Evaluating Modelling Performance: Sensitivity Analysis of Data Volume in Industrial Batch Processes

Simon Mählkvist [*,**] **Thomas Helander** [*]
Konstantinos Kyprianidis [**]

[*] *Kanthal AB, Hallstahammar, Sweden*
(e-mail: simonmkvst@gmail.com)
[**] *Future Energy Center, Mälardalen University, Västerås, Sweden*

Abstract: This study conducts a sensitivity analysis to evaluate the influence of varying data volumes on model performance within multi-product batch processes in the iron and steel industry. Nine machine learning models, encompassing both ensemble and parametric methods, were rigorously tested using a data withholding approach. The results demonstrate that ensemble models, particularly Random Forest and Gradient Boosting, consistently outperformed parametric models across different data volumes, showcasing superior generalisation and robustness to outliers. These findings underscore the importance of careful model selection and comprehensive data preprocessing in enhancing model performance and suggest that ensemble methods are particularly well-suited for complex industrial applications where data quality and volume are critical.

Keywords: Machine Learning, Model selection, Performance evaluation, Data volume sensitivity, Iron and steel industry, Industrial batch processes,

1. INTRODUCTION

The iron and steel industry, a cornerstone of global industrial development, is responsible for approximately 7.2% of global Green House Gas (GHG) emissions, highlighting its significant contribution to climate change (Ritchie et al., 2020). With global steel production anticipated to rise by approximately 30% by 2050, the demand for innovation and sustainable practices in this sector has become increasingly urgent (Yoro and Daramola, 2020). Thus, the advancement of more efficient production methods is not only an environmental imperative but also essential for the industry's long-term viability. Moreover, the development of accurate and reliable models can significantly contribute to reducing waste, facilitating process control, and improving overall product quality. By optimising the predictive capabilities of these models, industries can enhance their operational efficiency and sustainability, thereby achieving better outcomes both economically and environmentally.

In recent years, significant interest has been directed towards the application of Machine Learning (ML) techniques in industrial processes, driven by advancements in data acquisition technologies and the increasing complexity and volume of industrial data. These advancements have enabled the development of sophisticated models capable of processing vast amounts of data, thereby improving decision-making and operational efficiency within industrial contexts.

⋆ The authors gratefully acknowledge Kanthal AB, Automation Region Research Academy (ARRAY), and the Swedish Knowledge Foundation (KKS) for their support.

This study aims to systematically evaluate the impact of data volume and complexity on the performance of ML models in multi-product batch processes within the iron and steel industry. A sensitivity analysis is conducted to provide insights that will guide future model development and applications in industrial batch processes.

A rigorous and systematic approach has been adopted in this study, wherein the effect of varying data volumes on model accuracy and complexity is analysed to ensure a comprehensive examination of these critical factors. A diverse range of ML models, with varying degrees of complexity, has been selected to assess their performance across different scenarios. These models include traditional machine learning algorithms, which are recognised for their efficacy in handling tabular data. Neural networks were excluded from this analysis due to the tabular nature of the dataset, which does not inherently suit such models, as evidenced by Shwartz-Ziv and Armon's findings that ensemble models generally outperform deep neural networks on tabular data (Shwartz-Ziv and Armon, 2022).

To explore the relationship between data volume and model performance, a data withholding approach has been implemented, enabling an assessment of how error scores fluctuate with varying data volumes. The data volume in this study ranges from 10 to 10,000 samples, distributed across 10 logarithmically spaced steps. These steps include 10, 22, 46, 100, 215, 464, 1,000, 2,154, 4,642, and 10,000 samples. This logarithmic progression ensures that the analysis covers a broad range of data volumes, providing a nuanced understanding of how data availability impacts model performance.

Previous studies have also examined the impact of data volume on model performance, albeit with differing scopes and methodologies. For instance, Bailly et al. (2022) investigated the effect of data volume on model metrics using artificially generated datasets with volumes of 1,000, 10,000, and 100,000 samples. Their findings indicated that, within the specific setup employed, data volume did not significantly influence model metrics, suggesting that the relationship between data volume and model performance may be context-dependent and influenced by factors such as data characteristics and model selection. In another study, Ramezan et al. (2021) explored the effects of training sample size on the performance of six supervised ML algorithms in classifying a large-area high-spatial-resolution remotely sensed dataset. Their work demonstrated that, while larger training sets generally led to better performance, there was considerable variation in how different classifiers responded to changes in sample size. These variations underscore the complexity of the relationship between data volume and model performance, highlighting the need for context-specific analyses.

The dataset employed in this study originates from the production of thermocouple materials at Kanthal Hallstahammar, specifically from the key stages of melting and hot rolling. Thermocouples are vital components in temperature measurement and are among the most common methods used in industrial processes, including those in the iron and steel industry. Their widespread use underscores the practical significance of accurate and reliable temperature measurement in maintaining process efficiency and product quality. The dataset comprises measurements of chemical composition and Electromotive Force (EMF), with the objective of predicting the final properties after hot rolling based on initial measurements taken after melting. A more detailed description of the pre-processed dataset is provided in Section 3.1.

The overarching aim of this work is to enhance the modelling of industrial processes by adhering to the principle of Occam's razor, which advocates for simplicity in model design. While continuous advancements in research contribute to increasingly complex models, it is crucial to balance complexity with practical implementation. This study builds upon previous work, such as Rendall et al. (2019) and Mählkvist et al. (2023), which examines the trade-offs between model complexity and performance. Rendall et al. (2019) succinctly illustrated the relationship between modelling complexity and implementation challenges, providing a framework for assessing the practicality of complex models in real-world applications. Similarly, Mählkvist et al. (2023) evaluated the modelling complexity of different classification models, including Logistic Regression, Random Forest Classifier, and Support Vector Classifier, to determine the most suitable model for the specific data problem at hand. These studies underscore the importance of balancing model sophistication with practical considerations, such as ease of implementation and computational efficiency.

