Innovation in Computer and Data Sciences
2(2), 79–95
DOI: https://doi.org/10.64389/icds.2026.02294
https://sphinxsp.org/journal/index.php/icds/

Research article

Machine Learning Based Prediction of Alphabetic Optimality Criteria in Central Composite Designs

L. O. Ngonadi[1,*], **Sydney I. Onyeagu**[1], **F. C. Eze**[1], **Ifunanya Lydia Omeje**[1]

[1,*] Department of Statistics, Faculty of Physical Sciences, Nnamdi Azikiwe University, P.O. Box 5025 Awka, Nigeria

* **Correspondence:** lo.ngonadi@unizik.edu.ng

ARTICLE INFO

Keywords:
Optimality Criteria
Machine Learning
Central Composite Design
Prediction
Efficiency

Mathematics Subject Classification:
62M10, 62P05, 91G70, 91B84,60G10

Important Dates:
Received: 26 January 2026
Revised: 17 February 2026
Accepted: 23 February 2026
Online: 27 February 2026

ABSTRACT

This study analyses the effectiveness of machine learning approaches in predicting optimality criteria of Central Composite Designs (CCDs). The study focuses on three types of CCD: Rotatable Central Composite Design (RCCD), Spherical Central Composite Design (SCCD), and Face-Centered Central Composite Design (FCCD) for dimensions varying from $k = 3$ to $k = 10$. The purpose is to find out if geometric and structural features of CCD can be used as predictors of design efficiency. Geometric variables such as the number of factors, type of CCD, axial distance, and total number of experimental trials were utilised to forecast A-efficiency, D-efficiency, and G-efficiency. Linear Regression, Random Forest, XGBoost, Support Vector Regression (SVR), k-Nearest Neighbours (KNN), and Decision Tree. Six machine learning methods were used and evaluated by the value of R^2 and RMSE. It was found that machine learning nonlinear algorithms significantly outperformed the linear regression algorithm on both low- and high-dimensional datasets. The best results were obtained with SVR and XGBoost where the prediction of D-efficiency approached almost deterministic ($R^2 \approx 0.99$). Additionally, the total number of runs turned out to be the most important predictor of all three criteria.

1. Introduction

In order to build adequate representations for increasingly complex experimental setups that involve multiple variables and nonlinear relationships, it is crucial to develop simplified mathematical models for high dimensional and nonlinear interactions between experimental inputs and outputs. Among classical approaches in experimental design, Response Surface Methodology (RSM) still remains popular due to its

versatility. However, RSM based approaches, especially Central Composite Designs (CCD), suffer from significant computational limitations when dealing with numerous factors, as the number of design parameters increases rapidly while the information matrix loses its conditioning. Therefore, traditional efficiency measures that are based on matrix spectra might not always provide a reliable estimate of the actual performance of the design ([12, 11]).

With machine learning (ML) having emerged as a promising approach for modeling highly complex phenomena and optimizing performance metrics, researchers have been able to leverage ML algorithms that use ensemble approaches in combination with kernel methods and gradient boosting to fit nonlinear input-output relationships without relying on analytical expressions. This makes ML an increasingly popular choice across various branches of engineering and experimental sciences, including material science and experimental physics ([8, 18]).

Recently, numerous research papers demonstrated that ML-based approaches could be leveraged in experimental design to significantly boost predictive accuracy while substantially reducing experimental costs by modeling nonlinear processes effectively. Hybrid approaches that combine both RSM and ML often outperform their strictly statistical counterparts ([9, 14]). ML based workflows have also proven useful in optimizing manufacturing operations and identifying the optimal setup configurations of a particular system ([3, 2]).

Despite being applied primarily to solve engineering and scientific tasks related to design optimization and outcome prediction, ML has not yet found extensive use in validating the underlying theoretical principles behind various experimental design methodologies. In particular, this applies to CCD. Historically, the performance of CCD was evaluated based on alphabetic optimality criteria, namely, D-, A-, and G-efficiency that were calculated with the help of the information matrix ($X^\top X$). Alphabetic optimality criteria rely on the spectral properties of the matrix, specifically, its determinant (D-efficiency), the sum of inverse eigenvalues (A-efficiency), and maximum prediction variance (G-efficiency) which are directly correlated with the smallest eigenvalue. As the number of dimensions in CCD increases, the difference between eigenvalues grows larger and, hence, the reliability of efficiency measures drops sharply[4].

Recent work by [13] further examined the behaviour of Central Composite Designs under both classical and regularized optimality criteria in high-dimensional settings. Their findings highlighted the increasing challenges associated with evaluating design efficiency as dimensionality grows and demonstrated the usefulness of regularization techniques for improving the stability of information matrix-based criteria.

According to recent advances in high-dimensional statistics, the distribution of eigenvalues is crucial for the stability and predictive power of a model. This highlights the importance of establishing a relationship between the geometric features of a design and the spectral properties of the information matrix in high dimensions, which could be achieved with the help of ML techniques. Advances in interpretable machine learning (IML) enable researchers to explore the structural patterns behind predictive models in more detail and provide additional insights into their functioning ([10, 16]).

At the same time, there is still a major research gap regarding the interplay between the geometric properties of a design and the resulting efficiency measures. Specifically, the following chain of dependencies remains unexplored Design Geometry → Spectral Properties → Efficiency Measures.

