# Hybrid Machine Learning Characterization and Parameter Space Analysis using Interactive Visualization for Analyzing the Quality of the Virtual Nonwovens.

---

Viny Saajan Victor

*February, 2021*

Technical University Kaiserslautern

visual
information
analysis

Department of Computer Science

This thesis is submitted in fulfillment for the degree of Master of Science

# Hybrid Machine Learning Characterization and Parameter Space Analysis using Interactive Visualization for Analyzing the Quality of the Virtual Nonwovens.

Viny Saajan Victor

*1. Reviewer* Prof. Dr. Heike Leitte
Department of Computer Science
Technical University Kaiserslautern

*2. Reviewer* Dr. Andre Schmeißer
Transport Processes Department
Fraunhofer Institute for Industrial Mathematics

*Supervisors* Prof. Dr. Heike Leitte and Dr. Andre Schmeißer

February, 2021

**Viny Saajan Victor**

*Hybrid Machine Learning Characterization and Parameter Space Analysis using Interactive Visualization for Analyzing the Quality of the Virtual Nonwovens.*

This thesis is submitted in fulfillment for the degree of Master of Science, February, 2021
Reviewers: Prof. Dr. Heike Leitte and Dr. Andre Schmeißer
Supervisors: Prof. Dr. Heike Leitte and Dr. Andre Schmeißer

**Technical University Kaiserslautern**
Department of Computer Science
Erwin-Schrödinger-Straße 52
67663 and Kaiserslautern

# Declaration

I, Viny Saajan Victor, declare that this thesis titled "Hybrid Machine Learning Characterization and Parameter Space Analysis using Interactive Visualization for Analyzing the Quality of the Virtual Nonwovens" , and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

# Abstract

The quality of the technical textiles, that are nonwoven webs of fibers, depends on the process parameters involved in the production of these textiles. It is not practical to analyze the effect of these parameters in real-time. Hence, many simulation tools have been developed to depict this production process. The results of these simulations are used in the virtual production of the nonwovens. The quality of the nonwovens is then analyzed and mapped back to the process parameters. This approach consumes a lot of time and memory.

In this thesis, we incorporate Machine Learning to map the process parameters to the quality of the nonwovens to reduce the time and memory involved while analyzing the effect of these parameters on the quality of the product. We constructed a dataset using the virtual production tool to train the machine learning models.

We developed a visual analytical tool to assist the user in analyzing the effects of the process parameters on the product quality. The tool uses the machine learning model at the back end to predict the quality of the nonwovens based on the selected parameter setting. The main aim of the tool is to visually guide the user in finding the optimal combination of the parameters to obtain the desired product quality. This problem setting is termed as "Parameter Space Exploration" in visual analysis. The tool also provides multiple analysis to the user which aids him in decision-making.

# Acknowledgement

I would like to sincerely thank everyone who helped me and supported me during my thesis. I am extremely thankful to Prof. Dr. Heike Leitte for providing me an opportunity to write my thesis in Visual Information Analysis Department and supporting me throughout the thesis. I would like to express my heartfelt gratitude to my mentor Dr. Andre Schmeißer for his relentless support and encouragement throughout the learning process of this master thesis. I could not imagine having a better supervisor than him throughout my entire time in Fraunhofer ITWM.

I owe my deepest gratitude to God Almighty and my family for supporting me throughout my life. I am also highly indebted to all my friends for their immense love, help and motivation that helped me in every stage.

# Contents

# 1 Introduction

> *The greatest value of a picture is when it forces us to notice is what we never expected to see.*
>
> **— John Tukey**
> (American Mathematician)

## 1.1 Production Process of Nonwoven Fabrics

Technical textiles are nonwoven webs of fibers that find their applications in many branches such as textile, hygiene, automobile and building industries. One of the real-time applications currently is for the preparation of the face masks. The property of the fabric depends on their usage. An important common property for the quality assessment of the fabrics is the homogeneity of the fiber web.

One of the main objectives in the production of these nonwoven webs is the optimal design of the production process keeping the desired material specification. Mathematical models have been developed for the simulation and control of productions process [KMW09]. Based on these mathematical models, several simulation tools are developed to imitate the dynamics of production process. One such tool is the Fiber Dynamics Simulation Tool (FIDYST).

### 1.1.1 Fiber Dynamics Simulation Tool(FIDYST) :

FIDYST is a simulation tool developed by the Fraunhofer ITWM that simulates fibers in turbulent flows. FIDYST simulates the dynamics of elastic, line shaped objects in a very general way. Hence, there is a broad spectrum of different applications for FIDYST. Of particular importance are production processes of technical textiles. The simulations of the fiber dynamics are used to optimize the geometry of the production plant and the operating conditions. Goal of the optimization is an improved quality of the final product and reduced energy and raw material consumption at the same time. The Figure 1.1 shows the Graphical user interface of the tool.

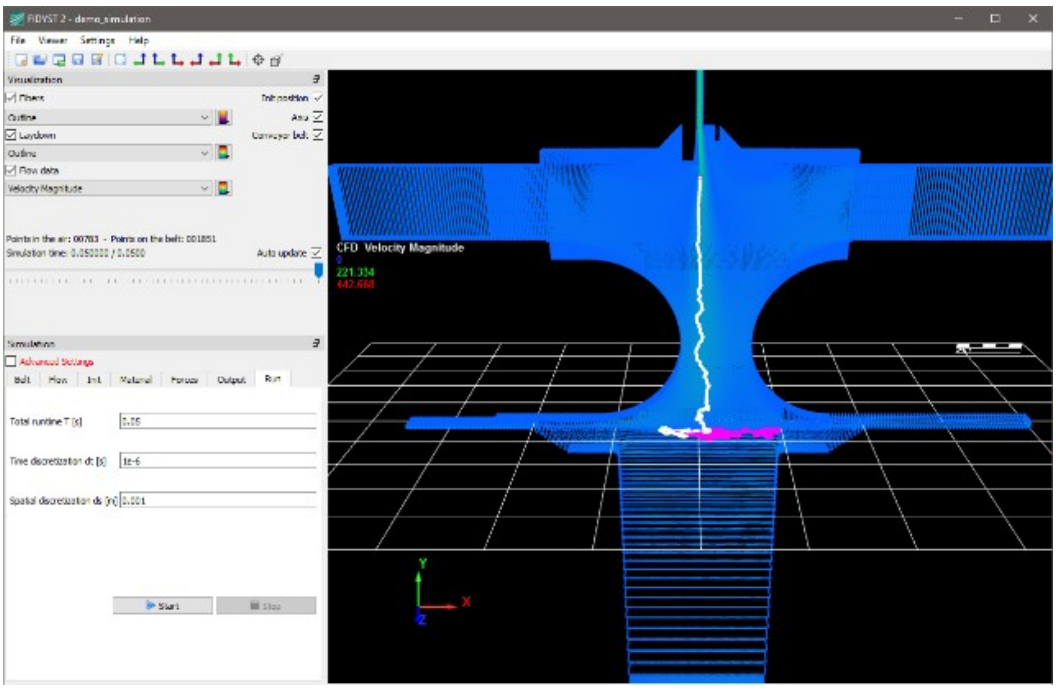



**Fig. 1.1:** Graphical User Interface of the FIDYST, showing a spun-bond airflow and filament simulation. Image Courtesy: Fraunhofer ITWM

## 1.2 Virtual Production of Nonwovens

The process of production of nonwovens consists of melting, spinning, swirling, and deposition. Nonwoven web is created by spinning and entangling a large number of filaments or fibers. Production of several nonwovens for the analysis in real time is not practical. Hence many methods and tools have been developed for virtualization of these processes to produce nonwovens [WMH15]. One such tool used to generate the virtual nonwoven web is the Software SURRO (Surrogate model).

### 1.2.1 Software SURRO(Surrogate model):

SURRO is the software developed by Fraunhofer ITWM which generates large-scale virtual nonwoven structures as shown in the Figure 1.2. It is based on a stochastic surrogate model for the simulation of filaments, which is mathematically defined by [KMW09] as the following system of stochastic differential equations

$$d\xi_s = \tau(\alpha_s)ds - d\gamma_s \tag{1.1}$$

$$d\alpha_s = -\nabla B(\xi_s) * \tau^{\perp}(\alpha_s)ds + AdW_s \tag{1.2}$$

$$B(\xi) = (\xi_1^2/\sigma_1^2 + \xi_2^2/\sigma_2^2)/2 \tag{1.3}$$

Where $\sigma_1$, $\sigma_2$ and $A$ are input parameters of the SURRO tool and are obtained by first performing physically-based simulations of a few individual filaments using FIDYST. $\sigma_1$ and $\sigma_2$ are standard deviations of a 2D normal distribution $N(\mu, \sum)$, centered around the spin position $\mu$. The linear belt movement is added by the function $\gamma_s$, A determines the strength of the stochastic influence.

$$\sum = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \tag{1.4}$$

## 1.3 Analysis of Nonwoven production processes using FIDYST and SURRO

First, FIDYST is used to compute a full simulation of one or several representative filaments including all relevant physical effects, e.g. turbulent airflow. Then, the FIDYST simulation is imported into SURRO and the lay-down is analysed. In an identification step, the imported data is fit to a set of stochastic parameters which can then be used to reproduce a large nonwoven of similar quality. The generated nonwoven structure can then be analysed with regard to weight distribution and homogeneity on different scales. The homogeneity is essential for the quality of the resulting nonwoven fabric and one of the criteria used to optimize the production process.

## 1.4 Machine Learning for the analysis of Nonwovens

Machine learning(ML) models have proven to be working well in various fields. The virtual production of large nonwovens samples using FIDYST and SURRO requires a lot of computation time and resources. To speed up the process, we can utilize the stochastic parameters of SURRO to train a machine learning model that predicts the final product quality. This approach is computationally efficient in analysing the

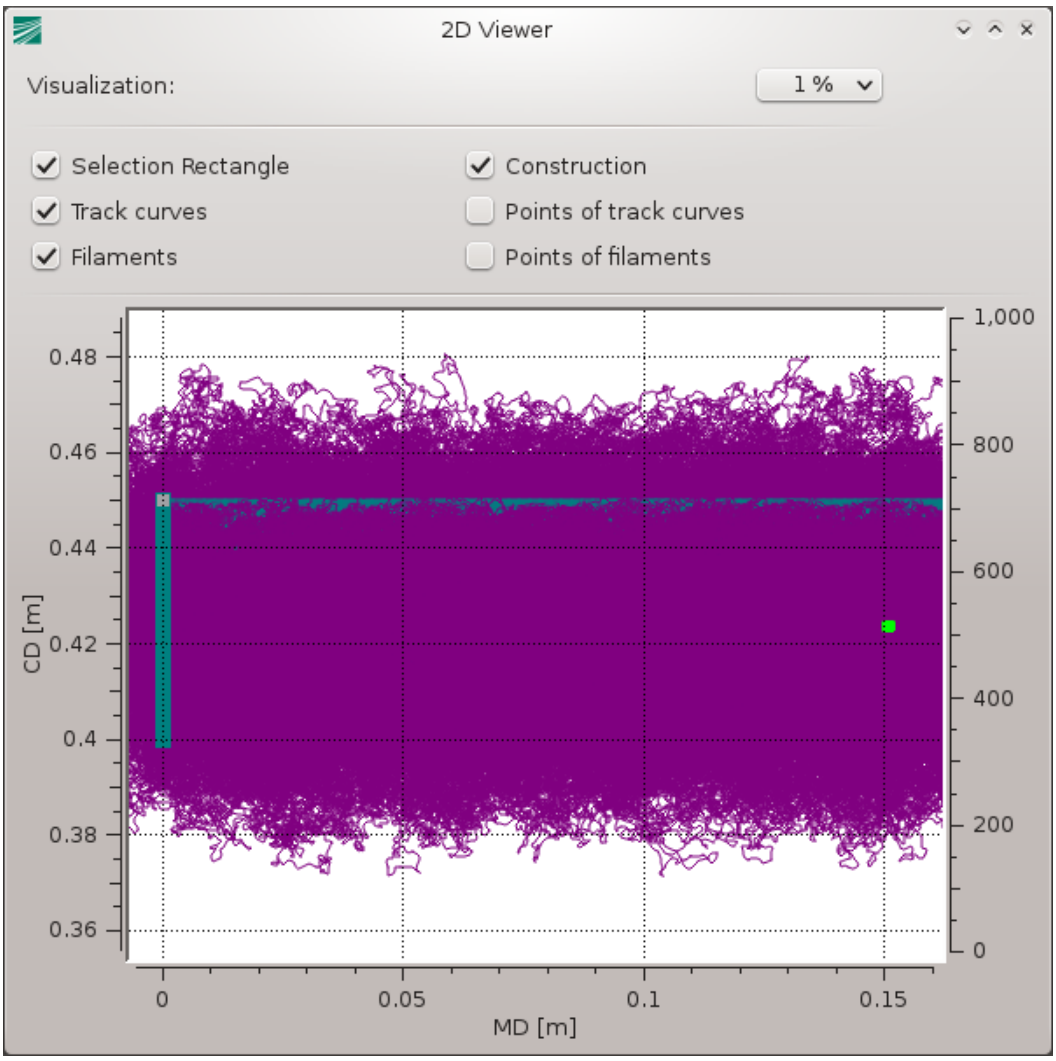



**Fig. 1.2:** 2D view of virtual nonwoven in SURRO. Courtesy: Fraunhofer ITWM

influence of the stochastic parameters on the product quality, which are then used to fine tune the process parameters to achieve optimization.

## 1.5 Motivation and Problem Statement

The thesis mainly focuses on achieving two goals:

First, train a machine learning model to replace the SURRO tool to map the input parameters to the quality of the nonwovens. The main intention here is to reduce computational time and resources.

Second, developing an interactive tool that allows the user to visually analyse the model parameters that are responsible for the quality of the nonwovens. This interactive tool uses the ML model for faster prediction of the interpolated data. It also aids the user to search the model parameter space thereby helping to find the optimal combination corresponding to the desired quality of nonwovens.

The Figure 1.3 explains the flowchart of the overall thesis structure with different components.

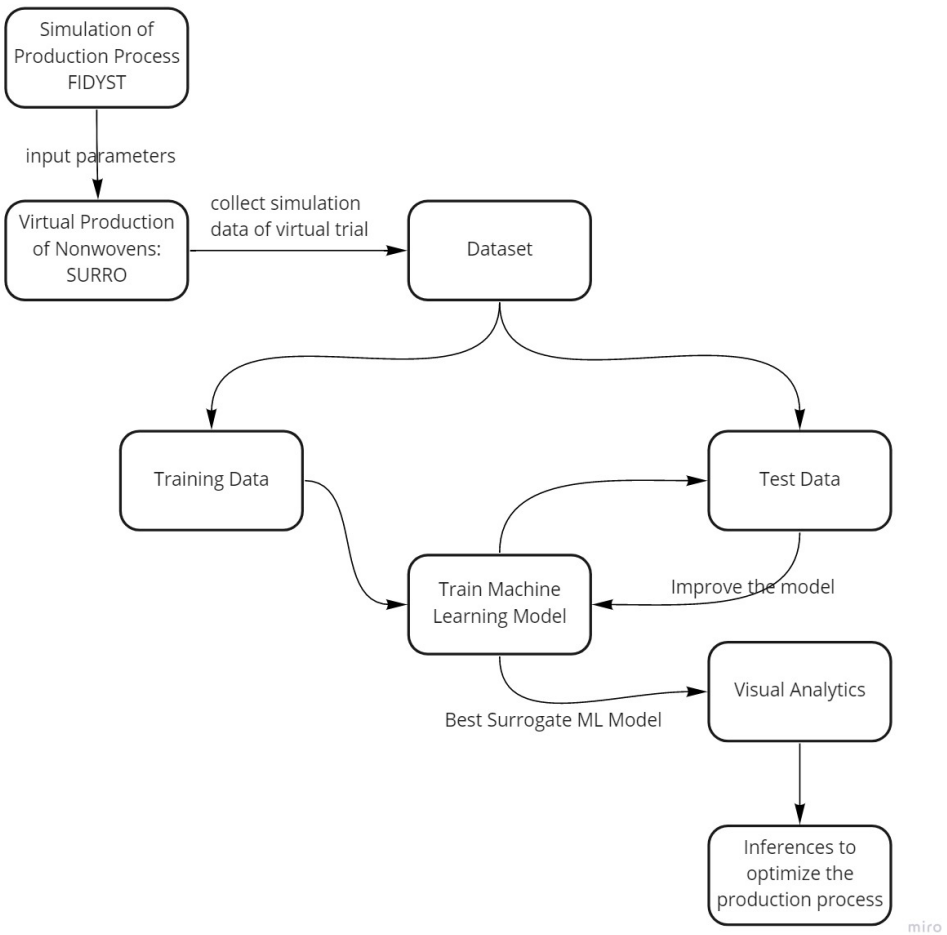



**Fig. 1.3:** Overview of thesis, showing different components

## 1.6 Thesis Structure

The rest of the thesis is organized as follows. Chapter 2 explains the related work for both machine learning and visual parameter space exploration topics. Chapter

3 talks about the simulation setup of the SURRO tool for collection of the data. It also explains the strategies used for preparing the input and output datasets using SURRO simulation. In Chapter 4, we evaluate different regression models on our dataset and selects the model with better accuracy. Chapter 5 gives the overview of the visual analytic tool and explains its components and their use cases for our analysis. In this chapter we also explain how the tool can visually guide the user in parameter space exploration. Chapter 6 states the inferences realized using our tool and provides arguments to support our claims and concludes the thesis with the scope for future work.