Through a comparative analysis, it is intended to evaluate whether models with specific characteristics offer superior insights into the industrial processes under study. It is hypothesised that some models will perform better with larger data volumes and that a diverse range of model characteristics will yield more comprehensive insights, particularly under conditions of data saturation. Ultimately, this research seeks to determine the optimal balance between data volume and model performance within the context of industrial batch process modelling.

2. METHODOLOGY

This section delineates the methodology employed in this study, encompassing details about the development environment, systematic data processing approaches, model training, and evaluation techniques. The approach has been designed to ensure robust and reproducible results through meticulous dataset handling, model selection, and hyperparameter tuning.

2.1 Coding and Dependencies

Python is the coding language used for this study. Besides arbitrary dependencies on Pandas, NumPy, and other common libraries, the package `scikit-learn` Pedregosa et al. (2011) is employed for the implementation of the ML models, as well as for hyperparameter tuning.

2.2 Systematic Approach for Datasets and Modelling

Dataset and Subset sampling

Dataset and Subset Sampling This study begins with the product datasets, denoted as $\mathcal{P}_x$, where $x \in \mathcal{L}_P$ and $\mathcal{L}_P$ represents a list of all product datasets, each identified by a Greek letter, such as $\mathcal{L}_P = [\alpha, \beta, \ldots]$.

A data withholding approach is employed to generate increasingly larger datasets by sampling from the original product datasets. These smaller datasets are referred to as subsets. Each subset derived from a product dataset $\mathcal{P}_x$ is denoted by $\mathcal{S}^x_{i,j}$, where $\mathcal{S}^x_{i,j}$ represents the i-th iteration of sampling from the j-th subset of the x-th product dataset $\mathcal{P}_x$.

In this notation, i varies from 1 to n, where n is the total number of iterations performed for each subset size. This allows the same volume to be sampled multiple times to capture a more representative dataset. For instance, if the sampling volume is 10 samples, n subsets of volume 10 are generated by randomly selecting samples.

The index j corresponds to the position within a list of predefined sampling sizes, denoted as $\mathcal{L}_V = [v_1, v_2, \ldots, v_j]$. Each element v_j in $\mathcal{L}_V$ defines the size of the subset $\mathcal{S}^x_{i,j}$, ensuring that $\mathcal{S}^x_{i,j} \subseteq \mathcal{P}_x$.

Each iteration i of a subset $\mathcal{S}^x_{i,j}$ not only represents a random sampling from $\mathcal{P}_x$ but also retains all elements from the previous smaller subset $\mathcal{S}^x_{i,j-1}$. Consequently, as j increases (i.e., as the subset size grows within the same iteration x), each new subset includes all samples from the preceding smaller subset for the same product dataset. This approach ensures that as the dataset increases in volume, it maintains the same reference samples, thereby preserving consistency across different subset sizes.

Modelling A list of machine learning models of arbitrary size is utilised for training. Each element in the list ($\mathcal{L}_M = [m_1, m_2, \ldots, m_k]$) denotes a model m_k indexed by k.

Table 1. Characteristics of Machine Learning Models Evaluated in the Study

Models (Abbreviations)	Parametric/ Non-Parametric	Regularization (None/L1/L2)	Linearity (Linear/Non-linear)	Sensitivity to Scaling/Outliers
Ordinary Least Squares Linear Regression (OLS)	Parametric	None	Linear	Sensitive
Ridge Regression (Ridge)	Parametric	L2	Linear	Sensitive
Least absolute shrinkage and selection operator (Lasso)	Parametric	L1	Linear	Sensitive
Decision Tree Regression (DTR)	Non-parametric	None	Non-linear	Robust
Random Forest Regression (RFR)	Non-parametric	None	Non-linear	Robust
Gradient Boosting Regression (GBR)	Non-parametric	None	Non-linear	Robust
Linear Support Vector Regression (LIN)	Parametric	L2	Linear	Sensitive
Polynomial Support Vector Regression (POLY)	Parametric	L2	Non-linear	Sensitive
Radial Basis Function Support Vector Regression (RBF)	Parametric	L2	Non-linear	Sensitive

Each model in the list $\mathcal{L}_M$ is trained individually on each subset $\mathcal{S}^x_{i,j}$ derived from the product datasets.

The naming convention for a model trained on a specific subset follows the format $\mathcal{M}^x_{i,j,k}$. This indicates that the model indexed k from the list $\mathcal{L}_M$ has been trained on subset $\mathcal{S}^x_{i,j}$, where x refers to the originating product dataset, i to the iteration, and j to the specific subset volume as defined by its position in the list of sampling sizes $\mathcal{L}_V$.

2.3 Model Description and Parameter Range

This section outlines the models to be implemented, detailing each model in the subsequent subsections. Additionally, it includes the parameters and their respective ranges used for hyperparameter estimation, where applicable.

The model list ($\mathcal{L}_M$) consists of 9 ML models, as shown in the first column of Table 1. Thus, the length of the list of models is $|\mathcal{L}_M| = 9$.

