Open reference architecture for engineering model spaces

Deliverable 4.1

Overview of surrogate modelling approaches, including an initial risk analysis on standards for re-usability and transferability

Project Coordinator	Olga Kattan, Philips Consumer Lifestyle B.V.		
Start date Project	April 1st 2024	**Duration**	36 months
Version	1.0		
Status	Final		
Date of issue	30-04-2025		
Dissemination level	Public		

Authors

Author	Beneficiary
Ali Kafalı	Acd Bilgi İşlem Ltd.Şti.
Eelco Galestien	Philips Consumer Lifestyle B.V.
Federico Raspanti	Eindhoven University of Technology
Mathias Verbeke	Katholieke Universiteit Leuven
Mike Holenderski	Eindhoven University of Technology
Nicolas Lammens	Siemens Industry Software NV
Sawan Singh Mahara	Eindhoven University of Technology
Tomas Molina	Thermo Fisher Scientific
Dilara Bayar	Acd Bilgi İşlem Ltd.Şti.
M. Oguz Tas	Inovasyon Muhendislik
Final editor’s address	Olga Kattan Philips Consumer Lifestyle B.V. Oliemolenstraat 5 9203ZN Drachten / Netherlands

Executive Summary

This deliverable provides a systematic overview of surrogate modeling approaches for approximating computationally expensive high-fidelity models in engineering applications. We examine various techniques including Response Surface Methods, Gaussian Process Models, Neural Networks, Support Vector Machines, Tree-based Methods, Ensemble Models, and Multi-fidelity Models, analyzing their mathematical foundations, implementation requirements, and practical considerations. Each approach is evaluated across multiple performance dimensions including training and prediction time, memory usage, data requirements, scalability, and interpretability. The document also presents methodologies for model validation, monitoring, and benchmarking to ensure surrogate models maintain accuracy and reliability throughout their lifecycle. Additionally, we conduct an initial risk analysis focusing on standards for surrogate model reusability and transferability, identifying challenges and potential solutions for broader adoption across industries. This deliverable contributes to the SmartEM project by establishing a foundation for efficient engineering model spaces that balance computational efficiency with predictive accuracy.

Table of Contents

1. Introduction

Surrogate modeling represents a critical approach in engineering design and analysis, where computationally intensive high-fidelity models are replaced with approximations that maintain acceptable accuracy while significantly reducing computational costs. This document provides a comprehensive overview of surrogate modeling approaches, with particular focus on their reusability and transferability across engineering domains.

In modern industrial settings, high-fidelity models (such as Computational Fluid Dynamics, Finite Element Analysis, and electron beam simulations) require substantial computational resources and time, limiting their application in design exploration, optimization, and real-time control scenarios. Surrogate models address this challenge by learning input-output relationships from a limited set of high-fidelity simulations or experimental data, enabling rapid evaluation of new design configurations.

The growing complexity of engineering systems has driven the development of various surrogate modeling techniques, each with distinct characteristics suited to different application contexts. This document examines these techniques, their mathematical foundations, practical implementation considerations, and evaluation methodologies. Additionally, we present an initial risk analysis focused on standards for ensuring reusability and transferability of surrogate models across different engineering applications and domains.

This deliverable contributes to the broader objective of establishing an open reference architecture for engineering model spaces, supporting the development of standardized approaches to surrogate model creation, validation, and deployment across industrial applications.

2. Fundamentals of Surrogate Modelling

2.1. Definition and Purpose

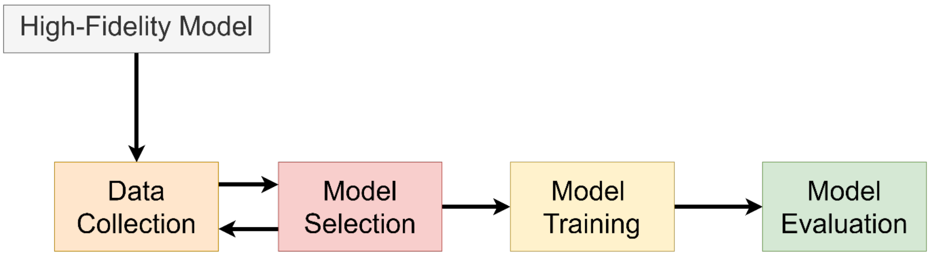
Surrogate modelling is a data-driven approach which aims to approximate a more complex, high-fidelity model, by learning to capture input-output relationships.

Especially in industrial settings, surrogate modelling helps reduce computational costs, in terms of both time and resources. For example, instead of running computationally expensive computational fluid dynamics (CFD) simulations to predict the drag on different car designs, once a surrogate model is trained on input-output examples of CFD simulations, it could be used to quickly estimate drag coefficients for different car designs.

This enables deployment in resource-constrained environments, as well as extensive design space exploration and system optimization that would be impractical with high-fidelity models. In the automotive example, this means engineers can rapidly explore many variations in vehicle shape to optimize aerodynamics, rather than being limited to testing just a handful of designs due to CFD computational constraints.

In this context, a key element to consider is the trade-off between computational efficiency and model fidelity. To address this, a wide range of design, development, evaluation, monitoring, and benchmarking techniques have been developed over the years. (Alizadeh, Allen, & Mistree, 2020) (Kudela & Matousek, 2022)

2.2. Key Components



2.2.1. Data collection

This step involves selecting sample points from the design space and obtaining corresponding outputs from the high-fidelity model. For instance, in aerodynamic optimization, this might mean running detailed CFD simulations for a chosen set of wing designs. Here the sampling strategy plays a very important role: we ideally want to cover as much of the design space as possible using as few samples as possible. Various sampling strategies, like Latin Hypercube Sampling and adaptive sampling, exist to help ensure good coverage of the design space with minimal computational expense (Loh & W., 1996) (Helton, J., Davis, & F., 2003) (Jain, A., Chang, & E., 2004). For a comprehensive overview of sampling and design of experiments (DoE) techniques, see (Arboretti, Ceccato, Pegoraro, & Salmaso, 2022).