# 2 Literature Survey

## 2.1 Machine Learning

Machine learning [JM15] is a data analysis tool that automates analytical model building. It is a branch of artificial intelligence that can learn from data, identify patterns and make decisions with minimal human intervention in real-time. Therefore, in our approach a machine learning model is used as a surrogate model for the fiber simulation tool for faster computation of the results and also to compute the results for interpolated data which helps in our analysis.

Machine learning comprises various learning algorithms. One of the popular methods being Regression Analysis [CH15], which is a statistical method that aims to determine the strength and character of the relationship between one or more dependent variables and independent variables.

### 2.1.1 Problem setting

Our input data contains five continuous input features which are used to predict seven continuous output values. Hence we need to use multiple-input, multiple-output regression as our machine learning method. We also need to examine the type of relationship between the input features and the output values(For example, linear or non-linear relationship).

### 2.1.2 Basis for Literature Survey

The literature survey was done by categorizing the relevant papers based on the following three criteria:

1. **Type of Input and Output Data:** [SI] Paper compared many regression models for the prediction of stock market movement using multiple independent variables. This problem is similar to our data setting in terms of the data type. The paper evaluates linear regression, polynomial regression and support

vector regression and concludes that support vector regression performs best for their dataset.

2. **Application Domain:** [Abo15] is using artificial neural network to achieve predictability of some of the woven fabric properties. The machine learning model used for this purpose is a multiple-input, multiple-output regression. In this paper neural network performed exceedingly better than multi-linear regression. [SSC19] presented the simulation of production process and how to make use of regression for forecasting in glove textile industry. The dataset is applied to many machine learning models to evaluate the performance. The paper compared Decision Trees, K-Nearest Neighbor, AdaBoost, Random Forests and Support Vector Regression. Random Forest Regression performed best for the dataset in terms of accuracy.

3. **Confidence and Interpretability of the model:** [BT03] In comparison to classical regression, the Bayesian approach is used to formulate regression where, the dependent variables are assumed to be drawn from a probability distribution than being estimated as a single value. Since these models give distribution for the regression coefficients rather than point estimates, they can be used to analyse the confidence on its predictions.

## 2.2 Visual Parameter Space Analysis

Parameter space analysis (PSA) is defined according to [Sed+14] as the systematic variation of model input parameters, generating outputs for each combination of parameters, and investigating the relation between parameter settings and corresponding outputs. If this process is facilitated by interactive visualization then it is termed as "Visual Parameter Space Analysis(VPSA)".

[Sed+14] paper provides a conceptual framework for VPSA, independent of their application domain. Before choosing the framework, the authors analysed 112 research papers in the area of VPSA and selected 21 papers. These 21 papers were split into 14 for training and 7 for validation to provide generic framework independent of the application domain.

The visualization approaches inspired from the above framework that are used in the thesis can be categorized into the following two groups:

### 2.2.1 Navigation Strategies

Once the data has been generated and the machine learning model has been selected, this data needs to be presented to the user for exploration analysis. We classified three distinctive strategies of how this data was made available for navigation.

1. **Informed Trial and Error:** If the computation of the output values takes a short amount of time, user can be provided with an option to try out different combinations of parameter settings for the analysis. We use the machine learning model for the predictions to obtain the results in real-time.
2. **Local to Global Navigation:** The local-to-global strategy starts with displaying one specific output and lets the user explore alternatives from there. Output for a very specific input parameter setting is shown and user can interactively change the input parameters from that setting thereby updating the output.
3. **Global to Local Navigation:** In this strategy, user is first presented the overview of the whole range of data, from where zoom and filter options can be used to show the details on demand.

### 2.2.2 Analysis Techniques

Once we have characterized how to present data to the user for navigation, the next important goal is to understand the tasks that users want to engage in when doing visual parameter space analysis. After understanding the requirement we utilize the following analysis techniques to guide the design and engineering processes.

1. **Data Partition:** In this strategy, the output data space is partitioned into clusters to find different types of model behaviours. They are related back to input parameter settings to analyse the data responsible for these behaviors.
2. **Sensitivity Analysis:** Sensitivity is termed as the uncertainty of the input parameter value. The variation of outputs are observed against the change in inputs. This analysis helps in distinguishing optimal parameter setting and the non-sensitive parameter setting which is explained in the later sections.
3. **Handling Statistical uncertainty:** Statistical uncertainty indicates the difference in the output for multiple non-deterministic runs of the model while keeping the parameter setting constant. If we have higher statistical uncertainty, it needs to be reduced for our model to be trustworthy.

# 3 Nonwoven Sample Simulation Setup and Dataset Creation

We simulate virtual nonwoven samples using the input features database. For each of the simulated sample, we calculate the CV values at seven different resolutions-[0.5,1,2,5,10,20,50]. The input features along with the calculated CV values is used as the training set for the machine learning models.

**Input Features:** These are the stochastic parameters provide to the SURRO tool, as the input. We need to know two directions in the spinning process of nonwovens which are used to explain some of the input features, machine direction and cross direction. The former is the direction in which the belt moves and the latter is the direction perpendicular to the machine direction. Figure 3.1 shows these two directions.

1. $Sigma\_1(\sigma_1)$: is the standard deviation of the 2D normal distribution of the fibres around the spinning nozzle in machine direction without the belt movement. The feature values ranges from 1mm to 50mm.
2. $Sigma\_2(\sigma_2)$: is the standard deviation of the 2D normal distribution of the fibres around the spinning nozzle in cross direction without the belt movement. The feature values ranges from 1mm to 50mm.
3. $A$: is the feature that contains all random effects of the production process, e.g. the influence of the turbulent flow during the fiber spinning and lay-down, fiber-fiber contacts. This feature describes how deterministic (the value 0) or stochastic (the value $\infty$) the simulated fiber lays down. The feature values ranges from 1 to 50.
4. $BeltSpinRatio(BSR)$: is the ratio of spinning speed and belt speed. The feature values ranges from 0.01 to 0.25.
5. $SpinPositionsPerMeter(SPM)$: is the number of spin positions per meter. The feature values ranges from 200 to 10000.

**Output:** The output corresponds to the coefficient of variation(CV) values at seven different resolutions obtained form SURRO for each row of input feature.

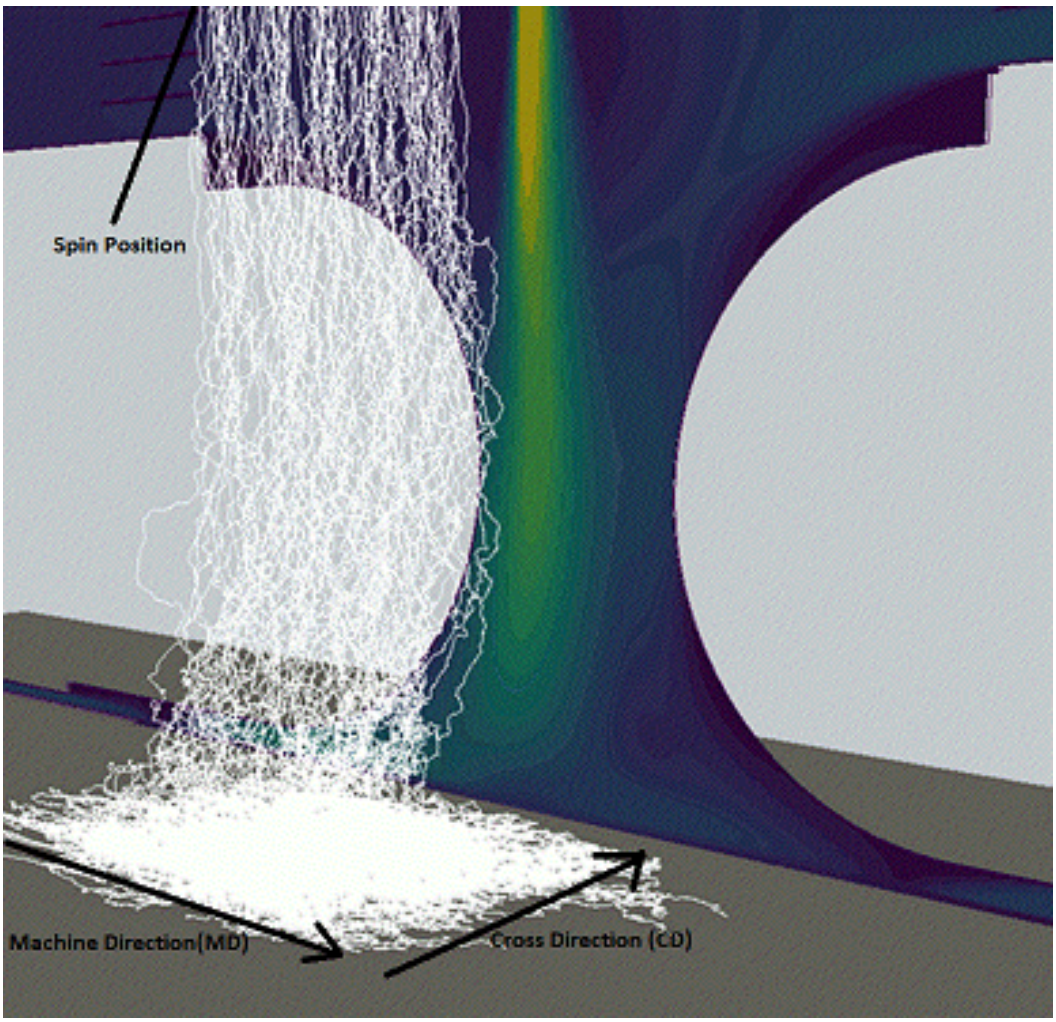



**Fig. 3.1:** Nonwoven spinning process showing the machine direction and cross direction. Image Courtesy: Fraunhofer ITWM

**coefficient of variation(CV):** The coefficient of variation (CV) is a statistical measure of the relative dispersion of data points in a data series around the mean. It determines the homogeneity of the nonwoven web. Lesser the CV value, lower the data dispersion and better the tensile strength of the nonwoven.

$$CV = \frac{\sigma}{\mu} \tag{3.1}$$

Where: $\sigma$ = Standard Deviation $\mu$ = Mean

## 3.1 Sample Region Size for the Simulation

While simulating the virtual nonwoven sample, we need to determine the size of the sample region which represents the whole nonwoven produced from the input features. Then we can construct only the fibers which are overlapping with this sample region. This greatly reduces the computation time and memory as we will not simulate the fibers that are outside the sample region. The SURRO tool is non-deterministic i.e. different simulation runs with the same parameter setting results in different outputs. Hence while selecting the size of the sample region, we ensure that the output variation across different runs is as small as possible for all the seven resolutions.

**Determination of statistical uncertainty:** First, we need to compare the CV values of the individual runs with same input parameter setting by computing the ratio of standard deviation and mean to see if they differ significantly. And we also need to verify whether this difference reduces if we go for larger sample region size.

We created a smaller database consisting of 3125 input rows using uniform sampling. This database was simulated with two sample region sizes.

1. $5cm * 5cm$
2. $15cm * 50cm$

We computed the corresponding std/mean values for individual resolution. The below figures shows the distribution of these values for resolution 20mm with respect to both the sample region sizes. From the Figure 3.2 we can clearly see that the spread between individual runs with same parameter setting is lower for larger sample size.

We further determined the input parameter setting with the highest std/mean for the smaller sample region size and compared it with the corresponding std/mean of the larger sample region size to measure the difference. The same analysis is done for the other way round. The results are shown in Table 3.1 and Table 3.2

From the Table 3.1, for resolution 20cm, a very high value of std/mean(0.78) can be seen. We determined the input parameter setting responsible for this value which was [8.89, 50, 50, 0.3, 1000]. We used this parameter setting for our further analysis because this setting quantifies the highest difference of spread among individual runs for the two sample region sizes (a factor of 10.8).

**Determination of the sample region size for the simulation:**

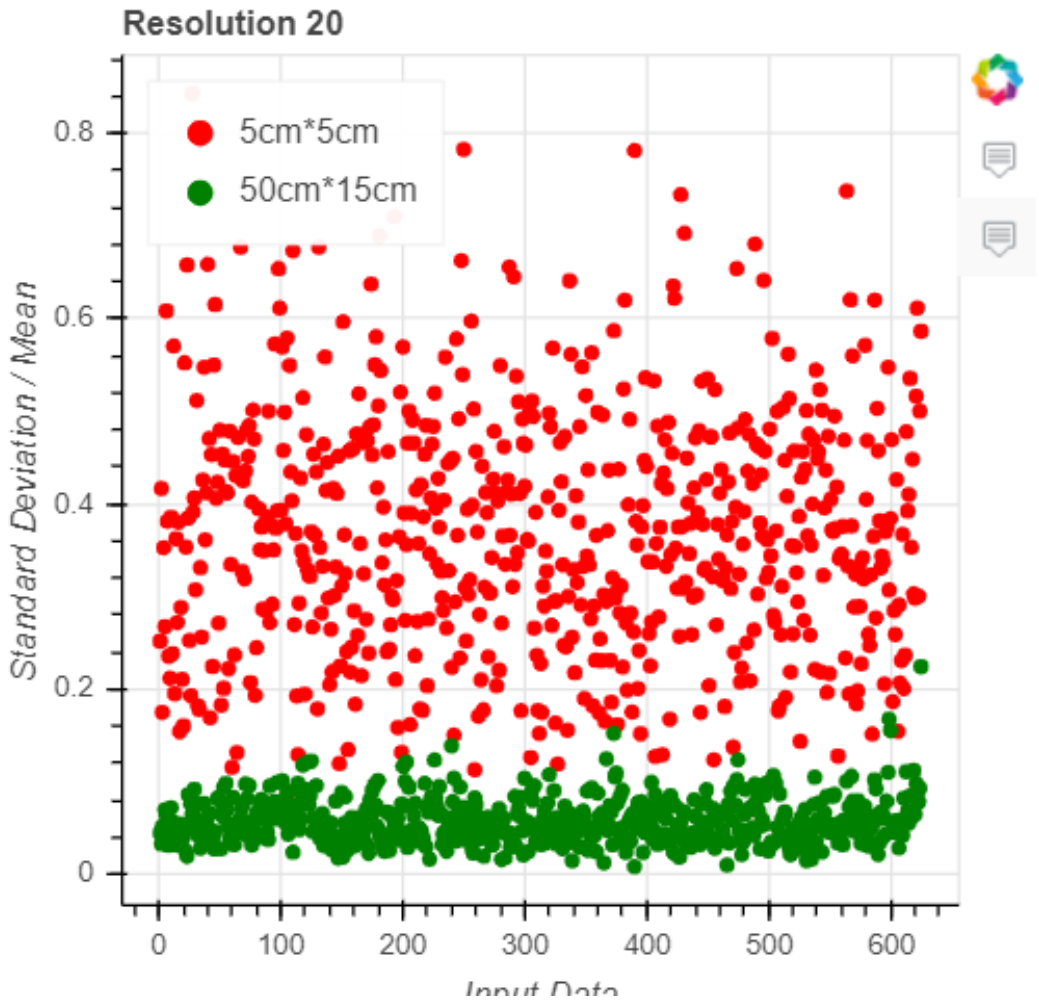



**Fig. 3.2:** Scatter-plot showing Std/Mean metric of individual samples for 2 different sample region sizes $Sigma\_1$ and $Sigma\_2$

We selected three sizes of the sample regions for our analysis.

