Predictive modeling of die filling of the pharmaceutical granules using the flexible neural tree

Article

Accepted Version

Ojha, V. K. ORCID: https://orcid.org/0000-0002-9256-1192, Schiano, S., Wu, C.-Y., Snásel, V. and Abraham, A. (2018)

It is advisable to refer to the publisher's version if you intend to cite from the work. See Guidance on citing.

To link to this article DOI: http://dx.doi.org/10.1007/s00521-016-2545-8

www.reading.ac.uk/centaur

Predictive Modeling of Die Filling of the Pharmaceutical Granules Using the Flexible Neural Tree

Varun Kumar Ojha[*1], Serena Schiano[2], Chuan-Yu Wu[2], Václav Snášel[1], and Ajith Abraham[3]

[1]*IT4Innovations, VŠB-Technical University of Ostrava, Ostrava, Czech Republic*
[2]*Dept. of Chemical and Process Engineering, University of Surrey, Guildford, GU2 7XH, UK*
[3]*Machine Intelligence Research Labs (MIR Labs), Auburn, WA, USA*

Abstract

In this work, a computational intelligence (CI) technique named flexible neural tree (FNT) was developed to predict die filling performance of pharmaceutical granules and to identify significant die filling process variables. FNT resembles feedforward neural network, which creates a tree-like structure by using genetic programming. To improve accuracy, FNT parameters were optimized by using differential evolution algorithm. The performance of the FNT-based CI model was evaluated and compared with other CI techniques: multilayer perceptron, Gaussian process regression, and reduced error pruning tree. The accuracy of the CI model was evaluated experimentally using die filling as a case study. The die filling experiments were performed using a model shoe system and three different grades of microcrystalline cellulose (MCC) powders (MCC PH 101, MCC PH 102, and MCC DG). The feed powders were roll-compacted and milled into granules. The granules were then sieved into samples of various size classes. The mass of granules deposited into the die at different shoe speeds was measured. From these experiments, a dataset consisting true density, mean diameter (d50), granule size, and shoe speed as the inputs and the deposited mass as the output was generated. Cross-validation (CV) methods such as 10FCV and 5x2FCV were applied to develop and to validate the predictive models. It was found that the FNT based CI model (in the cases of both CV methods) performed much better than other CI models. Additionally, it was observed that process variables such as the granule size and the shoe speed had a higher impact on the predictability than that of the powder property such as d50. Furthermore, validation of model prediction with experimental data showed that the die filling behavior of coarse

---

[*]Corresponding author

granules could be better predicted than that of fine granules.

1   Introduction

In pharmaceutical industries, it is well recognized that the variation in composition and the quality of tablets are determined by material properties and process conditions. One of the greatest challenges in pharmaceutical development is to identify the causal relationship between material properties, intermediate properties, final product properties, and process variables, which is crucial to obtain high-quality products. However, it is a challenging multifactorial problem.

Tablets are manufactured by compressing dry powders or granules in a die, i.e., the so-called die compaction. The die compaction process consists of three primary stages: die filling, compaction, and ejection [1]. Simple gravity effect can play a role during die filling of powders into the die. However, flow behavior during die filling process controls the tablet composition, the tablet properties as well as its segregation tendency [2–4]. Therefore, the study of die filling process parameters has a significant role in controlling tablet (drug) manufacturing industry.

Previous studies on die filling have been focused on such factors influencing the flow behavior as powder characteristics, apparatus features, and operating variables. For instance, the effect of density and particle size on gravity and suction filling of several grades of microcrystalline cellulose powders were explored in [5]. It was shown that fine particles present intermittent flow behavior due to their cohesive property, while larger particles are free flowing. The suction effect during die filling on a rotary press system was studied in [6], where the strong influence on the flow behavior of powders was observed due to air pressure build-up into an open cavity. Moreover, it was found that the lower punch has to be withdrawn at a higher speed by increasing the turret speed in order to improve the suction feed efficiency.

In [7], authors studied powder segregation behavior during die filling process for a mixture of fine and coarse powders from a stationary hopper. Segregation was found due to the movement of the fine particles on the powder bed during the process. Moreover, the reduction of segregation was possible with the decrease of fill time or enhancing the drop height.

Several studies reported in the literature are on apparatus features such as die width and feed frame design. For example, a study on die width proved that the filling density decreases proportionally with the die width [8]. In another study, the filling density was found correlated with the shoe speed, i.e., the filling density was observed to be increasing as the shoe speed was decreasing [9]. Additionally, in [10], authors studied the influence of different feed frames of powder blends and found that the feed frame had a significantly high impact on powder hydrophobicity due to over lubrication of the powders caused by the shear. This will lead to

an increase of flowability and a decrease in tablet tensile strength.

Similarly, different process conditions such as die filling in the air or in the vacuum, were studied [2], where it was shown that nose flow and bulk flow are two possible classifications of die filling depending on the shoe speed, the powder flow patterns. Moreover, a faster filling rate was observed in the vacuum than the filling rate in the air. Furthermore, in [11], authors proposed a new methodology based on the concept of critical shoe velocity to characterize powder flow, where a critical fill speed that suggests the minimum filling speed at which the die can be filled, completely. The die will not be full if the velocity is higher than the critical velocity. Hence, it was possible to determine the fill ratio at these higher speeds using the following equation:

$$\delta = \left(\frac{v_c}{v_s}\right)^m \tag{1}$$

where v_c is the critical velocity, v_s is the shoe speed, and m is a coefficient of value between 1.0 and 1.6.

Guo et al. [12] used a coupled discrete element method with computational fluid dynamics (DEM-CFD) to modeled die filling process, in which they explored the influence of powder properties (i.e., particle size and density) on die filling in the air and in the vacuum. They characterized flow in terms of mass flow rate and found that the mass flow rate is constant in the vacuum; whereas for light and small particles in the air-sensitive regime, it was found that the mass flow rate increases as the particle size and density increase, and for coarse and dense particles in the air-inert regime, a negligible effect on airflow was observed. Guo et al. [13] then investigated segregation induced by air during die filling, where DEM-CFD was used to simulate monodispersed mixtures. For die filling using a stationary shoe, segregation was induced due to the air presence and a lower concentration of light particles was found in the lower region of the die. Whereas, for die filling using a moving shoe, lighter particles predominantly gather in the leading edge of the flow stream.

Although several investigations on die filling have been proposed both experimentally and numerically, the influence of the granule property on die filling behavior and how this is correlated with the raw material properties and process variables are not fully understood. Moreover, it is of commercial benefit for pharmaceutical companies to reduce the development time and costs with the contemporary improvement of the pharmaceutical process design [14]. To achieve these goals, the use of the multilayer perceptron (MLP) on a limited set of experiment data for predictive modeling can be of great benefit [15]. Additionally, use of computational intelligence techniques is advantageous for various pharmaceutical processes [16, 17]. Hence, the objective of this work is to develop a predictive model that can accurately describe the die filling behavior of pharmaceutical granules. For this purpose, a computational intelligent (CI) technique, named flexible neural tree (FNT), is used for predicting die filling performance of pharmaceutical granules

2 Computational intelligence techniques

Computational intelligence techniques discover knowledge from data and create predictive models. Moreover, in a predictive modeling, a causal relationship between the independent variable $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$ and the dependent variable $\mathbf{d} = \langle d_1, d_2, \ldots, d_N \rangle$ is discovered. Specifically, the unknown parameter $\mathbf{w}$, which represents the learning component of a CI technique, are determined by minimizing the root mean square error e (RMSE) between the desired output $\mathbf{d}$ and the predicted output $y = \langle y_1, y_2, \ldots, y_N \rangle$. Hence, the training of a CI technique is a process of searching the optimum parameter $\mathbf{w}$. In this work, the RMSE e was computed as:

$$e = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - d_i)^2}, \tag{2}$$

where N is total samples in a training set. An additional performance measure used in this work was correlation coefficient r that quantifies the correlation between two the variables: the desired output d and the predicted output y. The correlation coefficient value r was computed as:

$$r = \frac{\sum_{i=1}^{N}((y_i - \text{mean}(y))(d_i - \text{mean}(d)))}{\sum_{i=1}^{N}(y_i - \text{mean}(y))^2 \sum_{i=1}^{N}(d_i - \text{mean}(d))^2} \tag{3}$$

where function $\text{mean}(\cdot)$ returns the average of the elements of a vector. The correlation coefficient r equal to 1 indicate the best performance and the correlation coefficient r equal to -1 indicates the worst performance.

The common CI techniques such as multilayer perceptron (MLP), Gaussian process regression (GPR), and reduced error pruning tree (REP-Tree) solves several industrial and real-world problems [18]. The concise definitions of these CI techniques are as follows:

- The MLP is a mathematical form of human-like learning, where a network of computational nodes arranged in a layered architecture is trained using a dataset [19].

- In REP-Tree, based on the dataset, a tree-like structure is created, where the tree's internal nodes are binary decision nodes and the leaf nodes are the output nodes [20].

- The GPR is a Gaussian distribution based extension of linear regression technique [21].