2.2.2. Model Selection

The choice of surrogate model type often goes together with data collection decisions. Different surrogate models have distinct characteristics that make them suitable for different applications. For example, kriging models are well suited for capturing nonlinear behaviour and providing uncertainty estimates, while neural networks might be preferred when dealing with high-dimensional problems with complex feature interactions.

When choosing the appropriate surrogate model, it is best to refer to existing literature about model selection (Jin, Chen, & Simpson, 2001) (Williams & Cremaschi, 2019). However, due to a lack of widely accepted standards for model selection, many decisions in this phase are left to the ML/AI engineers' expertise, and their understanding of the requirements. Alternatively, a number of frameworks have been developed to automatically select the most appropriate surrogate, based on the dataset's features (Jia, et al., 2020) (Li, Wang, Luo, Jiang, & Li, 2023) (Mehmani, Chowdhury, Meinrenken, & Messac, 2018) (Ben Salem & Tomaso, 2018).

2.2.3. Model Training

With model training we refer to the process of calibrating a surrogate model to capture the underlying relationships in the collected data. During this phase we optimize the surrogate model's parameters to minimize the discrepancy between its predictions and the outputs from the high-fidelity model, or the experimental data.

We typically begin with data preprocessing, which involves, for example, normalizing input variables to comparable scales, addressing missing values, and encoding categorical variables into numerical representations. This preprocessing improves numerical stability and reduces potential biases in the training procedure. See (Bala & Behal, 2024) for a complete overview of techniques.

Parameter optimization follows a loss function minimization approach, where the loss function quantifies the deviation between the predictions of the surrogate model, and the actual outputs. Common loss functions include mean squared error for regression tasks and cross-entropy for classification problems. The optimization algorithm iteratively adjusts model parameters to reduce this error metric across the training dataset. For a complete overview of loss functions, see (Ciampiconi, Elwood, Leonardi, Mohamed, & Rozza, 2023).

For models with significant hyperparameters that control model structure or learning behaviour, hyperparameter tuning is an essential component of the training process. Methods such as grid search, random search, or Bayesian optimization help identify optimal hyperparameter configurations.

The training process concludes with convergence checks, where parameter updates become sufficiently small or validation metrics plateau, indicating that further optimization would yield diminishing returns. The trained surrogate model can then proceed to evaluation and validation against independent test data.

2.2.4. Model Evaluation

In this step we assess the surrogate model's performance to ensure its reliability as a replacement for the high-fidelity model.

Evaluation begins by analysing the error of our surrogate model against the test data, using statistical error metrics. This preliminary evaluation allows us to check the robustness of the model. A useful and widespread evaluation technique is cross-validation, where we partition the dataset into multiple training and testing subsets. In K-fold cross-validation, a popular variant, we split data into k parts, training the model k times with different test subsets. This approach helps detect overfitting and provides confidence in the model's generalization capabilities across the input domain (Qiu, 2024).

For engineering applications, consider domain-specific tests check whether surrogate predictions satisfy physical constraints, evaluate performance near design boundaries, and assess sensitivity to input variations.

The evaluation concludes with validation against the high-fidelity model at selected test points, especially in regions of interest. This verification confirms the surrogate represents the original model's behaviour while increasing computational efficiency. Section 4 provides a more in-depth examination of advanced validation techniques, including model monitoring and model reuse.

3. Surrogate Models

Surrogate modelling is an essential approach in modern engineering design, where decision-making relies heavily on computational analysis. Given the high computational cost of simulations in fields like aerospace, engineers often seek faster yet accurate alternatives to traditional tools such as computational fluid dynamics or structural mechanics (Alexander I, Forrester, & Andy, 2008). Surrogate models, are developed to approximate the results of expensive simulation codes, enabling predictions across unexplored regions of the input space without running the original simulations. These models are particularly valuable when only a limited number of simulation runs are feasible due to time or resource constraints. The goal is to achieve significant speedups while maintaining useful accuracy, a trade-off that must be carefully balanced. In addition to approximating single high-fidelity codes, surrogates can calibrate lower-fidelity models, bridge different simulation fidelities, or integrate computational and experimental data (Alexander I, Forrester, & Andy, 2008). They are also useful in handling noisy or incomplete datasets, acting as filters that reveal underlying trends and fill missing values. Furthermore, surrogate models serve as tools for data mining, helping engineers identify key variables and gain deeper insight into functional relationships.

3.1. Overview

The table below compares seven surrogate modelling techniques: Response Surface Methods, Gaussian Process, Neural Networks, Support Vector Regression, Random Forest/Boosted Trees, Ensembles, and Multi-Fidelity models. Each technique is evaluated across eleven performance criteria including Training Time, Prediction Time, Memory Usage, Data Requirements, Manual Tuning, Dimensional and Dataset Size Scalability, Non-linearity Capture, and Interpretability. The table uses a colour scheme ranging from red (very poor) to green (very good), offering a visual guide to each model's suitability under different priorities described by the performance criteria. It also lists supported data modalities for each method in the final uncoloured column.

	Training Time	Prediction Time	Memory Usage	Data Requirements	Manual Tuning Required	Dimension Scalability	Dataset Size Scalability	Non-linearity Capture	Interpretability	Modality
Response Surface Methods	🟢	🟩	🟩	🟩	🟧	🟥	🟢	🟧	🟩	1,2
Gaussian Process	🟧	🟨	🟧	🟢	🟥	🟥	🟧	🟩	🟨	1,2
Neural Networks	🟥	🟢	🟧	🟥	🟧	🟢	🟩	🟩	🟥	1,2,3,4, 5,6,7
Support Vector Regression	🟧	🟨	🟨	🟨	🟧	🟨	🟧	🟢	🟧	1,2
Random Forest, Boosted Trees	🟨	🟢	🟨	🟧	🟢	🟢	🟩	🟢	🟨	1,2,3
Ensembles	🟥	🟧	🟥	🟥	🟨	🟨	🟢	🟩	🟧	1,2
Multi-Fidelity	🟧	🟨	🟧	🟨	🟥	🟢	🟢	🟢	🟨	1,2

Table 1: Overview of surrogate modelling techniques. This table compares different models (rows) based on their suitability when specific criteria (columns) are prioritized. The comparison assumes equivalent training conditions

and optimal alignment between data and model structure. The fit can be very poor, poor, neutral, good, very good. Data modalities are: Tabular (1), Numerical (2), Categorical (3), Text (4), Image (5), Audio (6), Video (7).