1. $5cm * 5cm$
2. $15cm * 50cm$
3. $25cm * 50cm$

The input parameter setting [8.89, 50, 50, 0.3, 1000] which was determined in the previous analysis was simulated 100 times for all the three sample sizes. Table 3.3, Table 3.4 and Table 3.5 shows the statistics for size $5cm * 5cm$, $15cm * 50cm$ and $25cm * 50cm$ respectively.

| Resolution (in mm) | Mean (5*5) | Std (5*5) | Std/Mean (5*5) | Mean (15*50) | Std (15*50) | Std/Mean (15*50) | Ratio (5*5) / (15*50) |
|---|---|---|---|---|---|---|---|
| 0.5 | 44.53 | 4.96 | 0.11 | 44.30 | 1.03 | 0.02 | 4.80 |
| 1.0 | 38.61 | 5.31 | 0.13 | 38.31 | 1.03 | 0.02 | 5.08 |
| 2.0 | 34.10 | 5.58 | 0.16 | 33.65 | 1.06 | 0.03 | 4.65 |
| 5.0 | 12.51 | 2.99 | 0.23 | 14.67 | 0.45 | 0.03 | 7.69 |
| 10.0 | 24.49 | 8.60 | 0.35 | 26.16 | 1.07 | 0.04 | 8.55 |
| 20.0 | 11.88 | 9.30 | 0.78 | 17.67 | 1.28 | 0.07 | 10.80 |

**Tab. 3.1:** Table showing metrics of input sample with highest std/mean for sample region size $5cm * 5cm$ and the corresponding metrics for sample region size $15cm * 50cm$

| Resolution (in mm) | Mean (5*5) | Std (5*5) | Std/Mean (5*5) | Mean (15*50) | Std (15*50) | Std/Mean (15*50) | Ratio (5*5) / (15*50) |
|---|---|---|---|---|---|---|---|
| 0.5 | 63.49 | 1.99 | 0.03 | 62.29 | 2.86 | 0.04 | 1.46 |
| 1.0 | 53.26 | 2.00 | 0.03 | 51.92 | 2.91 | 0.05 | 1.49 |
| 2.0 | 45.16 | 2.05 | 0.04 | 43.67 | 3.13 | 0.07 | 1.58 |
| 5.0 | 34.63 | 2.07 | 0.05 | 32.17 | 3.55 | 0.11 | 1.85 |
| 10.0 | 30.44 | 2.86 | 0.09 | 28.56 | 5.29 | 0.18 | 1.96 |
| 20.0 | 20.31 | 4.55 | 0.22 | 16.29 | 9.55 | 0.58 | 2.62 |

**Tab. 3.2:** Table showing metrics of input sample with highest std/mean for sample region size $15cm * 50cm$ and the corresponding metrics for sample region size $5cm * 5cm$

As we can see from the Table 3.3, the 5cm*5cm sample size has a high spread relative to the other two sizes across different runs. We can also observe that the spread is decreasing with increase in the sample region size. Hence we decided to go for the largest size which is 25cm*50cm. Yet as per the observation in the Table 3.5, the sample size 25cm*50cm has a large variation($20\%$) for the lowest resolution[50]. Hence it not adequate to have a single sample of this size.

According to Central Limit Theorem [Ros56], averaging $N$ samples should reduce the spread by a factor of $\sqrt{N}$. Therefore in our scenario, sampling the output 5 times for the same input setting should reduce the variance approximately by $9\%$.

This approach of simulating multiple samples for the same input parameter setting eliminates the need to opt for higher sample region sizes, which is a costly operation in terms of memory and computation.

**Sample Size Chosen: 25 cm * 50 cm**

**Number of Runs Chosen: 5**

| Resolution (in mm) | Min Value | Max Value | Mean | Std | Spread ($\frac{Std}{Mean}$) |
|---|---|---|---|---|---|
| 0.5 | 77.98 | 99.75 | 88.85 | 4.08 | 0.04 |
| 1.0 | 59.96 | 83.32 | 70.56 | 4.16 | 0.05 |
| 2.0 | 46.77 | 65.94 | 54.86 | 1.19 | 0.07 |
| 5.0 | 25.90 | 48.64 | 37.24 | 4.83 | 0.12 |
| 10.0 | 12.84 | 38.43 | 25.04 | 5.39 | 0.21 |
| 20.0 | 2.16 | 35.01 | 12.52 | 6.50 | 0.51 |

**Tab. 3.3:** Table showing characteristics of distribution 100 samples generated for input setting [8.89, 50, 50, 0.3, 1000] with sample size = $5cm * 5cm$

| Resolution (in mm) | Min Value | Max Value | Mean | Std | Spread ($\frac{Std}{Mean}$) |
|---|---|---|---|---|---|
| 0.5 | 85.31 | 92.69 | 89.17 | 1.59 | 0.01 |
| 1.0 | 68.39 | 74.54 | 70.89 | 1.23 | 0.01 |
| 2.0 | 51.34 | 61.31 | 56.25 | 1.38 | 0.02 |
| 5.0 | 35.49 | 42.56 | 38.94 | 1.37 | 0.03 |
| 10.0 | 24.00 | 32.36 | 27.19 | 1.61 | 0.05 |
| 20.0 | 14.19 | 25.51 | 18.07 | 2.20 | 0.12 |
| 50.0 | 3.88 | 14.54 | 7.98 | 2.40 | 0.30 |

**Tab. 3.4:** Table showing characteristics of distribution with 100 samples generated for input setting [8.89, 50, 50, 0.3, 1000] with sample size = $15cm * 50cm$

## 3.2 Construction of Nonwoven samples based on Sample Size

As discussed in the previous section we only construct the fibers which are overlapping with the sample region to reduce the time and memory consumption. From the property of a normal distribution [Gra06], we know that the percentage of values that lies with 2, 2.5 and 3 standard deviation away from mean in a normal distribution is $95\%$, $99\%$ and $99.7\%$ respectively.

**Hence we simulate**

1. $2.5\sigma_2 - 3\sigma_2$ of nonwoven in cross direction around rectangle.

2. A bit more than $3\sigma_1$ in machine direction around sample rectangle.

Because only those fibers have the chances of overlapping with our sample. Figure 3.3 shows the construction region of the nonwoven web based on $\sigma_1$ and $\sigma_2$ values.

| Resolution (in mm) | Min Value | Max Value | Mean | Std | Spread ($\frac{Std}{Mean}$) |
|---|---|---|---|---|---|
| 0.5 | 87.17 | 91.87 | 89.27 | 0.91 | 0.01 |
| 1.0 | 68.82 | 74.62 | 70.79 | 0.91 | 0.01 |
| 2.0 | 53.84 | 58.13 | 56.00 | 0.81 | 0.01 |
| 5.0 | 36.56 | 42.80 | 39.09 | 1.13 | 0.02 |
| 10.0 | 24.86 | 31.03 | 27.78 | 1.11 | 0.04 |
| 20.0 | 15.00 | 24.03 | 18.67 | 1.72 | 0.09 |
| 50.0 | 5.78 | 14.74 | 9.24 | 1.93 | 0.20 |

**Tab. 3.5:** Table showing characteristics of distribution with 100 samples generated for input setting [8.89, 50, 50, 0.3, 1000] with sample size $= 25cm * 50cm$

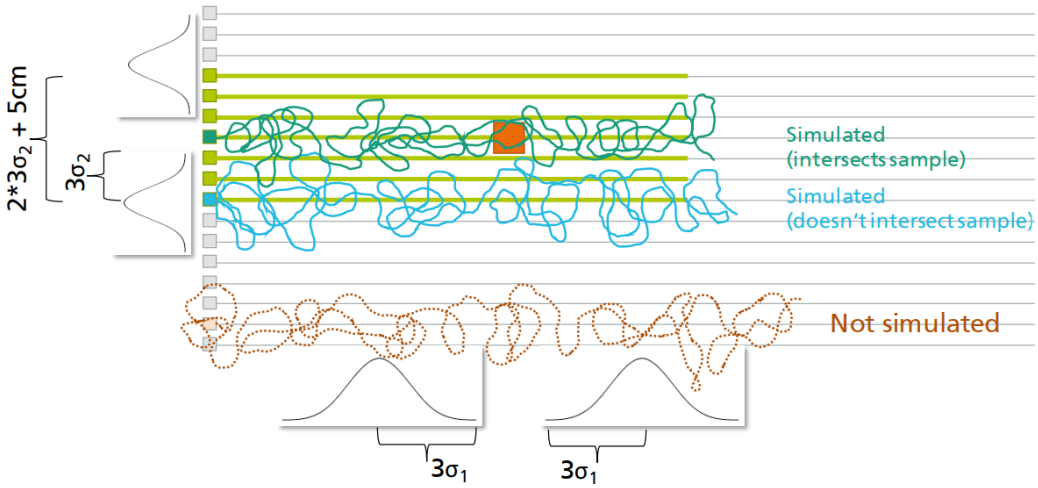



**Fig. 3.3:** Diagram showing the construction of simulated fibers based on $\sigma_1$ and $\sigma_2$ values

**Proposed setting of the nonwoven sample construction :** Figure 3.4 displays the total construction area of the sample.

Total construction: width W, height H

Sample rectangle: width $rx = 25cm$, height $ry = 50cm$

$h = max(3\sigma_2 + 5mm, 10mm)$ and $w = max(5\sigma_1 + 15mm, 100mm)$

$W = 2w + rx$ and $H = 2h + ry$

$BeltSpinRatio = \frac{BeltSpeed}{SpinningSpeed}$

Fiber length L to create width W: $L = \frac{W}{BeltSpinRatio}$

Number of spin positions: $\#Sp = (H/distance) + 1$

Number of points in fiber(N):

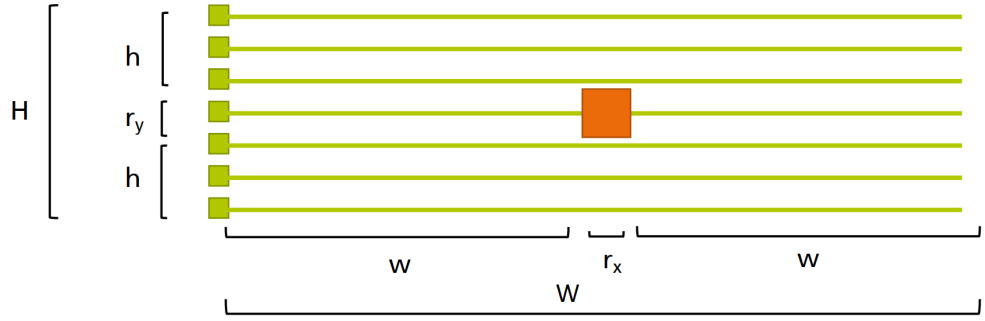



**Fig. 3.4:** Total construction size of the nonwoven sample

$L = N * ds$ where ds is Spatial Descretization which is a constant

$N = round(L/ds)$

## 3.3 Creation of Input Features Database

The correctness of the machine learning models is significantly decided by the quality of the data upon which it is trained on. Any kind of skewness or irregularity in the data leads to a biased model. Hence in this section we discuss a few techniques to generate a well balanced dataset for training our machine learning models:

1. **Design of Experiments(DOE) [WA15]:** is an applied branch of statistics which is used for the analysis and collection of data. It allows the manipulation of input features which helps in determining their effect on desired output.

   **Latin Hypercube Sampling (LHS) [HD03]:** It's a statistical method under DOE which is used to sample random numbers. Advantage of using this sampling is that it distributes samples evenly over sample space. We are using this technique to generate our database of input features because it covers most of the variance in the input parameters.

2. **Discrete Samples:** We also used few interesting discrete input values. The goal was to set up a smaller number of inputs on a lattice, and add additional inputs randomly distributed in the range using Latin Hypercube Sampling.

**sampling scale:** We started with linear scale but switched to logarithmic scale to reduce the response to skewness towards large values.

| | Sigma_1 | Sigma_2 | A | BeltSpinRatio | SpinPositionsPerMeterInverse | RandomSeeds |
|---|---|---|---|---|---|---|
| **0** | 1.000000 | 1.000000 | 1.000000 | 0.010000 | 200.000000 | [1213095470, 442944496, 634841805] |
| **1** | 1.000000 | 1.000000 | 1.000000 | 0.010000 | 200.000000 | [1642774584, 414207576, 1611613430] |
| **2** | 1.000000 | 1.000000 | 1.000000 | 0.010000 | 200.000000 | [1248361643, 518153081, 1721992811] |
| **3** | 1.000000 | 1.000000 | 1.000000 | 0.010000 | 200.000000 | [1834129115, 952397124, 1184676440] |
| **4** | 1.000000 | 1.000000 | 1.000000 | 0.010000 | 200.000000 | [1274595782, 707387979, 1557540557] |
| **...** | ... | ... | ... | ... | ... | ... |
| **311735** | 3.513361 | 20.232137 | 23.026302 | 0.180353 | 837.352405 | [1003094736, 591635343, 1961093723] |
| **311736** | 3.513361 | 20.232137 | 23.026302 | 0.180353 | 837.352405 | [953213396, 1283929761, 639338037] |
| **311737** | 3.513361 | 20.232137 | 23.026302 | 0.180353 | 837.352405 | [538647705, 364740067, 1005850105] |
| **311738** | 3.513361 | 20.232137 | 23.026302 | 0.180353 | 837.352405 | [1442998392, 648729299, 2074783063] |
| **311739** | 3.513361 | 20.232137 | 23.026302 | 0.180353 | 837.352405 | [635667682, 806371089, 1143299561] |

311740 rows × 6 columns

**Fig. 3.5:** Overview of the Input Database

## 3.4 Conclusion

We generated 50,000 samples from latin hypercube sampling and the combination discrete input values yielded us 12348 samples. As we discussed in the previous section we ran each sample 5 times. Hence the final database consists of 311740 rows as shown in the Figure 3.5.

**Output Data Generation:** We divided the input data into 16 batches of 20,000 rows each (except the last batch which had 11740 rows) and wrote a python script which sequentially takes each row of the batch and runs the SURRO simulation. The outputs of the simulations are the CV values at seven resolutions. These values along with the corresponding input rows are used as our final database for training machine leaning models.

# 4 Data Preparation and Regression Models

The characteristics of a machine learning model is predominantly determined by the data with which it is trained on. This chapter deals with introspecting the training data to get deeper insights about the same. This is then followed by determining suitable machine learning models which best capture the relationship between the input and output data.

It is common that in a multi-dimensional data setting, there could be a huge offset in the scales and ranges of different features. If features in the data are used in their original scale, the predictions of the models might not be reliable. For example in our problem setting $BeltSpinRatio$ is a continuous variable which takes the range between 0.01 and 0.25 and $SpinPositionsPerMeter$ is a discrete variable that takes the range between 200 and 10000. To ensure that these features contribute equally to the model training we perform feature scaling.

In recent times, extensive research is being done in the area of machine learning which has led to several learning algorithms. To determine which learning algorithm best fits our data, we evaluated algorithms such as linear regression [SL12], polynomial regression [Ost12], random forests [Bre01], bayesian regression [BT03] and neural networks [Spe+91]. A best model in terms of accuracy is chosen for the visual analysis.

## 4.1 Exploratory Data Analysis(EDA)

Before training the model we performed exploratory data analysis to summarize the generic characteristics from the dataset. The main purpose of EDA is to help look at data before making any assumptions. We can also identify obvious errors, as well as better understand the patterns within the data, detect outliers and find relations among the variables.

**Pairs Plot or Scatter plot Matrix:** We used pairs plot to determine the distribution of input variables. We also can find the correlation between input and output variables.

In the Figure 4.1, the diagonal represents the histograms describing the distribution of the variables. The upper triangle of the matrix shows the scatter plot of one variable with respect to the other. The triangle of the matrix describes the Pearson Correlation Coefficient between the variables.