These CI techniques are implemented as an open source library, i.e., as a software tool named WEKA [22]. A detailed description of these models is available in [23].

In comparison to the models above, the flexible neural tree (FNT) produces a tree-like model, where the nodes of the tree are similar to the MLP nodes. Hence, FNT differs from MLP in its structural configuration, and it differs from REP-Tree in its node type. Moreover, the tree-like structure in the FNT is created using genetic programming (GP) [24, 25]. Therefore, the basic advantage of using FNT over MLP and REP-Tree lies in its ability of automatic adaptation

into the tree-structure and the input feature selection with the help of GP [26]. Such an ability is necessary and has been used in several successful applications. Examples of such applications are cement decomposing-furnace production-process modeling [27]; exchange rate forecasting [28]; gene regulatory network reconstruction and time-series prediction from gene expression [29]; Internet traffic identification [30]; and protein desolation rate prediction [31].

It is known that for a good quality data, the predictive capability of MLP lies in its learning components, such as connection weights, network-architecture, activation-function, input features, and learning rules [32]. The design of FNT includes all these MLP-like learning components and offer an automatic adaptation of these components by using GP [24, 25]. The descriptions of the CI models are given below.

2.1 Multilayer Perceptron (MLP)

Usually, an MLP structure has three layers: an input layer, a hidden layer, and an output layer. Each layer of an MLP consists of several nodes, and each node receives real-valued inputs and produces outputs based on the weighted linear combination of inputs. Then, the magnitude of the node's output is computed by using a nonlinear activation function. In this work, resilient propagator algorithm was used to train an MLP [33].

2.2 Gaussian process for regression (GPR)

GPR is a statistical technique that extends multivariate Gaussian distribution. Let $\hat{y} = f(x) = \phi(\mathbf{x})^T \mathbf{w}$ be a regression model, where an input vector $\mathbf{x} = \langle x_1, x_2, \ldots, x_d \rangle \in X$ is mapped onto an N-dimensional feature space $\phi(\cdot)$, and $\mathbf{w}$ is the learning component. Then, a mean function $m(x)$ and a covariance function $k(x, x^{'})$ defines a Gaussian process $gp(\cdot)$. The mean and covariance functions are written as:

$m(x) = \mathbb{E}[f(x)],$
$k(x, x^{'}) = \mathbb{E}[(f(x) - m(x)), (f(x^{'}) - m(x^{'}))].$

and Gaussian process $gp(\cdot)$ may be written as:
$f(x) \sim gp(m(x), k(x, x^{'})).$

The mean function $m(x)$ and the covariance function $k(x, x^{'})$ are described in [21].

2.3 Reduced error pruning tree (REP-Tree)

Reduced error pruning tree depends on the creation of a decision tree. A decision tree resembles a tree-like structure whose non-leaf nodes are decision nodes. A decision node is tested (a binary decision) against an input feature and the leaf node are the outcomes of the test. In REP-Tree, a
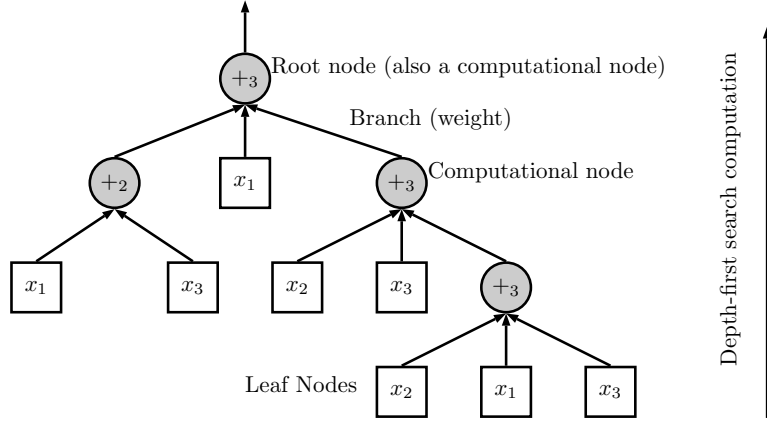
5

Figure 1: A typical FNT S with computational set $F = \{+_2, +_3\}$ and terminal set $T = x_1, x_2, x_3$.

decision tree is created using the information available in a training data and then the leaves and branches of the tree are pruned using the reduced-error pruning method to avoid over-fitting. In the reduced error pruning method, the nodes of the tree are pruned until the accuracy of the prediction is not affected. A detailed study of this algorithm is available in [20].

2.4 Flexible neural tree (FNT)

To minimize RMSE e so that it converges to zero, an FNT adapts its parameter by using learning algorithms (such as genetic programming and differential evolution). Such adaptation is performed by optimizing the components of tree-like structure: internal-nodes (computational nodes), branches (connection-weights), and leaf nodes (terminal nodes).

Mathematically, a union of computational node F and the terminal node T describes an FNT denoted as S, where a computational node represents an activation function, and a terminal node represents an input. Hence, an FNT can be represented as:

$$S = F \cup T = \{+_2, +_3, \cdots, +_N\} \cup \{x_1, x_2, \ldots, x_d\} \tag{4}$$

where $+_n$ ($n = 2$ or 3 or ... or N) indicates that a computational node that takes two or more arguments; whereas, the terminal node takes zero arguments. Fig. 1 is an illustration of FNT model.

The shaded circle in Fig. 1 represents a computational node (the root node is among the computational nodes), the branches of the tree indicate the weights of the model, and the leaf nodes (shown in square) of the tree indicates the selected input features. The root node of the tree represents the output of the model.The output of an FNT at the root node is computed using a bottom-up approach (as indicated by arrows in Fig. 1). The i-th computational node (shown in Fig. 2) of the tree (Fig. 1) receives n^i input through n^i connection-weights (branches) and two adjustable parameters a_i and b_i, which represent the arguments of the i-th computational node's activation function. The purpose of using an activation function at
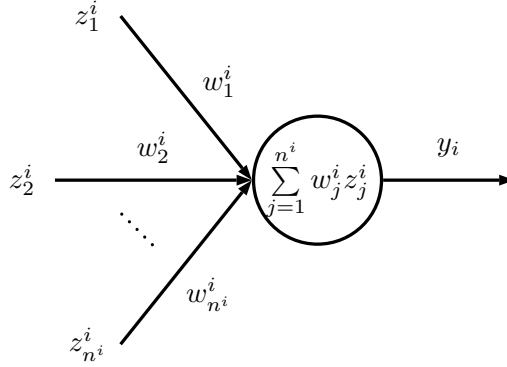
6

Figure 2: Illustration of an FNT's computational node. The variables z_j^i and w_j^i in the figure indicate the input and the weight. The variable y_i is the output of the node.

the computational node is to limit computational node's output within a certain range. For example, output of i-th node that contains a Gaussian function is computed as:

$$y_i = f_i(a_i, b_i, o_i) = \exp\left(-\left(\frac{(o_i - a_i)}{b_i}\right)\right) \tag{5}$$

where o_i is the weighted summation of the inputs to the i-th computational node (see Fig. 1), which is computed as:

$$o_i = \sum_{j=1}^{n^i} w_j^i x_j^i \tag{6}$$

where $z_j^i \in \{x_1, x_2, \ldots, x_d\}$ or $z_j^i \in \{y_1, y_2, \ldots, y_k\}$, i.e., z_j^i can be either an input feature (leaf node value) or the output of another node (computational node value). The weight w_j^i is the connection weight of real value in the range $[w_l, w_h]$. Now, two aspects are involved in the training of an FNT: tree structure optimization and parameter optimization. The optimization of these two aspects of FNT took place in two phases:

1) Structure optimization using genetic programming (GP)

2) Parameter optimization with differential evolution (DE).

2.4.1 Structure optimization phase

GP is a population-based evolutionary-inspired algorithm that makes use of genetic operators such as crossover and mutation and iteratively constructs an optimal tree-like structure (the structure that gives the lowest approximation error) through a vast topological search space [26]. In a crossover operation, the randomly selected subtrees of two trees are exchanged between each other's and in a mutation operation a randomly selected subtree is either deleted entirely or replaced by a randomly generated subtree. In this work, the mutation operators were designed as follows:

7

1) Replacing one terminal node with a new terminal node;

2) Replacing all terminal nodes with new terminal nodes;

3) Growing a new sub-tree and replacing a randomly selected function node;

4) Replacing a computational node with a terminal node.

Since GP is an evolutionary-inspired algorithm, only the best input features (regarding the predictability) are selected during the structure optimization phase.

