Archives of Computational Methods in Engineering
https://doi.org/10.1007/s11831-025-10371-z

RESEARCH

Novel Hybrid ANN-Interpolation Techniques for Predicting Mean Residence Time in Wet Twin Screw Granulation Application: A Critical Review

Mitchell Rae[1] · Vivek V. Ranade[2] · Gavin Walker[3] · Stefan Heinrich[4] · Rohit Ramachandran[5] · Mehakpreet Singh[1]

Received: 31 January 2025 / Revised: 4 April 2025 / Accepted: 16 August 2025

Abstract

In twin-screw granulation (TSG), the mean residence time (MRT) of materials significantly influences granule properties, such as size distribution and density, impacting the quality of the final product. Accurately estimating MRT is crucial because deviations can lead to overwetting, compaction issues, or insufficient granulation. This study presents a hybrid approach that combines machine learning and data interpolation techniques to model MRT as a function of process parameters, including feed flow rate, screw speed, screw configuration, and liquid-to-solid ratio. Our goal is to develop a predictive tool capable of handling coarse datasets for precise MRT estimation. By optimising the MRT, process control, efficiency, and batch-to-batch consistency can be improved, ensuring adherence to product specifications and facilitating cost-effective scale-up. This study explores the integration of various univariate and multivariate spline interpolation techniques with the nonlinear autoregressive with exogenous inputs (NARX) and multilayer perceptron (MLP) machine learning methods to enhance the accuracy of MRT. While numerous studies have utilised large datasets, this study examines a coarse dataset, applying various interpolation techniques to enhance data resolution and consequently improve the performance of the NARX machine learning model. This study examined training and testing datasets of different sizes, demonstrating the versatility and applicability of the coupled methodology. Our findings demonstrate the advantages of multivariate cubic spline interpolation with the NARX approach over MLP and Kriging with univariate interpolation methods. This paper presents a comprehensive review of existing interpolation techniques and their impact on modeling performance, addressing a critical gap in the current literature. The results show that the multivariate cubic spline interpolation with the NARX approach achieved a 72% reduction in the root mean square error (RMSE) and an 85% increase in the adjusted R^2 compared to the existing Kriging interpolation technique (Ismail et al. in Powder Technol 343:568–577, 2019). In terms of computational efficiency, the NARX approach with univariate and multivariate spline interpolations are 16 times more efficient than the Kriging interpolation technique.

✉ Mehakpreet Singh
Mehakpreet.Singh@ul.ie

Mitchell Rae
rae.mitchell@ul.ie

Vivek V. Ranade
Vivek.Ranade@ul.ie

Gavin Walker
Gavin.Walker@ul.ie

Stefan Heinrich
stefan.heinrich@tuhh.de

Rohit Ramachandran
rohit.r@rutgers.edu

[1] Mathematics Applications Consortium for Science and Industry (MACSI), Department of Mathematics and Statistics, University of Limerick, Limerick V94 T9PX, Ireland

[2] Multiphase Reactors and Process Intensification Group, Department of Chemical Sciences, University of Limerick, Limerick V94 T9PX, Ireland

[3] Bernal Institute, Department of Chemical Sciences, University of Limerick, Limerick V94 T9PX, Ireland

[4] Institute of Solids Process Engineering and Particle Technology, Hamburg University of Technology, Denickestraße 15, 21073 Hamburg, Germany

[5] Chemical and Biochemical Engineering, Rutgers University, 98 Brett Road, Piscataway, NJ 08854, USA

Published online: 14 September 2025

1 Introduction

The pharmaceutical industry requires precise and accurate machines to produce products with consistent and safe qualities. Twin screw granulation (TSG) is an increasingly popular method for producing fine-grain powder products. TSG provides control over several essential properties of the material, such as particle size, particle size distribution, compactibility, distribution of particles, solubility, and dissolution properties. Control over these properties is critical to the final quality of the product. Furthermore, confidence in the accuracy of these qualities substantially improves the safety and marketability of products, especially in the pharmaceutical industry. Extensive research has been conducted on the simulation and mathematical modelling of TSG processes in recent years [1, 2]. Research often aims to provide insight and predictability of response variables, such as mean residence time, particle size distribution, granule porosity and density, and many more response variables. Modelling these response variables based on independent variables such as screw speed, powder and airflow rates, and input materials will allow significant advancements in product manufacturing efficiency and optimisation.

Twin-screw granulation involves two identical screws that rotate within a barrel. The rotation of the twin screws forces the particles in the barrel to interact, encouraging physical processes such as aggregation, growth, fragmentation, and nucleation [3–6]. Figure 1 shows a simplified TSG for explanatory purposes.

It shows the basic dynamics of the TSG, material entry and exit, screw rotation and angle, and spacing between screws. These factors are critical to consider when calibrating TSG equipment and labs to industrial scale. Similarly, these factors, along with screw speed (v_{screw}, feeding rate, liquid-to-solid ratio (L/S), and initial particle size, are crucial to accurately model response variables, such as mean residence time (MRT). TSG designs can vary by simple factors such as screw positioning and movement; however, there are some design factors that require equipment replacements such as screw slanting angle, screw length, screw sections (e.g. kneading zones), material input locations, and screw shape. These factors contribute to the interaction rate of particles within the system, impacting the output particle properties. There are many existing approaches including discrete element methods [7, 8] machine learning approaches, [9–12] and population balance models [13–15] for modelling the powder particles within the TSG process, each holding both merits and demerits to their methods, individually [6]. Among all approaches, we focus on machine learning approaches for modelling the MRT in the wet twin screw granulation process.

Many data-driven models require large datasets, costing hours of expensive experimentation, and material waste to develop predictive models [1, 10, 12, 17]. However, it is becoming clear that large datasets are not always realistic, practical, or necessary [18]. This paper presents mathematical methods that reduce the necessity for copious and unnecessary data acquisition. This study also provides an overview of the TSG process and its key properties. It will briefly introduce some leading mathematical approaches to modelling TSG before focusing on the machine learning applications of predicting MRT correlating with the process parameters, including L/S, and powder flow rate (Q). The insights presented in this study address a significant gap in the literature by showcasing the effective use of coarse datasets to develop efficient predictive models. This approach highlights the potential to minimise unnecessary experimentation, thereby reducing experimental waste and associated costs.

1.1 Existing Modelling Techniques for TSG Process

The existing literature contains many approaches for modelling the TSG processes including the finite element method (FEM) [19], computational fluid dynamics (CFD) [9], and population balance modelling (PBM) [4, 20]. CFD and FEM approaches require the repeated calculation of complicated partial differential equations (PDEs), and their resolution depends on the size of the mesh over which the simulation is performed. These factors cause the FEM and CFD simulation approaches to have high computational costs. Additionally, PBMs can be used to reduce the computational

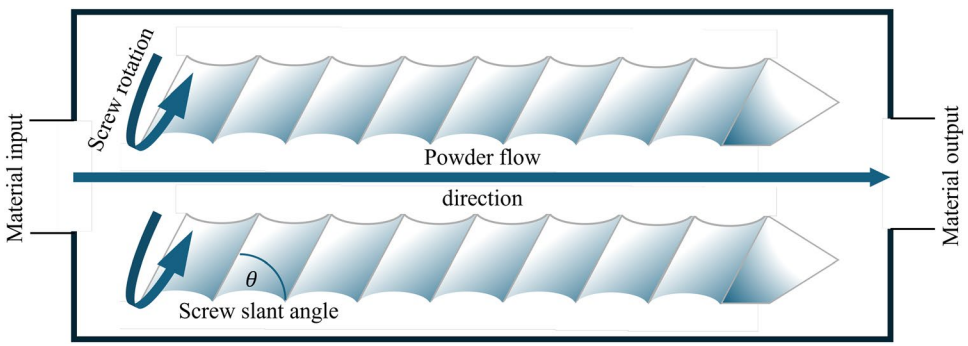


Fig. 1 Schematic of basic TSG mechanism

cost, however, analytical solutions are typically challenging to achieve; leading to the use of semi-analytical approximations which can significantly reduce the accuracy over extended temporal conditions [21–27]. The PBM approach determines or approximates the number density function of an integro-partial differential equation such as the Smoluchowski equation [28–34] or the Lotka-Volterra equations [35, 36]. PBMs generally describe the number density of a certain quality or property over time, and their complexity can vary significantly as they describe vastly different phenomena.

Machine learning (ML) and data-driven approaches have been popular alternative solutions for many modelling problems across all disciplines. This is because ML techniques, such as artificial neural networks (ANNs), require minimal computational cost and often provide highly accurate predictions. ANNs have been praised for their adaptability and robust performance without any physics-based knowledge. Recent advances have highlighted the potential for hybrid approaches that couple data-driven techniques with other relevant mathematical, physics, and chemistry principles; enhancing both precision and interpretability. For example, Dai et al. [37] presented a hybrid Gaussian process model, enhanced by the Wasserstein distance, which showcases the potential for soft sensing in extruder processes. Similarly, Jia et al. [38] combined physical principles with graph learning to predict process indicators, thereby bridging the gap between physics-based and data-driven methods. This study combines machine learning with interpolation techniques to optimise MRT prediction within TSG processes.

Interpolation tools can be used to expand a coarse dataset (as seen in Fig. 3) which effectively improves the performance of the training algorithm, thus improving the model predictive power [39, 40]. It is common for deep learning and machine learning approaches to rely on large datasets which can be impactiocal and unrealistic for some applications. Thus, the approach proposed in this study aims to demonstrates the possibility of effectively training models using coarse data, reducing the required experimental observations.

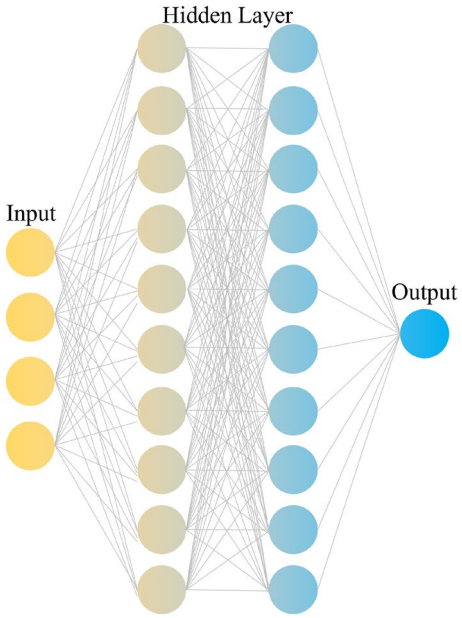


Fig. 2 Graphical representation of a simple fully connected ANN

1.2 Existing Machine Learning Approaches

Here, a brief overview of the relevant ML techniques and an introduction to ANNs provide the necessary knowledge for proceeding topics and research findings. First, machine learning encapsulates a broad range of statistical and mathematical processes that generally focus on predictive models and data analysis. Abundant ML statistical techniques include feature extraction and selection, anomaly detection, regression analysis, dimensionality reduction, clustering analysis, and many more which provide a comprehensive array of tools for gaining insight into complex datasets [41, 42]. Statistical ML methods streamline data analysis by offering powerful tools for understanding and extracting meaningful insights from complex datasets. They are advantageous over traditional statistical techniques because of their adaptability, automation, and ability to uncover hidden patterns without needing pre-labelled data. Additionally, ML techniques are capable of developing high-performance predictive models. ANNs are an increasingly popular ML method for predictive modelling and analysis as they can perform extremely well under various conditions without any physical understanding of the relevant system. The adaptability and automation of ANNs provide an attractive alternative to traditional predictive models while benefiting from low computational cost and low development effort. However, ANN predictive models are often considered to be “black box" models because the patterns and properties recognised by the ANN are challenging to extract. Therefore, ANNs models can often fail to provide desirable mathematical and physical insights of the system [43–45].