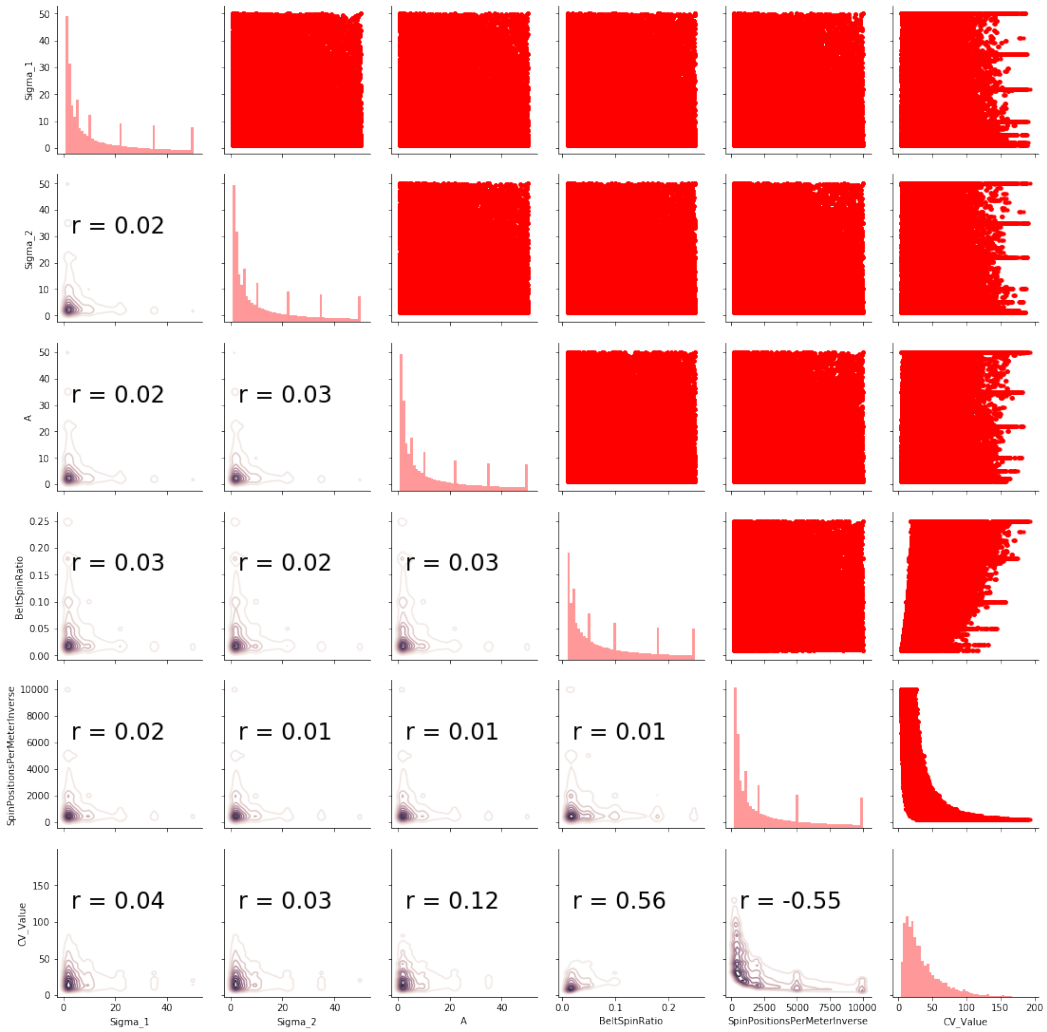



**Fig. 4.1:** Pair plot showing the relationship between the features and output for resolution 0.5

Pearson Correlation [Ben+09]: It's a statistical measure for determining the linear relationship between two continuous variables. Figure 4.2 shows the Pearson coefficients between input parameters and output values for resolution 1mm.

It is calculated as:

$$\rho_{x,y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.1}$$

where, $cov$ is the covariance, $\sigma_X$ is the standard deviation of X, $\sigma_Y$ is the standard deviation of Y

| | Sigma_1 | Sigma_2 | A | BeltSpinRatio | SpinPositionsPerMeterInverse | MeanWeight | CV_Value |
|---|---|---|---|---|---|---|---|
| **Sigma_1** | 1.000000 | 0.018842 | 0.023514 | 0.026356 | 0.016783 | 0.026507 | 0.065964 |
| **Sigma_2** | 0.018842 | 1.000000 | 0.025931 | 0.024188 | 0.012299 | 0.021179 | 0.043385 |
| **A** | 0.023514 | 0.025931 | 1.000000 | 0.025373 | 0.010811 | 0.023741 | 0.203328 |
| **BeltSpinRatio** | 0.026356 | 0.024188 | 0.025373 | 1.000000 | 0.008543 | 0.530158 | 0.535987 |
| **SpinPositionsPerMeterInverse** | 0.016783 | 0.012299 | 0.010811 | 0.008543 | 1.000000 | 0.639203 | -0.542472 |
| **MeanWeight** | 0.026507 | 0.021179 | 0.023741 | 0.530158 | 0.639203 | 1.000000 | -0.189641 |
| **CV_Value** | 0.065964 | 0.043385 | 0.203328 | 0.535987 | -0.542472 | -0.189641 | 1.000000 |

**Fig. 4.2:** Table showing the Pearson Correlation Coefficients between input parameters and the output for resolution 1mm

Spearman Correlation [CD10]: It measures the correlation between two variables using a monotonic function. Figure 4.3 shows the Spearman coefficients between input parameters and output values for resolution 1mm.

| | Sigma_1 | Sigma_2 | A | BeltSpinRatio | SpinPositionsPerMeterInverse | MeanWeight | CV_Value |
|---|---|---|---|---|---|---|---|
| **Sigma_1** | 1.000000 | 0.000182 | 0.004122 | 0.006463 | 0.004030 | 0.006186 | 0.051584 |
| **Sigma_2** | 0.000182 | 1.000000 | 0.005462 | 0.005574 | -0.000659 | 0.003451 | 0.027719 |
| **A** | 0.004122 | 0.005462 | 1.000000 | 0.007250 | 0.000039 | 0.004460 | 0.147446 |
| **BeltSpinRatio** | 0.006463 | 0.005574 | 0.007250 | 1.000000 | -0.004556 | 0.623769 | 0.578582 |
| **SpinPositionsPerMeterInverse** | 0.004030 | -0.000659 | 0.000039 | -0.004556 | 1.000000 | 0.764291 | -0.769878 |
| **MeanWeight** | 0.006186 | 0.003451 | 0.004460 | 0.623769 | 0.764291 | 1.000000 | -0.222887 |
| **CV_Value** | 0.051584 | 0.027719 | 0.147446 | 0.578582 | -0.769878 | -0.222887 | 1.000000 |

**Fig. 4.3:** Table showing the Spearman Correlation Coefficients between input parameters and the output for resolution 1mm

We calculated the correlation between input and output variables for all the resolutions as seen in the Figure 4.4.

From the initial exploratory data analysis we found that the input parameters $Sigma\_1$, $Sigma\_2$, $A$ and $BeltSpinRatio$ are positively correlated to the output and $SpinPositionsPerMeter$ is negatively correlated.

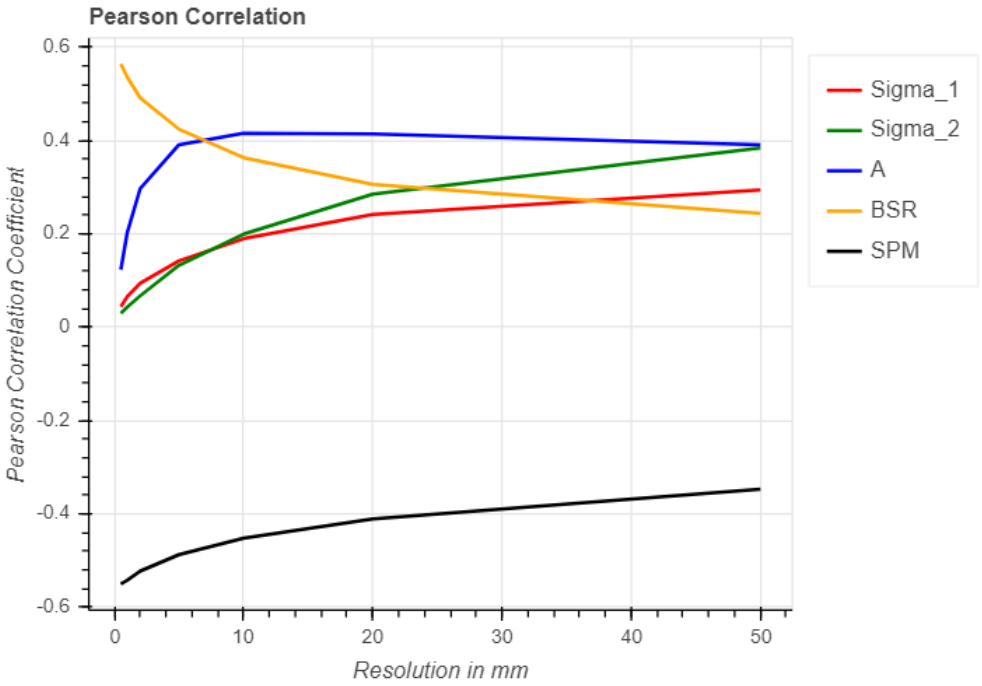



**Fig. 4.4:** Correlation of input parameters and CV Values at different resolutions.

## 4.2 Data Preparation:

In this section, we perform data preprocessing steps to tailor the data for our machine learning models. These steps include data cleaning, feature scaling and data splitting.

1. **Data Cleaning:** RandomSeeds is a part of input feature and included to reproduce the simulation run. Since it does not have any influence on the output values, we discard this feature before training the ML models.

2. **Feature Scaling:** As discussed previously, our input features are measurements of different units and these can introduce bias in the ML models. Hence, we standardize our data by re-scaling each feature to have properties of standard normal distribution with mean 0 and standard deviation 1.

   For each input value $x$, we calculate the re-scaled value(which is also called as z-score) using the formula

$$z = \frac{x - \mu}{\sigma} \tag{4.2}$$

where $z$ is the re-scaled value, $\mu$ and $\sigma$ are the mean and standard deviation of the corresponding feature distribution.

3. **Splitting the data into training, validation and testing sets:** To make our ML models generalized and not biased towards our dataset we split the dataset into $80\%$ training data and $20\%$ testing set. The training data is further divided into training set and validation set. The training set is used to train the model and validation set is used to tune the model hyper-parameters. The testing set is used for unbiased evaluation of the final model.

   Splitting the data into above three sets also prevents our models from two main hurdles in machine learning:

   a) Under-fitting: It is the case where the model has not trained enough that it gives unreliable predictions both on training and testing data set.

   b) Over-fitting: It is the case where the model error is very small, but the model is not generalized for new unseen data and hence unreliable. This is due to the model learning too much from the training data set. Such models will result in very bad accuracy for the testing set.

   Since we run same parameter setting 5 times, our database will contain 5 copies of this parameter setting. We have to make sure that these 5 rows should either fall completely in the training set or in testing set. This ensures the two sets are completely different from each other there by reducing chance of over-fitting. For this, we grouped out data with 5 input values and provided indexes to each groups. We then shuffled this indexes and split them into $80:20$ ratio for training and testing sets respectively.

## 4.3 Criteria for Selection of Regression Models

Our data consists of 5 continuous input features which are used to predict 7 continuous output values. This setting in statistics is called as multiple-input, multiple-output regression. So we need to train regression models for our dataset.

The following are the criteria for selecting regression models:

1. Accuracy of the model: This ensures that the prediction error is minimum.
2. Confidence on the model's prediction: This criterion ensures the model's confidence with respect to the computed coefficients is as high as possible.

3. Interpretability of the model: It becomes difficult to interpret the model as its complexity increases. Hence, we try to ensure that the model is relatively simple.

4. Scalability of the model: The performance of certain models deteriorates as the volume of the data grows. Hence, we need to ensure that the ML models are can scale with the data.

### 4.3.1 Metrics used for evaluation:

The models are evaluated using three metrics. Here, $n$ refers to the number of data points, $y_i$ and $\hat{y}_i$ refer to the actual and predicted values respectively for the data point $i$ and $\bar{y}$ is the mean value of the data points.

1. **Mean Absolute Percentage Error(MAPE):** [De +16] It is a statistical measure of how accurate a regression model is. It measures this accuracy as a percentage. Lower the value better the model. The value lies between $0$ to $100$. MAPE is calculated as:

$$MAPE = \left(\frac{1}{n} * \sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) * 100 \tag{4.3}$$

2. **Mean Squared Error(MSE):** [DJR+04] The mean squared error measures the average of squares of the errors. Lower the value better the model. MSE is calculated as:

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.4}$$

3. $R^2$ **Score (Coefficient of determination):** [Nag+91] It is the measure of how close the data points are to the fitted regression line. The value ranges between $0\%$ to $100\%$ with 0 being worst fit to 100 being best fit. It explains how much of the variance of actual data is explained by the predicted values.

$$R^2(y, \hat{y}) = 1 - \left(\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}\right) \tag{4.5}$$

### 4.3.2 Regression Models

We have evaluated our dataset using the following regression models:

1. **Multi-Linear Regressor:** We examined whether the relationship between the input parameters and output values is linear by trying to fit different flavors of linear regression model. Each of the variant differs in the type of regularization used in the optimization function. As we can observe from the Table 4.1, a linear model is not adequate to fit our data as its error rate is very high. This is hinting us that the relationship between the input parameters and output values is not linear.

| Regressor Flavor | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| Vanilla | 94.5723 | 94.55 | 0.60 |
| Ridge | 94.5719 | 94.55 | 0.60 |
| Lasso | 107.6147 | 99.09 | 0.49 |
| ElasticNet | 91.2243 | 117.51 | 0.48 |

**Tab. 4.1:** Prediction Results of Multi-Linear Regressor on the Dataset for all the resolutions

2. **Polynomial Regressor:** We tried to fit polynomial regression model for the data which is a type regression where the relationship to the output values are modelled as $n$th degree polynomial in input parameters.

   Selection of degree of the polynomial: In polynomial regression selecting the degree of the polynomial is very important. Choosing a very small degree would under-fit the model while selecting very large value would over-fit the model. To determine the right degree we trained our model with degrees 1-11 and recorded the model errors for each degree. As we can see from the Figure 4.5, the training, validation and testing error decreases initially as we go for higher degrees. At degree 9 , the validation and test errors start increasing and the training error is still decreasing indicating that the model is starting to over-fit the data. This is considered as the sweet spot in machine learning. Hence we selected degree 9 as a suitable degree for our model. The Table 4.2 shows the prediction error for polynomial regression with degree 9.

   Disadvantages of Polynomial Regression: The complexity of the model increases as we go for higher degrees. We observed that the training time increases with increase in the dataset size. So these models do not scale well for larger datasets.

3. **Random Forest Regressor:** It is an ensemble regressor which uses multiple decision trees, each training with different data sample chunks using a method

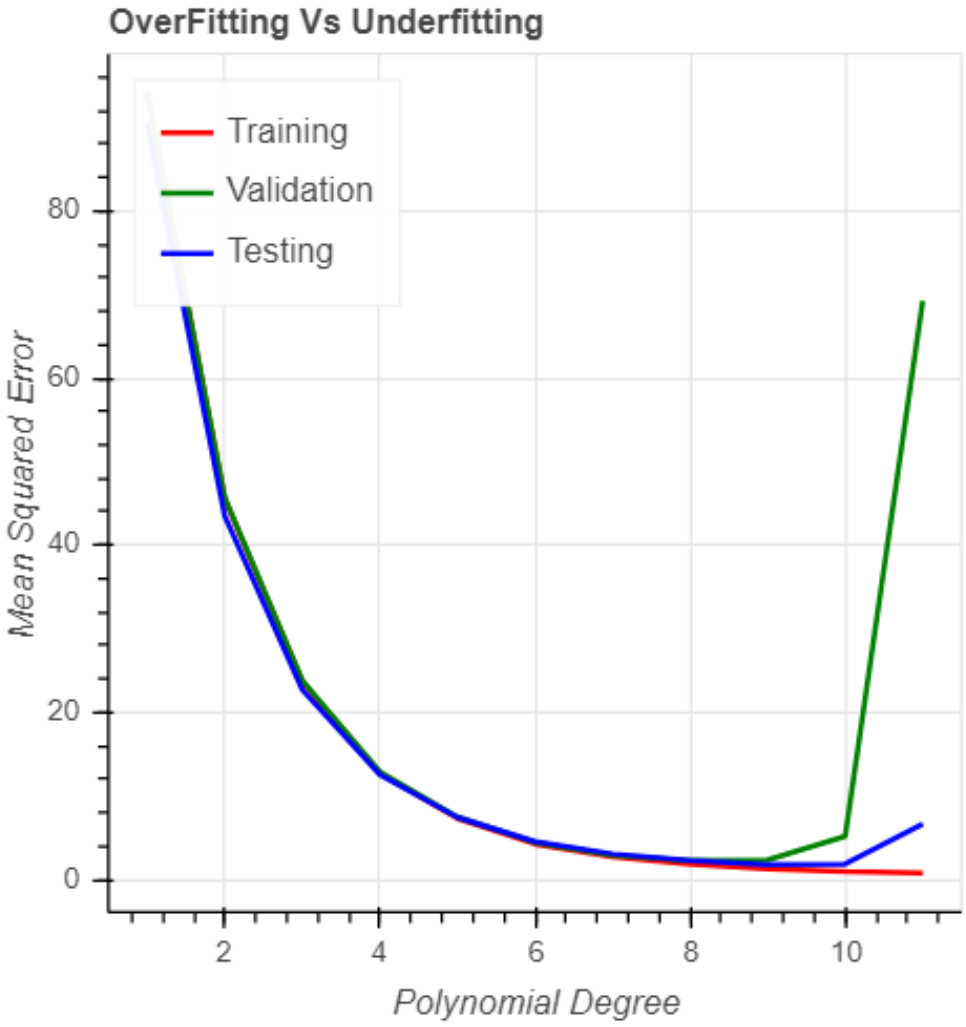



**Fig. 4.5:** Plot showing the selection of degree of a polynomial

called "Bootstrap Aggregation". It uses averaging to improve the accuracy of the model. Since each decision tree is training with a different chunk of data samples, over-fitting of the model is avoided. Table 4.3 shows the prediction errors for Random Forest Regressor on our dataset.

