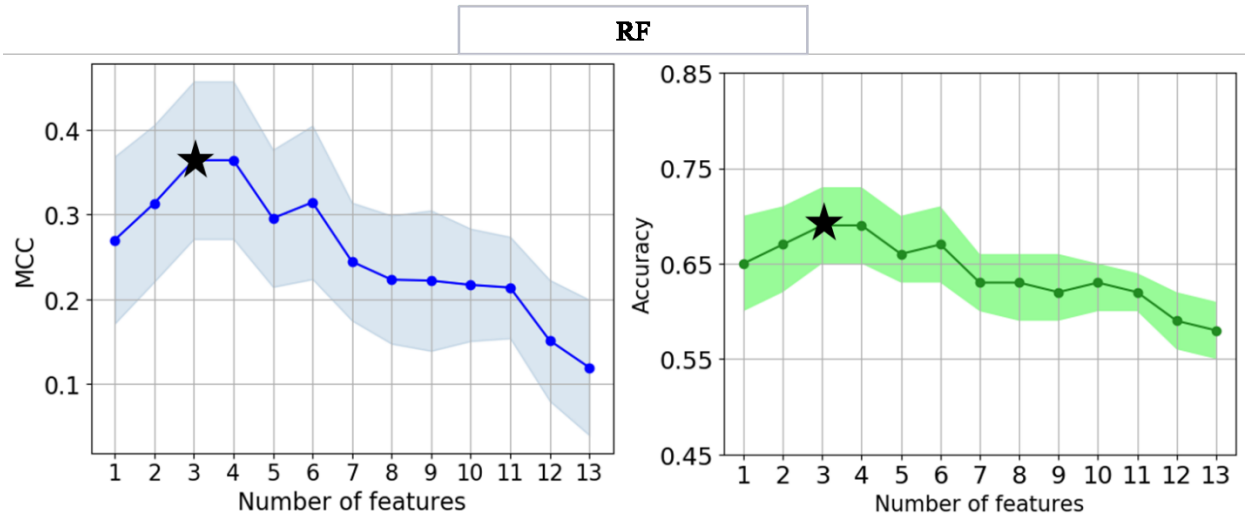




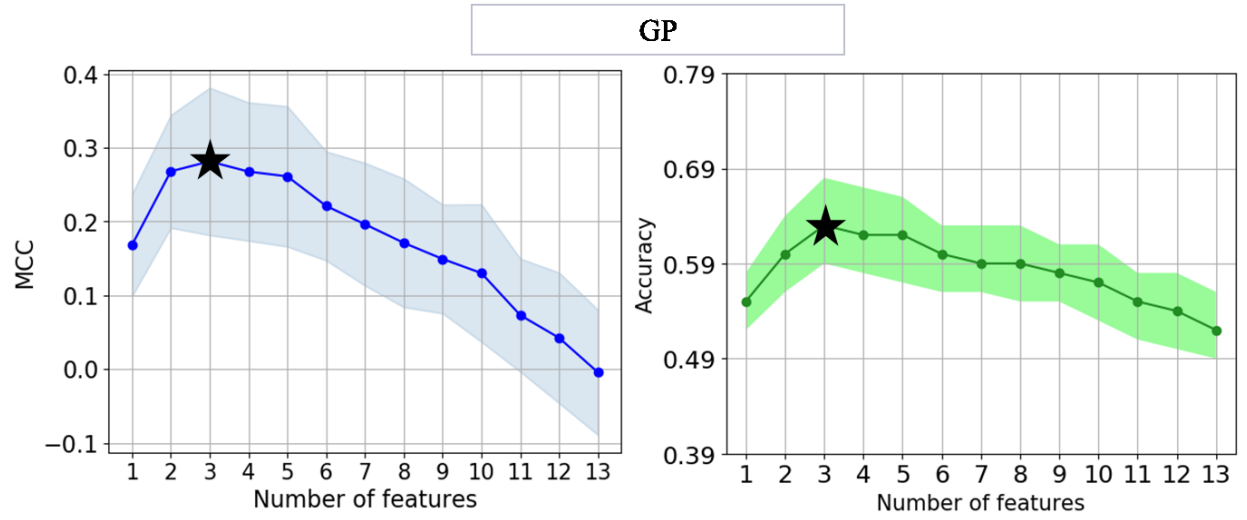
9.2.3 Bidirectional search

The graphs below show the changes in Matthew's Correlation Coefficient (MCC) and accuracy values as the Bidirectional search algorithm is used for feature selection. The table below

each plot shows the list of features which had the highest values of the metrics shown with a star on the graph.