Response Surface Methods emerge as the most efficient for low complexity tasks due to their low training time, minimal data requirements, and high interpretability. However, they perform poorly in capturing non linearity and handling high dimensions. Gaussian Processes and Neural Networks, while strong in capturing non linear relationships, suffer from high training time, memory usage, and poor scalability. Support Vector Regression is generally average across most parameters, offering a balanced but unremarkable trade off. Random Forest and Boosted Trees score well on dataset size scalability and non-linearity capture but require more manual tuning. Ensembles and Multi Fidelity models show poor scores in efficiency and scalability, limiting their use for real time or resource constrained applications. In terms of modality support, Neural Networks are the most versatile, handling all data types including images and audio, whereas others are mostly limited to tabular and numerical data. Overall, model selection should align with task complexity, data availability, and interpretability needs. For instance, interpretable low data scenarios favour Response Surface Methods, while complex, high dimensional tasks with ample data may benefit from Neural Networks or Ensemble methods despite their training overhead.

3.2. Details

3.2.1. Response Surface Methods

Model

In Response Surface Methods, we represent the model as a linear combination of polynomial functions of the input variables:

$$\hat{f}(x) = \sum_{i=0}^{p} \beta_i \phi_i(x)$$

where $\phi_i(x)$ represent polynomial basis functions up to order p and β_i represent unknown coefficients to be estimated.

Predictor

The RSM predictor is formulated as:

$$\hat{f}(x) = \Phi(x)^T \beta^*$$

where $\Phi(x) = [\phi_0(x), \dots, \phi_p(x)]^T$ is the vector of basis functions and β^* is the least-squares estimate of coefficients.

Properties and considerations

RSMs work well for problems where we can assume a relatively smooth response surface without sharp discontinuities or highly localized features. Consider for example a thermal management system for electronic components, where the relationship between design parameters (component spacing, heat sink dimensions, fan speed) and performance metrics (maximum temperature, thermal uniformity) needs to be modelled.

RSM would be best suited for early design stages of the system, where we need to quickly explore many different configurations. We usually begin with a second-order model unless domain knowledge suggests otherwise. In our example, a first-order model might be enough for basic heat spreading in simple geometries, but we may need a second-order model when accounting for thermal radiation and temperature-dependent material properties.

Sampling should try to balance covering as much of the design space as possible, with targeted sampling near regions we expect to be important. The number of samples should be at least (n+1)(n+2)/2 for a second-order model with n variables, though 1.5 to 2 times this number is recommended. For example, with 5 design variables, this means a minimum of 21 sampling points, though 30-40 points are recommended for robust fitting.

Advantages for Surrogate Modelling

RSMs are computationally efficient, with evaluation times typically orders of magnitude faster than finite element analyses or CFD simulations. This makes them suitable for integration into optimization loops and real-time applications. They also provide sensitivity information through their polynomial coefficients, making it easier to analyse each variable’s importance.

Limitations and Risk Mitigation

The polynomial basis limits RSMs’ ability to capture highly nonlinear behavior. To mitigate this, consider:

- Partitioning the design space into regions with similar behavior characteristics
- Using variable transformation techniques to linearize known nonlinear relationships
- Using mixed-order models where different variables exhibit different degrees of nonlinearity

When dealing with noisy data, weighted least squares fitting can improve model stability, by giving lower weight to noisy data points.

3.2.2. Gaussian Process Modelling (Kriging)

Model

The Kriging method represents the model as a combination of a polynomial trend and localized deviations:

$$\hat{f}(x) = \sum_{i=1}^{m} a_j g_j(x) + \varepsilon(x)$$

where $g_j(x)$ represent basis functions for the global model, a_j are unknown parameters, and $\varepsilon(x)$ is a zero-mean random error.

Predictor

The Kriging predictor is formulated as:

$$\hat{f}(x) = g(x)^T a^* + r(x)^T \gamma^*$$

where $g(x)$ is the vector of basis functions, a^* are the estimated parameters, $r(x)$ is the correlation vector between prediction and observed points, and γ^* are local adjustment weights.

Properties and Considerations

Kriging works well for problems with complex, spatially varying behavior that still maintains some degree of smoothness. Consider a thermal management system where we need to predict temperature distributions across a circuit board with multiple heat sources and cooling elements.

We can use Kriging when we need both accurate predictions and uncertainty quantification, for example when we want to be sure that maximum temperature limits are met with high confidence across the entire board, not just at the measured points.

The choice of correlation function determines how the model handles variations in the response surface. Gaussian kernels are suitable for smooth, infinitely differentiable functions, while Matérn kernels offer more flexibility in modelling different degrees of smoothness. For example, Gaussian kernels would work well for smooth temperature fields, while Matérn kernels would be better for cases with sharper thermal gradients near heat sources or cooling elements. The correlation parameters should be tuned through maximum likelihood estimation.

Advantages for surrogate modelling

Kriging is useful for cases where we need an accurate model of our system, as well as uncertainty quantification. The uncertainty estimates provide confidence intervals on predictions and identify regions of high uncertainty, making it easier to use adaptive sampling: new sampling points can be selected to either reduce overall prediction uncertainty (exploration) or focus on promising regions for optimization (exploitation). This makes Kriging particularly effective for expensive black-box optimization problems where sampling budget is limited.

Kriging is also well suited to combine a small number of expensive, high-fidelity simulations with a larger set of cheaper, low-fidelity data. For example, in thermal systems, we can combine detailed 3D simulations with simpler 1D models to efficiently achieve better accuracy.