Feature Importance: Random Forests implicitly calculates a measure for feature importance. Higher the value, higher is the importance of the feature for prediction.

The values are as follows:

$Sigma\_1$: 0.01686607, $Sigma\_2$: 0.02509503, A: 0.06872208, BSR: 0.35838257, SPM: 0.53093424

| Resolution(in mm) | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| 0.5 | 4.46 | 3.84 | 1.00 |
| 1 | 5.52 | 3.83 | 0.99 |
| 2 | 6.57 | 3.04 | 0.99 |
| 5 | 5.90 | 0.70 | 0.99 |
| 10 | 7.21 | 0.43 | 0.99 |
| 20 | 10.38 | 0.34 | 0.99 |
| 50 | 20.30 | 0.33 | 0.96 |
| All | 8.62 | 1.79 | 0.99 |

**Tab. 4.2:** Prediction Results of Polynomial Regressor with degree 9 on the Dataset

| Resolution(in mm) | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| 0.5 | 2.59 | 2.76 | 1.00 |
| 1 | 3.36 | 2.52 | 0.99 |
| 2 | 4.71 | 2.44 | 0.99 |
| 5 | 7.90 | 2.01 | 0.98 |
| 10 | 11.25 | 1.43 | 0.98 |
| 20 | 15.36 | 0.92 | 0.97 |
| 50 | 23.73 | 0.59 | 0.94 |
| All | 9.84 | 1.81 | 0.98 |

**Tab. 4.3:** Prediction Results of Random Forest Regressor on the Dataset

Hence the feature $SPM$ is of higher importance as per this algorithm.

4. **Bayesian Regressor:** In Bayesian regression, we assume that each of the coefficients computed by the regressor are drawn from probability distributions instead of them being single estimates. Hence, we determine the parameters(mean and variance) of the posterior distribution associated with these coefficients. This allows us to determine the confidence of the model prediction. We can see the results of Bayesian Regression model predictions in the Table 4.4.

5. **Neural Network Regressor:** An artificial neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output. A typical neural network fundamentally comprises units called nodes which are arranged in the form of layers. The input features are passed through these layers (input, hidden and output) with the series of non linear operations to get the final prediction.

| Resolution(in mm) | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| 0.5 | 4.07 | 3.27 | 1.00 |
| 1 | 5.04 | 3.25 | 0.99 |
| 2 | 6.03 | 2.56 | 0.99 |
| 5 | 5.51 | 0.57 | 1.00 |
| 10 | 6.93 | 0.37 | 0.99 |
| 20 | 9.60 | 0.30 | 0.99 |
| 50 | 16.86 | 0.28 | 0.97 |
| All | 7.72 | 1.51 | 0.98 |

**Tab. 4.4:** Prediction Results of Bayesian Polynomial Regressor with degree 9 on the Dataset

We design a Neural Network to learn the non-linear dependency of our output variables with respect to the input features. The trained network can then be used for predictions. Table 4.5 shows the prediction errors for Neural Network Regressor.

| Resolution(in mm) | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| 0.5 | 0.80 | 0.50 | 1.00 |
| 1 | 1.02 | 0.47 | 1.00 |
| 2 | 1.38 | 0.39 | 1.00 |
| 5 | 2.18 | 0.27 | 1.00 |
| 10 | 3.33 | 0.27 | 1.00 |
| 20 | 5.31 | 0.25 | 0.99 |
| 50 | 12.31 | 0.27 | 0.97 |
| All | 3.76 | 0.35 | 0.99 |

**Tab. 4.5:** Prediction Results of the Best Neural Network Regressor on the Dataset

Hyper-parameter optimization for best Neural Network Architecture: The accuracy of the neural network is determined by optimal choice of parameters that decide its architecture. These parameters are called hyper parameters and are often decided manually. Hence, to find the optimal hyper parameters keeping the desirable accuracy, we performed network parameter tuning.

Hyper Parameters and their corresponding ranges for tuning:

a) Number of Hidden Layers: 1-5.

b) Number of nodes in each layer: 8-1024 with increment of 8.

c) Activation functions: ['relu', 'tanh', 'sigmoid'].

The tuning resulted in the following architecture illustrated in Figure 4.6 and the activation function 'relu' was selected.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 256)               1536
_________________________________________________________________
dense_1 (Dense)              (None, 512)               131584
_________________________________________________________________
dense_2 (Dense)              (None, 512)               262656
_________________________________________________________________
dense_3 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_4 (Dense)              (None, 768)               197376
_________________________________________________________________
dense_5 (Dense)              (None, 7)                 5383
=================================================================
Total params: 729,863
Trainable params: 729,863
Non-trainable params: 0
```

**Fig. 4.6:** The Neural Network architecture with the best accuracy.

**Trade off the between Complexity and Accuracy** As discussed in the criteria section the previously chosen architecture in the Figure 4.6, has 729,863 trainable parameters which increases the complexity of the model. Hence in order to preserve the simplicity of the model, we performed the tuning step again with lower ranges for hyper parameters. This resulted in the simplified architecture which can be seen in the Figure 4.7

The accuracy of the simpler model was relatively lower than the model from Figure 4.6. However, as seen from the Tables 4.5 and 4.6 the reduction in the accuracy is acceptable, when compared to the huge reduction in the number of trainable parameters.

## 4.4 Automated Analysis of Regression models:

It is cumbersome to perform the previously discussed tasks every time there is an addition or modification in the data, choice of regression models etc. Hence, we implemented a framework that takes care of the following steps:

1. Prepares the data.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_13 (Dense)             (None, 16)                96
_________________________________________________________________
dense_14 (Dense)             (None, 32)                544
_________________________________________________________________
dense_15 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_16 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_17 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_18 (Dense)             (None, 7)                 231
=================================================================
Total params: 4,039
Trainable params: 4,039
Non-trainable params: 0
```

**Fig. 4.7:** The Neural Network architecture with reduced parameters.

2. Executes regression models - Saves the models and scalers.
3. Logs the results into a csv file as seen in Figure 4.8.
4. Selects the best model based on MAPE - Saves this model as a pickle file.

This makes our framework a plug-and-play model with which future integrations such as novel unseen data, novel regressor models, new evaluation metric etc. are made very easy. This in turn enables our visual analytic tool to automatically choose the best performing model provided by our framework.

## 4.5 Conclusion

After performing the series of experiments as discussed in this chapter, we concluded that the Neural Network Regressor performed best in terms of accuracy, efficiency and scalability. So it has been selected as the surrogate ML model for visual analysis.

| Resolution(in mm) | MAPE(%) | MSE | R2-Score |
|---|---|---|---|
| 0.5 | 1.62 | 0.78 | 1.00 |
| 1 | 1.90 | 0.71 | 1.00 |
| 2 | 2.46 | 0.67 | 1.00 |
| 5 | 3.78 | 0.45 | 1.00 |
| 10 | 5.19 | 0.37 | 0.99 |
| 20 | 7.82 | 0.35 | 0.99 |
| 50 | 16.37 | 0.34 | 0.96 |
| All | 5.59 | 0.52 | 0.99 |

**Tab. 4.6:** Prediction Results of Optimized Neural Network Regressor on the Dataset

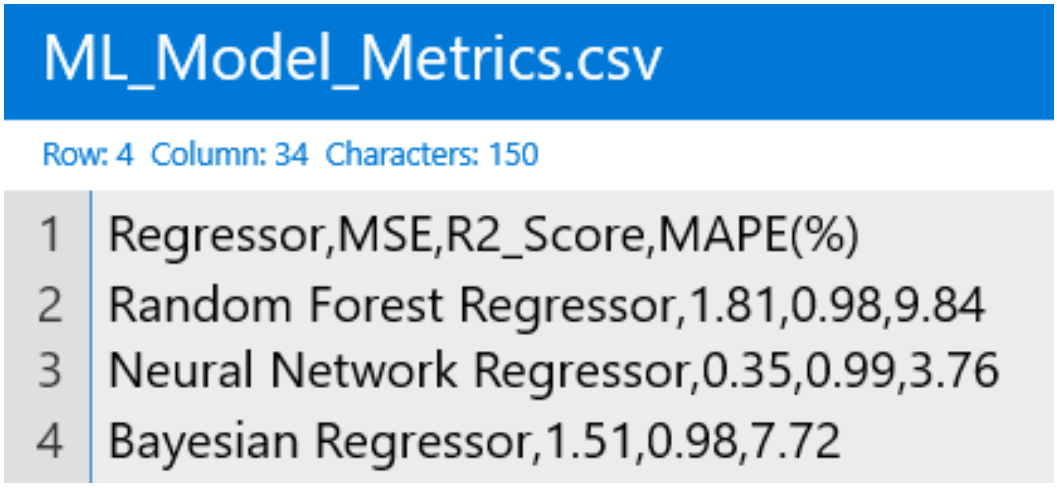



**Fig. 4.8:** The csv file comparing the metrics of ML models.

# 5 Visual Analytic Tool

This chapter involves the details about the interactive visualization tool built to assist the user to understand and analyse the data. It uses the the best regression model as a surrogate model for the analysis. The tool has four main objectives: first, understanding the relation between input parameters and output values and the sensitivity of the output values to minor changes in the input parameters. Second, visually aiding the users to reach their optimal set of input parameters which result in the desired output. Third, to show the patterns, clusters and anomalies in the data which help in better understanding of the data. Fourth, to help the user navigate through the data in real-time.

The technologies used to build this tool involve, Plotly, which is an interactive, open-source, and browser-based graphing library for Python. DASH, which is a productive Python framework for building web analytic applications. Both the above frameworks are used as a front end for our tool.

## 5.1 Parameter Tuner

The parameter tuner as shown in Figure 5.1 consists of 5 knobs to tune different parameters. The user can dynamically see the changes corresponding to different parameter settings on the graph and the table displayed beside the tuner. The functionalities of the tuner are explained in detail as follows:

1. **Analysis of the output for all the resolutions:** This allows the user to change the input parameter and see the corresponding CV values for all the seven resolutions. The output graph which is a line chart of CV values vs resolution, dynamically gets updated upon the change in input values. Each input feature is provided with sliders(knobs). The range of each of these sliders is set according to the range allowed for the respective input feature. Along with the slider, the user is also given an option to enter the input values. The values in the sliders and the respective input boxes are coupled with each other and are updated dynamically.

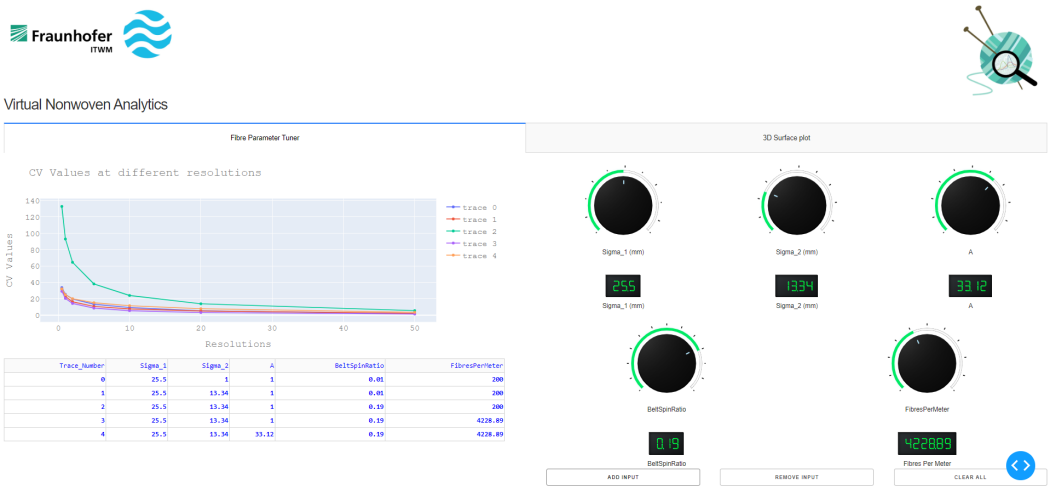



**Fig. 5.1:** Basic Parameter Tuner with sliders for Input Features

2. **Comparison between more than one parameter setting:** This allows the user to compare two or more parameter settings as shown in Figure 5.2 using the following actions:

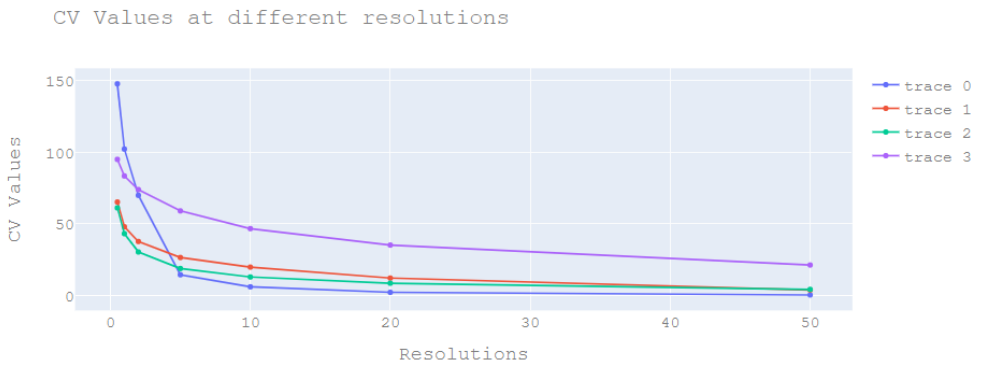
   a) Add the output of more than one parameter setting to the graph for comparison.

   b) Delete the last added output from the graph.

   c) Clear all the outputs from the graph to reset the analysis.

   This graph is also provided with the dynamic table which updates according to the operation performed above. The table contains the individual values of the input parameter corresponding each output on the graph.

3. **Analysis of the output for individual resolutions over a range:** If the user is interested to see the changes in one resolution, he can switch to individual resolution analysis where the user can allow one parameter to remain free and freeze the other four input parameters and plot CV values over the whole range of the free parameter as shown in the Figure 5.3. Keeping the frozen parameters unchanged, we generate 50 Latin Hypercube samples corresponding to the free parameter within its corresponding range and produce the output with each of these 50 parameter settings. The same is done for all the seven resolutions separately. The visualization shows the effect of the free parameter on the output while other parameters are frozen.




| Trace_Number | Sigma_1 | Sigma_2 | A | BeltSpinRatio | FibresPerMeter |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0.25 | 200 |
| 1 | 50 | 1 | 1 | 0.04 | 200 |
| 2 | 50 | 50 | 1 | 0.04 | 200 |
| 3 | 1 | 50 | 50 | 0.04 | 200 |

**Fig. 5.2:** UI Showing the comparison between 4 different nonwoven material for all the resolutions

4. **Expected Mean Weight calculation:** Expected Mean Weight is the weight of the nonwoven web per square meter. It is one of the important factors while analysing the quality of the nonwoven. It is calculated as follows:

$$ExpectedMeanWeight = titer * SpinPositionsPerMeter * BeltSpinRatio$$

Where SpinPositionsPerMeter and BeltSpinRatio are input features and titer is weight per unit length which is a constant user input.

