Titre: Title:	Predictive Modeling for Quality Control in Multi-Stage Manufacturing Systems Using Artificial Intelligence
Auteur: Author:	Luis Fernando Agredano Gonzalez
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Agredano Gonzalez, L. F. (2024). Predictive Modeling for Quality Control in Multi-Stage Manufacturing Systems Using Artificial Intelligence [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/61838/

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Predictive Modeling for Quality Control in Multi-Stage Manufacturing Systems Using Artificial Intelligence

LUIS FERNANDO AGREDANO GONZALEZ

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Décembre 2024

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Predictive Modeling for Quality Control in Multi-stage Manufacturing Systems Using Artificial Intelligence

présenté par **Luis Fernando AGREDANO GONZALEZ**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Hanane DAGDOUGUI, présidente
Soumaya YACOUT, membre et directrice de recherche
Ramy KHALIFA, membre

DEDICATION

I would like to dedicate this thesis to

***my father**,*

who was a great engineer but, above all, a man full of kindness and love.

He is my inspiration, and I strive every day to become like him.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Professor Soumaya Yacout, whom I greatly admire. Thank you for your guidance, patience, and for always looking out for the best interests of your students. Your support has been invaluable throughout this journey.

I would like to express my heartfelt gratitude to the jury members, Dr. Hanane Dagdougui and Dr. Ramy Khalifa, for accepting to be part of my defense. They made this moment truly enjoyable and enriching, and I deeply appreciate their valuable comments and feedback.

I also want to thank my mother for her unwavering support and constant encouragement to keep moving forward. To my father, who is an inspiration to me as an engineer, as a person, and as a father, I am profoundly grateful.

To my wife, who has been my unwavering support every step of the way, always by my side, bringing joy to each moment. Thank you for your love and encouragement, which have been a constant source of strength.

RÉSUMÉ

La prédiction des caractéristiques de qualité dans les systèmes de fabrication à plusieurs étapes contribue au maintien de spécifications de produits et à l'ajustement des procédés de fabrication pour diminuer le rebut et retravail des produits. Dans la fabrication aérospatiale, une prédiction précise de la qualité du produit à différentes étapes de la production permet des interventions opportunes, réduisant les défauts et garantissant que les produits finaux répondent à des exigences de qualité. Des modèles de prédiction permettent aux fabricants de contrôler la qualité des pièces tout au long de la production, en prenant des actions pour minimiser les pièces rejetées ou retravaillées et pour optimiser l'utilisation des ressources. L'utilisation des techniques d'intelligence artificiel (IA) dans la prédiction de la qualité des pièces produites aide à améliorer la qualité des produits et la fiabilité des procédés par des actions qui améliorent la performance globale.

Les modèles d'apprentissage automatique ont été largement étudiés et appliqués dans le contexte de la prédiction de la qualité dans les systèmes de fabrication à plusieurs étapes. Divers algorithmes ont démontré leur efficacité selon la revue de littérature. Ces résultats soulignent le potentiel de l'utilisation des techniques d'apprentissage automatique pour améliorer la précision des prédictions et l'efficacité opérationnelle dans les systèmes de fabrication à plusieurs étapes.

Dans cette étude, nous avons utilisé les algorithmes d'apprentissage automatique les plus efficaces selon la revue de littérature, notamment la régression par moindres carrés partiels (PLS), la régression par composantes principales (PCR), les machines à vecteurs de support (SVM) avec des fonctions de base linéaires et radiales, la forêt aléatoire et les k-plus proches voisins (KNN). Bien que l'apprentissage profond ait montré un grand potentiel dans la prédiction de la qualité, il nécessite une quantité substantielle de données pour s'entraîner efficacement. Compte tenu de nos contraintes de données, nous avons décidé de ne pas inclure d'architectures d'apprentissage profond dans cette étude.

Nous appliquons ces techniques à un système de fabrication à plusieurs étapes produisant des pièces de moteur d'avion. Le processus implique des inspections à différentes étapes intermédiaires utilisant des machines de mesure par coordonnées (CMM). Contrairement à de nombreuses études dans la littérature, nos prédictions sont basées uniquement sur des données d'inspection en cours de processus de fabrication, sans recourir à des paramètres de procédés ou des lectures de capteurs. Notre approche est basée sur la connaissance des données historiques sur les mesures des

caractéristiques de qualité (QC) à chaque étape de fabrication. De plus, notre approche permet non seulement de prédire les caractéristiques de qualité (QC) pour l'étape consécutive suivante, mais aussi pour des étapes plus avancées dans le processus de fabrication, y compris l'inspection finale, en utilisant des données des étapes antérieures. Cette capacité permet un contrôle qualité proactif et rapide, ainsi qu’une optimisation de l'ensemble du flux de production.

Les résultats montrent qu'une portion significative des caractéristiques de qualité peut être prédite avec une grande précision en utilisant les techniques susmentionnées. Nos modèles ont démontré la faisabilité de prédire non seulement l'étape consécutive suivante, mais aussi les étapes futures dans le système de fabrication.

Enfin, nous discuterons de certaines limitations de l'étude et des recommandations pour les travaux futurs.

ABSTRACT

The prediction of quality characteristics in multi-stage manufacturing systems contributes to maintaining product specifications and adjusting manufacturing processes to reduce scrap and rework. In aerospace manufacturing, predicting product quality at various stages of production allows for early interventions, reducing defects and ensuring that final products meet quality requirements. Predictive models enable manufacturers to control the quality of parts throughout production, taking actions to minimize rejected or reworked parts and optimize resource use. The use of artificial intelligence (AI) techniques in predicting the quality of produced parts helps improve product quality and process reliability through actions that enhance overall performance.

Machine learning models have been studied and applied in the context of quality prediction in multi-stage manufacturing systems. Various algorithms have demonstrated to perform well according to the literature review. These results highlight the potential of using machine learning techniques to improve the accuracy of predictions and operational efficiency in multi-stage manufacturing systems.

In this study, we used the most used machine learning algorithms according to the literature review, including partial least squares regression (PLS), principal component regression (PCR), support vector machines (SVM) with linear and radial basis functions, random forest, and k-nearest neighbors (KNN). Although deep learning has shown potential in quality prediction, it requires a substantial amount of data to train. Given our data constraints, we decided not to include deep learning architectures in this study.

We apply these techniques to a multi-stage manufacturing system producing aircraft engine parts. The process involves inspections at various intermediate stages using coordinate measuring machines (CMM). Unlike many studies in literature, our predictions are based solely on in-process inspection data, without relying on process parameters or sensor readings. Our approach is based on knowledge of historical data on quality characteristic (QC) measurements at each manufacturing stage. Additionally, our approach not only predicts the quality characteristics (QC) for the next consecutive stage but also for more advanced stages in the manufacturing process, including the final inspection, using data from earlier stages. This capability allows for proactive and rapid quality control, as well as optimization of the entire production flow.

The results show that quality characteristics can be predicted using the chosen techniques. Our models have demonstrated the feasibility of predicting not only the next consecutive stage but also future stages in the manufacturing system.

Finally, we will discuss some limitations of the study and recommendations for future work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LISTE OF SYMBOLS AND ABBREVIATIONS

MMS	Multi-stage Manufacturing System
QC	Quality Characteristic
DL	Deep Learning
ML	Machine Learning
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MSE	Mean Square Error
PLS	Partial Least Squares
PCR	Principal Component Regression
SMVLinear	Support Vector Machine with Linear Kernel
SMVRBF	Support Vector Machine with Radial Basis Function
KNN	K-Nearest Neighbors
RF	Random Forest
PCA	Principal Component Analysis
GA	Genetic Algorithm
CMM	Coordinate Measure Machine
MLR	Multiple Linear Regression
%EV	Percentage of Error compared to Data Variation
%ET	Percentage of Error compared to Tolerance
EDA	Exploratory Data Analysis
R^2	Coefficient of Determination

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Basic Definitions

Quality control is defined as a system that maintains a desired level of quality in a product or service in fabrication through feedback on product/service characteristics and implementation of remedial actions when deviations from specification occur. This involves monitoring and controlling the production processes to ensure that the output meets specified quality standards [1].

In a Multi- Stage Manufacturing Systems (MMS), quality prediction involves the use of statistical and machine learning models to predict future quality outcomes based on historical and real-time data. It allows for proactive actions to prevent out-of- specifications product , and to improve production efficiency by anticipating potential quality problems before they arise [1].

A quality characteristic is an inherent attribute or feature of a product that affects its ability to satisfy stated or implied needs. These characteristics can be structural, sensory, time-oriented, or ethical, and they are used to measure the quality of a product or service [1].

Tolerances or specifications refers to the allowable limits of variation in a physical dimension or measured value within which a product or process is considered acceptable. Specifications are precise statements that formalize the requirements of the customer and are used to ensure that products meet desired standards [1].

1.2 Evolution of Quality Control

Quality control has been a present aspect of industrial production since ancient times. The ancient Egyptians applied rigorous standards in their construction projects, while the Greeks and Romans implemented high standards in their architecture and craftsmanship. Throughout history, the approach to quality control has evolved, particularly during the Middle Ages and the Industrial Revolution, transitioning from individual artisans to systematic approaches involving foremen and specialized quality control departments [1].

1.3 Development of Quality Control Methods

In the early 20th century, formalized quality control methods such as the foreman quality control period and later the statistical quality control period were introduced. These methods were essential in ensuring product consistency and meeting the increasing demand for high-quality products.

Pioneers like Walter A. Shewhart, who developed the concept of statistical process control in the 1920s, and W. Edwards Deming, who later expanded on these ideas and applied them to improve quality management across industries, brought a scientific approach to quality management by utilizing statistical tools to monitor and control production processes.

With the rise of Industry 4.0, quality control has further advanced to incorporate technologies and data-driven approaches. The integration of sensors and automated monitoring systems has enhanced the capability to maintain quality standards throughout the manufacturing process[1].

1.4 Current Practices in Quality Control

Quality control today encompasses various methods and tools to ensure products meet specified standards. These methods include off-line quality control, online statistical process control, and acceptance sampling plans. Off-line quality control focuses on designing products and processes to prevent defects before production begins, for example by using Design of Experiments techniques. Statistical process control uses statistical methods to monitor and control production processes in real-time, ensuring that they operate within specified limits. Acceptance sampling plans are employed to decide whether to accept or reject a batch of products based on a sample[1].

1.5 Importance of Quality Prediction

In addition to quality control, quality prediction is important for anticipating potential quality problems before they occur. Predictive models use historical and real-time data to predict future quality outcomes, allowing for proactive actions to prevent the production of defects and to enhance production efficiency. Quality prediction tools can reduce costs associated with rework, scrap, and downtime, and improve the performance of manufacturing processes[2].

1.6 Advantages of Integrating Quality Prediction Tools

Integrating quality prediction tools into a quality management system offers several advantages. These tools enable manufacturers to:

- Anticipate and Mitigate Risks: By predicting potential quality problems, manufacturers can take preventive actions to reduce the likelihood of defects and production delays.
- Optimize Resource Utilization: Predictive models can help optimize the use of materials, labor, and equipment, leading to more efficient production processes.

- Enhance Decision-Making: Data-driven insights from predictive models support better decision-making, allowing managers to make informed choices about process improvements and resource allocation.
- Improve Product Quality and Consistency: By identifying and addressing potential issues early, quality prediction tools help ensure that products meet or exceed quality standards.
- Increase Customer Satisfaction: High-quality products lead to increased customer satisfaction and loyalty, which are important for maintaining a competitive edge in the market.

In conclusion, while traditional quality control methods focus on monitoring, detecting and correcting defects during the manufacturing process, quality prediction tools provide a proactive approach to managing quality. By integrating these tools into a comprehensive quality management system, manufacturers can enhance their ability to maintain quality standards, optimize production processes, and improve overall performance and customer satisfaction.

1.7 Research Motivation

The multi-stage manufacturing systems (MMS) contain several stages to produce a finished product with the required level of quality. They comprise various processes such as machining, stamping, assembly, turning and milling. [3]

Two characteristics of multi-stage manufacturing processes are:

1. The output quality characteristics at a particular stage are the inputs of the next stage. This means that the output of a stage depends not only on process variations at this stage but also on the variations propagated from the upstream stages. Thus, the final product is an accumulation of variations of all stages.
2. The product's final quality of the multi-stages system depends on the cumulative performance of the individual processes at each stage.

This complexity poses challenges specifically for automatic quality control. One aspect of quality control in various multi-stage manufacturing is the dimensional quality assurance of the product [4]. This difficulty arises from the uncertainties due to random variations in the production processes, which lead to random variations in the quality characteristics of the parts at the different stages of the manufacturing system. Hence, quality prediction in multi-stage manufacturing processes has been a research focus. The increasing availability of data and the adoption of artificial intelligence (AI) lead to the development and implementation of advanced prediction tools [4]. Several research have been accomplished to develop such quality prediction tools by using artificial intelligence. A comprehensive sample of this research is given in chapter 2 of this thesis.

1.8 Research Objective

According to the previous subsections, integrating quality prediction tools into quality management systems can provide benefits, such as anticipating potential problems and optimizing resource utilization. Building on this foundation, the primary objective of this research is to present, in Chapter 2, a comprehensive literature review of the research published on quality prediction within an MMS, to propose a methodology for predicting this using AI techniques, and finally to present an industrial application in Chapter 3. Conclusions and recommendations are presented in chapter 4. A general rule across this thesis, the predicted values of specific product's characteristics at a given stage $q+1$, is based on the corresponding characteristics inspected at the immediately preceding stage q. Furthermore, this research explores the possibility to predict the values of quality characteristics at the final stage Q based on the corresponding characteristics inspected at q. Where $q= 1, 2, 3, \ldots, Q$.

CHAPTER 2 LITERATURE REVIEW

2.1 Multi-Stage Manufacturing Systems (MMS)

Multistage manufacturing systems (MMS) are industrial processes involving multiple sequential stages to produce complex products, such as semiconductor components, automotive parts, or turbine blades. Each stage introduces new features or modifies existing ones to meet specific quality standards. The complexity of MMS arises from the interdependence of stages, where deviations and errors from upstream processes propagate and amplify in downstream operations. This accumulated effect creates nonlinear relationships between variables at different stages, significantly impacting final product quality[5].

2.1.1 Characteristics and Challenges of MMS

The defining feature of MMS is its interconnected nature, where variations in process parameters or quality measures in one stage affect subsequent stages. For example, in turbine blade manufacturing, small errors such as fixture-induced or thermal deviations accumulate across 7–12 stages, complicating efforts to maintain geometric precision and aerodynamic performance [6]. Unlike single-stage systems, the quality in MMS depends not only on current process variable, increasing the difficulty of effective monitoring and control.

Predicting quality within the same production stage is a well-studied problem with relatively low complexity, as it often involves static conditions and linear relationships. In contrast, predicting quality at future stages adds layers of complexity due to dynamic dependencies, nonlinearities, and stochastic variations inherent to MMS. For example, accumulated deviations or non-obvious interactions between parameters may lead to defects that are challenging to predict without advanced modeling techniques [7].

2.2 Machine learning for quality prediction in Multi-stage Manufacturing Systems

Data-driven methods for quality prediction in multistage manufacturing systems are divided into two main categories: Deep and conventional Machine Learning (ML) methods. The performance of conventional Machine Learning methods depends on the feature engineering, and expert knowledge may be needed. On the other hand, Deep Learning (DL) methods can reach the

prediction without feature engineering [8]. However, the training process of Deep Learning methods requires a large amount of data, while conventional Machine Learning methods are proven to be more stable when the amount of data is limited [9].

Machine learning aims to approximate an unknown function to model the relationships between input and output variables. DL is a branch of machine learning that contains multiple layers to represent or map the relationships between the inputs and the outputs. A layer comprises neurons that perform mathematical operations, and with a deep architecture, the model can handle many variables and approximate patterns in the data. DL models generate features as part of the learning process, which is an advantage over the conventional ML methods that need to generate features as an additional step before the learning process [10].

ML and DL models have been used in the manufacturing industry to address many challenges thanks to their ability to manage large-dimensional multivariate data and detect implicit relationships within large data sets in dynamic environments [11]. Regarding quality prediction in multi-stage manufacturing systems, ML algorithms for classification and regression have been pursued [5]. The regression algorithms are of particular interest to our study.

2.2.1 Regression Algorithms for Quality Prediction in muti-stage Manufacturing Processes:

Regression is a supervised learning method. It is used when the type of response variable to be predicted is continuous [5]. In the context of quality prediction, the goal is to predict the value of one or more quality characteristics.

