Particuology 115 (2026) 336–355

Contents lists available at ScienceDirect

Particuology

journal homepage: www.elsevier.com/locate/partic

Prediction of immersion mill performance and energy use via machine learning

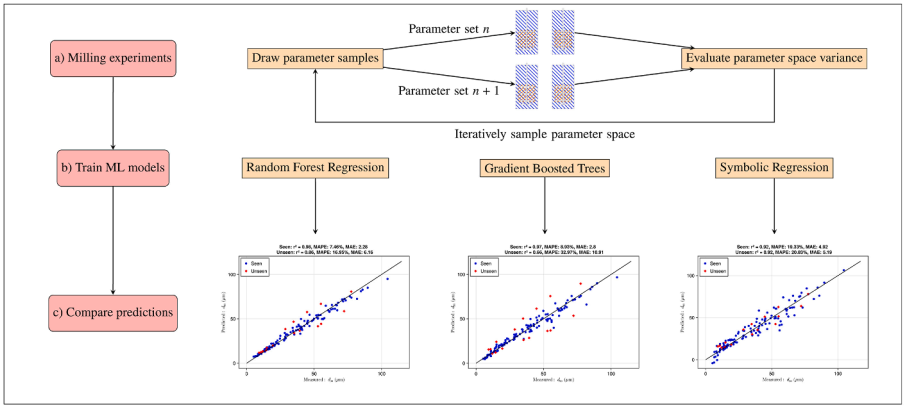
Daniel Weston[a,b,*], Li Liu[b], Christopher Windows-Yule[a]

[a] School of Chemical Engineering, University of Birmingham, Birmingham, B15 2TT, UK
[b] Johnson Matthey Technology Centre, P.O. Box 1, Belasis Avenue, Billingham, TS23 1LB, UK

HIGHLIGHTS

- Symbolic regression predicted immersion mill performance with a sparse dataset.
- The generated equations provided interpretable rules for process optimisation.
- This approach minimised energy use whilst achieving a target particle size.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Machine learning
Wet milling
Prediction
Optimisation

ABSTRACT

This study performs a comparative assessment of three Machine Learning (ML) models to determine their robustness in data-sparse industrial environments for predicting and optimising the performance and energy consumption of an immersion mill. By employing active learning strategies that target high-variance regions in process space, a limited experimental dataset (15 batches) was found sufficient to accurately predict mill performance, provided the correct model architecture is chosen. Three models were used: Random Forest Regression (RFR), Gradient Boosted Trees (GBT) and Symbolic Regression (SR). Based on performance when used on completely unseen data, as well as interpretability and usability, the SR models were found to be the most effective for this application. The simple algebraic form of the SR models allowed for direct use in exploring 'what-if' scenarios, and equally allowed either model to serve as a constraint for the other to minimise energy use to achieve a target particle size. An unexpected finding during this work was that the relationship between final particle size and impeller speed and grinding media content is weaker than for classic vertical stirred mill designs owing to transport and/or mixing mechanisms novel to this particular mill design. The result of this study is a set of predictive models that can be used in optimising the immersion milling process, and effectively responding to changes in feed Particle Size Distribution (PSD) whilst minimising energy use.

* Corresponding author. School of Chemical Engineering, University of Birmingham, Birmingham, B15 2TT, UK.
E-mail address: dtw545@student.bham.ac.uk (D. Weston).

https://doi.org/10.1016/j.partic.2026.05.023
Received 30 January 2026; Received in revised form 6 May 2026; Accepted 29 May 2026
Available online 10 June 2026

Nomenclature

Latin Symbols	
$C_0 \ldots C_{15}$	Symbolic Regression (SR) model constants
d_{10}, d_{50}, d_{90}	10th, 50th, and 90th percentiles of the cumulative particle size distribution (μm)
d_{90}^0	Initial 90th percentile particle size (μm)
E_m	Specific energy input (kJ kg^{-1})
I	Motor current (A)
k	Specific energy rate coefficient (kJ kg^{-1} min^{-1})
m	Mass of the slurry (kg)
N	Impeller speed (rpm or rad s^{-1})
P	Power draw (W)
$q_3^*(\log x)$	Logarithmic volume density (%)
R	Motor resistance (Ω)
S_i	Selection function for the ith size class
t	Batch time elapsed (min)
w	Slurry solids content (wt%)
x	Particle size (μm)
Greek Symbols	
α	L_1 regularisation coefficient
γ	Minimum objective function improvement required for further node splitting
η	Learning rate
ϕ	Grinding media content (wt%)

1. Introduction

Understanding the behaviour of wet stirred mills is important in many manufacturing contexts where size reduction is required (Daraio et al., 2020). Physical experiments allow insights into the complete extent of milling, which would be prohibitively expensive and complex for numerical experiments. Whilst there is a plethora of research conducted on typical wet attritor mills (Daraio et al., 2020; Kwade & Schwedes, 2007; Radziszewsk, 2013) where the grinding media and slurry occupy the same volume, there is little research conducted on immersion mill configurations. Fig. 1 illustrates the difference between the mills, focusing on the volume occupied by grinding media and slurry.

To the authors' knowledge there has, to date, been only one peer-reviewed article concerning immersion mills (Szilagyi & Nagy, 2019). The focus of that work was not, however, the milling process. Rather, the immersion mill mentioned in their study was a feature of one of three crystallisation units they sought to compare for crystallisation performance. Whilst they make use of Population Balance Models (PBMs) as is common in milling literature (as will be discussed below), their models were developed to predict crystal growth.

As such, the subsequent literature review will focus on vertically oriented wet attritor mills, which are the most similar to an immersion mill.

Global mining processes were responsible for ca. 1.73% of global energy consumption in 2015 (Aramendia et al., 2023) which is forecasted to increase by 200% to 800% by 2060. Comminution typically accounts for 80% to 90% of the energy use in mining (Jeswiet & Szekeres, 2016). Additionally, comminution is known to be extremely inefficient, especially with poor process parameter selection (Sterling et al., 2023). Because size reduction is a fundamental step in many processes, it is important to minimise the energy it requires and thus reduce its demand on global energy production.

Energy use optimisation in stirred milling typically relies on calculating the optimum stress energy to achieve the desired particle size

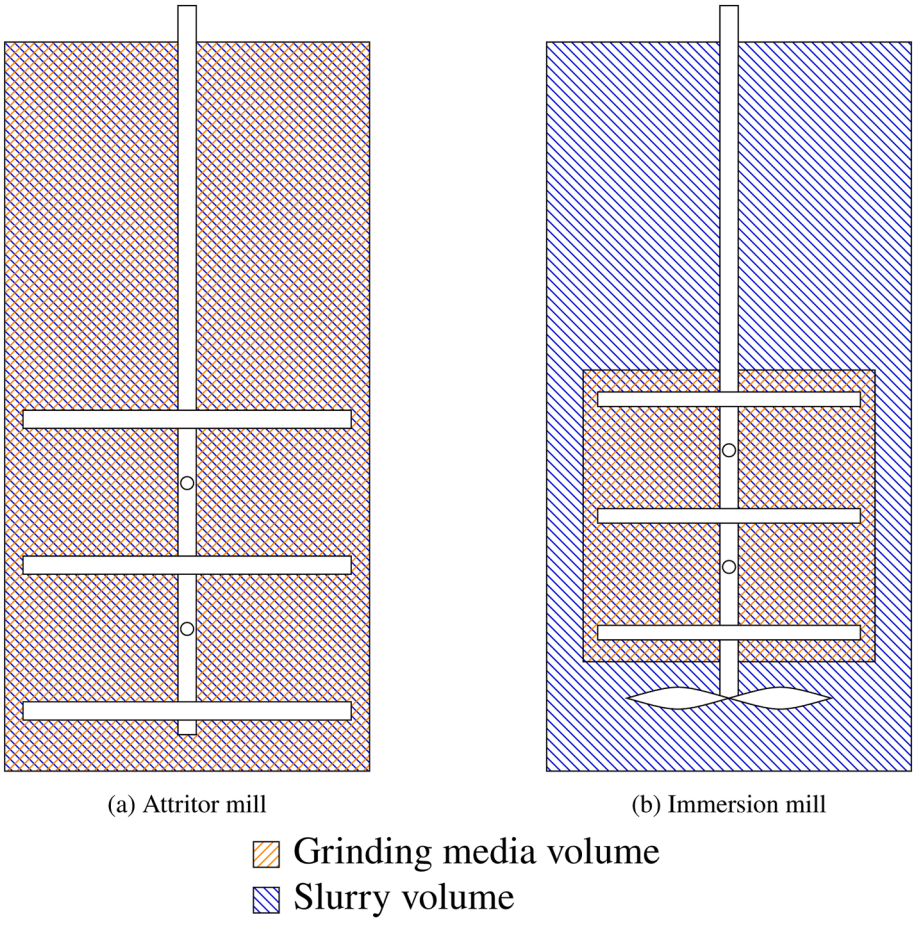


Fig. 1. Simple schematic illustrating the difference between an immersion mill and a more conventional attritor mill.

(Kwade, 2004). This is achieved by measuring specific energy input to reach the target particle size at different stress energies; the lowest required specific energy input corresponds to the optimal stress energy. Whilst this approach is demonstrated to work well (Kwade, 2004), the obvious drawback is that for any change in target particle size, one must repeat the series of experiments to determine the new optimal stress energy, wasting material.

A constraint for optimising a mill's energy consumption is the product Particle Size Distribution (PSD), which must be achieved to meet quality requirements.

Historically, empirical models such as the JK t_{10} model which has the form

$$t_{10} = A(1 - e^{-bE_m}) \tag{1}$$

have been used to link breakage energy and size reduction for coarse feed particles, but this has limited applicability for (ultra-)fine grinding setups as highlighted by Taylor et al. (2020), such as the immersion mill that is the focus of this study. Moreover, they discuss that t_{10} relationships were primarily derived from crushing (impact) behaviours, whereas the mechanisms of (ultra-)fine grinding are typically due to attrition, abrasion, and compression (Kumar et al., 2023); in particular, there is a lack of validation data for t_{10} models at sizes below 100 μm, which is also the order of magnitude that the feed sizes processed by the immersion mill in this study are.

Another popular method for predicting particle size is using a PBM, which is a series of differential equations that describe the rate of change in each particle size class with time. For a batch mill with n size classes the general form of a PBM for the ith size class (where $i = 1$ represents the coarsest class) is given by

$$\frac{dm_i}{dt} = -S_i m_i + \sum_{j=1}^{i-1} b_{ij} S_j m_j \tag{2}$$

where S_i is the selection function and b_{ij} is the breakage function, which describe the rate that class i breaks up and how these newly formed particles are distributed, respectively.

This approach works well for systems with limited numbers of size classes. However, for n size classes, there are n S_i parameters, and $\frac{n(n-1)}{2}$ for b_{ij}, meaning that for n size classes, one requires $\frac{n(n+1)}{2}$ parameters in a batch system; consequently, such models rapidly become impractical for large numbers of size classes; such numerous parameters necessitate large numbers of experiments and also poses extreme difficulty for an optimiser to fit such numerous degrees of freedom (Bilgili, 2024). As an example, using 5 size classes necessitates 15 parameters, for a given set of process conditions. Should it be assumed that the model parameters themselves are a function of process parameters, then the likelihood of statistically insignificant parameters increases (Bilgili, 2024). While mechanistically rigorous, the back-calculation of selection and breakage functions often requires extensive experimental campaigns to fit parameters, and these parameters often do not transfer well when process conditions (like solid loading) change. Machine Learning (ML) offers a potential bypass to this rigid parameterisation by learning the aggregated process dynamics directly (Lu et al., 2024).

In practice, one assumes the form for S_i and b_{ij} based on the system in question, trading generality for vastly improving the statistical significance of the fewer required parameters. For example, Bilgili et al. successfully used a PBM to predict the PSD for copper ore in a wet mill where S_i was assumed to be a power-law relationship between the impeller speed, solids loading and class size (made dimensionless by dividing by their maximum values), i.e. process parameters. In contrast, b_{ij} was assumed independent of process parameters and instead is solely a function of the parent particle and child particle size classes (Bilgili et al., 2024).

Despite the apparent success of their approach, Bilgili has raised the fact that errors in PBMs predictions are fraught with non-physical breakage parameters when fitted to experimental data (Bilgili, 2024). The key relevant reason in this study, as outlined by Bilgili is the assumption of first order breakage kinetics, which is less likely to be true beyond short milling times. Additionally, in a fine grinding system like the one used in this study, assumptions of first order breakage kinetics are rendered less valid when there is a high presence of fines and multi-particle interactions (Yang, 2018). There is work to develop non-linear PBMs, which mitigates the weaknesses of the first order assumption, but the system-dependent choice of expressions for selection and breakage remain (Capece et al., 2011).

This study aims to develop predictive models that can improve the effectiveness and efficiency of the immersion milling process; in particular by exploring process parameter effects on the energy usage and resultant particle size during a batch. The contemporary ML architectures do not rely on assumptions about the grinding mechanisms, which means that they have the opportunity to provide robust predictions of immersion mill performance. The selection of these specific model architectures was driven by the constraints of the dataset, namely its limited size, high sparsity, and the non-linearity inherent to the milling process. Due to the limited size, heavily parameterised models such as Artificial Neural Networks (ANNs) are highly susceptible to overfitting and poor generalisation on small tabular datasets (Grinsztajn et al., 2022). Next, the sparsity of the data degrades the performance of distance- and kernel-based methods such as Support Vector Machines (SVM) and Gaussian Process (GP) regression, where distance-based metrics are likely to degrade due to the limited number of points in the parameter space (Rasmussen & Williams, 2005). Further, the previously mentioned models have limited interpretability, which limits their usability in decision-making beyond just model output (Kuhn & Johnson, 2013). Finally, the non-linear aspect of the data precludes the usage of Partial Least Squares (PLS) regression, which is a linear method and would struggle to capture the non-linear particle size reduction behaviour (Shan et al., 2014). Thus, the three selected models are:

- Random Forest Regression.
- Gradient Boosted Trees.
- Symbolic Regression.

