Research Project

at the
Institute for Solids Process Engineering
and Particle Technology

Development of a machine learning model to predict particle size distributions in fluidized bed spray granulation

Agam Safaruddin (509699)

1st Examiner:	Prof. Dr.-Ing. habil. Prof. E.h. Dr. h.c Stefan Heinrich Technische Universität Hamburg
2nd Examiner:	Dipl.-Ing. Robert Kräuter Technische Universität Hamburg
Submission:	12.02.2025

Content

List of Figures

Nomenclature

Experiment

Symbol	Meaning	SI Unit
f_0	Frequency	Hz
v	Particle velocity	m/s
M	Magnification level	$-$
s	Interval width of spatial filters	m
m	Mass	kg
$\dot{m}$	Mass flow rate	kg/s
T	Temperature in Kelvin	K
P	Pressure in Pascal	Pa
$\dot{V}$	Volumetric flow rate	m^3/s

Machine Learning

Symbol	Meaning
Y	Label
x	Feature
N	Number of trees in Random Forest
T_i	Prediction of i_{th}
SS_{total}	Total sum of squares
SS_{res}	Residual sum of squares
R^2	Comparison of residual sum of squares to total sum of squares
Y_i	Predicted label
Y_{avg}	Average label
$\widehat{Y_i}$	Average predicted label

1. Introduction

The granulation process is a critical operation in industries such as pharmaceuticals and food production, and controlling the particle growth is essential for achieving the desired product characteristics. Predicting particle growth during granulation presents a challenge due to the complexity of the process which includes nucleation, growth, agglomeration, breakage and other phenomena.

This project focuses on developing a machine learning model to predict the particle size distribution at any time step during the granulation process. The process utilized Microcrystalline Cellulose Pellets (Cellets® 200) and Sodium Benzoate as the materials. The data were collected from nine granulation experiments with varying parameters of temperature and air flow speed. The data from the sensors, including the systems status, and process parameters provided the foundation for the training and testing of the machine learning model.

The predictive models are evaluated and are further tested on unseen experiment data to assess its generalizability. Two approaches are used, training and testing models on individual experiments and combining multiple experiments in the training dataset. The results of these approaches are compared to identify the suitable strategy for the model's predictability. This approach aims to enhance real time monitoring and control of the particle size, which in turn would enhance product quality and process efficiency.

2. Fluidized Bed Spray Granulation

Granulation is a process of enlarging particle grains or granules from powdery or solid substances. The process converts fine grains into flowing and dust free granules that are easy to compress into desired shapes. It is one of the most significant unit operations in the production of pharmaceutical dosage form, mostly tablets and capsules [1]. Beyond the pharmaceutical industry, this process is widely used in other sectors, including the food industries, where it could create products such as instant culinary powders and flavor enhancements. In the field of chemical engineering, granulation is used to produce fertilizers, detergents, and other chemical-based products by ensuring precise particle size distribution.

There are two types of granulations: wet and dry granulation. Wet granulation involves liquid additive to add layers onto the particles, then drying to solidify the grown particles. This method is preferred if the materials used in this process can handle moisture and heat. Dry granulation, on the other hand, forms the materials under pressure without any liquid additives and therefore it is suitable for heat or moisture sensitive products. [2]

Fluidized bed spray granulation is one of the methods of wet granulation. The process involves passing air through a bed of powders to suspend and mix the particles in the air, while a liquid solution is sprayed onto them. The particles would stick together or grow in size to create granules, which are then dried simultaneously in the same bed [3]. This method offers an easier handling of solids, a thorough mixing of particles, and good heat mass transfers between the bed and the liquid additive [4]. The system that was used in this experiment is called GlattGF3 (ProCell LabSystems) from Glatt Ingenieurtechnik GmbH.

2.1. Digital Twin

Based on the trilateral project called *Twin Guide: Digital Twins for Autonomous Control of Fluidized Beds* by TUHH SPE, Fraunhofer IFF, and Pergande Group. A Digital Twin is to be created and a part of it is an ML model to predict the output of the process.

2.2. Existing approach

Mechanistic models in Spray granulation are based on physical principles, such as mass and energy balances that can describe the dynamic behavior of the granulation process. These models aim to predict particle growing size and structure by considering the interactions between the liquid solution and the particles. The mass balance tracks the inputs, outputs, and internal transformations of material within the granulation process. Layering growth is a granulation mechanism where material accumulates around a particle [5].

These models integrate both mass balance principles and layering growth to enhance the ability to control and predict outcomes in spray granulation processes, by controlling the parameters such as spray rate, temperature, and airflow speed, then predicting its behavior in a larger-scale equipment, while ensuring consistent granule properties [6].

In this project, the focus shifts from mechanistic modeling to a data method approach using machine learning. The parameters of the experiment are the temperature and air flow speed, which result in different sensor data in the experiment. Using the sensor's data of the experiment such as temperature, humidity, pressure, and the particle sizes, etc., and with those data a model can be trained to predict the particle size at any given moment of the spray granulation process.

2.3. Experiment process

Nine experiments were done according to the design of the experiment and data were recorded via the sensors [Figure 1]. This would ensure different humidity levels inside the fluidized bed to see if effects as spray drying or agglomeration becomes more significant.

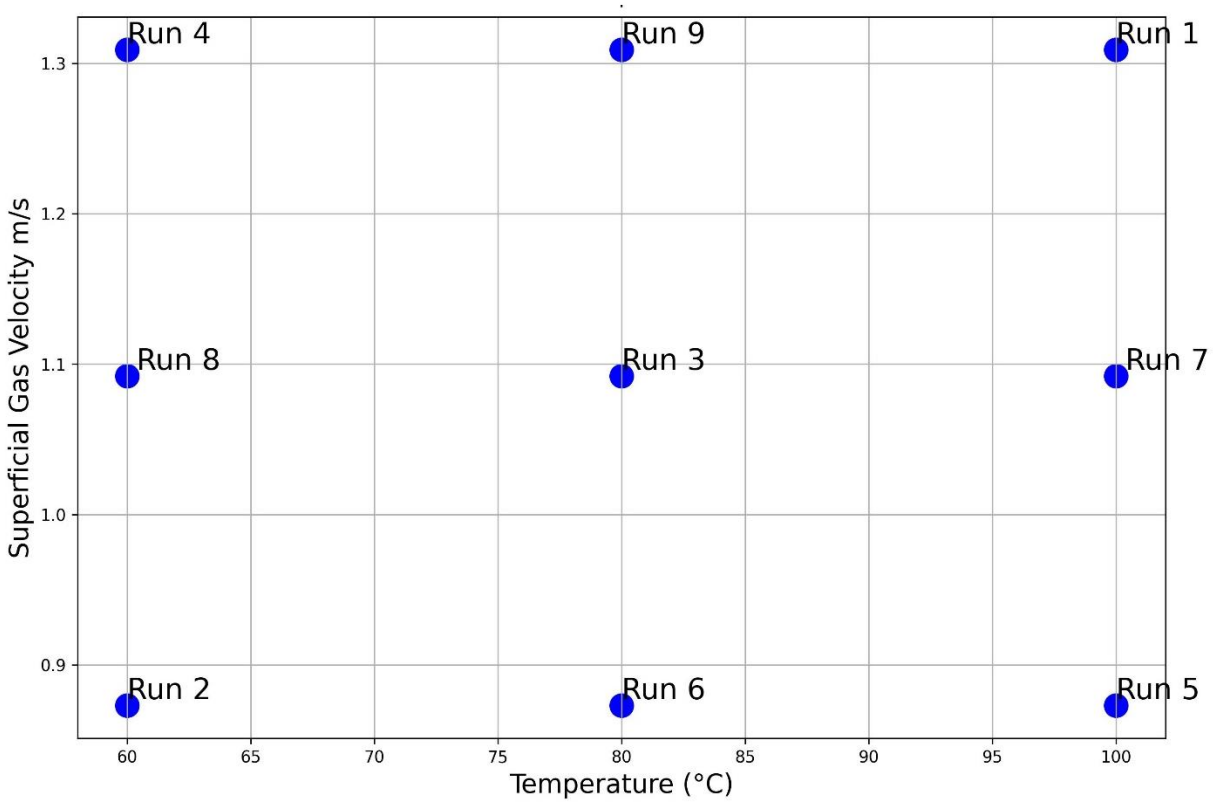

Figure 1: Design of experiments showing the fluidization temperature and the fluidization speed.

Materials of experiment:

- Cellets-200 (Microcrystalline Cellulose Pellets), from JRS-Pharma
- Sodium Benzoate
- Distilled Water

The Cellets-200 is made out of Microcrystalline Cellulose with a particle size of 200 Micrometer, it is a neutral carrier for a layering process of any material [7]. The Sodium Benzoate was mixed with distilled water to create the liquid solution, that would be sprayed onto the Cellets while it was being fluidized. The amount of solution made was based on how long the experiment would run. Using a spray rate of

$2.185\ g/min$ with $30\%\ solids\ mass\ concentration$ for $5\ hours$ experiment runtime, would require $1500g\ Sodium\ Benzoate + 3500g\ distilled\ water$.

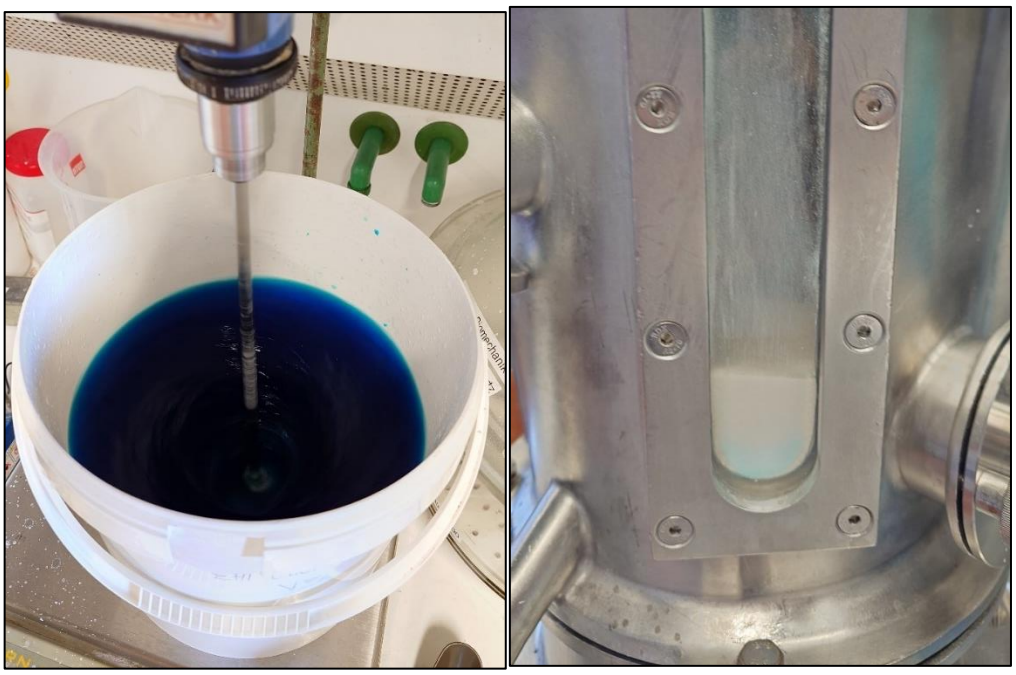

Figure 2: Mixing Sodium Benzoate Solution (Left). Cellets-200 inside Process Chamber (Right).

The GlattGF3 system was fitted with the Vario 3 Insert [Figure 3], with a working volume of $2.5 - 10\ Liters$ and a capacity of $0.4 - 4\ kg/h$ [8]. The system consists of multiple parts to run the granulation process. The inlet air is first heated up and the inlet air flow activated to their parameter's settings of the experiment. As soon as the temperature (measured via temperature probe) and the humidity are relatively stable, the first sample is then taken, afterwards the Spray Pump, with its connected pipe to the liquid solution, is then started. By using the top spray two-fluid nozzle, that connects both the pressurized air of $2.5\ Ba\ or\ 250\ kPa$, and the solution pump at $20.465\ g/min$ [Figure 4]. This creates a pressurized spray out

of the Spray Nozzle in order to atomize the liquid solution [Figure 5]. This would help the dispersion of the solution evenly to achieve consistent agglomeration. [9]

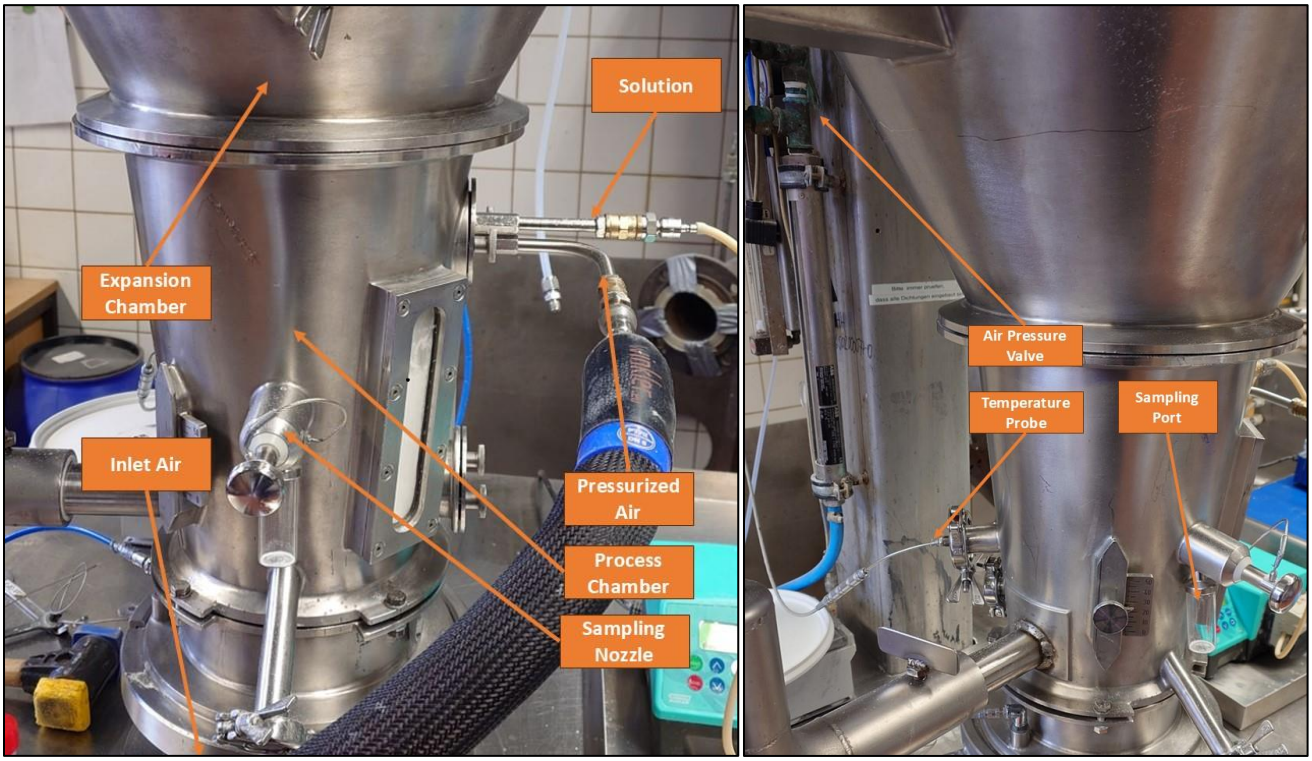

Figure 3: Vario 3 Insert fitted and connected to GlattGF3.