There are many designs and parameters to be considered when developing ANN. It is helpful to first consider the physical representation of an ANN. Figure 2 displays a simple ANN design consisting of round points representing the neurons, or nodes, of the network. Numerous lines connecting neurons to other neurons are called synapses or pathways. Data is fed into the input layer, with each node representing an individual input variable, and passed along the first set of synapses to reach the first hidden layer. Each synapse applies an activation or transfer function when information is passed from one node to the next. activation functions are used to map the data to a specific domain.

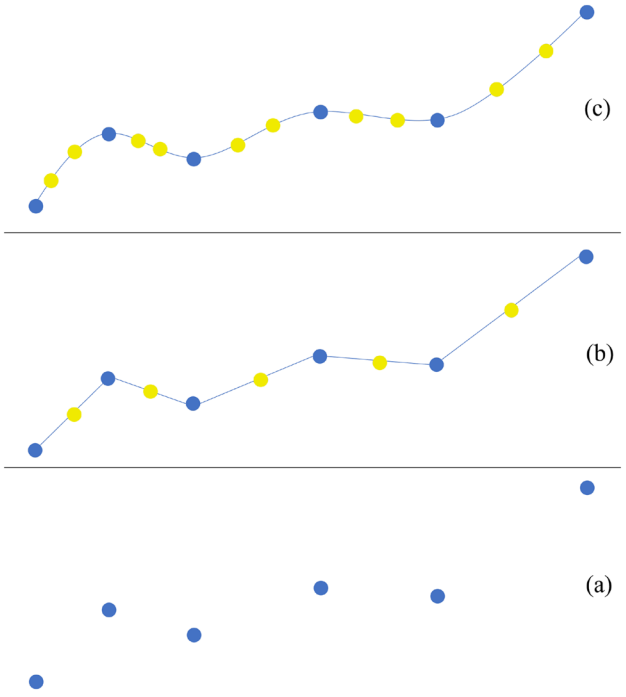


Fig. 3 Graphic demonstration of interpolation of the original dataset (**a**). Interpolation in (**b**) uses a linear interpolation equation and a ratio of $\frac{1}{2}$, whereas, **c** uses a nonlinear interpolation method and has an interpolation ratio of $\frac{1}{3}$

Popular activation functions include the sigmoid, hyperbolic tangent, and rectified linear unit (ReLu) functions. The sigmoid activation function uses $\sigma(x) = \frac{1}{1+e^x}$ to map all input values to the set $S_s \in (0, 1)$ The ReLU activation function uses $\mathrm{f(x)} = \max(0, \mathrm{x})$, mapping all negative values to 0 and all positive values to themselves. The *tanh* function uses $\tanh(x) = \frac{e^x - e^x}{e^x + e^x}$ to map input values to the set $S_t \in (-1, 1)$ [46, 47].

The nodes of the ANN include a *weight* which is a controlled variable assigned to each node. The weight represents the significance of the value at the node. The weights of each node are updated after every iteration of the training algorithm. Training an ANN is a clever mathematical and statistical process that modern technology is capable of utilising. There are several training algorithms and techniques for training an ANN by applying similar logic. They generally begin with a random weight distribution throughout the network, feed the data through, and calculate the difference between the ANN output (prediction) and the known output (expected). The difference, or error, is then used to update the weights such that the error decreases. By repeating this process over numerous iterations and with sufficient input and output data, the model trains itself to accurately predict the output for datasets that are separate from the data used for training. This is how the success of an ANN model is measured, based on the predictive ability of data reserved for testing. However, ANNs are not always trained effectively or successfully, producing incorrect predictions or unreliable results. Several factors contribute to ANN failure. Thankfully, novel advancements in training algorithms help improve the robustness of ANN training by applying specific mathematical algorithms that overcome common challenges and improve the adaptability and robustness of the ANN model. Popular and successful training algorithms that are seen repeatedly in the literature include the Levenberg–Marquardt (LM) [48, 49], Bayesian regularisation backpropagation (BR) [50–52], and stochastic gradient descent methods.

Major challenges faced by ANN include inherent variability in results owing to the randomness of initiation, getting trapped in local minima when reducing error between iterations, overfitting to the training data, and over-complicating the dataset. A significant variation observed between runs of the model can be caused by many factors. The main factor to consider is the initial weight distribution in the neural network (NN). This can lead to varying results if the model is trapped in different local minima in each attempt. The variability can be mitigated by constraining the initial weight distribution or using different training algorithms which may handle the dataset better. The training algorithm can also be a solution to the overfitting problem; some algorithms introduce small perturbations to escape local minima that may cause variations between each iteration.

Overfitting occurs when the model is essentially working as a line-of-best-fit model and projects the exact trend on the testing data, causing a discrepancy between the predicted and expected values if the trend does not continue perfectly between the training and testing datasets. Finally, a more obvious, yet still significant, issue is the over-complication of the dataset. This can occur because of poor preprocessing practices, poor interpolation tools, and incorrect or poor input feature identification. Some ANN designs have flexible and adaptable parameters, whereas others require delicate selection and tuning of parameters to perform well. For some ANNs, the number of layers, size of the hidden layers, and training algorithm can have significant impacts on the performance and require careful adjustment of these parameters to find the optimal setting. This is not always an explicit characteristic of the ANN design; rather, it is a result of pairing the ANN design with the dataset. Some ANNs perform well and are flexible and adaptable to certain types of datasets, as opposed to others.

Another challenge faced by ML techniques is the effective training of ANN with a small dataset. This study addresses this issue by coupling ANN and interpolation techniques

and demonstrating the results based on MRT predictions. The purpose of coupling interpolation with an ANN model is to improve its predictive performance. Interpolating the dataset provides more data points for the ANN to learn and detect patterns in the data. A larger dataset allows the model to interpret the patterns faster and with more reinforcement, thus leading to a stronger predictive model. An ANN that has been trained on the interpolated data will then be able to predict the future outputs corresponding to input variable adjustments which is not possible with interpolation alone.

2 Experimental Datasets and Preprocessing

The type of data used in ANN modelling and preprocessing practices can have significant impacts on the performance of the model. Coarse datasets have fewer observations with respect to the changes made to the variables. Developing a model with a coarse dataset can be challenging because coarse datasets make it difficult to discern between correlation and causation, interactions between variables may be overlooked, and the training of the model with coarse datasets may ultimately mislead the model and provide poor predictions. The challenges arising from coarse datasets can be alleviated by careful preprocessing. This study focuses on preprocessing methods to improve the functionality of coarse datasets. The dataset in this study is coarse, with only 96 total measurements, compared to datasets seen in studies such as [53]. Preprocessing can involve input feature selection and exclusion, data reorganisation, normalisation, interpolation, and many more techniques.

Data interpolation can be a useful tool for improving the performance of machine learning approaches. Interpolating a dataset can increase the size of the training data which can lead to more efficient training. However, interpolation, similar to all preprocessing methods, must be carefully considered to avoid the presentation of misleading results.

Figure 4 demonstrates a common example of how interpolation tools can effect the resulting predictions. In Fig. 4a, the data are not segregated into testing and training in a blocked format; rather, the testing and training data are mixed and interleaved. This is a perfectly normal method of data separation without interpolation; however, using this method after interpolation causes the model to essentially train with the testing data. This is more apparent as the number of interpolated points increases, and the difference between the interpolated and original points becomes indistinguishable. However, in Fig. 4b, the data are clearly separated, and the interpolated model uses the behaviour of only the first half of the dataset to predict the second half. In Fig. 4b, increasing the number of interpolated points causes the original and interpolated points to become closer and more similar; however, the clear separation from training to testing data means that the model will not be trained on data that is indistinguishable to the original data, as seen in Fig. 4a. This demonstrates the necessity for careful consideration of interpolation methods and data segregation for training and testing. Furthermore, careful consideration of the nature of the dataset and the purpose of the model should be considered when addressing the plausibility of these methods.

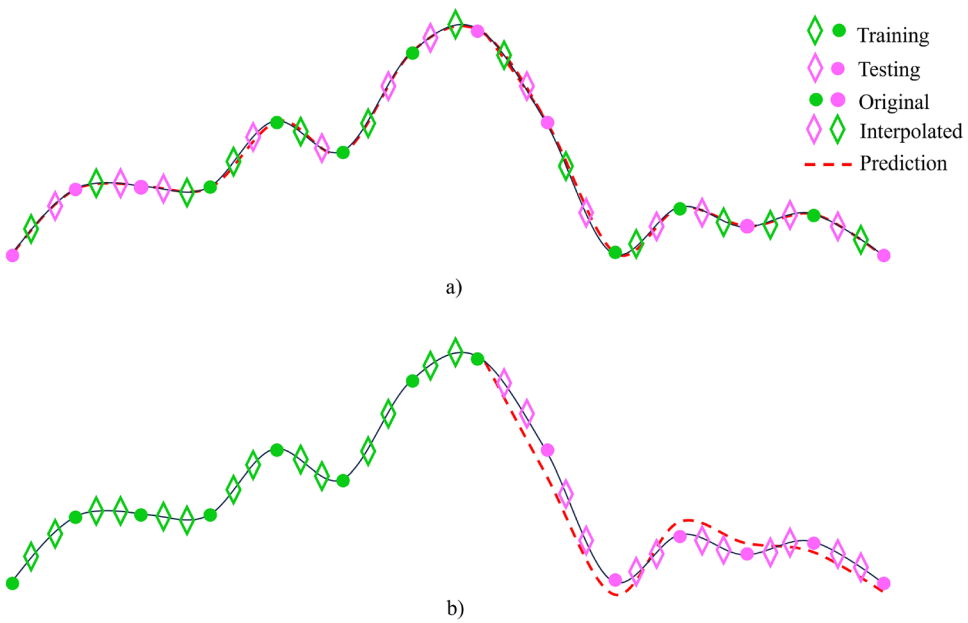


Fig. 4 Example effects of interpolation tools. **a** shows the common random selection of testing and training data after-interpolation, displaying how the predictions of the original data can be misleadingly accurate because of the nearby interpolants used in the training data. **b** represents a block-wise training and testing separation that considers the time-series nature of the data after interpolation; keeping all training data completely separate from the testing data

3 Machine Learning Approaches: NARX and MLP Algorithms

In this section, a concise overview of the competing MLP and NARX ANN structures is provided in a specific application to the problem presented in this paper. The MLP and NARX architectures have the same generalised blueprint; however, the NARX includes a feedback delay and input delay component. This key component causes significant performance differences [54, 55]. The model overview begins with the data input and concludes with MRT prediction.

The MLP uses a simple architecture with three inputs, five nodes in the first hidden layer, five nodes in the second hidden layer, and a single node output layer. Every layer is fully connected using the tansig transfer function, and the initial weight distribution throughout the network is random. The MLP uses an interpolation tool to increase the size of the dataset, providing more unique observations that are used to train the model. The interpolation tool is implemented before the data enter the input nodes of the MLP. This allows for the increased performance of the models' predictive power; however, the choice of the interpolation method significantly affects the model improvements. The choice of the interpolation tool should agree with the dataset variables, model architecture, and training algorithm.

Similarly, the NARX ANN uses the same basic architecture as the MLP: three input nodes, five nodes in the first hidden layer, five nodes in the second hidden layer, and a single output node for MRT. However, the NARX also includes an input delay and feedback delay component. The input delay allows the model to use previous input variable observations to inform the prediction of the current output. The input delay for this developed NARX ANN was set to zero, simply using current inputs to predict the current MRT. However, the feedback delay, which indicates the number of previous output predictions used to predict the current output, was 1:2. This implies that the previous output prediction is used to inform the current output prediction. This allows NARX to capture the temporal behaviour of time-series data better.

