minerals

MDPI

Article

Prediction and Optimisation of Copper Recovery in the Rougher Flotation Circuit

Bismark Amankwaa-Kyeremeh [1], Conor McCamley [2], Max Zanin [1,3], Christopher Greet [4], Kathy Ehrig [2] and Richmond K. Asamoah [1,*]

[1] University of South Australia, UniSA STEM, Future Industries Institute, Mawson Lakes, Adelaide, SA 5095, Australia
[2] BHP Olympic Dam, Adelaide, SA 5000, Australia
[3] School of Chemical Engineering, The University of Adelaide, North Terrace, Adelaide, SA 5005, Australia
[4] Magotteaux Pty Ltd., Rear of 31 Cormack Rd., Wingfield, SA 5013, Australia
* Correspondence: richmond.asamoah@unisa.edu.au

Abstract: In this work, the prediction and optimisation of copper flotation has been conducted in the rougher flotation circuit. The copper-recovery prediction involved the application of support vector machine (SVM), Gaussian process regression (GPR), multi-layer perceptron artificial neural network (ANN), linear regression (LR), and random forest (RF) algorithms on 15 rougher flotation variables at the BHP Olympic Dam. The predictive models' performance was assessed using linear correlation (r), root mean square error (RMSE), mean absolute percentage error (MAPE), and variance accounted for (VAF). A simulated annealing (SA) optimisation algorithm, particle swarm optimisation (PSO) algorithm, surrogate optimisation (SO) algorithm, and genetic algorithm (GA) were investigated, using the GPR predictive function, to determine the optimal operating condition for maximising copper recovery. The predictive function of the best-performing model was extracted and used in optimising the flotation circuit. The results showed that the GPR model developed with the matern 3/2 kernel function makes the most precise copper-recovery prediction as compared to the other investigated predictive models, obtaining r values > 0.96, RMSE values < 0.42, MAPE values $< 0.25\%$, and VAF values $> 94\%$. A hypothetical optimisation solution assessment showed that SA provides the best set of solutions for the maximisation of rougher copper recovery, obtaining a throughput of 638.02 t/h and a total net gain percentage of 14%–15.5% over the other optimisation algorithms with a maximum copper recovery of 94.76%. The operational benefits of implementing these algorithms have been highlighted.

Keywords: froth flotation; gaussian process regression (GPR); machine learning; copper recovery; random forest; simulated annealing (SA) optimisation; process optimisation

check for updates

Citation: Amankwaa-Kyeremeh, B.; McCamley, C.; Zanin, M.; Greet, C.; Ehrig, K.; Asamoah, R.K. Prediction and Optimisation of Copper Recovery in the Rougher Flotation Circuit. *Minerals* **2024**, *14*, 36. https://doi.org/10.3390/min14010036

Academic Editor: Luis Vinnett

Received: 27 October 2023
Revised: 11 December 2023
Accepted: 19 December 2023
Published: 28 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The demand for copper is ever-increasing, playing a critical role in the transition to a clean-energy economy. By the early 21st century, most of the rich copper oxide ores had already been mined out, leaving current deposits with a grade sometimes less than the tailings of earlier mining [1,2]. A study conducted by Calvo et al. [3] revealed that the average copper grade is constantly reducing overtime (25% reduction in just 10 years) with increasing energy consumption (46% energy increase) and total material production (30% production increase). The continuous increase in production (2%–3% per year forecasted production increase between 2010 and 2050) due to rising demand [4] requires more efficient methods across the mineral-processing value chain (e.g., comminution, flotation, and hydrometallurgy) while eliminating process waste [5–13].

Froth flotation has seen remarkable widespread applications in the mineral industry for different highly valuable commodities (e.g., copper, gold, zinc, and rare earth elements) [14–21]. The interdependence of the process variables extends performance

challenges to the flotation process where a change in feed mineralogy requires a corresponding change in the other flotation variables for an optimal outcome [22–31]. Failure to properly adjust the flotation conditions may result in a concentrate dilution and recovery of waste minerals which could contribute negatively to downstream unit efficacy and financial loss [28,32,33]. Based on the complexity of the process, froth flotation modelling to predict recovery and grade has been encouraged for the process control and optimisation of flotation plants [14,34–39]. In recent works by Gomez-Flores et al. [40] and Pu et al. [34], it was recommended that machine learning modelling is more suitable for the empirical modelling of a multivariate unit operation, especially when there are repeated patterns and high-quality measurements of the variables affecting the process. The implementation of these models for large-scale mining operations are scarce, and how different machine learning models perform is unclear.

In our previous work, a Gaussian process regression (GPR) algorithm was used to predict copper recovery from selected rougher flotation variables [33]. We further used GPR in investigating the role of pulp chemistry variables on copper-recovery prediction [28]. Allahkarami et al. [41] also applied artificial neural networks (ANNs) in estimating the copper recovery of a flotation concentrate based on the operational parameters of an industrial flotation process, yielding an r value of 0.92 for the testing phases. A key question, however, is whether simpler or easy to interpret machine learning models can be used, and what will be their predictive performance for process optimisation? For the industrial application of machine learning algorithms, it is normally advisable to investigate multiple algorithms and select the best performing model due to the uniqueness of each mineral processing plant and data distribution. Evaluating different predictive models on large-scale industrial data is vital in understanding and integrating machine learning into mineral-processing circuits.

In process optimisation, optimal parameters are determined for peak-process performance following the development of a process model that accurately relates process variables (inputs) and process key performance indicators (outputs) [42]. The predictive function of a good machine learning algorithm can be extracted as the objective function of a process and subsequently used for optimisation [43]. This ensures the implementation of a more efficient model with high predictive performance, allowing a more stabilised process. The main advantage of machine learning process optimisation is that thousands of possible solutions can be investigated to find the best ones. The limited flotation works that have been conducted in the literature using this technique include the work performed by Massinaei et al. [44] where they applied an ANN and gravitational search algorithm (GSA) in optimising the metallurgical performance of an industrial flotation column in terms of gas velocity, slurry solids, frother dosage, and froth depth. Recently, Jamróz et al. [45] also applied an ANN and evolutionary algorithm (EA) in optimising the copper-flotation enrichment process of a Polish copper ore by finding the optimal feed particle size, cleaning flotation time, and collector dosage. It is unclear whether ANN will work better for Australian copper mines, in terms of recovery prediction and which optimisation algorithm can produce the best results in maximising copper recovery within process constraints. How ANN compares with GPR and the other investigated predictive models is of interest. Four optimisation algorithms (simulated annealing (SA), particle swarm optimisation (PSO), surrogate optimisation (SO), and genetic algorithm (GA)) have been applied, using the best performing predictive model from the copper-recovery prediction as the objective function.

This study aims at investigating the predictive performance of selected supervised machine learning algorithms (artificial neural networks, Gaussian process regression, random forest, linear regression, and support vector machines) for optimising copper recovery. The optimisation study involved the application of a genetic algorithm, surrogate optimisation, simulated annealing, and particle swarm optimisation using the objective function extracted from the best-performance predictive machine learning model.

2. Research Methodology

This section presents the methodology for this research, involving data acquisition and preprocessing, model development, theoretical overview of the predictive and optimisation algorithms, and the models' performance assessment. A mean square error (MSE) assessment criterion as shown in Equation (1) was utilised. All the algorithms used in this work were carried out using MATLAB R2020a (64-bit version) software.

$$\text{MSE} = \left(\frac{1}{n}\right) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (1)$$

where $y_i = i$th true rougher copper-recovery value, $\hat{y}_i = i$th predicted rougher copper-recovery value, and $n =$ total number of observations.

2.1. Data Acquisition and Preprocessing

Data used in this study were obtained from BHP Olympic Dam, South Australia [46,47]. Specifically, data from the rougher circuit which consists of five flotation cells were used for this work. Online sensors were used to monitor process variables such as throughput, froth depth, and reagents (xanthate and frother), while copper recovery was estimated from an onstream analyser (OSA), applying Equation (2).

$$\text{Recovery, } R_i = \left(\frac{c_i}{f_i}\right)\left(\frac{f_{i-t_i}}{c_{i-t_i}}\right) \times 100\%, \; i = 1, 2, 3, \ldots, \quad (2)$$

where $c_i = i$th concentrate grade, $f_i = i$th feed grade, and $t_i = i$th tails grade.

A total of 1.4 million observations each on 15 rougher flotation variables, selected as the key recommended variables by the plant process engineers, with their corresponding recovery values were collected, representing 6 years of historical data with a confidence of 100%. The long span in terms of the data set considered for this work is to make sure almost all possible changes that occur in the plant are captured. The collected data set featured variables like feed grade, feed particle size, throughput, xanthate and frother dosages, air flow rate, and froth depth. The major issue with the data set was the transient operation observations. These are observations that were recorded in a nonsteady state usually after a plant shutdown. Since these observations are known to have a detrimental effect on the overall performance of a predictive model, they were flagged as outliers and were deleted before subjecting the remaining portion of the data set to further preprocessing. The further preprocessing of the data set was to clean the difficult-to-detect outliers, and this was carried out based on the domain knowledge of the steady operating bound of each rougher flotation variable. To ensure a same-size data set for the analysis, outliers detected in the data of a particular variable were deleted alongside with values in the remaining variables' data. The overall preprocessing of the data set resulted in 1,325,270 useful observations for the analysis. Table 1 outlines the various rougher flotation variables used for this work.

For the purpose of data confidentiality, standardised data, determined using Equation (3) (zscore transformation), are presented throughout this work. Figure 1 visualises the variation in the rougher flotation variables considered in this work.

$$z_i = \frac{s_i - \bar{s}}{s_s}, i = 1, 2, 3, \ldots, \quad (3)$$

where $z_i = i$th standardised observation, $s_i = i$th observation of a sample, $\bar{s} =$ mean of a sample, and $s_s =$ standard deviation of a sample. Visualisation of the variables is shown in Figure 1.

Table 1. Summary of variable types used in model development.

Rougher Flotation Variable		Variable Type
Feed grade (wt. %)		Input variables
Feed particle size (% passing 75 µm)		
Throughput (t/h)		
Xanthate dosage (mL/min)	Tank cell 1	
	Tank cell 4	
Frother dosage (mL/min)	Tank cell 1	
	Tank cell 4	
Air flow rate (m^3/h)	Tank cell 1	
	Tank cell 2	
	Tank cell 3	
	Tank cell 4	
	Tank cell 5	
Froth depth (mm)	Tank cell 1	
	Tank cell 2/3 *	
	Tank cell 4/5 *	
Copper recovery (%)		Output variable

* Froth depth of tank cells 2 and 4 also represent tank cells 3 and 5, respectively, as they are kept at same level.