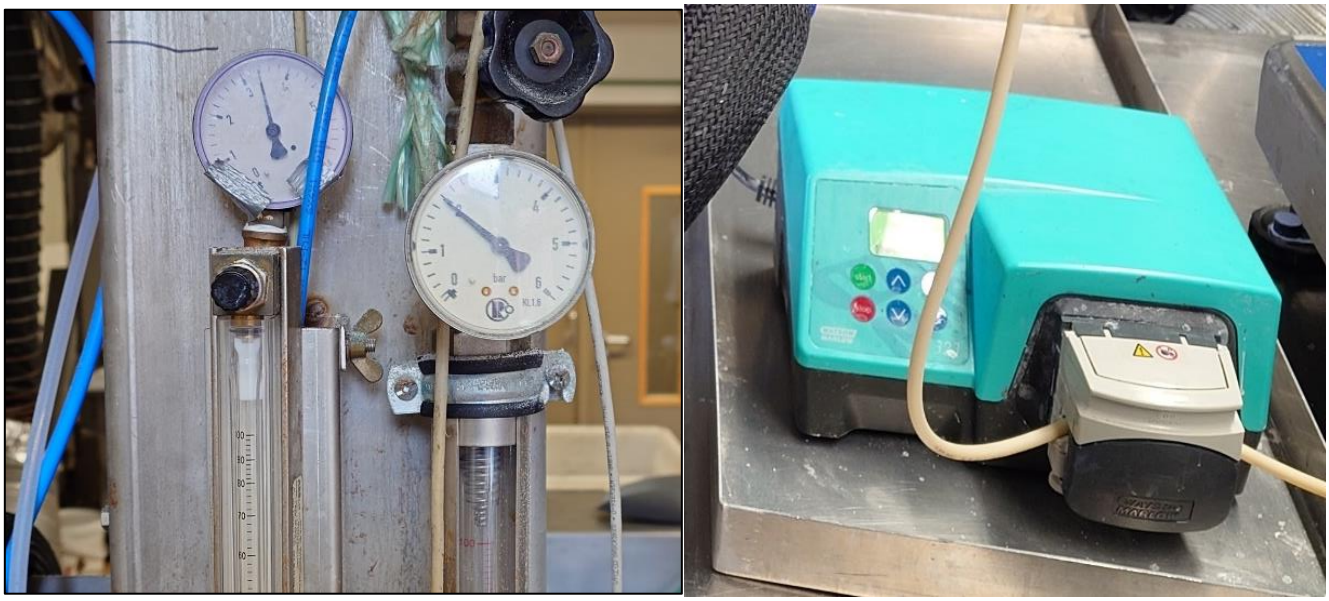

Figure 4: Air Pressure Valve (Left), Spray Pump (Right).

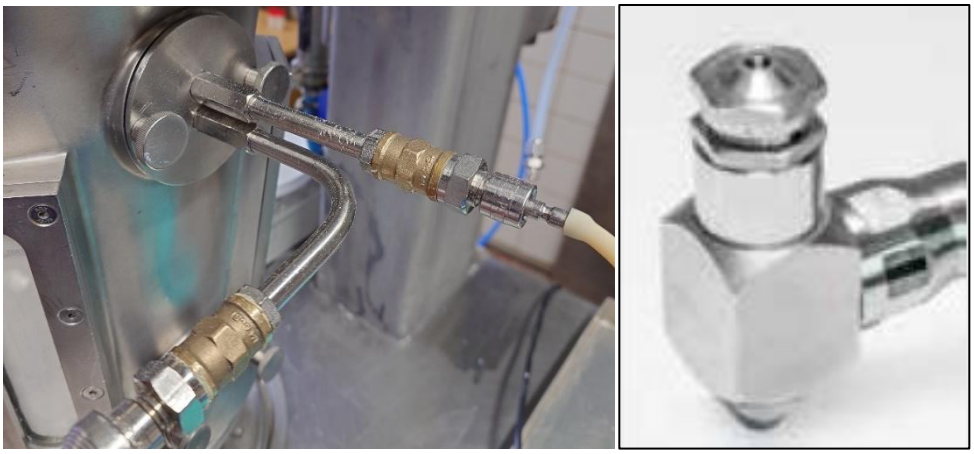

Figure 5: Two fluid nozzle in top spray configuration (Left). Spray Nozzle (Right) (Image from [1]).

During the run, samples were then taken every 17 minutes via the sampling nozzle (roughly 20ml per sample). The experiment ends until all the solution has been sprayed and the final sample would be taken. It is then followed by a drying stage and cooldown stage.

2.4. Data Gathering

The PLC of the GlattFG3 already logs data (temperature, pressure, flow, etc.) every second, whereas the particle size distribution was recorded with 2 methods; Parsum Probe and Camsizer.

The Parsum probe gathers data while the experiment runs (Inline Probe), it records the particle size distribution every 10 seconds. On the other hand, the Camsizer method (Offline) requires manual measurements; samples are poured into the Camsizer and it would then measure the particle sizes of said samples.

2.4.1. Methods

The equipment Parsum IPP 70-S [Figure 6], collects data using fiber-optic spatial filtering technique. The method involves using laser beam and shining it through a measuring cell and to a receiver made of 2 parts:

- **Single optical fiber:** Transmits the signal to a photodetector.
- **Spatial fiber-optic filter**: Maps the signal to 2 photodetectors.

Particles that pass through the measuring cell casts a shadow as they pass through the laser beam, when a particle crosses the single fiber, it creates a time impulse signal, and the width of that signal determines the particle's size, speed, and its random position relative to the single fiber [10]. After neglecting light diffraction and beam divergence, it is assumed that the shadow size matches the particles size. The particle velocity is then calculated by measuring the frequency (f_0) of the signal produces by the shadows, afterwards the particle velocity (v) is then calculated using the [Eq. 1]. s is the interval width of the spatial filters, M is the magnification of the imaging system. [11]

$$v = \frac{f_0 \cdot s}{M} \qquad \textit{Eq. 1}$$

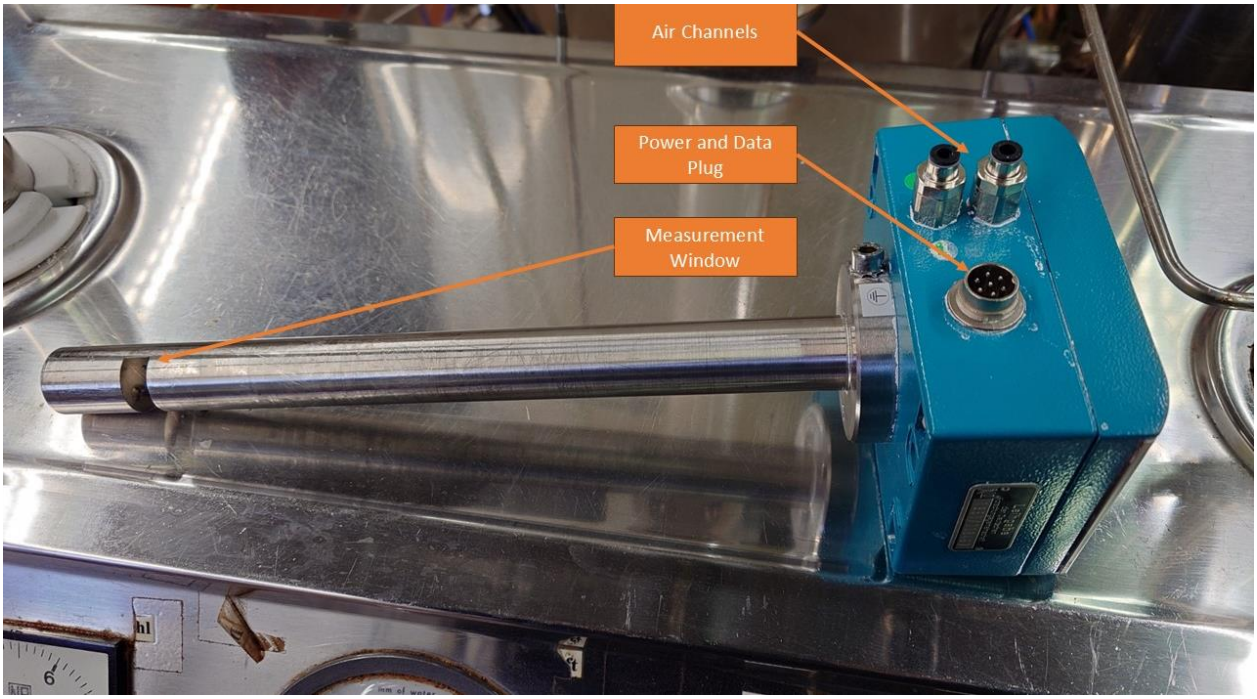

Figure 6: Parsum Probe IPP 70-S.

Another approach is by using an equipment Camsizer X2®, a particle size and shape analyzer [Figure 7].

The Camsizer X2® is an alternative method to measure particle sizes.

There are 3 modules that can be fitted into the Camsizer X2®:

- X-Jet, air pressure dispersion
- X-Flow, wet measurement
- X-Fall, gravity dispersion

The X-Fall can measure particle sizes up to $8mm$ and non-destructive (samples are recoverable). While the other two are destructive and X-Flow only goes up to $1mm$. The Parsum probe measured the biggest particle at $2.5119\ mm$, therefore, the X-Fall was chosen to measure the samples [12].

The samples that were taken during the experiment run [chapter 2.2], are then analyzed using the X-Fall module. The samples are poured a small amount (roughly $5\ grams$) onto the Chute and when the measurement is taken, the Hopper vibrates and slowly flows the samples into the X-Fall Chute. The Camsizer X2® then measures the sizes by using a Dual-Camera-Technology [Figure 8], that captures many images of the particles falling pass through its sight. With the captured images, the particles sizes are measured by its width, diameter, and length [Figure 9]. But in this experiment the X_{area} is of interest, because it can be taken as the average size combination of width and length of each particle. Each sample were analyzed three times to ensure reliability of the measurement.

Figure 7: Camsizer X2® (Left). X-Fall Module (Right) (Image taken from [2]).

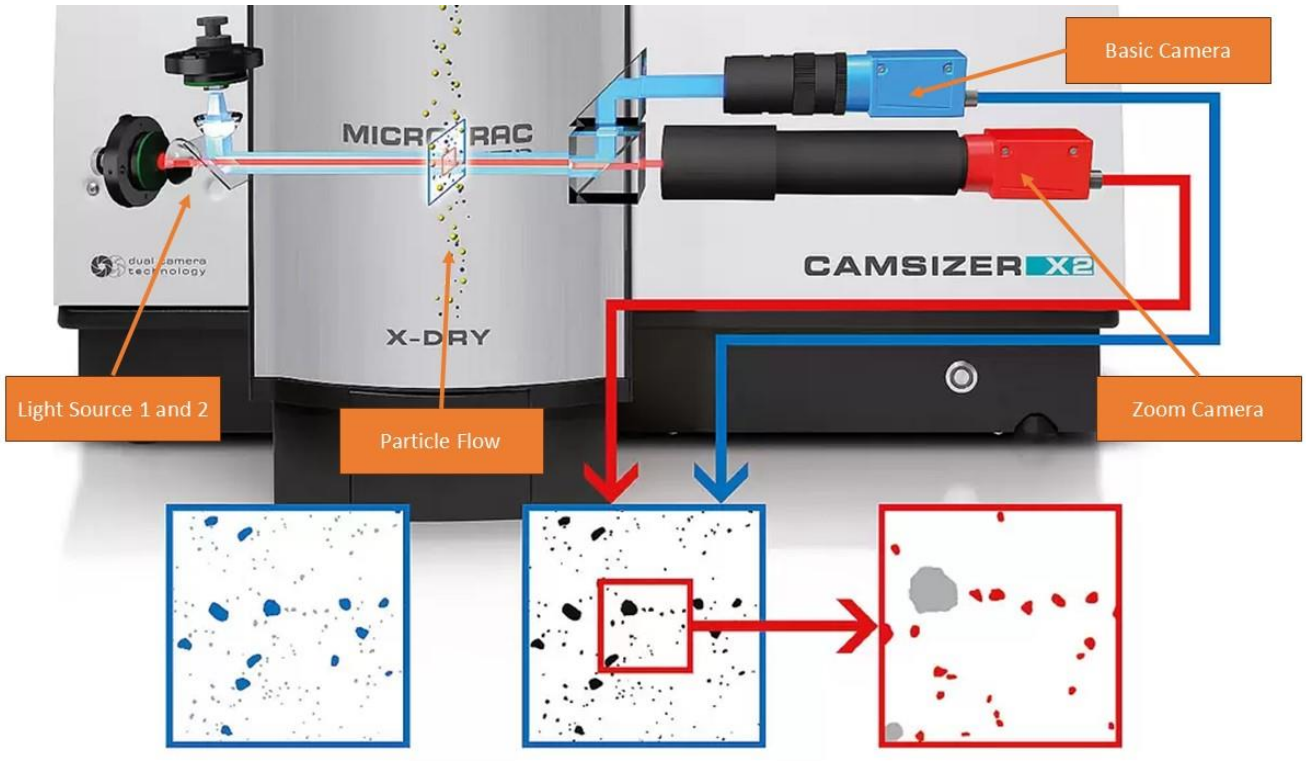

Figure 8: Dual Camera Technology of Camsizer X2® (Image taken from [2]).