The long short-term memory (LSTM) ANN is another highly popular ANN. It has recently been applied for purposes similar to those explored in this study [56]. Its defining characteristic is the existence of an LSTM layer in the ANN, without which the model would be more attuned to the MLP architecture. The LSTM layer functions as a complex decision gate. The values passed from the previous layer are fed into one of four gates: the forget gate, retain gate, cell candidate gate, or output gate. This allows the model to detect long-term trends within the dataset, often significantly improving the forecasting and predictability of the model. The LSTM was not thoroughly explored in this study because an effective comparison using the Bayesian regression training algorithm required the development of a custom training loop to include the Bayesian regression training algorithm in the LSTM framework or the development of a custom LSTM layer in the NARX framework in MATLAB.

4 Interpolation Techniques

Interpolation can be achieved using various methods. Different interpolation methods can be graded based on their respective computational cost, smoothness, and performance with respect to the number of interpolated points. Here, several popular interpolation methods have been introduced. Each of the following methods were examined in this study.

4.1 Linear Interpolation

One of the most simple interpolation tools, linear interpolation defines a straight line between two points and creates a new data point halfway between the two existing points. This linear interpolation can be mathematically defined as:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}, \tag{1}$$

where (x_0, y_0) and (x_1, y_1) are the known data points used to find the interpolants (x, y) between them. This method will develop only straight lines between the original data points which can often be seen as a significant limitation when interpolating continuous variables as it fails to capture the behaviour of the dataset.

4.2 Cubic Spline Interpolation

The cubic spline approach attempts to find cubic polynomials $q_i(x) = y$ to interpolate between (x_i, y_i) and $(x_{i+1}, y_{i+1})\ \forall\ i \in [0, n]$. Although there are multiple specific methods that fall under the "spline" group, the cubic spline is a simple and highly popular approach. The cubic spline follows some key assumptions that state:

$$\begin{cases} q_i(x_i) = q_{i+1}(x_i) = y_i, \\ q_i'(x_i) = q_{i+1}(x_i), \quad 1 \le i \le n-1, \\ q_i''(x_i) = q_{i+1}(x_i). \end{cases} \tag{2}$$

Here, q' and q'' are the first and second derivatives with respect to x, respectively. The algorithm used for interpolating with the cubic spline approach is as follows:

$$
\begin{aligned}
(x) = &\left(1-\frac{x-x_1}{x_2-x_1}\right) y_1 + \frac{x-x_1}{x_2-x_1} y_2 \\
&+ \frac{x-x_1}{x_2-x_1}\left(1-\frac{x-x_1}{x_2-x_1}\right)\left(1-\frac{x-x_1}{x_2-x_1}\right) \\
&((q_1'(x_1)(x_2-x_1)+(y_2-y_1)) \\
&+ \frac{x-x_1}{x_2-x_1}\left(-q_2''(x_2)(x_2-x_1)+(y_2-y_1)\right),
\end{aligned} \tag{3}
$$

or

$$
\begin{aligned}
q(x) =&(1-t(x))y_1 + t(x)y_2 \\
&+ t(x)(1-t(x))((1-t(x))(k_1(x_2-x_1) \\
&+ (y_2-y_1)) + t(x)(-k_2(x_2-x_1)+(y_2-y_1)),
\end{aligned} \tag{4}
$$

where $t(x) = \dfrac{x-x_1}{x_2-x_1}$, $k_1 = q_1'(x_1)$, and $k_2 = q_2''(x_2)$. Clearly, this is much more complicated than the linear interpolation approach; therefore, we can expect an increased computational cost for the cubic spline method.

4.3 Kriging Interpolation

The Kriging interpolation method, also known as Gaussian process regression, was originally developed for estimating the most likely distribution of gold in a given area based on sample data. This method has received much attention since its development in 1960 where it was designed as a method of spatial interpolation for gold mining in South Africa. The method has since led to many modified versions that target specific issues and purposes. Kriging involves two key steps: 1) fitting a variogram γ to find the covariance structure of the sample points, and 2) calculating the weights $z(x_i)$ of the clustered points in a neighbourhood of x_k [57]. The benefit of the Kriging method is that it considers two input variables (*x*, *y*) to interpolate the response (*z*) as opposed to alternative methods which are typically operate exclusively on one variable. Kriging is therefore often applied to spatial data using 2D coordinates as the input variables and measurable quantities such as density and total content as the response (*z*). The major limitation of Kriging is that increasing dimensionality is challenging, and it has strict assumptions that may not be met by many datasets. [57, 58].

The computations involved in Kriging are expensive and complicated compared with both linear and cubic interpolation. The Kriging method requires the calculation of an empirical variogram:

$$
\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [z(x_i) - z(x_i+h)]^2. \tag{5}
$$

Here, $\gamma(h)$ is the semivariance at lag distance *h*, $z(x_i)$ and $z(x_i+h)$ are the known observations at locations x_i and x_i+h, respectively. Next, we solve a system of linear equations to determine the weights for known data points. This requires constructing a covariance matrix C

$$
C_{ij} = \mathrm{Cov}\left(z\left(x_i\right), z\left(x_j\right)\right)
$$

Then, the covariance vector:

$$
c_i = \mathrm{Cov}\left(z\left(x_i\right), z\left(x_0\right)\right) \quad ,
$$

where *c* represents the covariance between the known data points and the target point $z(x_0)$. Then, augment for Kriging Weights:

$$
\begin{bmatrix} C & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{1} \end{bmatrix}.
$$

using a Lagrange multiplier (λ) to ensure unbiasedness. Here, w are Kriging weights and 1 is the vector of ones for the unbiasedness condition. We then solve the kriging weights for

$$
\mathrm{w} = C^{-1}\mathrm{c},
$$

and use these to predict the values at the interpolated locations using the following equation:

$$
z(x_0) = \sum_{i=1}^{N} w_i z(x_i).
$$

Finally, it is optional to calculate the Kriging variance to quantify the uncertainty of the predictions:

$$
\sigma^2(x_0) = \gamma(0) - \mathrm{c}^T C^{-1}\mathrm{c},
$$

where c^T denotes the transpose of c. $\gamma(0)$ is the sill which is the variance of the observed data $\gamma(0) = Var(z(x)) = \sigma^2$. In this study, the Kriging method was extended to include three input variables, making it a 3D method. Thus, x_i can be replaced with the three vectors $\mathrm{x}_i = [x_i, y_i, t_i]$ in all steps, and the distance between the points will become $\mathrm{x}_i - \mathrm{x}_0 = \sqrt{(x_i-x_0)^2 + (y_i-y_0)^2 + (t_i-t_0)^2}$.

4.4 Cubic Polynomial Interpolation

Cubic interpolation uses four points to interpolate a new point by designing a cubic polynomial from the four points and using the polynomial to estimate a point along the smooth curve. Consider the four points x_0, x_1, x_2, and x_3.

To interpolate a point between x_1 and x_2, say $x_{1+\frac{1}{2}}$, the following cubic interpolation formula is used:

$$\begin{aligned} f(x_0, x_1, x_2, x_3, x) =& x^3 \left(\frac{1}{2}x_0 - \frac{3}{2}x_1 + \frac{3}{2}x_2 - \frac{1}{2}x_3 \right) \\ &+ x^2 \left(x_0 - \frac{5}{2}x_1 + 2x_2 - \frac{1}{2}x_3 \right) \\ &+ x \left(-\frac{1}{2}x_0 + \frac{1}{2}x_2 \right) + x_1. \end{aligned} \tag{6}$$

Thus, we can interpolate $f(x)$ between any two values of x_i. Furthermore, when interpolating between endpoints x_0 and x_1 or between x_{n-1} and x_n, an extra point must be fabricated. Many simple methods can be used to accomplish this. Cubic interpolation is more computationally expensive than linear interpolation but less expensive than Kriging.

4.5 Makima Spline Interpolation

The Makima interpolation method is a modification of the Akima interpolation method [59]. The modification improves the weight calculations in flat regions where two slopes in the dataset converge. Makima uses the cubic polynomial function (4) as in the cubic spline method, and the slope between two points is given by

$$\delta_i = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}. \tag{7}$$

The derivative of the polynomial function is calculated as

$$d_i = \left(\frac{w_1}{w_1 + w_2} \right) \delta_{i-1} + \left(\frac{w_2}{w_1 + w_2} \right) \delta_{i-1},$$

where w_1 and w_2 are weights, which are calculated as:

$$\begin{aligned} w_1 =& |\delta_{i+1} - \delta_i| + \frac{|\delta_{i+1} - \delta_i|}{2}, \\ w_2 =& |\delta_i - \delta_{i-1}| + \frac{|\delta_i - \delta_{i-1}|}{2}. \end{aligned}$$

The derivative d_i is then used in the Hermite interpolation function:

$$\begin{aligned} f(x_i) =& f(x_{i-1}) H_1(x) + f(x_{i+1}) H_2(x) \\ &+ d_{i-1} H_3(x) + d_{i+1} H_4(x). \end{aligned} \tag{8}$$

Here, H are the Hermite functions [60]. This interpolation method is more computationally expensive than the linear interpolation method; however, it is less computationally expensive than the kriging method.

4.6 Multivariate Cubic Spline Interpolation

The multivariate cubic spline interpolation (MCSI) method is a more complicated version of the aforementioned univariate cubic spline. The univariate cubic spline develops a cubic polynomial between two points of a single variable and interpolates a point in between, whereas the multivariate method extends this approach to multiple dimensions, maintaining key properties such as smoothness and continuity. The basic idea of the multivariate cubic spline interpolation method is to take the given set of data points $\{\mathrm{x}_i, f(\mathrm{x}_i)\}$ in m-dimensional space ($x_i = [x_{i1}, x_{i2}, .., x_{im}]$), and construct the set $S(\mathrm{x}_i)$ such that $S(\mathrm{x}_i) = f(\mathrm{x}_i) \quad \forall \quad i = [1, 2, 3, ...n]$. This is performed using the multivariate cubic spline expression

$$S(\mathrm{x}_i) = \sum_{i=1}^{n} w_i K(||\mathrm{x} - \mathrm{x}_i||) + P(\mathrm{x}), \tag{9}$$

where K is a radial basis function, w_i are the weights to determine when solving the linear system of equations, $P(\mathrm{x})$ is a polynomial which ensures smoothness. There are many possible radial basis functions to use. Generally, they replace the interval-specific polynomials found in the univariate cubic spline interpolation method. The radial basis function can represent the multidimensional behaviour of the dataset, making it a key element in this method. A typical radial basis function is $K(r) = r^3$, where $r = ||\mathrm{x} - \mathrm{x}_i||$. The multivariate cubic spline method is closer in comparison to the Kriging method, as both methods consider the dataset as a multidimensional system and interpolate points in a multidimensional function. This helps the interpolation method to represent the multidimensional behaviour of the dataset. Figure 5 displays how Eq. 9 interpolates each variable of a dataset while maintaining maximum smoothness and allowing only slight deviations from the original observations.