2.4.2 Parameter optimization phase

DE is a population-based parameter optimization algorithm that makes use of crossover operator and iteratively searches through a search space to optimize a vector that represents the parameters of a model [34]. Let assume that H is the population of the tree parameter vectors. Hence, DE tries to find an optimal tree parameter $\mathbf{h}^*$ by using operators, such as selection, crossover, mutation, and recombination. At each iteration, the selection operator selects three random vectors from the population H: $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ such that $\mathbf{r}_1 \neq \mathbf{r}_2 \neq \mathbf{r}_3$ and the vector $\mathbf{g}$ that denotes the best solution in the population. Therefore, a new tree parameter vector $\mathbf{h}^{new}$ is computed as:

$$
\mathbf{h}^{new} \begin{cases} \mathbf{r}_1 + F \cdot (\mathbf{r}_1 - \mathbf{g}) + F \cdot (\mathbf{r}_2 - \mathbf{r}_3), & \text{if } \mathbf{u} < \mathbf{C} \\ \mathbf{r}_1, & \text{if } \mathbf{u} \geq \mathbf{C} \end{cases} \tag{7}
$$

where $\mathbf{u}$, is a vector of random values taken from a uniform distribution between 0 and 1, $\mathbf{C}$ is a vector of crossover probability, and F is a vector of mutation factor. This process is repeated until $\mathbf{h}^*$ is found. Hence, by combining structure optimization phase and parameter optimization phase, a general optimization procedure of FNT can be summarized in Fig. 3.

2.4.3 Limitations of FNT

The fundamental limitation of the FNN is in comparison to MLP is at its output node, i.e., the root node of the FNT (tree-like structure). Since the tree can have only one root node, one has to produce multiple models for multiple output prediction problems. Moreover, FNT can only offer multi-input-single-output models; whereas MLP can provide multi-input-multi-output models. Additionally, the proposed improvement in FNT in comparison to other CI techniques is its structure optimization. However, during cross-validation (CV) methods such as k-fold CV, the structure is determined only once (for only first fold) and for the rest of the $(k-1)$ folds only parameters were optimized. However, if the structure is determined for all the k folds. Then, k different FNT models should be obtained.
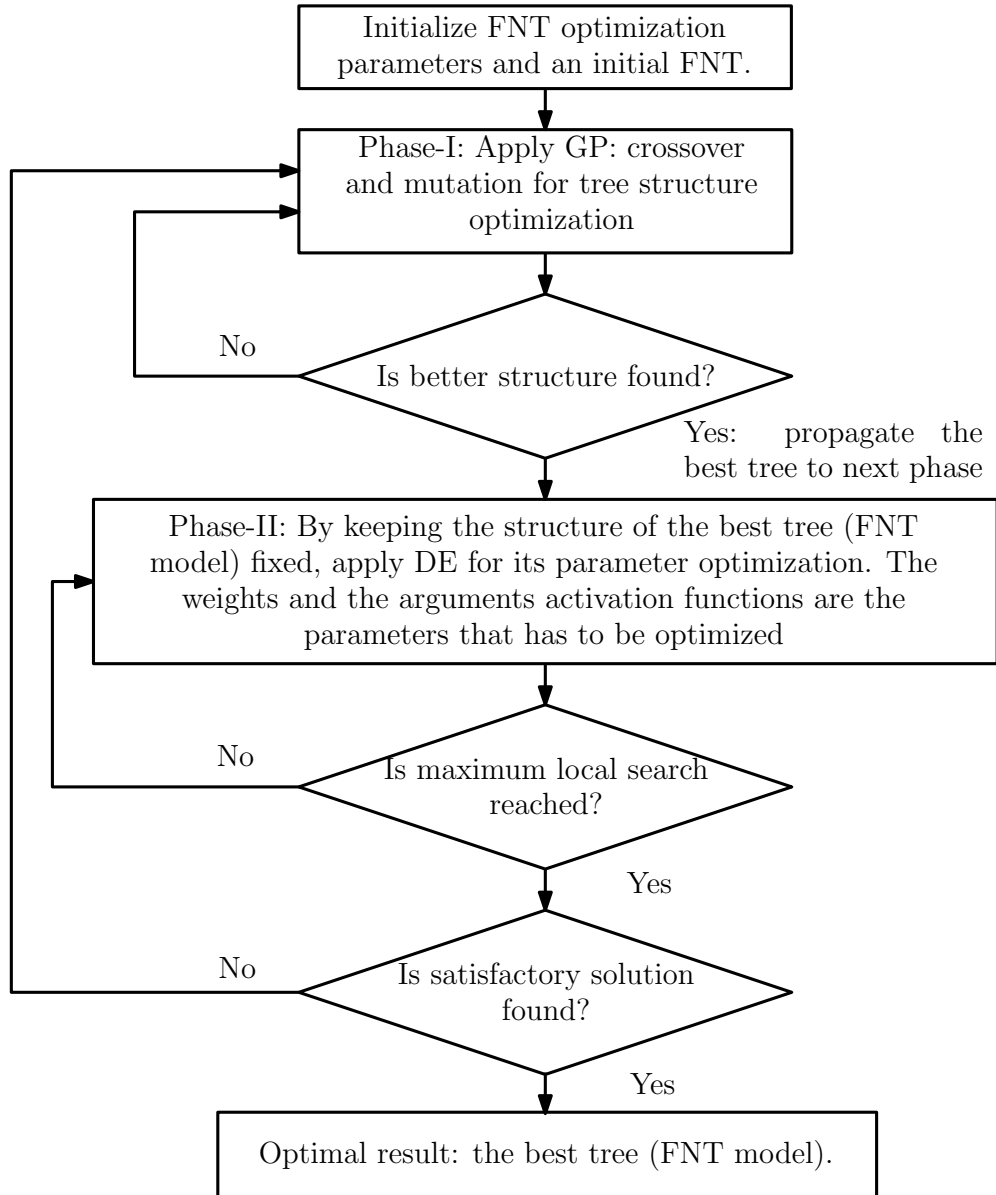
Figure 3: General FNT optimization procedure

3 Experimental study

3.1 Materials and methods

Microcrystalline cellulose (MCC) of three different grades was chosen as the raw materials, including Avicel PH-101, Avicel PH 102, and DG. A custom-made gravity-fed roll compactor with two counter rotating smooth rolls of 200 mm in diameter and 46 mm in width was used for ribbon production [35, 36]. The roll gap was set at 1.2 mm and the roll speed at 1 rpm roll speed. Ribbons were milled using a milling system (SM100, Retsch, Germany) equipped with a mesh size of 4 mm at a constant speed of 1,500 rpm. The granules were sieved into different size classes (0-90, 90-250, 250-500, 500-1000, 1000-1400, 1400-2360 µm) that were used for the die filling experiments. The corresponding upper size limit was used as the granule size in the current study (i.e., granules in the size range 0-90 µm were regarded as granules with a size of 90 µm), as commonly adopted in particle technology.

3.2 Data collection

True densities of the three MCC powders were determined using a Helium Pycnometer (AccuPyc II 1340, Micromeritics, UK). Particle size analysis was performed using a size analyzer (Camsizer XT, Retsch, UK). The experiments were run for 2-3 minutes and the data were collected. The mean diameter (d50) defined as the size value below which 50% of the particles lies, was then determined. Die filling experiments were performed using a model die filling system that consists of a shoe driven by a pneumatic driving unit, a positioning controller, and a displacement transducer [2]. For each granule size class, experiments were performed using seven different shoe speeds in the range of $10 \, \mathrm{mm \, s^{-1}}$ to $400 \, \mathrm{mm \, s^{-1}}$. For each die filling experiment, the powder mass deposited into the die was weighted, and the value was recorded. Each experiment was repeated three times. In total, 389 experiments (3 powders, 6 granule sizes, 7 speeds, 3 repeats) were performed, and 389 datasets were generated.

From the collected experimental data, four parameters were chosen as inputs for the modeling: true density and mean diameter (d50) of raw powders, granule size (µm), and shoe speed ($\mathrm{mm \, s^{-1}}$) and the deposited mass was the only output. Table 1 shows a selection of few samples (taken from 389 samples) of the generated dataset.

3.3 Predictive modeling

Predictive modeling with the CI techniques discussed in Section 2 was performed in the following manner. Two different cases, i.e., 10-fold cross validation (10FCV) and 5-cross-2-fold cross validation (5x2FCV) methods, were considered. In 10FCV, a dataset is first randomized (shuffled). Then, 90% of data samples are used for training of a model and the rest of 10% data

Table 1: Example of few data samples generated for modeling.

| #	Samples Name	Input				Output
		True density	d50 (μm)	Granules size	Shoe speed	Mass (g)
		Feature #1	Feature #2	Feature #3	Feature #4	
1	MCC PH 101	1581	59.83	90	10	12.81
2	MCC PH 101	1581	59.83	90	10	12.78
:	:	:	:	:	:	:
5	MCC PH 101	1581	59.83	90	20	12.3
6	MCC PH 101	1581	59.83	90	30	9.55
:	:	:	:	:	:	:
135	MCC PH 102	1570.3	94.7	250	50	13.45
136	MCC PH 102	1570.3	94.7	250	60	13.5
:	:	:	:	:	:	:
388	MCC DG	1785.6	52.33	2360	400	9.51
389	MCC DG	1785.6	52.33	2360	400	9.3
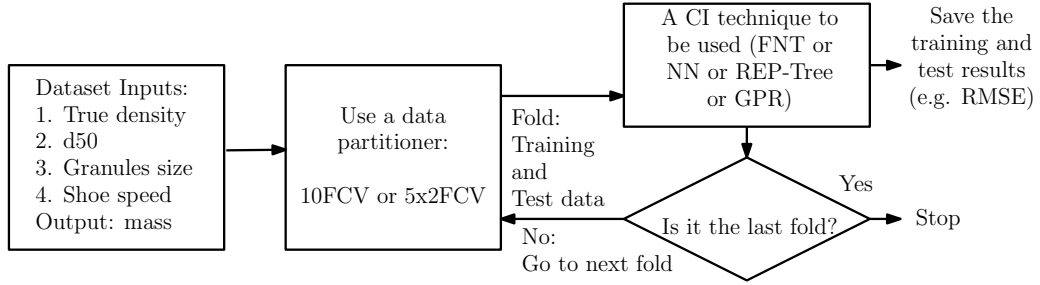
Figure 4: Flow chart illustrating the data flow during a predictive modeling.

samples are used for testing. This process is repeated for 10 times and each time a distinct set of 10% dataset is picked for testing of the model. Hence, called 10FCV. In the 5x2FCV, the data set is randomly divided into two sets, where each set is 50% in size of the entire dataset. When the first set is used for the training of a model, then the second set is used for the testing of the same model, and the vice-versa. This process is repeated for five times hence referred to as 5x2FCV validation. In each case, the four CI techniques discussed above (i.e., FNT, MLP, GPR, and REP-Tree) were used to develop the models, and their performance was evaluated. Fig. 4 illustrates the predictive modeling process adopted in this work.