Proving the presence of a deterministic relationship between efficiency measures and the corresponding geometric parameters of a CCD will help determine whether efficiency measures are sensitive to instability in high dimensions. Furthermore, it will facilitate the development of data-driven methodologies for evaluating the efficiency of designs without resorting to exclusively analytical methods.

A-, D-, and G-efficiency can be figured out using the information matrix, but our study isn't trying to replace those classic methods. We are looking at if machine learning can pick up on the hidden link between the shape of Central Composite Designs and their effectiveness scores. If it works, these models could speed things up by acting as quick stand-ins, letting us check big groups of designs without the usual intense math work. Plus, machine learning lets us explore how the structure of the designs ties into the matrix's properties and how efficient everything is. So, we use these models to predict and also to back up the theories behind designing response surfaces

Given the above considerations, the current study focuses on leveraging ML as a means of validating the theoretical relationships in RSM and demonstrating the dependence of CCD efficiency measures on geometric features.

The novelty of the paper is that, unlike other works where ML is applied to solve various problems within experimental design, the focus will be placed on validating the underlying theory of optimality criteria. Moreover, rather than simply developing algorithms to optimize designs, this study seeks to verify whether ML can be used to uncover structural patterns within CCDs.

2. Methodology

The methodology gives an overview of the machine learning approach used in predicting the optimality criteria of Central Composite Design (CCD) models. The method provides an analysis of the structure of CCD models, the geometric predictors, the machine learning algorithm, preprocessing, and the metrics used.

2.1. Problem Formulation

This section examines whether there exist relationships between optimality criteria for CCDs and their geometric and structural properties based on machine learning methods.

Let $x_i \in \mathbb{R}^d$ be a feature vector representing i-th CCD configuration, where d is the number of variables characterizing the design structure. Let $y_i^{(j)} \in \mathbb{R}$ be a response corresponding to j-th optimality criterion, where $j \in \{D, A, G\}$ represents D-efficiency, A-efficiency, and G-efficiency, respectively.

The task is formulated as a problem of regression where we estimate the function

$$f_j : \mathbb{R}^d \to \mathbb{R}$$

such that

$$y_i^{(j)} = f_j(x_i) + \varepsilon_i, \tag{1}$$

where ε_i is an error with $E[\varepsilon_i] = 0$ and $\mathrm{Var}(\varepsilon_i) < \infty$. Function $f_j(\cdot)$ is unknown and is approximated using machine learning methods. Models for different optimality criteria are built separately.

2.2. Design Matrix and Information Matrix

Each CCD is described via second-order response surface model, namely,

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon, \tag{2}$$

where $y \in \mathbb{R}$ is the response, $x_i \in \mathbb{R}$ is a coded value of the i-th factor, and $\beta_0, \beta_i, \beta_{ii}, \beta_{ij} \in \mathbb{R}$ are regression coefficients ([12, 11]).

Number of parameters p of the model is

$$p = 1 + k + \frac{k(k+1)}{2}. \tag{3}$$

Let $X \in \mathbb{R}^{N \times p}$ be a design matrix, where N is a number of experimental runs. Information matrix is defined as

$$M = X^\top X, \tag{4}$$

where $X^\top$ is a transpose of matrix X. This matrix contains information for estimating model parameters [1].

2.3. Regularisation of the Information Matrix

In high-dimensional settings, the matrix M can become singular or ill-conditioned due to multicollinearity. In order to make it invertible, ridge regularisation is performed:

$$M_\lambda = X^\top X + \lambda I, \tag{5}$$

where $\lambda > 0$ is a regularisation coefficient and $I \in \mathbb{R}^{p \times p}$ is the identity matrix. It guarantees positive definiteness of M_λ [6].

The regularization parameter was selected through a grid search over the range $\lambda \in \{10^{-10}, 10^{-9}, \ldots, 10^{-2}\}$. To ensure consistency across dimensions, the regularization coefficient was scaled according to the average magnitude of the information matrix:

$$\lambda_{\text{eff}} = \lambda \frac{\text{tr}(X^\top X)}{p}, \tag{6}$$

where p denotes the number of model parameters. This scaling ensures that regularization remains a constant proportion of the information contained in the design matrix regardless of dimensionality. The optimal value of λ was selected as the value that maximized the mean G-efficiency across all generated designs.

2.4. Optimality Criteria

Optimality criteria which serve as target variables in machine learning models are calculated based on the regularised information matrix M_λ.

Definition of D-efficiency is

$$D = \left(\frac{\det(M_\lambda)}{N^p} \right)^{1/p}, \tag{7}$$

where $\det(M_\lambda)$ is the determinant of M_λ, N is the number of runs, and p is the number of parameters [15].

A-efficiency is defined as

$$A = \frac{p}{\text{trace}(M_\lambda^{-1})}, \tag{8}$$

where $\text{trace}(M_\lambda^{-1})$ is the sum of diagonal elements of the inverse information matrix [1].

G-efficiency is defined as

$$G = \frac{1}{\max_{x \in R} x^\top M_\lambda^{-1} x}, \tag{9}$$

where $x \in \mathbb{R}^p$ is a point from design region R. Quadratic form $x^\top M_\lambda^{-1} x$ corresponds to prediction variance at this point [15].

2.5. Synthetically Generated Design Data

The data used for this study are 1080 synthetically generated design data. The data comprises of the number of Factors (k=3,4,...,10), type of Central Composite Design (RCCD, SCCD and FCCD), Replication of Factorial Points(R_f =1,2,3),Replication of Axial Points(R_α =1,2,3), Type of Factorial Core(full factorial), Axial Distance(α), Number of Factorial Runs(N_f), Number of Axial Runs (N_α), Number of Center Runs (N_c= 1, 3, 5, 7,10), and Total Number of Runs(N).