The rationale for selecting these models is that the tree-based (Random Forest Regression (RFR) and Gradient Boosted Trees (GBT)) and evolutionary (Symbolic Regression (SR)) models are naturally tolerant of sparsely sampled parameter space, and limit overfitting (Breiman, 2001; Chen & Guestrin, 2016; Cranmer, 2023). These types of models have also been successfully used in scenarios with similar dataset types and sparsity in parameter space (Cava et al., 2021; Chehreh Chelgani et al., 2021; Jenkins et al., 2025; Jones-Salkey et al., 2024). A more detailed description of each model is provided in section 2.4. The effectiveness of three different ML techniques is compared, assessing their performance for relatively small sample sizes representative of the amount of data typically available in real industrial scenarios.

2. Materials and methods

The mill utilised was a Hockmeyer HCP 1/4 immersion mill (Hockmeyer Equipment Corporation); a sketch of the unit is shown in Fig. 1 (b). The batch size for this unit is 4 L to 9 L, which is loaded into the chamber. Grinding media are loaded into the basket, which has a 0.25 L capacity. During operation, attritor pins inside the basket drive the grinding media, and a down-pumping impeller located externally on the bottom of the basket agitates the fluid. Both geometries are located on the same shaft, and so co-rotate at the same speed. The basket opening is such that grinding media are retained during normal operation, but fluid is allowed to pass through.

The material used for this study was dolomite (Omya UK Ltd, UK) which is a commonly studied material in the context of milling (Gao & Forssberg, 1993). A Scanning Electron Microscopy (SEM) image is

shown in Fig. 2.

2.1. Particle size measurement

The slurry PSD is measured in two intervals. Firstly, between times 0 min and 15 min inclusive, samples are taken every 3 min. For the rest of the batch, samples are taken every 15 min. The rationale for the sampling strategy and batch time is driven by initial observations for milling the material at 'aggressive' and 'relaxed' conditions - high and low impeller speeds, respectively. Measured PSDs indicated that during the first 15 min the size reduction rate is large and then tends to fall off past this point in time. Thus, selecting more samples early on provides useful dynamic information about the mill that is important to accurately predict. Fig. 3 illustrates the typical PSD evolution with time for the two aforementioned levels of impeller speed.

Samples were analysed using a Mastersizer 3000 (Malvern Panalytical, UK) fitted with a wet dispersion unit, which measures particle size by laser diffraction.

Samples were dispensed into the wet dispersion unit until the fraction of laser light blocked was in the range of 15% to 20%. In this range, the results obtained were at their most repeatable and least likely to contain outliers as determined by testing various obscuration ranges. As shown by Fig. 2, the dolomite particles are not spherical, so Malvern's non-spherical model was chosen for the dolomite samples; this model is also recommended for milling applications where particles are unlikely to resemble spheres (Malvern Panalytical, 2017). The samples were circulated for 90 s before measurement, where 5 measurements were taken, back-to-back.

2.2. Energy use measurement

Energy usage was measured from the mill Programmable Logic Controller (PLC) by a Raspberry Pi 5 (Raspberry Pi Ltd) utilising the *pylogix* library (Roeder, 2024). Amongst the available tags were the current drawn by the mill motor and the temperature inside the mill, which were both recorded.

The resistance of the mill motor was determined from reference values for current and power consumption at a known speed obtained during Factory Acceptance Testing (FAT) for the mill. Thus, the motor resistance was determined by solving

$$P = I^2 R \tag{3}$$

for the motor resistance, R. With the motor resistance known, the power draw at time t, $P(t)$ was calculated using the same equation. Finally, the specific energy, E_m input of the mill at sample time τ was calculated by

$$E_m(\tau) = \frac{1}{m}\int_0^{\tau} P(t)dt \tag{4}$$

where m is the mass of the slurry. It is important to emphasise that the power and subsequently calculated energy correspond to *electrical torque* provided to the motor rather than mechanical torque exerted by the motor itself.

2.3. Experiment settings

Process parameters were varied to understand how the mill's energy usage and performance are affected. Table 1 lists the parameters that were varied in this study.

The M^2E^3D library, which provides efficient parameter sampling methods and equation discovery (Nicusan & Windows-Yule, 2022), was used to generate the process settings for each of the batches. The M^2E^3D library has successfully been used to explore pharmaceutical blending (Jones-Salkey et al., 2024) and compression (Munu et al., 2025) systems; both of which have complex interactions between process parameters and responses. The former reference also provides a detailed description of using M^2E^3D for process parameter exploration. This was done in an iterative manner utilising the library's ability to target regions of process space with high variance, without having to specify all the process settings in advance. Unlike random sampling, which requires large N to cover a 5-dimensional space, this active learning approach maximises the information density of each experiment. By targeting the high

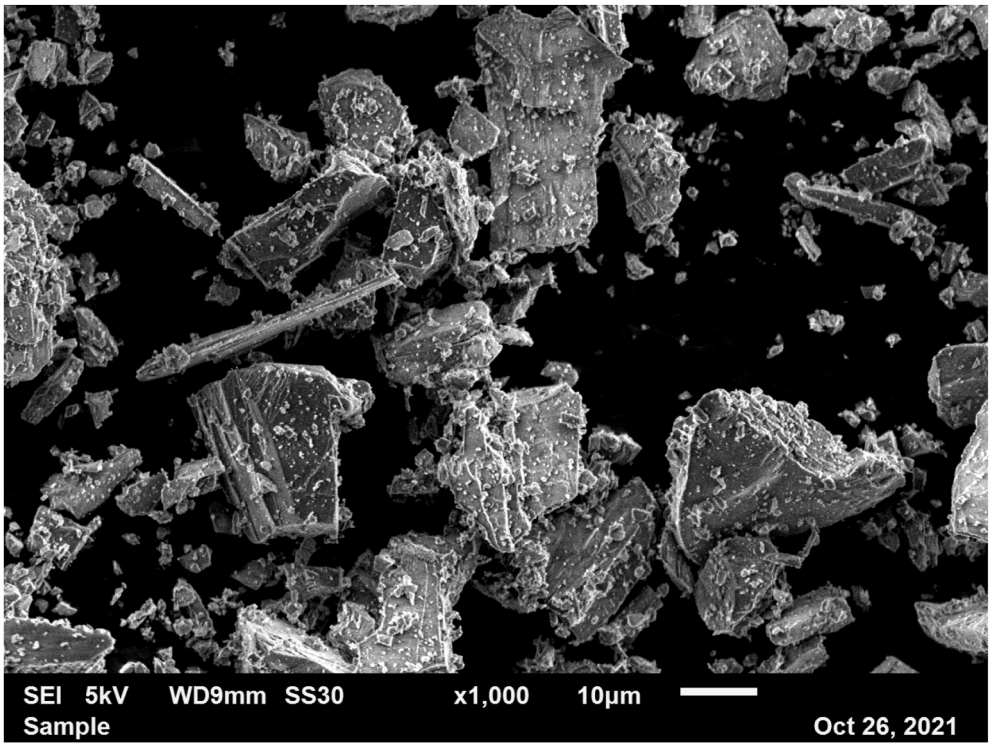
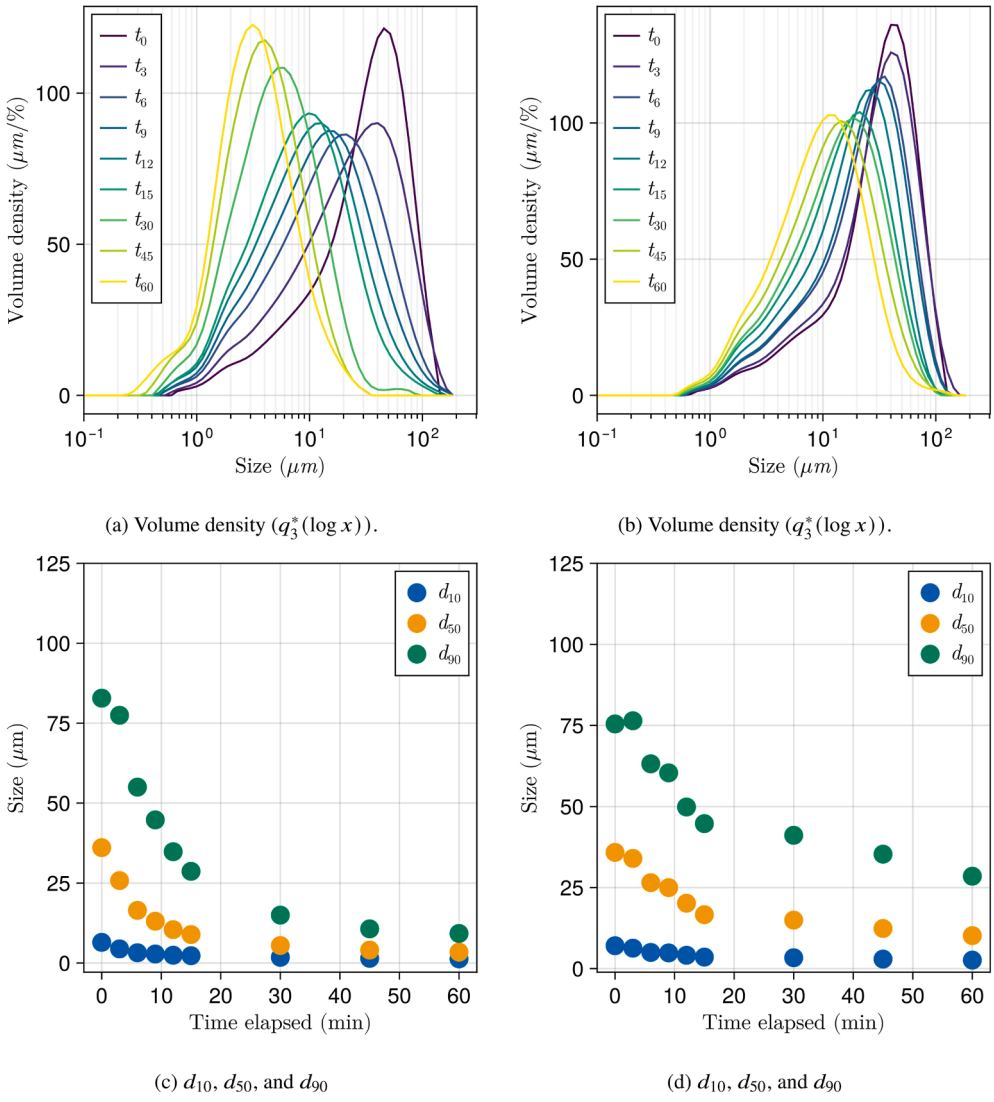


Fig. 2. SEM image of dolomite, image taken at JMTC Chilton.



(a) Volume density ($q_3^*(\log x)$).

(b) Volume density ($q_3^*(\log x)$).

(c) d_{10}, d_{50}, and d_{90}

(d) d_{10}, d_{50}, and d_{90}

Fig. 3. Example PSD evolution with time. In (a) and (c), the corresponding process conditions were $N = 2974$ rpm, $w = 30.26$ wt%, and $\phi = 60.69$ wt%. In (b) and (d), the corresponding process conditions were $N = 1690$ rpm, $w = 43.00$ wt%, and $\phi = 99.42$ wt%.

Table 1
Process parameters and ranges.

Parameter	Units	Range	
		Lower	Upper
Impeller Speed	rpm	1500	3000
Slurry solids content	wt%	30	45
Grinding media content	wt%	60	100

variance areas of parameter space, it is more likely that the obtained data represent the regions of the process space that provide statistically significant data to the models; thus, each batch presents a meaningfully different series of measurements.

In turn, the efficient coverage of parameter space reduces the number of data points required for training ML models, which is beneficial in industrial scenarios where data collection is expensive and time-consuming.

Firstly, 8 sets of process parameters were selected, and the data collected, which were fed back into M^2E^3D so that the library can evaluate the variance in the measured responses to inform where in process space it should sample from next. Subsequent samples were drawn in groups of 2 to 4 and their corresponding data fed back into M^2E^3D before more samples were drawn, which ensured regular updates to calculated response variances.

In total, 17 samples were drawn and used as the experiment settings. The data from 15 of these milled batches were used as training/testing data; the remaining two were held back as validation data, remaining *completely unseen* during model training. A summary of process inputs is found in Table 2.

2.4. Predictive modelling

The specific energy input and d_{90} are the responses that will be predicted by models outlined in the subsequent sections. As discussed in Section 1, energy use and particle size are of great interest in milling, thus make natural candidates for predictive modelling. As the energy use was continually measured during the entire batch duration, specific energy values were determined by evaluating Equation (4) for times elapsed that matched the particle size sampling cadence. That is, between times 0 min and 15 min inclusive, the specific energy values were calculated for time elapsed increasing in 3 min intervals. Beyond 15 min, the values were calculated for time elapsed increasing in 15 min up until 60 min inclusive. Particle size measurements were repeated 5 times per sample, so the mean d_{90} was used to form the d_{90} response.

Input features for the models are:

Table 2
Milling experiment parameters.