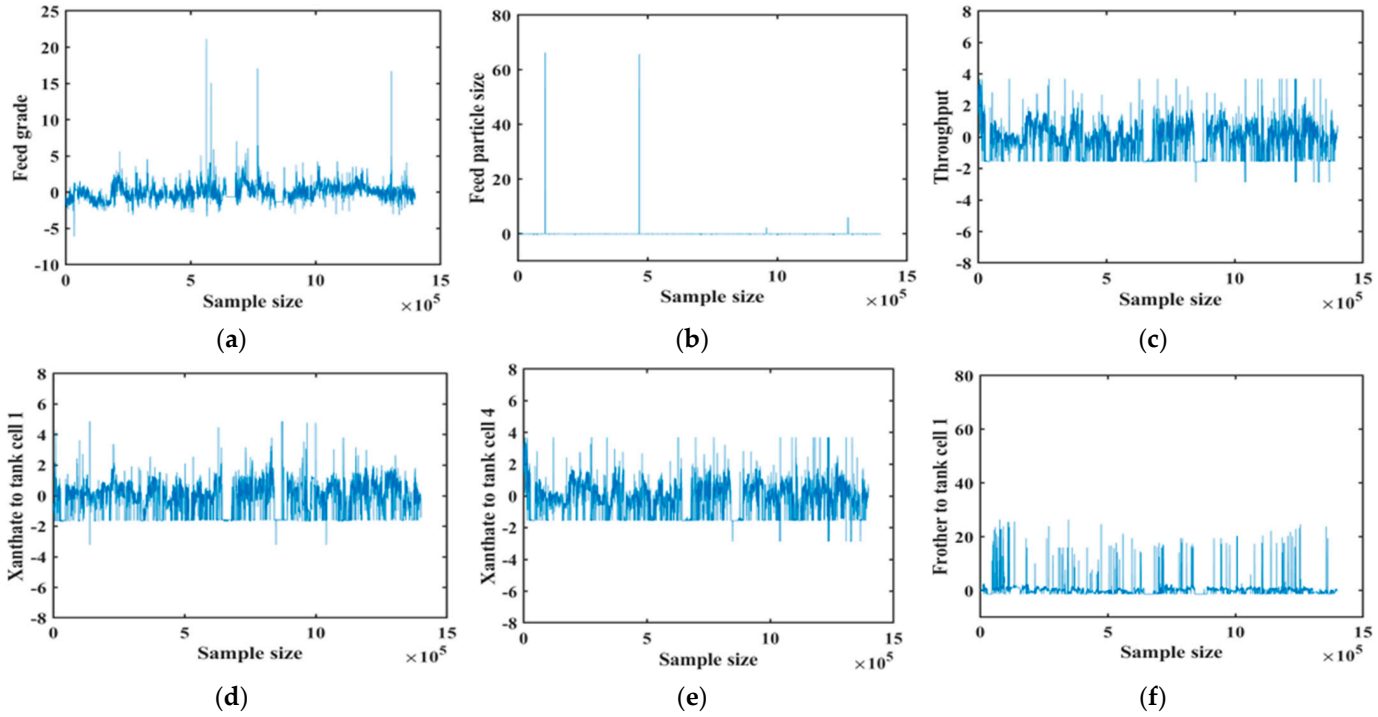


Figure 1. *Cont.*

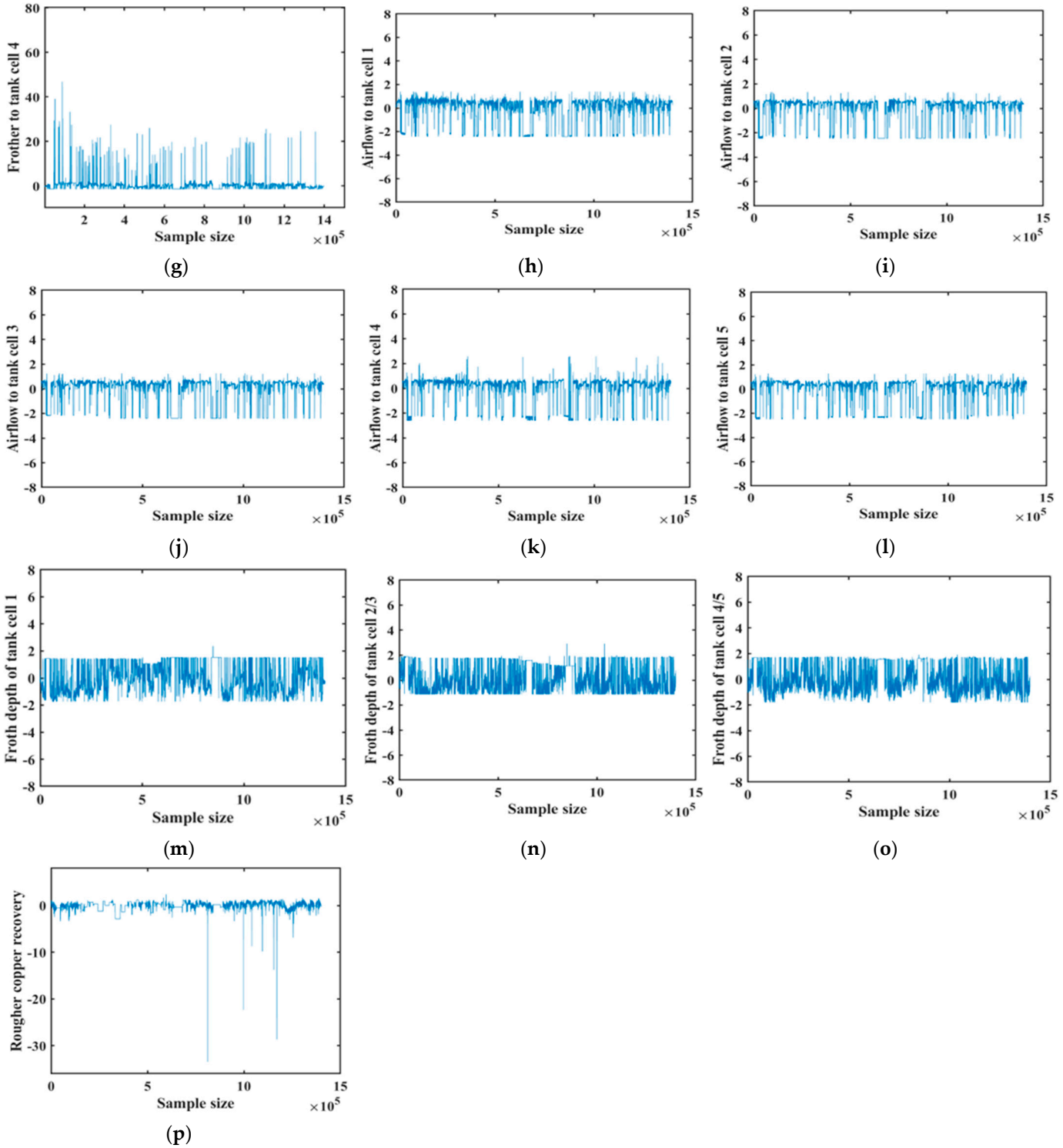


Figure 1. Visualisation of variations in (**a**) feed grade, (**b**) feed particle size, (**c**) throughput (**d**), xanthate to tank cell 1 (**e**), xanthate to tank cell 4, (**f**) frother to tank cell 1, (**g**) frother to tank cell 4, (**h**) airflow to tank cell, 1 (**i**) airflow to tank cell 2, (**j**) airflow to tank cell 3, (**k**) airflow to tank cell 4, (**l**) airflow to tank cell 5, (**m**) froth depth of tank cell 1, (**n**) froth depth of tank cell 2/3, (**o**) froth depth of tank cell 4/5, and (**p**) rougher copper recovery.

2.2. Predictive Model Development

Five predictive algorithms (SVM, GPR, ANN, LR, and RF) were used to assess the relationship between the input and output variable(s) outlined in Table 1. A total of

30,000 observations were randomly sampled out of the entire 1,325,270 observations for the training and validation of the models due to the high computational time that comes with the use of a large data set for model training. It must be noted that the best representative sample was chosen following repetitive random sampling until an error of less than 5% was attained between the sampled and population summary statistics. Figure 2 visualises the population and sampled data summary statistics of copper recovery and some rougher flotation variables for the purpose of brevity. A total of 30,000 observations was chosen as the optimal data size after a careful preliminary study (Table 2) using different data sizes while monitoring the computational time and training error. As clearly shown in Table 2, going beyond 30,000 observations only increases the computational cost with no significant improvement in model performance. The preliminary study was carried out using an LR model due to its fast execution time.

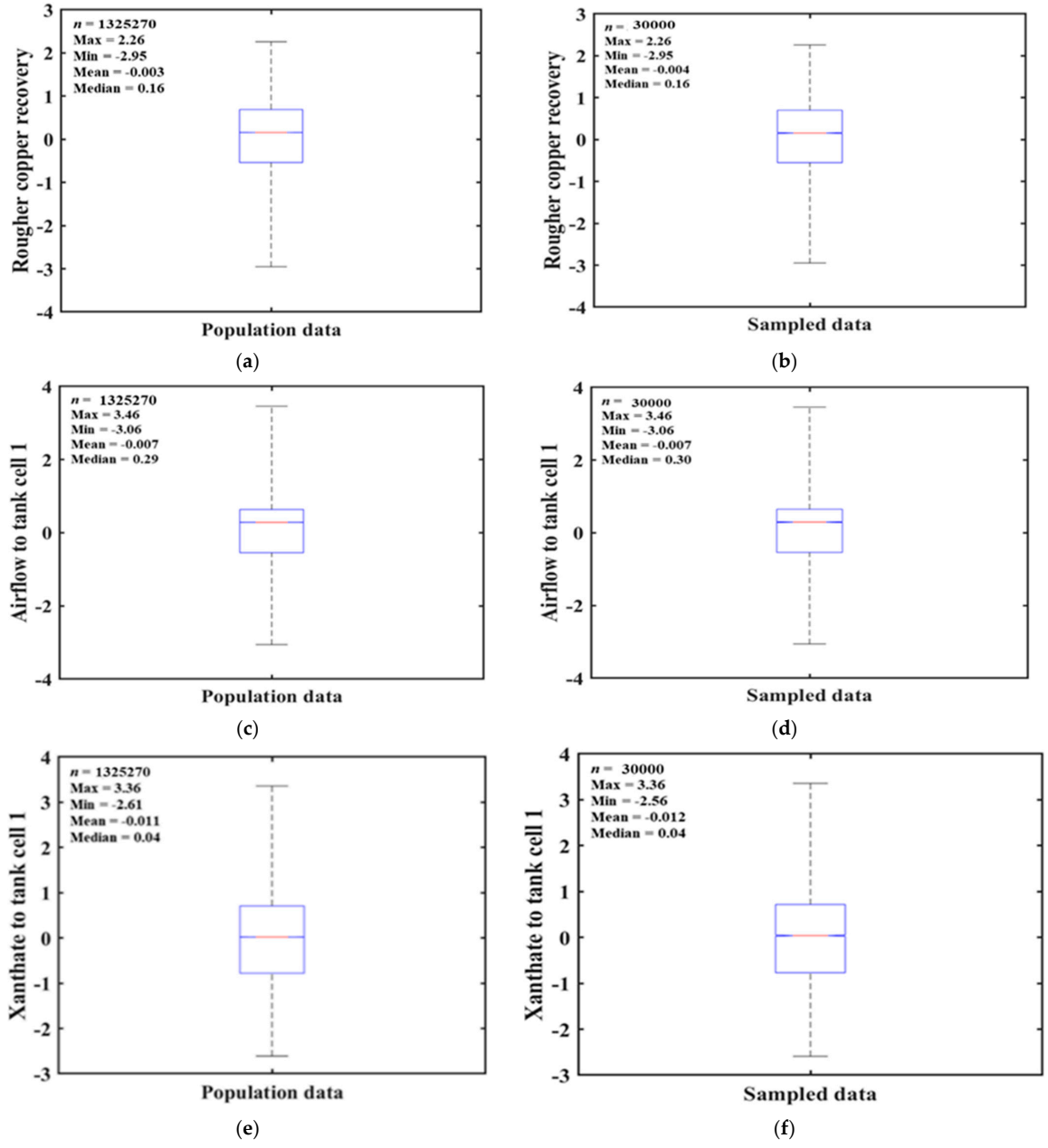


Figure 2. Summary statistics of (**a**) population data of rougher copper recovery, (**b**) sampled data of copper recovery, (**c**) population data of airflow to tank cell 1, (**d**) sampled data of airflow to tank cell 1, (**e**) population data of xanthate to tank cell 1, and (**f**) sampled data of xanthate to tank cell 1.

Table 2. Results of sample size and the corresponding computational time and training mean square error.

Sample Size, *n*	Computational Time, Seconds	Training MSE
10,000	60	2.30
20,000	138	1.92
25,000	162	1.85
30,000	180	1.79
45,000	216	1.79
60,000	270	1.80
80,000	312	1.79
100,000	360	1.78
120,000	396	1.78

To generate a good prediction, a hold-out cross-validation approach was used to randomly divide the sampled data set into an 80% (24,000 observations) training data set and 20% (6000 observations) validation data set. In as much as there is no general rule for the partitioning ratio, the rule of thumb is that the training data set must be significantly greater than the validation dataset to capture the entire characteristic of the data set. The remaining 1,295,270 observations were used as the testing data set. Each model was trained with the training data set and fitted with the training, validation, and testing data sets for performance assessment.

2.2.1. Support Vector Machine Algorithm

SVM is a popular supervised machine learning algorithm for solving both classification and regression problems, first identified by Vladimir Vapnik and his colleagues in 1992 [48]. Operating similarly to ANN, SVM can be considered as a two-layer network with linear and nonlinear weights in the first and second layers, respectively [49]. The main regression goal of SVM is to establish a predictive function given a training data set $T = \{(x_i, y_i)\}_{i=1,2,3,..n}$, where x_i denotes the ith multivariable inputs, y_i denotes the corresponding x_i output, and n is the total number of observations. To achieve this, the input x is first mapped into a feature space using some nonlinear function, finally constructing a linear model in this feature space. The linear regression function (in the feature space) implemented in SVM is expressed in Equation (4).

$$f(x) = \omega \cdot \varphi(\mathrm{x}) + \mathrm{b} \tag{4}$$

where $\cdot\varphi$ is a nonlinear function that maps x into a feature space, and b and ω are weight vectors and coefficients that have to be determined from the data. Regression coefficients are estimated in the high-dimensional feature space by minimising the sum of the empirical risk and the complexity term (Equation (5)).

$$\mathrm{R} = \mathrm{C} \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{L}_{\varepsilon}(\mathrm{f}(\mathrm{x_i}), \mathrm{y_i}) \frac{1}{2} ||\omega||_2, \tag{5}$$

where C = the additional capacity control parameter which determines the trade-off between model complexity and the extent to which errors larger than ε can be tolerated, $||\omega||$ = the regularisation term which denotes the Euclidean norm, and L_{ε} = ε-insensitive loss function (Equation (6)) that measures the empirical risk and has the advantage of selecting a subset of the input data for describing the regression vector ω. As shown in Equation (6), $\mathrm{L}_{\varepsilon} = 0$ if the difference between f(x) and y is less than ε [50].

$$\mathrm{L}_{\varepsilon}(\mathrm{f}(\mathrm{x_i}), \mathrm{y_i}) = \begin{cases} 0 & \text{for } |\mathrm{f}(\mathrm{x_i}) - \mathrm{y_i}| < \varepsilon \\ |\mathrm{f}(\mathrm{x}) - \mathrm{y}| - \varepsilon & \text{otherwise} \end{cases} \tag{6}$$

This implies that a nonlinear SVM regression function can be expressed as a function that minimises Equation (5) subject to Equation (6), as shown in Equation (7).

$$f(x, \alpha, \alpha^*) = \sum_{i=1}^{n} (\alpha_i - \alpha^*) k(x_i, x) + b \quad (7)$$

With $\alpha_i, \alpha_i{}^* \geq 0, i = 1, 2, 3, ..n$ and kernel function $k(x_i, x)$ describing the inner product in the D-dimensional space as shown in Equation (8).

$$k(x, y) = \sum_{i=1}^{D} \varphi_j(x) \varphi_i(y) \quad (8)$$

The coefficients $\alpha_i, \alpha_i{}^*$ are obtained by maximising Equation (9) subject to $\sum_{i=1}^{n} (\alpha_i - \alpha^*) = 0,\ 0 \leq \alpha_i, \alpha_i{}^* \leq C$.

$$R(\alpha, \alpha^*) = -\varepsilon um_{i=1}^{n} (\alpha_i^* + \alpha_i) + \sum y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i^* + \alpha_i)\ x\ (\alpha_i^* + \alpha_i) k(x_i, x) \quad (9)$$

Kernel functions are the main hyperparameters of SVM [51], and therefore in this work, Gaussian, linear, and polynomial kernel functions, as shown in Equations (10)–(12), were investigated in a preliminary study (Table 3) to ascertain the optimum kernel function using the training data set.

Table 3. Results of the various SVM models developed based on different kernel functions.

Kernel Function	Training MSE
Gaussian	0.83
Linear	0.92
Polynomial	0.87

(i) Gaussian kernel function

$$k(x, y) = \exp\left(-\frac{||x - y||^2}{2\alpha^2}\right) \quad (10)$$

(ii) Linear kernel function

$$k(x, y) = x, y \quad (11)$$

(iii) Polynomial kernel function

$$k(x, y) = (1 + x.y)^d \quad (12)$$

where $||x - y||$ = the Euclidean distance between x and y, d = degree of kernel, and α = free parameter.

As shown in Table 3, training MSE values in the range of 0.83–0.92 were recorded for the various SVM kernel functions that were investigated. From Table 3, it can be seen that the SVM model developed with the Gaussian kernel function (hereinafter referred to as SVM-Gaussian) produced the minimum MSE value and as such was selected as the optimal SVM model.

2.2.2. Gaussian Process Regression Algorithm

Gaussian process (GP) is a stochastic process defined by a finite collection of random variables with a joint Gaussian distribution [52]. A GP $t(x)$ is specified by its mean function $m(x)$ and kernel function $k(x, x')$ as shown in Equation (13) and Equation (14), respectively.

$$m(x) = E(t(x)) \quad (13)$$

$$k(x, x'; \theta) = E\left[(t(x) - m(x))(t(x') - m(x'))\right] \quad (14)$$

where θ is a set of hyperparameters, and $x, x' \epsilon$ X are random variables. $t(x)$ is shown in Equation (15).

$$t(x) \sim GP\left(m(x), k\left(x, x'\right)\right) \tag{15}$$

Similar to the goal of every regression problem, each output variable y is considered to be related to an underlying arbitrary function $t(x)$ that comes with an additive independent identically distributed Gaussian noise from the data, as expressed in Equation (16).

$$y = t(x) + \varrho \tag{16}$$

ϱ is an additive Gaussian noise with zero mean and variance σ_n^2, i.e., $\varrho \sim N\left(0, \sigma_n^2 I\right)$ with I as an identity matrix. Once a posterior distribution is attained, the predictive values from test data can be assessed. A joint Gaussian prior distribution can be established for a new test input x^* from a training output y and test output y^* as shown in Equation (17).