2.6. Feature Representation

Each CCD is represented by the feature vector

$$x_i = (k, R_f, R_\alpha, N_c, \alpha, N_f, N_\alpha, N, T), \tag{10}$$

where k is a number of factors, R_f and R_α represent replication of factorial and axial points, N_c is the number of centre runs, α is the axial distance, N_f and N_α are the numbers of factorial and axial runs, N is a total number of runs, and T represents the CCD type.

Continuous features are standardised as follows

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}, \tag{11}$$

where μ_j and σ_j are the mean and standard deviation of j-th feature, respectively.

2.7. Models Fitted

Six machine learning algorithms are fitted and compared:

2.7.1. Linear Regression Model

Linear regression is a parametric modeling approach that assumes a linear relationship between a response variable and a set of predictors. The model expresses the conditional mean of the response as a linear combination of the input variables [7]. The functional form of the model is given by

$$\hat{Y}(z) = \beta_0 + \sum_{j=1}^{p} \beta_j z_j \tag{12}$$

where $z = (z_1, z_2, \ldots, z_p)$ denotes the predictor vector, β_0 is the intercept term, and β_j are the regression coefficients associated with each predictor. Estimation of the parameters is typically achieved through the method of ordinary least squares, which minimizes the residual sum of squares between observed and predicted values. The objective function is defined as

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{13}$$

where y_i represents the observed response and $\hat{y}_i$ denotes the corresponding fitted value. Under standard assumptions, the least squares estimator is unbiased and has minimum variance among linear estimators [7].

2.7.2. Random Forest (RF)

A collection of decision trees trained on bootstrap samples of the data, predictions averaged across the decision trees [2]

$$\hat{Y}_{RF}(z_0) = \frac{1}{T} \sum_{t=1}^{T} \hat{f}_t(z_0) \tag{14}$$

where T is the total number of decision trees in the collection and $\hat{f}_t(z_0)$ is the prediction from the t-th tree trained on a bootstrap sample of the data. The averaging procedure helps avoid overfitting [2].

2.7.3. XGBoost (Gradient Boosting)

An ensemble of regression trees that are fit sequentially to the residuals of previous regression trees, with regularization to avoid overfitting [3]. XGBoost is a proven predictive modeling technique in diverse scientific areas.

$$\hat{Y}_{XGB}(z_0) = \sum_{m=1}^{M} f_m(z_0) \tag{15}$$

where M is the total number of boosting rounds and $f_m(z_0)$ is the m-th tree fitted to the residuals of the previous $m-1$ trees.

The objective function is given by

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \sum_{m=1}^{M} \Omega(f_m) \tag{16}$$

where $\Omega(f_m)$ is the penalty for the complexity of tree m given by the number of its leaves and the size of their weights [3].

2.7.4. Support Vector Regression (SVR)

The method involves mapping predictor variables into a high-dimensional feature space and fitting a linear model in this feature space. As a result, the method enables nonlinear prediction of responses in the original predictor space.

$$\hat{Y} = w^\top \phi(z) + b \tag{17}$$

subject to the constraint that the absolute difference between the prediction and observed values does not exceed the value of a given tolerance, where $\phi(z)$ maps the predictor vector into a higher dimensional feature space, w is the weight vector, and b is the bias term.

The radial basis function kernel is used:

$$\kappa(z_i, z_j) = \exp\left(-\gamma_s \|z_i - z_j\|^2\right) \tag{18}$$

where $\gamma_s > 0$ specifies the width of the kernel, resulting in nonlinear prediction in the original predictor space [17].

2.7.5. K-Nearest Neighbours (KNN)

Non-parametric method predicting a response for a new observation as an average response of its k nearest neighbours in the predictor space.

$$\hat{Y}(z_0) = \frac{1}{K} \sum_{i \in N_K(z_0)} y_i \tag{19}$$

where $\hat{Y}(z_0)$ is the predicted efficiency for a new design configuration z_0, K is the number of nearest neighbours, and $N_K(z_0)$ is the set of K training observations closest to z_0 in the predictor space [19].

2.7.6. Decision Tree (DT)

Decision trees are nonparametric models that partition the predictor space into mutually disjoint regions and predict constants within those regions [2]. In other words, the decision tree model is represented by a piecewise constant function of predictors:

$$\hat{Y}(z) = \sum_{m=1}^{M} c_m \cdot 1(z \in R_m) \tag{20}$$

where M is the number of terminal nodes (leaves), R_m denotes the m-th region of the predictor space, c_m is the predicted value in region R_m, and $1(\cdot)$ is the indicator function. The tree is built via recursive partitioning with a criterion based on minimizing the residual sum of squares. Given two disjoint regions R_1 and R_2, this criterion is

$$\sum_{i:z_i \in R_1} (y_i - c_1)^2 + \sum_{i:z_i \in R_2} (y_i - c_2)^2 \tag{21}$$

Recursive partitioning continues up until a certain termination criterion is met, after which the pruning procedure can be applied to reduce complexity and avoid overfitting.

2.8. Model Training Procedure

Machine learning algorithms are trained to discover the underlying functional relationship $f_j(x)$ between geometric design variables and optimality criteria. The following training procedure is rigorously specified for consistent and reliable results.