The collected dataset was pre-processed by normalizing the features of dataset between zero and one using a min-max normalization method. Table 2 lists the underlying parameters for the FNT, whereas, Table 3 describes the basic parameter set-up for other three CI techniques: MLP, GPR, and REP-Tree. The model accuracy was assessed using RMSE e defined in (2) and correlation coefficient r defined in (3).

To obtain the best model, the training and test performances of the models were observed using five performance indicators:

11

Table 2: Parameters settings and the values are chosen during FNT optimization.

#	Parameter Name	Definition/Purpose	Value
1	Tree height	The maximum levels of a tree.	5
2	Tree arity	Maximum siblings of a node in a tree.	4
3	Tree node type	Type of activation function used at nodes.	Gaussian
4	GP population	Total candidates in a GP population.	30
5	Mutation probability	The frequency of mutation operation.	0.2
6	Crossover probability	The frequency of crossover operation.	0.8
8	Tournament size	Total candidates for a tournament.	15
9	DE population	The initial size of the DE population.	50
10	Range of node search	The lower and upper bound of activation function arguments.	[0.0,1.0]
11	Range of edge search	The lower and upper bound of tree edges.	[-1.0,1.0]
13	Structure training	Maximum generations of GP training.	100 000
14	Parameter training	Maximum evaluations of parameter training.	10 000

Table 3: Parameters settings and the values are chosen during MLP, GPR, and REP-Tree training. The settings mentioned are those used in the software tool [22]

#	CI Technique	Parameter Name	Definition/Purpose	Values
1	MLP	Learning rate	Convergence speed.	0.3
2		Momentum rate	Magnitude of past iteration influence.	0.2
3		Hidden Layer	Maximum nodes at hidden layer.	350
4		Iterations	Maximum number of evaluations of parameter optimization	500
5	GPR	Kernel	The function used for implementing covariance function.	RBF Kernel
6	REP-Tree	No. Leaf Instances	Minimum No. of children per node.	2
8		Depth	Maximum limit of tree depth/level.	No limit
9		Pruning	Pruning of tree nodes.	Allowed

1) The RMSE e as per (2).

2) The correlation coefficient r as per (3).

3) The standard deviation (std) of correlation coefficient values of each fold.

4) The model's complexity. In the case of tree-based models (FNT and REP-Tree), it is the total number of nodes in the tree including computational node and leaf nodes (e.g., the complexity of the model shown in Fig. 1 is 12). In the case of MLP, it is the number of total computational nodes. Since GPR is an extension of the linear regression model, the complexity in the case of GPR was not commuted.

5) The number of selected input features.

3.4   Input feature analysis of die filling experiments

Feature analysis was conducted to understand the significance of the input features in the die filling experiments. For this purpose, at first, M many FNT models were created. Second, two performance dimensions were used: feature selection rate R and predictability score P. Feature selection rate R, which describes the total number of times a particular input feature set was appeared in the list of prepared out all the M models. The feature selection rate is defined as:

$$R_j = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}(A_j) \tag{8}$$

where R_j is the selection rate of j-th input feature set $A_j \in \mathbf{A}$ which is a power set $P\{$true density, d50, granule size, shoe speed$\}$, M is the total number of models in the list, and function $\mathbb{I}(A_j)$ is an identity function that returned "1" if j-th input feature set A_j is selected by the i-th model, otherwise, returned "0." Feature selection rate R equal to one is the highest (100% selection rate), and R equal to zero is the lowest (0% selection rate). In other words, the value of selection rate R equal to one means an input feature was selection by all the models in the prepared list and the value of R equal to zero means an input feature was selection by none of the models in the prepared list.

Since in the models in the list may not equal in their performances, the predictability score P based on the RMSEs of the models was computed. The predictability score P describes the predictability of an input feature. To compute predictability score P of j-th input feature set A_j, at first, the fitness F_j of the corresponding input feature set A_j was computed as:

$$F_j = \begin{cases} \sum_{i=1}^{M} e_i \cdot \mathbb{I}(A_j), & \text{if } |A_j| = 1 \\ \sum_{i=1}^{M} e_i \cdot \mathbb{I}(A_j) / \sum_{i=1}^{M} \mathbb{I}(A_j), & \text{if if } |A_j| \geq 1 \end{cases} \tag{9}$$

where e_i is the RMSE of i-th model. The fitness F_j for $|A_j|$ equal to one is the sum of RMSEs, and the F_j for $|A_j|$ greater than one is the average RMSEs of all models that selects subset A_j. Then, the predictability score P_j corresponding to an input feature set A_j was computed by normalizing the fitness of as:

$$P_j = \frac{F_j}{\max_{j=1 \text{ to } z} (F_j)} \tag{10}$$

where function $\max(\cdot)$ determines the maximum fitness value from all F_j. Similar to the selection rate R, the predictability score P equal to one indicates that the feature set has the highest impact on the predictability of the model and predictability score P equal to zero has the lowest.

4 Results and discussion

4.1 Model Prediction

4.1.1 Predictions using the 10FCV method

Table 4 describes the performance of the best models created by using FNT, MLP, GPR, and REP-Tree. In Table 4, the generated models are arranged in descending order (highest accuracy to lowest accuracy) of their test correlation values. It may be observed that the performance of the models created using FNT (when compared the test accuracies, i.e., correlation values) was better than the other CI techniques. Hence, the detail observation of the model Nos. 1, 2, and 3, which were created using FNT are provided.

The model No. 1 (see: row 1 of Table 4) produces a high correlation coefficient (on test set), i.e., 0.95, which indicates high predictability of this model over 10% of unknown samples. However, the model complexity was also high since the total function nodes, and leaf nodes in the created model amounted to 43. In addition, there was no feature selection performed by this model. In comparison to model No. 1, the model Nos 2 and 3 had lower test correlation coefficient (performance was slightly poor). However, their models complexity was simpler, and they offered feature selection, which was advantageous than the model No. 1. Fig. 5 illustrates the performance of the model using regression (scatter) plot and target against predicted value plot. In Fig. 5, each model Nos 1, 2, and 3 respectively were tested over 10% of test samples, i.e., 38 randomly chosen test samples, and the scattered plot and the target against predicted values plot were analyzed. It may be observed that the prediction curve follows the target curve. However, the lower values and outliers were slightly out of the reach in the prediction curve.

Table 4: Performance of the prediction models and validation over 10FCV.

Model No.	Model Type	Mean of RMSEs		Mean of r		Std over r		Model Complexity[1]	Selected Features[2]
		Train	Test	Train	Test	Train	Test		
1	FNT	2.0206	2.0571	0.93	0.95	0.0087	0.0383	43	1, 2, 3, 4
2		2.3891	2.3934	0.91	0.91	0.0083	0.0617	34	2, 3, 4
3		2.5491	2.2618	0.88	0.91	0.0078	0.0563	32	3, 4
4	REP-Tree	2.5751	3.1637	0.88	0.82	-	-	99	1, 2, 3, 4
5	GPR	2.9632	3.4023	0.86	0.79	-	-	-	1, 2, 3, 4
6	MLP	3.3687	3.4427	0.81	0.79	-	-	-	1, 2, 3, 4

Note: [1]Complexity is the sum of total nodes in the created tree-model. [2]Features Nos are assigned in Table 1

Table 5: Performance of the prediction models and validation over 5x2FCV

Model No.	Model Type	Mean of RMSEs		Mean of r		Std over r		Model Complexity	Selected Features
		Train	Test	Train	Test	Train	Test		
7	FNT	2.5075	2.6481	0.88	0.88	0.0415	0.0403	16	1, 2, 3, 4
8		2.6030	2.6792	0.88	0.87	0.0213	0.0217	17	2, 3, 4
9	REP-Tree	2.4460	3.6691	0.89	0.77	-	-	51	1, 2, 3, 4
10	MLP	2.7698	3.8730	0.86	0.76	-	-	-	1, 2, 3, 4
11	GP Reg.	2.7978	3.7869	0.87	0.75	-	-	-	1, 2, 3, 4

Note: Model Nos. are continued from Table 4

4.1.2 Predictions using the 5x2FCV method

In Table 5, a comparison of the best models created using FNT, MLP, GPR, and REP-Tree are provided. The models in Table 5 are arranged in descending order (the highest accuracy to the lowest accuracy) of their test correlation values. Similar to the modeling in 10FCV, the modeling in 5x2FCV and the performance of the models created using FNT outperformed the models created using MLP, GP, and REP-Tree. Hence, a detail observation on the model Nos. 7, and 8 are provided. Fig. 6 illustrates the performance of the model using regression (scatter) plot and target against predicted value plot.