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(X,X) + \sigma_n^2 I & k(X,x^*) \\ k(X,x^*) & k(x^*,x^*) \end{bmatrix}\right) \tag{17}$$

where $k(X, X)$ is n-order symmetric positive definite kernel matrix. $k(X, x^*)$ is the kernel matrix of test input x^* and training input X. GP can be used to estimate the test output y^* according to the posterior probability formula (Equation (18)) under the conditions of a given input x^* and a training data set T.

$$y^* | x^*, \ T \sim N\left(\overline{\hat{y}^*}, var(\hat{y}^*)\right) \tag{18}$$

where $\overline{\hat{y}^*}$ and $var(\hat{y}^*)$ are the predictive mean and predictive variance, respectively, as defined in Equations (19) and (20).

$$\overline{\hat{y}^*} = k(x^*, X)(k(X,X) + \sigma_n^2 I)^{-1} y = \sum_i^n \alpha_i k\left(x^i, x^*\right) \tag{19}$$

$$var(\hat{y}^*) = k(x^*, x^*)[k(X,X) + \sigma_n^2 I]^{-1} k(X, x^*) \tag{20}$$

The kernel function forms a critical component of the GP predictor as it helps to encode the assumption of the function to be learned. Whereas kernel functions can be specified by users, hyperparameters are learned from the training data using a gradient-based optimisation approach such as the maximisation of the marginal likelihood (Equation (21)) of the observed data with respect to the hyperparameters.

$$\log p\ (y|X, \theta) = -\frac{1}{2} y'(K + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log\left|K + \sigma_n^2 I\right| - \frac{n}{2} \log 2\pi \tag{21}$$

where y' = transpose of the vector y, θ = set of hyperparameters, and σ_n^2 = noise variance.

A number of kernel functions including squared exponential, rational quadratic, matern class (3/2 and 5/2), periodic, and Gaussian noise exist in the literature [52]. In this work, exponential, rational quadratic, matern 3/2, and matern 5/2 kernel functions were investigated in selecting the optimal kernel function, as shown in Table 4. Equations (22)–(25) have been used to show the mathematical expression of the various investigated kernel functions. From Table 4, MSE values in the range of 0.0001–0.0012 were recorded by the various kernel functions investigated in this work. This result indicates that, in general, all the various investigated GPR kernel functions have a good prediction capability. However, comparing the individual kernel performances, it can be observed that the GPR model developed with the matern 3/2 kernel function (hereinafter referred to as GPR-matern 3/2) had the best performance (minimum MSE value), and therefore was selected as the optimal GPR model.

Table 4. Results of the various GPR models developed based on different kernel functions.

Kernel Function	Training MSE
Exponential	0.0008
Rational quadratic	0.0012
Matern 3/2	0.0001
Matern 5/2	0.0010

(i) Exponential kernel function

$$k(x_i, x_j) = \sigma_f^2 \exp\left[-\frac{d}{l}\right] \tag{22}$$

(ii) Rational quadratic kernel function

$$k(x_i, x_j) = \sigma_f^2 \exp\left[1 + \frac{d^2}{2\alpha l^2}\right]^{-\alpha} \tag{23}$$

(iii) Matern 3/2 kernel function

$$k(x_i, x_j) = \sigma_f^2 \left(1 + \frac{\sqrt{3}d}{l}\right) \exp\left(-\frac{\sqrt{3}d}{l}\right) \tag{24}$$

(iv) Matern 5/2 kernel function

$$k(x_i, x_j) = \sigma_f^2 \left(1 + \frac{\sqrt{5}\,d}{l} + \frac{5d^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}d}{l}\right) \tag{25}$$

where the Euclidean distance between x_i and x_j, $d = ||x_i - x_j||$, σ_f^2 = the signal variance of the function, α = the shape parameter for the rational quadratic kernel function, and l = length scale.

The main hyperparameters that were optimised in this work were σ_n^2, l, σ_f^2, and α due to the mean function, kernel function, and noisy observations in the data.

2.2.3. Artificial Neural Network Algorithm

ANN is an adaptive system that operates similarly to the human brain by using interconnected neurons in a layered structure. ANN consists of an input layer of neurons, one or several hidden layer(s) of neurons, and finally, an output layer also consisting of neurons. The neurons typically have weights that are adjusted during the learning process. This adjustment causes a change in the signal strength of a particular neuron. Each neuron is capable of receiving an input signal, processing it, and sending it as an output signal. The computation of an output signal h_p of a neuron p in the hidden layer is shown in Equation (26).

$$h_p = \delta\left(\sum_{j=1}^{M} V_{ij} x_i + T_i^{hid}\right) \tag{26}$$

where δ is an activation or transfer function, M is the total number of input layer neurons, V_{ij} denotes neuron weight, and x_i is the input to the input layer neurons with T_i^{hid} representing the threshold terms of the hidden neurons.

Studies have proven that factors such as the training algorithm, the number of hidden layers, and the number of hidden layer neurons are the critical parameters in attaining an accurate ANN structure [53,54]. Among these factors, the type of training algorithm and the number of hidden layer neurons play a key role in the final network performance, as it has been proven that one hidden layer is enough to fit any continuous data [55]. Based on

this, one hidden layer was considered for this work. In order to determine the optimum training algorithm and the optimum number of hidden layer neurons, different training algorithms including Levenberg–Marquardt, Bayesian regularisation, gradient descent, scaled conjugate gradient, gradient descent with momentum, one-step secant backpropagation, and gradient descent with adaptive learning rate were tried in a preliminary study using a maximum of 31 hidden layer neurons, as shown in Figure 3. The maximum number of hidden layer neurons used in the preliminary study was based on a recommendation of Hecht-Nielsen [56] where he proposed a maximum number of $2(x_{iT}) + 1$ neurons with x_{iT} representing the total number of input variables (15). From Figure 3, the Bayesian regularisation training algorithm with 30 hidden layer neurons yielded the minimum MSE value; hence, this combination was chosen as the optimum training algorithm and optimum number of hidden layer neurons in this work. The input and output layer neurons used in this work were 15 and 1, respectively, as the number of input and output variables become the default number of neurons in ANN. This implies that the optimum ANN model used in this work had the structure 15-30-1.

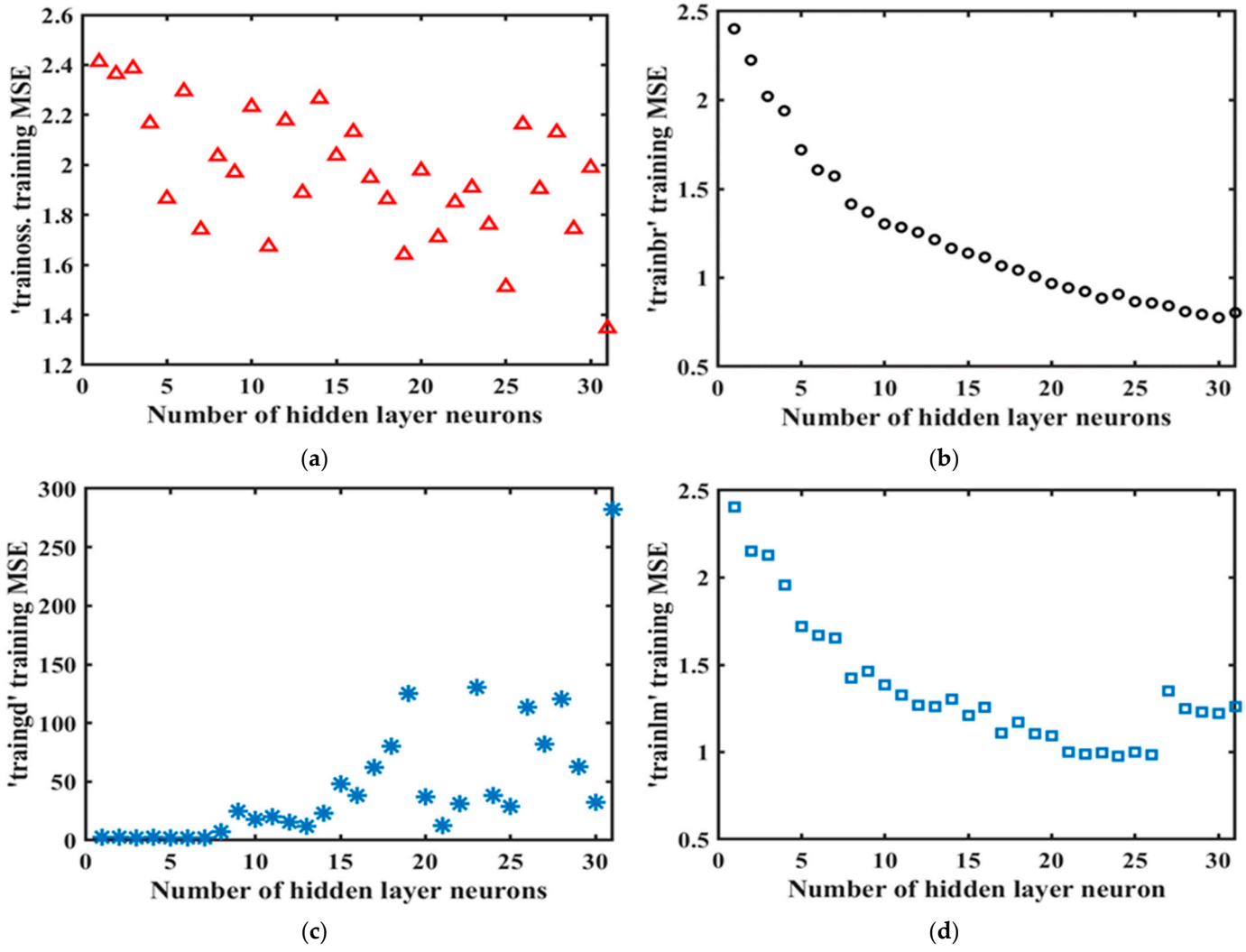


Figure 3. *Cont.*

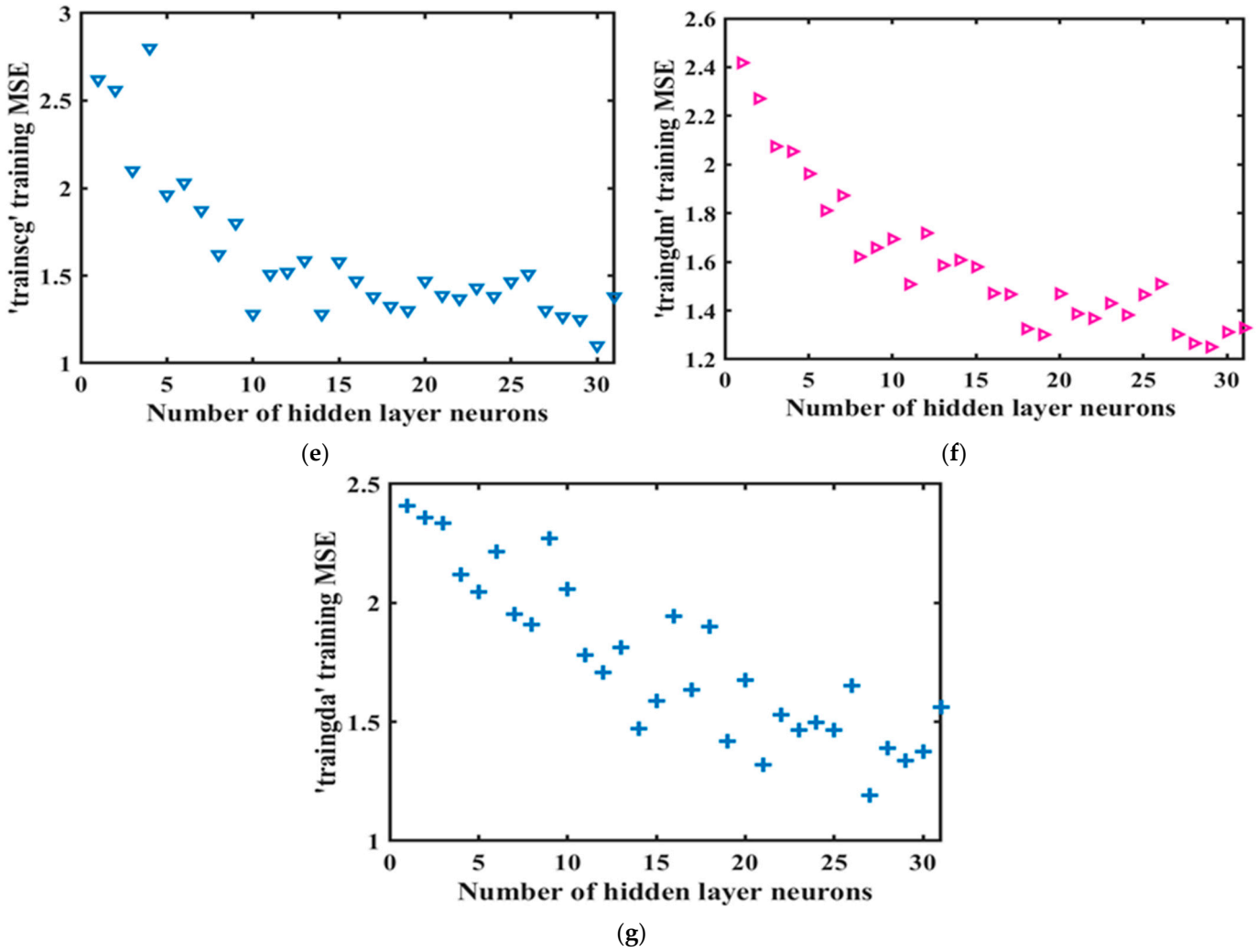


Figure 3. Training error results and number of hidden layer neurons using (**a**) one-step secant back-propagation, (**b**) Bayesian regularisation, (**c**) gradient descent, (**d**) Levenberg–Marquardt, (**e**) scaled conjugate gradient, (**f**) gradient descent with momentum, and (**g**) gradient descent with adaptive learning rate.

2.2.4. Linear Regression Algorithm

An LR model establishes the relationship between one output variable y_i and one or more input variables x_i. An LR model is often referred to as multiple linear regression (MLR) model in a situation where there is more than one input variable. The formulation of the MLR model used in this work is shown in Equation (27).

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_p x_{ip} + \varrho_i,\ i = 1, 2, 3, .., n \quad (27)$$

where $y_i = i$th output value, $\beta_p = p$th coefficient, $\beta_0 =$ constant term in the model, $x_{ij} = i$th observation of the jth variable, $j = 1, \ldots, p$, $\varrho_i = i$th noise term, that is, random error, and $n =$ total number of observations.

LR models are formulated with the following assumptions.

(1) ϱ_i values are not correlated.
(2) ϱ_i values have an independent and identical normal distribution with zero mean and constant variance, var. Thus,

$$E(y_i) = E\left(\sum_{k=0}^{K} \beta_k f_k(x_{i1}, x_{i2}, \ldots, x_{ip}) + \varrho_i\right)$$

$$= \sum_{k=0}^{K} \beta_k f_k(x_{i1}, x_{i2}, \ldots, x_{ip}) + E(\varrho_i)$$

$$= \sum_{k=0}^{K} \beta_k f_k(x_{i1}, x_{i2}, \ldots, x_{ip})$$

And

$$V(y_i) = V\left(\sum_{k=0}^{K} \beta_k f_k(x_{i1}, x_{i2}, \ldots, x_{ip}) + \varrho_i\right) = V(\varrho_i) = var$$

So, the variance of y_i is the same for all levels of x_{ij}.