Regarding the evaluation of the prediction's accuracy of regression algorithms, the most widely used performance indicators are shown in Table 2.1 [12], where:

- $\hat{y}^{p}_{i}$: the predicted value for the *i-th* observation and the *p-th* quality characteristic .
- y^{p}_{i} : the real value of the *i-th* observation and the *p-th* quality characteristic.

Where "*i*" represents the number of instances or observations in a given dataset. $i = 1, 2, \ldots, n$, and "*p*" is the quality characteristic to be predicted. $p = 1, 2, \ldots P$.

Table 2.1 Performance Indicators for Regression Algorithms

Indicator Name	**Equations**	
	Single output	**Multi-output**
Mean Error (*ME*)	$ME = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$ (2.1)	$ME = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^p{}_i - y^p{}_i)$ (2.6)
Mean Square Error (*MSE*)	$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ (2.2)	$MSE = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^p{}_i - y^p{}_i)^2$ (2.7)
Root Mean Square Error (*RMSE*)	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$ (2.3)	$RMSE = \frac{1}{P}\sum_{p=1}^{P}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}^p{}_i - y^p{}_i)^2}$ (2.8)
Mean Absolut Error (*MAE*)	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$ (2.4)	$MAE = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}^p{}_i - y^p{}_i\rvert$ (2.9)
Mean Absolut Percentage Error (*MAPE*)	$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{\lvert\hat{y}_i - y_i\rvert}{y_i}$ (2.5)	$MAPE = \frac{1}{P}\sum_{p=1}^{P}\frac{100}{n}\sum_{i=1}^{n}\frac{\lvert\hat{y}^p{}_i - y^p{}_i\rvert}{y^p{}_i}$ (2.10)

Indicators shown in Table 2.1 represent the prediction error of the model. Therefore, the lower the value, the better the regression model is.

2.3 Previous Work

In this subsection, six articles concerning the prediction of quality in multi-stage manufacturing systems are discussed. They all use ML or DL methods and face different challenges.

In[13], two quality characteristics are predicted in a two-stage continuous extrusion process. A pipe's internal and external diameter are these two quality characteristics of interest. The objective is to predict the final quality characteristics by taking as input the process variables from the two continuous stages at the same time as shown in figure 2.1. In both stages, there are process variables

such as temperatures, pressures, speed of rotation of the screw, and tension of the tensile mechanism. Those process variables are measured periodically by using sensors. In total, there are 15 process variables corresponding to the input data for the regression models. The authors tested eight algorithms to predict the internal and external diameter: Nearest K-Neighbors and two distance-weighted variants, multiple linear regression, support vector machine using linear kernel, polynomial, and radial basis, and finally, Neural Networks. The evaluation criteria chosen are *RMSE* and *MAE*. In conclusion, the best-performing models were the distance-weighted K-Nearest Neighbor and Support Vector Machine with the radial base kernel function. The authors proposed, as work to be done, that the selection of relevant process variables could improve the model's performance. Notably the final quality characteristics are predicted using process variables from both stages of a two-continuous stage extrusion process, providing a direct assessment of the final product. However, this approach limits opportunities for early intervention since predictions occur after both stages are complete, making it reactive rather than proactive.

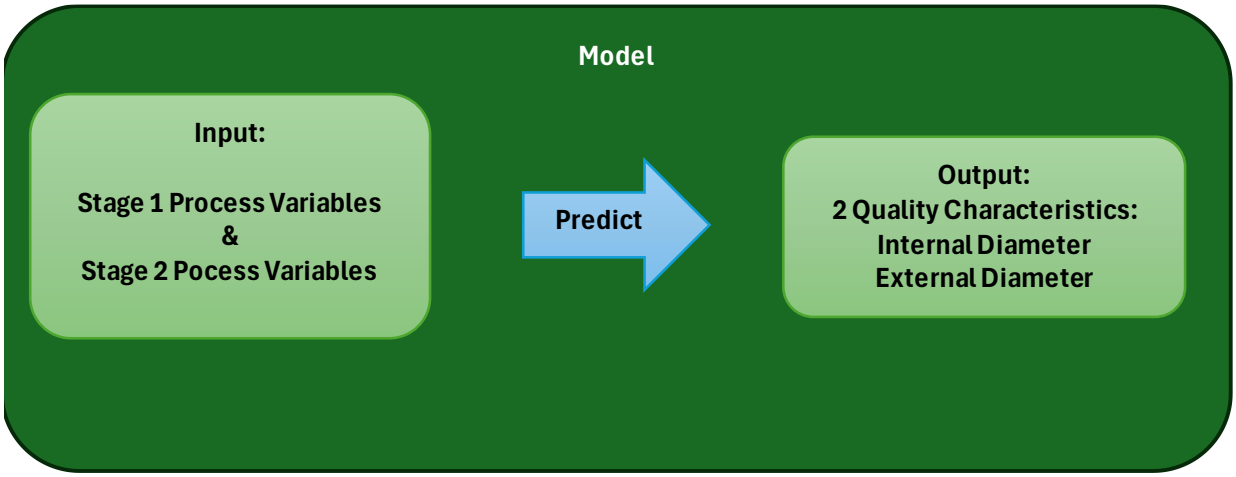

Figure 2.1 : Diagram of the model

In [14], the authors presented a two-stage assembly process. The first stage has 106 quality characteristics. The second stage has 107 quality characteristics. The objective was to predict the 107 quality characteristics of the second stage by taking as input the 106 quality Characteristics from the first stage.

The data set used includes 1000 observations. The most critical challenges are the presence of multicollinearity, the large number of predictor variables compared to the number of observations, and the large number of quality characteristics to be predicted. Multicollinearity occurs when two or more independent variables are correlated. In a regression model, this makes it difficult to determine the individual effect of each independent variable on the dependent variable since their contributions overlap. To overcome these problems, the authors proposed partial least squares regression (PLS), a technique where a set of orthogonal factors, called latent variables, are extracted from predictive variables. These latent variables have the best predictive power for response variables. Since nonlinear relationships are not rare in real industrial situations. A partial least squares nonlinear regression model is proposed to tackle the possible nonlinearity. PLS is a transformation technique that represents data in a higher-dimensional space using a specific kernel function (KPLS).In this work, the approach presented in [15] where a quadratic polynomic kernel is applied, is taken as reference to tackle the non-linearity problem challenge.

Both methods, PLS and KPLS, were compared, showing similar performance when the number of latent variables used as predictors goes from 1 to 4. From the fifth latent variable on, the KPLS performs better. *RMSE* was chosen as the evaluation metric.

In [16], two coal's quality characteristics were predicted and optimized in a multi-stage manufacturing system. This process involves uncontrollable and controllable process variables, and coal’s quality characteristics measured at each stage.

First, a regression model is fitted at each stage of the process. For the first stage of the process, the input variables correspond to the uncontrollable and controllable process variables while the two quality characteristics of the coal are the output variables to be predicted within the same stage. At the second stage of the process, the coal's quality characteristics from the previous stage also become input variables along with the process variables of the current stage. Figure 2.2 describes the modeling process. The algorithm chosen to fit the regression models was the Support Vector Machine with the radial base function because of its ability to adjust to nonlinear problems and its performance with small data sets. The algorithm's parameters were optimized using the "Grid Search" method. As for data pre-processing, outliers were identified using mustache box diagrams and then they were deleted from the model. Finally, the data was scaled using the standardization method.

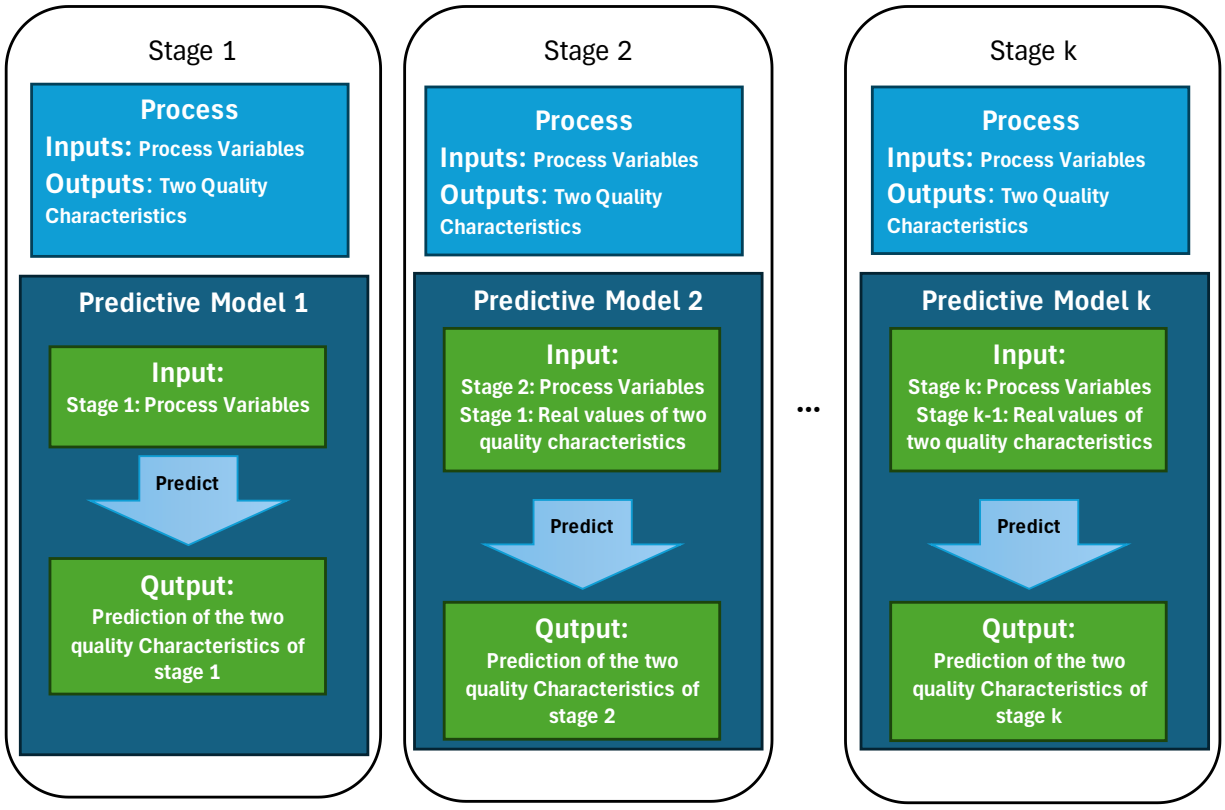

Figure 2.2: Modeling Process for Coal´s Quality Characteristics.

Secondly, the coal's quality characteristics were optimized using the genetic algorithm (GA). The optimization started at the last stage of the process, determining the target values of the coal's quality characteristics. The GA requires an evaluation function, which in this case, is the output of the regression model corresponding to the last stage. The coal's quality characteristics values of the previous stage are input variables in the current stage. Then, the GA returns the input variable´s values that will allow the quality characteristics of the coal in the last stage to reach the target value. The input values provided by the GA corresponding to the coal's quality characteristics will subsequently become the objective values for optimizing the previous stage, and so on, until the first stage of the process.

In [6], a multistage manufacturing system aimed at producing turbine blades is presented. There are up to 10 stages in the process, but the authors selected only three that are considered critical by the experts. Stage 1: Blade root milling; Stage 2: Comprehensive Accurate milling, and Stage 3: final inspection. The availability of controllable or uncontrollable process variables is absent.

However, the product's characteristics measurements are obtained from a CMM (Coordinate Measuring Machine) at the intermediate stages of the process

The objective was to predict the value of 10 quality characteristics at the final inspection. The input data for the model is the CMM measurements gathered at the key intermediate stages. Three models were developed: two separately predicted the ten quality characteristics at the final inspection from each key intermediate stage. The third model incorporated the measurements of both key intermediate stages as input data to predict the ten quality characteristics at the final inspection.

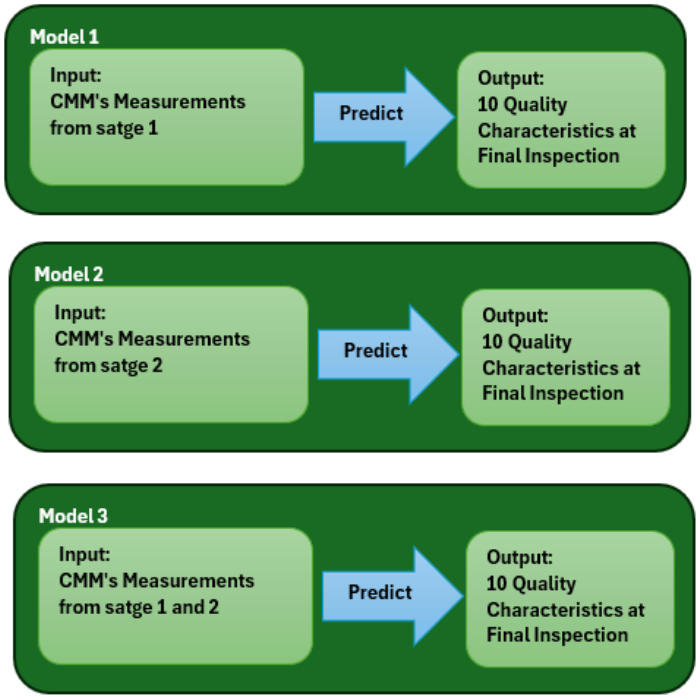

Figure 2.3: Diagram of Prediction Models

A sequential multistep deep learning architecture approach was developed for quality prediction. It employs an attentive transformer which is a type of neural network designed to select the most pertinent features. This architecture is designed as a modular system where each step or phase builds upon the output of the previous one, allowing specialized handling of different tasks within the sequence. [17].

The evaluation criteria were *RMSE* and *MAE*. Many algorithms were tested, but the proposed method showed the best performance. However, PCA (Principal Component Analysis) combined

with SVR (Support Vector Regression), and DBM (Deep Boltzmann Machine) showed good performance as well. According to the authors, the proposed method can better select effective feature combinations to predict the quality characteristics.

In [8], the authors introduced a novel approach termed the Path Enhanced Bidirectional Graph Attention Network, tailored for addressing the intricate challenges of quality prediction within multi-stage manufacturing systems. The system comprises two stages: the first stage includes three parallel-operating machines followed by a combiner which is a step that combines the flows from the three parallel machines into a single flow. The second stage consists of two machines. Each machine in the first stage presents 14 distinct process variables, and the combiner five process variables, while each machine in the second stage has nine process variables. These process variables serve as inputs for predicting 15 quality characteristics at each stage, with only the relevant stage's process variables used for predictions at that stage, except in the second stage where process variables from both stages are incorporated as input as shown in figure 2.2.

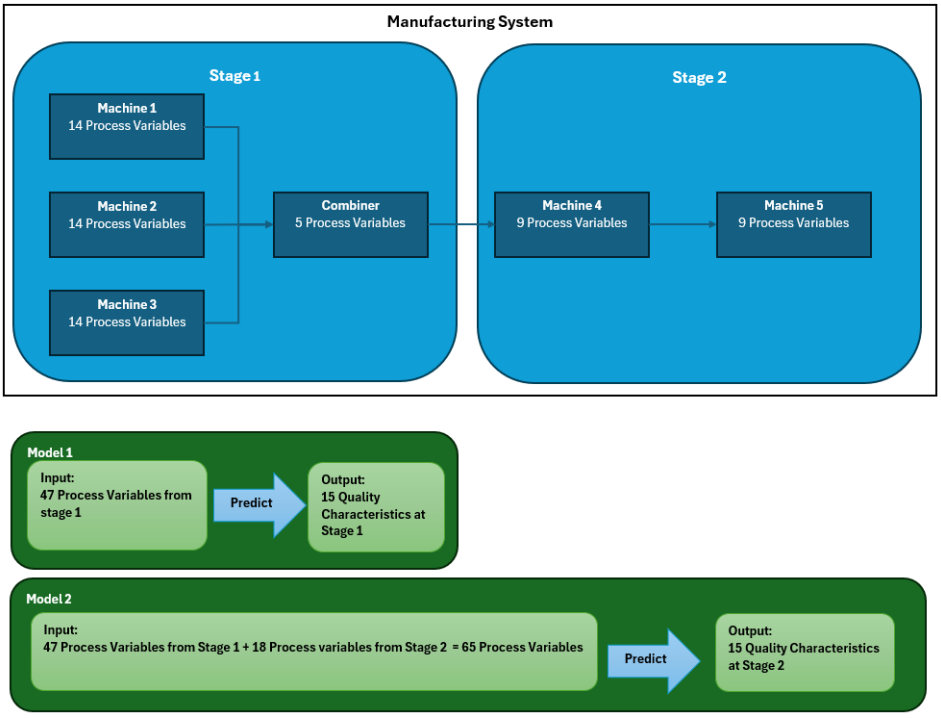

Figure 2.4: Multi-Stage continuous-flow Manufacturing System.