Table 3 compares these interpolation methods based on complexity, smoothness, and best use cases

5 Results and Discussion

In this section, the results of the linear, cubic, cubic spline, multivariate cubic spline, and makima interpolation methods are examined based on the predictive performance of the NARX ANN model for the mean residence time response variable. Predictions are made on data that have been reserved for testing and are completely untouched by the training of the ANN; interpolation is applied to the training and testing datasets. The model verification and comparison is done against the MLP and recently developed

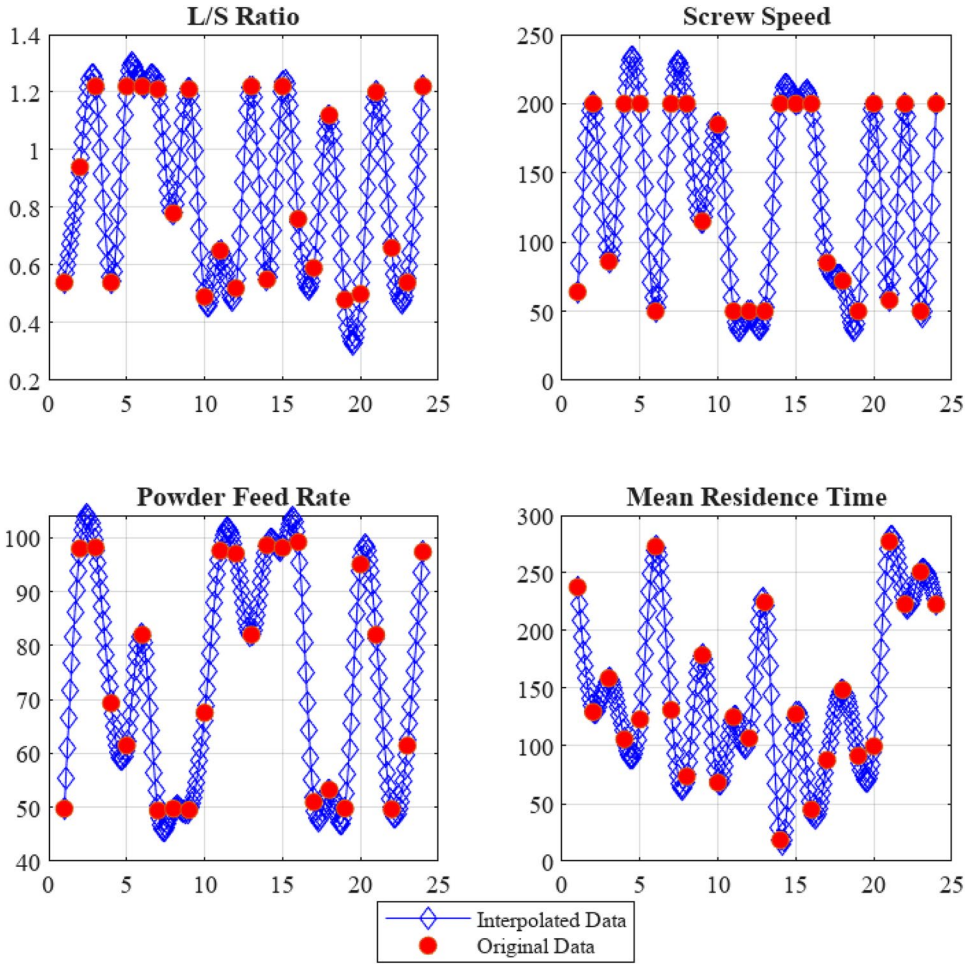


Fig. 5 Interpolation of the individual variables of the dataset which are then used as tensor products to provide a multidimensional interpolation. This interpolation ratio is $\frac{1}{10}$, thus each variable increases from 24 observations to 240

Table 3 Comparison of Interpolation Methods

Feature	Kriging	Linear	Makima	Cubic polynomial	Cubic spline	Multivariate cubic spline
Principle	Spatial correlation	Piecewise linear	Modified Akima: weighted slopes	Cubic polynomial	Smooth cubic polynomial	extends cubic spline to multidimensional data using radial basis functions
Computat-ional cost	High ($O(n^3)$)	Low ($O(n)$)	Moderate ($O(n)$)	Moderate ($O(n)$)	Moderate ($O(n)$)	High ($O(n^2)$)
Grid flexibility	Irregular domains	Regular grids	Regular grids	Regular grids	Regular grids	Irregular domains
Smoothness	Adaptive	None	Moderate (continuous 1st derivative)	Moderate	High (continuous 1st & 2nd derivatives)	high (first and second order partial derivatives)
noise Handling	Robust	Sensitive	Robust	Sensitive	Sensitive	Sensitive
Uncertainty estimates	Yes	No	No	No	No	No

Kriging approach [1]. Performance is measured based on RMSE, MAE, adjusted R^2, and computational cost. The performance of each method was further assessed based on the effects of smaller and larger interpolation ratios (the number of interpolated points per original point, also called interpolants). In addition, an overview of the investigative procedure for selecting and developing the ANN model is outlined.

5.1 Effect of Interpolation on the ANN Model Performance and Individual Parameter Impacts

The use of interpolation significantly improved the ANN predictive modelling ability. Table 4 shows that the ANN did not detect any patterns in the data and failed to produce a model without interpolation. However, after interpolation was applied, the ANN produced a model with an

Table 4 The statistics of the significance of each variable according to multiple linear regression model before interpolation and ANN modelling, after ANN without interpolation, and with both ANN modelling and interpolation. The cubic spline interpolation is used in these results

	p-value		
Variable	Without ANN or interpolation	With ANN and Without interpolation	With ANN and interpolation
L/S ratio (LS)	0.5146	0.2016	0.6130
Screw speed (SS)	0.8950	0.5875	1.3511×10^{-5}
Feed flow rate (FFR)	0.1936	0.3027	3.9549×10^{-5}
SS FFR	0.7096	0.1798	5.9909×10^{-6}
SS LS	0.6274	0.6320	5.5828×10^{-9}
FFR LS	0.1361	0.1232	3.0137×10^{-10}
Adj. R^2 of ANN model	-	0	≈ 0.9

Adj. $R^2 \approx 0.9$, depending on the interpolation ratio and technique, as seen later in Subsect. 5.3. The reason why the ANN is able to accurately model the MRT after applying interpolation can be demonstrated by fitting the dataset to a multiple linear regression model, using the predicted MRT values as the response variable. A multiple linear regression model is useful for providing p-values for each variable which quantify the significance of the variable to the response. Specifically, the p-value measures the strength of the evidence that suggests whether the variable significantly impacts the response variable. A p-value > 0.05 means there is evidence to suggest that the variable significantly impacts the response at the 0.05 level of significance, or a 95% confidence level. The magnitude of the p-value indicates the strength of the evidence; a p-value $= 0.001$ implies moderate to strong evidence, whereas a p-value $= 10^{-6}$ implies very strong evidence.

Table 4 displays the significance of each explanatory variable to the response variabel (MRT). It is evident that the use of interpolation significantly strengthens the models ability to identify the relationship between the input variables and the response MRT. However, the multiple linear regression model alone is not an effective predictive model; therefore, the ANN model should be fitted after interpolation to train the model on a dataset that displays strong relationships between the input and output variables. This leads to a more robust and better-performing ANN predictive model. Furthermore, the product of the two variables seen in Table 4 represents the interaction of the two variables. Therefore, it can be seen that the interaction between all combinations of the input variables has a significant effect on the MRT. These interaction terms describe a physical interaction in the system which will now be explained below.

The interaction between the screw speed and feed flow rate represents the effect of inconsistent feed flow caused by the amount of material in the chamber where the feed gate is located. A higher screw speed will evacuate the material from the input region faster and allow a consistent feed flow rate; however, a slower screw speed will cause build-up and compaction of the material, slowing the feed flow rate and ultimately affecting the MRT. This is presented by the p-value $= 5.9909 \times 10^{-6}$. Second, the interaction between the feed flow rate and L/S ratio is a result of the amount of wetted material that is possible if the feedflow rate varies. The L/S ratio will decrease if the feed flow rate is higher because the liquid will penetrate to a shallow depth, as opposed to a lower feed flow rate that allows more wettability of the powder. Finally, the interaction between the screw speed and L/S ratio is represented by a similar process. A higher or lower screw speed will affect liquid penetration into the dry material which affects the mixture friction and wettability, thus affecting the MRT. These findings are supported by existing results in [61, 62].

5.2 Validation of the Proposed Models with Existing Methods

For this study, NARX and MLP ANN models were designed to study the MRT of powder in the TSG. The specific parameters of the ANN included two hidden layers of size 5, fully connected using *tanh* transfer functions. Two training algorithms, (a) Bayesian regularisation (BR) and (b) Levenberg-Marquardt (LM), were used for the NARX and MLP models. Furthermore, the models were initiated with restrained initial weights to reduce the variability between runs. The model was trained for a maximum of 12,000 iterations, and the data were split into 50% for training and 50% for testing. Training was further divided into training and validation (25% and 25%, respectively). The splitting of the data was complete in block formatting; however, interleaved division was also used for demonstrative purposes, as discussed below. The development of this model required the investigation and exploration of several alternative designs and configurations.

The multilayer perceptron (MLP) faced significant setbacks in detecting patterns and forming reliable predictive models using interpolation ratios ranging from $\frac{1}{2}$ to $\frac{1}{100}$. The ANN architecture was extensively investigated by adjusting the hidden layer sizes, number of hidden layers, layer order, and activation functions. The investigation did not yield a predictive model with an adjusted R^2 above 0.4, and generally showed no sign of pattern recognition, as depicted in Fig. 6. Finally, experimentation with different training algorithms showed that the LM backpropagation and adaptive gradient descent training algorithms did not produce any significant improvement in the models.

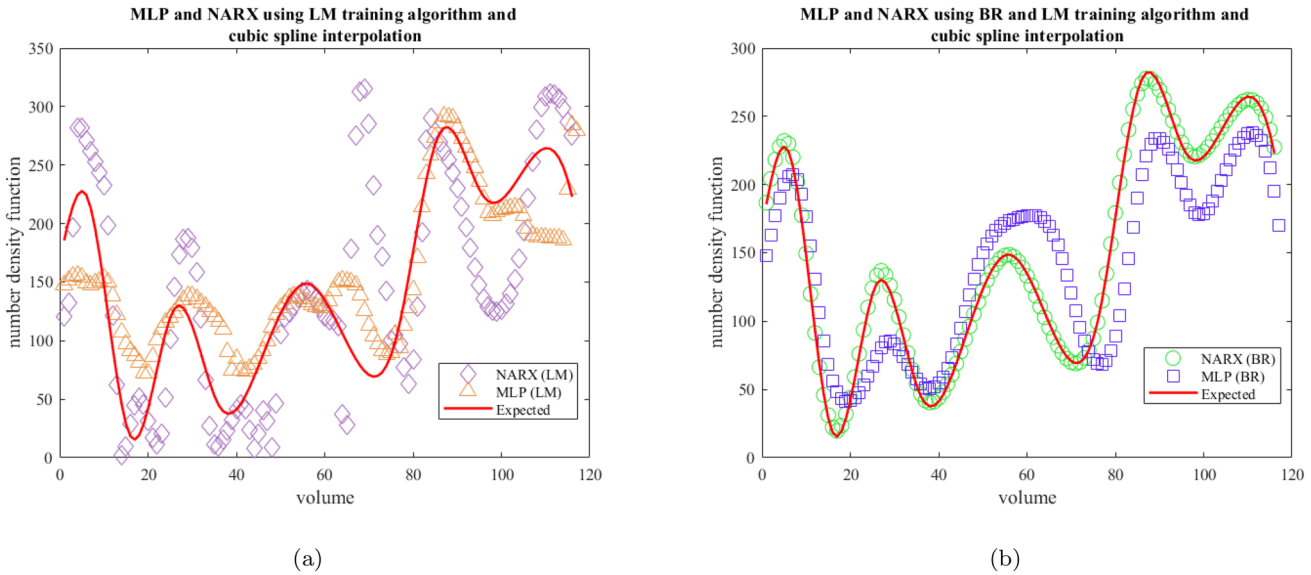


Fig. 6 Comparion of the MLP and NARX model predictions of MRT using univariate cubic spline interpolation with the LM training algorithm (**a**) and the BR training algorithm (**b**). Performance differences are observed with reference to the expected output in red