A log-uniform distribution is used to define the hyperparameter range for many of the parameters. This distribution is particularly useful for parameters that span several orders of magnitude, as it facilitates the exploration of a wide range of scales effectively. The log-uniform distribution is defined as:

$$\mathcal{U}(x, y) \tag{1}$$

where $\mathcal{U}$ is the log-uniform distribution, and x and y are the lower and upper bounds, respectively.

In addition, a random integer distribution is employed to define the range for integer-valued hyperparameters, such as the number of estimators in ensemble models or the depth of decision trees. This distribution is particularly useful when the hyperparameter must take discrete values within a specified range. The random integer distribution is defined as:

$$\mathcal{I}(a, b) \tag{2}$$

where $\mathcal{I}$ is the random integer distribution, and a and b are the lower and upper bounds, respectively. This distribution uniformly samples integer values between a and b, inclusive.

Ordinary Least Squares Linear Regression (OLS) The OLS is a widely used approach to linear modelling that fits coefficients for all dimensions in the datasets to minimise the residual sum of squares between the observed values and the values predicted by the model (James et al., 2013).

Ridge Regression (Ridge) The Ridge model, also known as Tikhonov regularisation, extends linear methods such as OLS by incorporating regularisation. This model addresses a regression problem using the l2-norm.

The method was introduced by Hoerl and Kennard (1970a) in their 1970a; 1970b works.

The primary parameter for the Ridge model is the regularisation parameter for the l2-norm. Details of the parameter and the range of values used are provided in Table 2.

Least absolute shrinkage and selection operator (Lasso) The Lasso, similar to Ridge, is a linear model trained with regularisation, but it uses the l1-norm instead. The term was introduced by Tibshirani (1996).

The parameters for the Lasso are similar to those of Ridge, focusing on the regularisation parameter. However, in the case of Lasso, the parameter regulates the l1-norm. Details of the parameter and the range of values used are provided in Table 2.

Table 2. Parameters for Ridge and Lasso Regression Models

Model	Parameter	Scope
Ridge	Alpha[a]	$\mathcal{U}(0.01, 100)$
Lasso	Alpha[a]	$\mathcal{U}(0.01, 100)$

[a] The alpha parameter regulates the regularisation strength of the model.

Decision Tree Regression (DTR) The DTR is the first of the non-parametric methods and it infers simple decision rules from the data James et al. (2013).

The key parameter for the DTR is the maximum number of features, which determines the number of features to consider when finding the best split. Selecting the appropriate maximum number of features is crucial for controlling the diversity of features considered at each split. Refer to Table 3 for details.

Random Forest Regression (RFR) RFR, also known as random decision forests, is an ensemble learning method used for regression tasks. This technique constructs a multitude of decision trees during the training phase. Each tree in the forest relies on the values of a random vector, which is sampled independently and follows the same distribution across all trees (Breiman, 2001).

The parameters for the RFR include the maximum number of features (as with DTR) and the number of estimators, which represents the number of trees in the forest. The choice of the number of trees is critical, as it impacts both the model's performance and the risk of over-fitting.

Refer to Table 3 for details.

Gradient Boosting Regression (GBR) GBR is an ensemble learning method used for regression tasks. This technique builds a series of decision trees sequentially, with each tree aiming to correct the errors made by its predecessor. The process involves fitting new models to the residual errors of the previous models, thereby improving accuracy with each iteration. The final model is a weighted sum of all individual models, resulting in a robust predictive model that minimises the overall prediction error (Hastie et al., 2009).

The primary distinction between GBR and RFR lies in their construction strategy: random forests build trees independently and combine their results, whereas gradient boosting builds trees iteratively, with each tree focused on correcting the errors of the previous ones.

The parameters that GBR shares with the previous tree-based models include the maximum number of features, the number of estimators, and the maximum depth, which defines how deep each tree can grow.

Refer to Table 3 for details.

Table 3. Parameters for Decision Tree, Random Forest, and Gradient Boosting Models

Model	Parameter	Scope
Decision Tree	Max Features[a]	$\mathcal{I}(1, 100)$
Random Forest	Max Features[a]	$\mathcal{I}(1, 100)$
	# Estimators[b]	$\mathcal{I}(100, 1000)$
Gradient Boosting	Max Features[a]	$\mathcal{I}(1, 100)$
	# Estimators[b]	$\mathcal{I}(100, 1000)$
	Max Depth[c]	$\mathcal{I}(1, 100)$

[a] Max features determine the number of features considered for each split.
[b] Number of estimators specifies the total number of trees in the ensemble.
[c] Max depth controls the maximum depth of each tree.

Linear Support Vector Regression (LIN) SVM enhances the traditional support vector machine regressor by employing kernels to expand the feature space, thereby accommodating non-linear characteristics (Boser et al., 1992; James et al., 2013). Three different kernels are utilised: linear (discussed in this section), polynomial, and Radial Basis Function Support Vector Regression (RBF), which are presented in the subsequent sections.

The linear kernel is the simplest form of kernel function. It maps the input features directly without any transformation, making it suitable for linearly separable data. The decision boundary is a straight line (or hyperplane in higher dimensions), which simplifies the computation and interpretation.

The parameters for the LIN model include the choice of kernel (in this case, linear), the regularisation parameter C, and epsilon.

For details on the parameters and their ranges, see Table 4.

Polynomial Support Vector Regression (POLY) The polynomial kernel maps the input features into a higher-dimensional space using polynomial functions. This allows it to capture non-linear relationships between the features. The degree of the polynomial determines the model's complexity, enabling it to fit more intricate patterns in the data (James et al., 2013).