(3) The true output values y_i are uncorrelated.

Based on the assumptions above, the fitted linear function can be expressed as shown in Equation (28).

$$\hat{y}_i = \sum_{k=0}^{K} b_k f_k(x_{i1,} x_{i2,} \ldots, x_{ip}) \; i = 1, 2, 3 \ldots, n \tag{28}$$

where $\hat{y}_i = i$th predicted output value, $b_k =$ fitted coefficients, and $x_{ip} = i$th observation of the pth variable, and $n =$ total number of observations.

The coefficients are estimated to minimise the mean squared error between the predicted output values $\hat{y}_i$ and true output values y_i i.e., $\hat{y}_i - y_i$ [57]. The addition of LR to this study allows the comparison of its relative performance compared with complex models.

2.2.5. Random Forest Algorithm

An RF algorithm is an ensemble method which constructs a set of tree predictors and uses averaging to make a final decision [58]. In RF, variables or a combination of variables are selected at each node to grow a predictor tree. To build a predictor tree, bagging (bootstrap sampling), a method which randomly generates a training data set from the original training data set with replacement, is used [59]. In bagging, the training data set consists of about two-thirds of the original training data set, leaving out about one third of the original training data set for each tree predictor grown. Each tree predictor $T_{L(\theta)}$ is dependent on a random vector θ which indicates the bagged samples from the original training data set L. The final predictor f is the average of all trees, as shown in Equation (29) [60].

$$f(x_n) = \frac{1}{k} \sum_{k=1}^{k} T_{L(\theta_k)}(x_n)_1^k \tag{29}$$

where x_n is the nth sample, and k is the total number of trees grown. The number of variables used at each node to grow a predictor tree and the maximum number of trees to be grown are user-defined parameters and the most important hyperparameters in random forest [61–63]. For this work, 5 variables were randomly selected to build predictor trees at each node following a $p/3$ (rounded down) recommendation by Hastie et al. [64], where p is the total number of variables (15). To obtain the optimal number of trees, a different number of trees were used in a preliminary study while monitoring the training error as shown in Figure 4. From Figure 4, it can be seen that the performance of the algorithm increases from 50 to 300 trees after which no significant improvement occurs. Therefore, the optimal RF model used in this work had 300 predictor trees, with each tree built from 5 randomly selected rougher flotation variables.

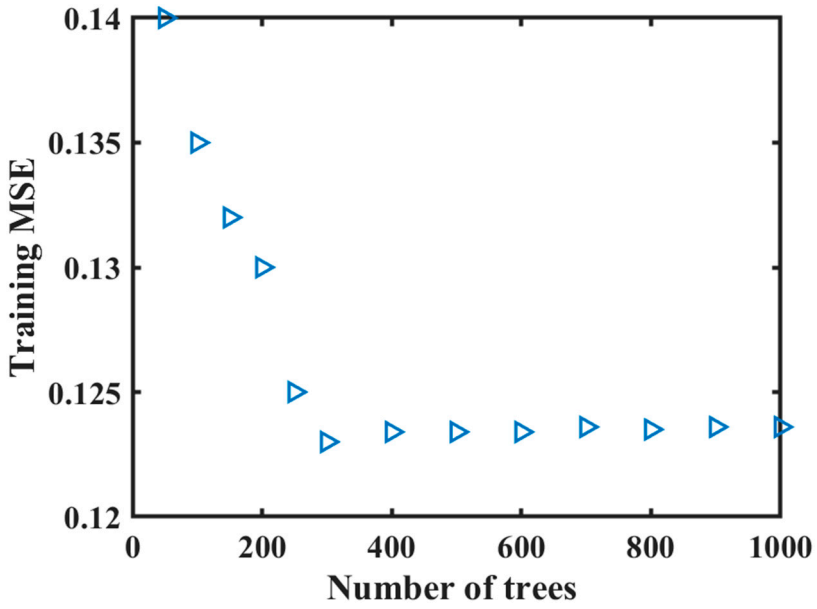


Figure 4. A visualisation of RF training errors and number of trees.

The variable selection method and pruning method are also key factors in designing a good predictor tree. From the literature, the most commonly used variable selection methods are the information gain ratio criterion and Gini index [65,66]. For this work, the Gini index which measures the impurity of a variable with respect to the output was used. With regards to pruning, Breiman [67] suggests that as the number of predictor trees increases, the generalisation error tends to converge even without pruning because of the strong law of large numbers [68].

2.3. Predictive Model Performance Assessment

To evaluate and compare the overall performance of the different predictive models, four performance assessment indicators were used. As shown in Equations (30)–(33), the correlation coefficient (r), root mean square error (RMSE), mean absolute percentage error (MAPE), and variance accounted for (VAF) were utilised in this work. For a good-performing model, r should be approaching 1, RMSE and MAPE should be approaching zero, with VAF close to 100% as possible. All model performance assessment indicators were computed at the 95% confidence interval.

$$r = \frac{\sum_{i=1}^{n}(y_i - \overline{y}) - (\hat{y}_i - \overline{\hat{y}})}{\sum_{i=1}^{n}(y_i - \overline{y})^2 \, \mathrm{X} \, \sqrt{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})^2}} \tag{30}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\left(\frac{1}{n}\right)\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{31}$$

$$\mathrm{MAPE} = \left(\frac{1}{n}\right)\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{32}$$

$$\mathrm{VAF} = \left(1 - \frac{var(y_i - \hat{y}_i)}{var(y_i)}\right) \times 100\% \tag{33}$$

where y_i = ith true rougher copper-recovery value, $\overline{y}$ = mean of true rougher copper-recovery values, $\hat{y}_i$ = ith predicted rougher copper-recovery value, $\overline{\hat{y}}$ = mean of predicted rougher copper-recovery values, and n = total number of observations.

2.4. Flotation Variable Optimisation

Optimisation of the various flotation variables was carried out to ascertain their optimal operation values which maximise copper recovery at the BHP Olympic Dam. As highlighted earlier, four optimisation algorithms (SA, PSO, SO, and GA) were applied using the best-performing predictive function as presented in the above sections. The formulation of the optimisation model for copper recovery is expressed in Equation (34). The boundary constraint for each rougher flotation variable in Equation (34) was ascertained in consultation with the metallurgical team at the BHP Olympic Dam and at least a 93% recovery was the proposed expected copper recovery. It should be further noted that rougher flotation variables listed in Equation (34) are the same variables used as the input variables for the development of the predictive model.

$$\{Minimise(-Copper\ recovery) \tag{34}$$

$s.t$
1.6 w.t% $\leq$ *Feed grade* $\leq$ 2.6 w.t%
78.0% $\leq$ *Feed particle size* $\leq$ 84.0%
350.0 t/h $\leq$ *Throughput* $\leq$ 900.0 t/h
37.0 mL/min $\leq$ *Xanthate to tank cell* 1 $\leq$ 190.5 mL/min
27.8 mL/min $\leq$ *Xanthate to tank cell* 4 $\leq$ 142.9 mL/min
28.1 mL/min $\leq$ *Frother to tank cell* 1 $\leq$ 180.7 mL/min
28.1 mL/min $\leq$ *Frother to tank cell* 4 $\leq$ 180.7 mL/min
900.0 m^3/h $\leq$ *Airflow to tank cell* 1 $\leq$ 1250.0 m^3/h
1100.0 m^3/h $\leq$ *Airflow to tank cell* 2 $\leq$ 1250.0 m^3/h
1100.0 m^3/h $\leq$ *Airflow to tank cell* 3 $\leq$ 1250.0 m^3/h
1100.0 m^3/h $\leq$ *Airflow to tank cell* 4 $\leq$ 1250.0 m^3/h
1100.0 m^3/h $\leq$ *Airflow to tank cell* 5 $\leq$ 1250.0 m^3/h
75.0 mm $\leq$ *Froth depth of tank cell* 1 $\leq$ 150.0 mm
75.0 mm $\leq$ *Froth depth of tank cell* 2/3 $\leq$ 150.0 mm
75.0 mm $\leq$ *Froth depth of tank cell* 4/5 $\leq$ 150.0 mm

2.4.1. Simulated Annealing Optimisation Algorithm

SA is a method for finding the solution to unconstrained and bound constrained optimisation problems. The algorithm mimics the physical process of heating a material and slowly decreasing its temperature to minimise system energy. The algorithm operates by first of all generating a random trial point. The distance of a new point from a current point or the extent of the search is chosen based on a probability distribution with a scale proportional to the current temperature. After this, the algorithm shifts the trial point if necessary, so it stays within the specified bounds. Each infeasible component of the trial point is shifted to a value randomly chosen between the violated bound and the feasible value of the previous iteration. The new point is compared to the current point, and if the new point is better than the current point, it becomes the next point. If the new point is worse than the current point, the algorithm can still accept it as the next point. The probability of acceptance $P(\text{accept new})$ is expressed in Equation (35).

$$P(\text{accept new}) = \frac{1}{1 + \exp\left(\frac{\Delta}{\max(T)}\right)} \tag{35}$$

where Δ = the difference between the new and old objective function to be minimised, and T = current temperature.

Since both Δ and T are positive, the probability of acceptance always falls between 0 and 0.5. A smaller T and larger Δ lead to a smaller acceptance probability and vice versa.

From here, T is systematically lowered, and the best found points are stored. T is updated using the relation expressed in Equation (36).

$$T = T_0\left(0.95^k\right) \tag{36}$$

where k = the annealing parameter or iteration number until reannealing, and T_0 = initial temperature.

The algorithm reanneals the annealing parameters to values lower than the iteration number causing an increase in temperature in each dimension. Annealing parameters are dependent on values of the estimated gradients of the objective function in each dimension, as expressed in Equation (37) [69,70].

$$k_i = \log\left(\begin{array}{c} \max(s_j) \\ \frac{T_0}{T_i}\frac{j}{s_i} \end{array}\right) \tag{37}$$

where k_i = annealing parameter of component i, T_0 = initial temperature of component i, T_i = current temperature of component i, s_i = gradient of objective function in direction i times the difference of bounds in direction i, and s_j = gradient of objective function in direction j times the difference of bounds in direction j.

The key parameter affecting the performance of the algorithm is the cooling schedule. Sufficiently high initial temperatures in the initial phase gives the algorithm the flexibility to search through the entire search space for a better solution and vice versa. The rate and the way of lowering the temperature also determine the speed of the algorithm such that decreasing the temperature slowly hinders the identification of the optimum solution by leaving too much freedom to search during a large number of iterations. The algorithm finally stops the search for the optimum solution when the average change in the objective function value is less than the function tolerance value. For this work, an initial temperature and a function tolerance value of 110 k and 0.000001, respectively, were specified. Figure 5 is a flowchart outlining the major stages of the algorithm.

2.4.2. Particle Swarm Optimisation Algorithm

The inspiration for this algorithm is a flock of birds or insects swarming such that each bird or bee is attracted to a best location it has found and also to the best location any member of the swarm has found. The algorithm operates by first of all generating initial particles and assigning initial velocities to them. The objective function is then evaluated at each particle location to identify the best function value and the best location. New velocities are chosen based on the current velocity, particles' individual best locations, and the best locations of their neighbours. Particles locations, velocities, and neighbours are iteratively updated until the relative change in the best objective function value is less than the function tolerance value [71,72]. The positions of particles are updated as shown in Equation (38).

$$x_{i(t)} = x_{i(t-1)} + v_i(t) \tag{38}$$

where $x_{i(t)}$ = current position of particle, $x_{i(t-1)}$ = previous position of particle, $v_i(t)$ = velocity vector, and $v_i(t)$ reflects the exchanged information and can generally be defined as shown in Equation (39).

$$v_i(t) = Wv_i(t-1) + C_1r_1\left(x_{pbesti} - x_{i(t)}\right) + C_2r_2\left(x_{leader} - x_{i(t)}\right) \tag{39}$$

where C_1 and C_2 are learning factors and are defined as constants, x_{pbesti} is the neighbourhood best position found, and x_{leader} is the position of the swarm leader. r_1 and $r_2\ \epsilon[0,\ 1]$ are randomly generated values, and W is the inertial weight defined within the algorithm [73,74].

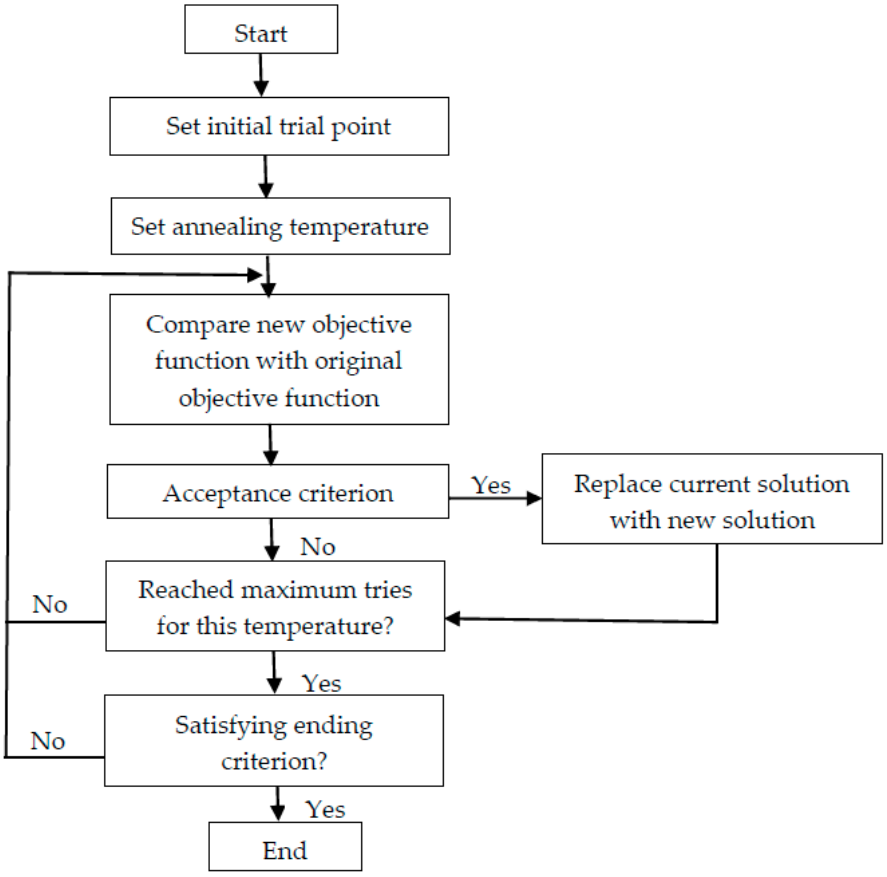


Figure 5. Flowchart outlining the major stages of the SA algorithm.