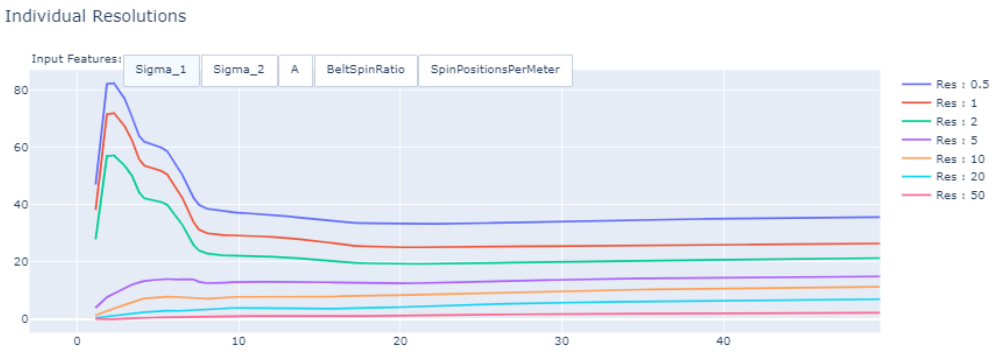



**Fig. 5.3:** Individual resolutions plot showing the changes in output for the whole range of $Sigma\_1$ values

The user is also provided with the Expected Mean Weight for the current parameter setting which aids the analysis. There is also an input field for changing the titer value if required.

## 5.2 3D Surface plot

3D Surface plot is created to give the user a better understanding of the influence of the input parameters by freezing a combination of two parameters. Though the user is not allowed to change these parameters, they act as free parameters, while, the other three parameters that the user is allowed to vary remain constant and act as frozen parameters for the generation of the data samples. The Figure 5.4 shows the overview of the 3D Surface Plot.

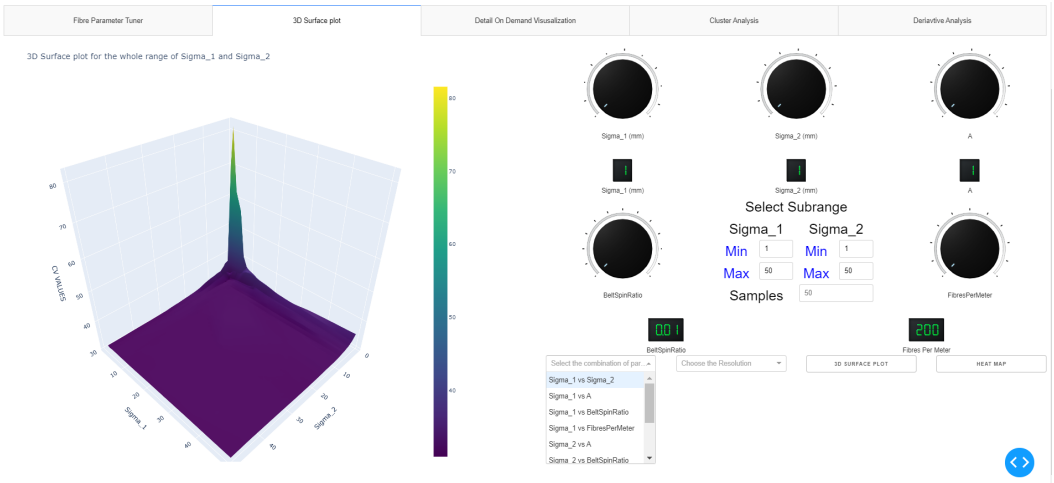



**Fig. 5.4:** 3D Surface Plot showing CV values for full range if $Sigma\_1$ and $Sigma\_2$ while other parameters kept constant

Creation of 3D Surface Plot involves the following steps:

1. 50 Latin Hypercube samples are generated for the two free parameters between the allowed range for these parameters.
2. The combination of the free parameters produces 2500 (i.e.,$50 * 50 = 2500$) samples.
3. The three frozen parameters are appended to each of the above combinations to form a dataset with 2500 input rows.
4. In the 3D Surface graph, the output for the selected resolution along z axis is plotted with the free parameter values along x and y axes.

The above graph helps the user to select the parameter setting of their interest for the analysis. It also allows them to observe the influence of two free parameters with respect to the output. This type of analysis is called local to global navigation in interactive visualization. The 3D Surface also helps user to visually analyse the local minima and maxima.

The additional visual aids provided for the analysis are:

1. Dynamic update of the plot in real-time: The plot is dynamically updated corresponding to the change in the input feature values made using sliders. The updates to the graph are in sync with the change in both input values from the sliders and the frozen combination of parameters. The graph is updated with additional 2500 values in real-time.

2. Sub-range selection for Details on Demand Analysis: The user is provided with an option to select a sub-range for the free parameters which lie in the specific region of interest on the surface of the graph. Upon selecting the sub-range, the graph will be updated with new 2500 values for the given range of input parameters. This shows the detailed analysis of a desired range of values.

3. Analysis for individual resolutions: User is provided with a drop down to select the resolution of their interest to get the individual resolution analysis.

## 5.3 Sensitivity Analysis

Sensitivity in our Context pertains to the robustness of the output values against small changes in the input parameter settings.

The input features do not directly affect the production process but they decide the values of the process parameters which in turn affect the product quality. We cannot ascertain the process parameters to exactly have the desired values in reality because of various factors in production such as manufacturing error, human error etc. If the input parameter setting is sensitive to small changes around it, it might result in a bad quality of the end product. Hence, during the parameter space exploration, along with finding optimal parameter settings for the desired output, the user needs to make sure that these settings are non-sensitive to small changes around them. To examine these effects, the user is provided with a sensitivity analysis tab [Ber+11] which is shown in the Figure 5.5.

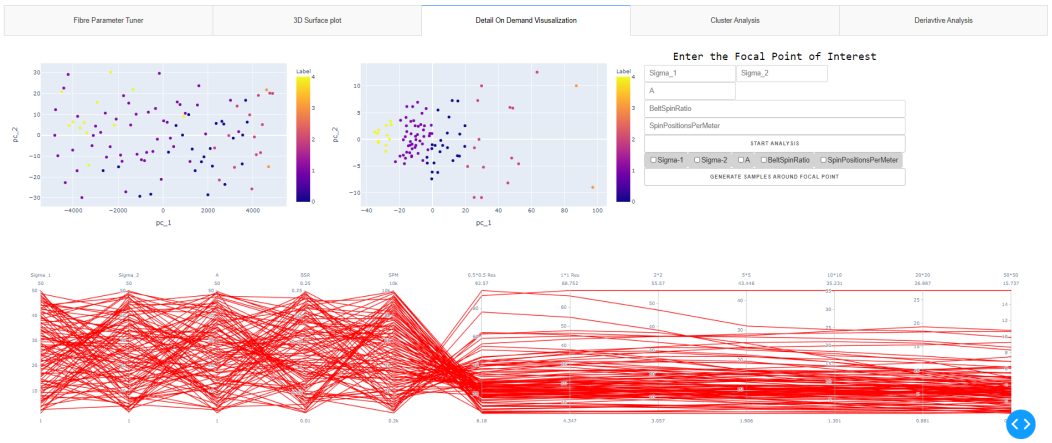



**Fig. 5.5:** Sensitivity Analysis tab showing the different components

### 5.3.1 Components:

The user interface of the sensitivity analysis tab contains 3 major components, the Input Space Graph, the Output Space Graph and the Parallel Plot. The Input Space Graph allows the user to select areas of interest, for which the corresponding CV values are then plotted in the Output Space Graph and in the Parallel Plot.

Before explaining the components, we need to know a dimensionality reduction approach: Principal Component Analysis [WEG87] which used for reducing the dimensions of a data point by projecting each of them onto only the first few principal components which results in lower-dimensional data while preserving as much of the variation in the data as possible.

**The Input Space Graph is constructed as follows:**

1. Create $n$ number of Latin Hypercube samples with dimensions $n * 5$.
2. Apply Principal Component Analysis on the input data.
3. Select the first two Principal components with highest variance to transform the data to dimension $n * 2$.
4. Assign colors to each of the data points base on their cluster id.
5. Show the data as scatter plot to the user.

**The Output Space Graph is constructed as follows:**

1. Predict CV values for the $n * 5$ input features using the ML model to get $n * 7$ output values.

2. Apply Principal Component Analysis on the output data.

3. Select the first two Principal components with highest variance to transform the data to dimension $n * 2$.

4. Assign colors to each of the data points base on their cluster id (the clustering is explained in the later section).

5. Show the data as scatter plot to the user.

Parallel Plot helps us to observe the influence of individual input features on each output value separately.

**The Parallel Plot is constructed as follows:**

1. It consists of 12 vertical lines.

2. The first 5 lines correspond to the input features and the range of each line corresponds to the range set for the input features.

3. The next 7 lines correspond to the CV values at seven different resolutions. The range of each line lies between the minimum and the maximum CV values corresponding to the resolution of that line.

### 5.3.2 Interaction Between the Components

The three main components of sensitivity analysis are in sync with each other. Upon selection of an input point in the Input Space Graph, the corresponding output component in the Output Space Graph is highlighted. A line passing through all the 5 input values and the corresponding 7 output values in Parallel Plot is also highlighted.

### 5.3.3 Aid for the Analysis:

The Sensitivity Analysis tab also includes five input fields corresponding to the five features that are obtained from the user. The entered values are used to predict the CV values. PCA is applied on both the user entered input and corresponding output values and they are updated on the Input and Output Space Graph respectively.

### 5.3.4 Analysis:

This sub-section talks about the analysis of the sensitivity of the input parameter setting. The selected input parameter for the analysis is called the focal point. The steps to carry out the analysis are as follows:

1. Selection of the focal point: The user selects the data point of his interest either from the Input Space Graph (by clicking on it) or by entering it manually. The selected focal point gets highlighted on all the three components.

2. Selection of area and population of local neighbourhood around the focal point: The user can generate $n$ data points around the focal point with a radius $r$. Here we set n=50 samples, and $r$ equal to $10\%$ of the feature range.

3. Updating the local neighbourhood: Two types of updates are provided to the user:

   a) Global Update: The values corresponding to each of the dimensions of input feature (focal point) are updated around their respective neighbourhood values.

   b) Local Update: The values corresponding to user-selected dimensions of input feature (focal point) are updated around their respective neighbourhood values.

4. Apply the changes: The 3 components are updated with the new neighbourhood values around the focal point.

This analysis for two such examples is explained in the Figure 5.6. Examples for global and local updates are explained in the Figure 5.7.

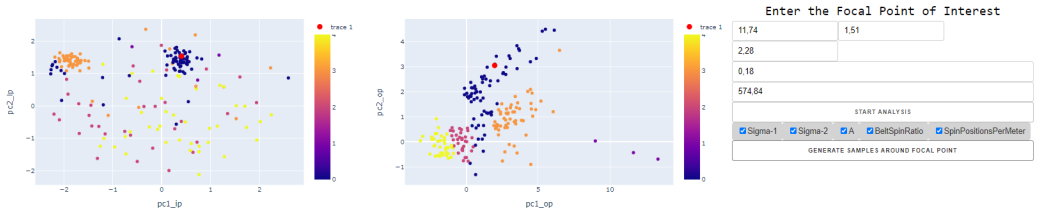



**Fig. 5.6:** Analysis showing the focal point(black dot) and the neighbourhood which are colored according to their cluster id for local update($Sigma\_2$,A,BSR)

| Focal Point | Neighbouhood Radius<br><br>R = 1/10*(range_max-range_min) | Neighborhood Population | Type of Update | Feature Value/<br>Feature Range = [Value-R, Value+R] |
|---|---|---|---|---|
| Sigma_1 = 5<br>Sigma_2 = 5<br>A = 1<br>BSR = 0.15<br>SPM = 200 | Sigma_1 = 1/10*(50-1) = 4.9<br>Sigma_2 = 1/10*(50-1) = 4.9<br>A = 1/10*(50-1) = 4.9<br>BSR = 1/10*(0.25-0.01) = 0.024<br>SPM = 1/10*(10000-200) = 980 | 50 | Local Update<br>User Selects:<br>Sigma_1<br>BSR<br>SPM | Sigma_1 = [5-4.9, 5+4.9] = [0.1, 9.9]<br>Sigma_2 = 5<br>A = 1<br>BSR = [0.15-0.024, 0.15+0.024] = [0.126, 0.174]<br>SPM = [200, 200+980] = [200, 1180] |
| Sigma_1 = 50<br>Sigma_2 = 25<br>A = 47<br>BSR = 0.25<br>SPM = 9500 | Sigma_1 = 1/10*(50-1) = 4.9<br>Sigma_2 = 1/10*(50-1) = 4.9<br>A = 1/10*(50-1) = 4.9<br>BSR = 1/10*(0.25-0.01) = 0.024<br>SPM + 1/10*(10000-200) = 980 | 50 | Global Update:<br>All features | Sigma_1 = [50-4.9, 50] = [45.1, 50]<br>Sigma_2 = [25-4.9, 25+4.9] = [20.1, 29.9]<br>A = [47-4.9, 50] = [42.1, 50]<br>BSR = [0.25-0.024, 0.25] = 0.226<br>SPM = [9500-980, 9500+980] = [8520, 10000] |

**Fig. 5.7:** Neighborhood Calculation around the Focal Point in Sensitivity Analysis

### 5.3.5 Conclusion:

In addition to the sensitivity analysis, the proposed framework can be leveraged to achieve the following navigation strategies in the interactive visualization.

1. **Visual Guidance to find the Local/Global Minima:** User can start at a certain parameter setting of his interest and generate points around it. He then selects the interesting point around this new neighborhood as the next focal point. User can repeat this until he finds the parameter setting of his interest.

2. **Global to Local Navigation:** The user is provided with the overview first and then the details are provided on demand.

## 5.4 Cluster Analysis

Clustering [XW08] is an unsupervised machine learning algorithm which involves automatic discovery of natural grouping in data. Clustering algorithms only interpret the data and find natural groups or clusters in feature space.

The goal of Cluster Analysis in our context is to find possible clusters in the output space to capture different types of model behaviours. These model behaviours are analysed by mapping the clusters found in the output space to the corresponding range of input space parameters. This analysis includes few important steps which are explained in the following subsections.

### 5.4.1 Partition of Output Space:

In this step, the output data points (CV values at all resolutions) are clustered based on two unsupervised machine learning algorithms.

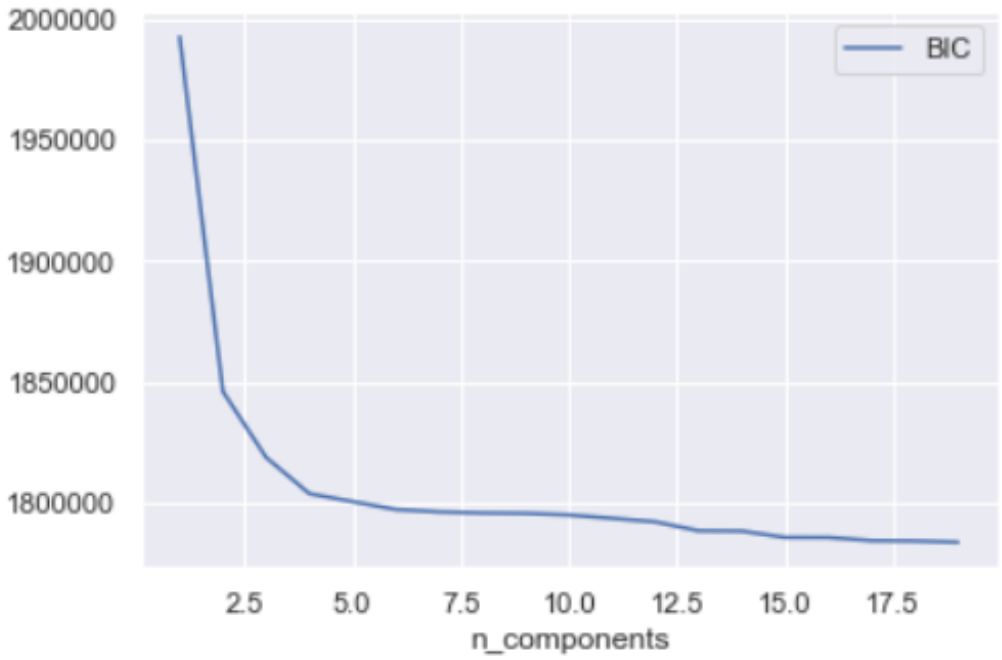



**Fig. 5.8:** Selection of number of clusters using Bayesian Information Criterion

Firstly, Principal Component Analysis(PCA) is applied on the entire output data to reduce the dimensions from seven to two. The selected two Principal components explain more than $99\%$ of the variance in the data and hence we will not lose much information by selecting these two components. These principle components are the linear combination of original dimensions.