To mitigate issues with noisy labels, a mask loss function was implemented in the pre-processing phase to selectively ignore certain data during loss calculation. The proposed solution features three main modules: Firstly, the Machine Embedding Pool (MEP) and a path encoder graphically represent and encode the relationships among the machines. This graphical representation captures the dependencies between machines and their process variables. Secondly, a Bidirectional Graph Attention Network extracts relevant process variables and contextual information from the dependency graph. This network employs an attention mechanism to enhance interaction understanding between machines, thereby improving predictions of quality characteristics. Lastly, a target-specific attention decoder predicts quality characteristics at different stages using the contextual information and process variables processed by earlier modules.

The study compared several machine learning and deep learning methods, with the proposed approach outperforming others in terms of *RMSE* and *MSE*. Comparable performances were observed in methods like Convolutional Neural Networks, Feedforward Networks, and Random Forests.

In [9], the multistage manufacturing process and the objectives are the same as those described in the previous article [8] (refer to figure 2.2), but a different approach is taken. The authors introduced a novel method called Contrastive Decoder Generator (CDG) for Few-Shot Learning in Product Quality Prediction. Few-Shot Learning is a technique in machine and deep learning that optimizes training when data is limited, which is particularly pertinent given the presence of noisy labels in the dataset.

The proposed CDG approach consists of three main modules: Firstly, a Machine Feature Encoder (MFE) encodes relevant process variables from the machines, capturing their interactions. The MFE is crucial for representation learning, aiming to develop a descriptive representation of the input data that highlights pertinent process variables while filtering out noise. Secondly, a Contrastive Feature Generator (CFG) is utilized. Contrastive learning methods are often employed as oversampling techniques to enhance the minority class representation in imbalanced datasets. Here, the CFG is used to generate knowledge about the stages and tasks contained in the hidden output representations of the MFE. Lastly, an Instance-Specific Decoder Generator (IDG) is introduced. Within generative approaches, the IDG generates specific instances conditioned on certain inputs or characteristics, allowing for customization or adaptation to specific requirements.

In this context, the IDG is designed to achieve quality predictions for new tasks in real-time without the need for additional training.

Several machine learning and deep learning techniques were evaluated, with the proposed CDG method outperforming others in terms of *RMSE* and *MSE*. Methods that showed performance close to the CDG included Convolutional Neural Networks, Feedforward Networks, and Random Forests.

Table 2.2 provides a comparison of the differences and similarities of the scenarios between the papers discussed in this section and our case study. S1, S2, and S3 represent stage 1, 2, and 3 respectively.

Table 2.2 : Comparison of the scenarios between the literature review and our case of study

Aspect/Articles	**[13]**	**[14]**	**[16]**	**[6]**	**[8], [9]**	**In this thesis**
Number of Stages	Two Continuous stages	Two Stages	Three Stages	Three Stages	Two Stages	Three Stages
Amount of available data for modeling	260 records	1000 records	40 records	Not specified	14,088 records	266 records
Availability of process variables	Yes	No	Yes	No	Yes	No
Number of independent variables	15 Process Variables from stage 1 and 2	106 of stage 1	S1:4, S2:13, S3:7	S1:80 S2:77 S3:118	S1:47 S2:18	S1:299 S2: 114 S3: 110
Number of dependent variables	2 QC	107 of stage 2	2 at each stage	10 at each stage	15 at each stage	56 at each stage

Table 2.2: Comparison of the scenarios between the literature review and our case of study (continuation and end)

Aspect/Articles	**[13]**	**[14]**	**[16]**	**[6]**	**[8], [9]**	**In this thesis**
Scope of prediction	Immediate next stage (Final inspection)	Immediate next stage	Within the stage	Final Inspection Prediction	Within the stage	Both: Immediate next stage and further stages including final inspection.

In summary, the articles discussed in this subsection have utilized a variety of techniques to address specific challenges encountered in multi-stage manufacturing systems. Each study focuses on predicting quality characteristics either within the same stage of the manufacturing process or in the immediately subsequent stage. The aim of this thesis is not only to predict the quality characteristics for the next stage but also for subsequent stages as well. This allows for a more comprehensive and forward-looking view of potential outcomes throughout the entire manufacturing process.

For our specific problem, the main challenges are high dimensionality, a limited amount of data and multiple variables to be predicted. Notably, in Stage 1, there are more independent variables than observations in the dataset, a situation that is not observed in any of the articles from the literature review. To overcome these problems, techniques such as Principal Component Analysis, Partial Least Square Regression, Principal Component Regression, Support Vector Machines, Random Forest, and K-nearest neighbors will be used as presented in the next section.

2.4 Multiple linear regression (MLR)

MLR is a statistical technique used to model the relationship between a dependent variable y and multiple independent variables represented by the matrix X.

The multiple linear regression model can be expressed in matrix form as follows:

$$y = X\beta + \epsilon \tag{2.11}$$

where:

y is an n ×1 vector of observations on the dependent variable. X is an n p matrix of observations on the independent variables, where n is the number of observations and p is the number of independent variables. β is a p ×1vector of unknown parameters (coefficients or weights) to be estimated. And ϵ is an n×1vector of errors or residuals, which represent the difference between the observed and predicted values of y.

The coefficients β are estimated using the method of ordinary least squares (OLS), which minimizes the sum of the squared differences between the observed values and the values predicted by the linear model. Mathematically, the OLS estimates of β are obtained by solving the following equation:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.12}$$

Here, X^T denotes the transpose of the matrix X, and $(X^TX)^{-1}$ is the inverse of the matrix X^TX.

Once the coefficients β are estimated, the predicted values of the dependent variable $\hat{y}$ can be computed as:

$$\hat{y} = X\hat{\beta} \tag{2.13}$$

2.5 Principal Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique. It aims to transform high-dimensional data into a lower-dimensional representation while retaining the most important information and minimizing information loss.

Let's consider a dataset of predictor variables with *n* data points, each residing in a *p*-dimensional space, represented as a matrix $\boldsymbol{X}$ with dimensions $n \times p$. Each row corresponds to a data point. The goal of PCA is to find a new set of orthogonal axes, along which the variance of the data is maximized. Those axes are called principal components. The first principal component is the

direction along which the data has the highest variance. It can be computed by finding the eigenvector $\boldsymbol{v_1}$ of the covariance matrix $\boldsymbol{C}$ corresponding to the largest eigenvalue λ_1.

$$\boldsymbol{C} = \frac{1}{n}\boldsymbol{X}^T\boldsymbol{X} \tag{2.14}$$

$$\boldsymbol{C}\boldsymbol{v}_1 = \lambda_1\boldsymbol{v_1} \tag{2.15}$$

Subsequent principal components are found similarly, but with the constraint that they are orthogonal to the previously found components.

The original data can be projected onto the new principal component axes to obtain the lower-dimensional representation. For *k* principal components, the projection matrix is formed by stacking the eigenvectors corresponding to the top *k* eigenvalues. $\boldsymbol{V}_k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k]$

The lower-dimensional representation $\boldsymbol{W}$ is given by:

$$\boldsymbol{W} = \boldsymbol{X}\boldsymbol{V_k} \tag{2.16}$$

2.6 Partial Least Square Regression

Partial Least Square Regression is useful when we need to predict a set of dependent variables *Y* or a single independent variable *y* from a large set of independent variables *X*. The problem often is that X^tX is singular either because the number of variables in *X* exceeds the number of observations or objects, or because of collinearities [18].

The goal of PLS is to extract latent variables (also known as components or factors) that capture the maximum covariance between *X* and y. The method iteratively constructs these latent variables, which are linear combinations of the original predictors, to explain both the variance in *X* and the covariance between *X* and y.

The PLS model can be summarized in the following steps:

1. Decompose *X* and y:

$$X = TL^T + E \tag{2.16}$$

$$y = UM^T + F \tag{2.17}$$

where:

- T is an $n \times k$ matrix of scores (latent variables) for X.
- L is a $p \times k$ matrix of loadings for X.
- E is an $n \times p$ matrix of residuals for X.
- U is an $n \times k$ matrix of scores for y.
- M is a $1 \times k$ vector of loadings for y.
- F is an $n \times 1$ vector of residuals for y.
- k is the number of components.

2. Maximize Covariance:

The latent variables T and U are chosen to maximize the covariance between X and y: Cov (T, y)

3. Construct the Regression Model:

Once the latent variables are obtained, the PLS regression model is constructed:

$$\hat{y} = X\hat{\beta} + \epsilon \tag{2.18}$$

where:

β is a $p \times 1$ vector of regression coefficients. ϵ is an $n \times 1$ vector of residuals. The coefficients β are estimated by projecting X onto the latent variables and then relating these projections to y. An important question is determining the number of latent variables that offer the best prediction for new observations. This is often addressed using cross-validation methods like bootstrapping [19].

2.7 Support Vector Machines for Regression (SVM)

Support Vector Regression (SVR) is a type of regression algorithm that leverages the concepts of Support Vector Machines (SVM) to predict continuous outcomes.

Let´s consider y is an $n \times 1$ vector of observations on the dependent variable. X is an $n \times p$ matrix of observations on the independent variables, where n is the number of observations and p is the number of independent variables. Each row of X represents a single sample, which consists of p independent variable values associated with a corresponding observation of the dependent variable in y. Thus, the i-th sample can be described as: (x_{i1}, x_{i2}, …, x_{ip}, y_i).

SVR aims to find a function that approximates the mapping of the independent variables X to a high-dimensional space where the relationships among the data points X and y can be modeled more effectively than in the original input space[13].

$$\hat{y} = W^T \emptyset(X) + b \tag{2.19}$$

Where W is the weight vector, $\emptyset(X)$ is the function to map X into a higher dimensional space and b is a bias term.

W and b are obtained by solving the following optimization problem [20]:

$$minimize\ \frac{1}{2}\|W\|^2 + Cost \sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{2.20}$$

$$subject\ to \begin{cases} y_i - W \cdot \phi(X_i) - b_i \leq \varepsilon + \xi_i,\ \xi_i \geq 0 \\ W \cdot \phi(X_i) + b_i - y_i \leq \varepsilon + \xi_i^*,\ \xi_i^* \leq 0 \\ i = 1, \dots, n \end{cases} \tag{2.21}$$

Where Cost is a constant that takes the role of a regulation parameter, ε is a precision parameter that represents the radius of the tube around the regression function and $\xi_i,\ \xi_i^*$ are the upper and lower training errors respectively that are subject to:

$$|y - (W\emptyset(X) + b)| \leq \varepsilon \tag{2.22}$$

By implementing the Langrage multipliers $\alpha,\ \alpha^*$ and a kernel function K, it is possible to express the model in the dual space as:

$$\hat{y} = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x_j) + b \tag{2.23}$$

Using a kernel function enables handling feature spaces of arbitrary dimensionality without the need to explicitly compute the mapping function $\emptyset(X)$. In this thesis the kernel functions that will be used are:

$$Linear: K(x_i, x_j) = x_i \cdot x_j \tag{2.24}$$

$$Radial\ Basis\ Function(RBF): K(x_i, x_j) = e^{\left(\frac{-\|x_i - x_j\|}{\sigma^2}\right)} \tag{2.25}$$

Where σ is an adjustable parameter.

All parameters must be carefully selected as they implicitly define the structure of the high-dimensional feature space ϕ(X), thereby controlling the complexity of the final solution.

Additionally, the performance of the SVR model is highly dependent on the regularization parameter Cost, the width of the tube Ɛ, and the parameters from the chosen kernels [20].

2.8 K-Nearest Neighbors Regression (KNN Regression)

K-Nearest Neighbors Regression is a non-parametric method that predicts the value of the dependent variable y based on the values of the k nearest neighbors in the feature space X. The algorithm assumes that similar observations (neighbors) will have similar output values.

Given a set of independent variables X and a dependent variable y, the KNN regression algorithm works as follows:

1. For a new observation, identify the k nearest neighbors based on a distance metric. The most used distance metric is Euclidean distance, but several other metrics can be used depending on the nature of the data and the problem at hand.
2. Compute the predicted value $\hat{y}$ as the average of the dependent variable values of the k nearest neighbors:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{2.26}$$

where y_i is the value of the dependent variable for the i-th nearest neighbor. i= 1, 2, …, k.

The number of neighbors k impacts the performance and results of the KNN regression model. When k is small, the model captures fine details in the data but will be sensitive to noise and outliers making it prone to overfit. When the k is large, the model will be more robust to noise and outliers. It produces smoother predictions and better generalizes to unseen data. However, if k is too large, the model may underfit [13], [21].

2.9 Random Forest Regression (RF)

Random Forest Regression is an ensemble learning method that constructs multiple decision trees during training and outputs the average prediction of the individual trees. It improves prediction accuracy and controls overfitting by averaging the results of a diverse set of decision trees. Further information about decision trees can be found in the appendix A.

Given a set of independent variables X and a dependent variable y, the Random Forest algorithm can be summarized as follows:

1. Randomly select subsets of the data with replacement (bootstrap sampling).
2. For each subset, grow a decision tree by recursively splitting the data based on the feature that maximizes the reduction in variance.
3. Aggregate the predictions from all trees to compute the final prediction:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t \tag{2.27}$$

Where T is the number of trees in the forest and $\hat{y}_t$ is the prediction from the t-th tree[22].

Some of the Parameters in Random Forest are:

1. Number of Trees: This parameter specifies the number of trees in the forest. Increasing the number of trees generally improves model performance, as the predictions are averaged over more trees, reducing overfitting. However, after a certain point, the performance gains diminish, and computational cost increases.
2. Number of Features: This parameter determines the number of features to consider when looking for the best split. For regression, typically using root square of the total number of features or a smaller subset of features works well to prevent overfitting and reduce the correlation between trees.
3. Tree Depth: This parameter controls the maximum depth of each tree. Limiting the depth helps prevent overfitting. Deeper trees capture more details and interactions in the data but can lead to overfitting if too deep.
4. Minimum Samples for Split: This parameter specifies the minimum number of samples required to split an internal node. Higher values prevent the model from learning too much from noise, promoting generalization.
5. Minimum Samples per Leaf: This parameter sets the minimum number of samples required to be at a leaf node. Setting this value higher helps smooth the model, as each leaf will contain more samples.
6. Bootstrap: This parameter indicates whether bootstrap samples are used when building trees. If set to True, each tree is built on a bootstrap sample of the data. If False, the whole dataset is used to build each tree.

CHAPTER 3 CASE STUDY: PREDICTIVE MODELING FOR QUALITY CONTROL IN MULTI-STAGE MANUFACTURING USING ARTIFICIAL INTELIGENCE

3.1 Introduction

In Chapter 1, we introduced the significance of maintaining quality control in multi-stage manufacturing systems. These systems encompass processes such as machining, stamping, and assembly, each contributing to the final product's quality. Variations and uncertainties in the production quality at each stage pose challenges of quality control, resulting in increased scrap, production delay, rework, and the necessity for multiple inspections. To mitigate these issues, in-process inspections are commonly employed for ongoing quality evaluation.

In Chapter 2, we reviewed the existing body of work on prediction of quality characteristics (QC) in multi-stage manufacturing systems. We found that machine-learning techniques have been used in previous studies. Based on this, we have selected the most used algorithms in the literature review for our research. These algorithms include Partial Least Square Regression (PLS-Regression) [14], [19], [23], Support Vector Machines for Regression (SVM) [6],[13], K-Nearest Neighbors Regression (K-NN) [13], Random Forest Regression (RF) [8], and Principal Component Regression (PCR). Although Principal Component Regression (PCR) is not explicitly used in any of the papers from the literature review, we have chosen to include it as a baseline model for comparison. In [13], linear regression serves this role, but since in our case the number of independent variables exceeds the number of observations, incorporating Principal Component Analysis is a more appropriate choice.

This thesis presents the application of artificial intelligence (AI) models to predict product quality in successive stages of multi-stage manufacturing processes, based on previous quality characteristics only, and in the absence of data concerning the controllable and uncontrollable process variables. In the published research, predictive models typically rely on process parameters to predict product quality. However, in this thesis we use quality inspection measurements from one stage to predict the quality characteristics (QCs) at subsequent stages. Specifically, we use the QCs' measurements from stage q to predict QCs at stage q+1, and we further extend predictions to stage q+j.