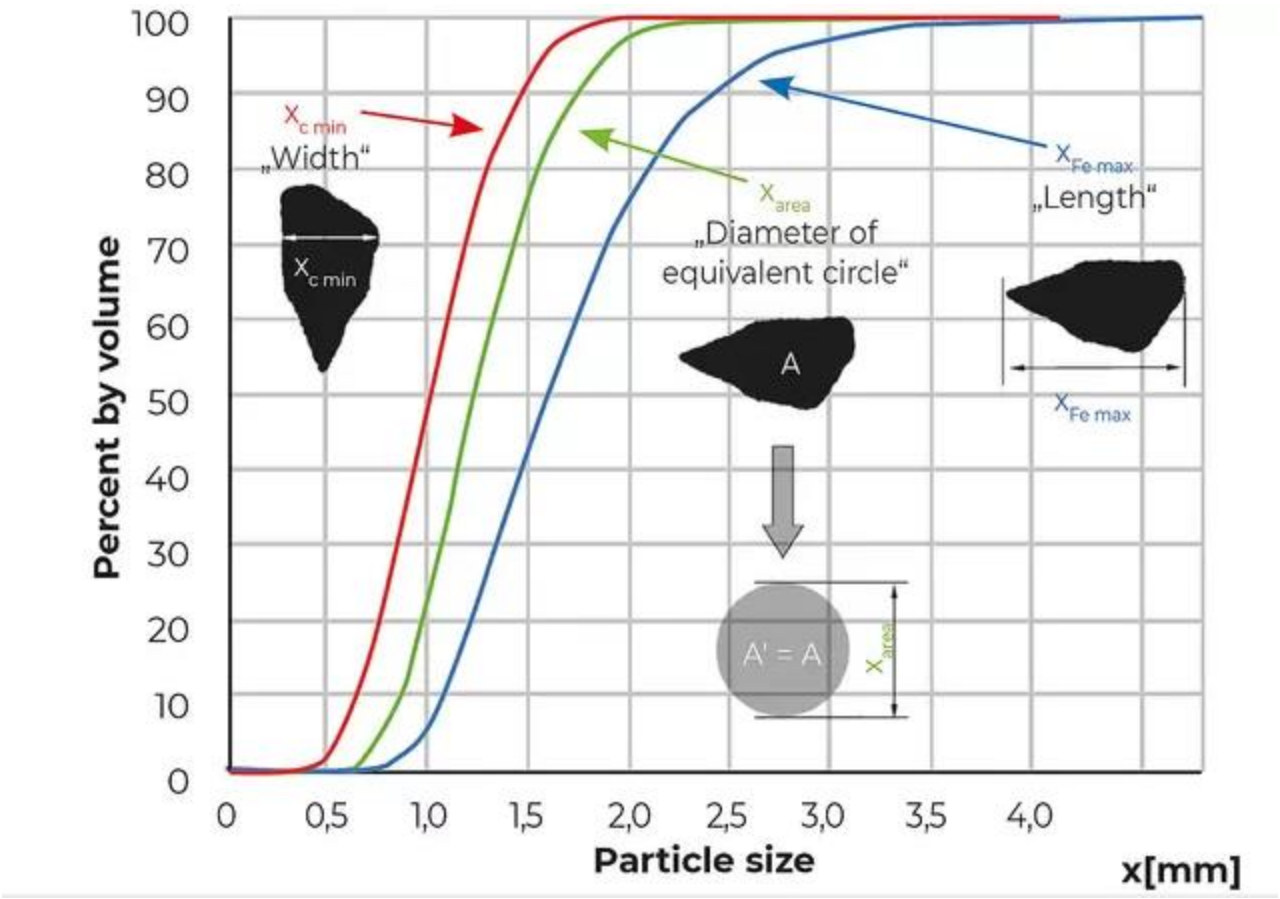

Figure 9: Camsizer X2 Particle Size measurement (Image taken from [2]).

2.4.2. Comparison of methods

Particle size analysis with laser diffraction (Parsum) interprets all measurements based on the diameter as if all the particles are assumed to be spherical in shape. Meanwhile, dynamic image analysis (Camsizer) provides multiple size definitions simultaneously.

2.5. Approach of Model Training

The development was done in Kaggle, an online python notebook environment with many available libraries supporting AI model development.

The bin particle sizes that were produced from the experiment showed a characteristically linear data, the particle sizes grew with respect to time. Hypothetically, if the growing size of single particle can be tracked, it would show a positive linear trend. In a multiple bin size output, it would also show a similar linear trend, but some linear trends would be negative and some would be positive. This is due to the changing of classes of particles sizes in respect to time. Therefore, a regression approach would be suitable the model to learn linear relationships.

In terms of classifying the bin sizes of particles, the Random Forest's method can classify the input data to their high probability or most likely size classes. Since the purpose of the model training is to predict the output bin particle sizes with respect to time, Random Forest Regression can predict continuous value

that the input data produces. In terms of handling complex multiple outputs, there are multiple methods for this such as Generalized Additive Models (GAM) and Gaussian Process Regression (GPR). But to handle any complex relationships that may occur in the output data, Random Forest Regressor has a higher predictive score in comparison with the GAM or GPR. [13]

Random Forest Regressor is an ensemble learning method of combining multiple decision trees, each tree is trained on a bootstrap sample of data and at each split, a random subset features is used. This creates a higher randomness among the trees, and as a result, it captures complex patterns in data while also maintaining generalizability [14]. All the output from all the trees is then averaged to create the output label Y, see [Eq. 2]. Where $T_i(x)$ is the prediction from the i_{th} tree and N is the total number of trees.

$$Y = \frac{1}{N}\sum_{i=1}^{N} T_i(x) \qquad \textit{Eq. 2}$$

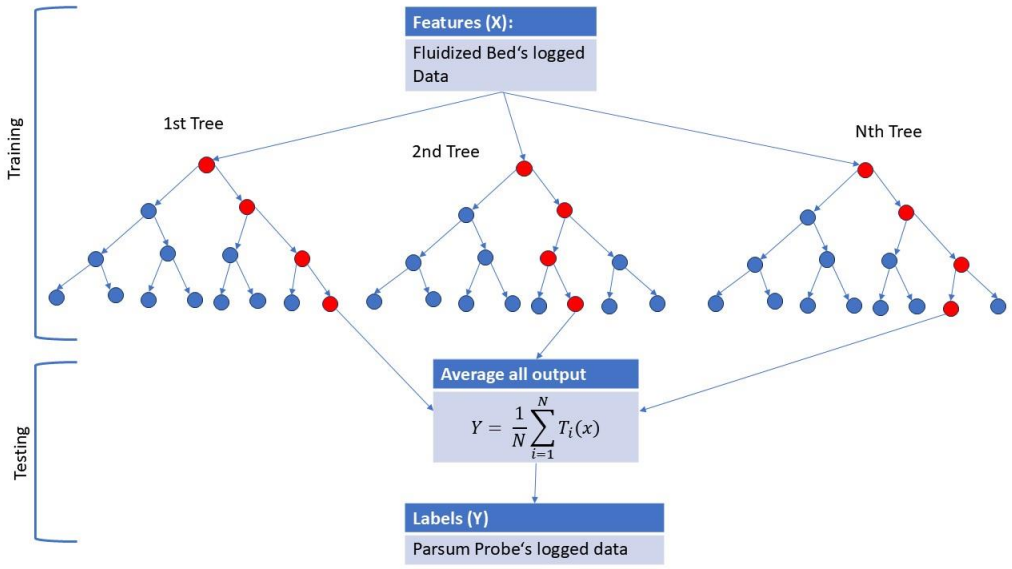

Figure 10: Random Forest Regressor Algorithm Structure (image made in PowerPoint).

Choosing suitable features for the example's input is necessary, because it directly influence the performance, accuracy, and interpretability of a model, that would output the class labels. One or more variables could cause or depend on the values of other variables and some could even be lightly associated with other values [15]. Therefore, a correlation matrix would be made to isolate the features that are highly correlated to the labels and help identify the strength of the correlation between all features [16]. The values in the matrix [Figure 11] represent the correlation as follows:

- 1: Maximum positive correlation, positive values represent a change in variable in the same direction to its comparison feature.
- 0: No correlation, zero or close to it means there is no correlation at all.
- -1: Maximum negative correlation, negative values represent a change in variable in the opposite direction to its comparison feature.

For example, the mass in the bed increases while the mass of the liquid solution decreases as the time runs. This represents a negative correlation value between the two features.

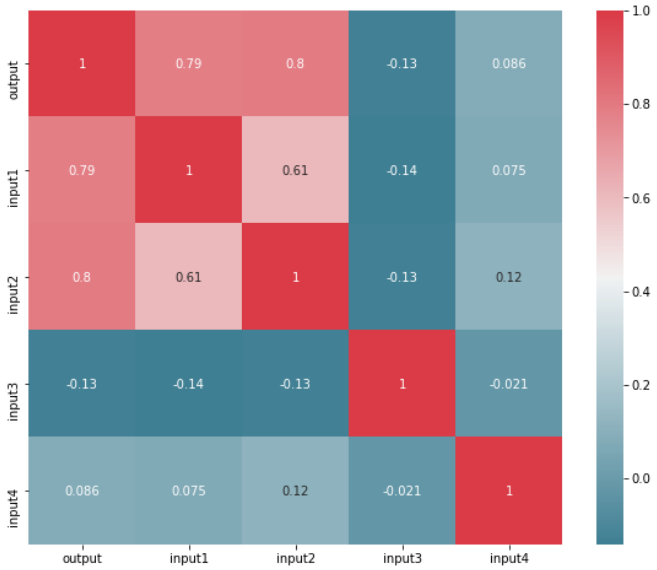

Figure 11: Correlation Matrix example (image from [16]).

Feature Importance is another method on identifying the impacts the features in Machine learning by assigning scores to it. This score indicates the relative importance of each feature in the model's prediction and it can also be used to rank features by their impacts to the model's predictive power. This gives insights to the relationships between features and labels and one could potentially filter out features that has a correlation close to zero as they have no impact between them and the output label. [17]

2.6. Model Evaluation

ML model requires evaluation to measure and improve the resiliency, robustness, and reliability before such models are deployed as an operational capability [18]. Since the data output of the experiment is almost linear, R^2 score evaluation is suitable for evaluating it. R^2 score is Coefficient of determination, to evaluate the quality of regression in machine learning.

R^2 is a comparison of the residual sum of squares (SS_{res}) with the total sum of squares (SS_{total}) [19]. A trained model would have many "missed" or close prediction to the real data, these distances between the real data to the predicted are calculated and summed up [Eq. 4]. This summed value is then compared to the distances between the real data to the mean data (average) [Eq. 3]. [Figure 12]

$$SS_{total} = \sum_{i=1}^{n} (Y_i - Y_{avg})^2 \qquad \textit{Eq. 3}$$

$$SS_{res} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \qquad \textit{Eq. 4}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{total}} \qquad \textit{Eq. 5}$$

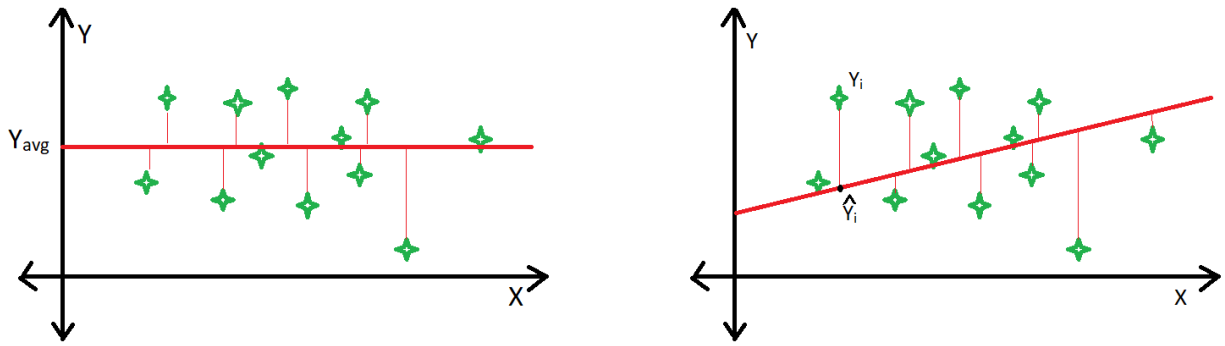

Figure 12: Average fitted model (Left). Best fitted model (Right) (Image from [20]).

3. ML Model Development

There was a total of 9 csv files developed from the experiment and each contained the machine's status and the Parsum Probe's size detection throughout the experiment.

3.1. Data Pre-Processing

The data contain many static data which represents the fluidized bed machine in the warming up phase, shutting down phase, and the clogging of the liquid solution spray. Some clogging occurred during the experiment run, where the liquid solution would leak out of the Brass Pipe Connector [Figure 13], due to the sodium benzoate solution drying up and clogging up the Spray Nozzle [Figure 5].

Figure 13: Brass Pipe Connector of the two-way nozzle where the blockage normally occurred.

S	T	U	V	W	X
setpoint_spray_rate[kg/s]	bed_temperature_sensor_position	fluidizatic	heater_status	nozzle_status	parsum_status
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	on	clogged
7	0.020465116	inside	on	on	clogged	clogged
7	0.020465116	inside	on	on	clogged	clogged
7	0.020465116	inside	on	on	clogged	clogged
7	0.020465116	inside	on	on	clogged	clogged

Figure 14: How the clogged data values were labeled in the csv file.

3.1.1. Cleaning

The data has been filtered to only include data where all equipment was fully functioning. That means all data are in non-idle state. The pump speed and spray rate columns are constant values, the same as the rest of the 9 experiments, and directly correlated to status of the nozzle [Figure 15]. Therefore, the status of the nozzle is used to represent them, whether they are 'on' or 'off'. The same goes to the other constant values such as Parsum status and position of the bed temperature probe, because those are statuses of data being recorded [Figure 16]. This ensures the necessary data being used, where the machine is running and data is recorded. Not only that, but also would help reduce the load in the performance when handling big data.

setpoint_pump_speed[1/s]	setpoint_spray_rate[kg/s]	bed_temperature_sensor_position	fluidization_status	heater_status	nozzle_status
0.0000	NaN	1	on	on	0
0.0000	NaN	1	on	on	0
0.0000	NaN	1	on	on	0
0.3167	0.020465	1	on	on	1
0.3167	0.020465	1	on	on	1

Figure 15: Before the removal of columns and correction of spray rate.

	bed_temperature_sensor_position	nozzle_status	parsum_status
3075	1	0	1
3076	1	0	1
3077	1	0	1
3078	1	1	1
3079	1	1	1

Figure 16: Redundant Columns removed.

	bed_temperature_sensor_position	nozzle_status	parsum_status
10439	1	1	1
10440	1	1	1
10441	1	1	1
10442	1	1	1
10443	1	1	1

Figure 17: non-running experiment rows removed.

setpoint_spray_rate[g/s]	bed_temperature_sensor_position	fluidization_status	heater_status	nozzle_status	parsum_status	sprayed_total[g]
0.000341	1	on	on	1	1	1.049178
0.000341	1	on	on	1	1	1.049519
0.000341	1	on	on	1	1	1.049860
0.000341	1	on	on	1	1	1.050202
0.000341	1	on	on	1	1	1.050543

Figure 18: Corrected setpoint_spray_rate[g/s] and added total mass liquid solution sprayed.

The 'setpoint_spray_rate[kg/s]' was discovered to be an error. The liquid solution spray rate was set to $20 g/min$ in the experiment, but the column name's rate is at kg/s and the values are actually at kg/min in the data [Figure 15]. Therefore, this was simply changed by simply converting using the conversion [Eq. 6] to get an appropriate rate of g/s [Figure 18].

$$1\ g/\mathrm{s} = \ 0.06\ kg/min \qquad \textit{Eq. 6}$$

Time was replaced with the accumulation of the solution in the bed, by simply accumulating the solution mass every second, then added to the data as a new feature called 'sprayed_total[g]'. This would be the new key variable (index variable) for the data [Figure 18].