First principal component coefficients = [ 0.36416881, 0.37780637, 0.38930036, 0.39593899, 0.39145764, 0.37781418, 0.34689816]

The explained Variance of the first principal component = 0.96360304

Second principal component coefficients = [-0.48823398, -0.39970777, -0.26307882, 0.00392448, 0.20315893, 0.39519199, 0.57895176]

The explained Variance of the second principal component = 0.03179121

Secondly, we use Gaussian Mixture Models (GMM) clustering [YLL12] to cluster the reduced output space. The more popular K-means clustering was not chosen for

this analysis as it works only if the clusters are circular and hence, will not consider the variance in our data. Thereby, we chose GMM Clustering which considers the variance in the data and handles clusters of different shapes.

Since, clustering depends on the number of clusters, choosing this number is one of the main aspects of this algorithm. The optimal number of clusters is chosen using Bayesian information criterion.

The Bayesian Information Criterion (BIC) **??** is a criterion for model selection among a finite set of models, where the model with the lowest BIC is preferred 5.8. The BIC balances the number of model parameters and the number of data points against a maximum likelihood function. In our context we prefer the number of clusters which minimises the BIC. As shown in the Figure 5.9, cluster number 5 is chosen as the optimal number.

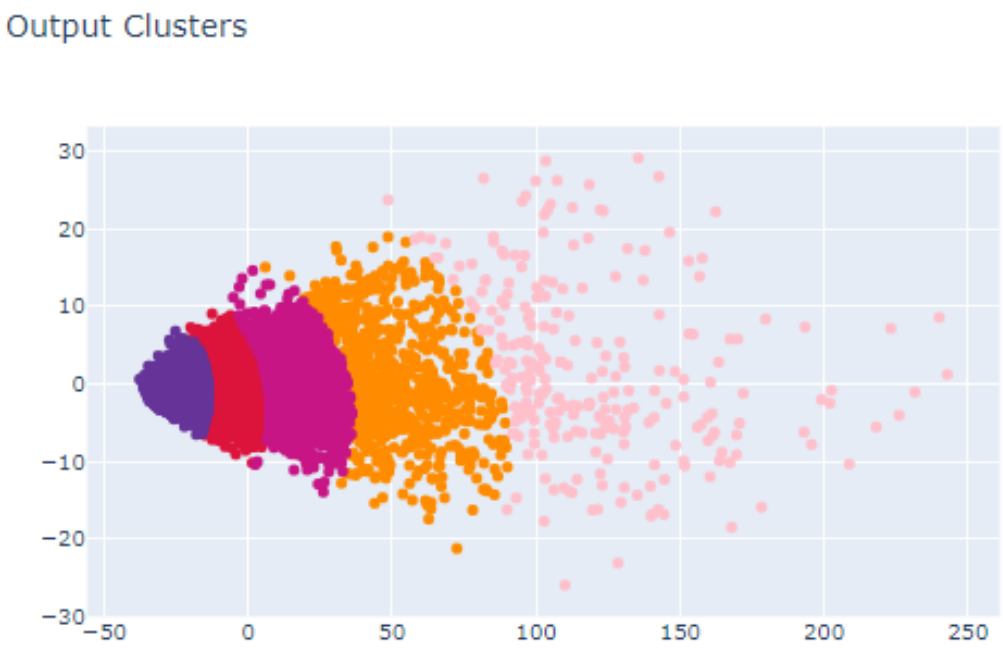



**Fig. 5.9:** Plot showing the five cluster obtained from output space

### 5.4.2 Assign a Quality Measure to the Partitioned Data

The 5 clusters obtained while partitioning the output space corresponds to 5 major model behaviours. To differentiate the clusters, a quality measure for each of is assigned. We chose average CV values in all the 7 resolutions as the quality measure

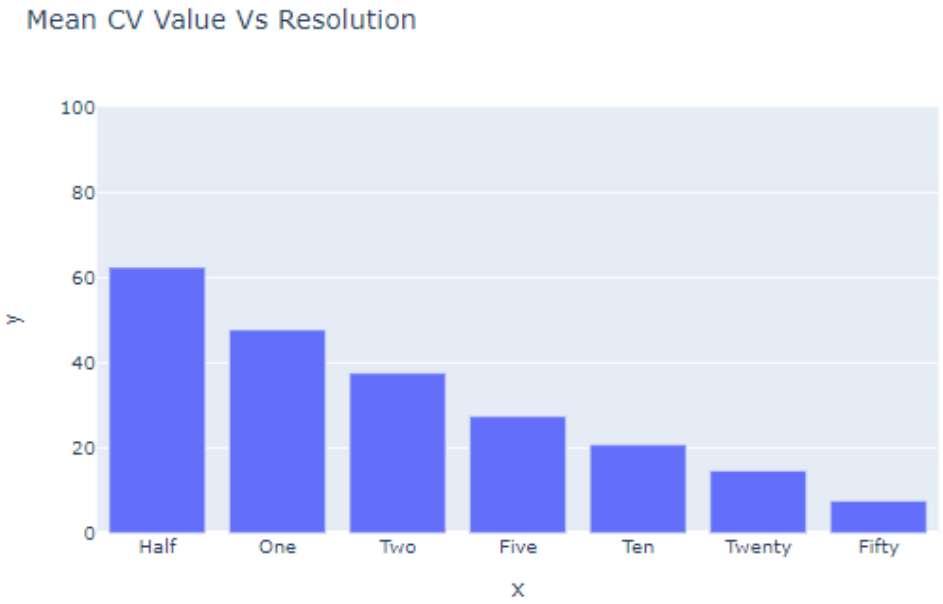



**Fig. 5.10:** Bar chart showing average cv values at all resolutions for user selected cluster

for each clusters. The clusters with lower average CV values are desired over the one with higher average CV values. This is visualized using a bar graph (Figure 5.10).

### 5.4.3 Mapping the clusters to the input space:

We need to map each of the clusters to input space to know about the range of values which generated these clusters. This gives a better overview for the user about the desired set of values for obtaining the output of their interest. Hence all the 5 clusters are mapped to the input space.

**Box-plot for analysing the distribution of input features:** We provide a box-plot for the distribution of each feature leading to the selected cluster. Box-plot displays Variations in the sample of statistical population without making any assumptions of the underlying data. It also shows the degree of dispersion, skewness and outliers in the data.

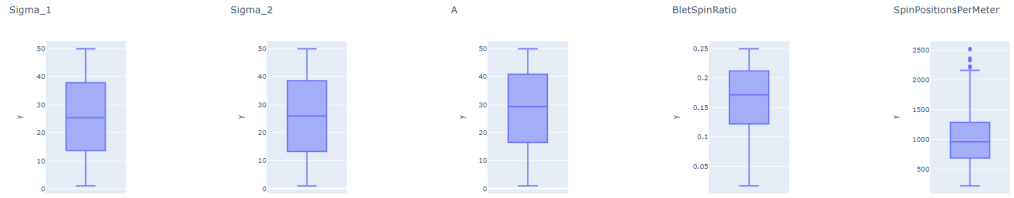



**Fig. 5.11:** Box plot showing the distribution input feature responsible for the generation of the user selected cluster

## 5.5 Combining Sensitivity and Cluster Analysis

The data points in the sensitivity analysis are colored according to their cluster numbers. For the newly entered user value and its output value, we calculate the cluster ids, color them accordingly and place them on the Input and Output Space graphs. This helps the user to already know the cluster to which his parameter setting of interest belongs hence aiding in the analysis. The cluster separation boundaries are clearly indicated and hence, the user can select the data points from the input space graph such their proximal neighbourhood does not lie in the undesirable cluster region.

## 5.6 Partial Derivative Analysis

Partial derivatives of the output with respect to the chosen input at fixed points is provided to the user to further help in visual analysis of the influence of input parameters on the output.

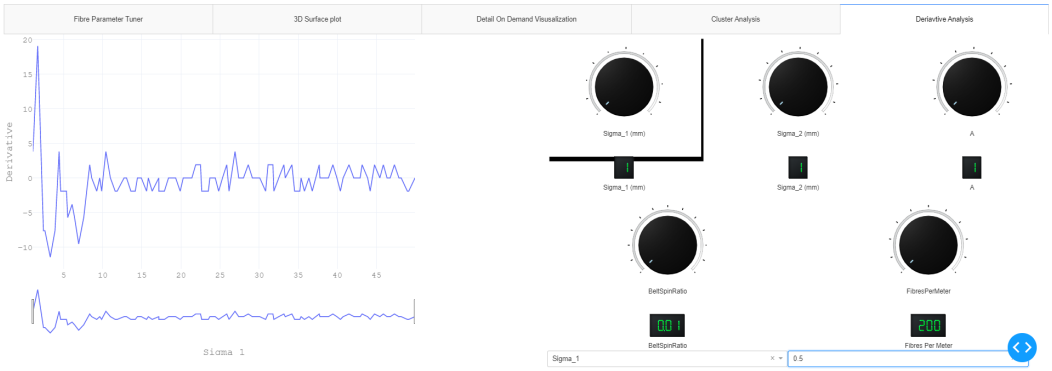



**Fig. 5.12:** Partial Derivative Analysis tab showing influence of $Sigma\_1$ for resolution $0.5$ on the output cv values

We used central finite difference [Str04] as an approximation for the partial derivatives.

For example we calculate the partial derivative of output with respect to the input feature $Sigma\_1$ at a fixed point $i$ as

$$\frac{\partial CV_i(S1,S2,A,BSR,SPM)}{\partial S1} = \frac{CV_i(S1+h,S2,A,BSR,SPM)-CV_i(S1-h,S2,A,BSR,SPM)}{2h}$$

for small value of $h$.

In the above equation, $Sigma\_1$, $Sigma\_2$, A, BeltSpinRatio and SpinPositionsPerMeter are abbreviated as S1, S2, A, BSR, and SPM respectively.

**Analysis:**

1. User selects the input feature of interest.
2. 50 LH Samples are generated within the selected input feature range. Each of these samples are combined with the current values of the remaining features.
3. Output values for the chosen samples are partially differentiated with respect to the selected feature.
4. The resulting output is shown as a line graph to the user.

**Aid for the Analysis:**

1. The line graph is provided with the range slider for the user to select the region of interest.
2. User can analyse the results for individual resolutions separately or together with the help of a drop down box.

# 6 Inferences and Conclusion

The purpose of a visual analytics framework is to perform analysis and draw out inferences. These inferences are in turn leveraged in decision making. In this chapter we discuss the inferences realized using statistical analysis and our visual analytic framework. We can use these inferences in optimal design of the production process of nonwovens.

The following are the inferences made in the thesis work:

## 6.1 CV Spread vs Sample Region Size:

During the selection of sample region size, we inferred:

1. The spread of CV values across different runs for the same parameter setting reduces with increase in the size of the sample region (as explained in 3.1).

2. For same input parameter setting, the spread decreases as it is sampled multiple times.

## 6.2 Optimal vs Non-sensitive Parameter Setting:

In our sensitivity analysis, we found that there are some parameter settings which are optimal but are very sensitive to small changes.

From figures 6.1 and 6.2, the input setting [15,36,27,0.23,5432] gives better CV values when compared to the setting [33,5,36,0.21,9927]. But if we subject these two settings for smaller changes around them, then second setting has shown more robustness for changes than the first setting.

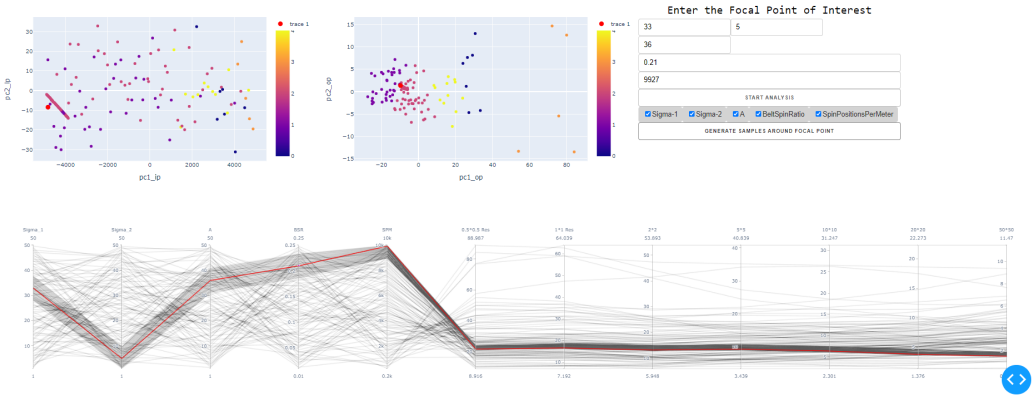



**Fig. 6.1:** Figure showing sensitivity analysis around the input setting [33,5,36,0.21,9927]

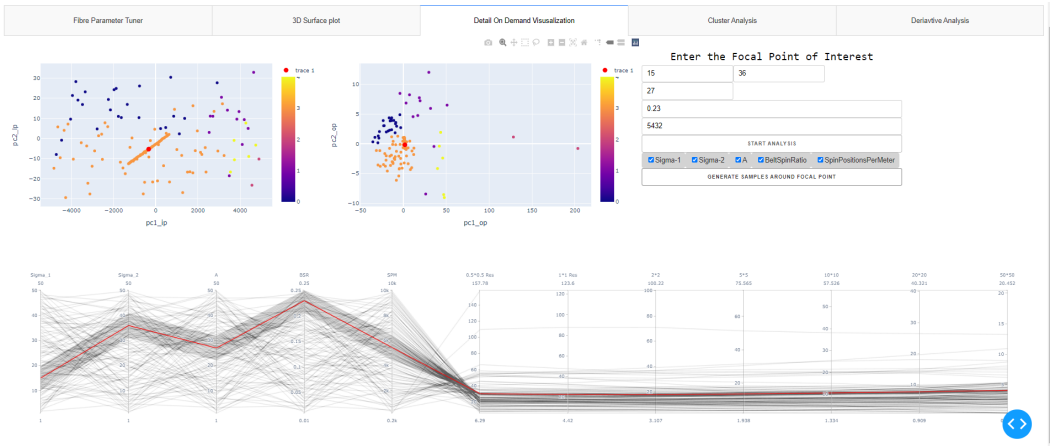



**Fig. 6.2:** Figure showing sensitivity analysis around the input setting [15,36,27,0.23,5432]

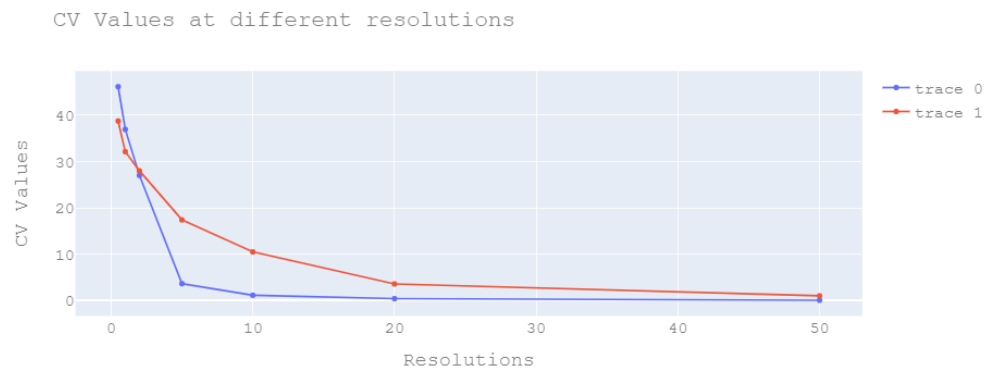
## 6.3 Desired resolutions for differentiating nonwovens

Different resolutions correspond to the same virtual material and are just different metrics to judge the quality. During our analysis, we observed that the higher resolutions do a better job in classifying nonwoven materials as "good" or "bad" compared to lower resolutions.