Table 5 Results from MLP and NARX each with LM or BR training algorithms for comparison between all interpolation methods. The data were acquired with an interpolation ratio of $\frac{1}{10}$. Statistics displayed in this table are the prevailing results and best represented the model performance

Interpolation method	Model	Training algorithm	RMSE	MAE	Standard deviation	Adj. R^2
Cubic spline (univariate)	NARX	BR	2.6	0.02376	0.033936	0.999
	NARX	LM	56.7	0.37369	0.5349	0.49
	MLP	BR	39.5	0.34277	0.47455	0.751
	MLP	LM	39.3	0.26235	0.37089	0.752
Makima (univariate)	NARX	BR	3.48	0.036688	0.060617	0.998
	NARX	LM	38.1	0.55819	0.90643	0.746
	MLP	BR	16.4	0.18886	0.39695	0.953
	MLP	LM	56.7	0.40395	0.56173	0.433
Linear (univariate)	NARX	BR	2.66	0.022736	0.032335	0.999
	NARX	LM	38.0	0.29572	0.46913	0.756
	MLP	BR	13.2	0.15158	0.33653	0.966
	MLP	LM	47.8	0.40039	0.80308	0.557
Cubic (univariate)	NARX	BR	6.3	0.083164	0.15773	0.993
	NARX	LM	52.5	0.36875	0.46822	0.528
	MLP	BR	51.6	0.61022	1.2105	0.544
	MLP	LM	61.3	0.54365	0.87251	0.358
Kriging (multivariate)	NARX	BR	23.9	0.15957	0.29356	0.733
	NARX	LM	24.5	0.16275	0.39188	0.719
	MLP	BR	23.7	0.16691	0.30036	0.737
	MLP	LM	37.4	0.33648	1.3327	0.343
Cubic spline (multivariate)	NARX	BR	2.97	0.022663	0.029876	0.999
	NARX	LM	13.0	0.18563	0.38015	0.972
	MLP	BR	13.7	0.18859	0.38094	0.969
	MLP	LM	31.4	0.27673	0.41839	0.838

In addition, the BR training algorithm for MLP significantly boosted the model performance. This is because BR is robust with small datasets because it penalises large weights to avoid fitting to noise and dynamically updates the regularisation term to prevent overfitting of the training data. The BR training algorithm is available in the *narxnet* ANN coding structure in MATLAB. It is worth mentioning that the NARX model uses 1:1 and 1:2 input delay and feedback delay, respectively, whereas the MLP ANN model uses an input delay of 1:1 and feedback delay of 1:1 (no

Table 1 The general algorithm outline of the MLP ANN architecture

	MLP algorithm
1.	Input layer: Feed input variables into the 3 input nodes with random weights w_{1j}
2.	First transfer function: The tansig activation function is applied to all nodes in the input layer, connecting every node to all nodes in the first hidden layer.
3.	First hidden layer: The first hidden layer has five nodes, all assigned with an initial random weight w_{2j}.
4.	Second transfer function: The values in the first hidden layer are fully connected to the second layer using the tansig activation function:
5.	Second hidden layer: The second hidden layer has five nodes, all assigned with an initial random weight w_{3j}.
6.	Third transfer function: All nodes in the second hidden layer are activated with the tansig activation function, which fully connects the second hidden layer to the output layer.
7.	Output layer: The output layer consists of a single node representing the predicted MRT variable.

Wights (w_{ij}) are updated after every iteration of values passing from the input layer to the output layer and compared to the known values of the training and validation datasets. Weights are updated using the selected training algorithm.

Table 2 The general algorithm outline of the NARX ANN architecture

	NARX algorithm
1.	Input layer: Feed input variables into the three input nodes with random weights w_{1j}.
2.	Input delay: Incorporate a 1:1 input delay.
3.	First transfer function: The tansig activation function is applied to all nodes in the input layer, connecting every node to all nodes in the first hidden layer.
4.	First hidden layer: The first hidden layer has five nodes, all assigned with an initial random weight w_{2j}.
5.	Second transfer function: The values in the first hidden layer are fully connected to the second layer using the tansig activation function:
6.	Second hidden layer: The second hidden layer has five nodes, all assigned with an initial random weight w_{3j}.
7.	**Third transfer function:** All nodes in the second hidden layer are activated with the tansig activation function, which fully connects the second hidden layer to the output layer.
8.	Output layer: The output layer consists of a single node that represents the predicted MRT variable.
9.	Feedback delay: Employ a 1:2 feedback delay.

Wights (w_{ij}) are updated after every iteration of values passing from the input layer to the output layer and compared to the known values of the training and validation datasets. Weights are updated using the selected training algorithm.

delays) [63]. By simplifying the delay terms in NARX to replicate the MLP architecture, we were able to highlight the pivotal role of the BR training algorithm in the success of the ANN model performance. The LSTM model was also investigated briefly; however, further validation of the LSTM model would require developing a custom BR training algorithm in the original LSTM MATLAB code or manually developing an LSTM layer directly into the *narxnet* MLP equivalent. This task is beyond the scope of this study and is therefore left as a future project.

The NARX ANN still demonstrated greater performance than the MLP with the BR training algorithm in terms of estimating the MRT for the TSG, as seen in Fig. 6. The comprehensive comparisons of performance parameters for the NARX and MLP using BR or LM training algorithms is provided in Table 5. It represents the most prevailing results; however, it is important to note that variations between each run of the model can be observed because of the randomisation of initial weight distributions in the ANN. The prevailing results are significantly more common and consistent than the alternative results; therefore, the latter are omitted here. The NARX ANN shows better accuracy because of the feedback delay applied in the NARX ANN, which helps capture temporal dependencies and patterns in the data. With a 1:1 input delay, the model considers only the current inputs to predict the output. Additionally, a 1:2 feedback delay allows the model to use previous predictions to predict the current output. This approach enables the model to capture temporal dependencies and provide a broader context for each prediction while avoiding the inclusion of potentially unnecessary or noisy data from earlier time steps. To capture a complete comparison of the developed NARX and MLP ANNs with integrated interpolation methods, the computational cost must be considered. The MLP and NARX, as mentioned before, follow an almost identical structure except for the feedbackl delay component of the NARX ANN (see Tables 1 and 2). This similarity results in an almost identical computational cost when the number of computations per iteration is considered. However, in actuality, the NARX model reaches an optimal state much sooner than the MLP. The models were instructed to cease training if $\mu \geq 10^{99}$, which is indicative of a model that is no longer improving. This causes the NARX to produce the final output predictions with an efficiency equal to or greater than that of MLP. While in some cases, the NARX and MLP have equal computational costs, NARX is often significantly faster than MLP. The NARX computational cost is of the order of tens to hundreds of seconds faster than the MLP, depending on which method is used and how many interpolants are taken. However, the NARX is never slower than the MLP. This demonstrates the significant advancement of the NARX over the MLP in all aspects of performance. Consequently, the NARX ANN with a 1:1 input delay and a 1:2 feedback delay emerged as the optimal structure and was chosen for modelling MRT to evaluate the various interpolation methods used throughout the remainder of the study.

To show the wide applicability of the NARX ANN models, the results were compared against Kriging interpolation [1]. In a previous study [1], Kriging interpolation

Table 6 The performance results of NARX model with each interpolation tool using interpolation ratio $\frac{1}{21}$

Method	RMSE	MAE	Standard deviation	Adj. R^2	Computational cost (in seconds)	Interpolation ratio	Training iterations
Kriging	4.61	0.089178	0.09028	0.989	640	1/21	7790
Makima	1.74	0.023288	0.050027	0.999	72	1/21	12000
Linear	14.6	0.17566	0.31082	0.959	12	1/21	2129
Cubic	2.97	0.047561	0.10971	0.998	23	1/21	12000
Cubic Spline	2.66	0.071229	0.16111	0.999	47	1/21	12000
Multivariate Cubic Spline	2.37	0.045041	0.12000	0.999	38	1/21	12000

Table 7 The performance results of NARX model with each interpolation tool using interpolation ratio $\frac{1}{15}$

Method	RMSE	MAE	Standard deviation	Adj. R^2	Computational cost (in seconds)	Interpolation ratio	Training iterations
Kriging	28.2	0.19474	0.2893	0.627	13	1/15	1949
Makima	5.18	0.093025	0.19451	0.995	36	1/15	12000
Linear	11.5	0.15932	0.35742	0.974	29	1/15	8401
Cubic	7.93	0.09074	0.13181	0.989	34	1/15	10698
Cubic Spline	5.22	0.041625	0.052618	0.996	142	1/15	12000
Multivariate Cubic Spline	1.34	0.028877	0.060293	0.999	32	1/15	12000

Table 8 The performance results of NARX model with each interpolation tool using interpolation ratio $\frac{1}{10}$

Method	RMSE	MAE	Standard deviation	Adj. R^2	Computational cost (in seconds)	Interpolation ratio	Training iterations
Kriging	28.3	0.20869	0.28924	0.623	6	1/10	1939
Makima	7.76	0.090741	0.17208	0.990	32	1/10	12000
Linear	15.1	0.25227	0.47242	0.957	14	1/10	3622
Cubic	6.3	0.083164	0.15773	0.993	17	1/10	3608
Cubic Spline	12.4	0.12877	0.23866	0.976	37	1/10	12000
Multivariate Cubic Spline	2.97	0.022663	0.029876	0.999	29	1/10	12000

was performed to produce 20 interpolants per observation. The interpolated dataset was then fed into an ANN model to predict the mean residence time, with an Adj. R^2 of 0.9242 and an RMSE of 13.9138. However, in this study, an advanced NARX ANN model was developed to process the same dataset. The data were interpolated similarly with a ratio of $\frac{1}{21}$ and fed into the NARX ANN. The resulting performance increased to an Adj. R^2 of 0.989 and an RMSE of 4.61. That is, Adj. R^2 increased by 6.48% and RMSE decreased by 66.9%. These results demonstrate the superiority of the proposed NARX ANN model for the predicted MRT of twin-screw granulators.

5.3 Validation of the Proposed Models with Different Interpolation Techniques

In this section, the NARX ANN model is integrated with various interpolation techniques and evaluated quantitatively to assess its accuracy and efficiency. The previous section demonstrated that the NARX model performed significantly better than the MLP model. Therefore, only the results obtained using the NARX model corresponding to different interpolations are discussed here. By interpolating the data using different methods and feeding the interpolated data into the NARX ANN model, the performance parameters demonstrate the effectiveness of the data interpolation. These results are summarised in Table 6 when using an interpolation ratio of $\frac{1}{21}$ similar to the existing Kriging approach [1].

The robustness of each interpolation tool was examined by assessing the respective performance of the NARX ANN model when applying various interpolation ratios. In Tables 7 and 8, the performance of the NARX ANN model is presented for each interpolation tool using interpolation ratios $\frac{1}{15}$ and $\frac{1}{10}$, respectively. It is clear that the model performance suffers significantly when using the Kriging method with fewer interpolants. This is likely due to the high dimensionality of the Kriging method and the structure of the interpolated data. Specifically, the Kriging interpolants are not structured in the same way as the alternative methods (linear interpolation, cubic interpolation, etc.), as indicated by the output curve. Kriging can produce interpolants

across an entire domain, giving rise to interpolated points that alternative methods are unable to capture. This clearly requires far greater computational cost for the NARX ANN model to effectively model the interpolated data using Kriging interpolation.