In Fig. 6, each model Nos. 7, and 8, respectively was tested over 50% of test samples, i.e., 194 randomly chosen test samples, and the scattered plot and the target versus prediction plot were analyzed. Similar to the trend as observed in Fig. 5, the trend observed in Fig. 6 says that the prediction curve follows the target curve. However, the lower values and the outliers were out of the reach of the prediction curve. The outlier is clearly visible in Fig. 6d. The outlier in our dataset was because of noise or bad value observed during die filling process.
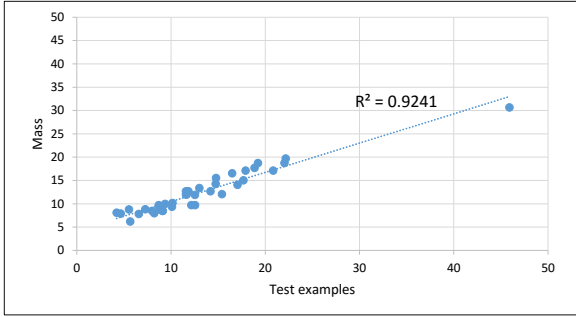
Fig. 7 and Fig. 8 are the illustrations of the created models, where the root node of the tree indicates the output of the models and the leaf nodes (Square boxes with numbers) indicates the input features. In Fig. 7 and Fig. 8, the input feature x_1, x_2, x_3, and x_4 indicate the features
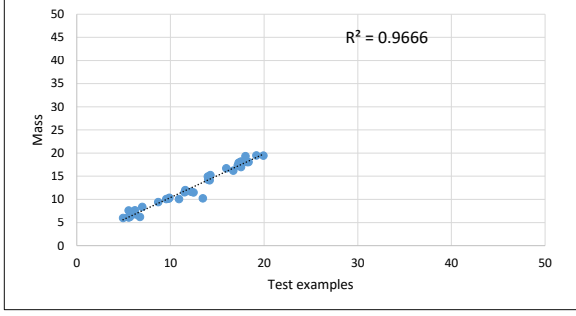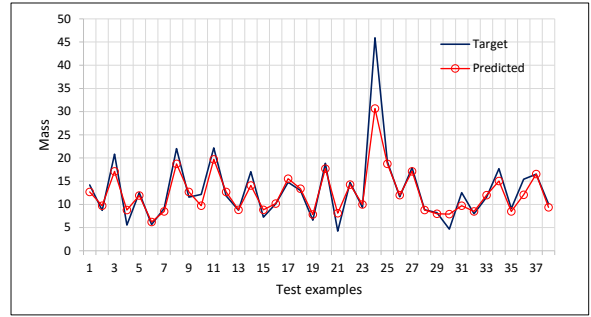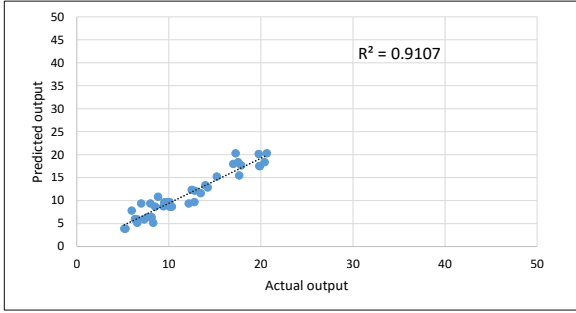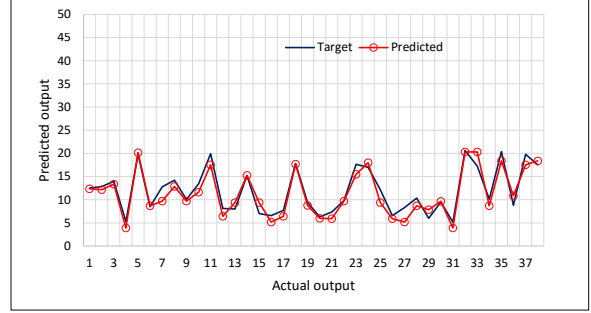
(a) Model No. 1
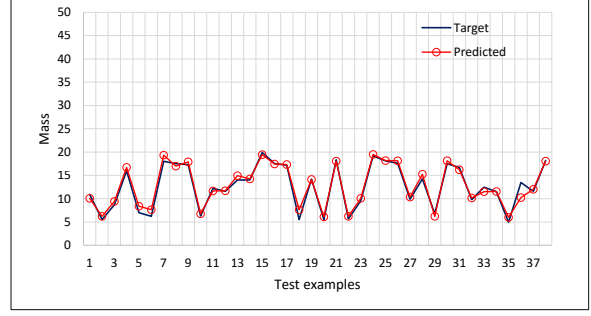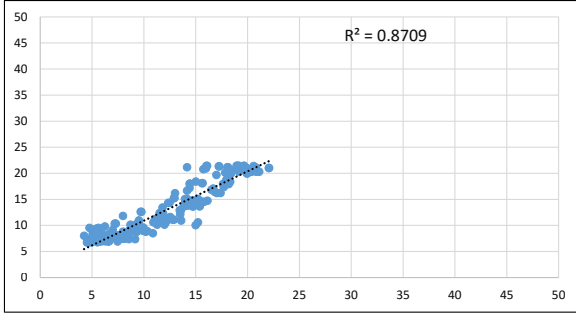
(b) Model No. 1

(c) Model No. 2
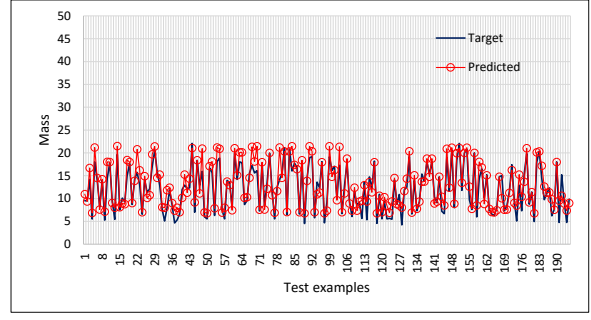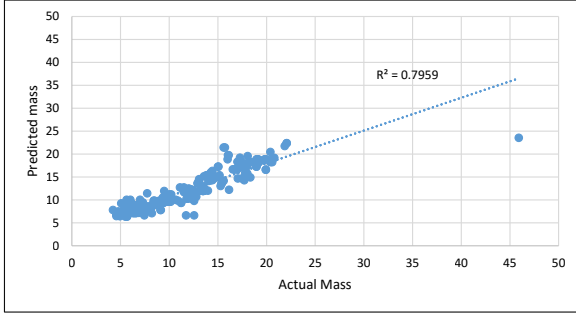
(d) Model No. 2

(e) Model No. 3

(f) Model No. 3

Figure 5: Models evaluation on unknown test samples: The regression plots (a), (c), and (e) indicates a high correlation between actual and predicted values. The plots (b), (d), and (f) shows the one-to-one mapping of target and prediction of the best models Nos. 1, 2, and 3 (see Table 4). The R^2 is the squared value of correlation coefficient r, where R^2 equal to one is the best performance and R^2 equal to zero is the worst performance.
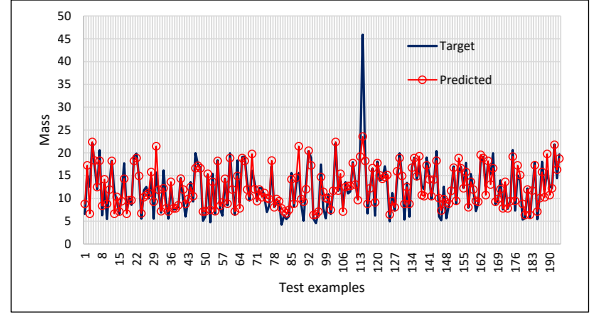
(a) Model No. 7   (b) Model No. 7

(c) Model No. 8   (d) Model No. 8

Figure 6: Models evaluation on unknown test samples. The regression plots (a) and (c) indicates a high correlation between actual and predicted values and plots (b) and (d) shows the one-to-one mapping of target and prediction of the best model Nos. 7 and 8 (Table 5). The R^2 is the squared value of correlation coefficient r, where R^2 equal to one is the best performance and R2 equal to zero is the worst performance.

true density, d50, granule size, and shoe speed, respectively.