We considered two nonwoven materials with the following parameter settings:

1. $Mat\_1$ = [1,1,1,0.01,200].
2. $Mat\_2$ = [1,5,1,0.01,200].

When we compared these two materials , as seen from Figure 6.3 the resolutions 0.5mm, 1mm and 2mm showed that $Mat\_2$ is better than $Mat\_1$ and remaining lower resolutions 5mm, 10mm, 20mm and 50mm showed that $Mat\_1$ is better than




| Trace_Number | Sigma_1 | Sigma_2 | A | BeltSpinRatio | FibresPerMeter |
|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0.01 | 200 |
| 1 | 1 | 5 | 1 | 0.01 | 200 |

**Fig. 6.3:** Figure comparing the cv values of two nonwoven materials

$Mat\_2$. However, when we generated actual samples for both settings, as we can see from the figures 6.5 and 6.4, $Mat\_1$ is bad nonwoven with horizontal stripes when compared to $Mat\_2$.

This shows that higher resolutions are better while comparing two materials compared to the lower resolutions.

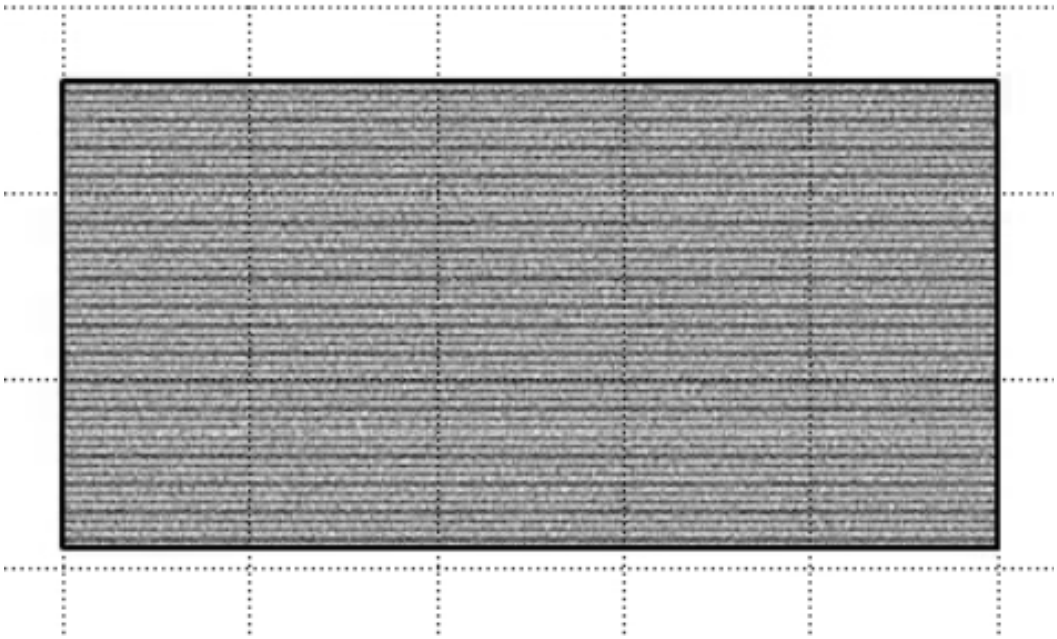


**Fig. 6.4:** Actual nonwoven material produced from the SURRO tool for $Mat\_1$

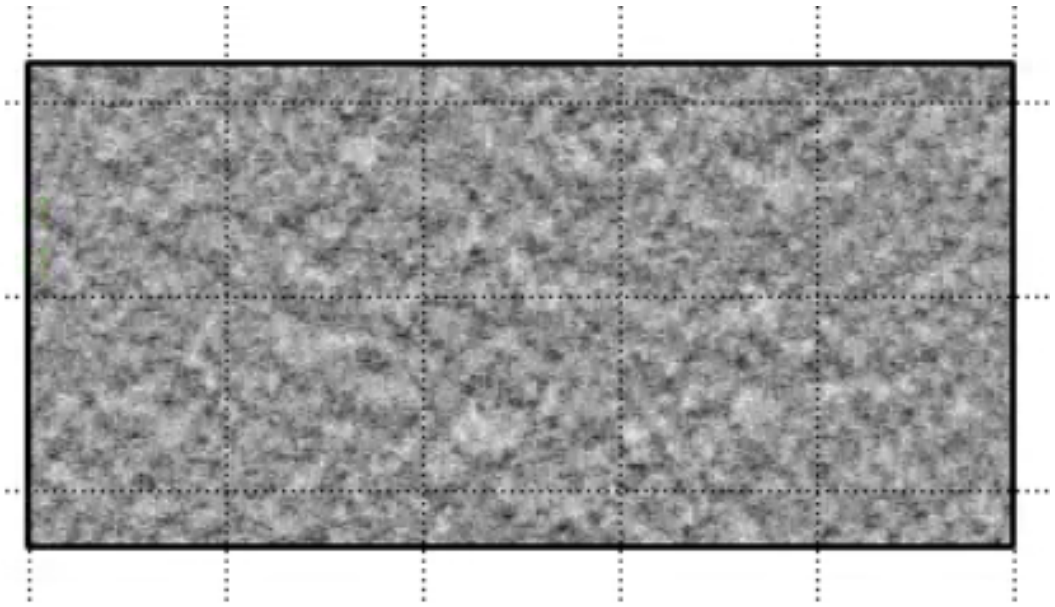


**Fig. 6.5:** Actual nonwoven material produced from the SURRO tool for $Mat\_2$

We could also realize this in our individual resolution plot 6.6 for the whole range of input feature($Sigma\_1$). There is a clear separation in the behaviour of the materials between higher and lower resolutions.

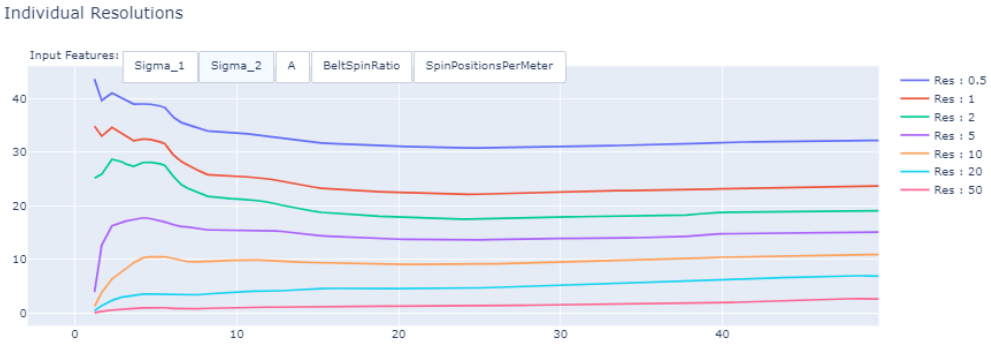



**Fig. 6.6:** Difference in the behavior of CV values at different resolutions

## 6.4 Local minimas in optimizing the process parameters:

During our analysis, we found that there are local minimas in at least one parameter direction while other parameters are fixed.

This is really useful, for example if the user is interested to find better quality where he can only change $Sigma\_1$ parameter while keeping others constant $[Sigma\_2 = 1, A = 1, BSR = 0.01, SPM = 200]$, he can find the local minima as shown in the Figure 6.7.

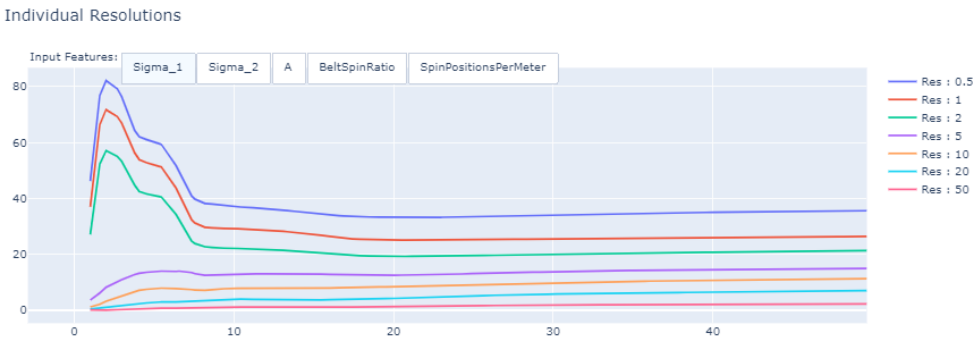



**Fig. 6.7:** Local minima in the direction of $Sigma\_1$ while other parameters are kept constant [1,1,0.01,200]

This is the similar case for local minimas in two parameter directions as shown below.

## 6.5 Outliers in Clusters

As we discussed in Section 5.4, the output clusters are mapped back to the input parameters. We then analysed the distributions of input parameters which led to the formation of each of these clusters. Although, there is a general trend in the range of input values, we observed certain outliers which are far away from the parameter distributions as shown in the Figure 6.8. These are very interesting points which can be further analysed as to why they are out of the distribution.

If we observe the distribution of the data points of the input parameter BletSpinRatio which are responsible for cluster 1. Majority of the points are up to the value 0.1 but we can see some outliers which are beyond 0.2. This are the interesting points and can be further analysed. These also emphasise on the fact that the relationship between input parameters and output values are not monotonous over the whole domain.

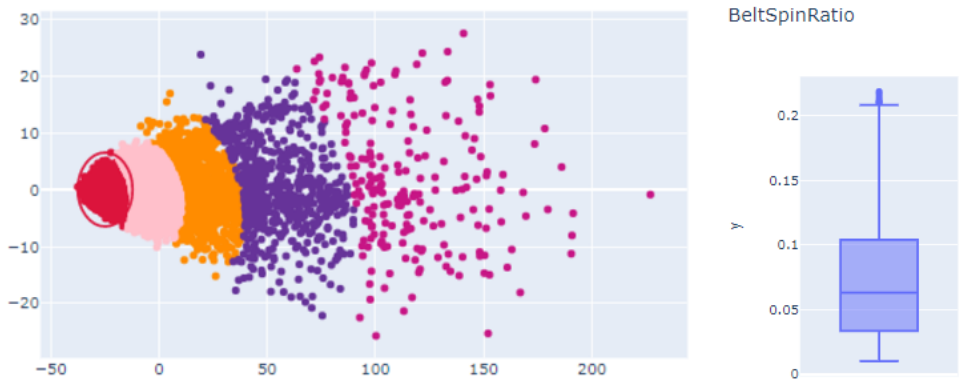



**Fig. 6.8:** Output space cluster 1 and the corresponding BeltSpinRatio distribution leading to the cluster

## 6.6 Future Work

So far, we are able to distinguish that higher resolutions are better when compared to lower resolution for differentiating the nonwoven materials. An extended analysis can be done by taking CV values for more resolutions in out dataset to determine the optimal resolution which differentiate the behaviours of the materials. We are using average CV values as a metric for comparing clusters. In future we can come up with a better metric to differentiate the clusters by analysing them. We can also incorporate the confidence of our machine learning model in the tool using the Bayesian regression model. We can use the fiber images instead of CV values and train these images over a Convolutional Neural Networks [ON15] to find a measure for determining the quality of the material.

# Bibliography


[Abo15] Ghada Ali Abou-Nassif. “Predicting the tensile and air permeability properties of woven fabrics using artificial neural network and linear regression models”. In: *Journal of Textile Science & Engineering* 5.5 (2015), p. 1 (cit. on p. 8).

[Ben+09] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. “Pearson correlation coefficient”. In: *Noise reduction in speech processing*. Springer, 2009, pp. 1–4 (cit. on p. 22).

[Ber+11] Wolfgang Berger, Harald Piringer, Peter Filzmoser, and Eduard Gröller. “Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction”. In: *Computer Graphics Forum*. Vol. 30. 3. Wiley Online Library. 2011, pp. 911–920 (cit. on p. 39).

[BT03] Christopher M Bishop and Michael E Tipping. “Bayesian regression and classification”. In: *Nato Science Series sub Series III Computer And Systems Sciences* 190 (2003), pp. 267–288 (cit. on pp. 8, 21).

[Bre01] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32 (cit. on p. 21).

[CH15] Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*. John Wiley & Sons, 2015 (cit. on p. 7).

[CD10] Christophe Croux and Catherine Dehon. “Influence functions of the Spearman and Kendall correlation measures”. In: *Statistical methods & applications* 19.4 (2010), pp. 497–515 (cit. on p. 23).

[DJR+04] Kalyan Das, Jiming Jiang, JNK Rao, et al. “Mean squared error of empirical predictor”. In: *Annals of Statistics* 32.2 (2004), pp. 818–840 (cit. on p. 26).

[De +16] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. “Mean absolute percentage error for regression models”. In: *Neurocomputing* 192 (2016), pp. 38–48 (cit. on p. 26).

[Gra06] Erik W Grafarend. *Linear and nonlinear models: fixed effects, random effects, and mixed models*. de Gruyter, 2006 (cit. on p. 16).

[HD03] Jon C Helton and Freddie Joe Davis. “Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems”. In: *Reliability Engineering & System Safety* 81.1 (2003), pp. 23–69 (cit. on p. 18).

[JM15] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260 (cit. on p. 7).

[KMW09] Axel Klar, Nicole Marheineke, and Raimund Wegener. “Hierarchy of mathematical models for production processes of technical textiles”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics* 89.12 (2009), pp. 941–961 (cit. on pp. 1, 2).

[Nag+91] Nico JD Nagelkerke et al. “A note on a general definition of the coefficient of determination”. In: *Biometrika* 78.3 (1991), pp. 691–692 (cit. on p. 26).

[ON15] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015) (cit. on p. 54).

[Ost12] Eva Ostertagová. “Modelling using polynomial regression”. In: *Procedia Engineering* 48 (2012), pp. 500–506 (cit. on p. 21).

[Ros56] Murray Rosenblatt. “A central limit theorem and a strong mixing condition”. In: *Proceedings of the National Academy of Sciences of the United States of America* 42.1 (1956), p. 43 (cit. on p. 15).

[SI] Nitin Nandkumar Sakhare and S Sagar Imambi. “Performance Analysis of Regression Based Machine Learning Techniques for Prediction of Stock Market Movement”. In: *International Journal of Recent Technology and Engineering (IJRTE), ISSN* (), pp. 2277–3878 (cit. on p. 7).

[SL12] George AF Seber and Alan J Lee. *Linear regression analysis*. Vol. 329. John Wiley & Sons, 2012 (cit. on p. 21).

[SSC19] Mine Seçkin, Ahmet Çağdaş Seçkin, and Aysun Coşkun. “Production fault simulation and forecasting from time series data with machine learning in glove textile industry”. In: *Journal of Engineered Fibers and Fabrics* 14 (2019), p. 1558925019883462 (cit. on p. 8).

[Sed+14] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. “Visual parameter space analysis: A conceptual framework”. In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (2014), pp. 2161–2170 (cit. on p. 8).

[Spe+91] Donald F Specht et al. “A general regression neural network”. In: *IEEE transactions on neural networks* 2.6 (1991), pp. 568–576 (cit. on p. 21).

[Str04] John C Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004 (cit. on p. 48).

[WMH15] Raimund Wegener, Nicole Marheineke, and Dietmar Hietel. “Virtual production of filaments and fleeces”. In: *Currents in Industrial Mathematics*. Springer, 2015, pp. 103–162 (cit. on p. 2).

[WA15] Steven A Weissman and Neal G Anderson. “Design of experiments (DoE) and process optimization. A review of recent publications”. In: *Organic Process Research & Development* 19.11 (2015), pp. 1605–1633 (cit. on p. 18).

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal component analysis”. In: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), pp. 37–52 (cit. on p. 40).

[XW08] Rui Xu and Don Wunsch. *Clustering*. Vol. 10. John Wiley & Sons, 2008 (cit. on p. 43).

[YLL12] Miin-Shen Yang, Chien-Yo Lai, and Chih-Ying Lin. “A robust EM clustering algorithm for Gaussian mixture models”. In: *Pattern Recognition* 45.11 (2012), pp. 3950–3961 (cit. on p. 44).

# List of Figures

# List of Tables