The alternative methods present good performance for interpolation ratios as low as $\frac{1}{10}$, and some methods are capable of good performance with as low as five interpolated points per observation. This demonstrates the robust performance of the alternative methods for interpolating the MRT for implementation into the NARX ANN predictive model. In Table 9, the RMSE and Adj. R^2 show that the cubic polynomial and cubic spline are the most successful interpolation methods for fewer interpolants. The Adj. R^2 of 0.953 and 0.952 and RMSE of 16.7 and 17.3 for the cubic polynomial and cubic spline techniques, respectively, show that both interpolation methods describe the MRT variable with a strong relation. Furthermore, the multivariate cubic spline demonstrates even higher explainability with Adj. $R^2 = 0.999$ and RMSE $= 2.97$, which is 31% less than the univariate cubic polynomial. This demonstrates the impact of the multidimensional approach on this dataset.

The cubic polynomial, cubic spline, and multivariate cubic spline interpolation methods demonstrated superior performance with fewer interpolants, providing a dataset that enabled the NARX ANN to accurately predict future MRT values. In contrast, the linear and Makima interpolation methods show weaker predictability, with only five interpolants per observation, as detailed in Table 9. The Kriging method performs poorly, achieving an adjusted Adj. R^2 of 0.627 with 15 interpolants per observation, compared to over 0.973 for the alternative methods (Table 7). However, with an interpolation ratio of $\frac{1}{21}$, Kriging improves dramatically, achieving an Adj. R^2 of 0.989. Still, the cubic polynomial, cubic spline, multivariate cubic spline, and Makima methods outperform Kriging, achieving Adj. R^2 values of 0.999, 0.998, 0.999, and 0.999, respectively, for the same interpolation ratio (Table 6).

Furthermore, the Kriging interpolation method has a high computational cost for computing all results compared to the alternative methods. All methods were implemented in the same NARX ANN model with the same training parameters; specifically, if $\mu > 10^{99}$, then the model would stop training, as $\mu > 10^{99}$ indicates that the model will no longer improve from further training iterations. This leads to a variation in the number of training iterations, as shown in Tables 6, 7, 8, 9. Irrespective, the computational cost of the Kriging method is much higher than that of the alternative methods, requiring over 600s to train just 7790 iterations. In contrast, the cubic method could train the full 12000 iterations in just 23s. The Kriging method, originally designed for spatial data interpolation, uses coarse datasets to estimate values within a spatial domain. When applied to nonspatial data, it functions similarly by treating inputs as spatial variables to interpolate nonspatial parameters. However, interpolating irrelevant or unrealistic data can lead to unnecessary computations, increasing the computational cost of the model.

The multivariate cubic spline interpolation method surpasses the alternative approaches for different interpolation ratios. Its ability to capture the multidimensional behaviour of the dataset provides a robust foundation for training the NARX ANN predictive model. With an Adj. R^2 of 0.978 using only five interpolants per observation, it demonstrates exceptional accuracy. Although its computational cost is higher than that of univariate methods, it remains lower than that of the Kriging method. Consequently, the multivariate cubic spline is the optimal interpolation method for this dataset, delivering the most accurate MRT predictions and the strongest input-output relationships for the NARX ANN model. The univariate cubic spline interpolation, on the other hand, performs slightly below the multivariate version. Furthermore, the difference between the multivariate and univariate approaches of the cubic spline output is highlighted in Fig. 7.

The results in Table 6 are associated to Figs. 8a, 9, 10b, which highlight the different characteristic curves for each interpolation technique. The difference in the curve shape of each method significantly impacts the performance of the model, thus reinforcing the importance of understanding the dataset before selecting an interpolation technique. This has been found in previous studies, such as Gnauck [64], which have discussed the importance of selecting an interpolation

Table 9 The performance results of NARX model with each interpolation tool using interpolation ratio $\frac{1}{5}$

Method	RMSE	MAE	Standard deviation	Adj. R^2	Computational cost (in seconds)	Interpolation ratio	Training iterations
Kriging	41.4	0.74396	1.4324	0.13	5	1/5	1322
Makima	37.4	0.38174	0.59069	0.753	8	1/5	992
Linear	29.9	0.30973	0.55405	0.828	6	1/5	359
Cubic	16.7	0.21159	0.37881	0.953	47	1/5	12000
Cubic Spline	17.3	0.18932	0.27737	0.952	15	1/5	1166
Multivariate Cubic Spline	11.6	0.096563	0.12719	0.978	29	1/5	433

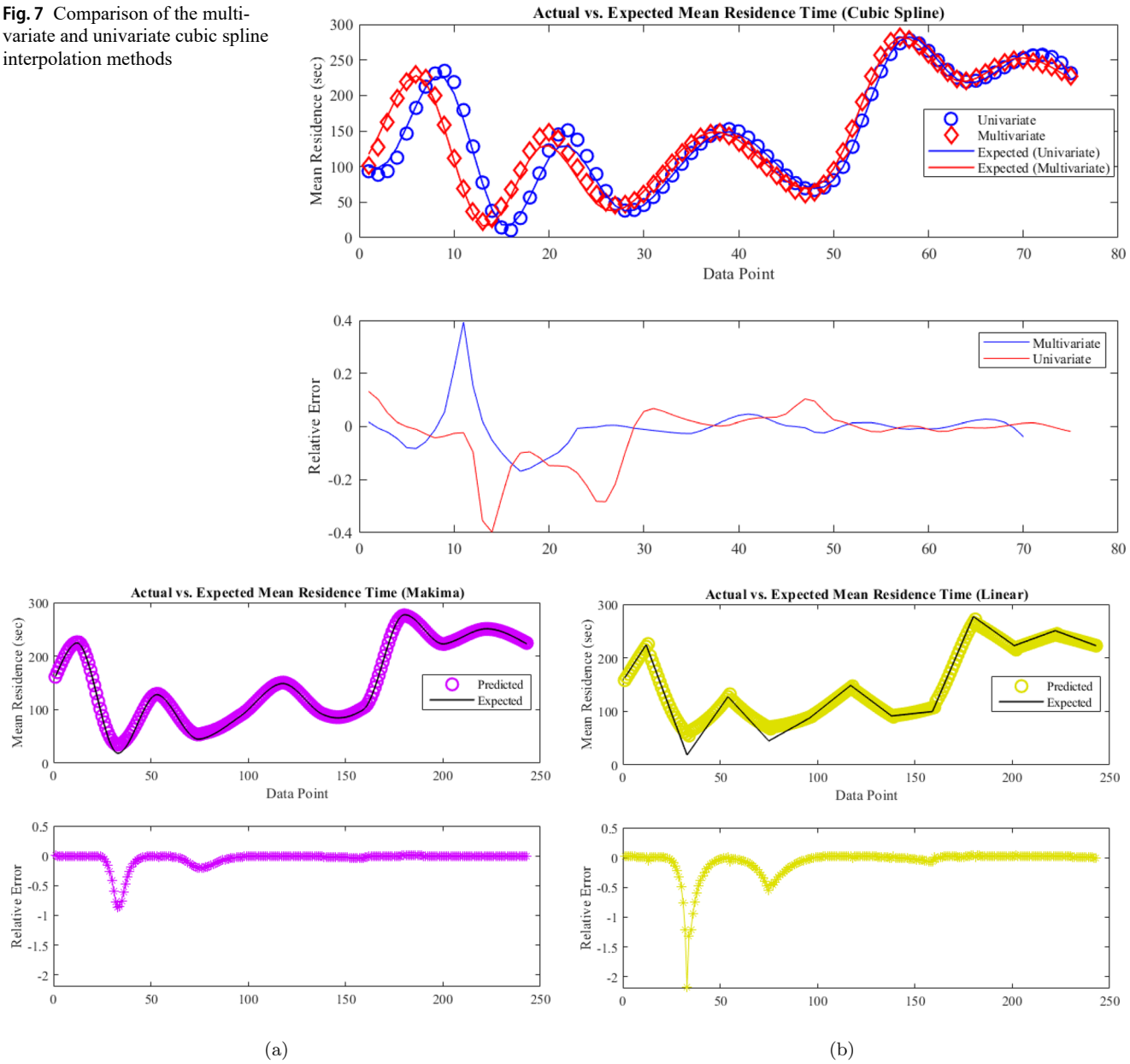


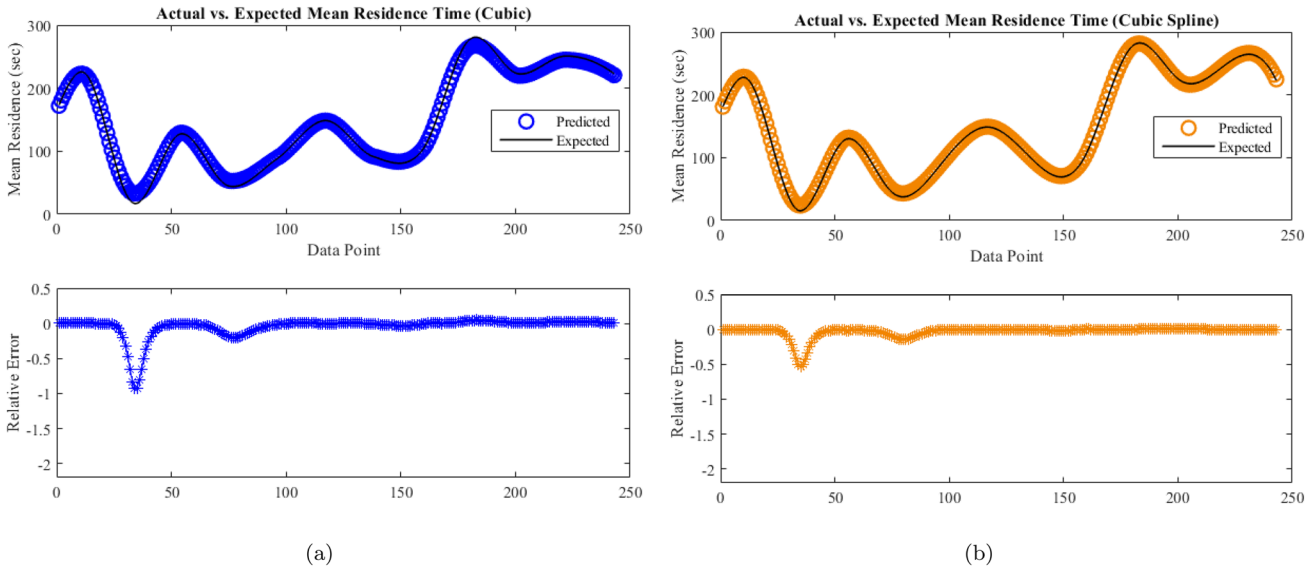
Fig. 7 Comparison of the multivariate and univariate cubic spline interpolation methods

Fig. 8 **a**: The predicted MRT curve and resulting error using the NARX model coupled with Makima interpolation. **b**: The predicted MRT curve and resulting error using the NARX model coupled with linear interpolation

method that is suitable and appropriate for specific applications and dataset characteristics. This supports the findings of this paper as the multivariate cubic spline is preserving the time-series characteristics of the original data and including the multidimensionality into the interpolation process.