The main constraints in our study are the limited amount of data and the high number of predictor variables. According to the literature, the previously mentioned algorithms, PLS-Regression, SVM, K-NN, RF, and PC-Regression are known for their ability to handle high-dimensional data and perform well even with small datasets, making them suitable choices for our research.

3.2 Problem Description of a Case Study

A multi-stage manufacturing system produces airplane engine parts. Approximately 300-dimensional quality characteristics are associated with each part. This manufacturing process consists of a total of 20 distinct stages. The manufacturing processes at all the stages are statistically in control. This means that the processes are statistically stable. The processes are also capable, which means that the capability indices of all processes are higher than 1.25. As such, the objective of this project is to increase the capability indices and to decrease the percentage defective parts to get closer to six sigma quality level, this means 3.4 ppm [24].

At various intermediate stages, in-process inspections are conducted to examine the dimensional attributes of the parts. These inspections are performed by several Coordinate Measure Machines. The primary objective of these inspections is to evaluate the product's quality and make any necessary adjustments to subsequent operations to compensate and mitigate any quality variations resulting from the previous stages, before further progression of the part in the production flow.

It is important to note that decisions regarding process adjustments are currently based solely on workers' experience. After inspection measurements are obtained at a given stage of the MMS, workers assess the likelihood of parts being out of specification and decide whether to adjust the process or the part. However, relying on experience poses several challenges: worker turnover, retirement, and variability in judgment can lead to inconsistent decisions. Additionally, such subjective evaluations are not always accurate, and unnecessary adjustments can introduce further variability into the process, ultimately harming product quality.

Figure 3.1 shows a representation of the MMS where Sq, Sq+1, and SQ stands for Stage q, Stage q+1 and Stage Q respectively.

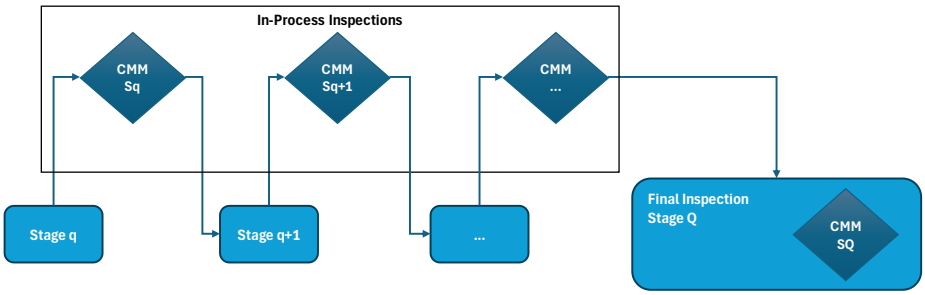

Figure 3.1 : Representation of the MMS

The information gathered from the in-process and final inspections is accessible, offering measurements of the part characteristics as it progresses in the MMS. Data of process parameters or sensor readings are not available for analysis. Moreover, the inspections are performed through sampling, lacking traceability of the inspected parts throughout the process. Consequently, the parts inspected at stage q differ from those inspected at stage *q+1* or the final stage Q. *q= 1, 2, 3, …, Q*.

To address these challenges of lack of traceability, unknown processes' parameters and sensors readings, and the lack of a sampling plan, four controlled lots containing 70 parts each, were produced. During each production run, every part was uniquely identified, and the inspection measurements were recorded and associated with each part. It is worth noting that the only difference between these controlled runs and regular production runs was the traceability of each part; all other aspects remained consistent with standard production processes.

This project focuses on predicting 56 prioritized quality characteristics, as identified by a multidisciplinary team from the manufacturing facility, using data from three specific stages: two consecutive intermediate stages, *q* and *q+1*, and a final inspection stage, *Q*. These stages were also defined by the same multidisciplinary team. Both the 56 QCs and the stages q, q+1 and Q were selected according to the multidisciplinary team's experience and historical data. The 56 prioritized QCs are the most likely to be out of specification and those 3 stages are the most critical where interventions may occur to save or reject parts. The 56 prioritized QCs are included in the inspections of these three stages; however, other QCs may not be present in all three stages.

The primary objective of predicting quality characteristics (QCs) is to reduce reliance on workers' experience for decision-making and to avoid unnecessary adjustments in the manufacturing process

3.3 Methodology

In this section, we describe the methodology used for the prediction of the QCs in a MMS at specific stage based on the QCs at the previous stages. First, we introduce the data and present an exploratory data analysis to understand the initial patterns and distributions. Next, we outline the steps for data preprocessing, including cleaning and transforming the data to ensure that it is usable for modeling. Following this, we introduce the evaluation metrics that are used to assess model performance. Finally, we discuss the model fitting process and the criteria for model selection, which ensure that the best model is chosen for the predictive tasks. Figure 3.2 shows an overview of the methodology section for a better understanding.

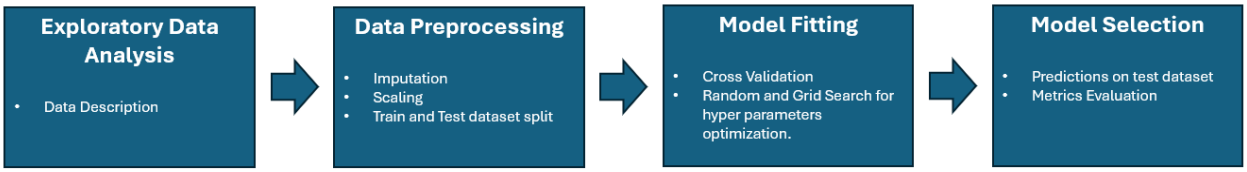

Figure 3.2: Methodology Overview

3.3.1 Exploratory Data Analysis (EDA)

The data from three stages were consolidated into a single dataset containing 21 columns. Each row in this dataset represents a single measurement of a part's quality characteristic. The most relevant columns for the analysis included:

- Date: Date and time of the inspection.
- Part: Identification of the part.
- Lot: The batch to which the part belongs.
- Operation: Stage 1,2…Q of the manufacturing process.
- Quality Characteristic: The name of the QC being measured.
- Measured Value: The result of the measurement.
- Nominal Value: The nominal value of the QC according to specifications.
- Upper Tolerance: Upper tolerance limit for the QC according to specifications.
- Lower Tolerance: Lower tolerance limit for the QC according to specifications.

The data was provided in a long format; however, for predictive modeling uses, a transformation into a wide format was necessary and was therefore performed.

Long format data is organized such that each row represents a single Quality Characteristic's measurement, and each column represents an attribute such as Part Id, QC, and measurement. In

this format, multiple Quality Characteristics' measurements for each part, are recorded in separate rows. Table 3.1 shows an example of dataset in a long format structure.

Table 3.1: Example of long format dataset

Part ID	Quality Charcteristic	Measurement
Part1	QC_A	0.1553
Part1	QC_B	0.6583
Part1	QC_C	0.4758
Part2	QC_A	0.4153
Part2	QC_B	0.797
Part2	QC_C	0.0322
Part3	QC_A	0.1331
Part3	QC_B	0.4634
Part3	QC_C	0.2253

Wide format data is organized such that each row represents a single part, and each column represents different QCs. This format is useful when performing analysis that requires all measurements for a part to be on the same row, such as many machine learning algorithms or statistical models. Table 3.2 shows the dataset in table 3.1 in a wide format structure.

Table 3.2 Example of wide format dataset

Part ID	QC_A	QC_B	QC_C
Part1	0.1553	0.6583	0.4758
Part2	0.4153	0.797	0.0322
Part3	0.1331	0.4634	0.2253

After the data is transformed into a wide format the data set is split into three different datasets, one for each operation *q, q+1,* and *Q*.

Table 3.3 provides a summary of the exploratory data analysis. The first row and second row indicate the stage and the batch respectively. The third row, "Number of QCs," indicates the number of quality characteristics measured at each stage. The fourth row, "Number of parts," shows the number of parts measured in each lot at each stage. Normally the number of parts in one lot is 70, then each part is identified with a consecutive number from 1 to 70. The fifth row, "Missing

Parts," lists the IDs of the parts that were not measured and, consequently, for which data is unavailable. The last row, "Missing Data," indicates the number of missing data points among the parts that were measured.

Table 3.3 :Summary of Exploratory Data Analysis

	Stage q				**Stage q+1**				**Stage Q**			
Lot	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Number of QCs	299	299	299	299	114	114	114	114	110	110	110	110
Number of parts	68	68	64	68	67	70	64	70	67	70	63	70
Missing Parts ID	NA	16, 43	65 to 70	27, 49	16	NA	65 to 70	NA	16	NA	49, 65 to 70	NA
Missing Data	0	1	0	0	0	0	0	0	0	0	54	0

The variations through each QC are represented by a statistical quality distribution. It is noticed that in all stages, some characteristics seem to be bimodal, and some others are represented by discrete continuous distribution as shown in Figure 3.3. Figure 3.3 is divided in two sections A and B that corresponds to the measurements of two different quality characteristics represented in a boxplot and violin plot at the same time. The blue line, which is the violin plot, shows graphically the density function. The density function is estimated using a kernel density estimation. This is a non-parametric way to estimate the probability density function and create a continuous curve that represents the underlying distribution. The violin plot shows the bimodal distribution in A. The black dots correspond to each observation and show that they all fall in only 18 and 12 distinct values in A and B respectively, revealing the discrete distribution. The red dot corresponds to an outlier which is present in most of the QCs. The boxplot, in green, is a graphical representation that was used to summarize the distribution of the quality characteristics, showing its central tendency, spread, and potential outliers. For example, when visualizing the Quality Characteristic Measurements (QCM) of A and B in figure 3.3, a boxplot provides a five-number summary: the minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum.

In a boxplot, the box represents the interquartile range (IQR), which covers the middle 50% of the data, between Q1and Q3. The line inside the box is the median (Q2), indicating the central value of the dataset. The whiskers extend from Q1 and Q3 to the smallest and largest values that are

within 1.5 times the IQR. Any points outside this range are considered outliers. Figure 3.3 was created in R[25] by using the libraries dplyr[26] and ggplot2[27], for data manipulation and graphics respectively.

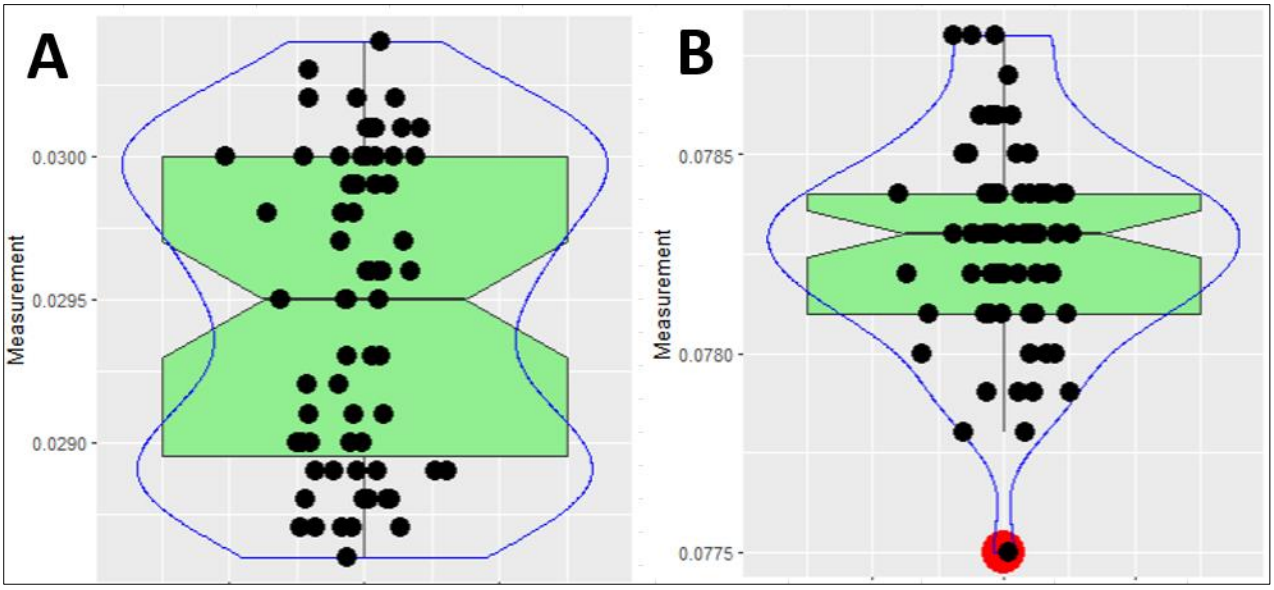

Figure 3.3: Box and Violin Plot of two QCs

The charts in Figure 3.3 display a univariate representation of two quality characteristics (QCs), labeled A and B at a specific stage of the manufacturing system (MMS). Principal component analysis (PCA) was employed to visualize the behavior of the parts in two dimensions at each stage, avoiding the need to examine individual QC charts for each stage. This approach reduces the analysis to a total of 523 charts: 299 for stage *q*, 114 for stage *q+1*, and 110 for stage *Q*. Figures 3.4, 3.5, and 3.6 show every single part of stage q, q+1, and Q, respectively, in a two-dimensional representation. The charts represent a reduction of the original quality characteristics into two principal components (PC1 and PC2), which explain the variability among the parts within each lot. Lots 1, 2, 3 and 4 are represented in orange, green, blue and purple respectively. These 3 figures show that lot 1 is separated from the other lots in the three stages, indicating that its QCs measurements' values differ from the rest of the lots. This suggests a process or material variation that is unique to Lot 1. The lots 2, 3, and 4 are closely grouped, indicating that their quality

characteristics are similar in the 3 stages, with slight variations between them, which suggests that after lot 1, no major changes in the production process were executed. The bimodal distribution observed in some QCs, as in figure 3.3 section A, appears because of the difference between Lot 1 and the rest of the lots (2, 3, and 4). If we plot only the QC data for Lot 1 or only for Lots 2, 3, and 4 separately, the distribution of some QCs would no longer be bimodal. Instead, it would show a more uniform distribution within each group. The bimodal shape happens only when we combine the data for all the lots, as Lot 1 has QC values that are clearly different from the rest.

The way the changes after lot 1 affected the quality of the part cannot be concluded only from the separation seen in the graphs. Quality depends on the tolerances of each QC. If the QC values in Lot 1 are further from the center of the tolerance range or exceed the acceptable limits, this may indicate lower quality. On the other hand, if the values in Lot 1 are better centered within the tolerance limits, its quality could be considered higher. Finally, the points that are distant from the 2 clusters confirm the presence of outliers.