4.1.3   Comparison between 10FCV and 5x2FCV methods

Each 10FCV and 5x2FCV methods have their advantages. This is the reason both methods were used for creation and validation of in this work. In 10FCV, a model uses a large sample for training. Thus, possess higher representativeness of real world (data samples) during learning, i.e., the model is trained efficiently. Whereas, in 5x2FCV a model used an equal proportion of data samples for training and testing (smaller training samples than 10FCV, but larger test samples than 10FCV). Thus, 5x2FCV possess higher generalization ability. Hence, at one hand, if a high test correlation coefficient obtained using 10FCV, it indicates that an effective predictive model can be created from the given dataset. Whereas, if a high test correlation coefficient obtained using 5x2FCV, it indicates that a general predictive model can be created from the given dataset.

The models 7 and 8 were created using 5x2FCV, and their test correlation values were found competitive to the model No. 1using 10FCV method. From our 5x2FCV results, it was found that the model Nos. 7 and 8 were simple in comparison to model Nos. 1, 2, and 3, but their accuracies (correlation coefficient) were slightly poorer in comparison. However, the
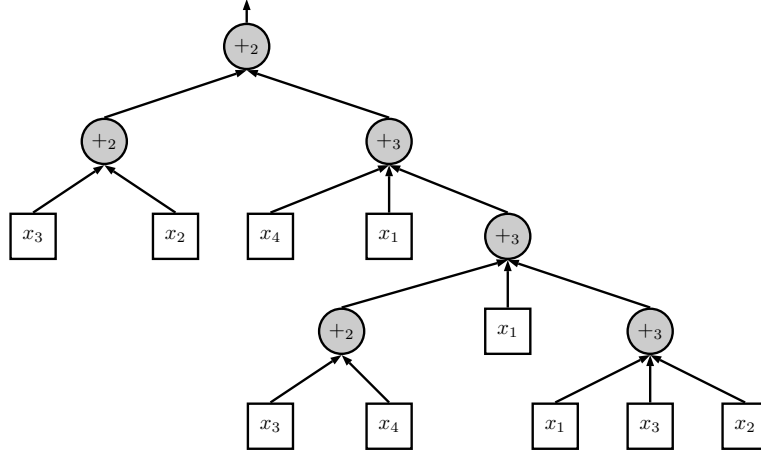
Figure 7: Tree-like structure of predictive model No. 7 created using 5x2FCV method: Complexity equal to 16 is the sum of the computational nodes (node in circles) and the leaf nodes (node in square).
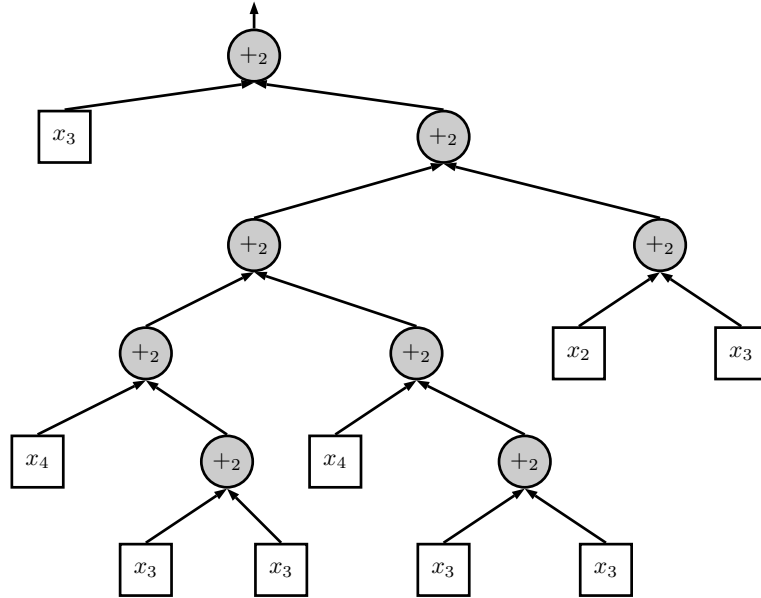

Figure 8: Tree-like structure of predictive model No. 8 created using 5x2FCV method: Complexity equal to 17 is the sum of the computational nodes(node in circles) and the leaf nodes (node in square).

model Nos. 7 and 8 were tested over 50% test samples. Hence, it describes that the deposited mass can be efficiently predicted by using die filling process variables knowledge. Moreover, choice of model is subjective. Since simple models possess higher generalization ability, the competitive accuracies of the model Nos. 7 and 8 to the model Nos. 1, 2, and 3 tells that one had to choose between the accuracies and the generalization ability of the models. Fig. 5 and Fig. 6 present the graphical visualization of the 10FCV and 5x2FCV models, respectively, where, the test performance of the models over each test samples were examined. The 5x2FCV models produced similar trend when the models were tested over 50% of samples. However, the 5x2FCV models (see Fig. 6(d)) did not predict the outlier as close as 10FCV models (see Fig. 5(d)) predicted it.

4.2 Feature analysis

A total 30 models were created using FNT for feature analysis. Since the evolutionary process was used during model creation, the created models selected the input features set that had the highest predictability. Therefore, the RMSEs and selected input feature set by the models were placed into a list. Subsequently, a comprehensive feature analysis was performed. For this purpose, two performance measure dimensions were adopted: feature selection rate R as defined in (8) and feature predictability score P as defined in (10). The feature analysis was categorized into two phases.

1) The identification of individual input features. Here, for (8) and (9) the feature set $|A_j|$ was set to one, which indicates that only one input feature was analyzed at a time. Since there were four input features in our dataset, in this phase, $P \subset A$, i.e., P was equal to $\{A_1, A_2, A_3, A_4\}$ (see Table 6 for the definition of $_1, \ldots, A_4$), i.e., z in (10) was equal to four.

2) The identification of feature subset, i.e., identification of the best combination of input feature. Here, for (8) and (9) the feature set $|A_j|$ can be one or two or three or four. After examining the selected feature by the models in the list, there were six different input feature subsets was found. Hence, in this phase, $Q \subset A$, i.e., Q was equal to $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ (see Table 7 for the definition of $A_1, \ldots, A_6$), i.e., z in (10) was equal to six.

4.2.1 Identification of the significance of individual input features

Table 6 describes the feature analysis results performed for the individual input features. The significance of individual features true density, d50, granule size, and shoe speed was examined. The features true density and d50 represents the powder properties. Whereas, the granule size and shoe speed represents the die filling process variables. It can be observed that the selection rate and predictability score of d50 (0.62069 and 0.58626) were higher than that of the selection rate and predictability score of true density (0.55173 and 0.54136). Therefore, d50 possess

Table 6: Significance of individual input features.

#	Input Features set	Selection Rate (R)	Predictability Score (P)
1	A_1= True density	0.55173	0.541356
2	A_2= d50	0.62069	0.586262
3	A_3= Granule size	1	1
4	A_4= Shoe speed	0.86207	0.92563

Table 7: Optimal subset of input features.

#	Input Feature set	Selection Rate (R)	Predictability Score (P)
1	A_1= True density, d50, Granule size, Shoe speed	0.31035	0.969497
2	A_2= d50, Granule size, Shoe Speed	0.17242	0.941601
3	A_3= True density, Granule size, Shoe speed	0.13793	1
4	A_4= Granule size, Shoe speed	0.24138	0.979663
5	A_5= True density, d50, Granule size	0.10345	0.493741
6	A_6= d50, Granule size	0.03448	0.470451

comparatively higher importance as a powder property than that of the true density. Similarly, the process variable granule size was more influential than that of shoe speed. However, the difference of significance level was marginal, but when the entire four input variables were compared, the process variables were having significantly higher selection rate and predictability score than that of the properties of the powder. This fact was also evident from feature subset analysis.

4.2.2 Identification of the best set input features.

Table 7 shows interesting findings, where it can also observe that the predictability score of subset A_4 (process variable granule size and shoe speed) and the subsets A_1, A_2 and A_3 (with both process variables combined with one or two powder properties) were higher compared to the subsets A_5 and A_6 (subsets where one of the process variables was not used for prediction).

The subset analysis produced a clear picture. It says that although the predictability score of subset A_3 was highest, the selection rate of subset A_1 was highest. This indicates that the evolutionary process often preferred to use the combination of entire features, i.e., the set A_1. However, among the subsets A_1, A_2, and A_3, the selection rate of subset A_2 was higher, which indicates d50 had the higher ability to represent powder properties than that of true density, but again the difference was marginal (say approximately higher by only four percent).