The label data (Parsum recorded particle size) contains few empty columns for some of its size classes. The lowest recorded particle size of all the 9-csv data is $39.8 \times 10^{-6} m$ or $0.0398 mm$, and the size classes below them were all zeros and their naïve values would always be zero. Therefore, the label columns are taken from 1 bin below that at $31.6 \times 10^{-6} m$ up to $2511.9 \times 10^{-6} m$. By removing the empty or zero labels, this provides a huge increase in performance during handling and training.

3.1.2. Filling empty data

The Parsum-probe logged data every 10 seconds [Figure 19], therefore there are 9 seconds of empty values interval between each log and the interval is constant. Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) was used to fill in the gaps of an almost non-linear data, PCHIP is also designed to not overshoot like other interpolation method such as Cubic Splines.

In Figure 20, only a portion of the data is shown as an example to clearly show the plots before and after the interpolation. The full data plot is much denser and the difference between before and after the interpolation can't be seen clearly. Hence, only a portion of the data is shown.

Parsum_particle_rate[1/s]	Parsum_volume_based_particle_size_cumulative_distribution_Q3_at_particle_size_7.9m*10^-6[%]	Parsu
NaN	NaN	
NaN	NaN	
NaN	NaN	
300.0	0.0	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
NaN	NaN	
171.0	0.0	
NaN	NaN	
NaN	NaN	

Figure 19: Parsum's logged data gaps.

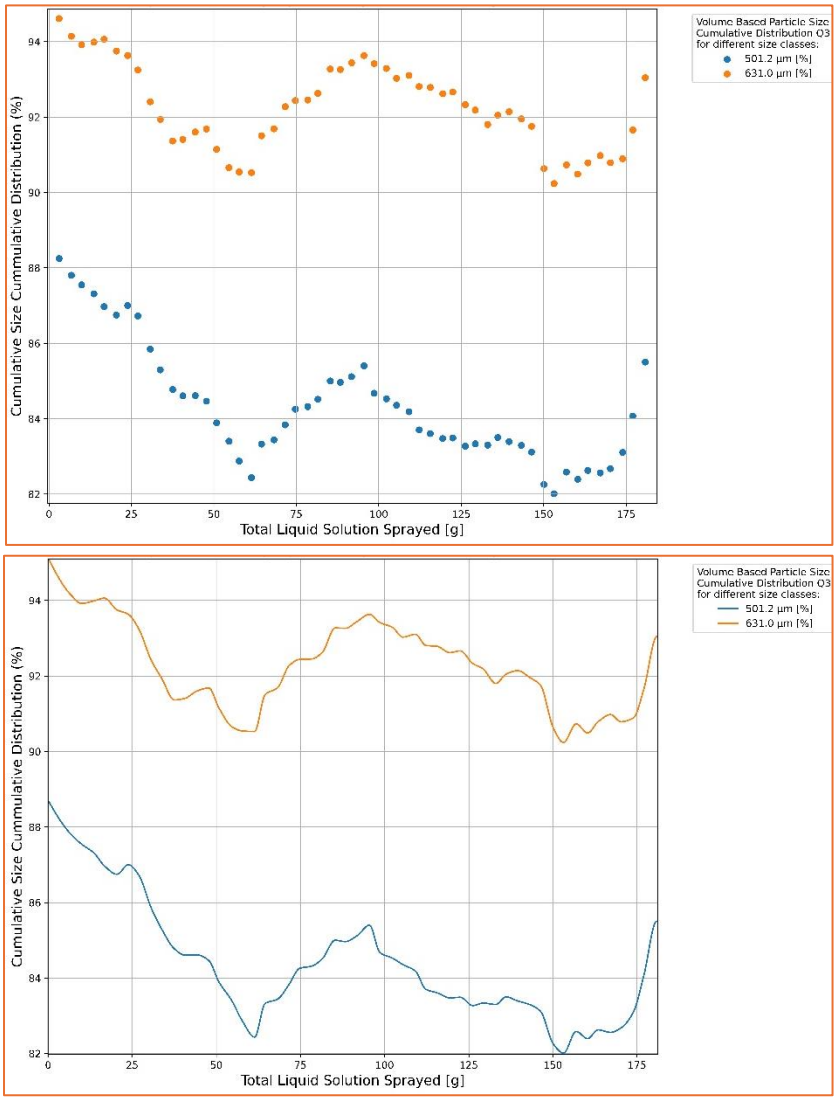

Figure 20: Sparse Data of the particle sizes (Above). Missing data filled using PCHIP (Below).

3.1.3. Analyzing features and labels

Correlation matrix (heatmap) was calculated and graphed on the data to see the most effective features on their labels [Figure 21]. But the heatmap showed different correlation values for every csv data; the

correlation between features were not constant for every experiment. The feature's correlation value seems to be changing in terms of the highest to lowest values, in some data a feature is at the highest value and some other data it is at the lowest, see [Appendix] for more details. Therefore, selecting the features for training would be different for every csv data in order to train a model out of it. Another approach was taken, feature importance calculation.

Feature importance was then calculated using Random Forest Regressor, where the features and the labels are trained. The features are scored based on their contribution to the labels (model's prediction) and then these scores are graphed [Figure 22]. But the result was the same as the correlation matrix, where all the features are inconsistent and changing for every csv data. The features seem to also be in different orders for every data. Although they are swapping with each other in ranks, there are 12 features that always occurs in every data in both Correlation matrix and Feature importance, therefore in both [Figure 21] and [Figure 22] only the 12 features were left after filtering out constant data, zero correlation values, and zero feature importance:

- Total Solution sprayed
- Inlet humidity before heating
- Bed temperature
- Outlet temperature
- Pressure drops from process chamber to filters
- Parsum particle rate
- Wet air flow rate before heating
- Inlet air temperature before heating
- Inlet air temperature after heating
- Outlet air humidity
- Pressure drops from wind box to process chamber

Since the two methods couldn't help in isolating the biggest factors of the output labels, mixing multiple data could be a way to average out the features across all the data. This could be done in multiple ways; the first method would be single data training, where each data of the 9 experiment is trained and tested. Second method would be a mixture of the data where 8 of the 9 data are combined and trained, but tested on 1 of the 9 data. This is then done with all the 9 data experiments, where each of them would take turns on being the testing dataset for the other 8 training dataset. Third method would also be a mixture of the 8 data, but with a partial mixture (20%) of the 1 data. Mixing all of them together however is not recommended, since it would not generalize too well, and also be stuck in the parameters of experiment's temperature and air flow rate settings, and could not predict any input outside of those parameters.

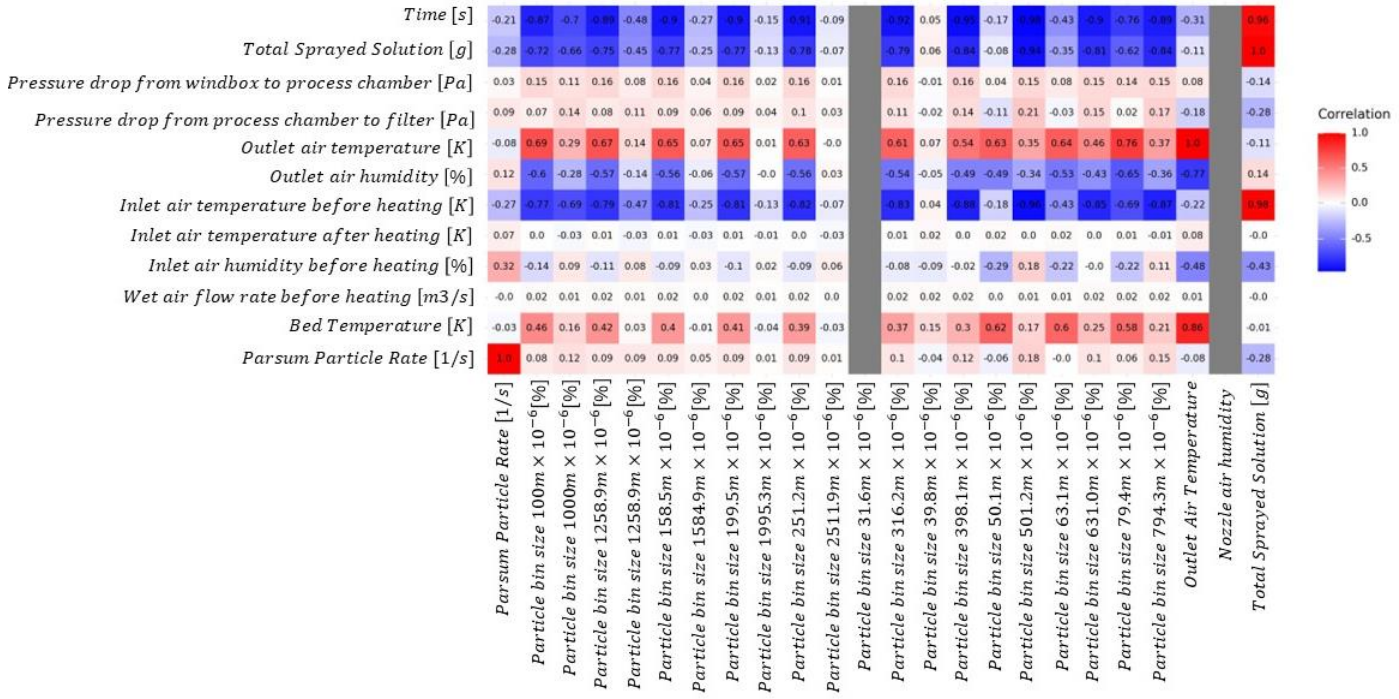

Figure 21: Correlation Matrix of experiment 5.

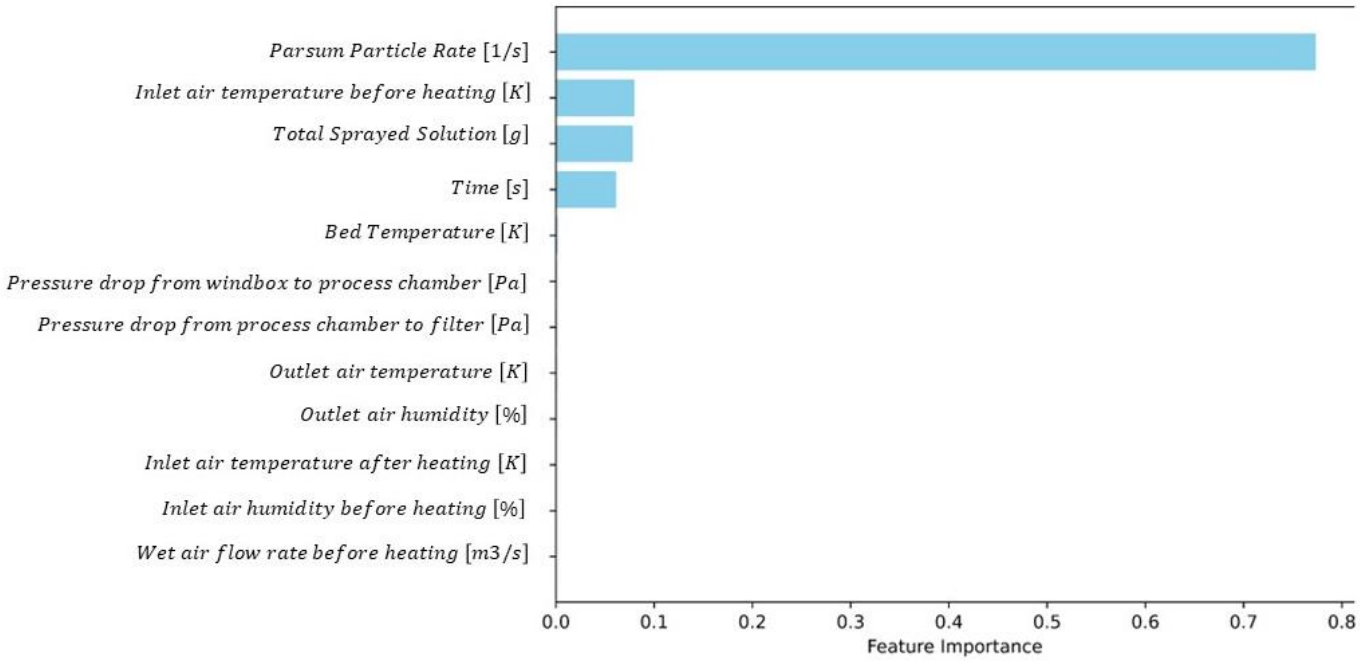

Figure 22: Feature Importance of Experiment 5.

3.2. Training and testing each individual data

The data was split into a training and test set with an 80/20 ratio for each dataset and then trained with Random Forest Regressor, creating 9 models in total. Each model number indicates the experiment data that it was trained on, see [Figure 1]. They were each evaluated using R^2 score evaluation and all had scores above 0.99, it can be seen, where the model is evaluated with test set [Table 1]. However, each of this

model cannot predict other data experiments with high precision, because it would result in drastically lower or even a negative R^2 scores, the score values marked in green are the positive scores. More elaboration in chapter 4.1.

Since there are 9 CSV data files, the script algorithm was simply looping through the 9 files. In each loop they are cleaned, split, and saved, to save time when reusing the model in the future. Each model is then loaded and evaluated.

Table 1: R^2 Scores of each model tested on all dataset.

	R^2 Score								
Model no.	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9
1	0.9995	-0.0048	0.0377	-0.6524	-1.5977	-6.1646	-2.2292	-5.6530	-1.2604
2	-0.351	0.999	-0.100	-0.112	-2.023	-4.846	-2.398	-5.860	-0.665
3	-1.235	-2.470	0.9997	-2.5141	-2.777	-6.0156	-3.1652	-5.5689	-0.7131
4	-0.2965	0.2707	-0.8687	0.9998	-1.1728	-6.1322	-1.8336	-6.5146	-1.6814
5	-16.32	-12.41	-9.4885	-19.84	0.9998	-0.8254	-0.3008	-0.7163	-2.8227
6	-18.89	-14.82	-8.3242	-29.98	-0.4654	0.9998	0.3735	0.5937	-2.3759
7	-22.21	-59.43	-18.24	-44.52	-1.181	-2.404	0.9998	0.0478	-5.6293
8	-20.981	-14.272	-9.7015	-29.566	-0.1384	0.4055	0.3732	0.9973	-2.6074
9	-4.287	-7.622	-1.5973	-12.127	-2.0867	-1.6241	-0.7299	-4.4331	0.9986

3.3. Training on eight data and testing on the ninth

In this method, the combined models were made by concatenating 8 datasets, and this would become the training set, then the last CSV file data would become the testing set. Since that the split is between the 8 training datasets and the 1 testing dataset, the splitting ratio would be inconsistent. Therefore, the number of examples of each train and test would be noted. There were 9 combinations of this for each CSV data [

Table 2], and the naming scheme of these combinations and the referred model name can be seen in both of the table below. Each data is trained, tested, and evaluated with R-Square scoring [Table 3].