For this work, the initial particles were generated at random and uniformly distributed within the specified constraint bound of each rougher flotation variable. Their initial velocities were also generated at random and uniformly distributed in the range of [−r, r], where r is the vector of initial ranges (difference between specified upper and lower bound of each rougher flotation variable). The function tolerance value was 0.000001. A flowchart outlining the various stages of the PSO algorithm is shown in Figure 6.

2.4.3. Surrogate Optimisation Algorithm

An SO algorithm approximates another function to search for a solution. An SO algorithm can be used to search for a point that minimises an objective function by evaluating it on thousands of points and taking the best value as an approximation to the minimiser of the objective function. To carry out SO, random sample points (using a quasirandom sequence) are first generated within the specified constrained bounds, and the expensive objective function is evaluated at these points. A surrogate of the expensive objective function is then created by interpolating a radial basis function γ through these points, as shown in Equation (40).

$$\gamma(x) = \gamma(\| x \|) \tag{40}$$

where $x =$ the input value.

A minimum value of the objective function is searched for by sampling several thousands of points from the randomly generated points. From here, a merit function $f_{merit}(x)$ (Equation (41)) is evaluated based on both the surrogate value at these points and the distance between them and the points where the expensive objective function has been evaluated. A best point is chosen from the evaluation as a candidate point, as measured by the merit function.

$$f_{merit}(x) = wS(x) + (1 - w)D(x), \text{ for a weight } w \text{ with } 0 < w < 1 \tag{41}$$

where $S(x)$ = scaled surrogate, and $D(x)$ = scaled distance.

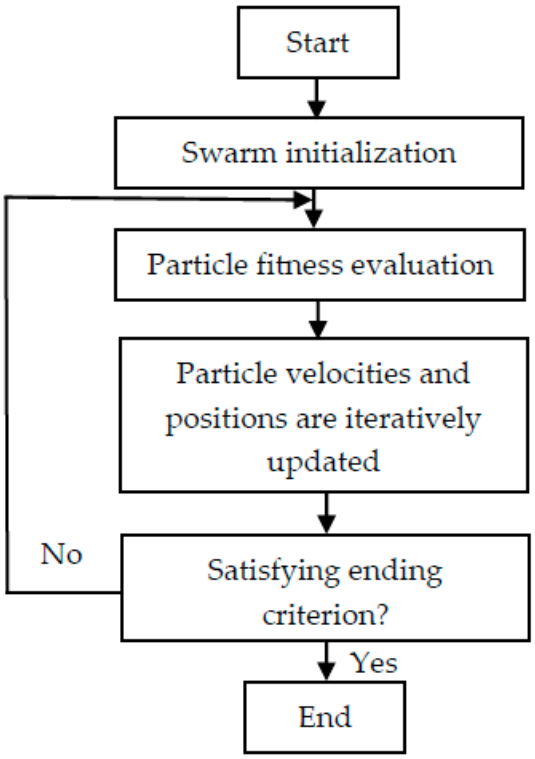


Figure 6. Flowchart outlining the major stages of the PSO algorithm.

The candidate point with the minimum merit function value (adaptive point) is used to evaluate the objective function value, simultaneously updating the surrogate value. If the objective function value at the adaptive point is lower than the current value, the search for the solution is considered complete, and the adaptive point value is set as the current value of the function value. If the otherwise occurs, the algorithm deems the search unsuccessful and does not change the current function value [75–77]. Figure 7 is a flowchart showing the major stages of the SO algorithm.

2.4.4. Genetic Algorithm

A genetic algorithm (GA) solves optimisation problems based on a natural selection process that mimics biological evolution [78,79]. The algorithm operates by creating a random initial population. A sequence of new populations is then created, and at each step, the algorithm creates the next population using individuals in the current generation. A new population is created by scoring each member of the current population through the computation of their individual fitness value (raw fitness scores). The raw fitness values are then scaled to convert them into a more usable range of values, also known as expectation values. The parents for the next generation are selected based on their expectation values. Some individual members in the current population with lower fitness values are considered to be elites and are passed on to the next generation. Children for the next generation are produced from parents by either combining the vector entries of a pair of parents (crossover) or by making changes (mutation) to a single parent. The current population is then replaced with the children to form the next generation. The algorithm is stopped when the relative change in the fitness value is less than the function tolerance value. The function tolerance value was specified to be 0.000001 in this work. Figure 8 is a flowchart showing the major stages of the algorithm.

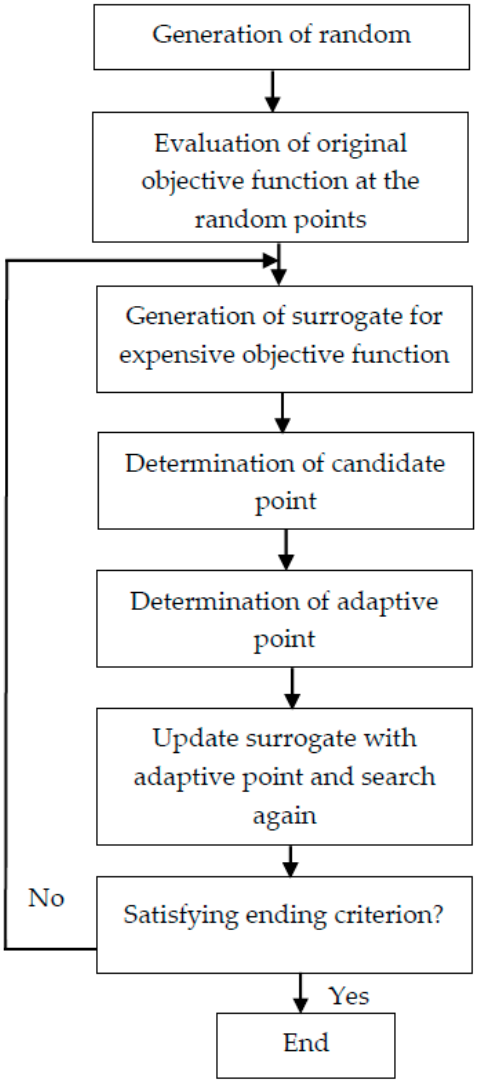


Figure 7. Flowchart outlining the major stages of the SO algorithm.

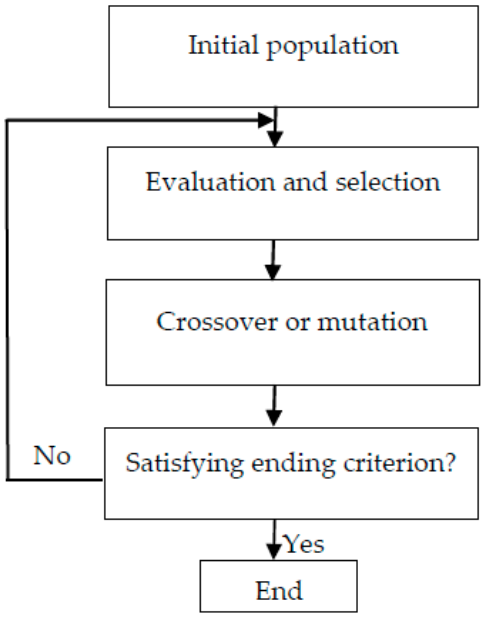


Figure 8. Flowchart outlining the major stages of the GA algorithm.

3. Results and Discussion

This section presents the results of the various techniques applied in this work. Specifically, a detailed discussion of the performance of the various predictive models has been presented together with the optimisation outcomes.

3.1. Predictive Model Performance

To assess the performance of the various predictive models, *r*, RMSE, MAPE, and VAF have been used as performance indicators with the results visualised in Figure 9. From Figure 9, an *r* indicator which gives an idea of the linear relationship between true and predicted rougher copper-recovery values was determined to be 0.87, 0.99, 0.87, 0.53, and 0.97 for SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively, when the trained models were fitted with the training data set. When the trained models were fitted with the validation data set, SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF recorded *r* values of 0.85, 0.97, 0.86, 0.50, and 0.95, respectively. With regard to fitting the trained models with the testing data set, 0.86, 0.97, 0.86, 0.51, and 0.95 were the *r* values recorded by SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively. Despite the very large testing data, compared with the training and validation data sets, the observed model performance values for the testing were close to the validation outcome. The highest *r* values obtained by the GPR-matern 3/2 model in each instance show quantitatively the strength of the linear relationship between true and predicted rougher copper-recovery values when the GPR-matern 3/2 model was used. This result further indicates the uniqueness of the predictive strength of the GPR-matern 3/2 model.

In terms of error statistics, RMSE and MAPE indicators were used in this work. RMSE values of 0.91, 0.01, 0.88, 1.52, and 0.35 were recorded by SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively, when the trained models were fitted with the training data set. When the trained models were fitted with the validation data set, SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF recorded RMSE values of 0.96, 0.41, 0.93, 1.56, and 0.60, respectively. Furthermore, 0.93, 0.41, 0.92, 1.54, and 0.59 were the RMSE values that were recorded by SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively, when the trained models were fitted with the testing data set. The lowest RMSE values obtained by the GPR-matern 3/2 model in each instance is a clear indication that GPR-matern 3/2 predicted rougher copper-recovery values are in better agreement with true rougher recovery values. From the MAPE results shown in Figure 5, it can be stated that the unexplained variability in true and predicted rougher copper-recovery values was 0.65%, 0.01%, 0.71%, 1.32%, and 0.23% for SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively, when fitted with the training data set. Fitting the trained models with the validation data set resulted in MAPE values of 0.69%, 0.24%, 0.74%, 1.34%, and 0.40% for SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF, respectively. Their corresponding testing data set fitting had MAPE values of 0.68%, 0.24%, 0.74%, 1.32%, and 0.39%. Once again, it is obvious that the GPR-matern 3/2 model was the best performing model as far as the MAPE indicator is concerned.

A VAF indicator was also used to verify the correctness of the models and how well they could make predictions. SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF trained models recorded VAF values of 74.72%, 99.99%, 76.24%, 26.54%, and 96.14%, respectively, when fitted with the training data set. Fitting with the validation data set resulted in VAF values of 71.72%, 94.70%, 73.50%, 24.50%, and 88.89% for SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF trained models, respectively. For fitting done with the testing data set, SVM-Gaussian, GPR-matern 3/2, ANN, LR, and RF trained models recorded VAF values of 74.87%, 94.65%, 73.32%, 25.62%, and 89.47%, respectively. These results indicate that the GPR-matern 3/2 model which outperformed SVM-Gaussian, ANN, LR, and RF can explain about 99.99%, 94.70%, and 94.65% of the potential variance in the predicted rougher copper-recovery values from the training, validation, and testing data sets, respectively.

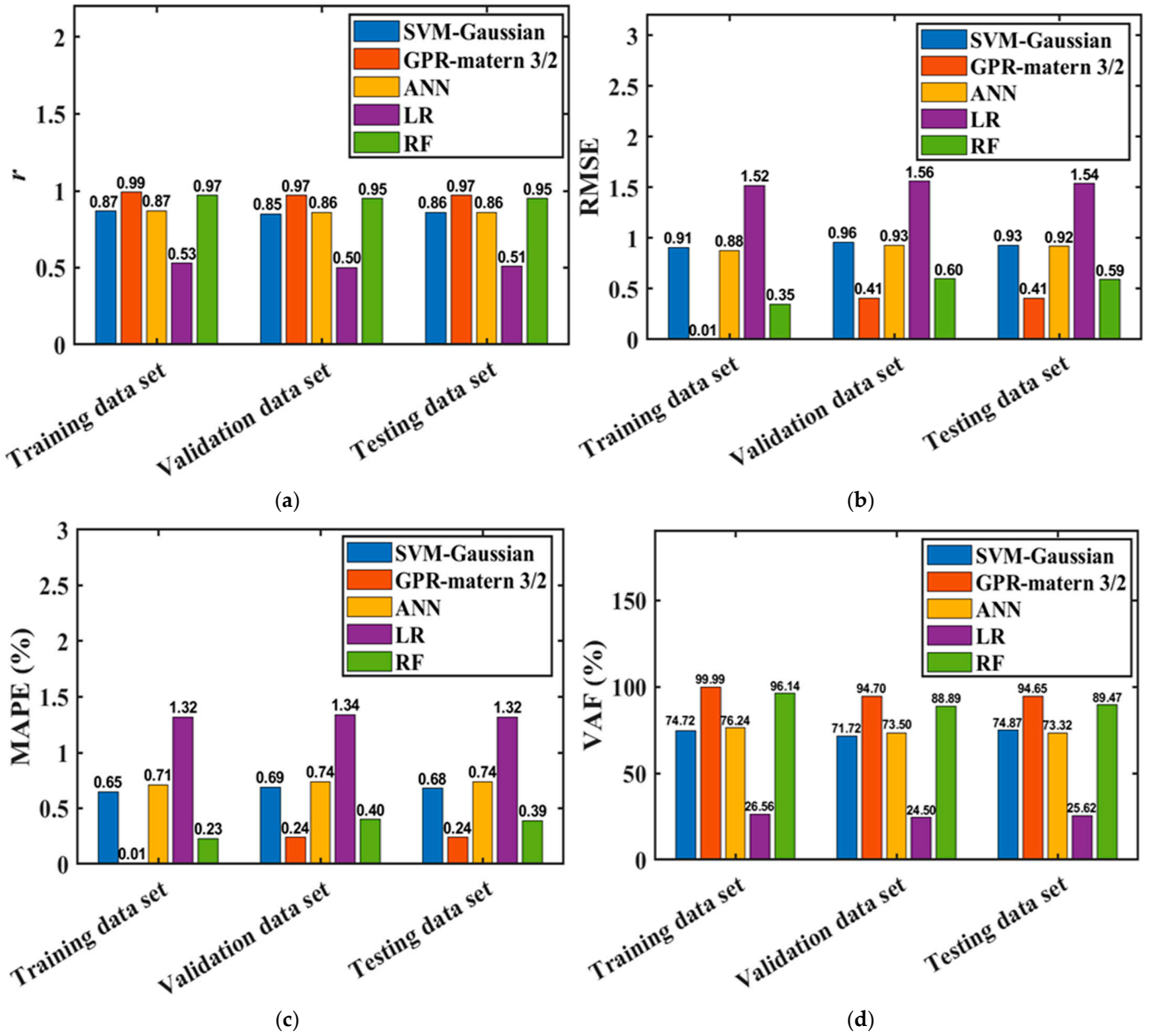


Figure 9. Visualisation of model performance using an (**a**) *r* indicator, (**b**) RMSE indicator, (**c**) MAPE indicator, and (**d**) VAF indicator.