For details on the parameters and their ranges, see Table 4.

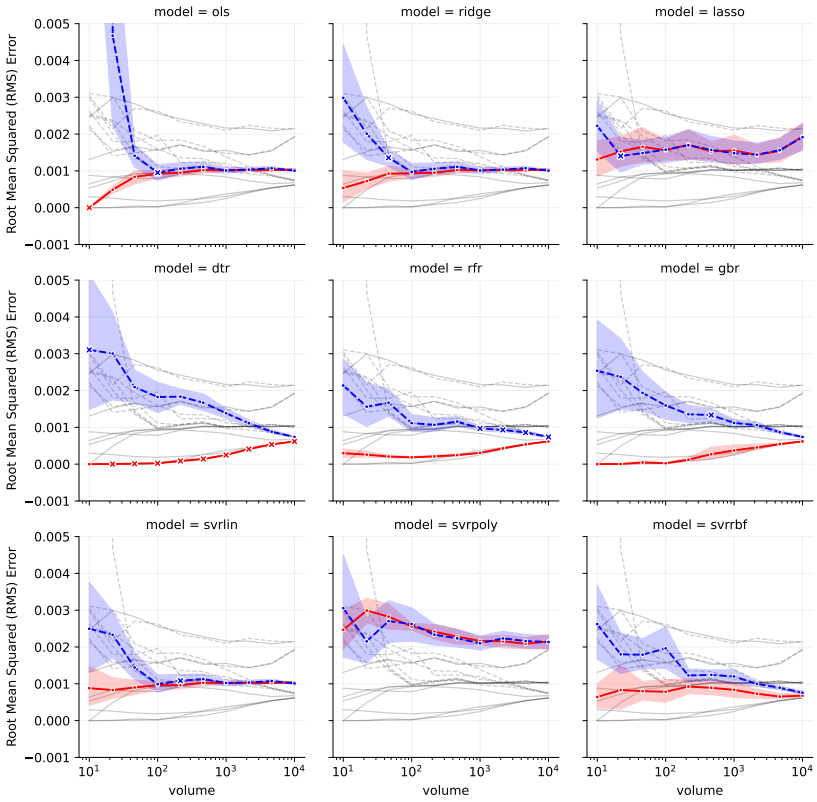
Radial Basis Function Support Vector Regression (RBF) The RBF kernel, also known as the Gaussian kernel, maps the input features into an infinite-dimensional space. It measures the similarity between data points based on their distance, allowing it to capture complex, non-linear relationships. The RBF kernel is particularly powerful for handling data that is not linearly separable (James et al., 2013).

For details on the parameters and their ranges, see Table 4.

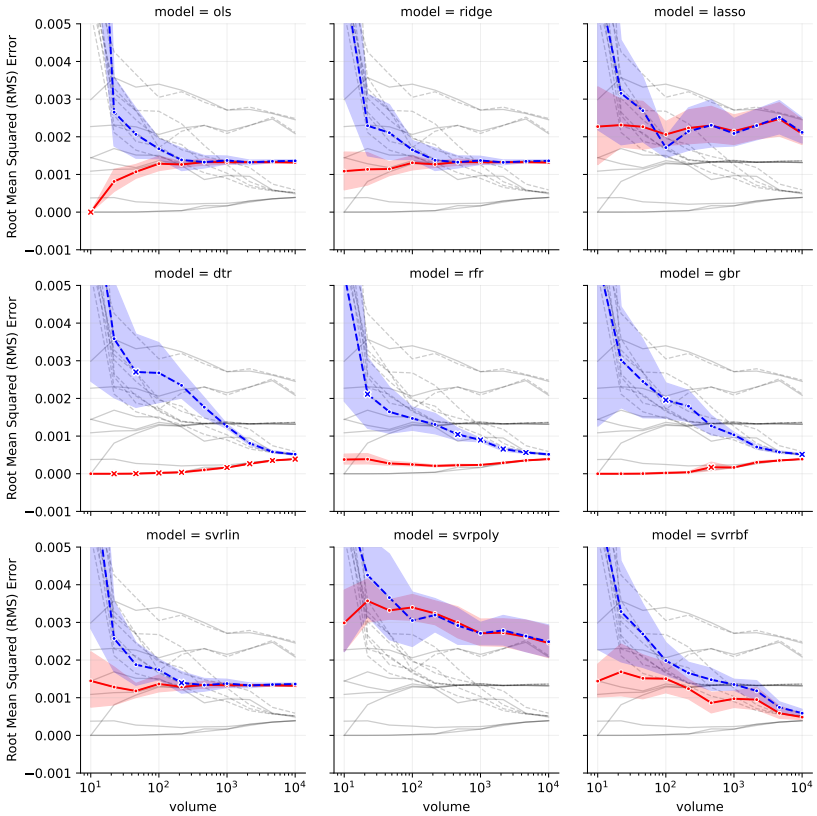
Table 4. Parameters for SVM Models with Different Kernels

Model	Parameter	Scope
Support Vector (Linear)	C[1]	$\mathcal{U}(0.1, 1.1)$
	Epsilon[2]	$\mathcal{U}(0.01, 1)$
Support Vector (Polynomial)	C[1]	$\mathcal{U}(0.1, 1.1)$
	Epsilon[2]	$\mathcal{U}(0.01, 1)$
	Gamma[3]	$\mathcal{U}(0.01, 100)$
	Degree[4]	$\mathcal{I}(1, 2)$
	Coef0[5]	$\mathcal{U}(0.01, 10)$
Support Vector (RBF*)	C[1]	$\mathcal{U}(0.1, 1.1)$
	Epsilon[2]	$\mathcal{U}(0.1, 1)$
	Gamma[3]	$\mathcal{U}(0.01, 100)$