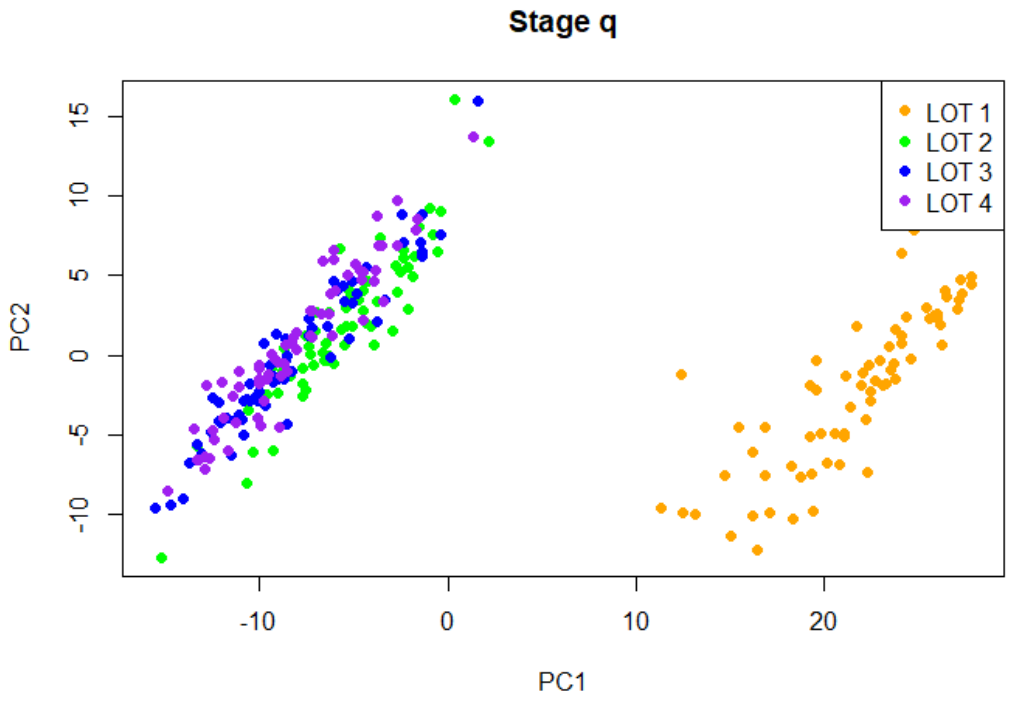

Figure 3.4 : PCA of QCs in stage q

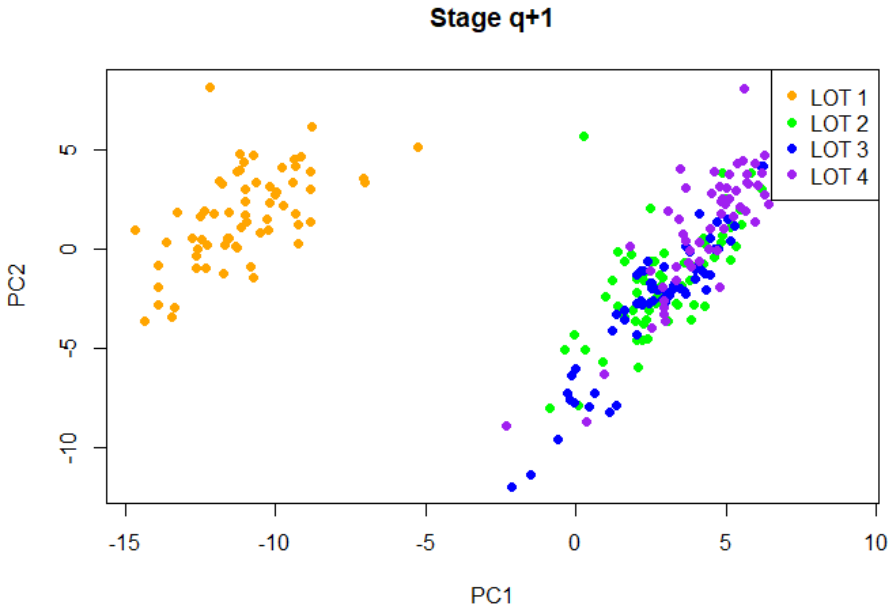

Figure 3.5 : PCA of QCs in stage q+1

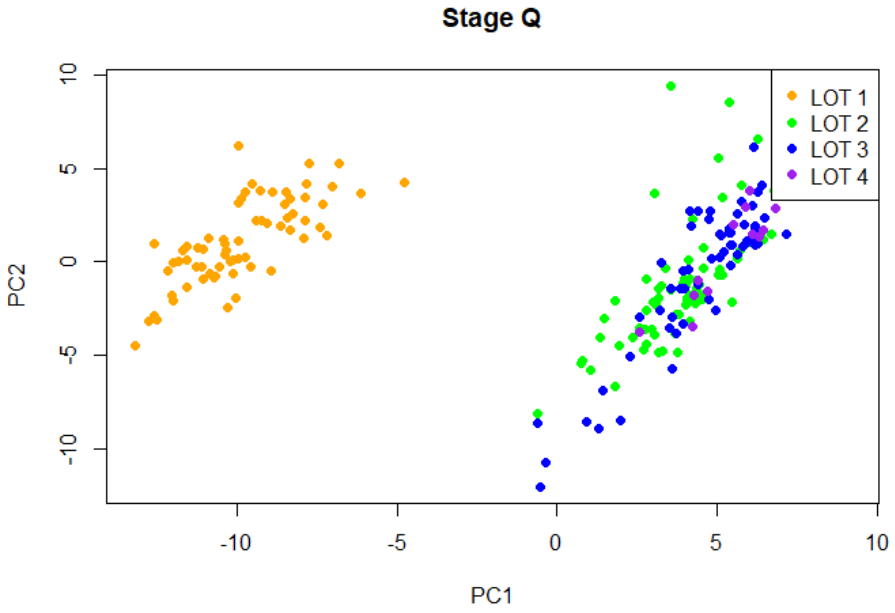

Figure 3.6 : PCA of QCs in stage Q

Not all the QCs were affected after production of lot 1. As an example, Figure 3.7 shows a run chart of a QC at stage q that does not show a shift after lot 1. Figure 3.8 shows other QC at stage q

that present a noticeable shift after lot 1. Figure 3.9 shows a smooth shift after lot 1, figure 3.10 shows a progressive shift, and finally, figure 3.11 shows a shift after each lot. The lots 1 ,2 3 and 4 were produced in that same chronological order, which allows us to use a run chart. A run chart is a graphical tool used to display data points in time order, helping to visualize trends, shifts, or patterns in a process over time[28].

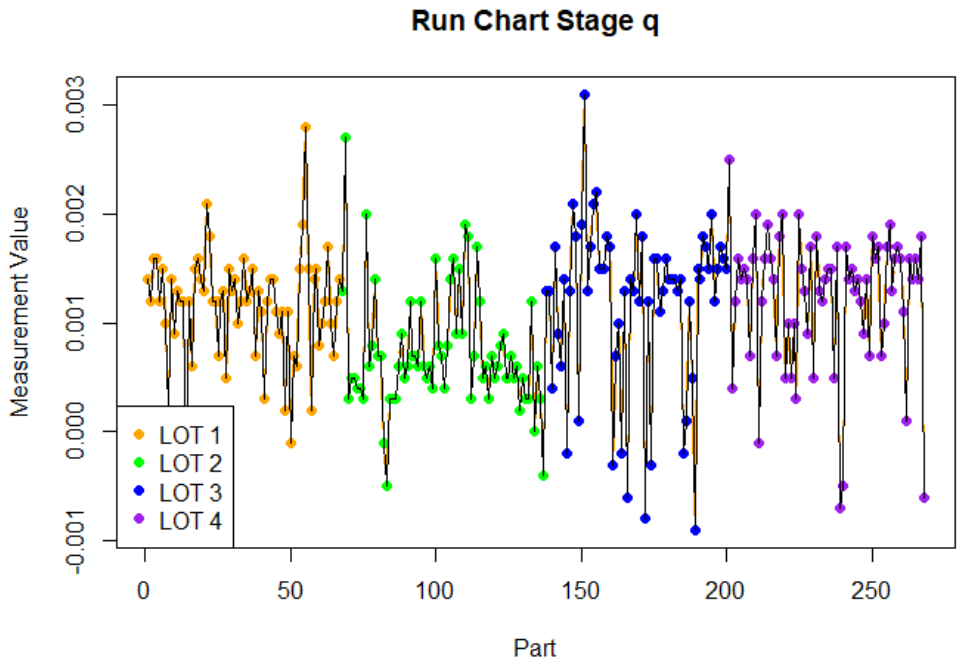

Figure 3.7 : Run Chart of a QC without shift

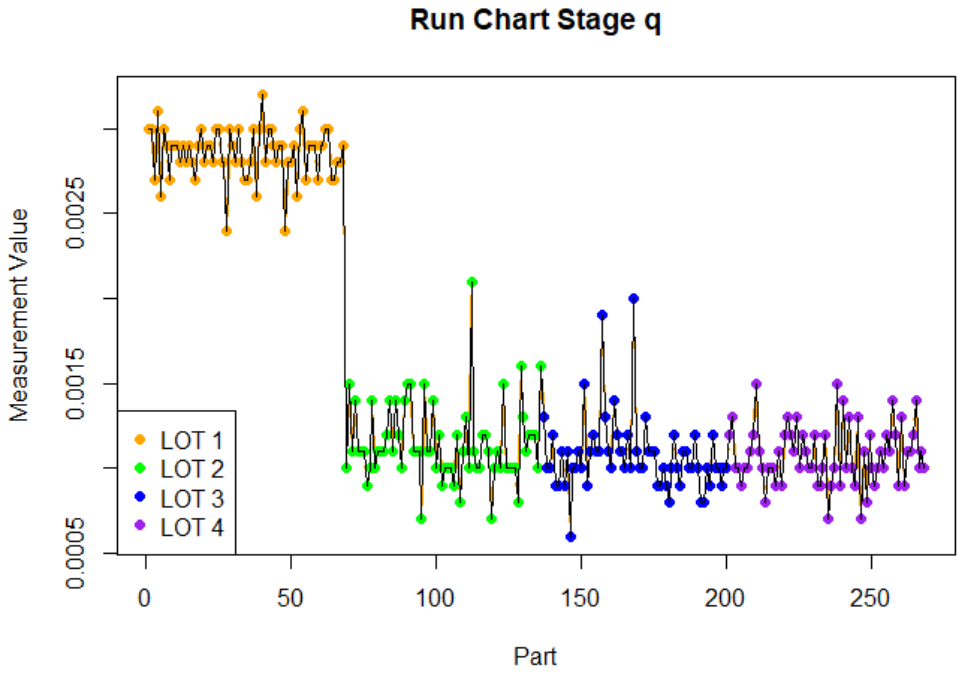

Figure 3.8 : Run chart of a QC with a drastic shift after lot 1

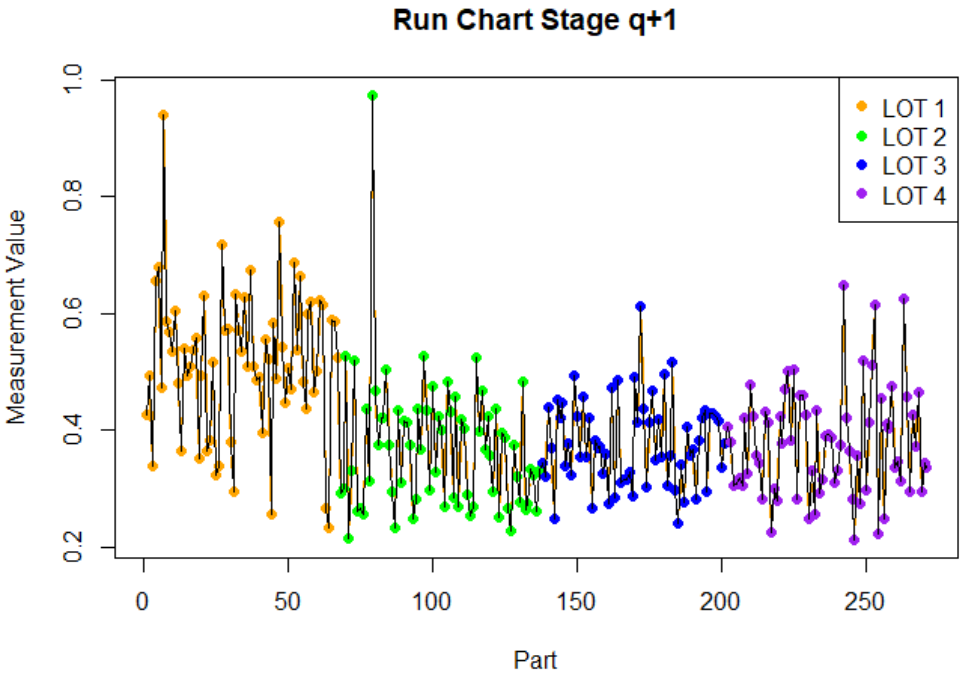

Figure 3.9 : Run chart of a QC with slight shift after lot 1

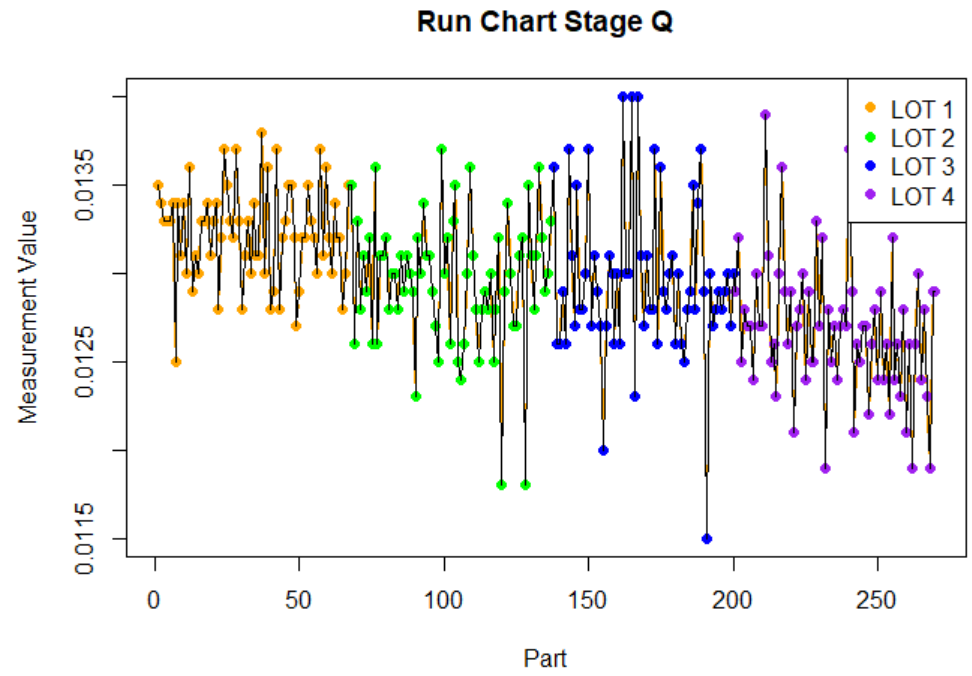

Figure 3.10 : Run chart with progressive shift

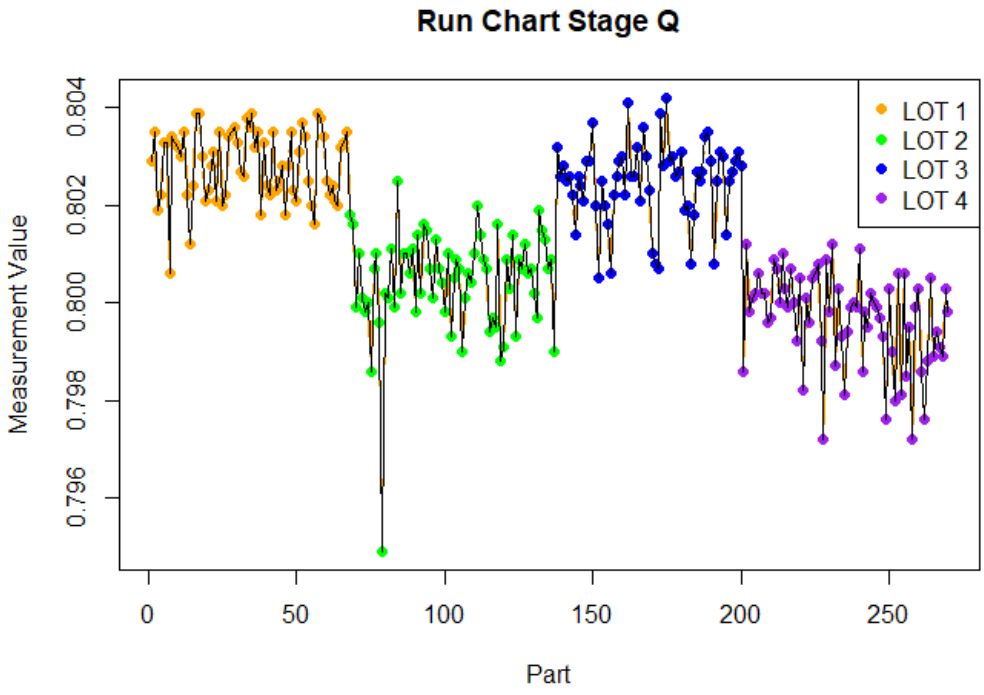

Figure 3.11 : Run chart with shift after each lot

Figures 3.7 to 3.11 show different examples of the variation of different QCs within specific stages of the MMS. According to the PCA visualizations (figures 3.4 to 3.6) we can conclude whether major changes in the production process occurred after lot 1 was produced. Table 3.4 shows how many QCs were affected at each stage after the production of lot 1. The QCs affected after the production of lot 1were visually identified by observing every run chart for each QC.

Table 3.4 : Number of QCs affected after production of lot 1

Stage	**Total Number of QC**	**Number of QCs Affected**	**Percentage of QCs Affected**
Stage q	299	255	85.28%
Stage q+1	114	70	59.82%
Stage Q	110	69	62.72%

3.3.2 Data Preprocessing

Let *X* and *Y* be the set of independent and dependent QCs respectively. Before fitting predictive models, preprocessing is performed in the input data *X* according to the Exploratory Data Analysis findings described in section 3.3.1.