It can clearly be seen that the GPR-matern 3/2 model produced the most precise copper-recovery prediction as compared to SVM-Gaussian, ANN, LR, and RF. The outstanding performance of the GPR-matern 3/2 model could be attributed to its intrinsic ability to add a prior knowledge and specification about the shape of the model by learning the hyperparameters which are relational to the training, validation, and testing data sets. This helps to capture the uncertainties in the data using the noise variance hyperparameter during the model formulation stage. The next best model from Figure 9 is the RF model. The performance of the RF model could be linked to the fact that it is an ensemble learner which builds multiple predictor trees and finds the average prediction value in solving a problem. This ensemble technique in RF makes it better than most standalone predictive models. Again, from Figure 9, it is obvious that SVM-Gaussian and ANN models had a similar performance, which was below the performance of both the GPR-matern 3/2 and RF models. This is mainly because both SVM-Gaussian and ANN models had a very similar learning ability during the training phase, as well as a similar generalisation capability

during the validation and testing phases. The poor performance of the LR model could be attributed to its inability to capture the complex nonlinear relationship between rougher flotation variables and rougher copper recovery. This is because the algorithm is quite potent in capturing linear relationships, which is contrary to the data set used in this work. For brevity, parity plots visualising the distribution of true and predicted copper-recovery values for all the investigated models using the testing data set are shown in Figure 10. It can be seen that the GPR-matern 3/2 model had the minimum spread of true and predicted copper-recovery values along its linear fit, confirming its unique predictive performance over the other investigated models (SVM-Gaussian, ANN, LR, and RF). This was followed closely by the RF model with SVM-Gaussian and ANN having a similar spread below that of the GPR-matern 3/2 and RF models. With the highest performance of GPR-matern 3/2, it was extracted for the optimisation studies.

3.2. Selection of Best Optimisation Solution

Four sets of solutions were found for the rougher flotation variables, as shown in Table 5. Furthermore, visualisations of the best objective function value at each iteration or generation for the various optimisation algorithms have also been presented in Figure 11.

From Table 5, it can be seen that all the optimisation algorithms found solutions within the specified constrained bounds of the individual rougher flotation variables, indicating their correctness. It can further be observed from Table 5 that the predicted copper-recovery objective function values were in line with the expected copper recovery ($\geq$93%). However, most of the solutions found for the various flotation variables showed subtle differences, making it dicey to easily select the best set of solutions. As such, a hypothetical analysis of the optimisation results was carried out considering a 24 h period.

For this hypothetical analysis, a focus was placed on feed grade, throughput, feed particle size, and reagent dosages (xanthate and frother) for economic and eco-friendly mineral separation, as shown in Table 6. A copper price of AUD6500 per tonne, xanthate and frother cost of AUD1.20 per litre and AUD1.30 per litre, respectively, were also assumed in this analysis.

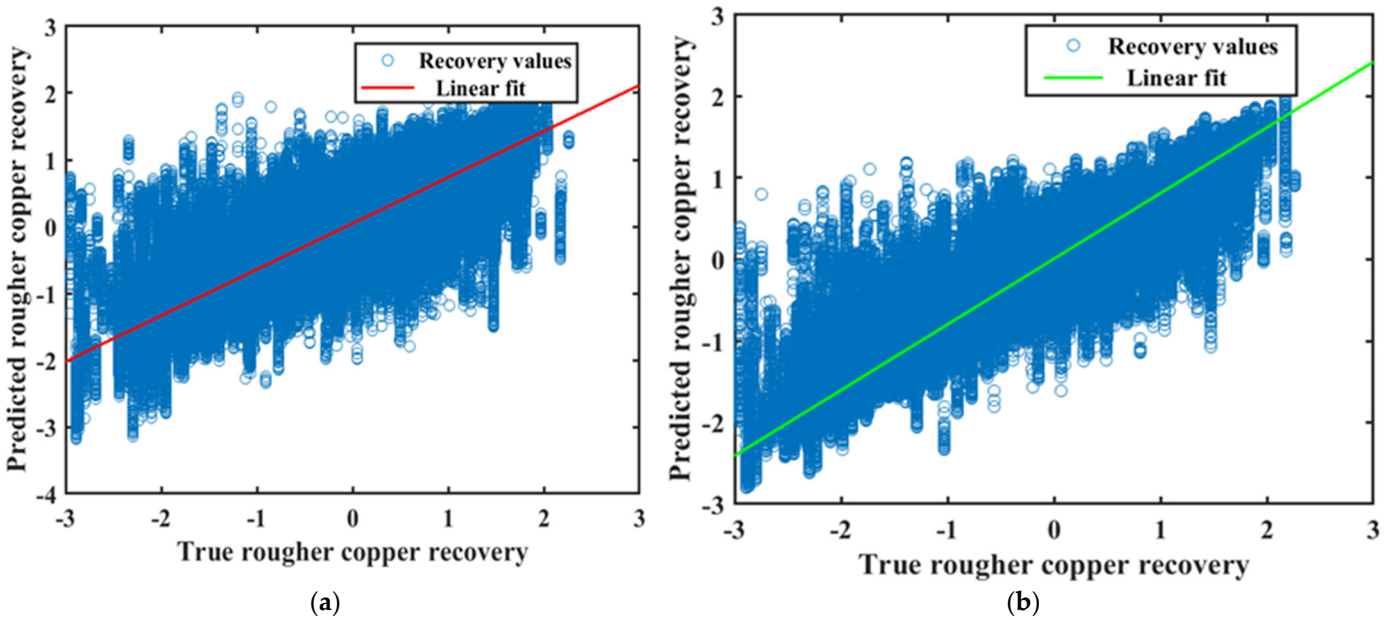


Figure 10. *Cont.*

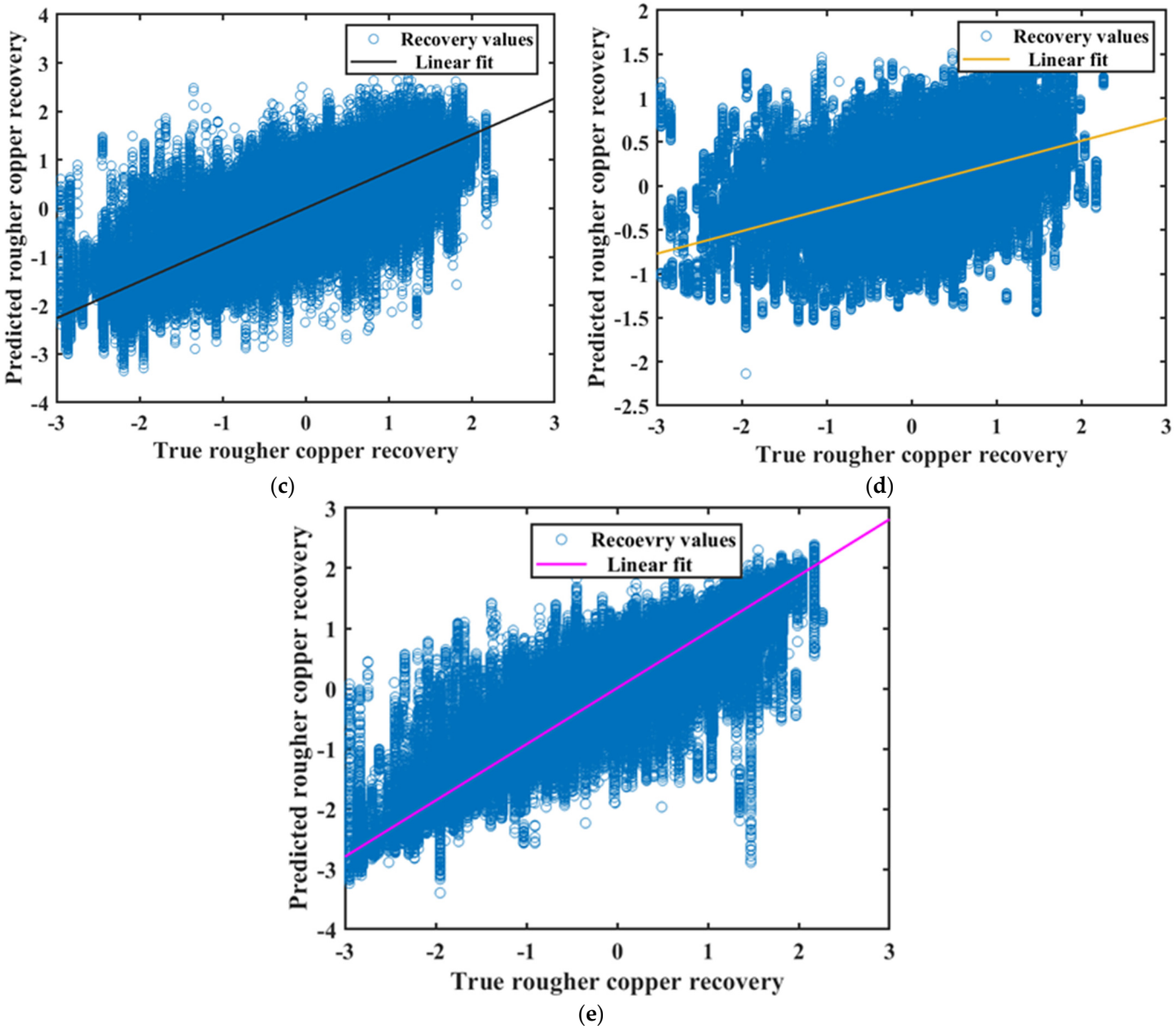


Figure 10. Parity plots visualising the distribution of true and predicted rougher copper-recovery values for the (**a**) SVM-Gaussian, (**b**) GPR-matern 3/2, (**c**) ANN, (**d**) LR, and (**e**) RF models using the testing data set.

The results showed a throughput of 15,312.48 t, 12,798.00 t, 12,819.36 t, and 12,869.76 t was recorded for the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA, respectively, in 24 h (Table 6). The high throughput value recorded by the SA optimisation algorithm is due to its relatively coarse grind size of 82.8% passing 75 µm as compared to the PSO algorithm, SO algorithm, and GA which all had a feed particle size value around 84% passing 75 µm. In as much as no data are presented on mill energy consumption in this work, it is obvious that applying the feed particle size solution found by the SA optimisation algorithm will conserve mill energy as compared to the feed particle size solutions found by the PSO algorithm, SO algorithm, and GA. Applying the respective feed grades, predicted copper recovery, and total throughput of material treated resulted in 371.46 t, 313.77 t, 314.29 t, and 319.19 t of copper for the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA, respectively, in 24 h. The monetisation of this recovered rougher copper was AUD2,414,490.00, AUD2,039,505.00, AUD2,042,885.00, and

AUD2,074,735.00, respectively, for the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA.

In terms of reagent consumption, as shown in Table 6, solutions found by the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA resulted in a total xanthate consumption of 133,387.20 mL, 103,478.40 mL, 103,507.20 mL, and 103,665.60 mL, respectively, in 24 h. These values resulted in a total xanthate cost of AUD160.06, AUD124.17, AUD124.20, and AUD124.40, respectively, for the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA. For frother consumption in the 24 h period, a total of 260,294.40 mL, 310,219.20 mL, 310,867.20 mL, and 309,571.20 mL was recorded by the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA, respectively. The cost for this frother consumption was estimated to be AUD338.38 for the SA optimisation algorithm, AUD403.28 for the PSO algorithm, AUD404.13 for the SO algorithm, and AUD402.44 for GA.

To complete the hypothetical analysis, the net gain at the end of the 24 h and total reagent cost for the same period were computed. The values realised were AUD2,413,991.56, AUD2,038,977.55, AUD2,042,885.00, and AUD2,074,735.00, respectively, for the SA optimisation algorithm, PSO algorithm, SO algorithm, and GA. These results indicate that the SA optimisation algorithm had a net gain percentage of 15.5%, 15.40%, and 14.07% over the PSO algorithm, SO algorithm, and GA, respectively.

In all the main benchmarks (overall throughput, feed particle size, xanthate and frother consumption, and net gain) used in this analysis, it is evident that the SA optimisation algorithm outperforms the other optimisation algorithms, except for xanthate consumption where the PSO algorithm, SO algorithm, and GA did marginally better than the SA optimisation algorithm, as shown in Table 6. Regardless of this, the overall net gain applying solution found by the SA optimisation algorithm is enough to compensate for the cost of xanthate consumed. Based on this, the SA optimisation algorithm was chosen to provide the best set of solutions for the maximisation of copper recovery even though it had the minimum predicted copper objective function value of 94.76% as against the PSO algorithm, SO algorithm, and GA which had objective function values >95%.

Table 5. Optimum operating values of the various rougher flotation variables.

Rougher Flotation Variable		SA Optimisation Algorithm	PSO Algorithm	SO Algorithm	GA
Feed grade (wt. %)		2.56	2.56	2.56	2.59
Feed particle size (% passing 75 μm)		82.80	84.00	84.00	83.99
Throughput (t/h)		638.02	533.25	534.14	536.24
Xanthate dosage (mL/min)	Tank cell 1	44.85	37.00	37.00	37.01
	Tank cell 4	47.78	34.86	34.88	34.98
Frother dosage (mL/min)	Tank cell 1	106.18	112.75	112.83	112.13
	Tank cell 4	74.58	102.68	103.05	102.85
Airflow rate (m^3/h)	Tank cell 1	1121.44	1231.35	1232.10	1232.14
	Tank cell 2	1182.66	1133.44	1132.42	1132.20
	Tank cell 3	1190.45	1134.58	1134.34	1134.70
	Tank cell 4	1211.91	1193.12	1192.62	1193.07
	Tank cell 5	1201.35	1250.00	1250.00	1249.95
Froth depth (mm)	Tank cell 1	148.73	149.99	150.00	149.93
	Tank cell 2/3 *	81.72	102.64	99.41	98.83
	Tank cell 4/5 *	149.53	150.00	150.00	149.95
Predicted rougher copper recovery (%)		94.76	95.77	95.77	95.76

* Froth depth of tank cells 2 and 4 also represent tank cells 3 and 5, respectively, as they are kept at same level.