[1] Regularisation Parameter for SVM applies uniformly across other SVM models to ensure consistency in regularisation and sensitivity.
[2] The epsilon parameter defines a margin of tolerance around the regression line within which no penalty is assigned for prediction errors.
[3] The gamma parameter controls the influence of a single training example and determines the spread of the kernel. This affects the smoothness of the decision boundary; lower values imply a broader spread, while higher values imply a narrower spread.
[4] The degree parameter specifies the degree of the polynomial function used to transform the data, determining the flexibility of the decision boundary by defining the highest power of the input features.
[5] The coef0 parameter represents the independent term in the kernel function and adjusts the influence of higher-order versus lower-order terms.



(a) Error Score for $\mathcal{P}_\alpha$



(b) Error Score for $\mathcal{P}_\beta$

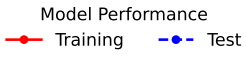


Fig. 1. Modelling Result for an Element of $\mathcal{P}_x$ Showing RMSE for All Models Over the $\mathcal{L}_V$

DOI: 10.3384/ecp212.057

2.4 Data Pre-processing

In this work, two product datasets were compiled, $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$, each representing different stages in the production process. The goal during pre-processing was to maintain at least 10,000 samples, which influenced the configuration of pre-processing steps. The pre-processing involved two key steps: feature selection and outlier removal.

Feature selection was conducted to identify and retain the most relevant variables by discriminating against those with low variance and high inter-correlation. A variance and correlation threshold was carefully estimated to ensure that the sample volume remained above 10,000, thus preserving the dataset's integrity while enhancing model performance.

Outlier removal was performed using the Interquartile Range (IQR) method. This method involved several steps:

1. Calculating the first quartile (Q_1) and the third quartile (Q_3). 2. Computing the IQR as IQR $= Q_3 - Q_1$. 3. Defining the lower bound as $Q_1 - 1.5 \times$ IQR. 4. Defining the upper bound as $Q_3 + 1.5 \times$ IQR. 5. Removing any data points that fell below the lower bound or above the upper bound.

Following the outlier removal, the dataset was scaled to standardise the features, ensuring that all variables contribute equally to the model's performance. Standardisation involved adjusting the features to have a mean of zero and a standard deviation of one, which is particularly important for machine learning models that are sensitive to the scale of the input data.

After these pre-processing steps, the dataset was split into training and testing sets for model evaluation. The separation between the training and test datasets effectively prevents overfitting, as is standard practice. To address underfitting, model parameters were allowed sufficient flexibility, managed through a trial-and-error approach. This approach was supported by a baseline guess informed by experience and conventional practices, ensuring that the models could adequately capture the underlying patterns in the data. The training set was used to fit the models, while the testing set was reserved for assessing the model's predictive performance on unseen data, thereby enhancing the model's ability to generalise and ensuring robust and reliable predictions.

2.5 Hyperparameter Estimation

A train-test split is implemented to ensure that the training process is conducted without any data leakage. Hyperparameter estimation is performed using a random grid search approach.

As demonstrated by Bergstra and Bengio (2012), the random search method offers significant advantages over conventional exhaustive grid search, particularly in terms of computational efficiency. It achieves comparable or even superior results while requiring fewer computational resources.

In a random grid search, hyperparameters are randomly sampled from a predefined list or distribution across a set number of iterations. This approach allows for a more effective exploration of the parameter space, increasing the likelihood of identifying optimal hyperparameters.

2.6 Evaluation

The Root Mean Squared Error (RMSE) is employed to evaluate the performance of each subset model $m^x_{i,j,k}$ on both training and test datasets. The RMSE depends on various factors, including the production database, iteration, volume, and model ($m(x, i, j, k)$). The RMSE is defined by Equation 3:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

where n is the number of observations, y_i represents the actual values, and $\hat{y}_i$ represents the predicted values.

This metric provides a robust measure of the model's predictive accuracy, with lower RMSE values indicating better model performance.

Grid-plot Evaluation To effectively interpret the complex system created by multiple layers of iterations, a structured framework for evaluating the results is necessary. The following approach is implemented in this work.

For each product dataset, a grid plot is created, containing one subplot for each model. Given that the number of models is 9, a 3 by 3 grid plot is used.

Each subplot, representing a specific model, displays how the train and test scores vary across the list of volumes. The y-axis shows the training and test error scores, while the x-axis represents the sampling volume. To enhance clarity, the x-axis is displayed on a logarithmic scale.

Additionally, each subplot shows the result scores for all iterations of the random grid search. The results are depicted as an area plot, with the mean indicated by a line (solid red for train data and dashed blue for test data).

To facilitate the comparison of model results within the same product dataset, faint but discernible lines are drawn in the background to represent other models. These backdrop lines correspond to the score type. Consequently, the x- and y-axes of all subplots are synchronised and shared.

Model Result Ranking To provide a comprehensive overview of model performance, a heat map ranking plot is created. This heat map shows which models achieve the best test scores (lowest RMSE) for each volume.

Each model in $\mathcal{L}_M$ is represented by an individual row on the y-axis.

Each volume in $\mathcal{L}_V$ has a corresponding column on the x-axis, increasing incrementally. Each cell in the heat map displays the rank of the model, ranging from 1 to $|\mathcal{L}_M|$.