First, all missing parts are removed to retain only the parts that exist in all three stages. This ensures the same number of parts in each stage. In total, 266 observations were retained for each stage. This means that 266 identified parts were tracked throughout the three stages.

It is decided to keep the outliers since the data is limited and because the production team did not find any assignable causes for them, the quantity of data is limited and to ensure that the analysis reflects the full variability observed in the process. However, outliers affect the prediction accuracy of the regression models**.** Among the six proposed algorithms, the most robust to outliers is Random Forest, as it uses decision trees that split the data based on rules rather than relying on metrics like mean or variance, which are sensitive to outliers. Next is SVM with the RBF kernel, which handles outliers by adjusting a non-linear boundary and reducing the influence of outliers, if they are not excessively far from the data. KNN follows, as its sensitivity to outliers depends on

the number of neighbors and scaling the data. PLSR and PCR are less robust, as both rely on variance to extract components, making them sensitive to outliers. Finally, Linear SVM is the least robust, since outliers can directly shift the linear decision boundary, affecting the model's performance.

The datasets are split into training and testing datasets with a proportion of 0.80 and 0.20 respectively.

The input data is scaled using the standardization method in equation 3.1, which is performed by subtracting the average value of the QC from each value of a data point and then dividing the resulting value by the standard deviation. The standardized QC has a mean of 0 and a standard deviation of 1.

$$Z_{pi}^{q} = \frac{X_{pi}^{q} - \mu_{p}^{q}}{\sigma_{p}^{q}} \tag{3.1}$$

Where:

- Z_{pi}^{q} is the scaled value corresponding to the *i-th* observation of the *p-th* independent variable from the *q-th* stage.
- X_{pi}^{q} is the original value corresponding to the *i-th* observation of the *p-th* independent variable from the *q-th* stage.
- μ_{p}^{q} is the mean of the *p-th* independent variable at the *q-th* stage.
- σ_{p}^{q} is the standard deviation of the *p-th* independent variable at the *q-th* stage.

Where i = {1, 2, ..., n}; p= {1, 2, ..., P}; q= {1, 2, ..., Q}.

For the missing data, the K-Nearest Neighbors (KNN) imputation method is used to fill in the gaps, thereby minimizing information loss. The number of neighbors was set to 5. This means the imputation uses the average of the 5 nearest neighbors to fill in the missing values. The Euclidean distance is used as a distance metric.

Finally, to reduce the dimensionality of the datasets that take the role of independent variables, a Principal Component Analysis (PCA) is performed and only components explaining up to 95% of the variance are retained. PCA is not utilized in conjunction with PLS Regression and PC Regression, as these algorithms inherently incorporate dimensional reduction as part of their processes and the number of components is part of their hyperparameters. It is worth noticing that

the dataset from Stage *q+1* plays both roles, as input to predict the characteristic at stage *Q* and as output for stage *q* (56 QCs). The data preprocessing is only done when it functions as an input.

3.3.3 Evaluation Metrics

The metric used for model fitting, and model selection is the Root Mean Square Error (*RMSE*). Once we have selected the models that lead to the smallest *RMSE*, the coefficient of determination (R^2), the mean absolute error (*MAE*), the error percentage compared to the data variation (*%EV*), and the error percentage compared to the tolerance *(%ET*) are computed and used to evaluate the models' performance in the test dataset. The *%EV* and *%ET*, shown in equations 3.2 and 3.3, are metrics that we developed to enhance precision assessment. The QCs to be predicted involve small-scale variations due to specifications that seeks Six Sigma quality. Therefore, evaluating models using only *RMSE* or *MAE* only will not be informative of the accuracy of the prediction model. For example, let us assume that the resulting *RMSE* for the prediction of a given QC is 0.0003. Initially, this might be viewed as a low *RMSE* value, based on the number itself. However, if the standard deviation of that QC is 0.00001, the *RMSE* can no longer be regarded as sufficient measure of the prediction quality, within that context. *%EV* and *%ET* offer representation of prediction accuracy by accounting for the variability in actual values (*%EV*) and the tolerances specific to the QCs (*%ET*), particularly informative when deciding on the implementation of a prediction model.

$$\%EV_p^q = \frac{MAE_p^q}{\max\left(y_p^q\right) - \min\left(y_p^q\right)} \tag{3.2}$$

$$\%ET_p^q = \frac{MAE_p^q}{\mathrm{UTL_p^q} - \mathrm{LTL_p^q}} \tag{3.3}$$

Where:

- MAE_p^q is mean absolute error of the *p-th* QC being predicted at the *q-th* stage.
- y_p^q is the *p-th* QC being predicted at the *q-th* stage.
- $\mathrm{UTL_p^q}$ and $\mathrm{LTL_p^q}$ are the upper and lower Tolerance limit respectively of the *p-th* QC being predictedat the *q-th* stage. Where $p=\{1,2,\ldots,56\}$, $q=\{1, 2, 3\}$

The coefficient of determination R^2 measures the proportion of the variance in the dependent variable, in our case the quality characteristic of the next stage, that is explained by the predictor variables, in this case the quality characteristics of the previous stages, in the regression model. In

other words, R^2 indicates how well the predictor variables in the model explain the variability of the dependent variable[29].

$$R^2 = 1 - \frac{SSres}{SStot} \tag{3.4}$$

$$SSres = \sum_i^n \left(y_i^{qp} - \hat{y}_i^{qp}\right)^2 \tag{3.5}$$

$$SStot = \sum_i^n \left(y_i^{qp} - \overline{y^{qp}}\right)^2 \tag{3.6}$$

Where in a given stage *q* of a MMS:

- $\hat{y}_i^{qp}$: the predicted value for the *i-th* observation of the *p-th* quality characteristic from the *q-th* stage.
- y_i^{qp} : the real value of the *i-th* observation of the *p-th* quality characteristic from the *q-th* stage.
- $\overline{y^{qp}}$: is the average of the *p-th* quality characteristic from the *q-th* stage.

Where "*i*" represent the number of or observations in a dataset. *i = 1, 2, …, n*. And "*p*" is the quality characteristic to be predicted. *p = 1, 2, …P*.

SSres and *SStot* are the Sum of squares of residuals and the Total Sum of Squares respectively.

3.3.4 Model Fitting and Model Selection.

Let's keep in mind that every QC to be predicted has its own prediction model. Figure 3.12 shows a visual representation of the modeling approach. Notably three modeling approaches are proposed. $Models_{q,q+1}$take as independent variables or inputs the QC of stage q, to predict 56 QCs from stage q+1. Similarly, $Models_{q+1,Q}$ take as input the QCs of stage q+1 to predict 56 QCs of stage Q. Finally, $Models_{q,Q}$, take as input the QCs of stage q to predict 56 QCs of stage Q.

Another possible modeling approach is to use data from both stage q and stage q+1 as input variables to predict the QCs of stage Q. However, this approach will not be developed in this thesis because it was considered more interesting to predict stage Q directly using data from stage q without relying and waiting on the QCs measurements of stage q+1. This approach increases reactivity and anticipation.

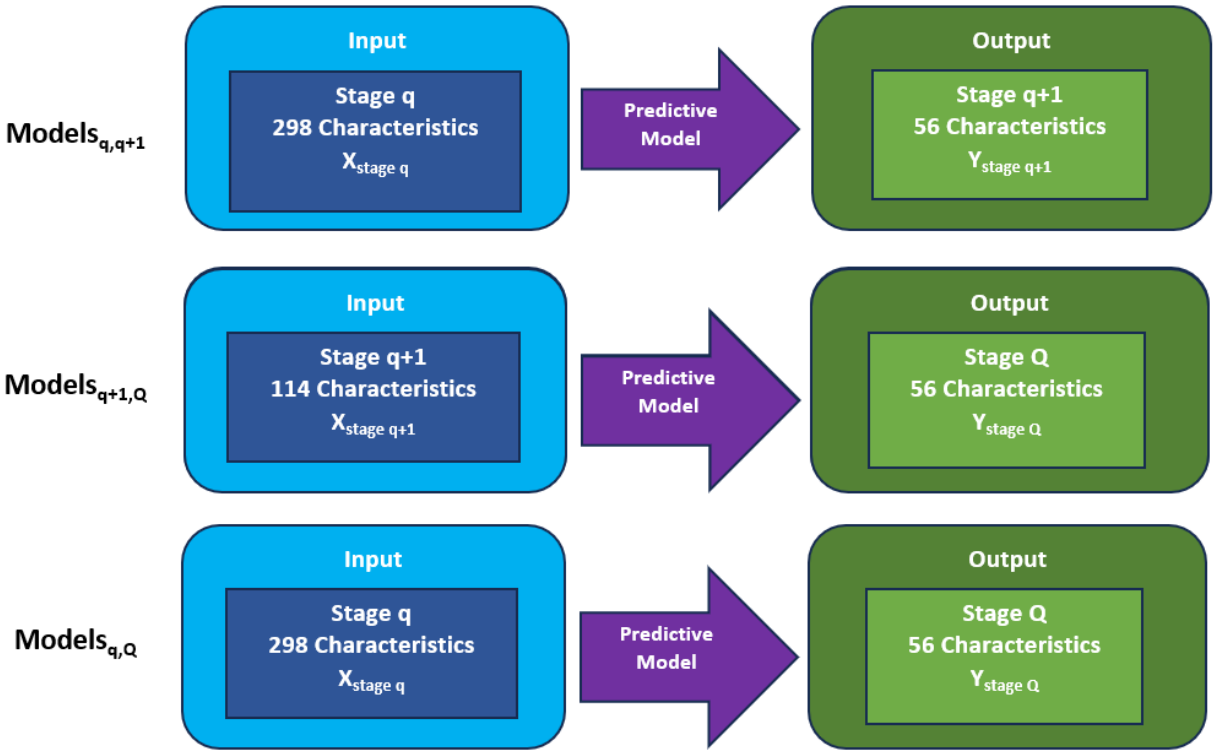

Figure 3.12: Modeling approach representation.

All pre-processing and modeling is performed by using the R library Caret[30].

The caret package (Classification and Regression Training) in R is a toolkit for building predictive models. It provides a unified interface to streamline the process of training, tuning, and evaluating machine learning models. Caret allows to pre-process data, select optimal features, and choose the best model configuration through cross-validation.

Each algorithm is adjusted to predict each characteristic of "Y_{q+1}" and "Y_Q" using the training data. Hyperparameter tuning is conducted using random search and cross-validation for Random Forest, and gride search and cross validation for PLS-Regression, PCR, SVMRBF, SVRLinear and KNN.

Random search is chosen for Random Forest because it is the algorithm with the largest number of hyperparameters involved, which makes an exhaustive search computationally expensive. Random search allows for an exploration of the hyperparameter space by randomly sampling combinations of hyperparameters. In the case of Random Forest, 20 random combinations of hyperparameters were evaluated. The hyperparameters tuned were:

- The minimum number of samples in a node for splitting. (min.node.size)
- The number of randomly chosen features considered at each split. (mtry)
- The method used to determine the best split. (splitrule)

This process iterates through all combinations of hyperparameters randomly selected, allowing the identification of optimal parameters that maximize predictive performance.

An example of the explored hyperparameter combinations for Random Forest during random search is shown in Table 3.5. We observe that the best combination of hyperparameter is the second one with the lowest *RMSE*.

Table 3.5 : Example of random search for Random Forest

min.node.size	**mtry**	**splitrule**	***RMSE***
3	7	extratrees	0.00033818
3	23	extratrees	0.00028856
4	2	extratrees	0.00043004
5	9	maxstat	0.00032161
5	10	variance	0.00031001
7	9	extratrees	0.0003491
7	21	maxstat	0.00030257
9	10	extratrees	0.00035234
9	11	extratrees	0.00031754
9	9	extratrees	0.00029376
9	5	maxstat	0.00030305
11	5	variance	0.00030362
14	5	maxstat	0.00037582
14	5	variance	0.00034741
16	12	variance	0.00030997
18	12	variance	0.00030095
19	5	extratrees	0.00031984
19	25	maxstat	0.00031733
20	13	extratrees	0.00031995

Grid search was employed for the rest of the models since they have fewer hyperparameters than Random Forest. Grid search explores all possible combinations of hyperparameters within a given range which make it computationally expensive as the number of hyperparameters increases.

For PLS-Regression and PC-Regression, the only hyperparameter is the number of latent variables and principal components respectively. For both algorithms the range of search is set from 1 to 50.

For KNN the only hyperparameter to be tuned is the number of neighbors. The range of search is: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, and 24 neighbors.

For SVR-Linear the single hyperparameter is the Cost. The range of search is set to 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64 and 128.

Lastly, SVR-RBF has two hyper parameters: Cost and sigma. The range of search for cost is set to 0.1, 1,2,4,8, 10 and 20. The range of search for sigma is set to: 0.01, 0.015, 0.03, 0.1, 1, 5 and 10. Table 3.6 shows an example of the explored hyperparameter combinations for the algorithm SVR-RBF. The fourth combination of hyperparameters yield to the lowest *RMSE*.

Table 3.6 : Example of grid search for SVR-RBF

Sigma	**Cost**	***RMSE***
0.01	0.1	0.00050609
0.01	1	0.00022376
0.01	2	0.00020788
0.01	4	0.00020044
0.01	8	0.00020063
0.01	10	0.00020137
0.01	20	0.00020884
0.015	0.1	0.00048889
0.015	1	0.00022908
0.015	2	0.00021635
0.015	4	0.00021035
0.015	8	0.00021367
0.015	10	0.00021613
0.015	20	0.00022052
0.03	0.1	0.00047683
0.03	1	0.00025978
0.03	2	0.00024769
0.03	4	0.00024893
0.03	8	0.00025139
0.03	10	0.00025222
0.03	20	0.00025221
0.1	0.1	0.00053481
0.1	1	0.00039071

Table 3.6: Example of grid search for SVR-RBF (continuation and end)

Sigma	**Cost**	***RMSE***
0.1	2	0.00037592
0.1	4	0.00037544
0.1	8	0.00037544
0.1	10	0.00037544
0.1	20	0.00037544
1	0.1	0.00058272
1	1	0.0005762
1	2	0.00057259
1	4	0.0005728
1	8	0.0005728
1	10	0.0005728
1	20	0.0005728
5	0.1	0.00058273
5	1	0.00057628
5	2	0.00057268
5	4	0.0005729
5	8	0.0005729
5	10	0.0005729
5	20	0.0005729
10	0.1	0.00058273
10	1	0.00057628
10	2	0.00057268
10	4	0.0005729
10	8	0.0005729
10	10	0.0005729
10	20	0.0005729

Cross-validation is a model performance evaluation technique that involves dividing the training dataset into "k" subsets and iteratively train the model with the k-1 datasets that form one training dataset, then test and evaluate the model on each remaining subset. The model's accuracy is the average of the k tests' accuracies. In our case, "k" is set to 10. To select the best hyperparameter combination, each set of hyperparameter combinations are trained and evaluated using cross - validation. In tables 3.4 and 3.5, the *RMSE* values represent the average *RMSE* obtained from the

10-folds-cross validation. The combination of hyperparameters with the lowest *RMSE* average is selected as the best model for a specific algorithm.

This process is performed with the six algorithms for each QC to predict the value of the predicted QC by each of the three modeling approaches as shown in figure 3.12.

After model adjustment, we have six predictive models (one for each algorithm) for each QC of Y_{q+1} and Y_Q totaling 1008 models. The goal is to select the best model for each QC. That means 168 models. 56 Model *q,q+1*to predict the QC of stage *q+1* taking as input X_q, 56 Models *q+1, Q* to predict the QCs of stage *Q* taking as input $X_{q+1,}$ and 56 Models *q,Q* to predict the QCs of stage *Q* by taking as input X_q as shown in figure 3.12. To achieve this, each algorithm predicted the QC of the testing dataset, and the model with the lowest *RMSE* value is chosen. We relied on *RMSE* since we aim to retain the best model for more accurate predictions. *RMSE* penalizes errors due to its quadratic nature.

3.4 Results

3.4.1 Summary of metrics

The following three tables, 3.7, 3.8 and 3.9, present a summary of the models' performance. Each table presents the average results of the modeling approaches according to figure 3.12. As a reminder, each modeling approach is composed of 56 predictive models, one per QC to be predicted. The results shown in the next three tables contain a summary of the 56 models. Appendix B shows the detailed results for each single model.