Let the dataset be given as

$$D = \{(x_i, y_i)\}_{i=1}^{n}, \tag{22}$$

where n represents the number of CCD configurations, $x_i \in \mathbb{R}^d$ represents the feature vector corresponding to the i-th observation, and $y_i \in \mathbb{R}$ is the optimality criterion of this observation.

2.8.1. Data Splitting

To assess the generalisation capability of the machine learning models, the dataset D is split into a training set D_{train} and a testing set D_{test} as follows: $D_{train} \cup D_{test} = D$ and $D_{train} \cap D_{test} = \emptyset$. A stratified random split is performed with the proportion $|D_{train}| = 0.8n$ and $|D_{test}| = 0.2n$.

The training dataset is used for model training, whereas the testing dataset is utilized for out-of-sample performance assessment [5].

2.8.2. Cross Validation

To avoid overfitting and improve generalisability, K-fold cross validation is employed. The dataset D_{train} is partitioned into $K = 5$ mutually exclusive subsets as follows:

$$D_{train} = \bigcup_{k=1}^{K} D_k. \tag{23}$$

Given a particular subset k, the model is trained using the subset $D_{train} \setminus D_k$ and tested using D_k.

The cross validation performance estimate is calculated as

$$\hat{R} = \frac{1}{K} \sum_{k=1}^{K} R_k, \tag{24}$$

where R_k is the performance metric associated with fold k. This technique increases the stability of the performance estimation and assists in proper hyperparameter selection [5].

2.8.3. Feature Scaling and Preprocessing

Continuous predictor variables are preprocessed via standardization. The transformation is defined as follows:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}, \tag{25}$$

where x_j represents the original value of the j-th predictor variable, μ_j represents the mean of the variable over the training dataset, and σ_j is the corresponding standard deviation.

This preprocessing step is especially important for distance-based and kernel-based algorithms such as k-Nearest Neighbors and Support Vector Regression.

For categorical predictor variables, such as CCD family types, one-hot encoding is used to generate binary indicator variables.

2.8.4. Hyperparameter Selection

Hyperparameters control how complex a model is and affect its predictions. To find the best values, we used grid search along with five-fold cross-validation. We figured out which setup performed the best based on cross-validated prediction errors and stuck with that one.

For Random Forest regression, we landed on 800 decision trees (ntree = 800) and three predictors per split (mtry = 3). With XGBoost regression, the ideal setup included a learning rate of $\eta = 0.06$, a tree depth

of five (max_depth = 5), a subsampling ratio of 0.80 (subsample = 0.80) and a column sampling ratio of 0.80 (colsample_bytree = 0.80), and 600 boosting rounds (nrounds = 600).

As for Support Vector Regression (SVR), we chose a radial basis function kernel with a penalty parameter C of 10, a kernel bandwidth γ of 0.10, and an ϵ-insensitive loss parameter ϵ of 0.01. The k-Nearest Neighbours (KNN) model had its sweet spot when k equaled ten. For the Decision Tree model, we settled on a complexity parameter of 0.001 (cp = 0.001) and a max tree depth of ten (maxdepth = 10). Linear Regression didn't need any adjustments beyond the defaults. These selected hyperparameter values were then used across the board for predicting A-efficiency, D-efficiency, and G-efficiency.

2.8.5. Model Fitting

Given the training dataset D_{train}, machine learning models output predictions via a predictive function $\hat{f}(x)$.

Model fitting for regression tasks is achieved by minimizing the loss function L. The widely used loss function for regression is the residual sum of squares:

$$L = \sum_{i=1}^{n} \left(y_i - \hat{f}(x_i)\right)^2, \tag{26}$$

where y_i is the value of the observed response and $\hat{f}(x_i)$ is the predicted response.

Minimization of this loss function is performed by different approaches depending on a specific machine learning algorithm. These include recursive partitioning for decision tree models, constrained optimization for kernel-based models, and least squares estimations for linear models.

2.9. Model Evaluation

The primary performance metric used in this study is the coefficient of determination, which is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{27}$$

where $\hat{y}_i$ is the predicted response value, y_i is the observed value of the response, and $\bar{y}$ is the sample mean. The coefficient of determination shows how much variation in the response variable is explained by the model.

Root Mean Squared Error (RMSE) is given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{28}$$

This metric represents an absolute prediction error, providing an estimate of model performance.

Model evaluation involves ranking them according to their predictive performance. Let R_m^2 be the coefficient of determination associated with model m. Then, the optimal model is identified via

$$R_m^2 = \max_{m \in M} R_m^2, \tag{29}$$

where M is the set of machine learning models considered in this study. This evaluation strategy is employed to rank different algorithms for each optimality criterion.

Feature importance is assessed for tree-based machine learning algorithms. The decrease in impurity attributed to predictor j at tree t is denoted as $\Delta I_{j,t}$.

The overall feature importance measure I_j is given by the sum of all such impurity decreases across all trees as

$$I_j = \sum_{t \in T} \Delta I_{j,t}, \tag{30}$$

where T is the set of trees in the ensemble. This feature importance analysis enables identification of geometric features contributing the most to prediction accuracy.

To explore model consistency across dimensional regimes, the dataset D is divided into low dimensional and high dimensional datasets. Namely, for CCDs with $k \leq 6$ dimensions, low-dimensional models are obtained, while models with $k > 6$ are considered to be high-dimensional. Separate machine learning models are trained for these datasets. In order to quantify the degree of model overfitting, the difference between train and test performance is calculated as $\Delta = R^2_{train} - R^2_{test}$.