Run	Impeller speed (rpm)	Slurry solids content (wt%)	Grinding media content (wt%)
Reserved for training/testing			
1	1637	31.37	78.02
2	2325	43.93	87.82
3	1788	34.47	92.31
4	2878	36.72	96.75
5	2545	33.06	67.45
6	1952	40.48	63.08
7	1560	38.47	82.12
8	2721	42.13	77.36
9	2156	31.05	73.49
10	2761	30.00	60.00
11	2761	45.00	70.00
12	2650	30.19	89.93
13	1521	32.05	65.34
14	1526	44.74	99.31
15	2974	30.26	60.69
Unseen by training/testing			
16	2376	34.67	73.29
17	1690	43.00	99.42

- Impeller speed (N).
- Slurry solids content (w).
- Grinding media content (ϕ).
- Time elapsed (t).
- Initial d_{90} (d_{90}^0).

The decision to include the initial d_{90} is justified because this allows the model to account for variability in the initial PSD. Variability in the initial d_{90} and the model's ability to account for this fact facilitates investigations into the impacts of altering the input material PSD, for example if there were a change in supplier or if there were a change (planned or unplanned) to an upstream process that provides the material to be milled.

The models in Sections 2.4.1 and 2.4.2 have many hyperparameters that affect model performance, these were tuned by using the *Optuna* (Akiba et al., 2019) library. There are many possible parameters to optimise, so only significant ones identified will be explained in the proceeding sections. Hyperparameter importance is determined via the functional ANOVA (fANOVA) implementation in *Optuna* (Hutter et al., 2014). An individual parameter's significance (or lack thereof) is then determined by the fraction of the total variance explained by each hyperparameter, sorted in descending order. The objective function for tuning the model was the mean Root-Mean Square Error (RMSE) from performing Leave One Out Cross-Validation (LOOCV) on the training data. One batch was left out per fold, a subsample from the training data, to serve as testing data, meaning that for each candidate set of hyperparameter values, the model was trained 15 times and tested against the corresponding held back batch. The RMSE was calculated from the test prediction. Thus, the objective function constructed seeks to outline how well, on average, does the model perform against a variety of unseen data. Cycling through each batch also ensures that the risk of overfitting is minimised, as a minimal objective function value in this case means that the test data is adequately predicted by the model - the model simply 'memorising' the data would provide a poor test score.

Additionally, the hyperparameter tuning was performed in two stages:

1. Search space includes all hyperparameters, optimisation ran for a short time to screen for significant ones.
2. Search space includes only hyperparameters deemed significant, optimisation allowed to run for as long as needed.

The rationale for this procedure is to identify the hyperparameters that have a tangible amount of significance to the model and focus on tuning them whilst leaving all others set at sensible defaults. For this study, the significant parameters are determined as the smallest amount that contribute to at least 90% of the total variance explained by the hyperparameters. After this threshold, it is likely that the additional required computation is not worth the minimal improvement in the model's predictive abilities. The results of hyperparameter tuning for the models described below in sections 2.4.1 and 2.4.2 are presented in section 3.1. In contrast, the SR model primarily relies upon *a priori* operator selections, so these are detailed directly in its methodology.

2.4.1. Random Forest Regression

RFR is an ensemble learning technique that utilises the output from multiple decision trees to predict a response y based on input features $\boldsymbol{x}$. Each tree is independently trained, and is a series of decisions based on the input features $\boldsymbol{x}$ made at each node of the tree. Each decision contains a threshold value for a feature, and the path taken along the tree depends on the feature's position to the threshold. Eventually, the tree will contain leaf nodes (also known as terminal nodes) which contain the predicted response value, *according to that individual tree*. The predicted response $\hat{y}(\boldsymbol{x})$ is determined by averaging over each tree's individual prediction of $\hat{y}(\boldsymbol{x})$. The interested reader is referred to (Breiman, 2001) for further details on random forests and how RFR is constructed.

In this study, the RFR implementation in *SciKit-Learn* (Pedregosa et al., 2011) was used. During training, each tree has access to their own randomly subsampled dataset from the original dataset. Hence, the trees are trained with diverse training data which helps improve generalisation ability. Each decision tree is also given a random subset of the input features to use to inform where node splits (decisions) should take place. The combination of using random data and input features ensures decorrelation between different decision trees in the random forest.

2.4.2. Gradient Boosted Trees

Gradient Boosted Trees are an improvement on traditional Gradient Boosting (GB) methods. Alongside many performance improvements, the prediction accuracy is improved by incorporating both L1 and L2 regularisation into the objective function - penalising model complexity inherently. Like with RFR, the model is an ensemble of decision trees, but unlike RFR, the trees are trained sequentially, improving upon the predictions made by the previous trees. Thus, each successive tree is trained on the residuals of the previous trees' predictions. The interested reader is referred to (Friedman, 2001) for further details on GBTs. In this study, the *XGBoost* implementation of GBT is utilised (Chen & Guestrin, 2016).

2.4.3. Symbolic Regression

The two previous approaches can suffer from a lack of interpretability. At a certain complexity, the constructed trees are difficult for a human to parse and understand; this makes interpreting the decisions made to reach a particular prediction difficult to reason with, to understand *why* the prediction has the value it does. Further, it can be difficult to intuitively understand the importance of input features in the model without comprehensive statistical analysis of the model. In turn, the process understanding can thus be inhibited by complex trees. It is worth noting that the lack of interpretability mentioned above is on a relative scale. Tree-based methods are still generally more interpretable than 'black-box' models such as ANNs, which despite their apparent effectiveness in chemical engineering scenarios, are difficult to interrogate as noted by Jones-Salkey et al. (Jones-Salkey et al., 2024).

In contrast, SR constructs closed-form expressions for predicted response $\hat{y}$ for given input features $\boldsymbol{x}$

$$\hat{y} = f(\boldsymbol{x}) \tag{5}$$

that are both easier to understand and can also hint at the underlying

physics governing measured responses. In this study, M^2E^3D as a frontend to *SymbolicRegression.jl* (Cranmer, 2023) was utilised, in addition to generating to the process parameters used to collect the data in the first place.

During training, a list of binary operators, $f(x, y)$ (e.g. addition, subtraction), unary operators, $f(x)$ (e.g. log, sqrt) and user-defined functions, f are provided as ingredients to construct trees that represent expressions which are fitted against the training data. An example for the expression $y = A(1 - e^{-t/\tau})$ is shown in Fig. 4.

Both the loss function and complexity are targeted for minimisation during SR. Intuitively, there is a trade-off between prediction accuracy and model complexity; a selection of models at the Pareto front is returned for the user to select where along the front is acceptable for their use.

In contrast to the other techniques mentioned above, hyperparameter tuning in SR is different. There are few significant numerical parameters that one can vary with respect to model prediction performance, where relevant, their values were set according to guidance in the *PySR* documentation. These numerical parameters and their settings (in brackets) are:

- Parsimony (0.001): Scaling factor that penalises complexity.
- Max size (40): Maximum allowed equation complexity.

The core hyperparameters for SR instead are operator selection and constraint and were manually tuned. In this study, the following operators were selected:

- Binary: +, -, *,/, ^ (where ^ is defined as pow(base, exponent))
- Unary: log, exp

The justification for selecting these operators is based on their prevalence in the literature when characterising mills. In other domains, other operator choices may be justified, such as sin, cos and other trigonometric functions. The pow operator was subject to the constraint that it can contain expressions of complexity no more than 1, i.e. values can be raised only by a single power, which helps prevent nonsensical solutions which may 'fit' well but lack physical meaning. Finally, the input features and responses were assigned their physical units and a penalty score 10^5 was set for solutions which do not result in a homogenous equation - this value is orders of magnitude larger than the values encountered in the scope of this study, so it should enforce homogeneity.

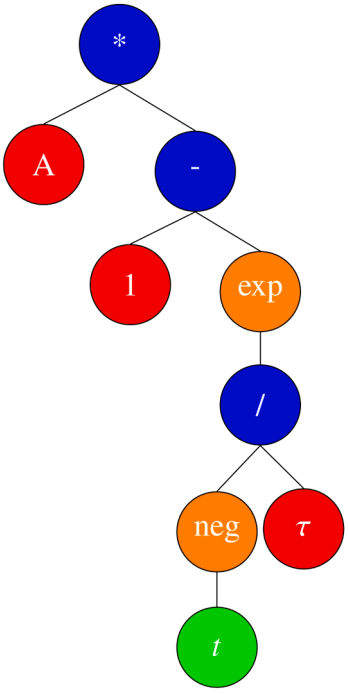


Fig. 4. Expression tree for equation $y = A(1 - \exp(\frac{-t}{\tau}))$ where neg represents negation. Binary and unary operators are coloured blue and orange respectively. Input features and constants are coloured green and red respectively.

3. Results

In this section, the training processes for the RFR and GBT models are presented, and then the performance of the three models is evaluated using data that remain wholly unseen during any stage of the training process; they were not present in hyperparameter tuning or model training.

3.1. Model training

3.1.1. Random Forest Regression

Results from hyperparameter tuning are shown in Figs. 5 and 6. This section and the subsequent one for GBT tuning utilise the results from the short, screening phase for the optimisation history and parameter importance plots, whilst the longer, final phase is used to create the plot showing the performance across each LOOCV fold.

The optimisation history for specific energy prediction during the short, hyperparameter screening phase is shown in Fig. 5(a). After ca. 45 trials, the improvement in objective function value saw diminishing returns.

Fig. 5(b) shows that the minimum samples per leaf is by far the most important hyperparameter for specific energy predicted by RFR. The next two most important parameters have a total weight of 0.128. An explanation of these hyperparameters is as follows:

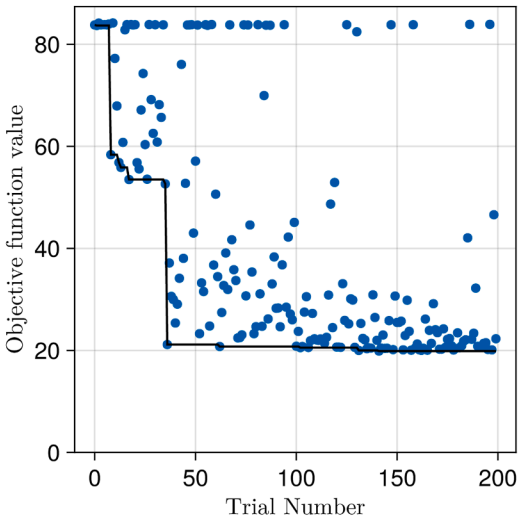
- Minimum samples per leaf (min_samples_leaf): Splits at any node are only permitted if the subsequent left and right branches have at least this minimum number of samples each.
- Minimum samples to split (min_samples_split): The minimum number of samples required to consider splitting a node.
- Number of trees (n_estimators): The number of decision trees that comprise the random forest.

For further explanation of additional parameters, see https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

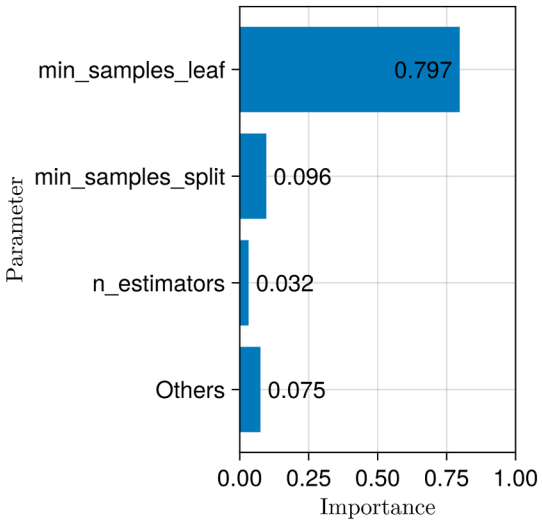
Performance across each LOOCV fold for the 'best' trial in tuning hyperparameters for the specific energy model is shown in Fig. 5(c). The plot shows that batch 4 in particular is important training data, given the high RMSE compared to any of the other folds. Batch 12 also demonstrates this effect, to a lesser degree when held back. Generally, the test error was lower than the training error. This is likely due to the sensitivity of the model to specific high-variance batches (e.g. batch 4) within the small dataset. In the LOOCV outlined, this 'difficult' batch is training data for $N - 1$ folds, which inflates the training error for the majority of the runs. As will be discussed in Section 3.3, this did not negatively affect the performance on the unseen data. An additional explanation is that the strong linear relationship between specific energy and time as implied by Equation (4) means that the underlying relationship is easy for the model to predict. It must be noted that the model predicts unseen data well (as explained later in Section 3.3) despite the test error being unintuitively lower than the training error.

A similar number of trials were required for a lack of further improvement in the objective function value for d_{90} prediction, as shown in Fig. 6(a). There were 4 significant hyperparameters, 1 more than for specific energy (see Fig. 6b).

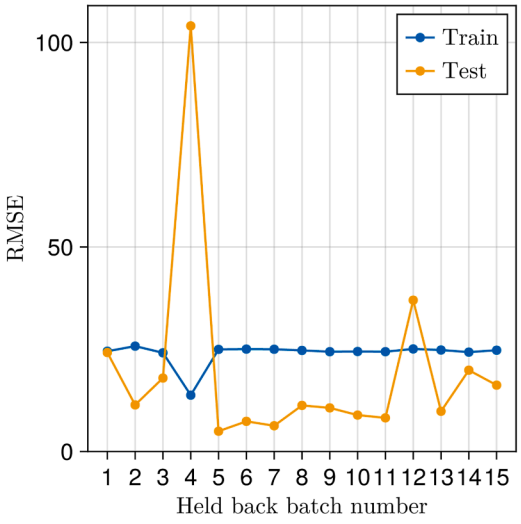
A shared feature from both responses is that the minimum number of samples per leaf is the most important hyperparameter as shown by their respective importance graphs (Figs. 5b and 6b) where this



(a) Optimisation history. The black line indicates the current lowest objective function value at trial n.



(b) Hyperparameter importance.



(c) LOOCV fold performance.

Fig. 5. RFR hyperparameter tuning performance for specific energy response.