### 4.2.3 Accuracy of Model Predictions

Fig. 9 and Fig. 10 show the comparison between die filling experimental results and predicted results from the model No 1 of the 10FCV method using FNT. This model was chosen because it has the highest value of R^2, which leads to a better fitting between experimental and predicted data. More specifically, Fig. 9 and Fig. 10 present the mass collected after each experiment as a function of the shoe speed of the three different MCCs powders for six different granule size ranges. It shows that there is a decrease of mass deposited into the die with increasing shoe speed for all the materials and granules size ranges investigated. Moreover, a general increase of deposited mass at a consistent shoe speed was found with the increasing granule size. MCC DG tends to have higher mass deposited values compare to MCC PH 102 and MCC PH 101 at all the shoe speeds considered and for all the different size ranges analyzed, exception for the granule size range 250-500 µm (Fig. 9(c)). MCC PH 101 and MCC PH 102 show an identical trend in all the experiments performed. Interestingly, results for finest granules (Fig. 9(a) and Fig. 9(b)) appears to have larger variations (due to the lower experimental reproducibility) compare to those for coarser granules (Fig. 10(b) and Fig. 10(c)). It is clear that the models give better predictions for coarser granules (Fig. 10(b) and Fig. 10(c)) than for finer granules (Fig. 9(a) and Fig. 9(b)) for all the materials under investigation. In particular, the model gives almost identical values to the measured ones for coarser granules, which proves that the FNT method used can predict die filling behavior for such materials with high accuracy. The accuracy of the model appears to rely on the consistency (or scattering) of the experimental data.

5   Conclusions

Flexible neural tree (FNT) was used to predict die filling behavior of MCCs granules of different size ranges. Two main methods were investigated, 10-fold cross-validation (CV) and 5x2-fold CV. Computational intelligence (CI) models were developed using FNT, multilayer perceptron, reduced error pruning tree, and Gaussian process regression. It was observed that the flexible neural tree models performed better than other CI techniques. Additionally, by examining FNT models of each method, it was found that 10-fold CV was a more efficient method with a higher correlation coefficient than the 5x2 fold CV. The experimental results were used as inputs and outputs of the FNT models. The constructed model efficiently predicted the deposited mass based on the knowledge gathered from the experimental data. Similarly, the feature analysis discovered that the shoe speed and the granule size are more significant in terms of governing the deposited mass than the raw powder properties (true density and d50). Interestingly, die filling behavior of coarser granules are easier to predict than fine granules for all the materials considered. This is due to the higher reproducibility of the experimental data for larger granules.
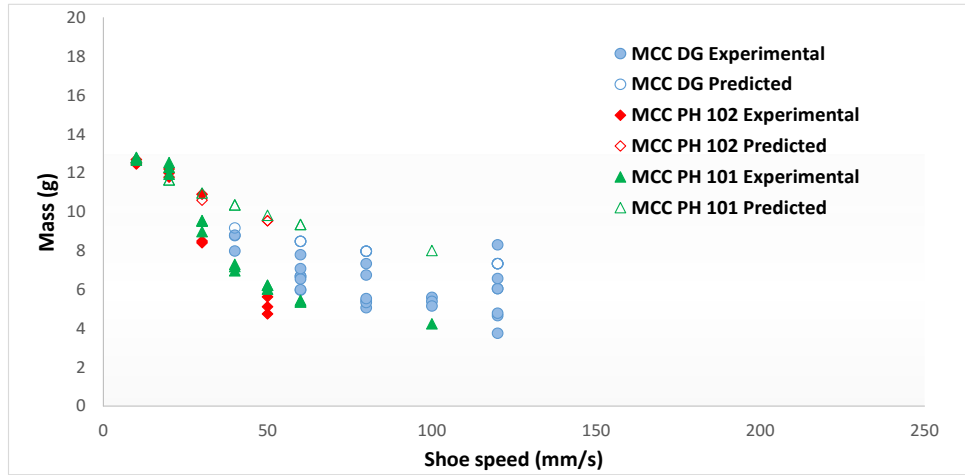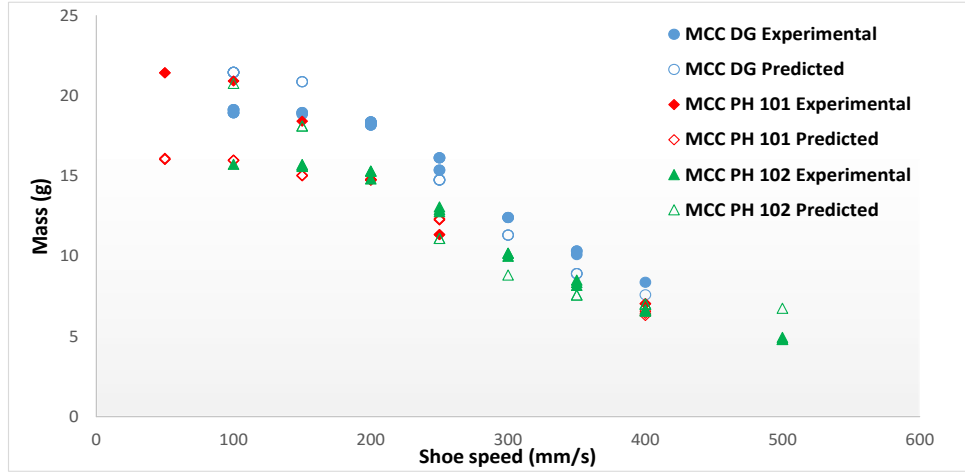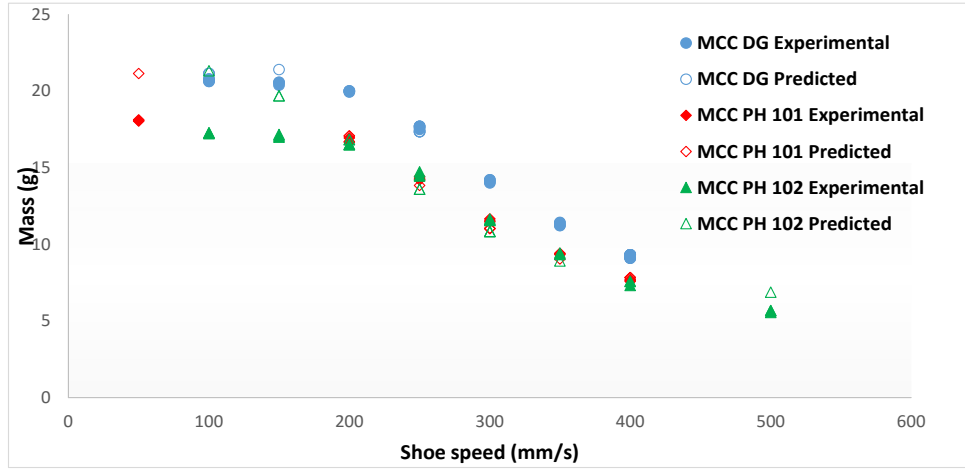
(a)
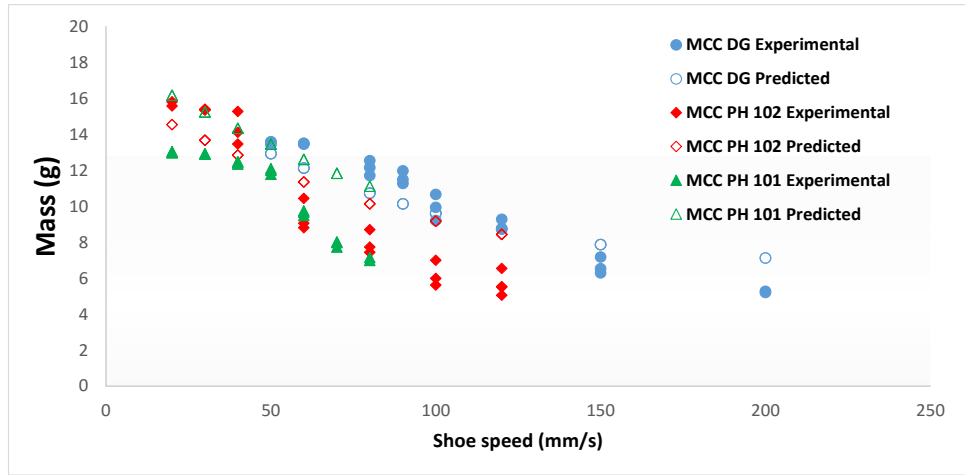
(b)
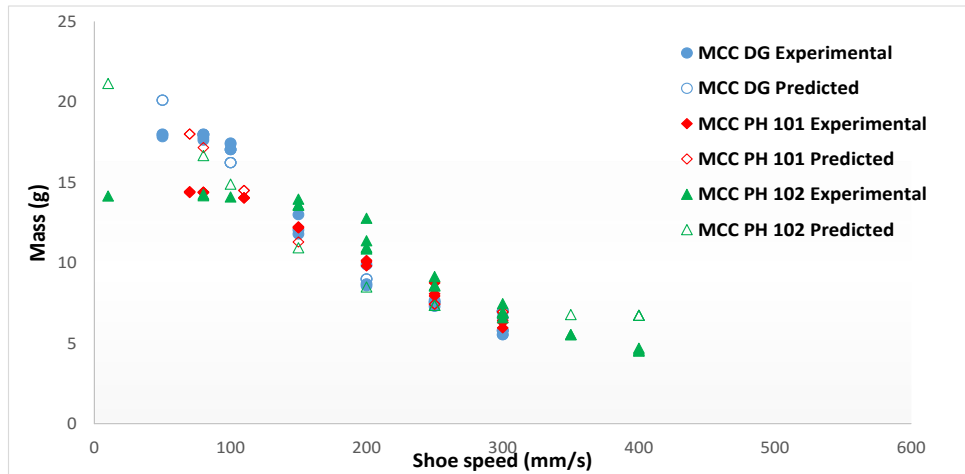
(c)