3. Results and Discussion

The results and discussion illustrate the accuracy with which the machine learning models can predict A-, D-, and G-efficiency in Central Composite Designs across various dimensional classes. Additionally, this part of the research examines the impact of geometric attributes on design optimality through model performance measures and variable importance.

3.1. Predictive and Structural Validation Across Dimensional Bands

To investigate whether geometric features alone are sufficient to predict alphabetic efficiency criteria across different dimensionality we partition the dataset into two groups which are Low dimensionality for $k = 3$ to 6 and High dimensionality for $k = 7$ to 10. The efficiency criteria (A-efficiency, D-efficiency, and G-efficiency) were modelled using the following geometric features: Number of factors (k),Type of CCD (FCCD, RCCD, SCCD), Axial distance (α) and Total number of runs (N)

Six machine learning algorithms were compared: Linear Regression, Random Forest, XGBoost, Support Vector Regression (SVR), k-Nearest Neighbours (KNN) and Decision Tree (DT). An 80/20 train test split was used to train and evaluate the models. Performance was assessed using the coefficient of determination (R^2) and the Root Mean Squared Error (RMSE).

3.2. A-Efficiency

From Table 1 , it is observed that nonlinear models outperform linear regression in both dimensional groups. In the low-dimensional group, the SVR model achieved the highest predictive performance with $R^2 \approx 0.79$. In the high-dimensional band, SVR again performed best with $R^2 \approx 0.91$.

Table 1. Model Comparison for A-Efficiency Across Dimensional Groups

	Low Dimension (k = 3 to 6)		High Dimension (k = 7 to 10)	
Model	RMSE	R^2	RMSE	R^2
LM	0.1334	0.6522	0.1025	0.8442
RF	0.1237	0.7171	0.0777	0.9094
XGBoost	0.1252	0.6969	0.1072	0.8467
SVR	0.1058	**0.7851**	0.0777	**0.9126**
KNN	0.1993	0.2188	0.1650	0.6461
Tree	0.1312	0.6588	0.0904	0.8757

The predictive relationships are illustrated in Figure 1 (Low dimension) and Figure 2 (High dimension), which display the observed versus predicted values for the best performing model in each group. In both figures, majority of the predicted values lie close to the reference line, confirming a nonlinear relationship between A-efficiency and geometric structure.

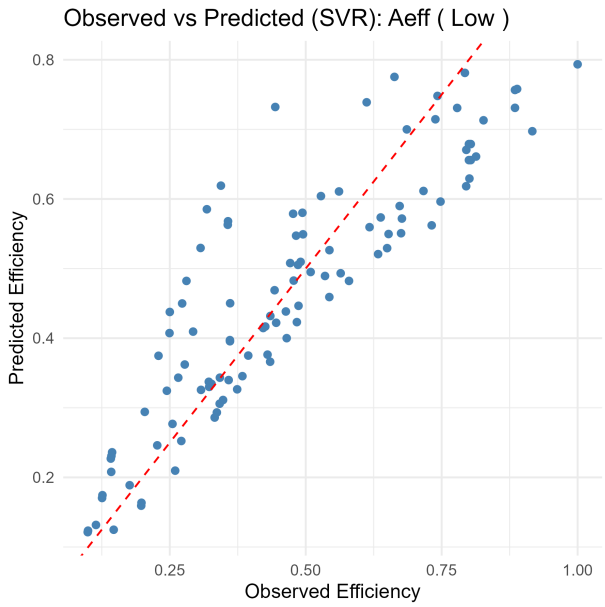

Figure 1. Observed vs Predicted A efficiency SVR model for Low dimensional group

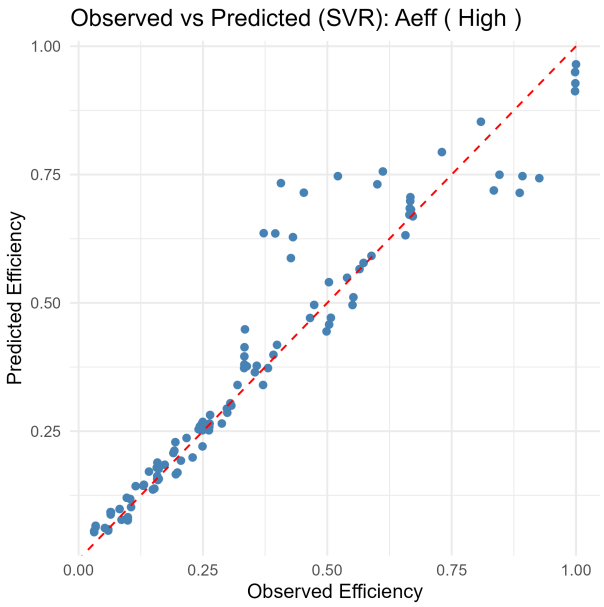

Figure 2. Observed vs Predicted A efficiency SVR model for high dimensional group

The structural dominance of the geometric components is presented in Figure 3, which shows the variable importance for both dimensional group simultaneously. In the low dimension, total run size (N) exhibit high influence on A-efficiency.