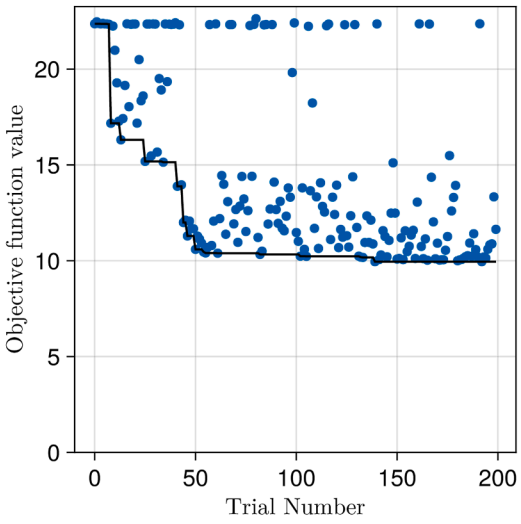
hyperparameter is shown to have the greatest fraction of explained variance. However, the importance of this parameter was less for d_{90} than for specific energy. Another shared important parameter was the minimum samples to split. An explanation of the identified significant hyperparameters is as follows:

- Minimum samples per leaf (min_samples_leaf): Splits at any node are only permitted if the subsequent left and right branches have at least this minimum amount of samples each.
- Maximum leaf nodes (max_leaf_nodes): Caps the maximum amount of terminal nodes in each tree - this promotes simpler trees that are easier to interpret. The penalisation of complexity additionally helps guard against overfitting.
- Minimum samples to split (min_samples_split): The minimum number of samples required to consider splitting a node.
- Complexity parameter (ccp_alpha): Controls the maximum complexity of the decision trees in the forest. For each tree, subtrees are formed, starting from the root node all the way to the complete tree. Each subtree is assigned a complexity value, which considers both the error of the subtree and its size. Subtrees are iteratively pruned until their complexity value is smaller than the complexity parameter; thus, this parameter acts as a regularisation term, penalising overly complex trees.

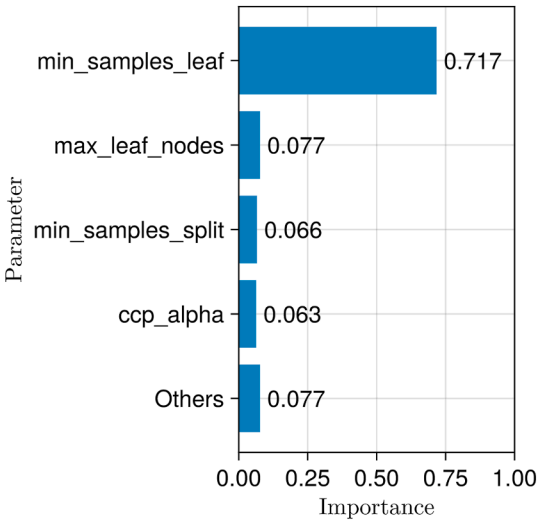
Performance across the LOOCV folds for the d_{90} (shown in Fig. 6c) appears as expected with respect to train and test error, with the test error typically higher. Overall, there was a greater variation in test performance concerning d_{90} prediction, which implies that the underlying relationship between the response and the input features is more complex.

3.1.2. *Gradient Boosted Trees*

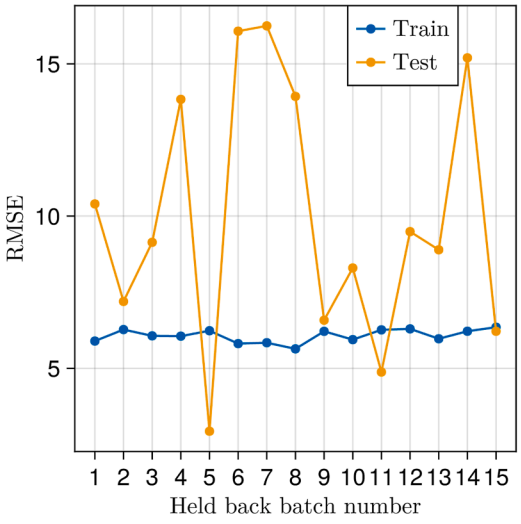
Hyperparameter tuning was performed similarly to above in Section 3.1.1. Results from hyperparameter tuning are shown in Figs. 7 and 8 The optimisation history for specific energy prediction as shown in Fig. 7 (a) shows a similar number of trials were required for a lack of further



(a) Optimisation history. The black line indicates the current lowest objective function value at trial n.



(b) Hyperparameter importance.



(c) LOOCV fold performance.

Fig. 6. RFR hyperparameter tuning performance for d_{90} response.

improvement in the objective function value as for the RFR model.

Fig. 7(b) shows that the minimum weight required per child is by far the most important hyperparameter for the model; the remaining significant hyperparameters have a total weight of 0.119. An explanation of these hyperparameters is as follows:

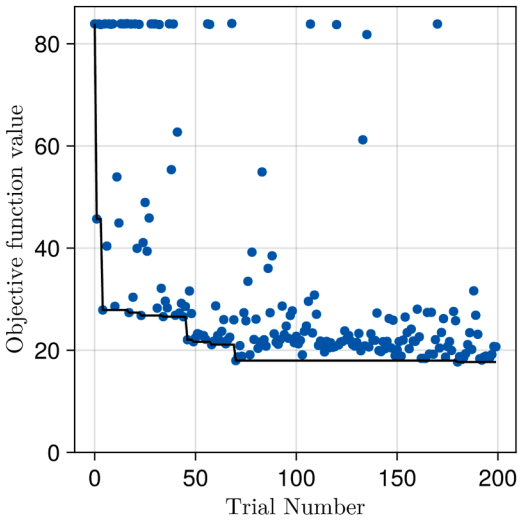
- Minimum weight required per child (min_child_weight): The minimum weight required for further partitioning to continue; if a tree is partitioned and a child node has weight below this value then this node is considered a terminal node.
- Subsample ratio (subsample): Fraction of training data sampled prior to tree growth. This aims to penalise overfitting.
- (max_bin): The maximum number of bins used to discretise continuous features; finer discretisation (more bins) improve quality of tree splits.
- η (eta): The learning rate, which controls how much the contribution of each subsequent tree is scaled down; a smaller value means that more trees are required to achieve the same performance, but the model is less likely to overfit.
- α (alpha): Coefficient applied to the absolute value of model weights in the objective function; this is a form of L1 regularisation that penalises large weights.
- γ (gamma): The minimum improvement in the objective function required to consider further splitting of tree nodes; a larger value means that the model is more conservative in its tree growth.

For further explanation of additional parameters, see https://xgboost.readthedocs.io/en/stable/parameter.html.

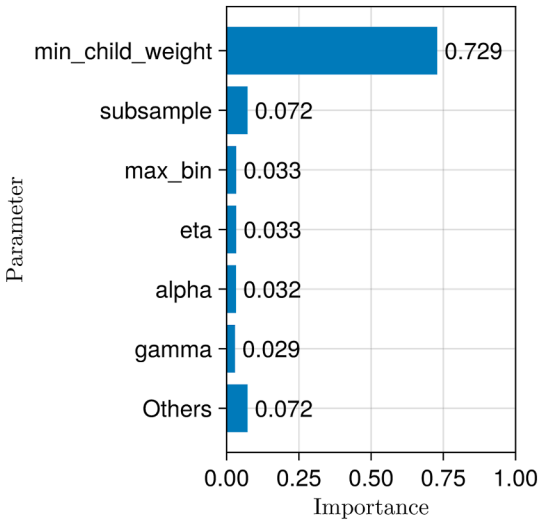
The fold performance (see Fig. 7c) for the GBT approach is very similar to the RFR model, sharing that fact that batches 4 and 12 appear to be important training data. Like with the RFR, performance on unseen data was good, despite the unusual training and test RMSE values.

Similarly to the tuning process for specific energy, there was a steep decrease in the objective function value within a short number of trials almost initially in the as shown in Fig. 8(a). Whilst there was a similar trend of improvement even after 150 trials, there were more step changes in improvement for the d_{90} response.

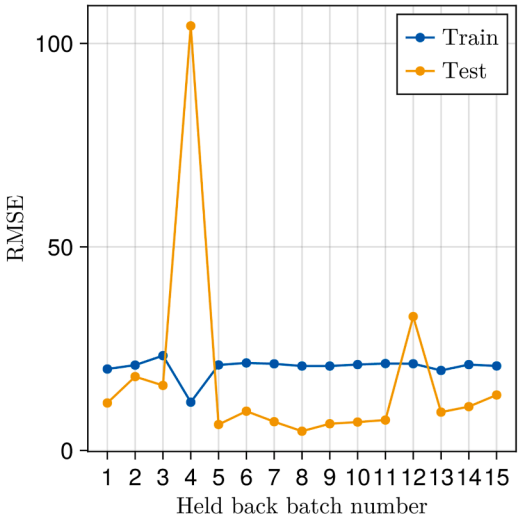
Like with the specific energy response, the d_{90} response shares that



(a) Optimisation history. The black line indicates the current lowest objective function value at trial n.



(b) Hyperparameter importance.



(c) LOOCV fold performance.

Fig. 7. GBT hyperparameter tuning performance for specific energy response.

the minimum weight required per child is the most important hyperparameter. Interestingly, only 1 more significant hyperparameter was identified - the subsample ratio, which together were the top two most important hyperparameters for the specific energy response. An explanation of the significant parameters is as follows:

- Minimum weight required per child (min_child_weight): The minimum weight required for further partitioning to continue; if a tree is partitioned and a child node has weight below this value then this node is considered a terminal node.
- Subsample ratio (subsample): Fraction of training data sampled prior to tree growth. This aims to penalise overfitting.

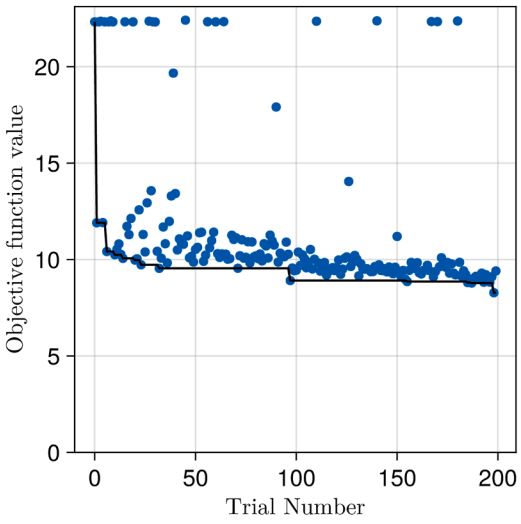
As shown in Fig. 8(c), the fold performance was very similar to the RFR model for the d_{90} response.

3.2. Trends

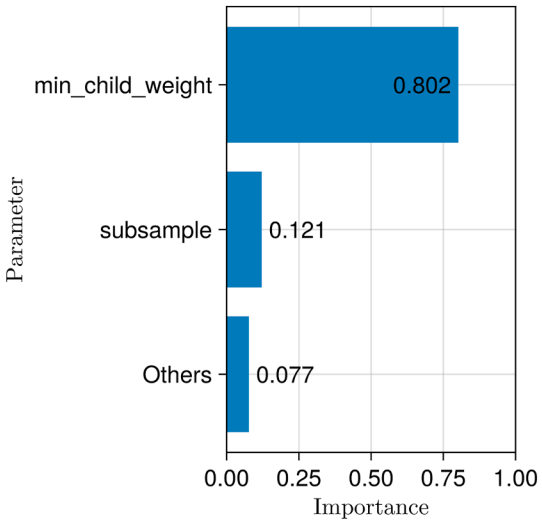
The correlation between input features and responses is shown in Fig. 9.

For the specific energy response shown in Fig. 9(a), time elapsed was strongly positively correlated, which makes sense given the formulation of Equation (4), potentially overshadowing other input features which showed noticeably less correlation. Positive correlations were observed for impeller speed and grinding media content, which is expected due to the greater kinetic energies (and thus required electrical torque). It is stressed that the calculated energy is based on current supplied to the motor, which is different from mechanical power measured at the impeller shaft (accounting for transmission and frictional losses) that exhibits a much stronger dependence on impeller speed (Shah, 1991). Additionally, the motor is fed by a Variable Frequency Drive (VFD), meaning that the input power is based on required torque and has a non-linear, complex relationship with impeller speed.

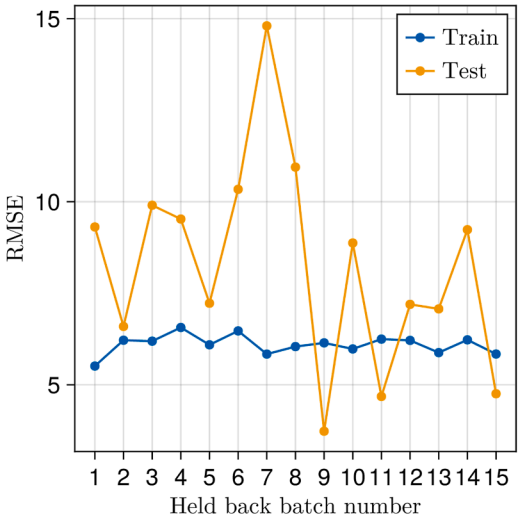
Between the input features, a noticeable positive correlation (0.42) between the initial d_{90} and solids content was observed. An explanation for this lies in the batching process, where each sample was dispersed at an identical speed for an identical period of time, following the exact same protocol for all batches. Thus, during batching, slurries with higher



(a) Optimisation history. The black line indicates the current lowest objective function value at trial n.



(b) Hyperparameter importance.



(c) LOOCV fold performance.

Fig. 8. GBT hyperparameter tuning performance for d_{90} response.

solids content experience less mixing for the same energy input due to their increased viscosity. A future improvement on the methodology in this study would be to first identify a batching protocol that provides identical dispersion across the solids content range to decouple the two features.