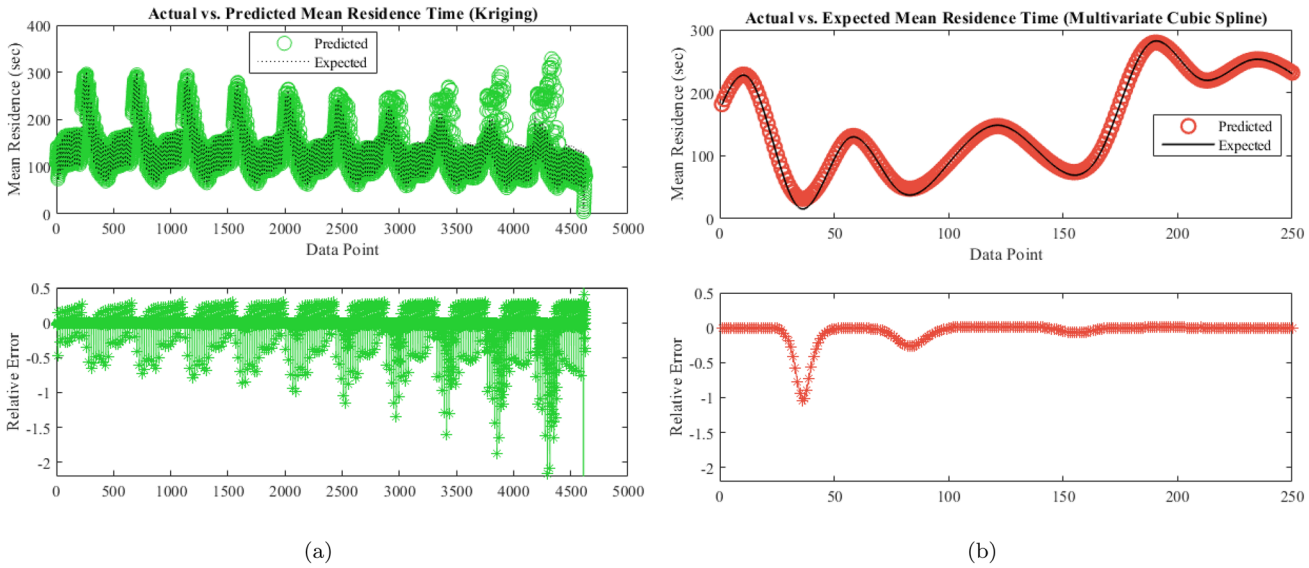
6 Conclusions, Suggestions and Future Aspects of the Study

This paper provides an overview of the TSG process and an ANN designed to model the MRT of granules in a TSG based on three input variables: the L/S ratio, powder flow rate, and screw speed. This study focused on using various univariate and multivariate interpolation methods to enhance functionality and model performance when using a dataset containing only 96 original data points (4×24 measurements). The results indicated that the cubic spline interpolation method



(a) (b)

Fig. 9 **a**: The predicted MRT curve and resulting error using the NARX model coupled with cubic polynomial interpolation. **b**: The predicted MRT curve and resulting error using the NARX model coupled with univariate cubic spline interpolation



(a) (b)

Fig. 10 **a**: The predicted MRT curve and resulting error using the NARX model coupled with multivariate Kriging interpolation. **b**: The predicted MRT curve and resulting error using the NARX model coupled with multivariate cubic spline interpolation

performed best among the four univariate methods, with minimal computational cost and error for the interpolation ratio of $\frac{1}{5}$. However, with more interpolants, the Makima and cubic univariate interpolation methods surpassed the cubic spline interpolation method, thereby demonstrating superior performance. In addition to the univariate methods, two multivariate interpolation methods were investigated. The Kriging and multivariate cubic spline methods yielded contrasting results, and the multivariate cubic spline methods exhibited superior computational efficiency compared to the Kriging method.

The Kriging method required at least 21 interpolants per observation to provide a dataset that could successfully train the ANN model. Conversely, the multivariate cubic spline method coupled with the ANN successfully modelled and predicted the MRT using an interpolation ratio of $\frac{1}{5}$ (5 interpolants). Furthermore, the multivariate cubic spline interpolation method surpassed the Kriging and univariate methods on all performance parameters except computational cost (slightly different). The multivariate cubic spline interpolation method with an interpolation ratio of $\frac{1}{5}$ achieved an RMSE of 11.6 and an adjusted R^2 of 0.978, representing

a 72% decrease in the RMSE and an 85% increase in the adjusted R^2 compared with the Kriging method.

Therefore, the multivariate cubic spline interpolation method demonstrates the greatest success for preprocessing the coarse TSG dataset, as evidenced by the resulting performance of the NARX ANN predictive model. The NARX ANN coupled with mutivariate cubic spline interpolation and BR training algorithm is the superior model configuration. This finding underscores the potential of combining advanced interpolation techniques with machine learning models to effectively analyse and predict complex processes in pharmaceutical manufacturing, contributing to the production of high-quality and safe products.

6.1 Future Work

Further analysis and assessment of hybrid ML interpolation models should be conducted using new experimental datasets that target the interpolants of the model. This will provide a robust and comprehensive assessment of the model performance and further validate the results. The model in this study can be further improved by establishing an effective way for MRT predictions to transition from interpolated to non-interpolated data. Numerous deep learning ANN approaches, aside from the MLP and NARX considered in this study, have gained considerable traction in many fields. The adoption of deep learning methods can significantly enhance the predictive ability of the coupled ML and interpolation model when using small datasets. Furthermore, the coupling of physics principles with this model would enhance the interpretability and predictive abilities of the model. These factors should be considered before implementing the model as an online MRT monitoring system.

Acknowledgements We would like to extend our gratitude to the Mathematics Applications Consortium for Science and Industry (MACSI) and the Pharmaceutical Manufacturing Technology Centre (PMTC) at the University of Limerick for generously providing the resources that made this research possible. I would also like to express our gratitude to the reviewers, whose suggestions helped us enhance the quality of the article.

Author Contributions MR: Conceptualization; Formal analysis; Investigation; Methodology; MATLAB coding; Validation; Writing—original draft; Writing—review & editing VVR: Formal analysis; Investigation; Methodology; Validation;Writing—review & editing GW: Formal analysis; Investigation; Methodology; Validation;Writing—review & editing SH: Formal analysis; Investigation; Methodology; Validation;Writing—review & editing RR: Formal analysis; Investigation; Methodology; Validation;Writing - review & editing MS: Conceptualization; Formal analysis; Investigation; Methodology; Supervision; Validation; Writing—original draft; Writing - review & editing

Funding Open Access funding provided by the IReL Consortium

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The author whose name is listed immediately certify that he has no affiliations with or involvement in any organization or entity with any financial interest.

References

1. Ismail HY, Singh M, Darwish S, Kuhs M, Shirazian S, Croker DM, Khraisheh M, Albadarin AB, Walker GM (2019) Developing ANN-Kriging hybrid model based on process parameters for prediction of mean residence time distribution in twin-screw wet granulation. Powder Technol 343:568–577. https://doi.org/10.1016/j.powtec.2018.11.060
2. Zheng C, Zhang L, Govender N, Chuan-Yu W (2021) DEM analysis of residence time distribution during twin screw granulation. Powder Technol 377:924–938. https://doi.org/10.1016/j.powtec.2020.09.049 (https://www.sciencedirect.com/science/article/pii/S003259102030913X)
3. Seem TC, Rowson NA, Ingram A, Huang Z, Shen Yu, de Matas M, Gabbott I, Reynolds GK (2015) Twin screw granulation - A literature review. Powder Technol 276:89–102. https://doi.org/10.1016/j.powtec.2015.01.075 (https://www.sciencedirect.com/science/article/pii/S0032591015001023)
4. Ismail HY, Shirazian S, Singh M, Whitaker D, Albadarin AB, Walker GM (2020) Compartmental approach for modelling twin-screw granulation using population balances. Int J Pharm 576:118737. https://doi.org/10.1016/j.ijpharm.2019.118737. (https://www.sciencedirect.com/science/article/pii/S0378517319307823)
5. Ismail HY, Singh M, Albadarin AB, Walker GM (2020) Complete two dimensional population balance modelling of wet granulation in twin screw. Int J Pharm 591:120018. https://doi.org/10.1016/j.ijpharm.2020.120018. https://www.sciencedirect.com/science/article/pii/S0378517320310036
6. Ismail HY, Singh M, Shirazian S, Albadarin AB, Walker GM (2020) Development of high-performance hybrid ann-finite volume scheme (ann-fvs) for simulation of pharmaceutical continuous granulation. Chem Eng Res Des 163:320–326
7. Sampat C, Bettencourt F, Baranwal Y, Paraskevakos I, Chaturbedi A, Karkala S, Jha S, Ramachandran R, Ierapetritou M (2018) A parallel unidirectional coupled DEM-PBM model for the efficient simulation of computationally intensive particulate process systems. Comput Chem Eng 119:128–142. https://doi.org/10.1016/j.compchemeng.2018.08.006. (https://www.sciencedirect.com/science/article/pii/S0098135418303594)