The script algorithm was similar to the previous method, it loops through the 9 CSV data file. Each iteration would then be processed the same, then set into the testing set, afterwards all the rest of the CSVs that are not of that iteration number would then be concatenated together, then processed, and inserted into the training set.

Table 2: 9 Combinations of 1 vs 8 testing and training split.

Dataframe Name	Single Test set	Combined Training set
Dataset 9	9^{th}	1-8^{th}
Dataset 8	8^{th}	1-7 and 9^{th}
Dataset 7	7^{th}	1-6 and 8-9^{th}
Dataset 6	6^{th}	1-5 and 7-9^{th}
Dataset 5	5^{th}	1-4 and 6-9^{th}
Dataset 4	4^{th}	1-3 and 5-9^{th}
Dataset 3	3^{rd}	1-2 and 4-9^{th}
Dataset 2	2^{nd}	1,3 and 9^{th}
Dataset 1	1^{st}	2-9^{th}

Table 3: Result of model training, testing, and evaluation of 1 vs 8 dataset.

Combined Model Name	Dataframe Name	Data split of Train, Test, Feature X, and Label Y				R^2 Score
		X Train	Y Train	X Test	Y Test	
9	Dataset 9	105465	105465	15195	15195	-2.8137838
8	Dataset 8	106398	106398	14262	14262	-1.9178874
7	Dataset 7	104381	104381	16279	16279	-0.7747885
6	Dataset 6	105960	105960	14700	14700	-2.1301392
5	Dataset 5	110120	110120	10540	10540	-0.5920556
4	Dataset 4	103692	103692	16968	16968	-19.0925898
3	Dataset 3	110257	110257	10403	10403	-0.6362660
2	Dataset 2	107289	107289	13371	13371	-2.1601698
1	Dataset 1	111718	111718	8942	8942	-5.1330582

3.4. Training on eight datasets with a partial dataset from the ninth

In this method, it is the same as in chapter 4.2. But 20% of the ninth dataset would also be concatenated into the 8-training dataset, then they would be tested on the left over ninth dataset [Table 4]. Each data is trained, tested, and evaluated with R-Square scoring [Table 5].

Table 4: 9 Combinations of 1 vs 8 dataset, plus 20% of dataset 1.

Dataframe Name	Single Test set	Combined Training set
Dataset 9	9^{th}	$1\text{-}8^{th}$ + 20% 9^{th}
Dataset 8	8^{th}	1-7 and 9^{th} + 20% 8^{th}
Dataset 7	7^{th}	1-6 and $8\text{-}9^{th}$ + 20% 7^{th}
Dataset 6	6^{th}	1-5 and $7\text{-}9^{th}$ + 20% 6^{th}
Dataset 5	5^{th}	1-4 and $6\text{-}9^{th}$ + 20% 5^{th}
Dataset 4	4^{th}	1-3 and $5\text{-}9^{th}$ + 20% 4^{th}
Dataset 3	3^{rd}	1-2 and $4\text{-}9^{th}$ + 20% 3^{th}
Dataset 2	2^{nd}	1 and $3\text{-}9^{th}$ + 20% 2^{th}
Dataset 1	1^{st}	$2\text{-}9^{th}$ + 20% 1^{th}

Table 5: Result of model training, testing, and evaluation of 1 vs 8 plus 20% of dataset 1.

Combined Model Name with 20%	Dataframe Name	Data split of Train, Test, Feature X, and Label Y				R^2 Score
		X Train	Y Train	X Test	Y Test	
9	Dataset 9	108504	108504	12156	12156	0.6606136
8	Dataset 8	109250	109250	11410	11410	0.7361095
7	Dataset 7	107637	107637	13023	13023	0.8226085
6	Dataset 6	108900	108900	11760	11760	0.6578140
5	Dataset 5	112228	112228	8432	8432	0.6779078
4	Dataset 4	107086	107086	13574	13574	0.3832651
3	Dataset 3	112338	112338	8322	8322	0.5902589
2	Dataset 2	109963	109963	10697	10697	0.3783014
1	Dataset 1	113506	113506	7154	7154	0.3731891

4. Results and conclusion

4.1. Single trained model evaluation and inference

In each model, that were trained and evaluated individually within the same CSV file, the results of the R^2 score were very high as shown in [Table 1]. But if any of the model is used for inference of other CSV files, the result of the R^2 score would become negative and therefore causes bad predictions.

The experiment-5 (Temperature: 80°C, Air Flow: 100m^3/h) cumulative size distribution result can be seen in Figure 23, and in around $450g$ of total solution sprayed there is a big drop for all the bin sizes. The data cleaning process has removed a big chunk of this data, this removed data are the blockage that occurred during the experiment run. Hence the reason for the plot to have a huge drop.

During the blockage mentioned in chapter 3.1, the sizes of the particles continued to grow due to the moist chamber inside the process chamber. In the plot, the size classes would likely continue to go in a negative trend. Therefore, when some data are removed, the particle size before stopping and after continuing wouldn't connect seamlessly.

This drop causes multiple patterns throughout the inference. In Figure 24, model-5 was trained using experiment-5 data and used for inference on the features to predict the output label sizes, naturally the result is almost identical.

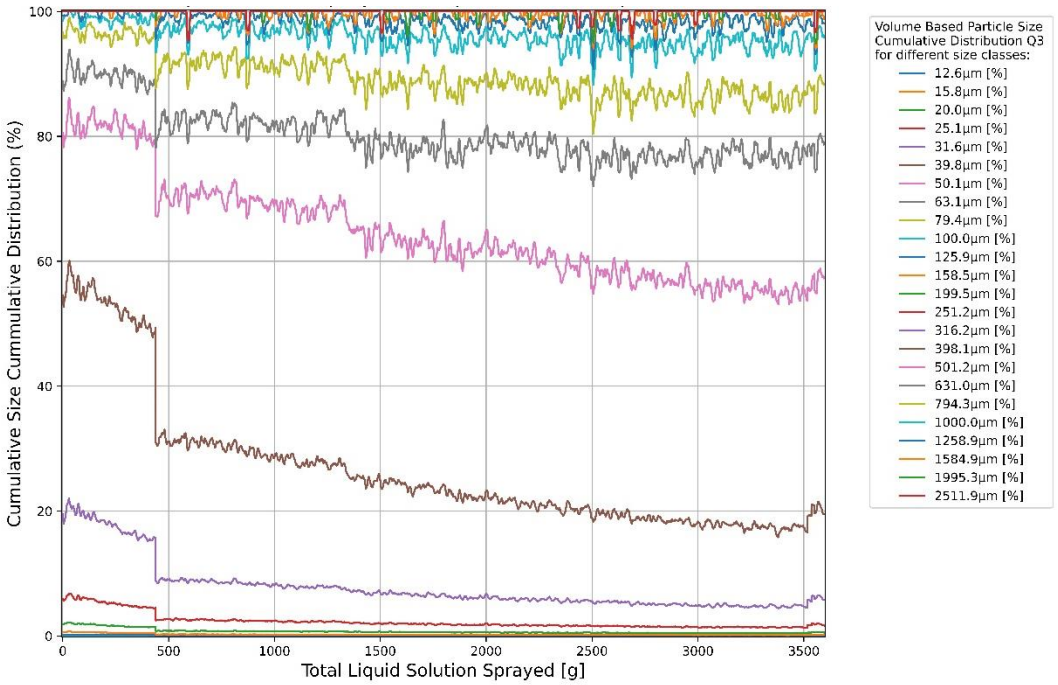

Figure 23: Real Data of Experiment-5.

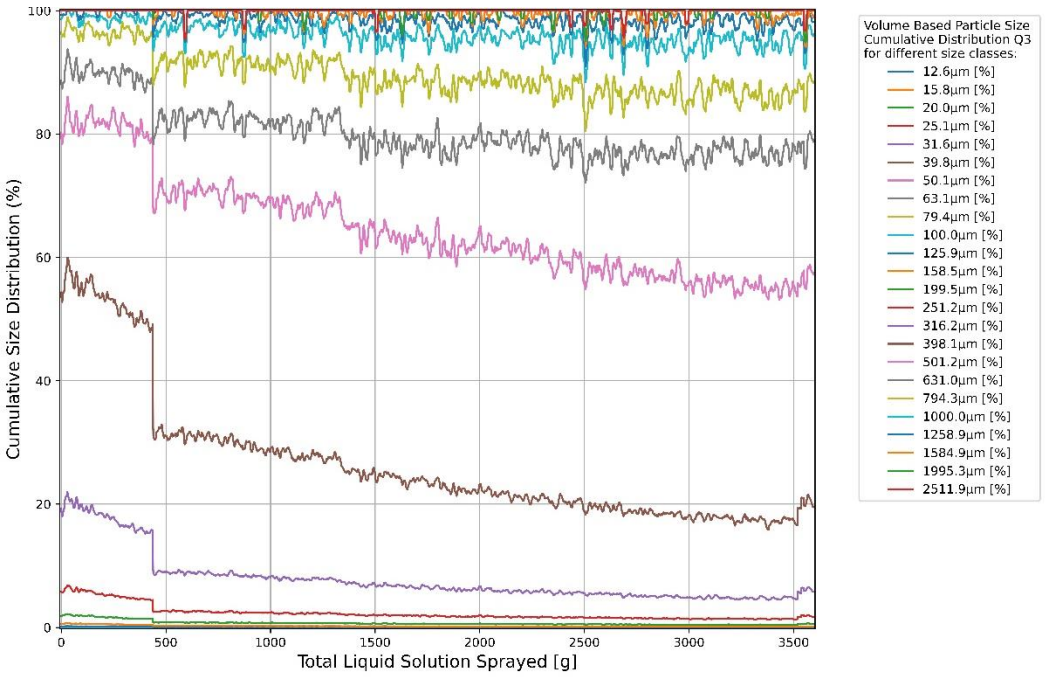

Figure 24: Prediction of experiment-5 using model-5.

Another inference was done to predict the input features of experiment-6 (Temperature: 80°C, Air Flow: 80m^3/h). It can be seen that the cumulative size distribution, predicted by the model, does follow the negative linear trend of the real result of experiment-6, however they are slightly offset and resulted in a negative R^2 score of -0.7883. As mentioned before regarding the big drops in experiment-5 data, it can be seen in Figure 25 there are multiple drops in the prediction results. This shows that removing rows of a characteristically linear data would have a significant impact to the linear regression approach in machine learning. A suggestive approach, is by using interpolation to fill in the data that was removed.

Another inference was done by using model-6 to predict experiment-7 (Temperature: 100°C, Air Flow: 100m^3/h), this experiment is much "cleaner" in the sense of no interruption in data logging. It yields much better results compare to model-5 with an R^2 score of 0.3729, although significantly lower than the desired R^2 score of above 0.9, it is still a positive value. The comparison of the real experiment-7 data and the prediction using model-6 can be seen in Figure 27. This supports the R^2 results of Table 1, but it may not be suitable at predicting other datasets. The reason for the model-6 to be able to have R^2 score for predicting experiment-7 and experiment-8 is due to the feature importance of the dataset of experiment 6, 7, and 8 as they have similar features that affects the learning output. These features could be taken into consideration when training any dataset of various parameters.

In Figure 30, the predicted Cumulative distribution (Q3) with model-6 fits very close to the real data of experiment 6, but the predicted data is slightly shifted to the right which indicates a slower growth to the real data. Figure 31 is the predicted data of experiment 7 using the same model-6, it shows a much bigger shift compared to predicting experiment 6. Figure 26, Figure 28, Figure 29 show much more details of each separate data of before and after running the experiment including the predicted data using model-6 on experiment 7.

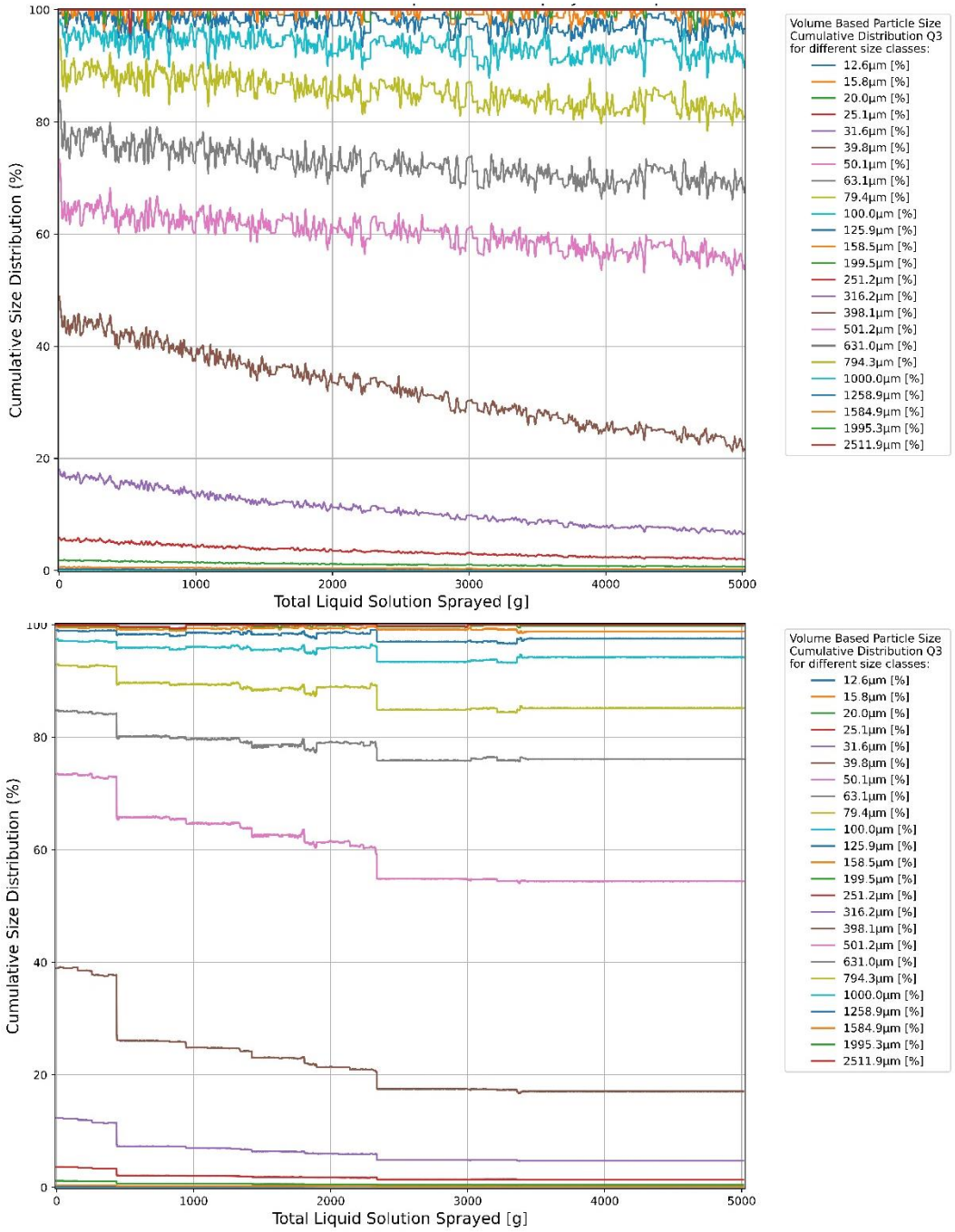

Figure 25: Comparison of real data (Above) of experiment 6 to the predicted data (Below) of experiment 6 using model 5.