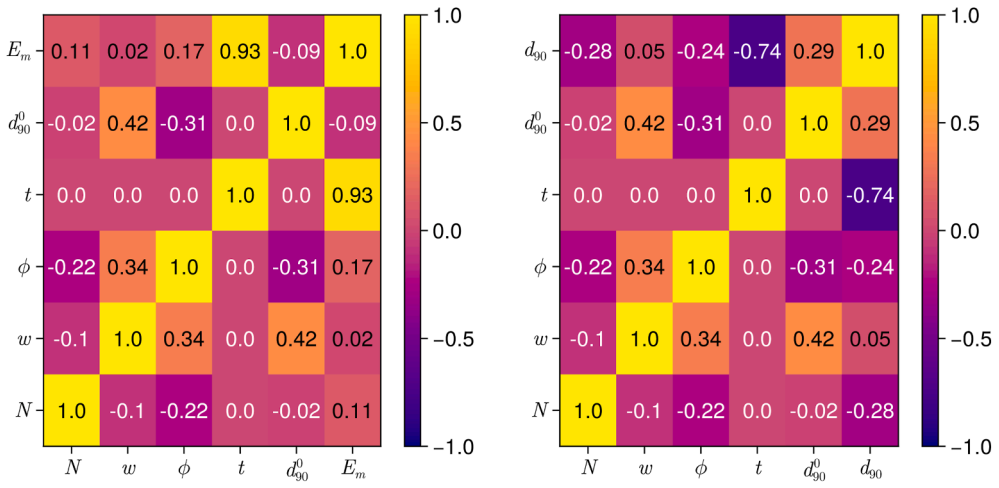
In contrast, the d_{90} response shown in Fig. 9(b) shows a negative correlation with time. This is expected, as with enough time (should there be enough stress energy present) the particles will reduce in size during milling. Other negative correlations included the impeller speed and grinding media content, both of which are proportional to stress energy (Schilde et al., 2011); with more stress energy, finer particle sizes can be achieved which means the negative correlations observed make sense.

Unexpectedly, the magnitudes for speed and grinding media content were less than the literature implies. An explanation is that this particular mill configuration only grinds particles inside the basket - there is a transport and/or mixing element to grinding that is missing in 'classic' attritor mill configurations.

During data collection, it was noticed that at particularly low speeds the particle size remained unchanged at the first 2 to 3 sampling points that the measured d_{90}; occasionally the size even increased, presumably due to unabated flocculation occurring. Thus, even though a low speed implies low stress energies, that was not the sole cause of poor milling performance as measured early on in a batch.

3.3. Random Forest Regression

Visualising the decision trees created in RFR can theoretically help understand the relationship between the input features and the responses via the decisions taken to get to a terminal node. Figs. 10 and 11 show a single tree taken from the generated ensemble blue(e.g. tree 9 of the 37 total decision trees in the specific energy forest). It is important to note that these decision trees can grow in complexity; for both responses, the entire selected tree is deliberately presented at full scale to visually demonstrate practical limitations of the RFR model. In particular, the tree shown in Fig. 11 is large and difficult to follow. Therefore, whilst RFR is theoretically an interpretable ML technique, this interpretability has practical limits.



(a) Specific energy. (b) d_{90}.

Fig. 9. Correlation heatmaps for the specific energy (E_m) and d_{90} responses, as a function of the following input features: impeller speed (N), slurry solids content (w), grinding media content (ϕ), batch time elapsed (t), and the initial d_{90} (d_{90}^0)

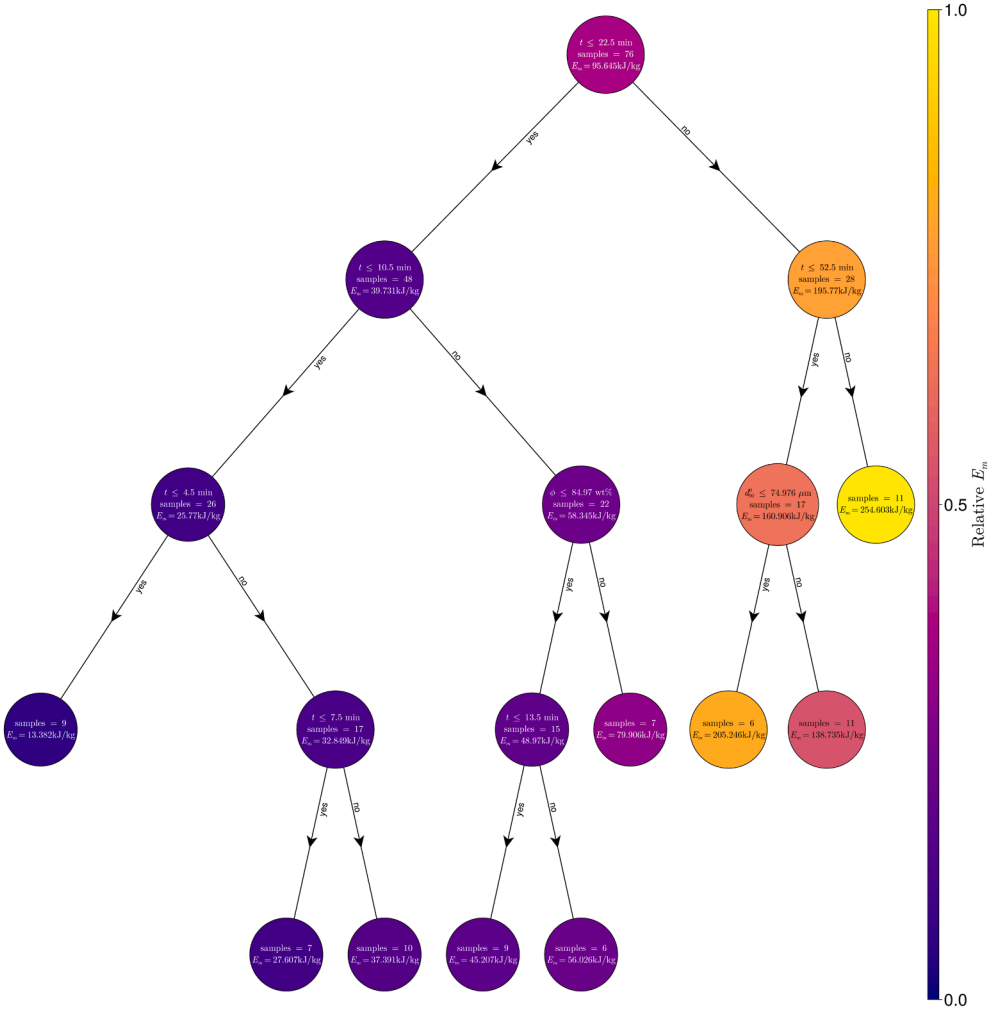


Fig. 10. RFR decision tree 9 of 37 (i.e., the 9th tree in an ensemble of 37). Lighter colours indicate higher specific energy. In each node, E_m denotes the mean specific energy prediction at that node for that particular tree, and 'samples' denotes the number of training data points that reach that node, based upon preceding splitting criteria. For non-terminal nodes, the input feature upon which a decision is made, and its threshold value are shown, where N is the impeller speed, w is the slurry solids content, ϕ is the grinding media content, t is the batch time elapsed, and d_{90}^0 is the initial d_{90} of the batch.

Unsurprisingly, most of the decisions for the specific energy response (Fig. 10) involve time elapsed, matching the strong correlation shown in Fig. 9(a). The lighter nodes on the right show higher values of specific energy, following the path that the time elapsed is greater than 22.5 min. If the time elapsed (t) is also less than or equal to 52.5 min (i.e., in the middle of the batch), then an initial d_{90} (d_{90}^0) of less than ca. 75 μm

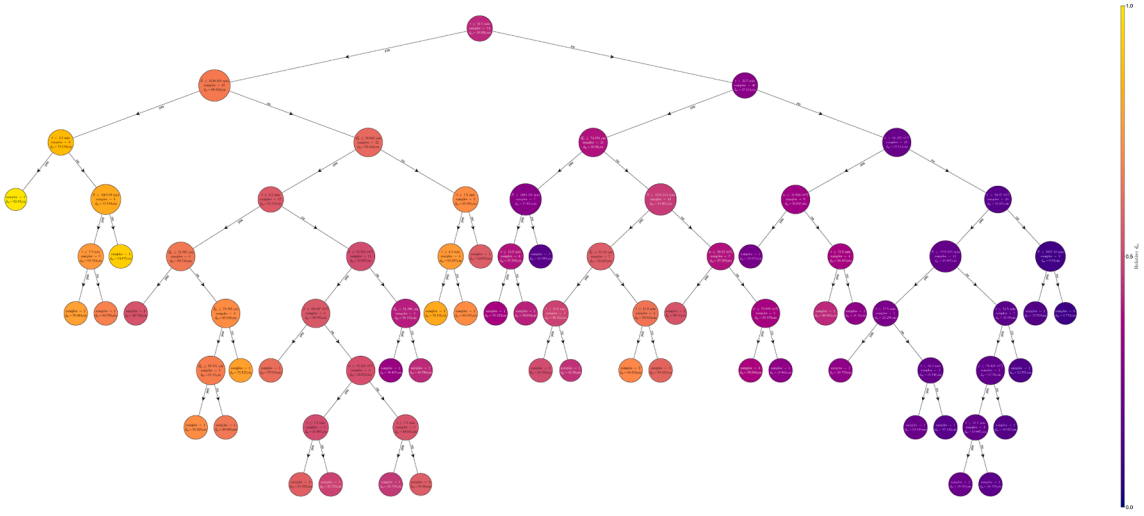
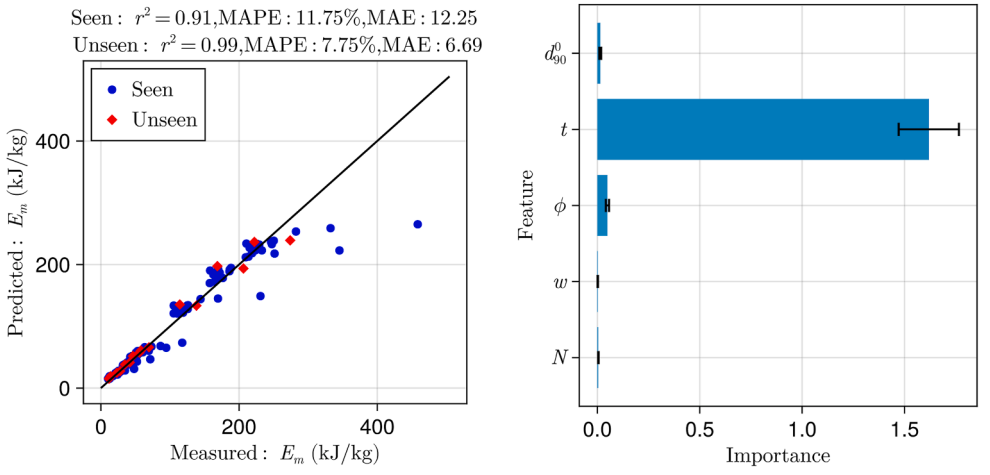
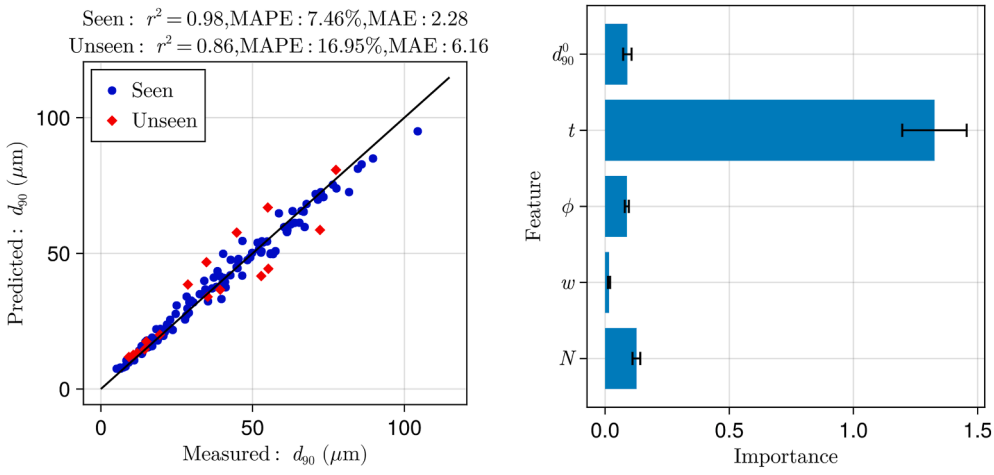


Fig. 11. RFR decision tree 1 of 27 (i.e., the 1st tree in an ensemble of 27). The scale of this tree visually demonstrates the interpretability limits of RFR. Lighter colours indicate higher d_{90}. In each node, d_{90} denotes the mean d_{90} at that node for that particular tree, and 'samples' denotes the number of training data points that reach that node, based upon preceding splitting criteria. For non-terminal nodes, the input feature upon which a decision is made, and its threshold value are shown, where N is the impeller speed, w is the slurry solids content, ϕ is the grinding media content, t is the batch time elapsed, and d_{90}^0 is the initial d_{90} of the batch.



(a) Parity plot (specific energy).

(b) Feature importance (specific energy).



(c) Parity plot (d_{90}).

(d) Feature importance (d_{90}).

Fig. 12. RFR model performance and feature importance for specific energy (top row) and d_{90} (bottom row). In b) and d), N is the impeller speed, w is the slurry solids content, ϕ is the grinding media content, t is the batch time elapsed, and d_{90}^0 is the initial d_{90} of the batch.

results in higher overall energy consumption. Provided that there is sufficient stress energy for size reduction during the entire batch, then as the particle size is further reduced, the slurry viscosity and hence resistance to motion, increases, necessitating greater energy input.