Limitations and Risk Mitigation

Computational cost scales cubically with sample size, making it challenging for large datasets. Consider:

- Using local Kriging methods that focus on nearby points
- Applying dimension reduction techniques for high-dimensional problems
- Using sparse approximations when dealing with more than 1000 sampling points

The correlation matrix may become unstable with points sampled close to each other. To improve numerical stability, a small positive value (10^{-8} to 10^{-6} times the variance) can be added to the diagonal of the matrix. This adjustment, known as regularization, prevents computational issues when inverting the matrix, making the Kriging model more reliable.

3.2.3. Artificial Neural Networks

Model

In Artificial Neural Networks (ANNs), we represent the model as a composition of multiple layers of interconnected neurons:

$$\hat{f}(x) = h^{(L)}\left(h^{(L-1)}\left(\ldots h^{(1)(x)}\right)\right)$$

where h(l) represents the l-th layer transformation:

$$h^{(L)(z)} = \sigma\left(W^{(L)} z + b^{(L)}\right)$$

Where $W^{(L)}$ is the weight matrix, $b^{(L)}$ is the bias vector, and σ is a nonlinear activation function. $W^{(L)}$ and $b^{(L)}$ are learning during training, with the goal to minimize the chosen loss. The choice of loss function depends on the task - typically mean squared error for regression and cross-entropy for classification.

Predictor

The ANN predictor propagates input through the network layers:

$$\hat{f}(x) = W^{(L)}\sigma\left(W^{(L-1)}\sigma\left(\cdots\sigma\left(W^{(1)}x + b^{(1)}\right)\cdots\right) + b^{(L-1)}\right) + b^{(L)}$$

The final output depends on the network architecture and activation functions chosen for the task.

Properties and Considerations

Neural networks are well-suited for problems with complex nonlinearities, high dimensionality, and large datasets. They are useful when the relationship between inputs and outputs cannot be described by simple mathematical functions. For instance, in thermal management, neural networks can capture complex relationships between geometric parameters, material properties, and thermal performance that involve many interactions and nonlinearities.

The architecture choice significantly impacts model performance. Deeper networks (more layers) are appropriate when the problem involves hierarchical feature extraction or requires multiple levels of abstraction. Wider networks (more neurons per layer) are better suited when the problem requires extensive parallel feature processing. In practice, start with a simple architecture (2-3 hidden layers) and gradually increase complexity if needed.

Modern implementations typically use ReLU activation functions for hidden layers due to their computational efficiency and ability to mitigate vanishing gradient problems. For the output layer, linear activations are standard for regression problems, while sigmoid or softmax functions are used for classification tasks.

Advantages for Surrogate Modelling

Neural networks are useful for capturing highly nonlinear relationships without requiring prior assumptions about the functional form. Their flexibility allows them to automatically learn relevant features from data, reducing the need for manual feature engineering. This makes them valuable for complex systems where the underlying physics is either unknown or too complex to model directly.

According to the data types, many specialized architectures are available. Convolutional neural networks (CNNs) efficiently process spatial data like temperature distributions across surfaces. Recurrent neural networks (RNNs) and transformers handle time-dependent phenomena such as transient thermal responses. Graph neural networks can model relationships in complex networked systems.

Limitations and Risk Mitigation

Neural networks typically require larger datasets than other surrogate models. When data is limited, consider:

- Data augmentation techniques to artificially expand the training set.
- Transfer learning from pre-trained models in related domains.
- Regularization methods like dropout and L2 regularization to prevent overfitting.

Neural networks are often difficult to interpret, making it challenging to understand how predictions are made. Techniques such as feature importance analysis, SHAP values, or layer-wise relevance propagation can help improve interpretability.

Training neural networks requires careful hyperparameter tuning. Use grid search, random search, or Bayesian optimization to find optimal configurations for learning rate, network depth/width, and regularization parameters.

3.2.4. Support Vector Machine

Model

In Support Vector Machines, we represent the model as a decision boundary that separates data points while maximizing the margin between classes. The model aims to find the optimal hyperplane that provides the widest possible separation between classes while minimizing classification errors. For nonlinear problems, the model employs kernel functions to transform the input space into a higher-dimensional feature space where linear separation becomes possible.

Predictor

The SVM predictor classifies new data points based on their position relative to the decision boundary, using support vectors (the data points closest to the decision boundary) and the chosen kernel function. For regression tasks (SVR), the predictor estimates values based on a similar principle, using a loss function that allows for small deviations from the true values.

Properties and Considerations

SVMs are well suited for classification and regression problems that require robustness against outliers and noise. They work best with medium-sized datasets where the underlying function has moderate complexity. For example, in thermal applications, SVMs can provide reliable temperature predictions even when sensor data contains occasional measurement errors or when the system experiences brief anomalous conditions.

The kernel selection should be based on the expected characteristics of the underlying function. Linear kernels work well when the relationship is approximately linear or when the feature space is high-dimensional. Polynomial kernels capture specific orders of interactions between features. Radial Basis Function (RBF) kernels, the most used, model a wide range of nonlinear relationships by measuring similarity based on distance in feature space.

Advantages for Surrogate Modelling

SVMs provide excellent generalization from limited data thanks to their margin-based formulation. By focusing on the most informative data points, they discard redundant information and resist overfitting. This makes SVMs useful when acquiring training data is expensive or time-consuming.

The convex optimization problem in SVM training guarantees a global optimum, avoiding the local minima issues that affect many other nonlinear methods. This leads to more consistent performance across different initializations and reduces the need for multiple training runs.

Limitations and Risk Mitigation

The computational complexity scales poorly with dataset size, typically between $O(n^2)$ and $O(n^3)$. For larger datasets, consider:

- Using linear SVMs, which scale better.
- Applying random sampling or clustering to reduce dataset size.

Feature scaling is important for SVM performance. Inputs should be normalized to prevent features with larger ranges from dominating the distance calculations. Standard scaling (zero mean, unit variance) or min-max scaling to [0,1] range are common approaches.