The top three models are colour-coded individually, while the remaining models share a single colour, as indicated by the colour bar on the right.

3. RESULTS AND DISCUSSION

The results and corresponding discussion are presented in this section. First, the outcomes of the data pre-processing stage are detailed in Section 3.1. This is followed by a description of the subset sampling process in Section 3.2. Next, the details of the hyperparameter search are provided in Section 3.3. The modelling scores are then presented in Section 3.4, followed by the analysis of the model rankings in Section 3.5.

3.1 Preprocessing

The pre-processing stage resulted in two datasets, $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$, each ultimately containing an equal number of features. However, only half of these features were shared between the two datasets. During the feature selection process, different subsets of features were identified as relevant or superfluous for each dataset, leading to the retention of distinct feature sets in $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$. Given that both datasets are derived from the same processes, it is expected that they share some underlying characteristics, which is reflected in the final selection of features.

The features retained after pre-processing for both $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ are summarised in Tables 5 and 6. To ensure consistency in subsequent analyses, the features shared between $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ were ordered and enumerated in a manner that aligns corresponding features representing the same properties. Each feature was assigned a consistent subscript across both datasets, allowing for direct comparison and facilitating the interpretation of the model results.

Table 5. Features Selection Outcome for α and β

Features	α	β
Initial	22	22
Removed	14	14
Kept	8	8
Missing Value Ratio*	0.96	0.95

* Constant value what ratio dictating the threshold for feature removal due to missing values.

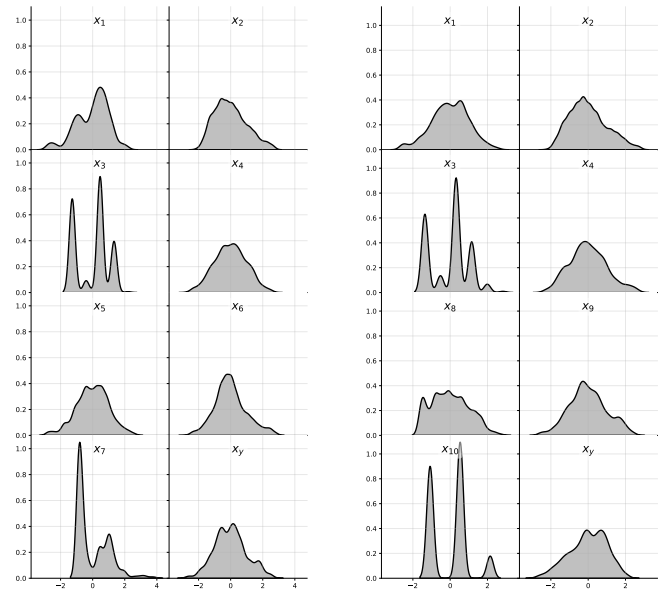
Table 6. Outlier Removal Results for α and β

Samples	α	β
Initial	12072	13597
Removed	1667	1543
Kept	10405	12054

3.2 Subset Sampling

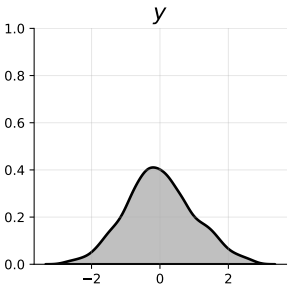
Subsets $\mathcal{S}^x_{i,j}$ were extracted for each volume in the list $\mathcal{L}_V$ and sampled A times, resulting in $|\mathcal{L}_V| \times A$ permutations per product dataset. The process begins with 10 samples and progresses to 10,000 samples in 10 steps, with the sampling volumes defined as $\mathcal{L}_V = [v_1 = 10, v_2 = 22, v_3 = 46, v_4 = 100, v_5 = 215, v_6 = 464, v_7 = 1000, v_8 = 2154, v_9 = 4642, v_{10} = 10000]$.

Thus, $|\mathcal{L}_V| = 10$, states that the length of the list of sampling volumes is 10.

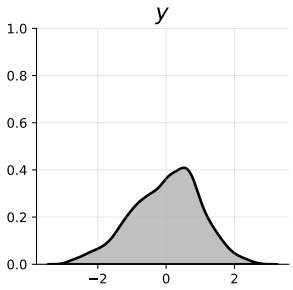


(a) Distribution of $\mathcal{P}_\alpha$'s X Features

(b) Distribution of $\mathcal{P}_\beta$'s X Features



(c) Distribution of $\mathcal{P}_\alpha$'s Y Feature



(d) Distribution of $\mathcal{P}_\beta$'s Y Feature

Fig. 2. Feature Distributions

3.3 Hyperparameter Estimation

For each element in the model list $\mathcal{L}_M$, the best fit was determined for every dataset in $\mathcal{S}$. Table 7 presents the mean values of the hyperparameters selected by the search process for the estimators.

3.4 Modelling Score

This subsection presents and discusses the variation in model scores as the volume of data in the production datasets increases. Figure 1 contains two sub-figures, 1a and 1b, which illustrate how the scores of all models change with increased data volume (see subsection 2.6.1 for details) for $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$, respectively. In general, the initial data volumes exhibit considerable volatility and are not given significant weight in the overall analysis of results. This volatility is reflected in the variation of the scores. Unless explicitly stated, both $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ are discussed collectively in the following analysis.