Table 3.7 presents the model's Performance Summary to Predict Y_{q+1} Using X_q as independent variables. The average *RMSE* and *MAE* indicate a low error rate being 0.00032703 and 0.00024423, respectively. The R^2 average of 0.7496994. Notably, the boxplot for R^2 in figure 3.13 shows that 75% of the models have an R^2 value higher than approximately 0.63, and 50% above 0.80, highlighting the robustness of most models. The *%EV* averages at 6%, with a maximum value reaching 12%. For our purpose, we considered these results acceptable, and less acceptable if they exceed 10%. This rule depends on the criticality of the quality characteristic being predicted, according to the production experts. The boxplot for *%EV* in Figure 3.13 shows that 75% of the models have *%EV* values below 8%, further emphasizing the reliability of the models. The *%ET* values' average is 2%. Figure 3.13 shows that 75% of the models have *%ET* values below 3%, and

the maximum value is 6% which indicates that the error of the models compared to the tolerances' range remains low for all the predictive models.

Table 3.7 : Modelq,q+1 performance summary to predict Y_{q+1} taking as independent variables X_q

	RMSE	***R^2***	***MAE***	***%EV***	***%ET***
Average	0.00031949	0.73185399	0.00023655	6%	2%
Maximum	0.00084244	0.9730143	0.00067847	13%	6%
Minimum	0.00011072	0.00176	0.0000898	3%	1%
Standard Deviation	0.00017317	0.24303772	0.0001302	2%	1%
Range	0.00073172	0.9712543	0.00058867	10%	4%

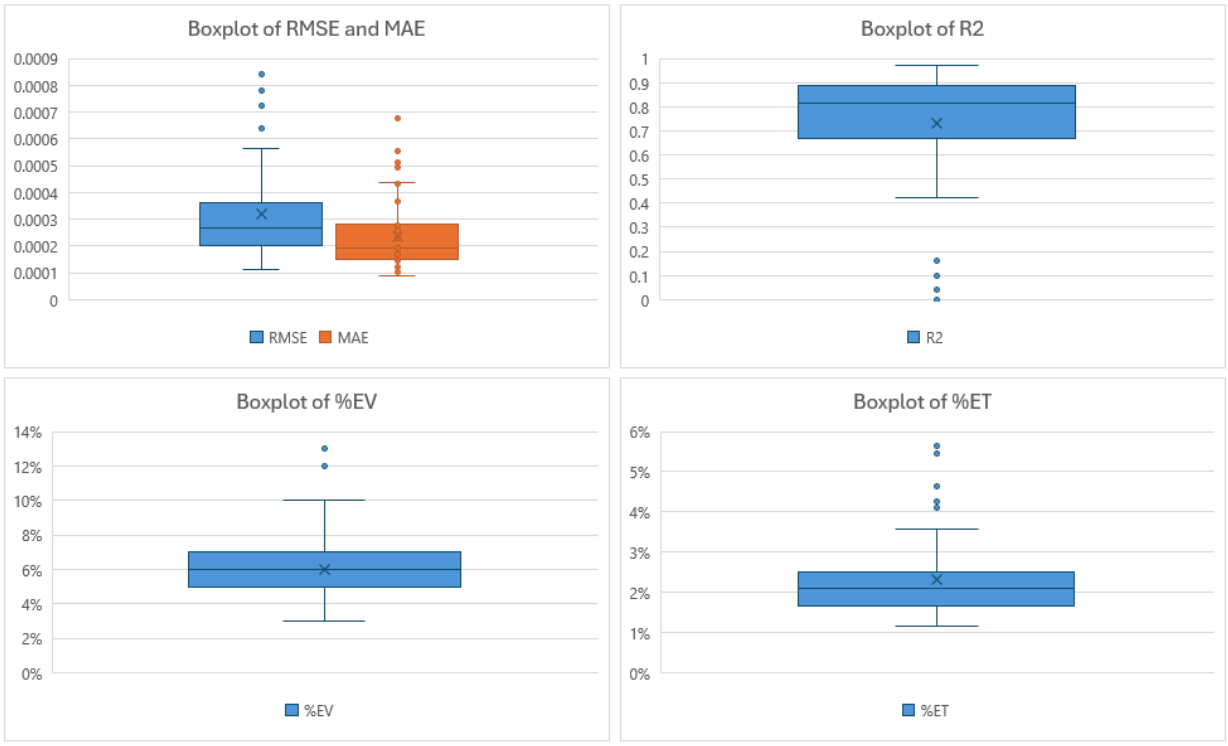

Figure 3.13 : Boxplots of Model's performance summary to predict Y_{q+1} taking as independent variables X_q

In Table 3.8: Modelq+1,Q performance summary to predict Y_Q taking as independent variables X_{q+1}, the average *RMSE* and *MAE* are 0.00026233 and 0.00019347, respectively. The R^2 average is 0.8043. Notably, the boxplot for R^2 in Figure 3.14 shows that 75% of the models have an R^2 value higher than approximately 0.68, and 50% above 0.84. The *%EV* averages at 5%, with a maximum value reaching 9%. Figure 3.14 shows that most models (75%) have *%EV* values below

6. The *%ET* values' average is 1.8%. Figure 3.14 shows that 75% of the models have *%ET* values below 2%, and the maximum value is 2.7%, which indicates a low error of the models compared to the tolerances' range for all the predictive models.

Table 3.8 : Modelq+1,Q performance summary to predict Y_Q taking as independent variables X_{q+1}

	RSME	***R^2***	***MAE***	***%EV***	***%ET***
Average	0.00026233	0.80431383	0.00019347	5.0%	1.8%
Maximum	0.00073259	0.98965855	0.00049922	9.0%	2.7%
Minimum	0.00012681	0.32098906	0.00010369	2.0%	0.9%
Standard Deviation	0.00013597	0.1605568	9.2743E-05	1.7%	0.4%
Range	0.00060578	0.6686695	0.00039553	7.0%	1.9%

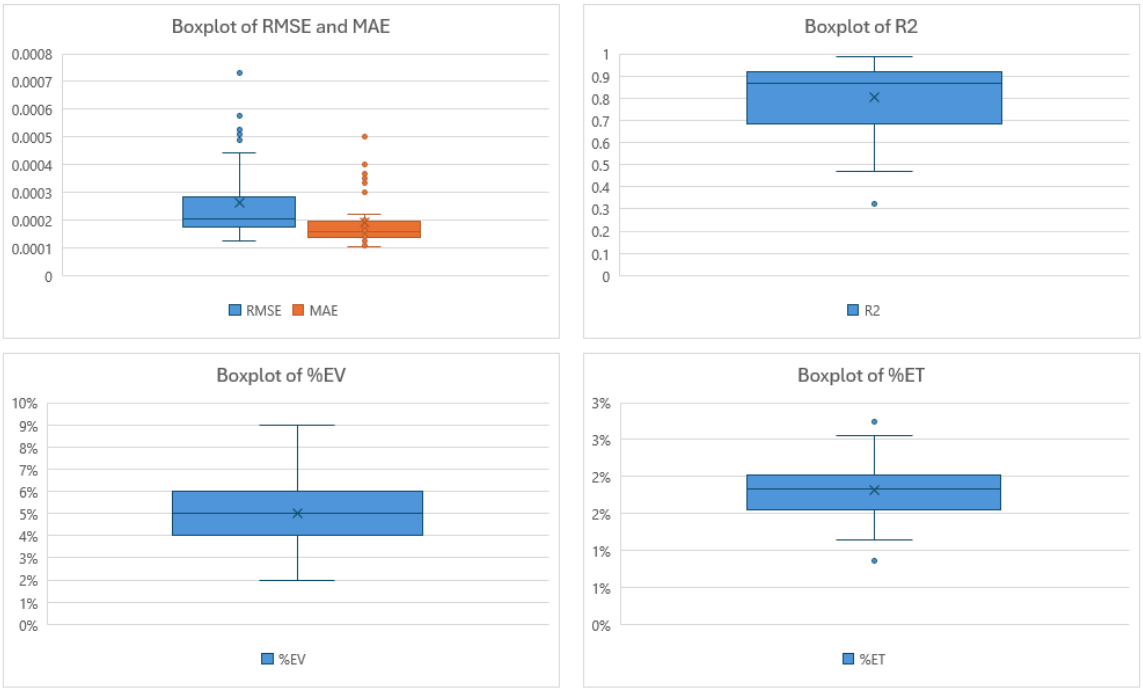

Figure 3.14 : Boxplots of Model's performance summary to predict Y_Q taking as independent variables X_{q+1}

In Table 3.9: Model's Performance Summary to Predict Y_Q Using X_q as independent variables, the metrics indicate solid performance. The average *RMSE* and *MAE* suggest a low error rate, being 0.00032703 and 0.00024423, respectively. The R^2 average is 0.7496994. From Figure 3.15 we

observe that 75% of the models have an R^2 value higher than approximately 0.62, and 50% above 0.82.

The *%EV* averages at 5.8%, with a maximum value reaching 12.1%. As commented before, it may be considered less acceptable if it exceeds 10%, depending on the criticality of the quality characteristic being predicted. The boxplot for *%EV* in figure 3.15 reveals that 75% of the models have *%EV* values below 7%.

The *%ET* values' average is 2.2%. Figure 3.14 shows that 75% of the models have *%ET* values below 2.5%, with a maximum value of 4.9%, indicating that the models' error compared to the tolerances' range remains low across all predictive models.

Table 3.9 : Modelq,Q performance summary to predict Y_Q taking as independent variables X_q

	RSME	***R^2***	***MAE***	***%EV***	***%ET***
Average	0.00032703	0.7496994	0.00024423	5.8%	2.2%
Maximum	0.00081625	0.97940568	0.00063217	12.1%	4.9%
Minimum	0.00012811	0.0000524	0.0000975	2.6%	1.2%
Standard Deviation	0.00020019	0.22499227	0.00015044	1.9%	0.8%
Range	0.00068814	0.97935328	0.00053467	9.5%	3.6%

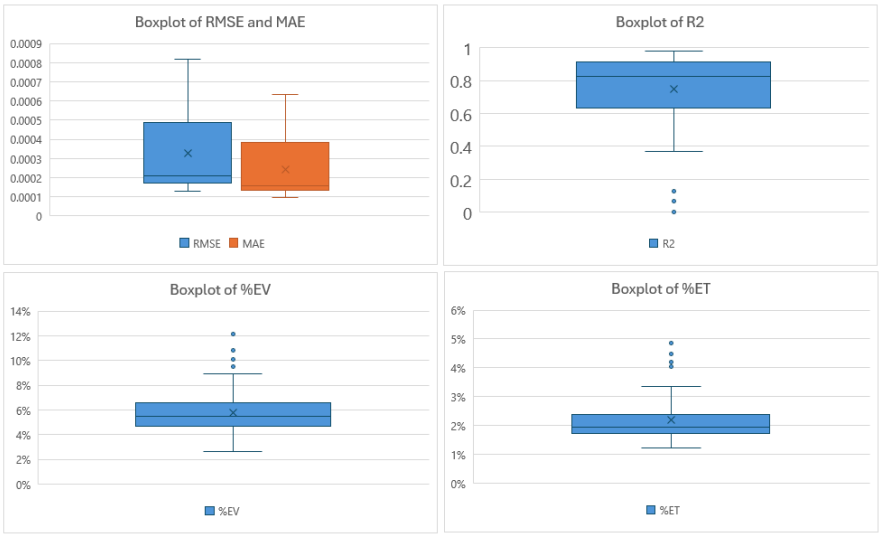

Figure 3.15: Boxplots of Model's performance summary to predict Y_Q taking as independent variables X_q

3.4.2 Plot of predicted QCs values vs actual QCs values

In this section, the quality of the predictions of QCs is obtained by comparison of the prediction to the actual values in the test data of various models. These are visually presented in figures 3.15 to 3.26. For each of the three modeling approaches, an example of four QCs prediction will be shown: one with a high value of R^2 value, two with intermediate values R^2 values, and one with a low value R^2 value.

In Figures 3.16 to 3.27, the red lines and points represent the values predicted by the model, while the black lines and points represent the actual values. The abbreviation of the algorithm used is shown in the upper right corner, the R^2 value is displayed in the lower left corner, and the *RMSE* value is in the lower right corner.

We can observe that when the R^2 value is high, the predicted values closely follow the actual values. Conversely, when the R^2 value is low, the model fails to accurately predict the actual values. This is not surprising, as R^2 measures the proportion of the variance in the dependent variable, Y, that is predictable from the independent variables. A higher R^2 value indicates a better fit, meaning the model explains a larger portion of the variance in the actual values, resulting in more accurate predictions. Conversely, a lower R^2 value indicates that the model does not explain much of the variance, leading to poorer predictive performance.

Figures 3.16 to 3.19 show the graphs of $models_{q,q+1}$ to predict Y_{q+1} taking as independent variable X_q. Figures from 3.120 to 3.23 show the graphs of $models_{q+1,Q}$ to predict Y_Q taking as independent variable $X_{q+1,}$ and finally, figures from 3.24 to 3.27 show the graphs of $models_{q,Q}$ to predict Y_Q taking as independent variables X_q.