Parameter selection significantly impacts performance. The regularization parameter (C) and kernel parameters (e.g., RBF bandwidth γ) should be determined through cross-validation. Grid search with logarithmic spacing is useful for initial exploration and should be followed by finer searches in promising regions.

3.2.5. Random Forest and Boosted Trees

Model

In Random Forest, we represent the model as an ensemble of decision trees:

$$\hat{\mathrm{f}}(\mathrm{x}) = \frac{1}{\mathrm{B}} \sum_{\mathrm{b}=1}^{\mathrm{B}} \mathrm{T}_{\mathrm{b}}(\mathrm{x})$$

where $T_b(x)$ represent individual decision trees, *B* is the number of trees in the forest, and each tree is trained on a bootstrap sample of the data.

Predictor

The Random Forest predictor is formulated as:

$$\hat{f}(x) = \sum_{b=1}^{B} w_b T_b(x)$$

where w_b represents the weights (uniform for Random Forests, learned in boosting) and $T_b(x)$ denotes individual tree predictions. The final prediction is computed as either an average or weighted combination of tree outputs.

Properties and Considerations

Tree ensemble methods are useful for complex engineering problems where the relationship between variables contains both linear and nonlinear components, and the input variables have different scales and types. They can handle mixed variable types without preprocessing, are robust against outliers, and can calculate feature importance. For example, a manufacturing process optimization problem where we need to predict product quality based on a mix of continuous parameters (temperature, pressure, time) and categorical variables (material type, machine used).

For time-critical applications where training speed matters but some accuracy can be sacrificed, Extremely Randomized Trees offer faster training. For highest accuracy, especially when dealing with imbalanced data or rare events prediction (like failure prediction in reliability engineering), Gradient Boosting typically outperforms other tree-based methods.

These methods handle mixed variable types without preprocessing and provide built-in feature importance measures through split statistics. The hierarchical structure limits outlier influence, and the algorithm's structure enables parallel computation, but the bounded nature of tree predictions makes extrapolating beyond the training data limited.

The hyperparameter selection depends on the specific application. For problems with high noise, limit tree depth (typically between 3-10) and use a larger number of trees (100-500) to promote better generalization. For cleaner datasets with complex patterns, deeper trees can capture more intricate relationships. Cross-validation should be used to find the optimal trade-off between model complexity and generalization ability.

Advantages for Surrogate Modelling

Tree ensembles excel in handling nonlinear relationships without requiring assumptions about the underlying function form. They capture variable interactions through their hierarchical splitting structure - for instance, if thermal performance depends on the specific combination of material type and thickness (not just each independently), tree ensembles will automatically model this interaction without explicit cross-terms. Unlike many other models, they don't require input scaling - whether temperature is measured in Celsius or Kelvin won't affect the model's performance, as the algorithm only cares about the relative ordering of values when making splits.

The built-in feature importance reveals which variables have the greatest impact on the output, helping to identify key design parameters and offering much better interpretability compared to methods like ANNs. For example, in a thermal management system, tree-based models can quickly identify whether fan speed, heat sink geometry, or component spacing has the greatest impact on maximum temperature.

Tree ensembles also offer natural handling of missing data and outliers, making them robust for real-world engineering data where measurements may be incomplete or contain errors. The ensemble nature of this approach reduces variance, making predictions more stable compared to single models. For instance, in structural analysis, where small changes in loading conditions might cause large variations in a single simulation, the ensemble approach averages out these fluctuations to provide more reliable predictions of structural responses across the design space.

Limitations and Risk Mitigation

Tree ensembles struggle with extrapolation beyond the training data. To mitigate this:

- Ensure training data covers the entire operating range of interest.
- Use physics-based rules as post-processing filters on model outputs (e.g., enforcing that predicted temperatures must satisfy conservation of energy or that structural deformations remain within material elastic limits).
- Monitor prediction confidence and flag predictions made in sparse data regions.

For high-dimensional problems (>50 variables), feature selection or dimensionality reduction techniques should be applied before training. Principal Component Analysis (PCA) or domain-knowledge-based selection can improve both performance and interpretability.

When dealing with large datasets (>100,000 samples), consider using subsampling techniques or distributed implementations to manage computational complexity. For streaming or online learning, incremental variants like Online Random Forest can be employed to update models with new data without complete retraining.

3.2.6. Ensembles of surrogates

Model

In surrogate ensembles, we represent the model as a combination of multiple individual surrogate models:

$$\hat{f}(x) = \sum_{\mathrm{i=1}}^{\mathrm{M}} w_i \, \widehat{\mathrm{f}_\imath}(\mathrm{x})$$

where $\widehat{f_\imath}(x)$ represents individual surrogate models (which could be of different types), M is the number of models in the ensemble, and w_i represents the weight assigned to each model.

Predictor

The ensemble predictor is formulated as:

$$\widehat{f_\imath}(x) = g\left(\widehat{f_1}(x), \widehat{f_2}(x), \dots, \widehat{f_M}(x)\right)$$

where g is a combination function that could be a simple weighted average, a voting scheme for classification problems, or a more complex stacking approach where a meta-model learns the optimal combination of base model predictions.

Properties and Considerations

Ensembles of surrogates are useful for problems with varying degrees of nonlinearity throughout the design space, and when different surrogate types have complementary strengths for the problem at hand. For example, in computational fluid dynamics, RSMs might capture global trends, while Kriging models better represent local variations. Combining these approaches can give us more accurate predictions across the entire design space.

When building ensembles, heterogeneous ensembles (different model types) usually provide better performance improvements than homogeneous ensembles (same model type with different parameters). A common approach is to combine fast, simple models like polynomial response surfaces with more complex models like neural networks or Kriging to balance computational efficiency with accuracy.

Advantages for Surrogate Modelling

Ensemble methods are generally more accurate, since thy leverage the strengths of different modelling approaches. The ensemble averaging effect reduces the risk of overfitting and increases robustness.