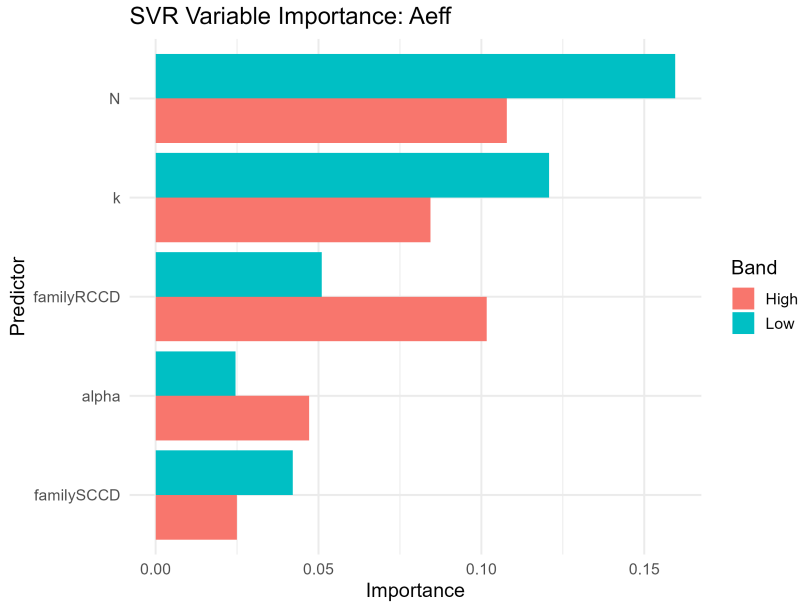

Figure 3. Variable Importance for A-Efficiency Across Low (k = 3–6) and High (k = 7–10) Dimensional band

Also, in the high dimensional group, total run size is still the dominant predictor. This indicates the importance of number of runs in a design experiment.

3.3. D-Efficiency

The near deterministic predictive behaviour of D-efficiency is shown in Table 2. In the low-dimensional band, XGBoost achieved $R^2 \approx 0.99$, while in the high-dimensional band, SVR achieved $R^2 \approx 0.98$. These results suggest that D-efficiency is almost entirely governed by geometric structure.

Table 2. Model Comparison for D-Efficiency Across Dimensional Groups

	Low Dimension (k = 3 to 6)		High Dimension (k = 7 to 10)	
Model	RMSE	R^2	RMSE	R^2
LM	0.0836	0.8435	0.0797	0.8944
RF	0.0516	0.9591	0.0431	0.9702
XGBoost	0.0261	**0.9854**	0.0383	0.9761
SVR	0.0311	0.9801	0.0371	**0.9783**
KNN	0.1474	0.5432	0.1360	0.7388
Tree	0.0764	0.8712	0.0668	0.9248

This strong relationship is visually confirmed in Figure 4 (Low dimension) and Figure 5 (High dimension), where the predicted values lie very close to the reference line in both cases.

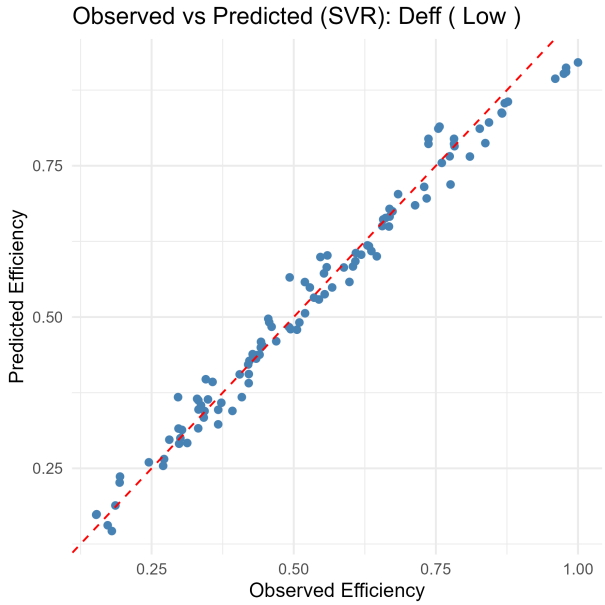

Figure 4. Observed vs Predicted D efficiency SVR model for low dimensional group

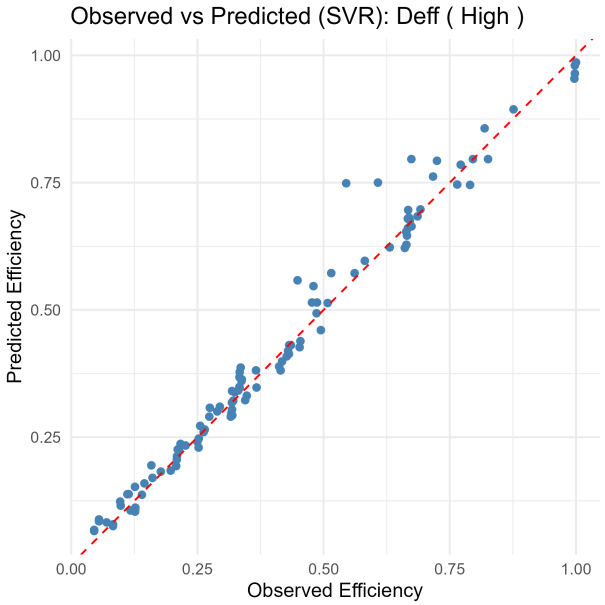

Figure 5. Observed vs Predicted D efficiency SVR model for high dimensional group

Variable importance results are presented in Figure 6. For both dimensional group, total run size (N) is observed to be the dominant predictor.