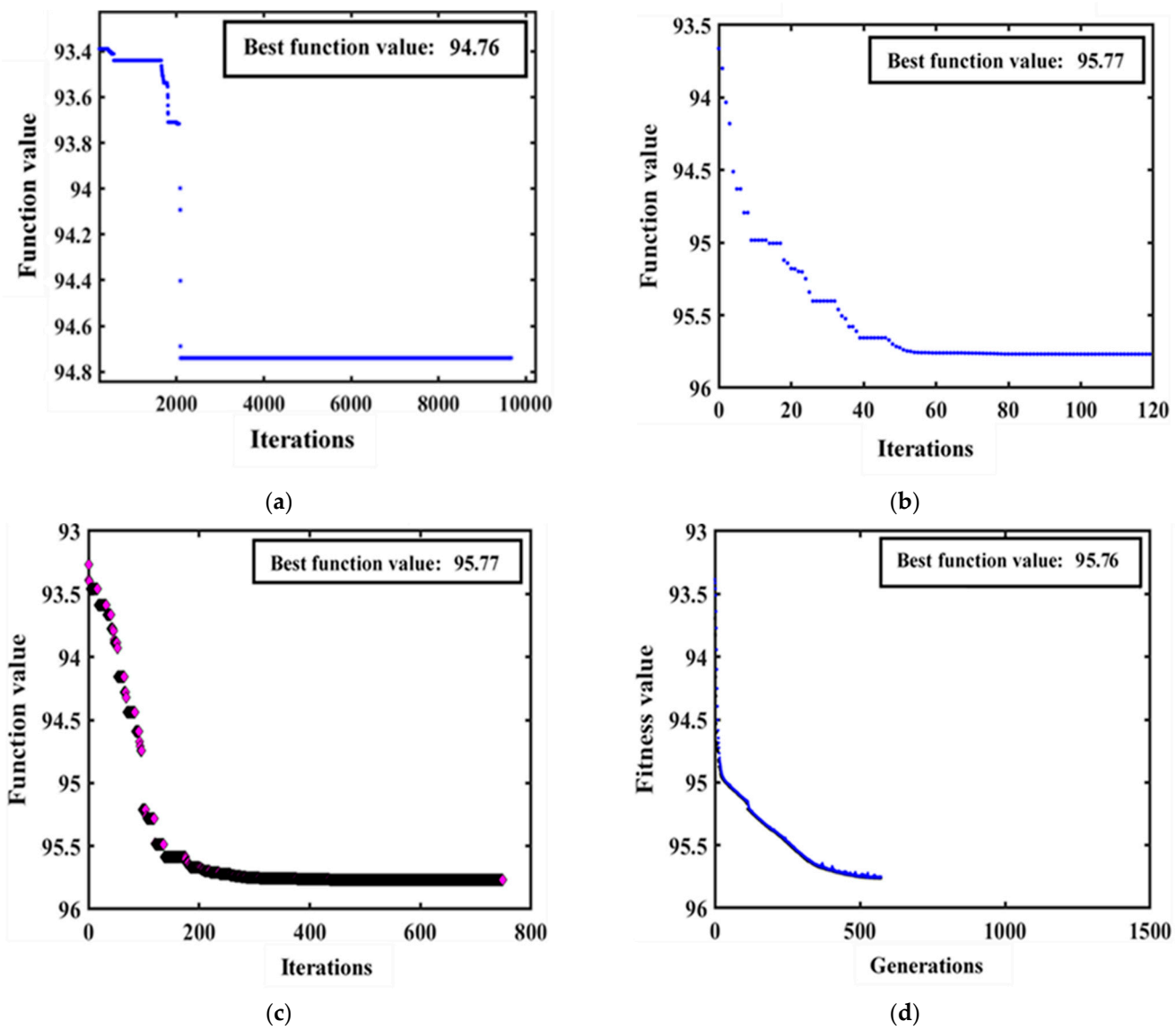


Figure 11. Visualisation of the best function values of the (**a**) SA optimisation algorithm, (**b**) PSO algorithm, (**c**) SO algorithm, and (**d**) GA.

3.3. Effect of Feed Grade and Particle Size Variation

Further analysis was carried out on the solutions found by the SA optimisation algorithm to ascertain how variation in the most difficult-to-control variables will affect copper recovery. This analysis focused on feed grade and feed particle size as it is quite difficult to keep these variables at their optimal operating points. To carry out this analysis, sixty typical plant observations were generated between the specified constraint bounds (Equation (34)) of both feed grade and feed particle size. The predictive function of the developed GPR model was then used to simulate copper recovery under three scenarios. In the first scenario, feed grade was varied using its generated observations while maintaining all other rougher flotation variables at their optimal operating values as found by the SA optimisation algorithm. The same approach was repeated for feed particle size in the second scenario. With regards to the last scenario, both feed grade and feed particle size were varied simultaneously with all other rougher flotation variables kept at their optimal operating values. The simultaneous variation was achieved by generating all the possible combinations between the two initially generated sixty observations of feed grade and feed particle size, which resulted in 3600 observations each. The visualisation of the simulation results is shown in Figure 12.

Table 6. Hypothetical analysis of the optimisation results.

Process Variables		SA Optimisation Algorithm	PSO Algorithm	SO Algorithm	GA
Feed grade (wt. %)		2.56	2.56	2.56	2.59
Feed particle size (% passing 75 μm)		82.80	84.00	84.00	83.99
Throughput (t/h)		638.02	533.25	534.14	536.24
Throughput (t/24 h)		15,312.48	12,798.00	12,819.36	12,869.76
Xanthate dosage (mL/min)	Tank cell 1	44.85	37.00	37.00	37.01
	Tank cell 4	47.78	34.86	34.88	34.98
Xanthate consumed in 24 h (mL/24 h)	Tank cell 1	64,584.00	53,280.00	53,280.00	53,294.40
	Tank cell 4	68,803.20	50,198.40	50,227.20	50,371.20
Total xanthate consumed in 24 h (mL/24 h)		133,387.20	103,478.40	103,507.20	103,665.60
Xanthate per tonne of copper produced in 24 h (mL/t)		359.08	329.79	329.34	324.77
Frother dosage (mL/min)	Tank cell 1	106.18	112.75	112.83	112.13
	Tank cell 4	74.58	102.68	103.05	102.85
Frother consumed in 24 h (mL/24 h)	Tank cell 1	152,899.20	162,360.00	162,475.20	161,467.20
	Tank cell 4	107,395.20	147,859.20	148,392.00	148,104.00
Total frother consumed in 24 h (mL/24 h)		260,294.40	310,219.20	310,867.20	309,571.20
Frother per tonne of copper produced in 24 h (mL/t)		700.73	988.68	989.11	969.86
Predicted Rougher copper recovery (%)		94.76	95.77	95.77	95.76
Rougher Copper recovered (t/24 h)		371.46	313.77	314.29	319.19
Dollar value of copper recovered in 24 h		AUD2,414,490.00	AUD2,039,505.00	AUD2,042,885.00	AUD2,074,735.00
Total xanthate cost in 24 h		−AUD160.06	−AUD124.17	−AUD124.20	−AUD124.40
Total frother cost in 24 h		−AUD338.38	−AUD403.28	−AUD404.13	−AUD402.44
Net gain in 24 h		AUD2,413,991.56	AUD2,038,977.55	AUD2,042,356.67	AUD2,074,208.16

Figure 12a shows that if feed grade varies within its constraint bound (1.6–2.6 wt.%), while keeping all other rougher flotation variables at their optimum values, copper recovery would increase continuously with an increasing feed grade to almost the upper limit of its constraint bound, recording values in the range of 93.44%–94.77%.

With regard to varying feed particle size within its constraint bound (78%–84% passing 75 μm), while maintaining all other rougher flotation variables at their optimum values, as visualised in Figure 12b, it can be observed that copper recovery would increase continuously until the optimum feed particle size (82.80% passing 75 μm) is attained, beyond which it will begin to decline. Copper recovery would be 94.49% at the coarsest milling (78% passing 75 μm), 94.76% at the optimum feed particle size, and 94.73% at the finest milling (84% passing 75 μm). This phenomenon is due to the fact fine milling results in the significant liberation of copper which further enhances recovery. However, as the optimum feed particle size is exceeded, slimes are generated which causes a decline in recovery.

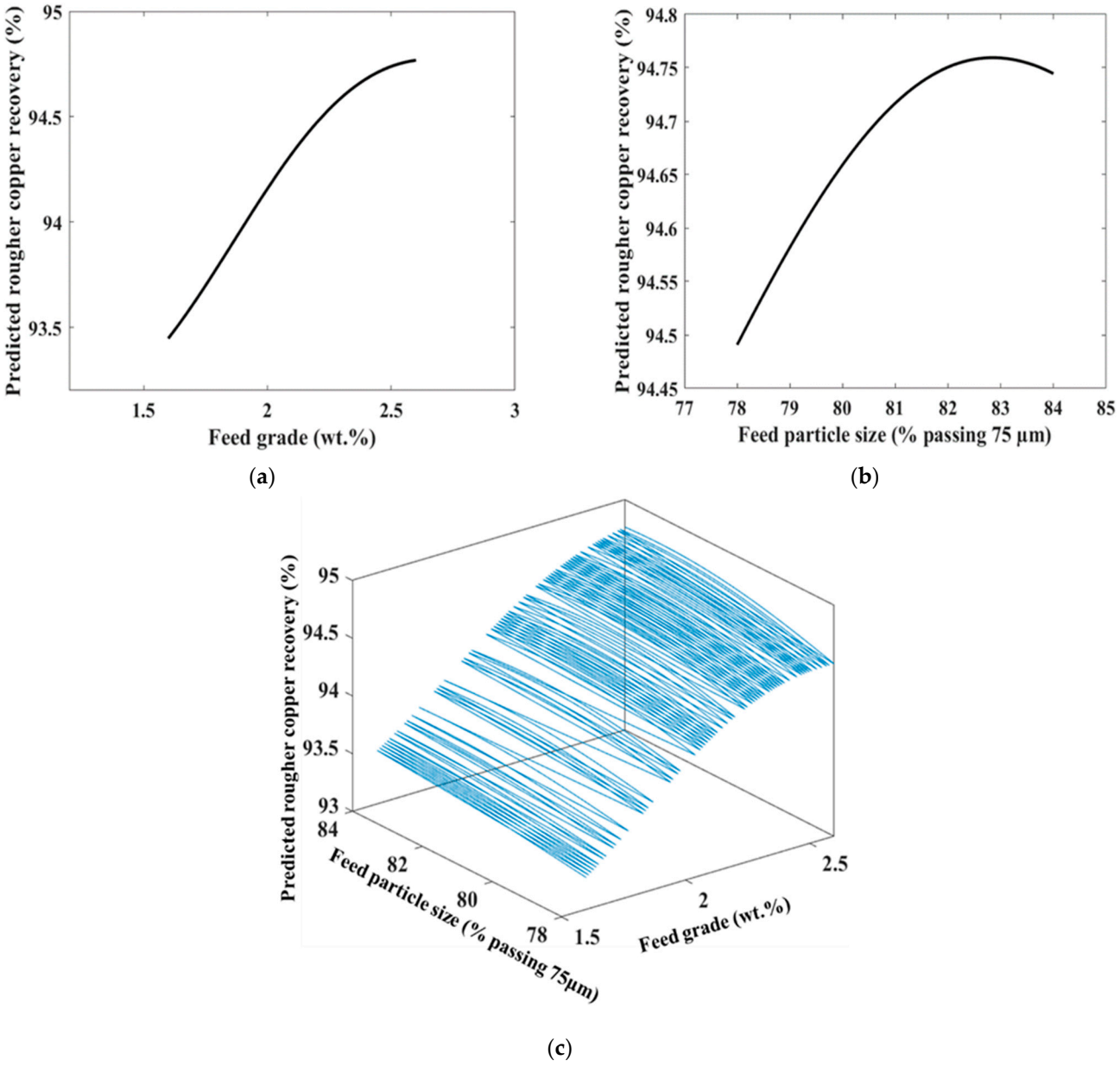


Figure 12. Visualisation of the variation effect of (**a**) feed grade, (**b**) feed particle size, and (**c**) both feed grade and feed particle size on copper recovery.

This simultaneous variation (Figure 12c) also had a similar performance in the range of 93.28%–94.77% copper recovery. These results confirm that near optimal copper recovery can still be realised within the constraint bounds of feed grade and feed particle size once all other rougher flotation variables are maintained at their optimum operating values as found by the SA optimisation algorithm. As highlighted earlier on, the outperformance of the SA optimisation algorithm over the other investigated optimisation algorithms could be linked to its ability to avoid being trapped in a local minimum and also accepting some changes that cause an increase in the objective function.

4. Conclusions

The prediction of copper recovery using support vector machine (SVM), Gaussian process regression (GPR), multi-layer perceptron artificial neural network (ANN), linear regression (LR), and random forest (RF) algorithms, followed by optimisation studies has been investigated with large industrial data from the BHP Olympic Dam. The individual

predictive model performance assessment showed that the GPR model developed with the matern 3/2 kernel function (GPR-matern 3/2) makes the most precise copper-recovery prediction, obtaining correlation coefficient (r) values > 0.96, root mean square error (RMSE) values < 0.42, mean absolute percentage error (MAPE) values $< 0.25\%$, and variance accounted for (VAF) values $> 94\%$ from the training, validation, and testing data sets. With the objective function extract from the GPR model, the optimisation of various rougher flotation variables for the maximisation of copper recovery was then investigated using a simulated annealing (SA) algorithm, particle swarm optimisation algorithm (PSO), surrogate optimisation (SO), and genetic algorithm (GA). A hypothetical analysis carried out on the set of solutions found by the investigated optimisation algorithms indicated that the SA algorithm finds the best set of solutions for the maximisation of copper recovery. Further analysis carried out on the solutions found by the SA algorithm indicated that near optimal copper recovery (>93%) can still be realised when feed grade and feed particles size are varied within their constraint bounds while maintaining all other rougher flotation variables at their optimum operating values. Implementing these machine learning solutions may enhance overall copper recovery while stabilising the operation. Extending the critical variable ranges (e.g., feed particle size and feed grade) beyond those investigated can be studied in the future to further understand their contributions.

Author Contributions: Conceptualization, B.A.-K., C.G. and R.K.A.; methodology, B.A.-K. and R.K.A.; validation, B.A.-K., M.Z, K.E., C.G. and R.K.A.; formal analysis, B.A.-K., M.Z. and R.K.A.; investigation, B.A.-K., C.M. and R.K.A.; resources, K.E., C.M., M.Z., C.G. and R.K.A.; data curation, B.A.-K., C.M., K.E. and R.K.A.; writing—original draft preparation, B.A.-K.; writing—review and editing, B.A.-K., C.M., M.Z., K.E., C.G. and R.K.A.; visualization, B.A.-K., C.M., M.Z., K.E., C.G. and R.K.A.; supervision, M.Z., K.E., C.G. and R.K.A.; project administration, M.Z., K.E., C.G. and R.K.A.; funding acquisition, M.Z., K.E., C.G. and R.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: Financial support from the Future Industries Institute of the University of South Australia is also acknowledged. Support from the Australia-India Strategic Research Fund for the Recovery of the Battery Materials and REE from Ores and Wastes is also acknowledged. Support from the Australian Research Council Centre of Excellence for Enabling Eco-Efficient Beneficiation of Minerals (grant number CE200100009) is gratefully acknowledged.

Acknowledgments: This research has been supported by the South Australian Government and the BHP Olympic Dam through the PRIF RCP Industry Consortium. The authors would also like to thank the BHP Olympic Dam for approving the publication of this article.

Conflicts of Interest: Christopher Greet is employees of Magotteaux Pty Ltd. The paper reflects the views of the scientists and not the company.