The structure of the decision trees predicting the d_{90} response are more balanced by comparison (see Fig. 11). As noted in Section 3.2, the correlations shown by Fig. 9(b) show that d_{90} is inversely related to the grinding media content and the impeller speed. The same trends are observed for the decision tree shown. As a 'good' d_{90} is small, the areas of interest are the darker nodes.

On the right of the figure, one can observe that the predicted d_{90} values decrease for the false branches from decisions relating to grinding media content and impeller speed, thus demonstrating the inverse relationship.

The observations noted in Section 3.2 are illustrated in the decision tree. Namely, that at low impeller speeds early in the batch, the poor mixing hinders the grinding action inside the basket. That is, for impeller speeds below ca. 2400 rpm and solids content more than ca. 36.5 wt% the predicted d_{90} for this particular decision tree is worse.

Fig. 12a demonstrates the RFR model predictive ability for unseen data. Performance is better for lower energies, where there are more data points for the RFR to fit to compared to the more sparse entries at higher specific energy values. It is difficult to differentiate the points on the graph into the two batches that the data are from, which suggests that the RFR is generalising well. Outliers in the parity plot correspond to the data from batch 4. As noted in Section 3.1.1, this batch also had noticeable impact on the model RMSE when it was the test batch during LOOCV. During the experiments, the particle size reduction was considerably faster than other batches. However, the final particle size for batch 12 ended up similar (5 μm vs 6.5 μm) but the specific energy input for batch 4 was 459 kJ kg^{-1} vs 332 kJ kg^{-1}. Now, as particle size decreases, the successive energy input required for further reduction increases non-linearly, but with the close final d_{90} values it appears that the conditions for batch 4 resulted in more inefficient milling than for other batches; this is perhaps the reason for batch 4's outlier status.

The importance of the input features is shown in Fig. 12(b). By far the most important parameter is time elapsed (t), which as explained previously is due to its explicit role in how specific energy input is calculated. The next two important features are grinding media content (ϕ) and the initial d_{90} (d_{90}^0) respectively.

The strong time dependence of the model is unsurprising, as the specific energy is explicitly defined as a function of time. Fig. 13 illustrates a parity plot assuming that specific energy solely depends on time elapsed - so a model of the form $E_m = kt$ would apply. The performance of this model is noticeably worse than for the RFR model, as noted by the higher Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) values, and lower r^2 score. In particular, the unseen data points straddle the parity line, with the spacing between each point increasing as milling progresses. Thus, the quality of prediction becomes increasingly incorrect further into milling; the implication here, is that the value for k is affected by process conditions and cannot be assumed to be independent of them. This explicitly demonstrates that the ML model captures non-linear rheological interactions rather than simply integrating a constant power draw, validating the necessity of the model over a simple linear baseline.

Fig. 12(c) shows that the RFR model is somewhat able to predict unseen data but not to the same degree as the specific energy model was able to as shown by the increased MAPE score for unseen predictions. Additionally, the r^2 value for the d_{90} response is lower than it was for energy, implying that the model accounts for less of the variance present in the data. Inspection of Fig. 12(c) shows that the prediction quality improves with time elapsed (decreasing particle size) where the points lie far closer to the parity line. Interestingly, the points lie almost equally either side of the parity line, implying that the model is able to pick up on some of the relationships between the input features and the response.

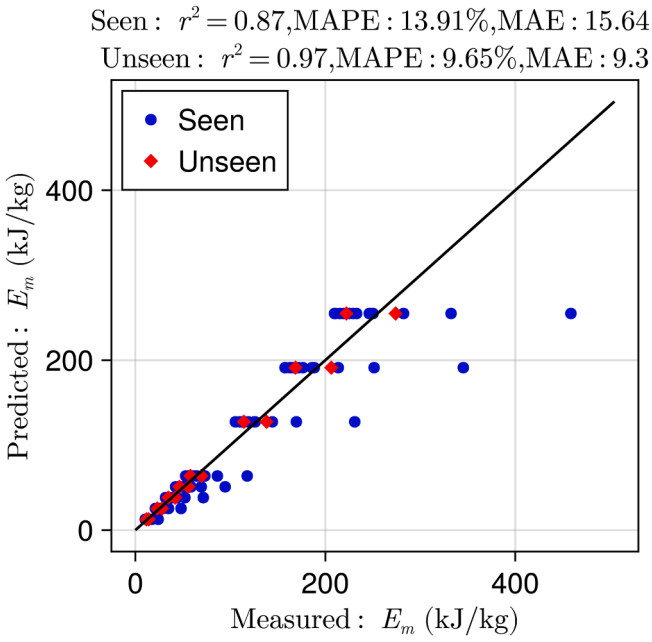


Fig. 13. Linear model ($E_m = kt$) performance for specific energy.

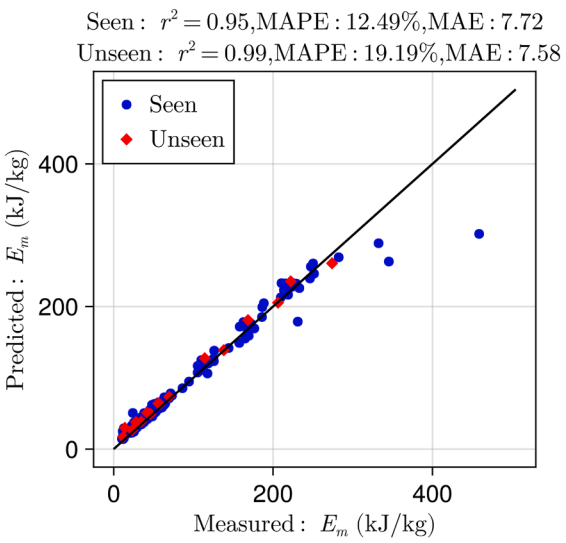
Whilst there is a still a strong time dependence for d_{90} also, other input features have greater importance than they did for predicting specific energy as shown in Fig. 12(d). Grinding media content and impeller speed having noticeable importance again ties back to the stress energy quantity present. Additionally, the initial d_{90} had a similar importance, which can explained by the fixed batch time as there is only a limited amount of time for size reduction - a larger initial size increases the amount of reduction required to achieve the same final d_{90} as using a smaller initial value.

3.4. Gradient Boosted Trees

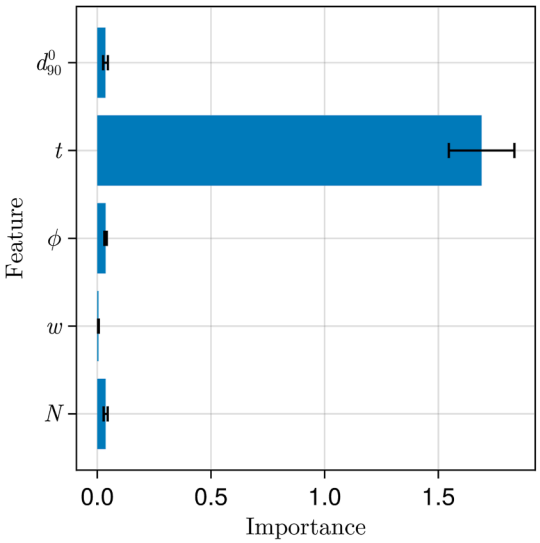
Fig. 14(a) shows that whilst the GBT model is able to explain most of the variance in the data, the prediction quality compared to the RFR was less. This is shown by the higher MAPE and MAE scores for unseen data. In the seen data case, the MAE score was lower for the GBT model but the MAPE score was higher, suggesting that the GBT model performed better for higher specific energy values, i.e. later on in a batch compared to the RFR model. However, in the unseen case, as both scores are high, the GBT model was worse throughout the entire batch. Both r^2 scores being 0.99 but demonstrably different error scores indicate that this quantity is not a meaningful way to distinguish between the two approaches.

In contrast to the RFR model, the GBT placed more importance on the impeller speed (N) and grinding media content (ϕ). This could mean that the GBT is less blinded to the effects of other features than time elapsed than the RFR model was.

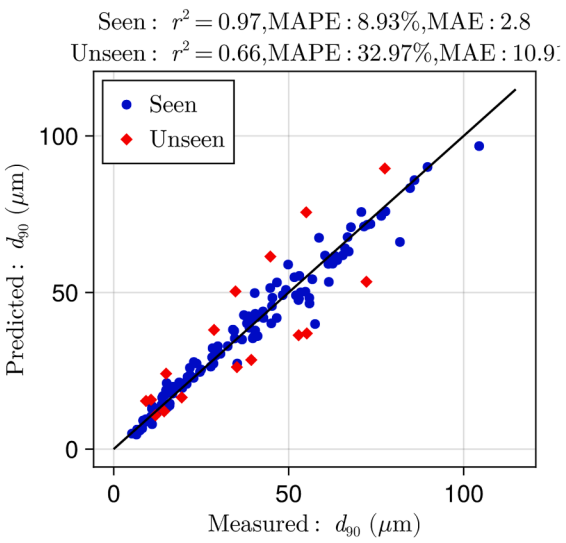
Predicting the d_{90} is where the differences between the GBT and RFR models appear in this study. As shown in Fig. 14(c), the unseen data lie further from the parity line than for the RFR model. Complementing the visual inspection is the fact that the MAPE score for the unseen data is *double* that of the RFR model for unseen data, whereas for the seen data, it is only marginally worse. This variability between architectures is a key finding: while GBT is powerful, it is prone to overfitting in this sparse-data regime. The conclusion drawn from this observation is that the GBT model is not able to generalise as well as the RFR model for the d_{90} response. Likewise, the MAE score was also higher for the GBT model. The r^2 score for the GBT model was unsurprisingly lower than for



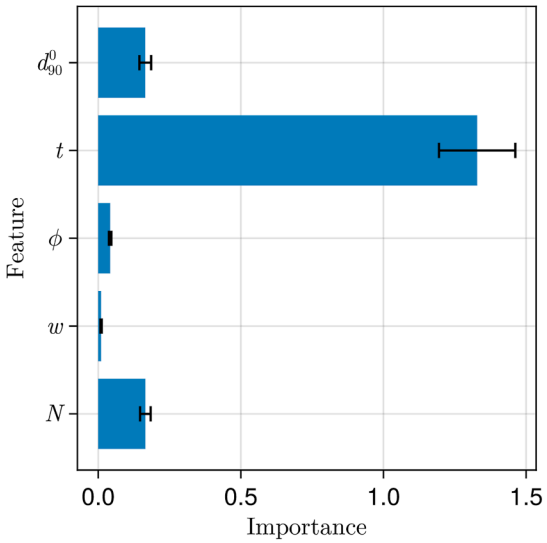
(a) Parity plot (specific energy).



(b) Feature importance (specific energy).



(c) Parity plot (d_{90}).



(d) Feature importance (d_{90}).

Fig. 14. GBT model performance and feature importance for specific energy (top row) and d_{90} (bottom row). In (b), N is the impeller speed, w is the slurry solids content, ϕ is the grinding media content, t is the batch time elapsed, and d_{90}^0 is the initial d_{90} of the batch.

the RFR model (0.66 compared to 0.86), which indicates that the GBT model accounts for less of the variance in the data.

As with specific energy, the GBT placed noticeably different importance on the input features compared to the RFR model as shown in Fig. 14(d). In particular, greater importance was placed on the initial d_{90} and impeller speed, whilst less was placed on the grinding media content. The poorer performance could thus be explained by the GBT underestimating the role that the grinding media content plays in size reduction.

3.5. Symbolic Regression

Unlike the RFR and GBT models, there are no trees to visualise with SR to illustrate decisions made. Instead, the generated equations are shown below for both responses:

$$E_m = (t + C_0)k \tag{6}$$

$$k = \left(\left(C_1 - \frac{C_2}{\phi - C_3}\right)(N - C_4) + \frac{C_5}{\phi + w - C_6} + \frac{C_7}{d_{90}^0 - C_8} + \frac{C_9}{\phi}\right) \tag{7}$$

$$d_{90} = \frac{C_{10}\phi}{\frac{t}{\left(\frac{C_{11}}{N(\phi - C_{12})}\right)^{C_{13}}} + \frac{t + C_{14}}{w} + \frac{C_{15}}{d_{90}^0}} \tag{8}$$

C_0 through C_{15} carry the units required for dimensional consistency (as enforced by the dimensional constraint discussed in section 2.4.3)

$$C_0 = 0.325 \text{ min} \tag{9}$$

$$C_1 = -0.0208 \text{ kJ kg}^{-1} \text{ min}^{-1} \text{ s rad}^{-1} \tag{10}$$

$$C_2 = 1.30 \text{ kJ kg min s rad} \tag{11}$$

$$C_3 = 125.0 \text{ wt\%} \quad (12)$$

$$C_4 = 99.6 \text{ rad s}^{-1} \quad (13)$$

$$C_5 = 0.180 \text{ kJ kg}^{-1} \text{ min}^{-1} \text{ wt\%} \quad (14)$$

$$C_6 = 99.6 \text{ wt\%} \quad (15)$$

$$C_7 = 1.52 \text{ kJ kg}^{-1} \text{ min}^{-1} \text{ µm} \quad (16)$$

$$C_8 = 105.0 \text{ µm} \quad (17)$$

$$C_9 = 231.0 \text{ kJ kg min wt\%} \quad (18)$$

$$C_{10} = 1.48 \text{ min µm wt\%}^2 \quad (19)$$

$$C_{11} = 1.84 \times 10^4 \text{ rads}^{-1} \text{ wt\%}^{1.398} \quad (20)$$

$$C_{12} = 42.9 \text{ wt\%} \quad (21)$$

$$C_{13} = 2.51 \text{ [–]} \quad (22)$$

$$C_{14} = 12.9 \text{ min} \quad (23)$$

$$C_{15} = 57.1 \text{ min µm wt\%}^{-1} \quad (24)$$

where ϕ is the grinding media content, w is the slurry solids content and d_{90}^0 is the initial, and N is the impeller speed. To maintain dimensional consistency in the SR model, the impeller speed was converted from rpm to rad s^{-1}.