8. Metta N, Ierapetritou M, Ramachandran R (2018) A multiscale DEM-PBM approach for a continuous comilling process using a mechanistically developed breakage kernel. Chem Eng Sci 178:211–221. https://doi.org/10.1016/j.ces.2017.12.016
9. AlAlaween WH, Khorsheed B, Mahfouf M, Reynolds GK, Salman AD (2020) An interpretable fuzzy logic based data-driven model for the twin screw granulation process. Powder Technol 364:135–144. https://doi.org/10.1016/j.powtec.2020.01.052 (https://www.sciencedirect.com/science/article/pii/S0032591020300644)
10. Sampat C, Ramachandran R (2022) Physics-constrained autoencoder neural network for the prediction of key granule properties in a twin-screw granulation process. In Yoshiyuki Yamashita and Manabu Kano, editors, computer aided chemical engineering, volume 49 of 14 international symposium on process systems engineering, pages 1687–1692. Elsevier, January. https://doi.org/10.1016/B978-0-323-85159-6.50281-5. URL https://www.sciencedirect.com/science/article/pii/B9780323851596502815
11. Wang M, Krishna Kumar YT, Feng TQ, Wang M (2024) Machine learning aided modeling of granular materials: a review. archives of computational methods in engineering 1–38
12. Pant R, Singh R, Gehlot A, Akram SV, Gupta LR, Thakur AK (2024) A systematic review of additive manufacturing solutions using machine learning, internet of things, big data, digital twins and blockchain technologies: a technological perspective towards sustainability. archives of computational methods in engineering 1–16
13. Shirazian S, Ismail HY, Singh M, Shaikh R, Croker DM, Walker GM (2019) Multi-dimensional population balance modelling of pharmaceutical formulations for continuous twin-screw wet granulation: Determination of liquid distribution. Int J Pharm 566:352–360 https://www.sciencedirect.com/science/article/pii/S0378517319304442. Publisher: Elsevier
14. Kotamarthy L, Dan A, Karkala S, Parvani S, Román-Ospino AD, Ramachandran R (2023) Twin-screw granulation: Mechanistic understanding of the effect of material properties on key granule quality attributes through the analysis of mixing dynamics and granulation rate mechanisms. Adv Powder Technol 34(9):104137. https://doi.org/10.1016/j.apt.2023.104137. (https://www.sciencedirect.com/science/article/pii/S0921883123002029)
15. Arthur TB, Rahmanian N (2024) Process Simulation of Twin-Screw Granulation: A Review. Pharmaceutics 16(6):706. https://doi.org/10.3390/pharmaceutics16060706https://www.mdpi.com/1999-4923/16/6/706. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute
16. Singh M, Shirazian S, Ranade V, Walker GM, Kumar A (2022) Challenges and opportunities in modelling wet granulation in pharmaceutical industry - A critical review. Powder Technol 403:117380. https://doi.org/10.1016/j.powtec.2022.117380 (https://www.sciencedirect.com/science/article/pii/S0032591022002741)
17. Panwar A, Shirazian S, Singh M, Walker GM (2021) Comprehensive modelling of pharmaceutical solvation energy in different solvents. J Mol Liq 341:117390
18. Rather IH, Kumar S, Gandomi AH (2024) Breaking the data barrier: a review of deep learning techniques for democratizing ai with small datasets. Artif Intell Rev 57(9):226
19. Ishikawa T, Amano T, Kihara S-I, Funatsu K (2002) Flow patterns and mixing mechanisms in the screw mixing element of a co-rotating twin-screw extruder. Polym Eng Sci 42(5):925–939. https://doi.org/10.1002/pen.11002
20. Singh M, Ranade V, Shardt O, Matsoukas T (2022) Challenges and opportunities concerning numerical solutions for population balances: a critical review. J Phys A Math Theor 55(38):383002
21. Yadav S, Singh M, Singh S, Heinrich S, Kumar J (2024) Modified variational iteration method and its convergence analysis for solving nonlinear aggregation population balance equation. Comput Fluids 274:106233. https://doi.org/10.1016/j.compfluid.2024.106233 (https://www.sciencedirect.com/science/article/pii/S0045793024000653)
22. Yadav S, Das A, Singh S, Tomar S, Singh R, Singh M (2024) Coupled approach and its convergence analysis for aggregation and breakage models: Study of extended temporal behaviour. Powder Technol 439:119714. https://doi.org/10.1016/j.powtec.2024.119714 (https://www.sciencedirect.com/science/article/pii/S0032591024003565)
23. Yadav N, Singh M, Singh S, Singh R, Kumar J (2023) A note on homotopy perturbation approach for nonlinear coagulation equation to improve series solutions for longer times. Chaos Solitons Fractals 173:113628. https://doi.org/10.1016/j.chaos.2023.113628 (https://linkinghub.elsevier.com/retrieve/pii/S0960077923005295)
24. Yadav S, Keshav S, Singh S, Singh M, Kumar J (2023) Homotopy analysis method and its convergence analysis for a nonlinear simultaneous aggregation-fragmentation model. Chaos Solitons Fractals 177:114204. https://doi.org/10.1016/j.chaos.2023.114204 (https://linkinghub.elsevier.com/retrieve/pii/S0960077923011062)
25. Yadav N, Singh M, Singh S, Singh R, Kumar J, Heinrich S (2024) An efficient approach to obtain analytical solution of nonlinear particle aggregation equation for longer time domains. Adv Powder Technol 35(3):104370. https://doi.org/10.1016/j.apt.2024.104370 (https://www.sciencedirect.com/science/article/pii/S0921883124000463)
26. Sriwastav N, Das A, Shardt O, Kumar J, Singh M (2025) A meshfree approach for the rennet-induced coagulation equation: Spline based multistage bernstein collocation method and its convergence analysis. Appl Math Model 143:116035
27. Keshav S, Singh S, Huang Y, Kumar J, Singh M (2024) Explicit and approximate solutions for the fragmentation equation in the presence of source and efflux terms: a coupled meshfree approach and its convergence analysis. Kinetic and Related Models 18:520–540
28. Smoluchowski MV (1916) Drei Vortrage uber Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen. Zeitschrift fur Physik, 17: 557–585, January. URL https://ui.adsabs.harvard.edu/abs/1916ZPhy...17..557S. Publisher: Springer ADS Bibcode: 1916ZPhy...17..557S
29. Singh M, Kaur G, De Beer T, Nopens I (2018) Solution of bivariate aggregation population balance equation: a comparative study. React Kinet Mech Catal 123(2):385–401
30. Singh M, Kumar A, Shirazian S, Ranade V, Walker G (2020) Characterization of simultaneous evolution of size and composition distributions using generalized aggregation population balance equation. Pharmaceutics 12(12):1152
31. Mehakpreet Singh G, Kaur JK, De Beer T, Nopens I (2018) A comparative study of numerical approximations for solving the smoluchowski coagulation equation. Braz J Chem Eng 35(4):1343–1354
32. Singh M, Sriwastav N, Shardt O (2024) Efficient mass-preserving finite volume approach for the rennet-induced coagulation equation. Chaos Solitons Fractals 181:114692
33. Ansari Z, Rae M, Kumar J, Singh M (2024) Optimizing numerical performance of enzymatic coagulation models: Insight into proteolysis and gelation dynamics. Phys Fluids 36(11)
34. Ansari Z, Rae M, Singh M (2024) Two moments preserving sectional approach for an enzymatic coagulation equation. Phys Fluids 36(6)
35. Wangersky PJ (1978) Lotka-Volterra Population Models. Annu Rev Ecol Syst 9:189–218 https://www.jstor.org/stable/2096748. Publisher: Annual Reviews

36. Bacaër N (2011) Verhulst and the logistic equation (1838). In Nicolas Bacaër, editor, A Short History of Mathematical Population Dynamics, pages 35–39. Springer, London. ISBN 978-0-85729-115-8. https://doi.org/10.1007/978-0-85729-115-8_6
37. Dai Y, Yang C, Zhixiang G, Yao Y, Liu Y (2025) Hybrid factors latent gaussian process modeling with wasserstein distance for soft sensing of extruder processes. Chemom Intell Lab Syst 105387
38. Jia M, Jiang L, Guo B, Liu Y, Chen T (2025) Physical-anchored graph learning for process key indicator prediction. Control Eng Pract 154:106167
39. Gupta N (2013) Artificial neural network. Netw Complex Syst 3(1):24 (https://iiste.org/Journals/index.php/NCS/article/view/6063)
40. Shen C (2018) A transdisciplinary review of deep learning research and its relevance for water resources scientists. Water Resour Res 54(11):8558–8593. https://doi.org/10.1029/2018WR022643
41. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2016) Feature selection for high-dimensional data. Prog Artif Intell 5(2):65–75. https://doi.org/10.1007/s13748-015-0080-y (https://doi.org/10.1007/s13748-015-0080-y)
42. Nassif AB, Talib MA, Nasir Q, Dakalbab FM (2021) Machine learning for anomaly detection: a systematic review. IEEE Access 9:78658–78700. https://doi.org/10.1109/ACCESS.2021.3083060 https://ieeexplore.ieee.org/abstract/document/9439459. Conference Name: IEEE Access
43. Zhang Z, Beck MW, Winkler DA, Huang B, Sibanda W, Goyal H (2018) Opening the black box of neural networks: methods for interpreting neural network models in clinical applications. Ann Transl Med 6(11):216. https://doi.org/10.21037/atm.2018.05.32 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6035992/)
44. Savage N (March 2022) Breaking into the black box of artificial intelligence. Nature
45. Castelvecchi D (2016) Can we open the black box of AI? Nature News 538(7623):20
46. Lederer J (2021) Activation functions in artificial neural networks: a systematic overview, January . URL http://arxiv.org/abs/2101.09957. arXiv:2101.09957 [cs] version: 1
47. Sharma S, Sharma S, Athaiya A (2020) Activation functions in neural networks. IJEAST 04(12):310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054 (https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf)
48. Levenberg K (1944) A Method for the Solution of Certain Non-Linear Problems in Least Squares. Q Appl Math 2(2):164–168 https://www.jstor.org/stable/43633451. Publisher: Brown University
49. Marquardt DW (1963) An Algorithm for Least-Squares Estimation of Nonlinear Parameters. J Soc Ind Appl Math 11(2):431–441. https://doi.org/10.1137/0111030 URL https://epubs.siam.org/doi/10.1137/0111030. Publisher: Society for Industrial and Applied Mathematics
50. Dan Foresee F, Hagan MT (1997) Gauss-Newton approximation to Bayesian learning. Proceedings of International Conference on Neural Networks (ICNN'97) 3:1930–1935. https://doi.org/10.1109/ICNN.1997.614194. URL http://ieeexplore.ieee.org/document/614194/. Conference Name: International Conference on Neural Networks (ICNN'97) ISBN: 9780780341227 Place: Houston, TX, USA Publisher: IEEE
51. MacKay DJC (1992) Bayesian interpolation. Neural Comput 4(3):415–447. https://doi.org/10.1162/neco.1992.4.3.415 (https://doi.org/10.1162/neco.1992.4.3.415)
52. Magris M, Iosifidis A (2023) Bayesian learning for neural networks: an algorithmic survey. Artif Intell Rev 56(10):11773–11823
53. Sampat C, Ramachandran R (2024) Optimizing energy efficiency of a twin-screw granulation process in real-time using a long short-term memory (LSTM) network. ACS Eng Au 4(2):278–289
54. Di Piazza A, Piazza MCD, Vitale G (2016) Solar and wind forecasting by NARX neural networks. Renew Energy Environ Sustain 1:39. https://doi.org/10.1051/rees/2016047 URL https://www.rees-journal.org/articles/rees/abs/2016/01/rees160047-s/rees160047-s.html. Publisher: EDP Sciences
55. Kahani M, Ahmadi MH, Tatar A, Sadeghzadeh M (2018) Development of multilayer perceptron artificial neural network (MLP-ANN) and least square support vector machine (LSSVM) models to predict Nusselt number and pressure drop of TiO2/water nanofluid flows through non-straight pathways. Numer Heat Trans Part A Appl 74(4):1190–1206. https://doi.org/10.1080/10407782.2018.1523597 Publisher: Taylor & Francis
56. Dan A, Liu B, Patil U, Manuraj BNM, Gandhi R, Buchel J, Chundawat SPS, Guo W, Ramachandran R (2025) Machine learning model-based design and model predictive control of a bioreactor for the improved production of mammalian cell-based bio-therapeutics. Control Eng Pract 156:106198
57. Ding Q, Yiren Wang Yu, Zheng FW, Zhou S, Pan D, Xiong Y, Zhang Y (2024) Subsurface geological profile interpolation using a fractional kriging method enhanced by random forest regression. Fract Fract 8(12):717. https://doi.org/10.3390/fractalfract8120717 URL https://www.mdpi.com/2504-3110/8/12/717. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute
58. Webster R, Oliver M (2001) Geostatistics for environmental scientist, 2nd. Statistics in Practice, January
59. Dan EL, Dînşoreanu M, Mureşan RC (2020) Accuracy of six interpolation methods applied on pupil diameter data. In 2020 IEEE international conference on automation, quality and testing, robotics (AQTR), pages 1–5, May. https://doi.org/10.1109/AQTR49680.2020.9129915. URL https://ieeexplore.ieee.org/abstract/document/9129915
60. Pal J, Chakrabarty D (2023) Infilling of missing data in groundwater pollution prediction models using statistical methods. Hydrol Sci J 68(15):2208–2222
61. Van Snick B, Kumar A, Verstraeten M, Pandelaere K, Di Jens Dhondt G, Pretoro TD, Beer CV, Vanhoorne V (2019) Impact of material properties and process variables on the residence time distribution in twin screw feeding equipment. Int J Pharm 556:200–216
62. Kumar A, Alakarjula M, Vanhoorne V, Toiviainen M, De Leersnyder F, Vercruysse J, Juuti M, Ketolainen J, Vervaet C, Remon JP et al (2016) Linking granulation performance with residence time and granulation liquid distributions in twin-screw granulation. Exp Investig Eur J Pharm Sci 90:25–37
63. Bhushan B, Singh M, Hage Y (2012) Identification and control using mlp, elman, narxsp and radial basis function networks: a comparative analysis. Artif Intell Rev 37:133–156
64. Gnauck A (2004) Interpolation and approximation of water quality time series and process identification. Anal Bioanal Chem 380:484–492