References

1. Graedel, T.; Bertram, M.; Fuse, K.; Gordon, R.; Lifset, R.; Rechberger, H.; Spatari, S. The contemporary European copper cycle: The characterization of technological copper cycles. *Ecol. Econ.* **2002**, *42*, 9–26. [CrossRef]
2. Asamoah, R.K. Specific refractory gold flotation and bio-oxidation products: Research overview. *Minerals* **2021**, *11*, 93. [CrossRef]
3. Calvo, G.; Mudd, G.; Valero, A.; Valero, A. Decreasing ore grades in global metallic mining: A theoretical issue or a global reality? *Resources* **2016**, *5*, 36. [CrossRef]
4. Kuipers, K.J.; van Oers, L.F.; Verboon, M.; van der Voet, E. Assessing environmental implications associated with global copper demand and supply scenarios from 2010 to 2050. *Glob. Environ. Chang.* **2018**, *49*, 106–115. [CrossRef]
5. Owusu, K.B.; Karageorgos, J.; Greet, C.; Zanin, M.; Skinner, W.; Asamoah, R.K. Predicting mill feed grind characteristics through acoustic measurements. *Miner. Eng.* **2021**, *171*, 107099. [CrossRef]
6. Owusu, K.B.; Zanin, M.; Skinner, W.; Asamoah, R.K. AG/SAG mill acoustic emissions characterisation under different operating conditions. *Miner. Eng.* **2021**, *171*, 107098. [CrossRef]
7. Owusu, K.B.; Skinner, W.; Asamoah, R. Feed hardness and acoustic emissions of autogenous/semi-autogenous (AG/SAG) mills. *Miner. Eng.* **2022**, *187*, 107781. [CrossRef]
8. Joseph-Soly, S.; Asamoah, R.K.; Skinner, W.; Addai-Mensah, J. Superabsorbent dewatering of refractory gold concentrate slurries. *Adv. Powder Technol.* **2020**, *31*, 3168–3176. [CrossRef]
9. Joseph-Soly, S.; Asamoah, R.; Addai-Mensah, J. Superabsorbent recycling for process water recovery. *Chem. Eng. J. Adv.* **2021**, *6*, 100085. [CrossRef]

10. Asamoah, R.K. EDTA-enhanced cyanidation of refractory bio-oxidised flotation gold concentrates. *Hydrometallurgy* **2020**, *193*, 105312. [CrossRef]
11. Asamoah, R.K.; Skinner, W.; Addai-Mensah, J. Enhancing gold recovery from refractory bio-oxidised gold concentrates through high intensity milling. *Miner. Process. Extr. Met.* **2019**, *129*, 64–73. [CrossRef]
12. Asamoah, R.K.; Skinner, W.; Addai-Mensah, J. Pulp mineralogy and chemistry, leaching and rheological behaviour relationships of refractory gold ore dispersions. *Chem. Eng. Res. Des.* **2019**, *146*, 87–103. [CrossRef]
13. Asamoah, R.K.; Baawuah, E.; Greet, C.; Skinner, W. Characterisation of metal debris in grinding and flotation circuits. *Miner. Eng.* **2021**, *171*, 107074. [CrossRef]
14. Quintanilla, P.; Neethling, S.J.; Brito-Parada, P.R. Modelling for froth flotation control: A review. *Miner. Eng.* **2021**, *162*, 106718. [CrossRef]
15. Forson, P.; Zanin, M.; Skinner, W.; Asamoah, R. Differential flotation of pyrite and arsenopyrite: Effect of hydrogen peroxide and collector type. *Miner. Eng.* **2021**, *163*, 106808. [CrossRef]
16. Forson, P.; Skinner, W.; Asamoah, R. Decoupling pyrite and arsenopyrite in flotation using thionocarbamate collector. *Powder Technol.* **2021**, *385*, 12–20. [CrossRef]
17. Forson, P.; Zanin, M.; Skinner, W.; Asamoah, R. Differential flotation of pyrite and Arsenopyrite: Effect of pulp aeration and the critical importance of collector concentration. *Miner. Eng.* **2022**, *178*, 107421. [CrossRef]
18. Forson, P.; Zanin, M.; Abaka-Wood, G.; Skinner, W.; Asamoah, R.K. Flotation of auriferous arsenopyrite from pyrite using thionocarbamate. *Miner. Eng.* **2022**, *181*, 107524. [CrossRef]
19. Dankwah, J.B.; Asamoah, R.K.; Zanin, M.; Skinner, W. Dense liquid flotation: Can coarse particle flotation performance be enhanced by controlling fluid density? *Miner. Eng.* **2022**, *180*, 107513. [CrossRef]
20. Dankwah, J.; Asamoah, R.; Zanin, M.; Skinner, W. Influence of water rate, gas rate, and bed particle size on bed-level and coarse particle flotation performance. *Miner. Eng.* **2022**, *183*, 107622. [CrossRef]
21. Dankwah, J.; Asamoah, R.; Abaka-Wood, G.; Zanin, M.; Skinner, W. Influence of bed material density on fluidized bed flotation performance: A study on the flotation performance of quartz and alumina beds in the HydroFloat. *Miner. Eng.* **2023**, *203*, 108321. [CrossRef]
22. Bergh, L.; Yianatos, J. The long way toward multivariate predictive control of flotation processes. *J. Process Control* **2011**, *21*, 226–234. [CrossRef]
23. Karimi, M.; Akdogan, G.; Bradshaw, S. A computational fluid dynamics model for the flotation rate constant, Part I: Model development. *Miner. Eng.* **2014**, *69*, 214–222. [CrossRef]
24. Laurila, H.; Karesvuori, J.; Tiili, O. Strategies for instrumentation and control of flotation circuits. *Miner. Process. Plant Des. Pract. Control* **2002**, *2*, 2174–2195.
25. Mathe, Z.; Harris, M.; O'Connor, C. A review of methods to model the froth phase in non-steady state flotation systems. *Miner. Eng.* **2000**, *13*, 127–140. [CrossRef]
26. Rao, S.R. Chapter 3—Physical and Physico-Chemical Processes. In *Waste Management Series*; Rao, S.R., Ed.; Elsevier: Amsterdam, The Netherlands, 2006; pp. 35–69.
27. Shean, B.J.; Cilliers, J.J. A review of froth flotation control. *Int. J. Miner. Process.* **2011**, *100*, 57–71. [CrossRef]
28. Amankwaa-Kyeremeh, B.; Ehrig, K.; Greet, C.; Asamoah, R. Pulp chemistry variables for gaussian process prediction of rougher copper recovery. *Minerals* **2023**, *13*, 731. [CrossRef]
29. Forson, P.; Skinner, W.; Asamoah, R. Investigating the selective flotation of auriferous arsenian pyrite from refractory ores using thionocarbamate. *Powder Technol.* **2023**, *426*, 118649. [CrossRef]
30. Asamoah, R.K.; Zanin, M.; Gascooke, J.; Skinner, W.; Addai-Mensah, J. Refractory gold ores and concentrates part 1: Mineralogical and physico-chemical characteristics. *Miner. Process. Extr. Met.* **2019**, *130*, 240–252. [CrossRef]
31. Asamoah, R.K.; Zanin, M.; Amankwah, R.K.; Skinner, W.; Addai-Mensah, J. Characterisation of Tectonic Refractory Gold Ore. In Proceedings of the CHEMECA 2014, Perth, Australia, 28 September–1 October 2014.
32. Asamoah, R.K.; Zanin, M.; Skinner, W.; Addai-Mensah, J. Refractory gold ores and concentrates part 2: Gold mineralisation and deportment in flotation concentrates and bio-oxidised products. *Miner. Process. Extr. Met.* **2019**, *130*, 269–282. [CrossRef]
33. Amankwaa-Kyeremeh, B.; Zhang, J.; Zanin, M.; Skinner, W.; Asamoah, R.K. Feature selection and Gaussian process prediction of rougher copper recovery. *Miner. Eng.* **2021**, *170*, 107041. [CrossRef]
34. Pu, Y.; Szmigiel, A.; Chen, J.; Apel, D.B. FlotationNet: A hierarchical deep learning network for froth flotation recovery prediction. *Powder Technol.* **2020**, *375*, 317–326. [CrossRef]
35. Al-Thyabat, S. On the optimization of froth flotation by the use of an artificial neural network. *J. China Univ. Min. Technol.* **2008**, *18*, 418–426. [CrossRef]
36. Aldrich, C.; Marais, C.; Shean, B.; Cilliers, J. Online monitoring and control of froth flotation systems with machine vision: A review. *Int. J. Miner. Process.* **2010**, *96*, 1–13. [CrossRef]
37. Hodouin, D. Methods for automatic control, observation, and optimization in mineral processing plants. *J. Process Control* **2011**, *21*, 211–225. [CrossRef]
38. Amankwaa-Kyeremeh, B.; Greet, C.; Zanin, M.; Skinner, W.; Asamoah, R.K. Selecting key predictor parameters for regression modelling using modified Neighbourhood Component Analysis (NCA) Algorithm. In Proceedings of the 6th UMaT Biennial International Mining and Mineral Conference, Tarkwa, Ghana, 5–6 August 2020; pp. 320–325.

39. Amankwaa-Kyeremeh, B.; Greet, C.; Zanin, M.; Skinner, W.; Asamoah, R.K. Predictability of rougher flotation copper recovery using Gaussian process regression algorithm. In Proceedings of the 6th UMaT Biennial International Mining and Mineral Conference, Tarkwa, Ghana, 5–6 August 2020; pp. 1–8.
40. Gomez-Flores, A.; Heyes, G.W.; Ilyas, S.; Kim, H. Prediction of grade and recovery in flotation from physicochemical and operational aspects using machine learning models. *Miner. Eng.* **2022**, *183*, 107627. [CrossRef]
41. Allahkarami, E.; Nuri, O.S.; Abdollahzadeh, A.; Rezai, B.; Chegini, M. Estimation of copper and molybdenum grades and recoveries in the industrial flotation plant using the artificial neural network. *Int. J. Nonferr. Met.* **2016**, *05*, 23–32. [CrossRef]
42. Schmidt, J.; Marques, M.R.G.; Botti, S.; Marques, M.A.L. Recent advances and applications of machine learning in solid-state materials science. *NPJ Comput. Mater.* **2019**, *5*, 83. [CrossRef]
43. Cook, R.; Monyake, K.C.; Hayat, M.B.; Kumar, A.; Alagha, L. Prediction of flotation efficiency of metal sulfides using an original hybrid machine learning model. *Eng. Rep.* **2020**, *2*, e12167. [CrossRef]
44. Massinaei, M.; Falaghi, H.; Izadi, H. Optimisation of metallurgical performance of industrial flotation column using neural network and gravitational search algorithm. *Can. Met. Q.* **2013**, *52*, 115–122. [CrossRef]
45. Jamróz, D.; Niedoba, T.; Pięta, P.; Surowiak, A. The Use of Neural Networks in Combination with Evolutionary Algorithms to Optimise the Copper Flotation Enrichment Process. *Appl. Sci.* **2020**, *10*, 3119. [CrossRef]
46. Ehrig, K.; McPhie, J.; Kamenetsky, V. Geology and mineralogical zonation of the Olympic Dam iron oxide Cu-U-Au-Ag deposit, South Australia. In *Geology and Genesis of Major Copper Deposits and Districts of the World, a Tribute to Richard Sillitoe*; Hedenquist, J.W., Harris, M., Camus, F., Eds.; Society of Economic Geologists: Littleton, CO, USA, 2012; Volume 16, pp. 237–268.
47. Amankwaa-Kyeremeh, B.; Skinner, W.; Asamoah, R.K. Comparative study on rougher copper recovery prediction using selected predictive algorithms. In Proceedings of the International Future Mining Conference, Sydney, Australia, 6–10 December 2021; pp. 1–10.
48. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 1995.
49. Bray, M.; Han, D. Identification of support vector machines for runoff modelling. *J. Hydroinform.* **2004**, *6*, 265–280. [CrossRef]
50. Tabari, H.; Kisi, O.; Ezani, A.; Talaee, P.H. SVM, ANFIS, regression and climate based models for reference evapotranspiration modeling using limited climatic data in a semi-arid highland environment. *J. Hydrol.* **2012**, *444-445*, 78–89. [CrossRef]
51. Duan, K.; Keerthi, S.; Poo, A.N. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing* **2003**, *51*, 41–59. [CrossRef]
52. Rasmussen, C.E. Gaussian Processes in Machine Learning. In *Summer School on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2003.
53. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; IEEE Cat. No. 04CH37541.
54. Zhu, Q.-Y.; Qin, A.K.; Suganthan, P.N.; Huang, G.-B. Evolutionary extreme learning machine. *Pattern Recognit.* **2005**, *38*, 1759–1763. [CrossRef]
55. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
56. Hecht-Nielsen, R. Kolmogorov's mapping neural network existence theorem. In Proceedings of the International Conference on Neural Networks, San Diego, CA, USA, 21–24 June 1987; IEEE Press: New York, NY, USA, 1987.
57. Kutner, M.H.; Nachtsheim, C.J.; Neter, J.; Li, W. *Applied Linear Statistical Models*; McGraw-Hill Irwin Boston: Columbus, OH, USA, 2005; Volume 5.
58. Dietterich, T.G. *Ensemble Methods in Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2000.
59. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
60. Auret, L.; Aldrich, C. Interpretation of nonlinear relationships between process variables by use of random forests. *Miner. Eng.* **2012**, *35*, 27–42. [CrossRef]
61. Singh, B.; Sihag, P.; Singh, K. Modelling of impact of water quality on infiltration rate of soil by random forest regression. *Model. Earth Syst. Environ.* **2017**, *3*, 999–1004. [CrossRef]
62. Bernard, S.; Heutte, L.; Adam, S. *Forest-RK: A New Random Forest Induction Method*; Springer: Berlin/Heidelberg, Germany, 2008.
63. Callens, A.; Morichon, D.; Abadie, S.; Delpey, M.; Liquet, B. Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. *Appl. Ocean Res.* **2020**, *104*, 102339. [CrossRef]
64. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer Series in Statistics; Springer: New York, NY, USA, 2009.
65. Quinlan, J.R. Learning with continuous classes. In Proceedings of the 5th Australian Joint Conference on Artificial Intelligence, Hobart, Tasmania, 16–18 November 1992; World Scientific: Singapore, 1992.
66. Breiman, L. *Classification and Regression Trees*; Routledge: New York, NY, USA, 2017.
67. Breiman, L. *Random Forests*; University of California Berkeley TR567: Berkeley, CA, USA, 1999.
68. Feller, W. *An Introduction to Probability Theory and Its Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2008; Volume 2.
69. Ingber, L. Adaptive simulated annealing (ASA): Lessons learned. *Control Cybern.* **1996**, *25*, 33–54.
70. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]

71. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
72. Mezura-Montes, E.; Coello, C.A.C. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm Evol. Comput.* **2011**, *1*, 173–194. [CrossRef]
73. Sadrossadat, E.; Basarir, H.; Luo, G.; Karrech, A.; Durham, R.; Fourie, A.; Elchalakani, M. Multi-objective mixture design of cemented paste backfill using particle swarm optimisation algorithm. *Miner. Eng.* **2020**, *153*, 106385. [CrossRef]
74. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [CrossRef]
75. Regis, R.G.; Shoemaker, C.A. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* **2007**, *19*, 497–509. [CrossRef]
76. Powell, M.J. The theory of radial basis function approximation in 1990. In *Advances in Numerical Analysis*; Oxford University Press: Oxford, UK, 1992; pp. 105–210.
77. Wang, Y.; Shoemaker, C.A. A general stochastic algorithmic framework for minimizing expensive black box objective functions based on surrogate models and sensitivity analysis. *arXiv* **2014**, arXiv:1410.6271.
78. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
79. Yu, T.-L.; Santarelli, S.; Goldberg, D.E. Military antenna design using a simple genetic algorithm and hBOA. In *Scalable Optimization via Probabilistic Modeling*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 275–289.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.