These equations were selected from the candidates at the Pareto front of each response according to the 'best' criterion available in *PySR*. This criterion selects the equation with the highest score that has a loss no worse than 1.5 × the most accurate model's. Score in *PySR* is defined as the negative of the derivative of the log-loss function with respect to the equation complexity. In other words, the 'best' criterion permits a slightly higher loss than the most accurate model to reduce the complexity which promotes generalisability.

Equation (6) shows that time has a clear linear relationship with the specific energy input. Given that all other input features are fixed for the duration of the batch, then this reduces the relationship down to $E_m \approx kt + c$ where k is the rest of the equation that $(t + 0.325)$ is multiplied by and $c = 0.325k$.

This links to the observations in Section 3.3, suggesting that a linear model with process-independent gradient (k) for the specific energy input is too simplistic. One could observe that as milling progressed (increasing specific energy), the unseen data points were further and further apart which shows diverging rates of increase and that the rate of energy increase is *not* independent of process conditions. By making the gradient a function of process parameters, the model had far better predictive abilities; the MAE for the linear model vs the SR model being 9.3 vs 2.67, respectively. For d_{90}, the generated equation shows that its value is proportional to $\frac{1}{t}$. As the impeller speed, grinding media and solids content all form denominator terms relative to the time term, this means they in essence form rate constants that affect the progression of d_{90}. The solids content 'rate constant' is proportional to the reciprocal of its value, meaning that more dilute slurries will reach lower d_{90} faster, which makes sense since there are fewer particles to successfully grind.

Practically, there is a trade-off between rate to the target size and the amount of material milled, analogously to rate versus yield when considering chemical reactions where the equilibrium position matters. In the other time-containing term, the impeller speed and grinding media content both form a somewhat proportional relationship with their 'rate constant'. That is, as either of those quantities is increased, the rate of size reduction will increase accordingly. Interestingly, both terms are raised to the power of approx. 2.5, showing a strong non-linear dependence on these input features.

Compared to the tree-based methods, the SR models generated are considerably easier to interpret and give physical significance to. Whilst a single decision tree is possible to follow, an ensemble quickly becomes too complex for one to follow in its entirety. Additionally, the SR approach has minimal hyperparameters that need tuning, rather, the user is asked to simply reason about the domain for the model and the operators that may help describe the present relationships.

The parity plot in Fig. 15(a) shows that the generated Equation (6) is able to easily predict the specific energy across the entire range of the two previously unseen batches. Performance of the SR model is marginally better than the previous 2 approaches, so whilst promising, is not a definitive means of comparing the different approaches.

Feature importance for the generated equations was determined using Global Sensitivity Analysis (GSA) which ranks input features based on their impact on the subsequent output variance. In contrast to local techniques, GSA considers the entire parameter space and is able to handle nonlinearity (Wainwright et al., 2014).

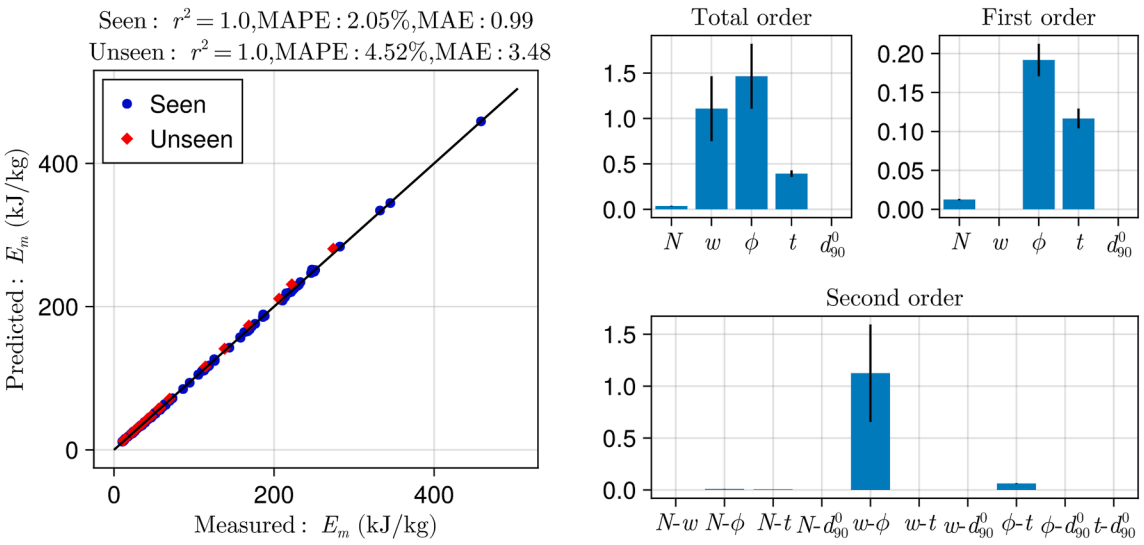
The *GlobalSensitivity.jl* package was used to perform sensitivity analysis on the generated equations by computing the Sobol indices for each equation (Dixit & Rackauckas, 2022). Sobol indices represent the portion of the total variance attributable to a given input feature, quantified by the total-order index. Further partitioning can be done into n^{th} order indices which represent the variance attributable to n-th order interactions between input features. In this study, the first and second order Sobol indices were computed, representing the contributions of singular and pairwise input feature interactions, respectively (Sobol, 2001).

The Sobol indices for Equation (6) are shown in Fig. 15(b). Looking at the total order indices, the effect of the initial d_{90} does little for the output variance of Equation (6), as the term is far smaller than for any of the other features. The greatest effects are from the solids (w) and grinding media (ϕ) content. Despite the strong linear relationship implied by Equation (6), the time elapsed (t) appears to have less of an effect on the variance, perhaps due to its linear nature; the magnitudes of the total and first-order indices for time are similar, so time has mostly first-order (linear) effects on the specific energy prediction. In contrast, the solids and grinding media content exhibit second order interactions with each other as noted by the fact that their magnitudes are similar for total-order as solids-grinding media second order interaction term. Their respective first order indices are far lower, especially for solids content.

Prediction of unseen d_{90} via SR appears to be more robust than either of the previous methods as evidenced by the lower MAPE and MAE values. Fig. 15(c) demonstrates that the predictions lie far closer to the parity line. Thus, alongside the superior interpretability of SR-generated equations compared to tree-based methods, the predictions are also more accurate.

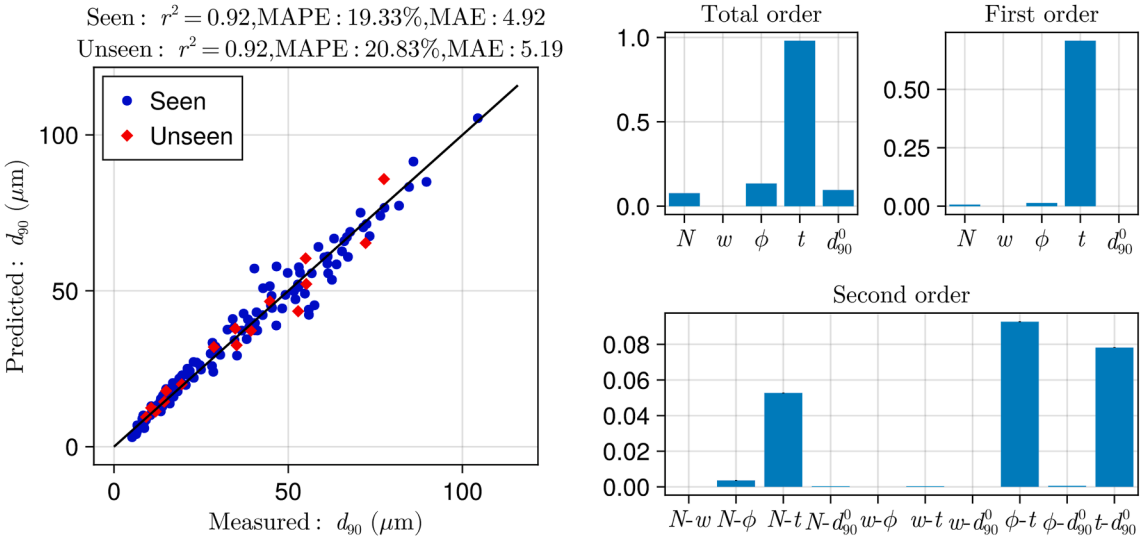
It must be highlighted that the SR model had a limited number of operators it was allowed to utilise, and within that subset, decided that exp and log were not required, instead using only the 4 basic arithmetic operators and a single power term.

The Sobol indices for Equation (8) are shown in Fig. 15(d). Time elapsed (t) showed strong effects on the output variance, which were mostly first-order in nature, given the relative magnitudes for the total and first-order indices. Impeller speed (N) and grinding media content (ϕ) both shared second-order interactions with time elapsed, again from comparing the magnitude of their total and second-order indices, presumably relating to the extremely non-linear term in Equation (8) featuring these three input features. However, compared to time elapsed, their total order indices are far lower, suggesting that the impact of impeller speed and grinding media content is less. This observation links back to earlier discussions in Section 3.2 where for a batch process like milling, once there is sufficient energy, with enough time then the particles will reduce in size; thus, time elapsed is likely to overpower other parameters. Interestingly, the solids content has little effect on the output variance, which could suggest that the size



(a) Parity plot (specific energy).

(b) Sobol indices (specific energy).



(c) Parity plot (d_{90}).

(d) Sobol indices (d_{90}).

Fig. 15. SR model performance and feature importance for specific energy (top row) and d_{90} (bottom row). In (b), the Sobol indices represent the output variance attributable to each input feature, where N is the impeller speed, w is the slurry solids content, ϕ is the grinding media content, t is the batch time elapsed, and d_{90}^0 is the d_{90} of the batch.

reduction is insensitive to the solids content (w) within the ranges used in this study; it is possible that a limiting value may exist at higher concentrations. For the immersion mill system, the limit could be due to poor slurry flow if the viscosity is too high. Finally, the initial d_{90} (d_{90}^0), unlike for specific energy, has a mostly second-order interaction with time on the output variance. An explanation is that for the same conditions, a larger initial d_{90} necessitates longer milling time to achieve a target end value.

3.6. Model suitability and limitations

While all three models successfully fit the training data, their ability to predict unseen data and therefore usefulness in an industrial application varies. In the context of this study, an evaluation of their strengths and limits is outlined below.

3.6.1. Random Forest Regression

The RFR model was able to handle the limited dataset (15 batches), where it showed good interpolative accuracy for specific energy input, and moderate accuracy for d_{90}. The bootstrapping aspect of the model helped to mitigate overfitting to specific high-variance datapoints such as in batch 4, which ensured reliability. However, this type of model cannot be used to extrapolate behaviour outside the bounds of the training data; the returned values end up capped, due to the thresholds being determined by the training data. Additionally, the decision trees for d_{90} in particular ended up being complex, which limited the physical insights available to help understand the behaviour of the immersion mill.

3.6.2. Gradient Boosted Trees

The GBT model was more sensitive to non-time features than the RFR

model, which suggests that the model architecture was less overshadowed by other features by the batch time elapsed (t) feature in comparison. However, its performance on predicting d_{90} was considerably poorer than for the RFR model. It is possible that the dataset sparsity negatively interacted with the residual-correcting architecture of the GBT model; the model was memorising noise, rather than the underlying physics. In this particular application, the GBT architecture is not suitable for use in predicting d_{90}.

3.6.3. Symbolic Regression

The SR model was able to generate closed-form algebraic expressions that can be directly utilised, without need for complex ML software stacks. Such expressions are also easily interpretable, meaning that, for instance, it was possible to observe that the model had deduced an inverse relationship between batch time and d_{90}. However, the architecture is reliant on domain expertise; the operators chosen need to be able to accurately describe the physics of the system. Additionally, the form of Equation (6) can result in unphysical outcomes (e.g. negative specific energy), with certain combinations of parameters which means that it must be carefully used to ensure sensible predictions.

3.7. Application: minimising energy usage

The development of models for predicting specific energy and d_{90} allows for the optimisation of the milling process, and/or exploration of 'what-if' scenarios. This section explores these hypothetical scenarios using the SR models, as they are the best performing models (as discussed previously) and most amenable to usage in optimisation routines as they are purely algebraic in nature. The target d_{90} values were explicitly selected to represent typical final product specifications for this particular mill configuration. It must be noted that whilst the predicted optimal specific energy inputs for these exact parameter combinations have not been subject to experimental verification, the strong performance of SR against wholly unseen data in *the exact same system* (Fig. 15a) provides confidence in the predictions outlined below.

Optimisation was carried out using the *Optimisation.jl* package (Dixit & Rackauckas, 2023), specifically making use of the *Ipopt* backend. The objective function was set to *minimise* the specific energy input, with the constraints being that:

- The d_{90} must be *exactly* the target value.
- The specific energy cannot be negative.

The latter constraint is to ensure that the optimisation routine does not try to find a solution that requires negative energy input, which is not physically possible. Negative input being possible is a criticism of Equation (6) where the correct combination of input features could lead to a negative value for k. However, in the context of this section, the negative values can be prevented by careful choice of constraints.

The first scenario explored is changing the target d_{90} value whilst keeping the same feed (same initial d_{90}) - all other input features are allowed to vary (within bounds set out in Table 1). The results of the optimisation scenarios are shown in Table 3.