Most models show convergence between training and testing scores as data volume increases, with Lasso being a notable exception. Thus, it can be concluded that generalisation improves with an increase in data volume. How-

Table 7. Hyperparameter Result

Model	**Parameter**	α				β			
		10^1	10^2	10^3	10^4	10^1	10^2	10^3	10^4
Ridge	Alpha	5.130	5.190	3.393	7.206	16.470	7.628	2.797	5.435
Lasso	Alpha	0.026	0.017	0.012	0.033	0.171	0.018	0.018	0.019
Decision Tree	Max Features	36.0	35.0	43.0	46.5	47.5	73.5	57.0	66.5
Random Forest	Max Features	45.0	61.0	54.0	27.0	49.5	39.5	59.0	44.5
	# Estimators	327.0	269.0	490.0	636.5	291.5	657.5	604.0	257.0
Gradient Boosting	Max Depth	34.5	37.5	10.5	23.0	56.0	63.5	15.0	25.5
	Max Features	39.5	6.5	25.5	50.5	42.5	4.5	34.0	61.5
	# Estimators	563.5	599.5	513.5	215.0	605.0	579.0	588.0	693.5
Support Vector (Linear)	C	0.047	0.032	0.071	0.154	0.082	0.088	0.057	0.101
	Epsilon	0.024	0.016	0.022	0.028	0.049	0.022	0.029	0.027
Support Vector (Polynomial)	C	0.614	0.281	0.254	0.443	0.145	0.346	0.208	0.289
	Epsilon	0.512	0.119	0.114	0.123	0.636	0.117	0.107	0.130
	Gamma	0.867	0.519	0.263	5.295	0.637	0.852	0.094	0.132
	Degree	2.0	3.0	2.0	2.0	2.0	3.0	2.0	2.0
	Coef0	0.187	0.197	1.117	0.607	0.091	0.644	0.570	0.461
Support Vector (RBF)	C	0.145	0.134	0.168	0.185	0.028	0.103	0.477	0.166
	Epsilon	0.017	0.021	0.024	0.019	0.034	0.031	0.033	0.017
	Gamma	0.300	0.182	0.082	2.177	0.260	0.077	0.136	0.979

ever, a point of diminishing returns in generalisation is discernible at different volumes and to varying degrees.

Lasso exhibits interesting behaviour, where training and testing scores converge quickly but then fluctuate as data volume increases, resulting in subpar overall performance. To explain this atypical behaviour, it is worth considering Table 1, which highlights Lasso's unique use of l1 regularisation. Furthermore, as shown in Table 7, the Lasso hyperparameter (Alpha) remains relatively static as volume increases, indicating that the method may be incompatible with this data or that the parameter ranges need revision.

The training score generally increases monotonically for all models except Lasso and RBF. The previous explanation for Lasso is insufficient when considering RBF, but since RBF shows significant improvement with larger volumes, this is not a major concern.

Examining the score spread, it is clear that ensemble models outperform the other models. A noteworthy runner-up is RBF, which, at larger volumes, approaches the performance of the ensemble models. This suggests that the data is well-suited for a non-parametric approach. Additionally, since all non-linear models, except POLY, show strong performance, it implies that the data has a non-linear nature. Alternatively, it may also indicate that the hyperparameter estimation and parameter ranges are insufficient to fully capture the underlying data patterns.

Referencing Table 1, it is possible that the robustness of ensemble methods to outliers gives them an advantage, suggesting that the pre-processing approach may have been inadequate for models sensitive to outliers.

3.5 Model Ranking

To make the overall performance of the models more discernible, their rankings across different data volumes are presented in Figs. 3 and 4 for $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$, respectively (see subsection 2.6.2 for details).

The ranking for $\mathcal{P}_\beta$ shows convergence earlier than for $\mathcal{P}_\alpha$, meaning it stabilizes at a lower data volume. Specifically, $\mathcal{P}_\beta$ reaches saturation at a volume of approximately $10^{2.3}$, while $\mathcal{P}_\alpha$ does not reach saturation until a volume of around 10^3.

The differences between $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ can be attributed to the distinct sets of features retained after pre-processing, even though both datasets originate from the same underlying processes. Although each dataset contains approximately 10,000 samples and initially had an equal number of features, the final set of features for $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ differs. This suggests that the selected features contribute differently to the modelling process, with certain features being more relevant or informative for one dataset than the other.

The divergence in feature selection underscores the varying impact of these features on the predictive models. Some features may offer greater predictive power or relevance depending on the specific context of each dataset, which in turn influences the point at which the model rankings stabilise

4. CONCLUSION

This study has demonstrated that most machine learning models show consistent improvement in predictive performance, as evidenced by a reduction in test RMSE,