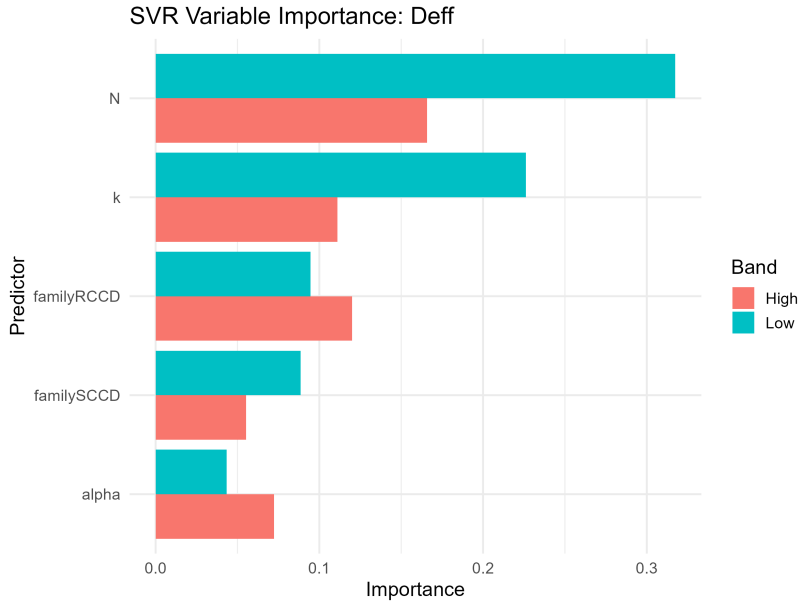

Figure 6. Variable Importance for D-Efficiency Across Low (k = 3–6) and High (k = 7–10) Dimensional

This behaviour is consistent with the determinant formulation of D-efficiency, which reflects the global spectral spread of the information matrix.

3.4. G-Efficiency

The model comparison results for G-efficiency are presented in Table 3. In the low-dimensional band, Random Forest achieved $R^2 \approx 0.90$, while in the high-dimensional band, SVR achieved $R^2 \approx 0.88$. Although predictive performance remains strong, it is slightly lower than that observed for D-efficiency.

Table 3. Model Comparison for G-Efficiency Across Dimensional Group

	Low Dimension (k = 3 to 6)		High Dimension (k = 7 to 10)	
Model	RMSE	R^2	RMSE	R^2
LM	0.1385	0.7498	0.1064	0.8194
RF	0.0916	**0.8998**	0.0875	0.8753
XGBoost	0.0905	0.8922	0.1176	0.7985
SVR	0.1047	0.8694	0.0876	**0.8760**
KNN	0.2293	0.3587	0.1638	0.6082
Tree	0.1025	0.8792	0.1020	0.8285

The observed versus predicted relationships are illustrated in Figure 7 (Low dimension) and Figure 8 (High dimension). Compared to the D-efficiency plots, a slightly greater dispersion around the identity line is observed, reflecting the localised nature of G-efficiency, which depends on maximum prediction variance rather than global determinant structure.

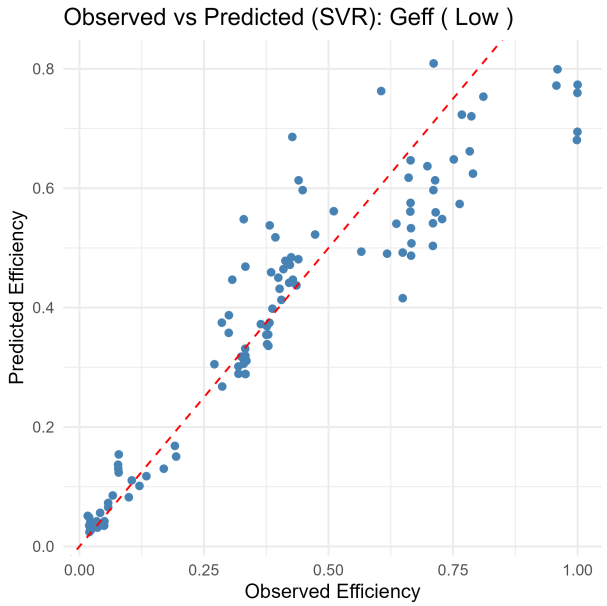

Figure 7. Observed vs Predicted G efficiency SVR model for low dimensional group

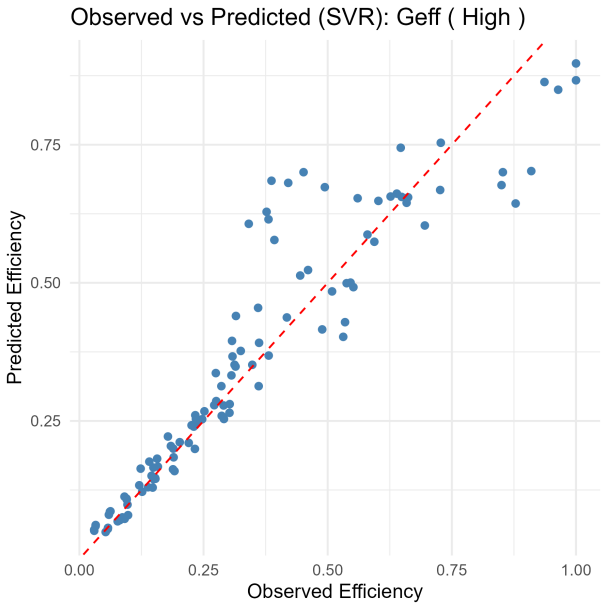

Figure 8. Observed vs Predicted G efficiency SVR model for high dimensional group

The structural dominance patterns are shown in Figure 9. In the low-dimensional regime, G-efficiency behaviour is influenced primarily by number of runs. Also, in the high dimensional group, number of runs

remains dominant.