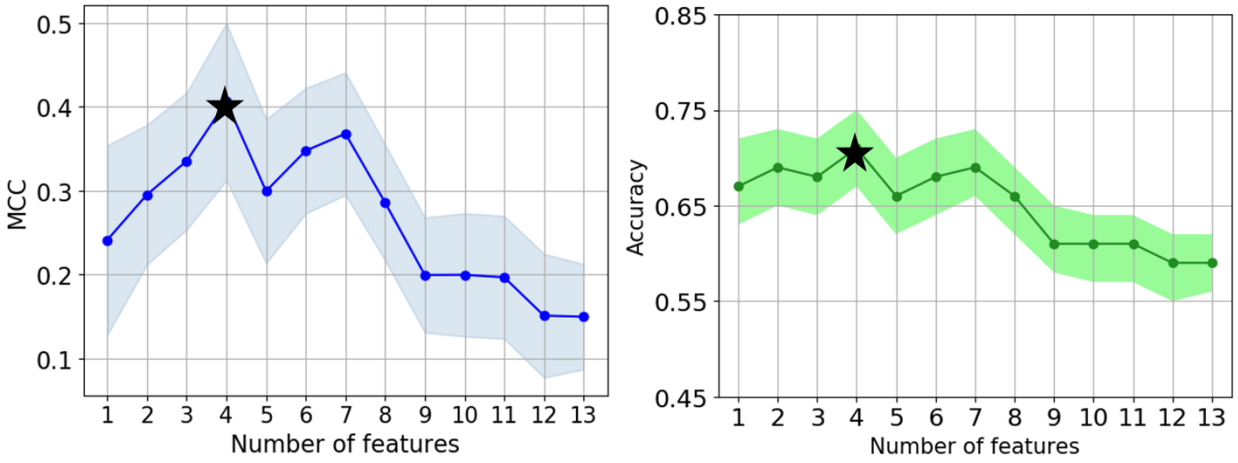


Cell concentration	PEG-fb concentration	Media1



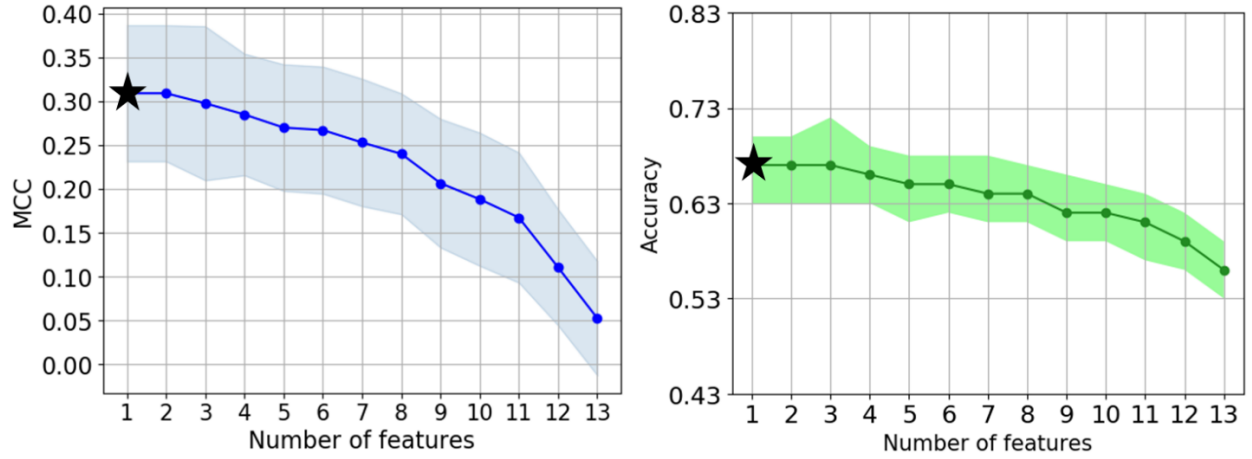
Passage post freeze	CHIR/S/V	Media1

SVM



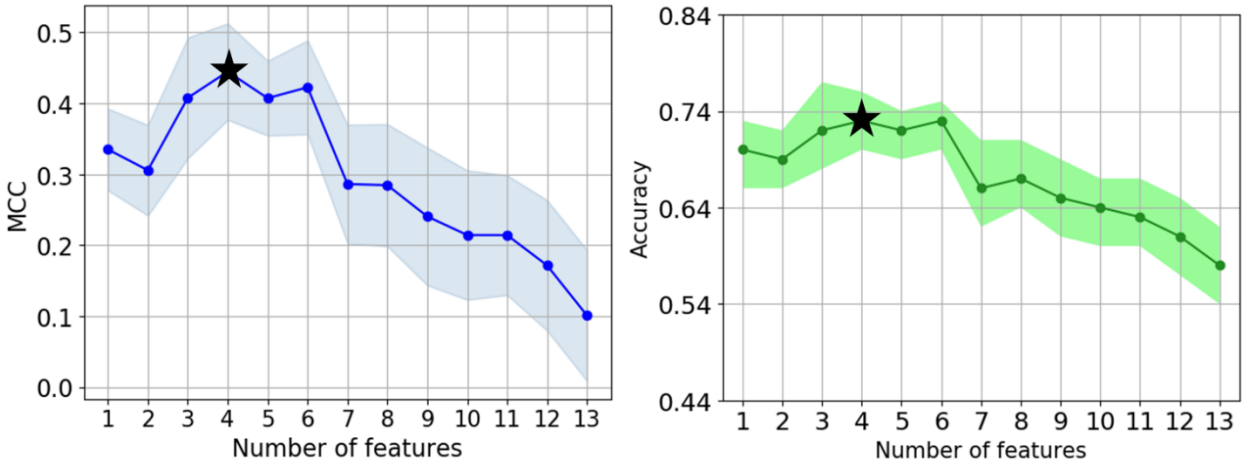
CHIR concentration	PEG-fb concentration	Post-freeze passage	S/V/CHIR

RF_{PCA}



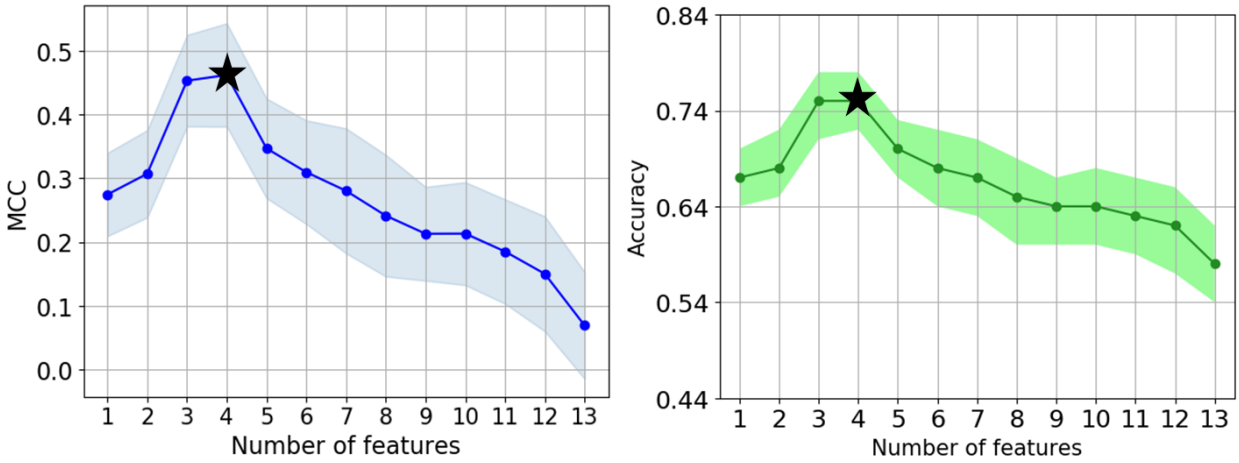
PC8

SVM_{PCA}



PC2	PC7	PC8	PC9

GP_{PCA}



PC4	PC5	PC8	PC13

Chapter 10 – **Appendix 3**

Table A3.1. Test functions with one uncertain parameter to investigate the impact of number of input dimensions.

Function	**Input range**	**Uncertain parameter distribution**
Ackley	$x_i \in [-32.768, 32.768]$	$k \sim N(20, 2)$
Dixon-Price	$x_i \in [-10, 10]$	$k \sim N(2, 0.2)$
Griewank	$x_i \in [-600, 600]$	$k \sim N(4000, 400)$
Levy	$x_i \in [-10, 10]$	$k \sim N(10, 1)$
Rastrigin	$x_i \in [-5.12, 5.12]$	$k \sim N(10, 21)$
Rosenbrock	$x_i \in [-5, 10]$	$k \sim N(100, 10)$
Schwefel	$x_i \in [-500, 500]$	$k \sim N(418.9829, 41.89829)$
Styblinski-tang	$x_i \in [-5, 5]$	$k \sim N(16, 1.6)$
Zakharov	$x_i \in [-5, 10]$	$k \sim N(0.5, 0.05)$

Table A3.2. Two-dimensional test functions to investigate impact of number of uncertain parameters.