The ensemble framework provides uncertainty quantification through the variance of individual model predictions. This built-in error estimation is useful for risk assessment and reliability analysis. For example, in structural design optimization, regions where ensemble models disagree indicate higher uncertainty, signalling the need for additional high-fidelity simulations at those locations.

Ensembles can be built incrementally, starting with simpler models and adding more complex ones only where needed. This allows us to focus expensive modelling efforts on challenging regions of the design space where simpler models perform poorly.

Limitations and Risk Mitigation

The increased computational cost of training and evaluating multiple models can be significant. To mitigate this:

- Use simpler models as filters, applying more complex models only in regions where additional accuracy is needed
- Implement parallel training and evaluation to reduce time overhead
- Prune redundant models that don't significantly contribute to ensemble performance

Ensemble methods introduce additional hyperparameters related to model weighting and combination strategies. Use cross-validation to determine the best weights. It is also possible to use dynamic weighting, using performance metrics on recent validation data to guide the weighting adjustment.

When combining models with significantly different accuracy, poor models can degrade overall performance if not properly weighted. Consider using a stacked approach where a meta-model learns which base models perform best in different regions of the design space.

3.2.7. Multi Fidelity Models

Model

In multi-fidelity modelling, we represent the model as a combination of information from simulations or experiments at different fidelity levels:

$$\widehat{f_{hi}}(x) = \widehat{f_{lo}}(x) + \delta(x)$$

where $\widehat{f_{hi}}(x)$ is the high-fidelity approximation, $\widehat{f_{lo}}(x)$ is the low-fidelity model, and $\delta(x)$ is a discrepancy function that captures the difference between fidelity levels.

Predictor

The multi-fidelity predictor is formulated as:

$$\widehat{f_{hi}}(x) = \rho \cdot \widehat{f_{lo}}(x) + \delta(x)$$

where ρ is a scaling factor that correlates the low and high-fidelity models, and $\delta(x)$ is typically modelled as a Gaussian process or other surrogate model trained on the residuals between the scaled low-fidelity predictions and high-fidelity data.

Properties and Considerations

Multi-fidelity modelling is useful when high-fidelity simulations are computationally expensive, but lower fidelity approximations are readily available. For example, in thermal management system design, a few detailed 3D conjugate heat transfer simulations can be combined with many inexpensive 1D thermal resistance network calculations to create an accurate surrogate across the entire design space at a fraction of the computational cost.

The approach works best when there is consistent correlation between fidelity levels. Consider a thermal analysis problem where both 1D heat transfer approximations and detailed 3D finite element models are available. Even if the 1D model doesn't capture all spatial effects, as long as it correctly identifies general trends (e.g., how thickness affects heat dissipation), the multi-fidelity approach can leverage this correlation to improve predictions.

The discrepancy function modelling is very important. For problems where the discrepancy varies smoothly across the design space, Gaussian processes work well. When the discrepancy shows more complex patterns, neural networks or tree-based methods might be more appropriate.

Advantages for Surrogate Modelling

Multi-fidelity models significantly reduce the computational cost of surrogate modelling by extracting maximum value from cheaper sources. This approach enables surrogate modelling in applications where the required number of high-fidelity simulations would otherwise be prohibitively expensive.

The framework accommodates hierarchical information from multiple fidelity levels, not just two. For example, in electronics cooling design, models could incorporate 1D resistor networks, 2D computational fluid dynamics, and 3D conjugate heat transfer simulations at different levels of detail, creating a surrogate that spans the entire spectrum.

Multi-fidelity approaches integrate well with existing engineering workflows where simplified models are already used in early design stages.

Limitations and Risk Mitigation

The quality of multi-fidelity models depends heavily on the correlation between fidelity levels. When correlation is weak or inconsistent, consider:

- Segmenting the design space into regions with similar correlation structures.
- Using nonlinear mapping functions between fidelity levels instead of simple scaling.
- Adding intermediate fidelity levels to create a smoother transition between models.

Sampling across fidelity levels is challenging. As a general guideline, allocate resources following a pyramid structure: many low-fidelity samples forming the base, fewer medium-fidelity samples in the middle, and a small number of high-fidelity samples at the top. The exact ratios depend on relative costs and correlations between levels.

The discrepancy model can become unreliable when extrapolating to regions without high-fidelity data. Uncertainty quantification should be used to identify when predictions rely heavily on extrapolated discrepancy terms.

4. Model Evaluation

4.1. Model Validation

Model validation is the process of rigorously assessing whether a surrogate model reliably approximates the behaviour of the original high-fidelity system across the input space. Since surrogate models are often employed to replace computationally expensive simulations or experiments, even small prediction errors can propagate into significant inaccuracies during optimization or decision-making. Thus, establishing robust validation techniques is critical.

A common validation approach involves partitioning the available dataset into training and test sets. The surrogate is calibrated on the training set and its predictive accuracy is then evaluated on the test set using quantitative metrics, such as mean squared error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2) for regression, and accuracy, precision, recall, and F1-score for classification (Forrester & Keane, 2009).

These metrics provide a straightforward measure of how closely the surrogate replicates the high-fidelity model's outputs. For example, (Mehmani, et al., 2015) extensively used k-fold cross validation and bootstrapping to assess error metrics and to compute confidence intervals for surrogate predictions, ensuring that the model's performance is robust to variations in the training data.

Beyond simple error quantification, advanced uncertainty quantification methods are increasingly integrated into model validation. Techniques such as Gaussian process regression not only yield point predictions but also provide predictive variances or confidence intervals, which can be used to identify regions of the input space where the surrogate may be less reliable. This probabilistic framework enables the estimation of error bounds, allowing practitioners to assess the risk associated with surrogate predictions. In a related study, (Ling, Y., Mahadevan, & S., 2013) extended classical Bayesian hypothesis testing to validate both deterministic and stochastic models, offering a systematic method to compare surrogate outputs against experimental or high-fidelity simulation data.