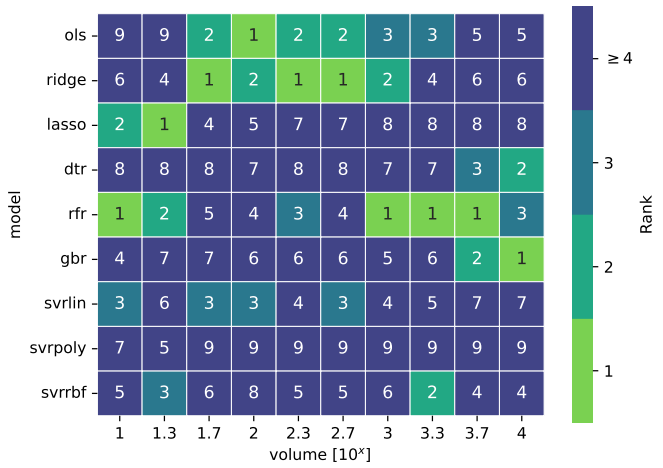


Fig. 3. Model Rank for $\mathcal{P}_\alpha$

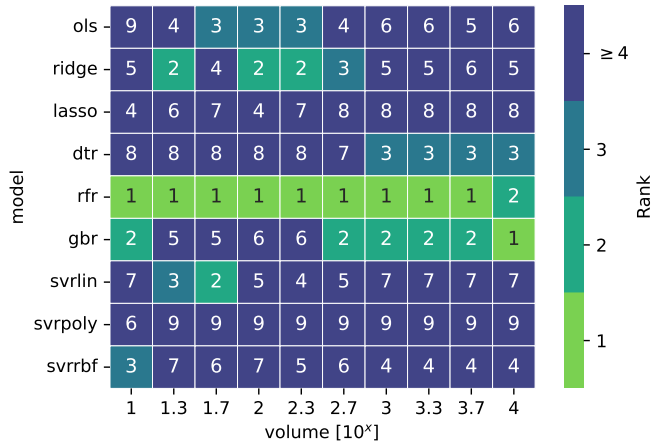


Fig. 4. Model Rank for $\mathcal{P}_\beta$

with increasing data volume. This finding underscores the importance of larger datasets in enhancing model generalisation, which is crucial in the context of industrial batch process modelling.

Among the models evaluated, ensemble models such as RFR and GBR consistently outperformed other models across various data volumes. Their robustness to outliers and ability to capture complex, non-linear relationships make them particularly effective for the datasets used in this study.

In contrast, non-ensemble models, especially those sensitive to outliers, generally underperformed relative to ensemble methods. Models employing L1 regularisation, such as Lasso, exhibited less stability and improvement in performance, suggesting that the chosen regularisation method may not be optimal for this data.

The disparity in performance between ensemble and non-ensemble models may be attributed to the latter's greater sensitivity to outliers. While stricter outlier removal could potentially enhance the performance of non-ensemble models, it would also reduce the number of available data samples, potentially limiting the study's scope.

The analysis of model rankings revealed a notable difference in the convergence times between models trained on the $\mathcal{P}_\beta$ and $\mathcal{P}_\alpha$ datasets, despite both being derived from the same type of product and processes. This disparity illuminates the effectiveness of the introduced framework in detecting subtle variations in dataset complexity, particularly in terms of the variation of selected features, which significantly impacts the amount of data required for models to achieve saturation.

These findings highlight the importance of careful model selection and robust data pre-processing in industrial applications. Given the superior performance of ensemble models, they should be prioritised in future research within similar contexts. However, non-ensemble models may require more sophisticated pre-processing and parameter tuning to achieve comparable performance. These conclusions provide a foundation for further work aimed at improving model accuracy and robustness in industrial settings, potentially through enhanced data handling techniques and the inclusion of more complex models.

The findings of this study have broader implications beyond the iron and steel industry, extending to other sectors that rely on industrial batch processes, such as the chemical, pharmaceutical, and food processing industries. These industries share common challenges in managing complex, multi-product operations where model performance is heavily influenced by data volume and quality. The demonstrated superiority of ensemble models in handling non-linear relationships and their robustness to outliers suggests that similar approaches could be highly effective in these related industries. Moreover, the insights gained from addressing model sensitivity to outliers and the impact of dataset complexity can inform best practices in these sectors, where optimising process efficiency and product quality is equally critical. By adopting the strategies outlined in this study, industries with comparable batch processing challenges can enhance their predictive modelling capabilities, leading to more sustainable and efficient operations.

REFERENCES

Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., and Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. *Computer Methods and Programs in Biomedicine*, 213, 106504. doi: 10.1016/j.cmpb.2021.106504.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Boser, B.E., Guyon, I.M., and Vapnik, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152.

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

Hastie, T., Tibshirani, R., Friedman, J.H., and Friedman, J.H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 2. Springer.

Hoerl, A.E. and Kennard, R.W. (1970a). Ridge regression: Applications to nonorthogonal problems. *Technometrics : a journal of statistics for the physical, chemical, and engineering sciences*, 12(1), 69–82.

Hoerl, A.E. and Kennard, R.W. (1970b). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics : a journal of statistics for the physical, chemical, and engineering sciences*, 12(1), 55–67.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning*, volume 112. Springer.

Mählkvist, S., Ejenstam, J., and Kyprianidis, K. (2023). Cost-sensitive decision support for industrial batch processes. *Sensors*, 23(23), 9464.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Ramezan, C.A., Warner, T.A., Maxwell, A.E., and Price, B.S. (2021). Effects of Training Set Size on Supervised Machine-Learning Land-Cover Classification of Large-Area High-Resolution Remotely Sensed Data. *Remote Sensing*, 13(3), 368. doi:10.3390/rs13030368.

Rendall, R., Chiang, L.H., and Reis, M.S. (2019). Data-driven methods for batch data analysis – A critical overview and mapping on the complexity scale. *Computers & Chemical Engineering*, 124, 1–13. doi:10.1016/j.compchemeng.2019.01.014.

Ritchie, H., Roser, M., and Rosado, P. (2020). CO2 and greenhouse gas emissions.

Shwartz-Ziv, R. and Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, 81, 84–90. doi:10.1016/j.inffus.2021.11.011.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1), 267–288.

Yoro, K.O. and Daramola, M.O. (2020). Chapter 1 - CO2 emission sources, greenhouse gases, and the global warming effect. In M.R. Rahimpour, M. Farsi, and M.A. Makarem (eds.), *Advances in Carbon Capture*, 3–28. Woodhead Publishing. doi:10.1016/B978-0-12-819657-1.00001-3.