Function	Input range	Uncertain parameter distribution
Beale	$x_i \in [-4.5, 4.5]$	$k_1 \sim N(1.5, 0.15)$ $k_2 \sim N(2.25, 0.225)$ $k_3 \sim N(2.625, 0.2625)$ $k_4 \sim N(1, 0.1)$
Bohachevsky1	$x_i \in [-100, 100]$	$k_1 \sim N(2, 0.2)$ $k_2 \sim N(0.4, 0.04)$ $k_3 \sim N(3, 0.3)$ $k_4 \sim N(0.7, 0.07)$
Bohachevsky2	$x_i \in [-100, 100]$	$k_1 \sim N(2, 0.2)$ $k_2 \sim N(0.3, 0.03)$ $k_3 \sim N(3, 0.3)$ $k_4 \sim N(0.3, 0.03)$
Bohachevsky3	$x_i \in [-100, 100]$	$k_1 \sim N(2, 0.2)$ $k_2 \sim N(0.3, 0.03)$ $k_3 \sim N(3, 0.3)$ $k_4 \sim N(0.3, 0.03)$
Booth	$x_i \in [-10, 10]$	$k_1 \sim N(2, 0.2)$ $k_2 \sim N(7, 0.7)$ $k_3 \sim N(2, 0.2)$ $k_4 \sim N(5, 0.5)$

Table A3.2. Two-dimensional test functions to investigate impact of number of uncertain parameters. (Cont.)

Function	**Input range**	**Uncertain parameter distribution**
Branin	$x_1 \in [-5, 10]$ $x_2 \in [0, 15]$	$k_1 \sim N(1, 0.1)$ $k_2 \sim N(5.1, 0.51)$ $k_3 \sim N(5, 0.3)$ $k_4 \sim N(10, 1)$
Bukin	$x_1 \in [-15, -5]$ $x_2 \in [-3, 3]$	$k_1 \sim N(100, 10)$ $k_2 \sim N(0.01, 0.001)$ $k_3 \sim N(0.01, 0.001)$ $k_4 \sim N(10, 1)$
Currin1	$x_i \in [0, 1]$	$k_1 \sim N(2300, 230)$ $k_2 \sim N(1900, 190)$ $k_3 \sim N(100, 10)$ $k_4 \sim N(500, 50)$
Currin2	$x_i \in [0, 1]$	$k_1 \sim N(21.15, 0.2115)$ $k_2 \sim N(2.17, 0.217)$ $k_3 \sim N(1.38, 0.138)$ $k_4 \sim N(5.26, 0.526)$
Frankes	$x_i \in [0, 1]$	$k_1 \sim N(0.75, 0.075)$ $k_2 \sim N(0.75, 0.075)$ $k_3 \sim N(0.5, 0.05)$ $k_4 \sim N(0.2, 0.02)$

Table A3.2. Two-dimensional test functions to investigate impact of number of uncertain parameters. (Cont.)

Function	**Input range**	**Uncertain parameter distribution**
Lim1	$x_i \in [0, 1]$	$k_1 \sim N(30, 3)$ $k_2 \sim N(5, 0.5)$ $k_3 \sim N(4, 0.4)$ $k_4 \sim N(5, 0.5)$
Lim2	$x_i \in [0, 1]$	$k_1 \sim N(35, 3.5)$ $k_2 \sim N(5, 0.5)$ $k_3 \sim N(15, 1.5)$ $k_4 \sim N(11, 1.1)$
Six hump camel	$x_1 \in [-3, 3]$ $x_2 \in [-2, 2]$	$k_1 \sim N(4, 0.4)$ $k_2 \sim N(2.1, 0.21)$ $k_3 \sim N(4, 0.4)$ $k_4 \sim N(4, 0.4)$
Three hump camel	$x_i \in [-10, 10]$	$k_1 \sim N(2, 0.2)$ $k_2 \sim N(1.05, 0.105)$ $k_3 \sim N(6, 0.6)$ $k_4 \sim N(1, 0.1)$