Another layer of validation involves the use of external data. By running additional high-fidelity simulations—data not included in the surrogate's training process—and comparing these with the surrogate's predictions, researchers can assess the model's extrapolation capability. Although similar to the earlier validation approaches, real external data from the source itself can uniquely assess generalisability which may not be fully captured through estimates obtained by a finite dataset, which were previously discussed. This external validation is particularly useful when the system exhibits nonlinearities or when measurement noise is present. In quantitative systems pharmacology (QSP), for instance, surrogate models have been validated by direct comparisons with full QSP model simulations, often resulting in high R^2 scores and low MSE values while dramatically reducing computation times.

4.2. Model Monitoring

To maintain the performance of the surrogate model throughout the process, it is important to regularly validate with new data and monitor critical performance metrics (e.g. MAE, RMSE). This validation allows early detection of new system behaviours that deviate from the conditions under which the model was trained or unexpected changes such as concept drift.

Incremental learning or periodic retraining approaches can be used to minimize performance degradation of the model (Wu, et al., 2019) (Ven, M, Tuytelaars, & Tolias, 2022) (Lopes, et al., 2023). Especially in models that can estimate uncertainty (e.g. Kriging), it should also be monitored whether the prediction intervals reflect model reliability. In the event of unexpected changes that exceed performance thresholds, automatic warning systems can be activated to quickly take steps such as retraining the model or rollback to a previous stable version. This ensures that the operational reliability and accuracy of the surrogate model is maintained throughout the intended application.

4.3. Similarity quantification between engineering datasets

An important aspect to enable the reuse of historical engineering datasets is the ability to retrieve similar designs in an efficient way. This requires an effective measure to quantify the similarity between different simulation datasets. In the state-of-the-art, several approaches have been proposed to compare formally disjoint, heterogeneous datasets.

A first set of approaches is based on transforming the datasets first into an embedding or latent space, in which the datasets can be compared more easily (e.g., due to a reduced dimensionality of the feature space). Prominent state-of-the-art techniques for text or image similarity (Lin, et al., 2022)) are based on contrastive learning (Chen, et al., 2020), a technique for self-supervised representation learning in which a model is trained to learn a representation of examples such that similar samples are closer in the vector space, while dissimilar ones are far apart.

A second main group of techniques is based on Optimal Transport (OT) theory, which is a flexible geometric method for comparing probability distributions. For example, the Optimal Transport Dataset Distance (Alvarez-Melis, D., Fusi, & N., 2020) is an OT-based approach to compare two (labelled) datasets. A practical example of using OT theory for similarity quantification involve comparing datasets from simulations of a power semiconductor on a PCB. These simulations might include temperature data at various points (discrete), material properties (categorical), and other parameters like heat flux (continuous). OT could be used to compare the thermal distributions between the two datasets, accounting for differences in the number of variables or the types of data. This approach would help quantify the similarity in thermal behavior, even with varying amounts of data, and identify key features (such as heat dissipation rates or material properties) that drive differences in performance, ultimately aiding in design optimization for better thermal management. Only limited studies are available that compare the similarity between simulation or engineering datasets. (Linton, et al., 2019) propose a data similarity measure for scientific datasets, based on a combination of similarity metrics, nonlinear dimension reduction, clustering methods and validity measures. Yet, the work is focused on time series data. Engineering datasets typically also include discrete and categorical data and, more challengingly, the datasets might differ in amount and type of variables per dataset. The similarity model of (Kohl, et al., 2020) is based on entropy of physical systems, leveraging a Siamese multiscale feature network. Yet, this metric focusses solely on volumetric simulations.

4.4. Surrogate model reuse through transfer learning

Reusing data-driven models across different datasets for a similar type of problem can be established through transfer learning. Two main types of techniques can be distinguished in the state-of-the-art. In data-based transfer learning, a transfer of the knowledge via the adjustment and the transformation of the data is considered, whereas in model-based transfer learning the source model is (partly) adapted and reused to construct the target model.

In the context of SmartEM, the main challenge in adopting transfer learning is in the heterogeneity of the simulation datasets and models. While related work is limited, there is emerging interest in the state-of-the-art on heterogeneous transfer learning (Weiss, et al., 2016), which allows the data in two domains to be represented with different feature spaces.

Recent examples include the work of (Moon, S., Carbonell, & J., 2017) who propose Attentional Heterogeneous Transfer, a heterogeneous transfer learning approach that selects and attends to an optimized subset of source samples to transfer knowledge from, and builds a unified transfer network that learns from both source and target knowledge, including a novel unsupervised transfer loss. (He, et al., 2020) mainly focus on the difference in data quality and propose Transfer Learning with Weighted Correspondence, which utilizes instance-correspondence (IC) data to adapt the source domain to the target domain. Rather than treating IC data equally, the method assigns solid weights to each IC data pair depending on the quality of the data. The recent work of (Bica, I., Schaar, & M., 2022) uses representation learning to handle heterogeneous feature spaces and proposes an architecture with shared and private layers to transfer information between potential outcome functions across domains.

5. Risk analysis on standards for re-usability and transferability

Understanding the potential risks associated with surrogate modeling is crucial for ensuring effective re-usability and transferability.

Surrogate models are often trained on specific datasets, which means they might not generalize well to other use cases or contexts, even within the same domain. Creating surrogate models requires significant expertise and domain knowledge to ensure good results, whether developing a new physics-based model or a machine learning model. Additionally, surrogate models may expose information about the data they were trained on through adversarial machine learning methods, posing a risk in industrial settings where sensitive intellectual property data might be exposed.

The main risks concerning standards for re-usability and transferability of surrogate models include the lack of standardization, which can hinder the re-usability and transferability within various development groups. A clear model development process, including data preprocessing steps and training parameters, is essential. While there are standards focusing on data formats and interfaces, such as the widely used Modelica FMI software standard, there is a lack of standards for data semantics. These standards are necessary for re-using and transferring surrogate models within and between domains. One possible approach is to start by defining ontologies to describe the model semantics for specific domains, such as printers, shavers, and electron microscopes.