For all target d_{90} values, the optimisation routine found that the mill is to be run at maximum speed for optimal energy use. For the 5 μm target and 7 μm targets, a full hour is required to achieve the target d_{90}. Further, increasing the target d_{90} to 9 μm reduced the batch time by almost 25%. Interestingly, the required solids content increased with increasing target size, whilst the grinding media content decreased. In a practical sense, this means that the mill throughput needs to decrease with decreasing target size to maintain optimal energy usage. To reiterate, the scenario explored here is hypothetical, it is entirely possible to fix any of the input features as required by the process.

Table 3
Predicted specific energy input for different d_{90} targets.

	Target d_{90} (μm)		
Variable	5	7	9
Impeller speed (rpm)	3000	3000	3000
Slurry solids content (wt%)	37.72	44.32	45.0
Grinding media content (wt%)	81.88	75.28	74.60
Batch time (min)	60.0	60.0	47.66
Specific energy input (kJ kg^{-1})	123.74	72.24	54.45

The second scenario explored is changing the initial d_{90} to represent a disturbance in the feed material, such as a change in the material supplier or an upstream process change/disturbance. The results of the optimisation scenarios are shown in Table 4.

Interestingly, in all scenarios, the mill should be run at maximum speed to achieve the target d_{90}. The required batch time is also approximately the same as for the initial scenario - the full hour being required for all but the largest target d_{90}. Between the initial d_{90} values, each input feature shows little variation but in the same direction. This means that whilst there is a meaningful relationship between the initial d_{90} and the optimal process parameters, the relationship is quite weak. Recalling Fig. 15(b), the initial d_{90} has a miniscule effect on the output variance for both responses, showing that this mill configuration is not sensitive to a 25% change in the initial d_{90} in either direction. This is a positive finding, as it means that the mill can be run at the same process parameters for a wide range of initial d_{90} values, thus reducing the need for re-optimisation after a disturbance.

Both scenarios demonstrate the models' ability to support decision-making and reduce the energy consumption of the mill. Rather than a trial-and-error approach, the operator can instead input the initial d_{90} (d_{90}^0), as well as the target d_{90} to the model. In return, the optimisation routine can return the optimal setpoints for the solids content (w), impeller speed (N), grinding media content (ϕ), and the required batch time (t).

4. Conclusions

This work has explored the parameter space of an immersion mill, and used data gathered during the process to create and validate multiple machine learning models to predict and optimise mill performance. These models were:

Table 4
Predicted specific energy input for different d_{90} targets, considering the scenario where the initial d_{90} is either bigger or smaller than usual.

Variable	Initial d_{90}		
	60	80	100
Target d_{90}: 5 μm			
Impeller speed (rpm)	3000.0	3000.0	3000.0
Slurry solids content (wt%)	37.93	37.72	37.60
Grinding media content (wt%)	81.67	81.88	82.0
Batch time (min)	60.0	60.0	60.0
Specific energy input (kJ kg^{-1})	121.90	123.74	124.62
Target d_{90}: 7 μm			
Impeller speed (rpm)	3000.0	3000.0	3000.0
Slurry solids content (wt%)	44.59	44.32	44.15
Grinding media content (wt%)	75.01	75.28	75.45
Batch time (min)	60.0	60.0	60.0
Specific energy input (kJ kg^{-1})	70.42	72.24	73.1
Target d_{90}: 9 μm			
Impeller speed (rpm)	3000.0	3000.0	3000.0
Slurry solids content (wt%)	45.0	45.0	45.0
Grinding media content (wt%)	74.60	74.60	74.60
Batch time (min)	46.65	47.66	48.26
Specific energy input (kJ kg^{-1})	53.4	54.45	54.86

- Random Forest Regression.
- Gradient Boosted Trees.
- Symbolic Regression.

By utilising smart sampling strategies that target high-variance regions in the 5-dimensional parameter space, a small number of batches (15) were sufficient to gain useful, accurate insights into the immersion mill's performance. Only requiring a small number of runs to understand the specific energy and particle d_{90} for the mill minimised material usage during the experiments and enabled rapid discernment of important process parameters.

Based on unseen data prediction accuracy and interpretability, the SR models performed the best in this study. For the unseen data, the SR model for specific energy had a MAE of 3.48 kJ kg^{-1} and for d_{90} 5.19 μm. Additionally, the SR models were easily able to be used to set mill process parameters to achieve target d_{90} sizes whilst minimising energy usage, seeing as how they were purely algebraic in nature.

The findings within this study form the basis for further work which, for instance, could explore different materials and thus incorporate material properties into the predictive models. As the sampling method is readily extensible, then pre-existing data can be fed into subsequent studies - if any new features are identified then these too are easily added.

CRediT authorship contribution statement

Daniel Weston: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Li Liu:** Writing – review & editing, Supervision. **Christopher Windows-Yule:** Writing – review & editing, Supervision.

Acknowledgements

Special thanks go to Dr Carl Tipton and Dr Ben Watson for their assistance in configuring the Raspberry Pi for data acquisition.

This work was undertaken as part of an EngD project at the Centre for Doctoral Training in Formulation Engineering, with funding from EPSRC award EP/S023070/1 and Johnson Matthey.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*.

Aramendia, E., Brockway, P. E., Taylor, P. G., & Norman, J. (2023). Global energy consumption of the mineral mining industry: Exploring the historical perspective and future pathways to 2060. *Global Environmental Change, 83*, Article 102745. https://doi.org/10.1016/j.gloenvcha.2023.102745

Bilgili, E. (2024). Population balance modeling of milling processes: Are we falsifying breakage kinetics and distribution via back-calculation methods? *Powders, 3*(2), 190–201. https://doi.org/10.3390/powders3020012

Bilgili, E., Toprak, A., Altun, D., & Altun, O. (2024). Pbm of an industrial-scale vertical wet stirred media mill (higmill): Assessment of back-calculation and hybrid methods. *Powder Technology, 440*, Article 119760. https://doi.org/10.1016/j.powtec.2024.119760

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Capece, M., Bilgili, E., & Dave, R. (2011). Identification of the breakage rate and distribution parameters in a non-linear population balance model for batch milling. *Powder Technology, 208*(1), 195–204. https://doi.org/10.1016/j.powtec.2010.12.019

Cava, W. L., Orzechowski, P., Burlacu, B., de França, F. O., Virgolin, M., Jin, Y., Kommenda, M., & Moore, J. H. (2021). Contemporary symbolic regression methods and their relative performance. arXiv:2107.14351 https://doi.org/10.48550/arXiv.2107.14351 https://arxiv.org/abs/2107.14351.

Chehreh Chelgani, S., Nasiri, H., & Tohry, A. (2021). Modeling of particle sizes for industrial hpgr products by a unique explainable ai tool- a "conscious lab" development. *Advanced Powder Technology, 32*(11), 4141–4148. https://doi.org/10.1016/j.apt.2021.09.020

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, KDD '16* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785. URL.

Cranmer, M. (2023). Interpretable machine learning for science with PySR and symbolicRegression.jl. arXiv:2305.01582 [astro-ph, physics:physics] http://arxiv.org/abs/2305.01582.

Daraio, D., Villoria, J., Ingram, A., Alexiadis, A., Stitt, E., & Marigo, M. (2020). Investigating grinding media dynamics inside a vertical stirred mill using the discrete element method: Effect of impeller arm length. *Powder Technology, 364*. https://doi.org/10.1016/j.powtec.2019.09.038

Dixit, V. K., & Rackauckas, C. (2022). Globalsensitivity. jl: Performant and parallel global sensitivity analysis with julia. *Journal of Open Source Software, 7*(76), 4561.

Dixit, V. K., & Rackauckas, C. (2023). *Optimization.jl: A unified optimization package*. https://doi.org/10.5281/zenodo.7738525

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gao, M.-W., & Forssberg, E. (1993). A study on the effect of parameters in stirred ball milling. *International Journal of Mineral Processing, 37*(1), 45–59. https://doi.org/10.1016/0301-7516(93)90004-T

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). *Why do tree-based models still outperform deep learning on tabular data?*. https://doi.org/10.48550/arXiv.2207.08815

Hutter, F., Hoos, H., & Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance. In E. P. Xing, & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning, vol. 32 of proceedings of machine learning research* (pp. 754–762). Bejing, China: PMLR. URL https://proceedings.mlr.press/v32/hutter14.html.

Jenkins, B. D., Nicusan, A. L., Neveu, A., Lumay, G., Francqui, F., Seville, J. P. K., & Windows-Yule, C. R. K. (2025). Identifying optimal regression models for dem simulation datasets. arXiv:2508.05308 https://arxiv.org/abs/2508.05308.

Jeswiet, J., & Szekeres, A. (2016). Energy consumption in mining comminution. *Procedia CIRP, 48*, 140–145. https://doi.org/10.1016/j.procir.2016.03.250. the 23rd CIRP Conference on Life Cycle Engineering.

Jones-Salkey, O., Windows-Yule, C., Ingram, A., Stahler, L., Nicusan, A., Clifford, S., Martin de Juan, L., & Reynolds, G. (2024). Using ai/ml to predict blending performance and process sensitivity for continuous direct compression (cdc). *International Journal of Pharmaceutics, 651*, Article 123796. https://doi.org/10.1016/j.ijpharm.2024.123796

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

Kumar, A., Sahu, R., & Tripathy, S. K. (2023). Energy-efficient advanced ultrafine grinding of particles using stirred mills–a review. *Energies, 16*(14). https://doi.org/10.3390/en16145277

Kwade, A. (2004). Mill selection and process optimization using a physical grinding model. *International Journal of Mineral Processing, 74*, S93–S101. https://doi.org/10.1016/j.minpro.2004.07.027. special Issue Supplement Comminution 2002.

Kwade, A., & Schwedes, J. (2007). Chapter 6 wet grinding in stirred media mills. In *Handbook of powder technology*. Elsevier. https://doi.org/10.1016/S0167-3785(07)12009-1.

Lu, M., Xia, Y., Bhattacharjee, T., Klinger, J., & Li, Z. (2024). Predicting biomass comminution: Physical experiment, population balance model, and deep learning. *Powder Technology, 441*, Article 119830. https://doi.org/10.1016/j.powtec.2024.119830

Malvern Panalytical. (2017). Mastersizer 3000 customer training course. URL https://www.malvernpanalytical.com/en/assets/malvern%20panalytical%20-%20mastersizer%203000%20training%20materials_tcm50-97739.pdf.

Munu, I., Nicusan, A. L., Crooks, J., Pitt, K., Windows-Yule, C., & Ingram, A. (2025). Using in-line measurement and statistical analyses to predict tablet properties compressed using a styl'one compaction simulator: A high shear wet granulation study. *International Journal of Pharmaceutics, 669*, Article 125098. https://doi.org/10.1016/j.ijpharm.2024.125098

Nicusan, A.-L., & Windows-Yule, K. (2022). M2E3D: Multiphase materials exploration via evolutionary equation discovery. URL https://github.com/uob-positron-imaging-centre/MED.

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., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Radziszewsk, P. (2013). Assessing the stirred mill design space. *Minerals Engineering, 41*, 9–16. https://doi.org/10.1016/j.mineng.2012.10.012

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. The MIT Press. https://doi.org/10.7551/mitpress/3206.001.0001

Roeder, D. (2024). *Pylogix*. URL https://github.com/dmroeder/pylogix.

Schilde, C., Beinert, S., & Kwade, A. (2011). Comparison of the micromechanical aggregate properties of nanostructured aggregates with the stress conditions during stirred media milling. *Chemical Engineering Science, 66*, 4943–4952. https://doi.org/10.1016/j.ces.2011.07.006

Shah, Y. (1991). Design parameters for mechanically agitated reactors. In *Vol. 17 of advances in chemical engineering* (pp. 1–206). Academic Press. https://doi.org/10.1016/S0065-2377(08)60115-5.

Shan, P., Peng, S., Bi, Y., Tang, L., Yang, C., Xie, Q., & Li, C. (2014). Partial least squares–slice transform hybrid model for nonlinear calibration. *Chemometrics and Intelligent Laboratory Systems, 138*, 72–83. https://doi.org/10.1016/j.chemolab.2014.07.015

Sobol, I. (2001). Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation, 55*(1), 271–280. https://doi.org/10.1016/S0378-4754(00)00270-6. the Second IMACS Seminar on Monte Carlo Methods.

Sterling, D., Breitung-Faes, S., & Kwade, A. (2023). Experimental evaluation of the energy transfer within wet operated stirred media mills. *Powder Technology, 425*, Article 118579. https://doi.org/10.1016/j.powtec.2023.118579

Szilagyi, B., & Nagy, Z. K. (2019). Model-based analysis and quality-by-design framework for high aspect ratio crystals in crystallizer-wet mill systems using gpu acceleration enabled optimization. *Computers & Chemical Engineering, 126*, 421–433. https://doi.org/10.1016/j.compchemeng.2019.04.025

Taylor, L., Skuse, D., Blackburn, S., & Greenwood, R. (2020). Stirred media mills in the mining industry: Material grindability, energy-size relationships, and operating conditions. *Powder Technology, 369*, 1–16. https://doi.org/10.1016/j.powtec.2020.04.057

Wainwright, H. M., Finsterle, S., Jung, Y., Zhou, Q., & Birkholzer, J. T. (2014). Making sense of global sensitivity analyses. *Computers & Geosciences, 65*, 84–94. https://doi.org/10.1016/j.cageo.2013.06.006. tOUGH Symposium 2012.

Yang, Y. (2018). *A study of fine particle grinding in vertically stirred media mills via positron emission particle tracking and the discrete element method*. Birmingham, UK: University of Birmingham. Ph.D. thesis.