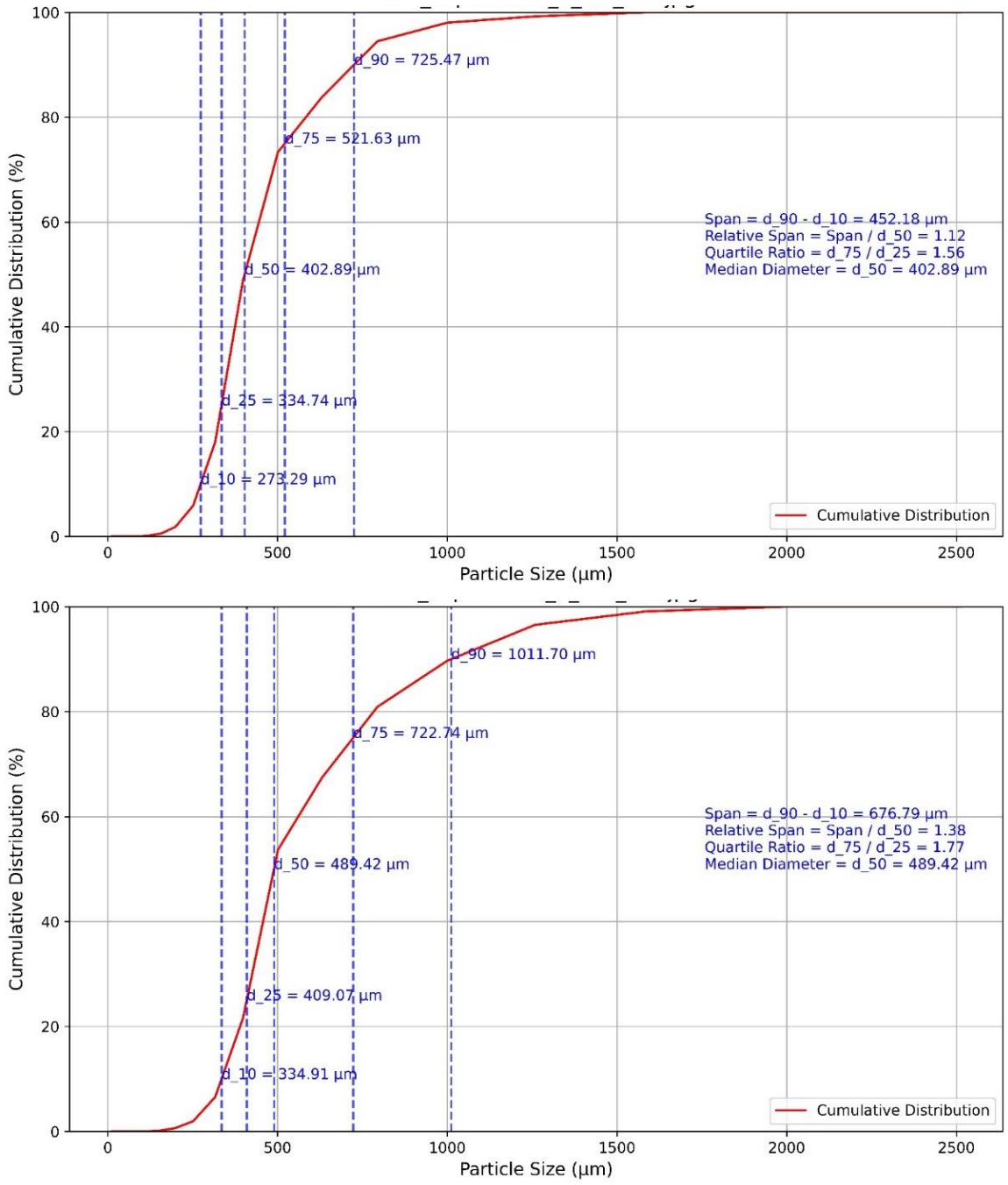

Figure 26: Before (Above) and after (Below) real data cumulative particle size distribution of experiment 6.

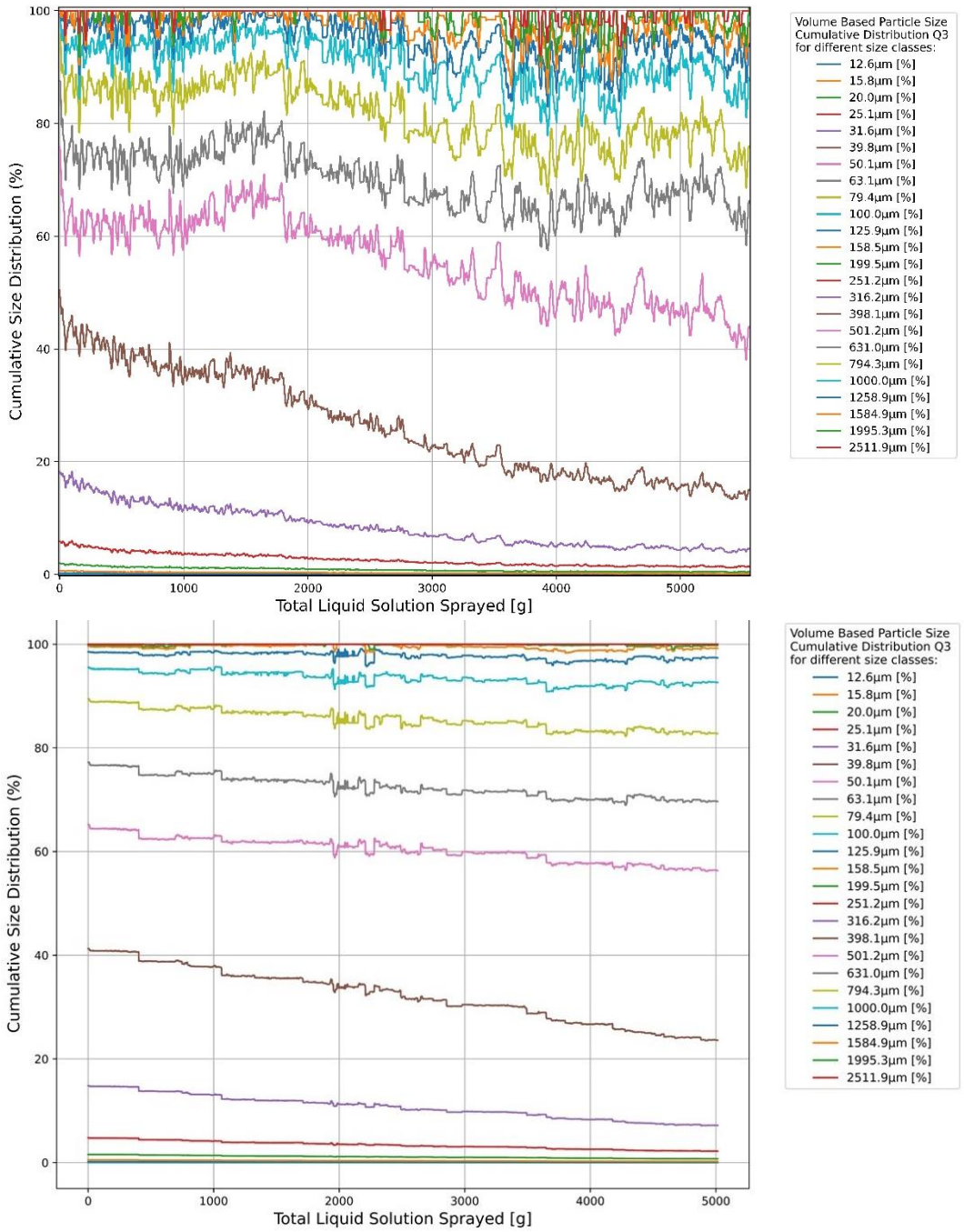

Figure 27: Comparison of real data (Above) of experiment 7 to the predicted data (Below) of experiment 7 using model 6.

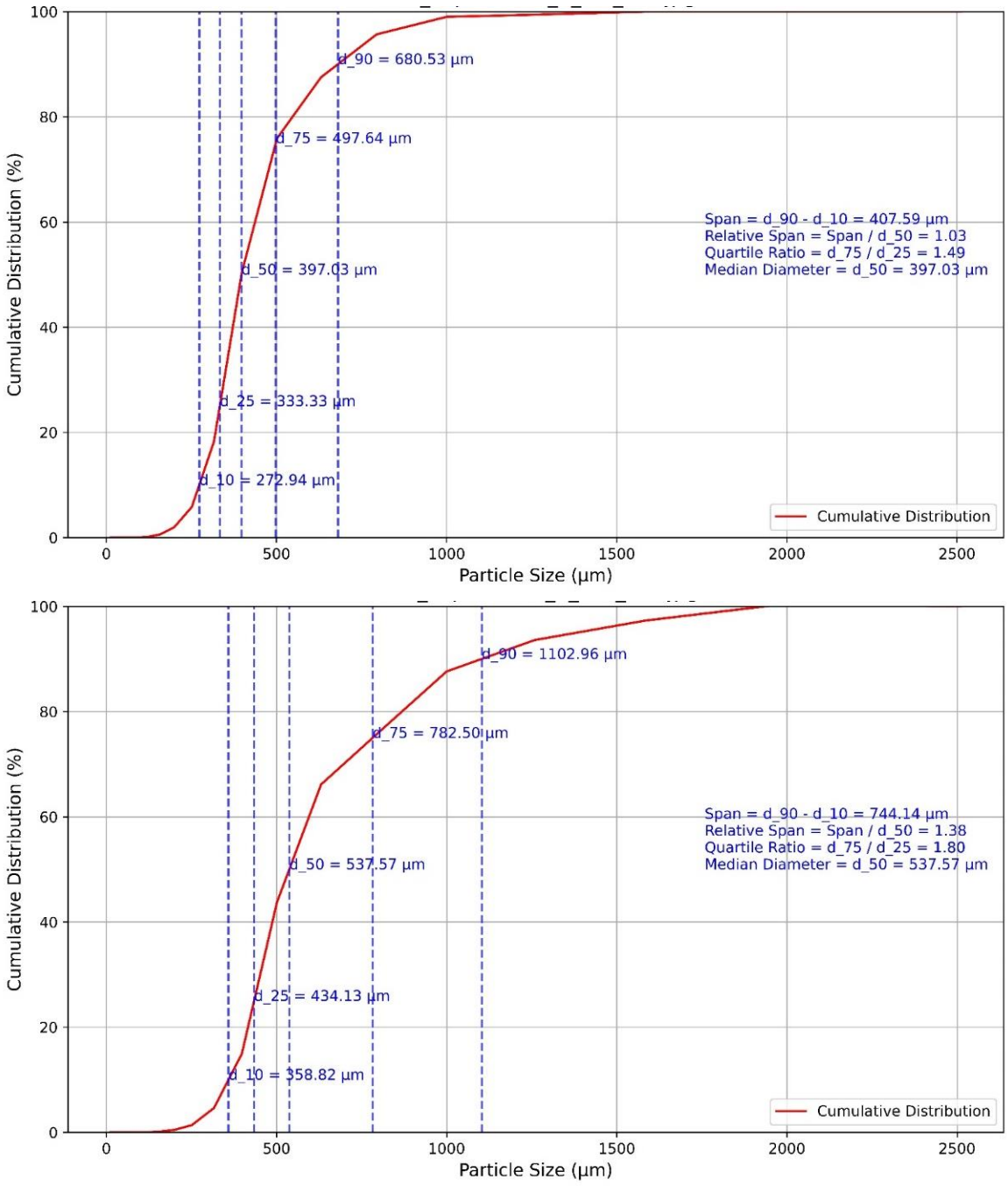

Figure 28: Before (Above) and after (Below) real data cumulative particle size distribution of experiment 7.

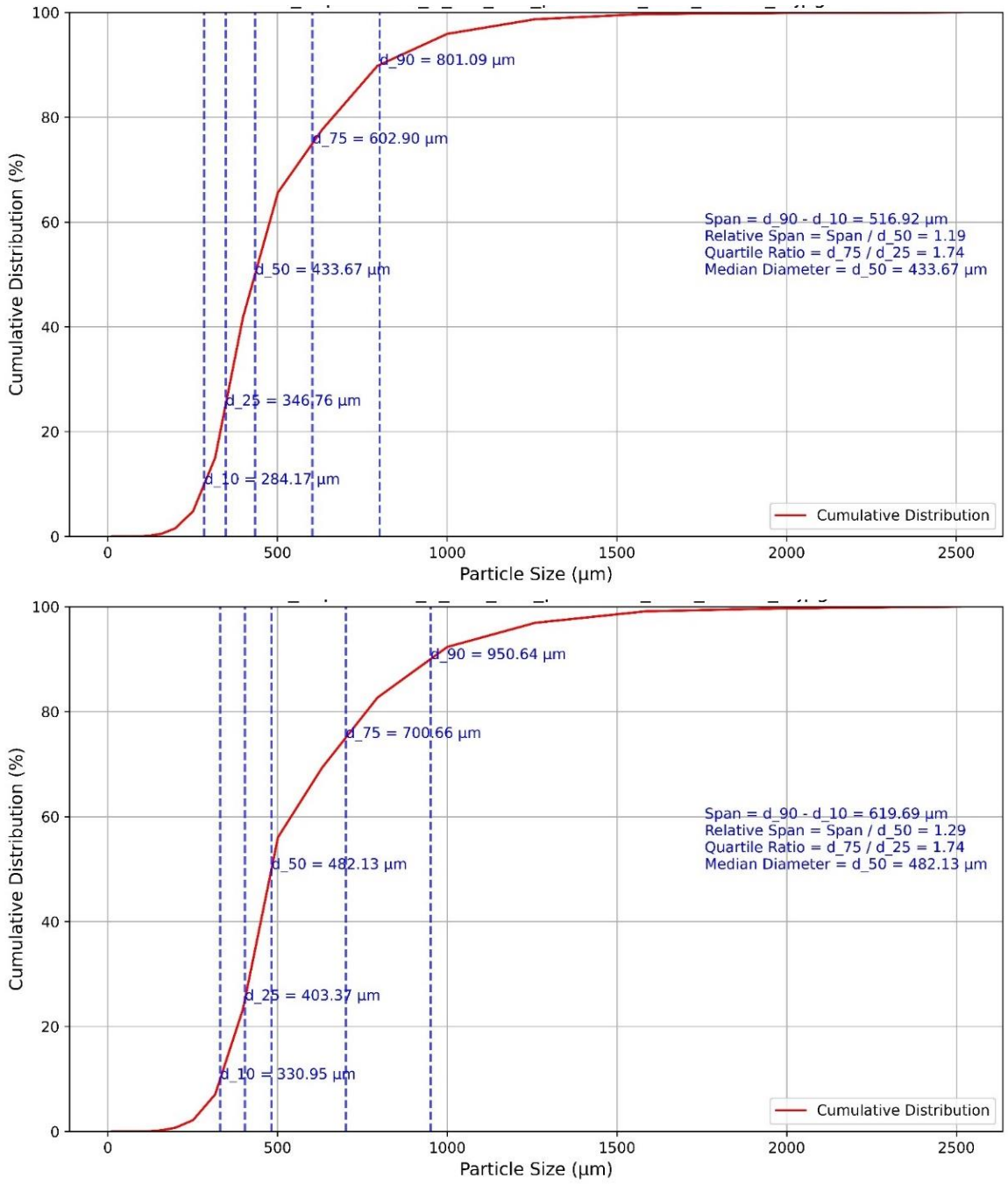

Figure 29: Before (Above) and after (Below) Predicted data cumulative particle size distribution of experiment 7 using Model 6.