6. Bibliography

Williams, B. A., & Cremaschi, S. (2019). Surrogate model selection for design space approximation and surrogatebased optimization. In *Computer aided chemical engineering* (Vol. 47, pp. 353–358). Elsevier.

Mehmani, A., Chowdhury, S., Meinrenken, C., & Messac, A. (2018). Concurrent surrogate model selection (COSMOS): optimizing model type, kernel function, and hyper-parameters. *Structural and Multidisciplinary Optimization, 57*, 1093–1114.

Li, J., Wang, H., Luo, H., Jiang, X., & Li, E. (2023). A ranking prediction strategy assisted automatic model selection method. *Advanced Engineering Informatics, 57*, 102068.

Kudela, J., & Matousek, R. (2022). Recent advances and applications of surrogate models for finite element method computations: a review. *Soft Computing, 26*, 13709–13733.

Jin, R., Chen, W., & Simpson, T. W. (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and multidisciplinary optimization, 23*, 1–13.

Jia, L., Alizadeh, R., Hao, J., Wang, G., Allen, J. K., & Mistree, F. (2020). A rule-based method for automated surrogate model selection. *Advanced Engineering Informatics, 45*, 101123.

Ben Salem, M., & Tomaso, L. (2018). Automatic selection for general surrogate models. *Structural and Multidisciplinary Optimization, 58*, 719–734.

Arboretti, R., Ceccato, R., Pegoraro, L., & Salmaso, L. (2022). Design of Experiments and machine learning for product innovation: A systematic literature review. *Quality and Reliability Engineering International, 38*, 1131–1156.

Alizadeh, R., Allen, J. K., & Mistree, F. (2020). Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design, 31*, 275–298.

Bala, B., & Behal, S. (2024). A Brief Survey of Data Preprocessing in Machine Learning and Deep Learning Techniques. *2024 8th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 1755-1762.

Ciampiconi, L., Elwood, A., Leonardi, M., Mohamed, A., & Rozza, A. (2023). A survey and taxonomy of loss functions in machine learning.

Forrester, A. I., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 0376-0421.

Qiu, J. (2024). An Analysis of Model Evaluation with Cross-Validation: Techniques, Applications, and Recent Advances . *Advances in Economics, Management and Political Sciences*, 69-72.

Ven, V. d., M, G., Tuytelaars, T., & Tolias, A. S. (2022). Three types of incremental learning. . *Nature Machine Intelligence*, 1185-1197.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., & Fu, Y. (2019). Large scale incremental learning. *IEEE/CVF conference on computer vision and pattern recognition*, 374-382.

Lopes, F., Leal, A., Pinto, M. F., Dourado, A., Schulze-Bonhage, A., Dümpelmann, M., & Teixeira, C. (2023). Removing artefacts and periodically retraining improve performance of neural network-based seizure prediction models. . *Scientific Reports*, 5918.

Loh, & W., L. (1996). On Latin hypercube sampling. *The Annals of Statistics*, 2058-2080.

Helton, J., C., Davis, & F., J. (2003). Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. *Reliability Engineering & System Safety*, 23-69.

Jain, A., Chang, & E., Y. (2004). Adaptive Sampling for Sensor Networks. *Proceedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction With VLDB 2004*, 10-16.

Jin, R., Chen, W., Simpson, & T., W. (2001). Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria. *Structural and Multidisciplinary Optimization*, 1-13.

Hansen, N., Auger, A., Ros, R., . . . D. (2021). COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *Optimization Methods and Software*, 114-144.

Mehmani, A., Chowdhury, S., Messac, & A. (2015). Predictive Quantification of Surrogate Model Fidelity Based on Modal Variations With Sample Density. *Structural and Multidisciplinary Optimization*, 253-373.

Ling, Y., Mahadevan, & S. (2013). Quantitative Model Validation Techniques: New Insights. *Reliability Engineering & System Safety*, 217-231.

Myers, R., C., Augustin, F., Huard, J., . . . C., M. (2023). Using Machine Learning Surrogate Modeling for Faster QSP VP Cohort Generation. *CPT: Pharmacometrics & Systems Pharmacology*, 1047-1059.

Bica, I., Schaar, v. d., & M. (2022). Transfer Learning on Heterogeneous Feature Spaces for Treatment Effects Estimation. *arXiv Preprint* , arXiv:2210.06183.

He, Y., Jin, X., Ding, G., . . . S. (2020). Heterogeneous Transfer Learning With Weighted Instance-Correspondence Data. *Proceedings of the AAAI Conference on Artificial Intelligence* , 4099-4106.

Moon, S., Carbonell, & J., G. (2017). Completely Heterogeneous Transfer Learning With Attention—What and What Not to Transfer. . *IJCAI*, 1-2.

Weiss, K., Khoshgoftaar, T., M., Wang, & D. (2016). A Survey of Transfer Learning. *Journal of Big Data*, 1-40.

Kohl, G., Um, K., Thuerey, & N. (2020). Learning Similarity Metrics for Numerical Simulations. *International Conference on Machine Learning* , 5349-5360.

Lin, N., Qin, G., Wang, J., . . . D. (2022). Research on the Application of Contrastive Learning in Multi-Label Text Classification. *arXiv Preprint*, arXiv:2212.00552.

Chen, T., Kornblith, S., Norouzi, M., . . . G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *International Conference on Machine Learning*, 1597-1607.

Alvarez-Melis, D., Fusi, & N. (2020). Geometric Dataset Distances via Optimal Transport. *Advances in Neural Information Processing Systems*, 21428-21439.

Linton, P., Melodia, W., Lazar, A., . . . K. (2019). Understanding Data Similarity in Large-Scale Scientific Datasets. *IEEE International Conference on Big Data (Big Data)*, 4525-4531.

Alexander I, J., Forrester, A. S., & Andy, J. K. (2008). *Engineering Design via Surrogate Modelling A Practical Guide.* John Wiley & Sons Ltd.