Figure 9: Comparison of experimental results and model predictions for 3 MCC granules of different size ranges: a) 1-90 µm, b) 90-250 µm, c) 250-500 µm.
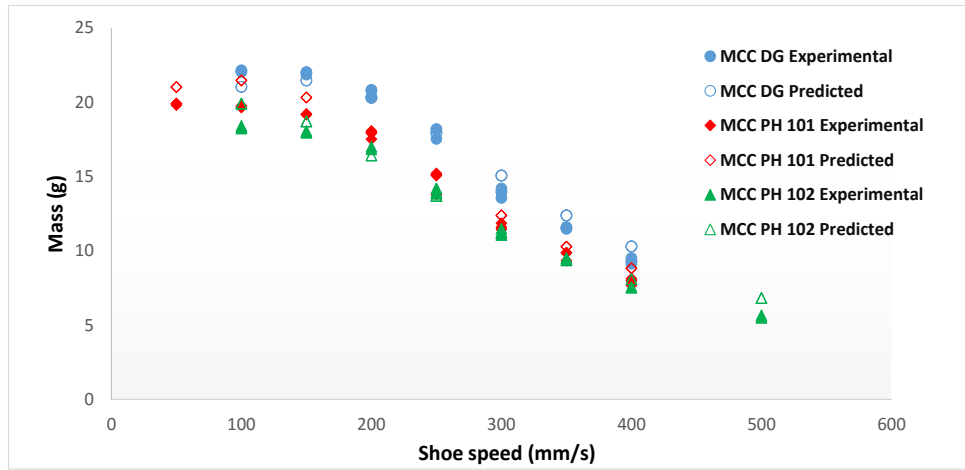
(a)

(b)

(c)

Figure 10: Comparison of experimental results and model predictions for 3 MCC granules of different size ranges: a) 500-1000 µm, b) 1000-1400 µm, and c) 1400-2360 µm.

Acknowledgments

References

[1] O. Coube, A. Cocks, and C.-Y. Wu, "Experimental and numerical study of die filling, powder transfer and die compaction," *Powder Metallurgy*, vol. 48, no. 1, pp. 68–76, 2005.

[2] C.-Y. Wu, L. Dihoru, and A. C. Cocks, "The flow of powder into simple and stepped dies," *Powder Technology*, vol. 134, no. 1, pp. 24–39, 2003.

[3] L. Schneider, I. Sinka, and A. Cocks, "Characterisation of the flow behaviour of pharmaceutical powders using a model die–shoe filling system," *Powder Technology*, vol. 173, no. 1, pp. 59–71, 2007.

[4] C.-Y. Wu, "Dem simulations of die filling during pharmaceutical tabletting," *Particuology*, vol. 6, no. 6, pp. 412–418, 2008.

[5] L. Mills and I. Sinka, "Effect of particle size and density on the die fill of powders," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 84, no. 3, pp. 642–652, 2013.

[6] S. Jackson, I. Sinka, and A. Cocks, "The effect of suction during die fill on a rotary tablet press," *European journal of pharmaceutics and biopharmaceutics*, vol. 65, no. 2, pp. 253–256, 2007.

[7] L. Lawrence and J. Beddow, "Some effects of vibration upon powder segregation during die filling," *Powder Technology*, vol. 2, no. 2, pp. 125–130, 1968.

[8] G. Bocchini, "Influence of small die width on filling and compacting densities," *Powder Metallurgy*, vol. 30, no. 4, pp. 261–266, 1987.

[9] E. Rice and J. Tengzelius, "Die filling characteristics of metal powders," *Powder Metallurgy*, vol. 29, no. 3, pp. 183–194, 1986.

[10] R. Mendez, F. J. Muzzio, and C. Velazquez, "Powder hydrophobicity and flow properties: effect of feed frame design and operating parameters," *AIChE Journal*, vol. 58, no. 3, pp. 697–706, 2012.

[11] C.-Y. Wu and A. Cocks, "Flow behaviour of powders during die filling," *Powder Metallurgy*, vol. 47, no. 2, pp. 127–136, 2004.

[12] Y. Guo, K. Kafui, C.-Y. Wu, C. Thornton, and J. P. Seville, "A coupled dem/cfd analysis of the effect of air on powder flow during die filling," *AIChE journal*, vol. 55, no. 1, pp. 49–62, 2009.

[13] Y. Guo, C.-Y. Wu, and C. Thornton, "The effects of air and particle density difference on segregation of powder mixtures during die filling," *Chemical Engineering Science*, vol. 66, no. 4, pp. 661–673, 2011.

[14] C. Zhao, A. Jain, L. Hailemariam, P. Suresh, P. Akkisetty, G. Joglekar, V. Venkatasubramanian, G. V. Reklaitis, K. Morris, and P. Basu, "Toward intelligent decision support for pharmaceutical product development," *Journal of Pharmaceutical Innovation*, vol. 1, no. 1, pp. 23–35, 2006.

[15] J. Bourquin, H. Schmidli, P. van Hoogevest, and H. Leuenberger, "Advantages of artificial neural networks (ANNs) as alternative modelling technique for data sets showing nonlinear relationships using data from a galenical study on a solid dosage form," *European Journal of Pharmaceutical Sciences*, vol. 7, no. 1, pp. 5–16, 1998.

[16] C.-Y. Wu and Y.-C. Hsu, "Optimal shape design of an extrusion–forging die using a polynomial network and a genetic algorithm," *The International Journal of Advanced Manufacturing Technology*, vol. 20, no. 2, pp. 128–137, 2002.

[17] D. Kim and B. Kim, "Application of neural network and fem for metal forming processes," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 6, pp. 911–925, 2000.

[18] H.-K. Lam and H. T. Nguyen, *Computational intelligence and its applications: evolutionary computation, fuzzy logic, neural network and support vector machine techniques.* World Scientific, 2012.

[19] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural networks and learning machines.* Pearson Education Upper Saddle River, 2009, vol. 3.

[20] R. Kohavi and J. R. Quinlan, "Data mining tasks and methods: Classification: decision-tree discovery," in *Handbook of data mining and knowledge discovery.* Oxford University Press, Inc., 2002, pp. 267–276.

[21] C. E. Rasmussen and C. Williams, *Gaussian processes for machine learning.* The MIT Press, 2006, vol. 2, no. 3.

[22] "Weka," http://www.cs.waikato.ac.nz/ml/weka/index.html (accessed on: May 2016).

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[24] Y. Chen, B. Yang, and J. Dong, "Nonlinear system modelling via optimal design of neural trees," *International Journal of Neural Systems*, vol. 14, no. 02, pp. 125–137, 2004.

[25] Y. Chen, B. Yang, J. Dong, and A. Abraham, "Time-series forecasting using flexible neural tree model," *Information sciences*, vol. 174, no. 3, pp. 219–235, 2005.

[26] R. Poli, W. B. Langdon, N. F. McPhee, and J. R. Koza, *A field guide to genetic programming*. Lulu. com, 2008.

[27] Q. Shou-Ning, L. Zhao-lian, C. Guang-qiang, Z. Bing, and W. Su-juan, "Modeling of cement decomposing furnace production process based on flexible neural tree," in *Information Management, Innovation Management and Industrial Engineering, 2008. ICIII'08. International Conference on*, vol. 3. IEEE, 2008, pp. 128–133.

[28] Y. Chen, P. Wu, and Q. Wu, "Foreign exchange rate forecasting using higher order flexible neural tree," *Artificial Higher Order Neural Networks for Economics and Business, IGI Global Publisher*, pp. 94–112, 2008.

[29] B. Yang, Y. Chen, and M. Jiang, "Reverse engineering of gene regulatory networks using flexible neural tree models," *Neurocomputing*, vol. 99, pp. 458–466, 2013.

[30] Z. Chen, L. Peng, C. Gao, B. Yang, Y. Chen, and J. Li, "Flexible neural trees based early stage identification for ip traffic," *Soft Computing*, pp. 1–12, 2015.

[31] V. K. Ojha, A. Abraham, and V. Snasel, "Ensemble of heterogeneous flexible neural tree for the approximation and feature-selection of poly (lactic-co-glycolic acid) micro- and nanoparticle," in *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015*. Springer, 2016, pp. 155–165.

[32] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[33] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *Neural Networks, 1993., IEEE International Conference on*. IEEE, 1993, pp. 586–591.

[34] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[35] J. Zhang, C. Pei, S. Schiano, D. Heaps, and C.-Y. Wu, "The application of terahertz pulsed imaging in characterising density distribution of roll-compacted ribbons," *European Journal of Pharmaceutics and Biopharmaceutics*, 2016.

[36] S. Schiano, C.-Y. Wu, A. Mirtic, and G. Reynolds, "A novel use of friability testing for characterising ribbon milling behaviour," *European Journal of Pharmaceutics and Biopharmaceutics*, vol. 104, pp. 82–88, 2016.