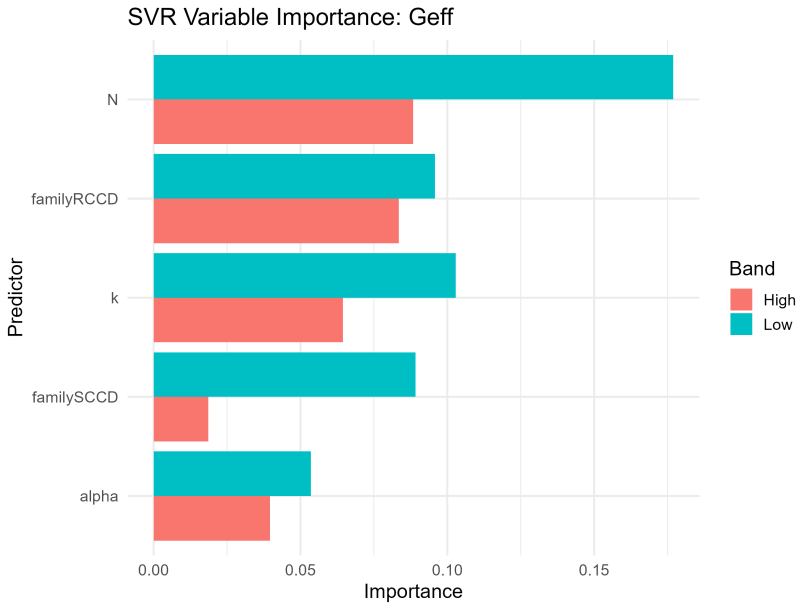

Figure 9. Variable Importance for G-Efficiency Across Low (k = 3–6) and High (k = 7–10) Dimensional

3.5. Overfitting Assessment

To assess overfitting, we calculated the difference between the training and testing performance for the SVR model, using $\Delta = R^2_{\text{train}} - R^2_{\text{test}}$ as shown in Table 4 The discrepancy in R^2 values between training and testing is tiny for all cases, under 0.02, showing the model performs well on new data. Sometimes, the testing R^2 even topped the training R^2, this can result from the train-test split randomness but still confirms no overfitting occurred.

Table 4. Overfitting Assessment for the SVR Model Across Efficiency Criteria and Dimensional Bands

Response	Band	Train R^2	Test R^2	Δ
D-efficiency	Low	0.9809	0.9801	0.0008
D-efficiency	High	0.9785	0.9783	0.0002
A-efficiency	Low	0.7660	0.7851	-0.0191
A-efficiency	High	0.9097	0.9126	-0.0029
G-efficiency	Low	0.8588	0.8694	-0.0106
G-efficiency	High	0.8827	0.8760	0.0067

4. Conclusion

The study shows that machine learning models can accurately predict A-efficiency, D-efficiency, and G-efficiency in Central Composite Designs based on their geometric features. Nonlinear models like Support Vector Regression and XGBoost performed way better than Linear Regression, no matter if the data was

low- or high-dimensional. D-efficiency was the easiest to predict, coming really close to an R^2 of 0.99, which suggests it strongly depends on the CCD's geometry. The total number of runs emerged as the most important factor for all criteria and dimension groups.

Also, the similar accuracy rates between training and testing imply that these models aren't overfitted and work well overall. This means that not only are machine learning algorithms great at catching patterns, but they could also become a handy tool for figuring out design efficiencies.

That said, the study has some limitations. It only used synthetic CCD data and only looked at dimensions from $k = 3$ to $k = 10$. So, there's room for improvement. For instance, future work could check how these models fare with real-world datasets and different types of designs. Also, researchers might look into trying out other regularization strategies, and using an alternative approach such as permutation importance to improve feature importance analysis.

Conflict of Interest

The authors declare there is no existing conflict of interest.

References

1.Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS*. Oxford University Press, Oxford.

2.Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

3.Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

4.Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. Johns Hopkins University Press, 4th edition.

5.Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, 2nd edition.

6.Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

7.James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning*. Springer, 2nd edition.

8.Jumper, J., Evans, R., Pritzel, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

9.Luján-Moreno, G. A., Howard, P. R., Rojas, O. G., and Montgomery, D. C. (2018). Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications*, 109:195–205.

10.Molnar, C. (2022). *Interpretable Machine Learning*. Lulu.com, 2nd edition.

11.Montgomery, D. C. (2017). *Design and Analysis of Experiments*. Wiley, 9th edition.

12.Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M. (2016). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 4th edition.

13.Ngonadi, L. O., Diab, L. S., Onyeagu, S. I., Eze, F. C., Obulezi, O. J., and Aldukeel, A. (2026). High-dimensional evaluation of central composite designs under classical and regularized optimality criteria. *Symmetry*, 18(5):814.

14.Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P., and Buddhakulsomsiri, J. (2022). Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ann, svm, and dbn. *Mathematical Problems in Engineering*, 2022:1–17.

15.Pukelsheim, F. (2006). *Optimal Design of Experiments*. SIAM.

16.Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

17.Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

18.Vinuesa, R., Azizpour, H., Leite, I., et al. (2020). The role of artificial intelligence in achieving the sustainable development goals. *Nature Communications*, 11(1):233.

19.Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11):218.