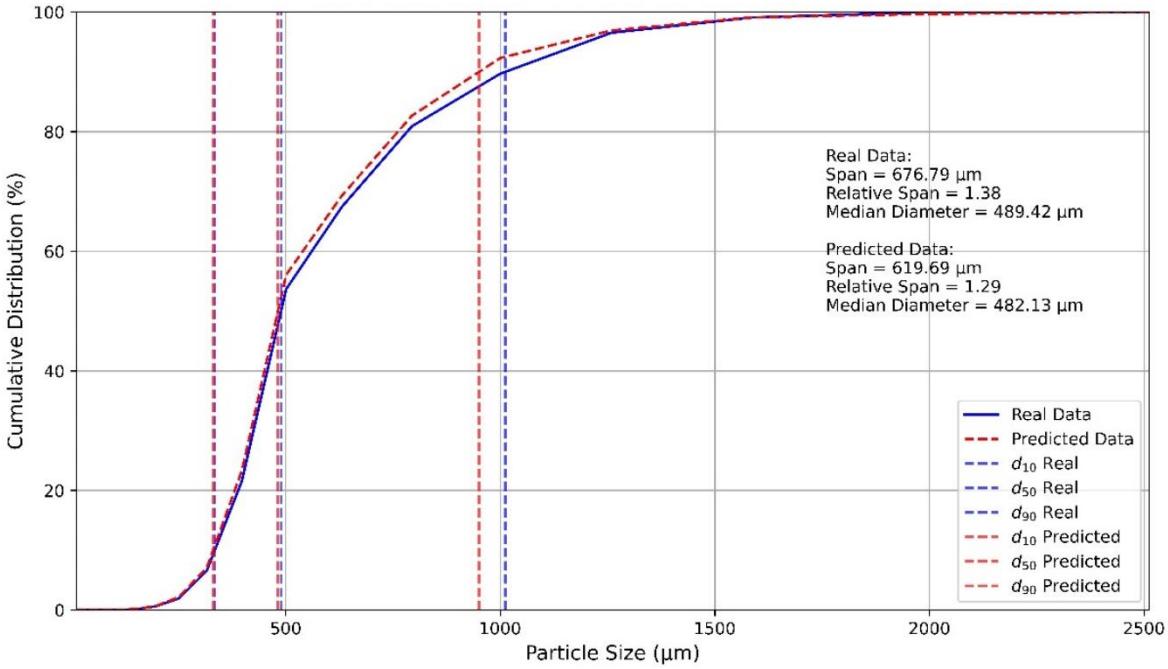

Figure 30: Combined plot of real data and predicted data of experiment 6 with model 6.

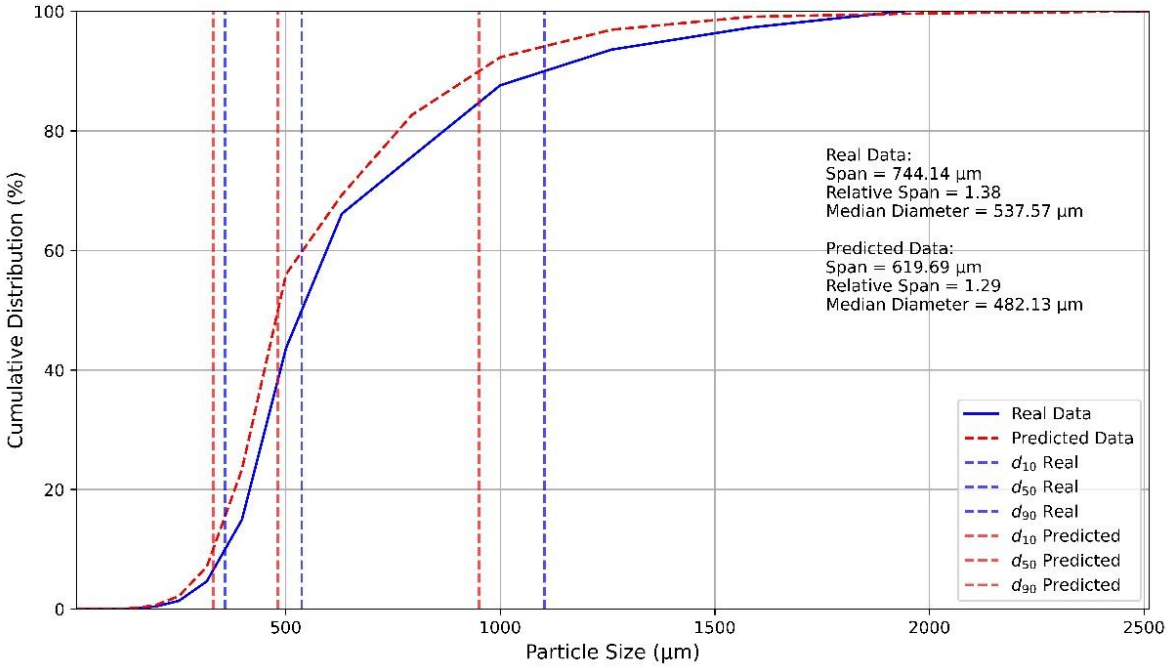

Figure 31: Combined plot real data and predicted data of experiment 7 with model 6.

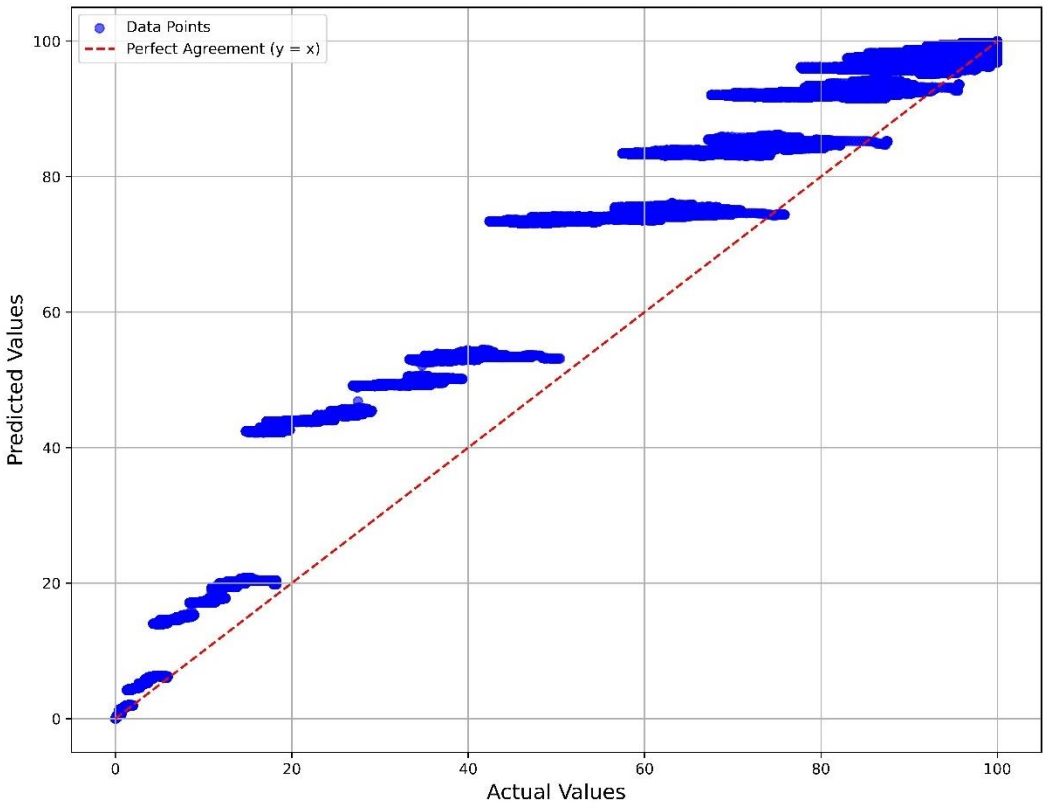

Figure 32: Parity Plot of Experiment 7 with its prediction using Model 6.

4.2. Combined CSV data trained model

Since the R^2 scores of the combined models are all negative, the results of the inference are not satisfiable and can be seen in [Figure 33]. Neither combining nor uncombining the Experiment-5 data (the clogged experiment) change its results. Since the combination of 1 vs 8 method all resulted in a negative R^2 score when introduced to an unseen dataset, testing the model with one of the datasets that was used in training would rather be useless, because it would naturally get a positive R^2 score. Therefore, this method can be concluded to not be suitable approach for training this model. Perhaps for future experiment, combining only a few trainings dataset of possibly 4 or 5 and then test against the rest of the dataset could result in new findings.

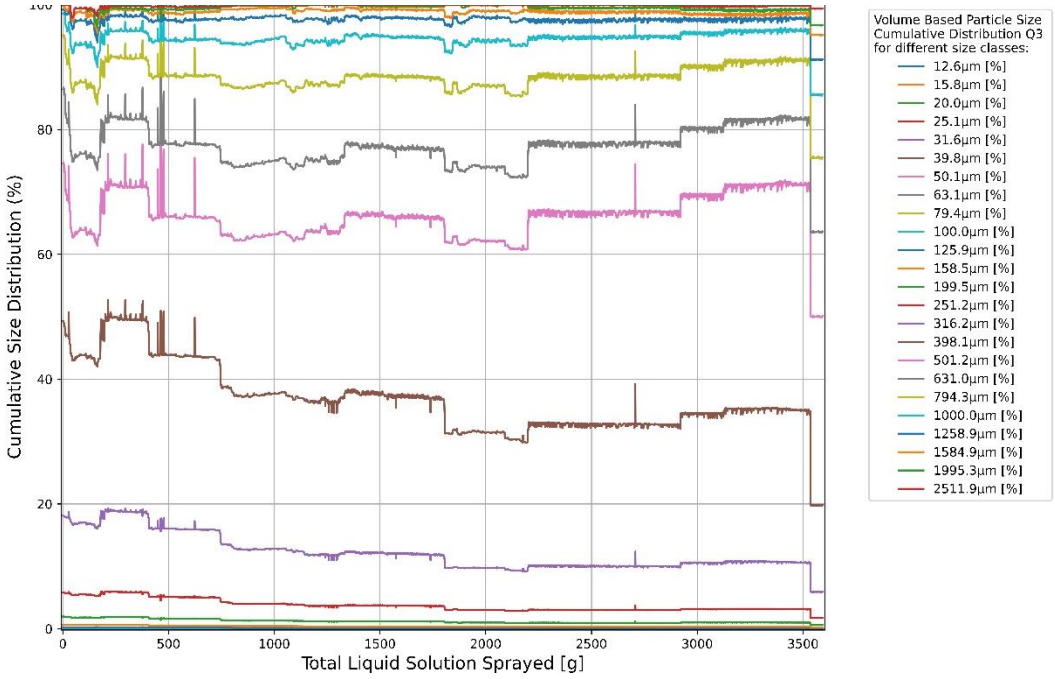

Figure 33: Predicted data of Experiment-5 using combined model-5-combined.

4.3. Combined CSV data + 20% trained model

In this method, the model resulted in a much better R^2 score, which can be seen in [Table 5] and that is expected, because the part of the test dataset is used in the training as well. But this entails another problem, now the model is “stuck” within the limits of the 9 experiment parameters and can only predict those datasets, which beats the purpose of a machine learning model. Therefore, this method is not recommended to achieve generalizability of a model.

4.4. Weak Spots of Random Forest Regressor and solution

If the feature importance varies from dataset to dataset, the algorithm may struggle to converge on a consistent set of “important” features, which leads to instability in the model’s performance. Features that are crucial in one dataset may not be selected as frequently in the bootstrap samples for other datasets, which reduces the influence on predictions. This issue is rather common where the dataset’s features and labels changes among datasets, and the features have multicollinearity, where multiple features are strongly correlated, which causes the model during training to split importance among them arbitrarily.

Possible solutions to improve the model’s predictability of unsee data; creating composite indices to identify relevant aspects of multiple features, this would help transform features to reduce variability across datasets. Another solution is applying dimensionality reduction techniques, which could reduce the feature set to a smaller number of consistent variables. This minimizes the variability in feature importance by focusing on much consistent subsets of information.

Bibliography

[1] A. Hafsa, S. Azeez, and D. Shahidulla, “GRANULATION TECHNIQUES: AN OVERVIEW,” *WORLD JOURNAL OF PHARMACY AND PHARMACEUTICAL SCIENCES*, vol. 8, pp. 525–546, Mar. 2022, doi: 10.20959/wjpps20195-13774.

[2] “Granulation Techniques in Pharmaceutical Manufacturing | Fabtech.” Accessed: Feb. 10, 2025. [Online]. Available: https://fabtechnologies.com/granulation-techniques/

[3] C. G. Philippsen, A. C. F. Vilela, and L. D. Zen, “Fluidized bed modeling applied to the analysis of processes: review and state of the art,” *Journal of Materials Research and Technology*, vol. 4, no. 2, pp. 208–216, Apr. 2015, doi: 10.1016/j.jmrt.2014.10.018.

[4] M. Kraume, *Transportvorgänge in der Verfahrenstechnik: Grundlagen und apparative Umsetzungen*. Berlin, Heidelberg: Springer, 2020. doi: 10.1007/978-3-662-60012-2.

[5] M. Orth, P. Kieckhefen, S. Pietsch, and S. Heinrich, “Correlating Granule Surface Structure Morphology and Process Conditions in Fluidized Bed Layering Spray Granulation,” *KONA*, vol. 39, no. 0, pp. 230–239, Jan. 2022, doi: 10.14356/kona.2022016.

[6] P. Kieckhefen, S. Pietsch-Braune, and S. Heinrich, “Product-Property Guided Scale-Up of a Fluidized Bed Spray Granulation Process Using the CFD-DEM Method,” *Processes*, vol. 10, no. 7, p. 1291, Jun. 2022, doi: 10.3390/pr10071291.

[7] M. Langner, I. Kitzmann, A.-L. Ruppert, I. Wittich, and B. Wolf, “In-line particle size measurement and process influences on rotary fluidized bed agglomeration,” *Powder Technology*, vol. 364, pp. 673–679, Mar. 2020, doi: 10.1016/j.powtec.2020.02.034.

[8] P. Labsystem, “Glatt. Integrated Process Solutions.”.

[9] X. Zhou, “Development of a Spray Nozzle Model using the Flowsheet Simulation Framework Dyssol,” Hamburg University of Technology, Hamburg, 2020.

[10] M. Harders, “Inline-Vermessung der Partikelgröße in einem Sprühgranulationsprozess,” Hamburg University of Technology, Hamburg, 2024.

[11] P. Dieter, D. Stefan, E. Günter, and K. Michael, “In-line particle sizing for real-time process control by fibre-optical spatial filtering technique (SFT),” *Advanced Powder Technology*, vol. 22, no. 2, pp. 203–208, Mar. 2011, doi: 10.1016/j.apt.2010.11.002.

[12] Microtrac Retsch GmbH, “Particle Size and Shape Analyzer CAMSIZER X2,” Germany, 2024.

[13] J. Z. Kosicki, “Generalised Additive Models and Random Forest Approach as effective methods for predictive species density and functional species richness,” *Environ Ecol Stat*, vol. 27, no. 2, pp. 273–292, Jun. 2020, doi: 10.1007/s10651-020-00445-5.

[14] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[15] Govardhan211103, “Correlation among features and between feature | output-label, Intuition and Implementation,” Medium. Accessed: Feb. 10, 2025. [Online]. Available: https://govardhan211103.medium.com/correlation-among-features-and-between-feature-output-label-intuition-and-implementation-1fe66a1332a9

[16] M. A. Hall, “Correlation-based Feature Selection for Machine Learning”.

[17] S.-W. Hwang *et al.*, “Feature importance measures from random forest regressor using near-infrared spectra for predicting carbonization characteristics of kraft lignin-derived hydrochar,” *J Wood Sci*, vol. 69, no. 1, p. 1, Jan. 2023, doi: 10.1186/s10086-022-02073-y.

[18] L. Li, Y. Huang, X. Cui, X. Cheng, and X. Liu, “On Testing and Evaluation of Artificial Intelligence Models,” in *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, Aug. 2023, pp. 92–97. doi: 10.1109/ICSECE58870.2023.10263364.

[19] M. Chung, "R^2," in *An Introduction to R*, vol. 0, 2 vols., in Encyclopedia of Research Design, vol. 0. , Taiwan, pp. 1187–1191.
[20] "R-squared in Regression Analysis in Machine Learning," GeeksforGeeks. Accessed: Feb. 10, 2025. [Online]. Available: https://www.geeksforgeeks.org/ml-r-squared-in-regression-analysis/

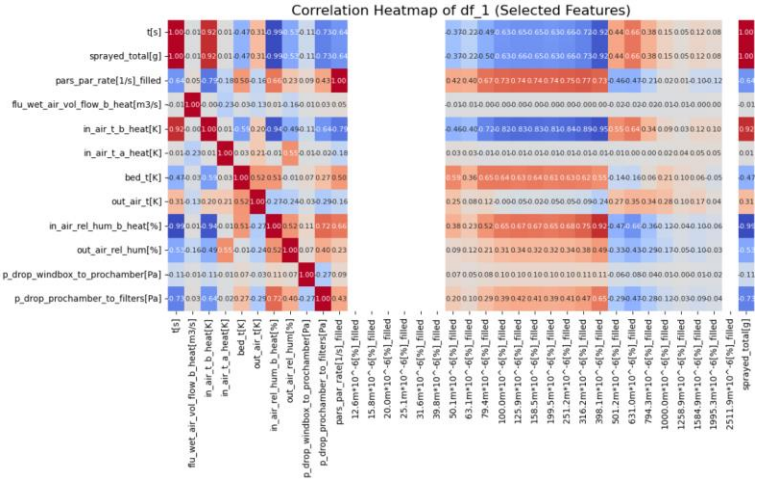

Figure A - 1: Correlation matrix Dataset 1.

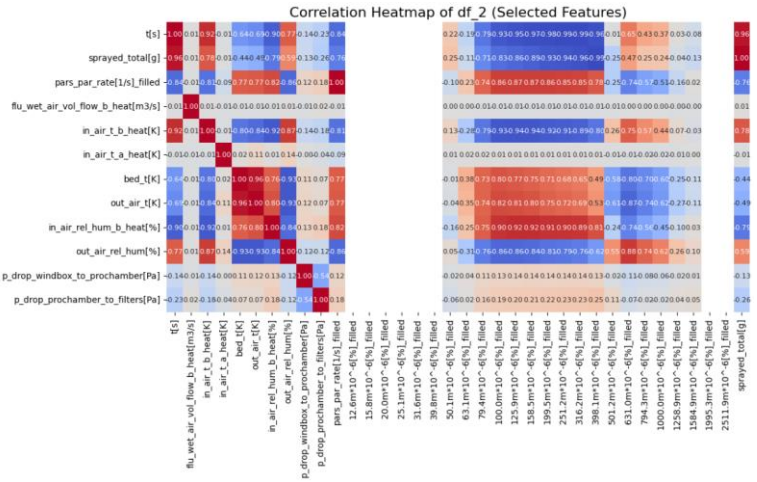

Figure A - 2: Correlation matrix Dataset 2.

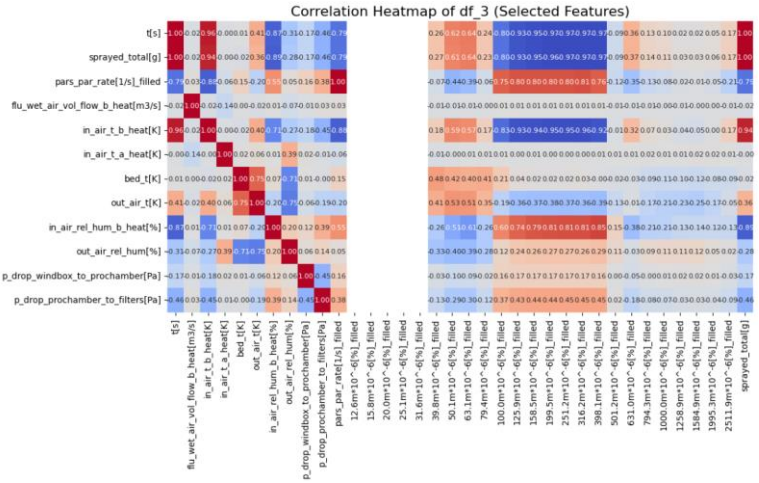

Figure A - 3: Correlation matrix Dataset 3

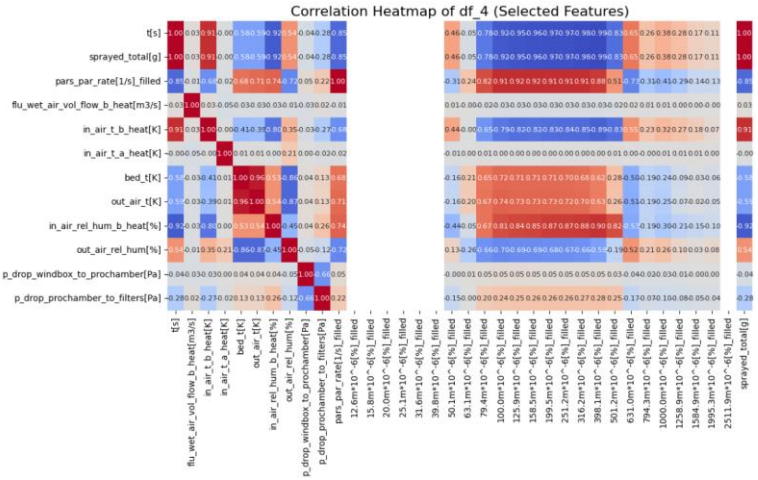

Figure A - 4: Correlation matrix Dataset 4

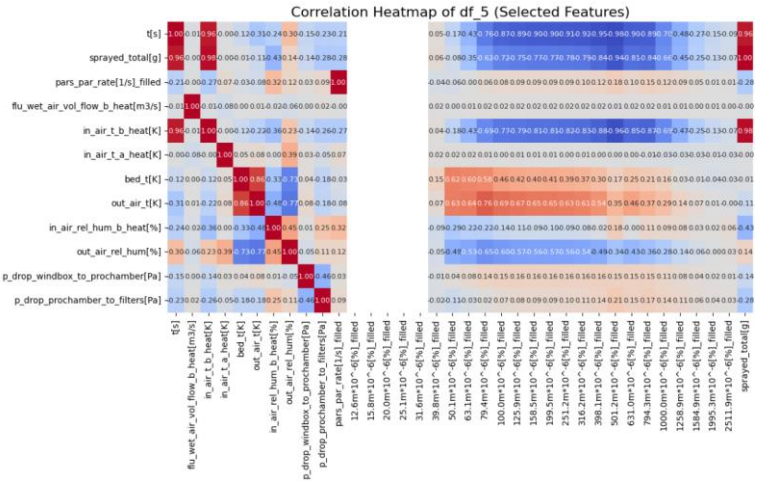

Figure A - 5: Correlation matrix Dataset 5

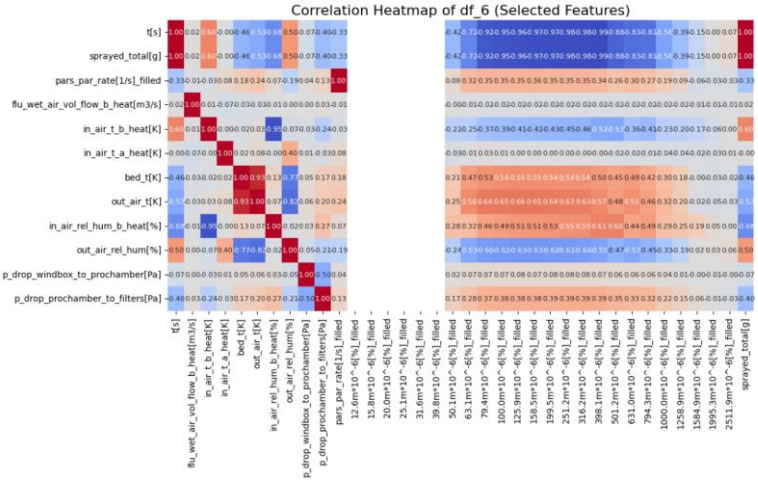

Figure A - 6: Correlation matrix Dataset 6

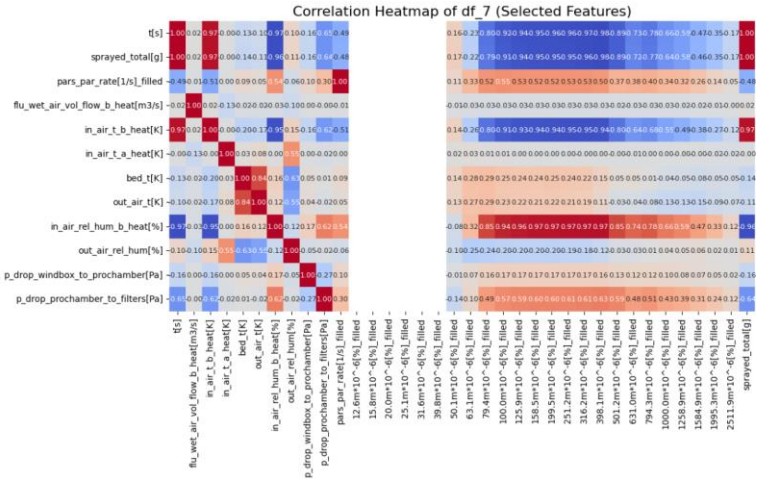

Figure A - 7: Correlation matrix Dataset 7

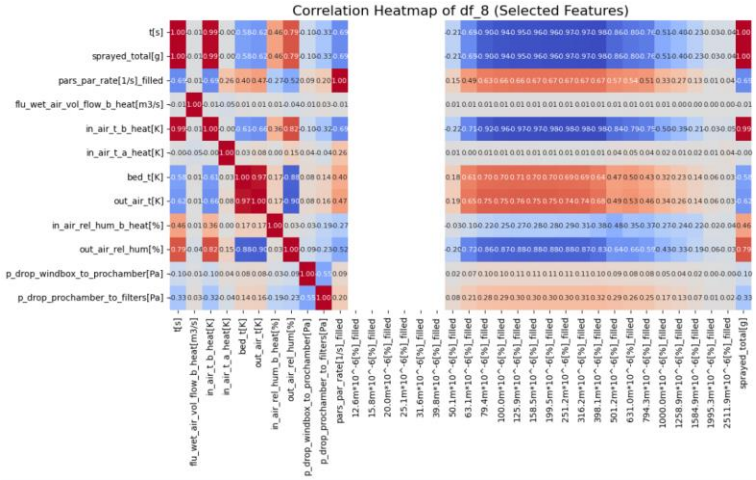

Figure A - 8: Correlation matrix Dataset 8

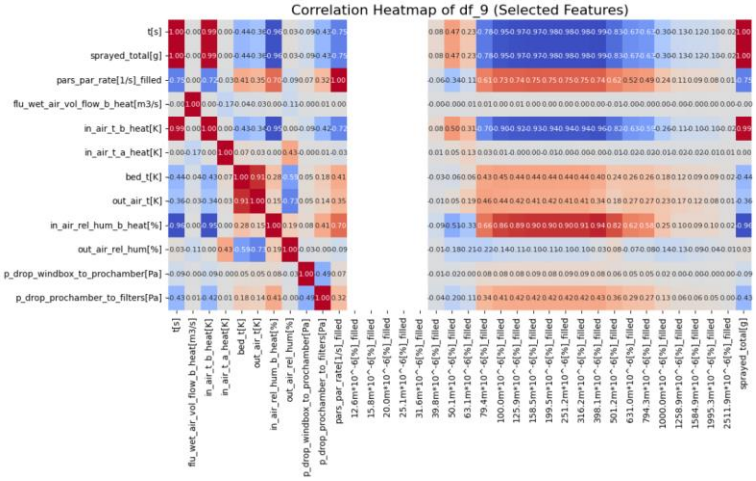

Figure A - 9: Correlation matrix Dataset 9