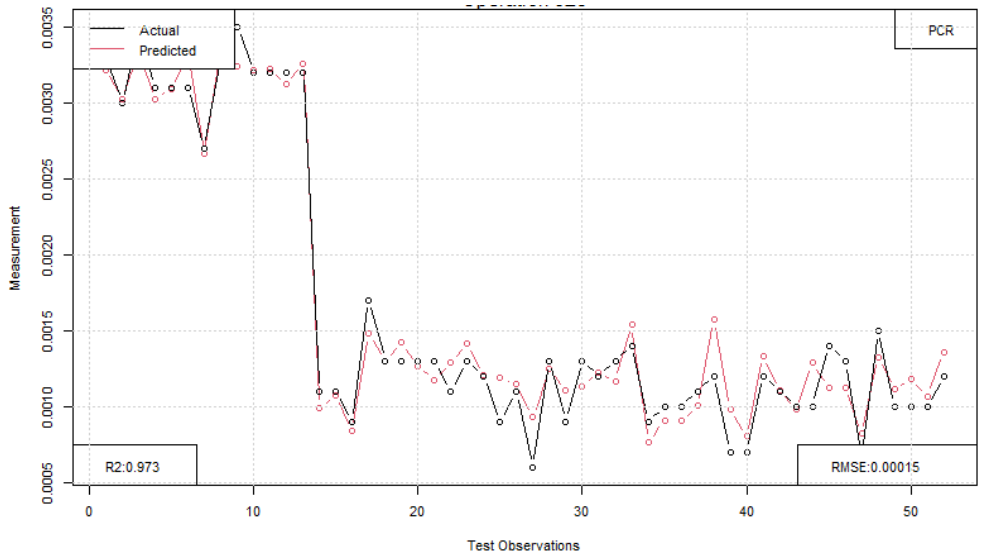

Figure 3.16: Prediction vs Actual values of QC_53

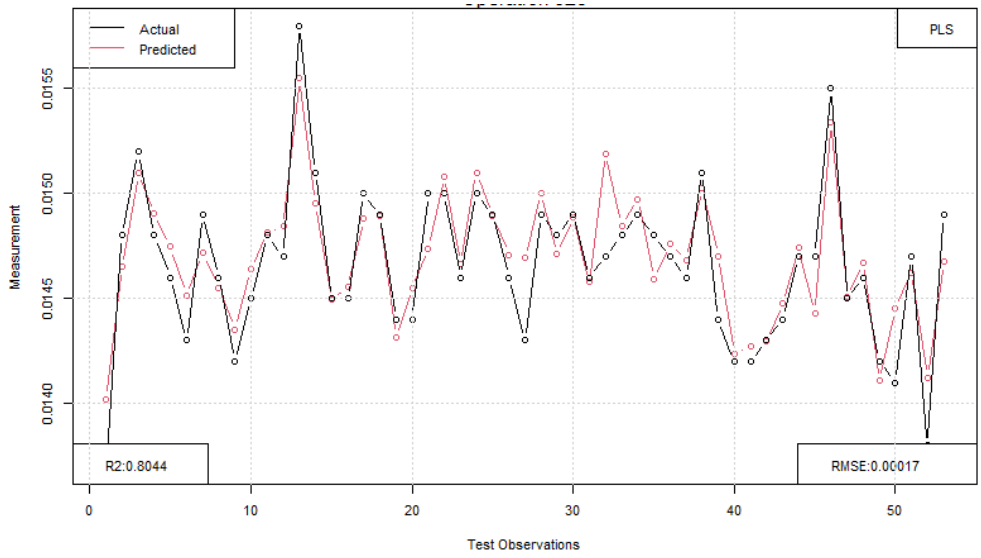

Figure 3.17: Prediction vs Actual values of QC_37

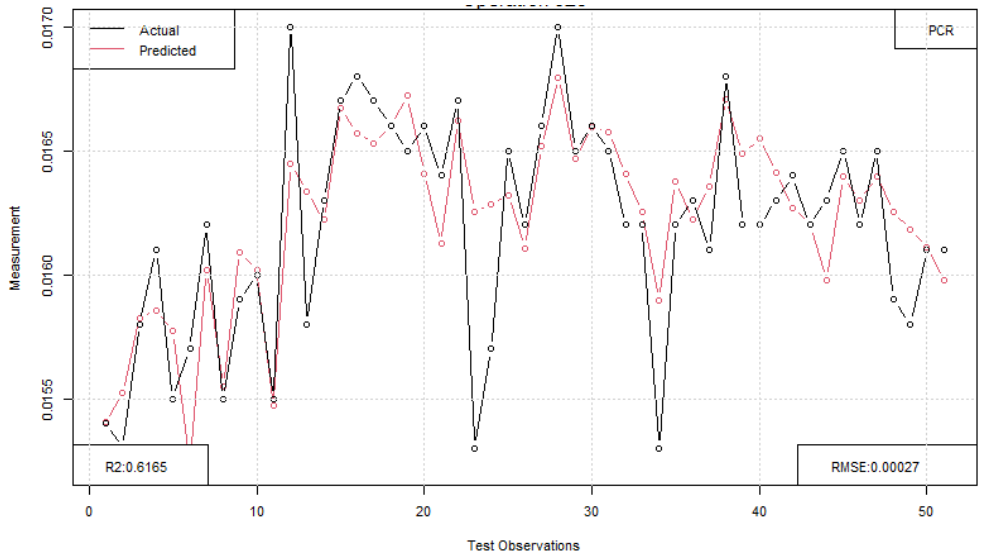

Figure 3.18 : Prediction vs Actual values of QC_39

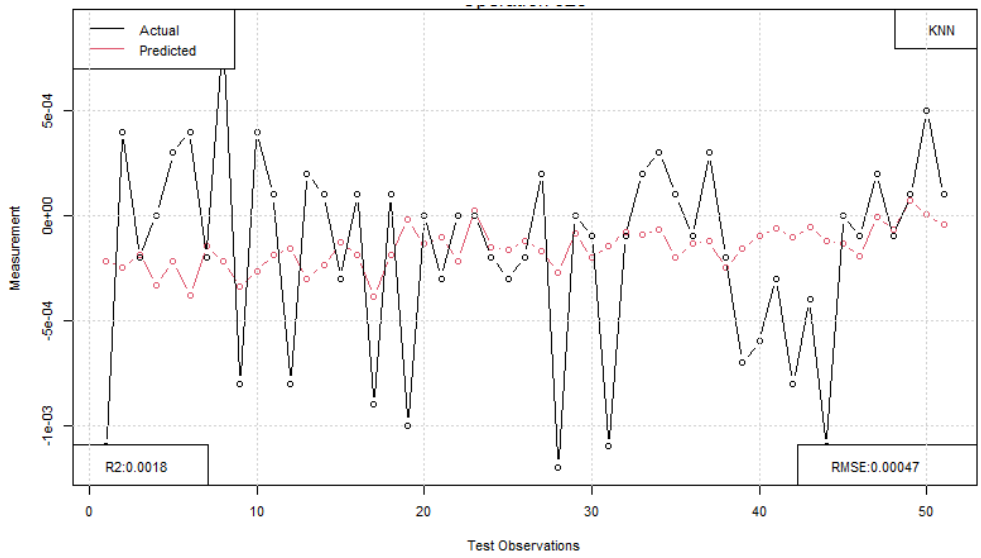

Figure 3.19 : Prediction vs Actual values of QC _5

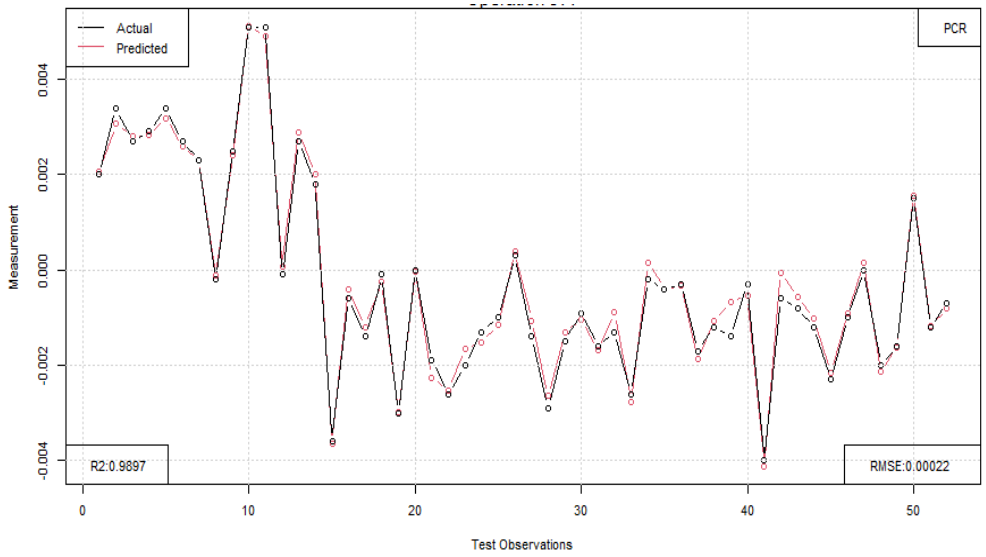

Figure 3.20 : Prediction vs Actual values of QC_9

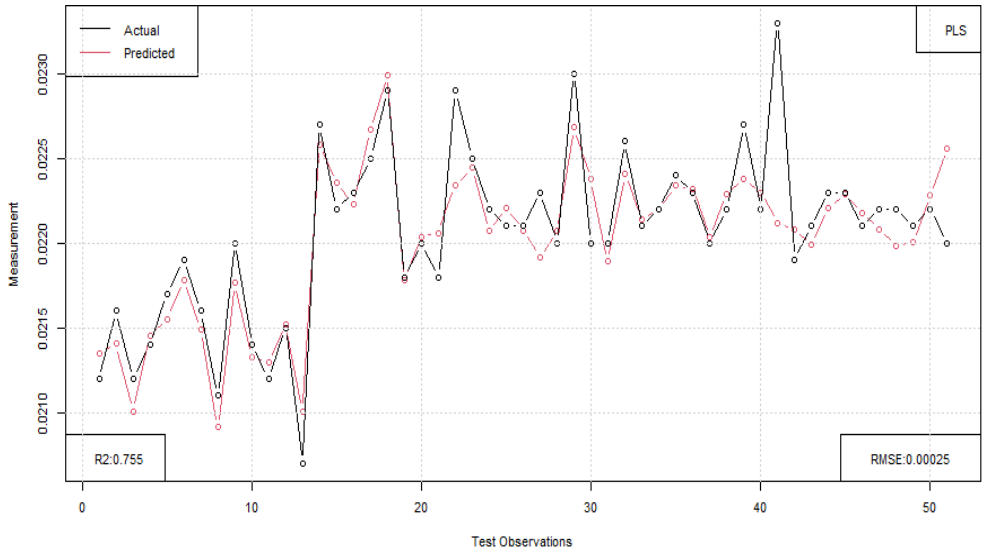

Figure 3.21 : Prediction vs Actual values of QC_21

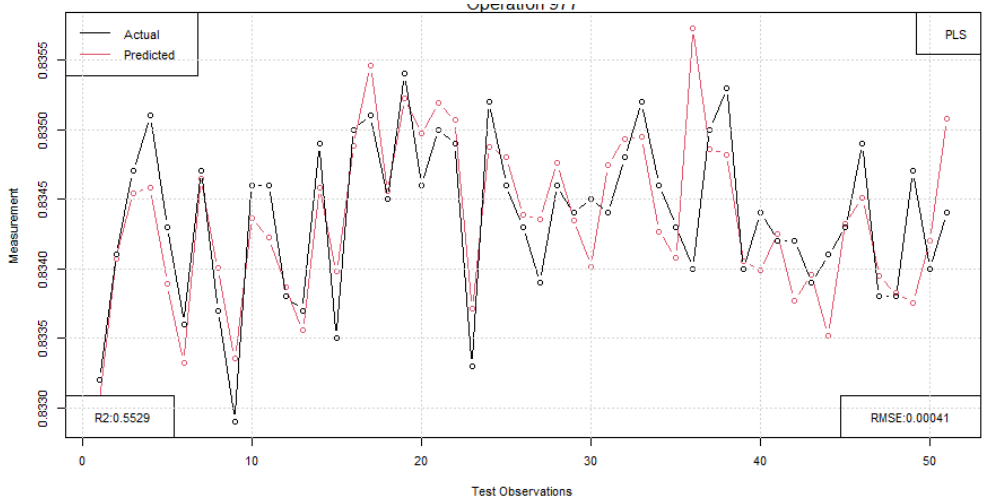

Figure 3.22 : Prediction vs Actual values of QC_20

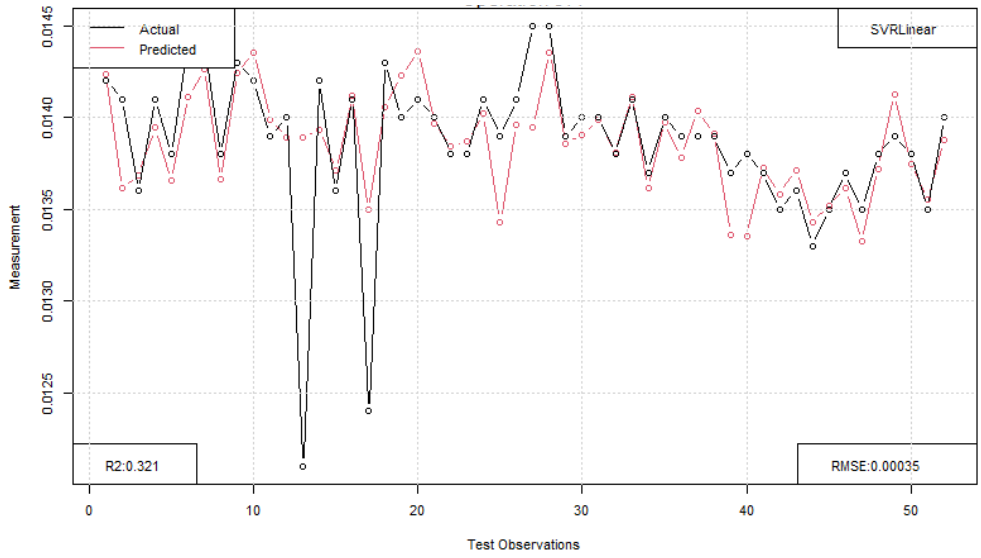

Figure 3.23 : Prediction vs Actual values of QC_36

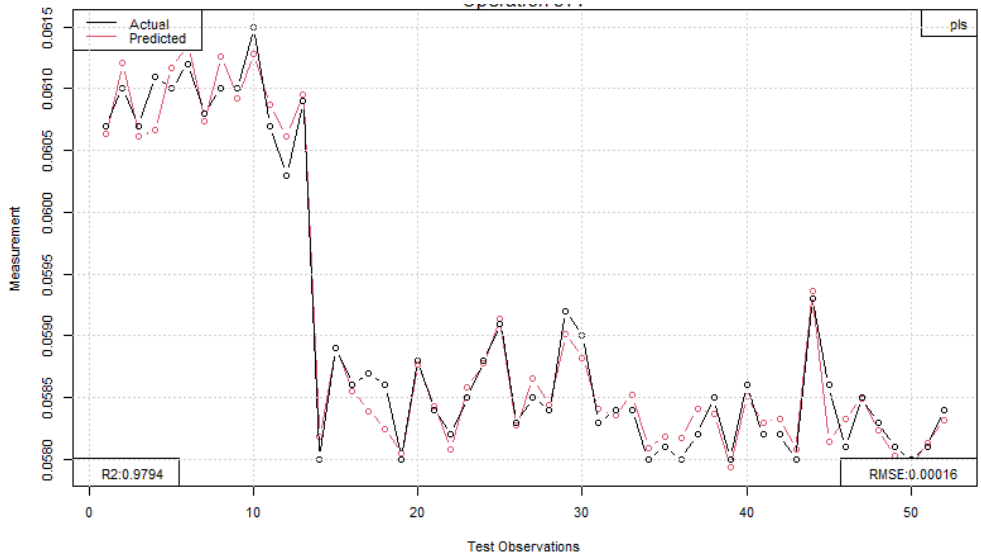

Figure 3.24 : Prediction vs Actual values of QC_42

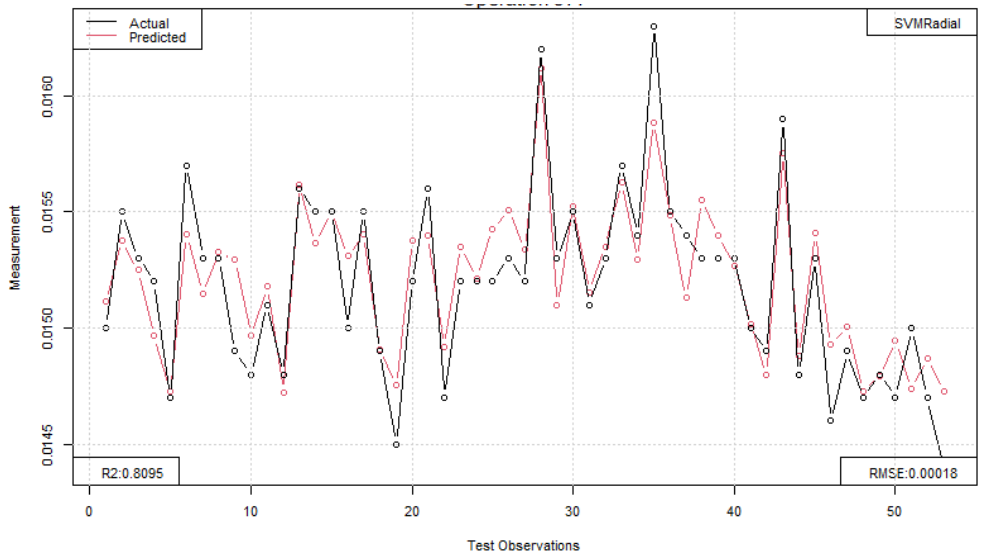

Figure 3.25 : Prediction vs Actual values of QC_31

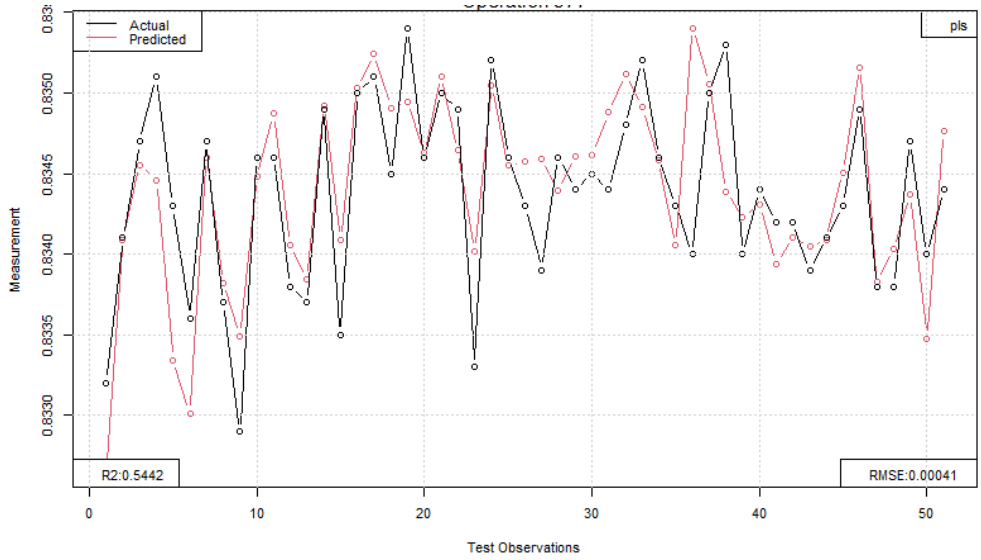

Figure 3.26 : Prediction vs Actual values of QC_20

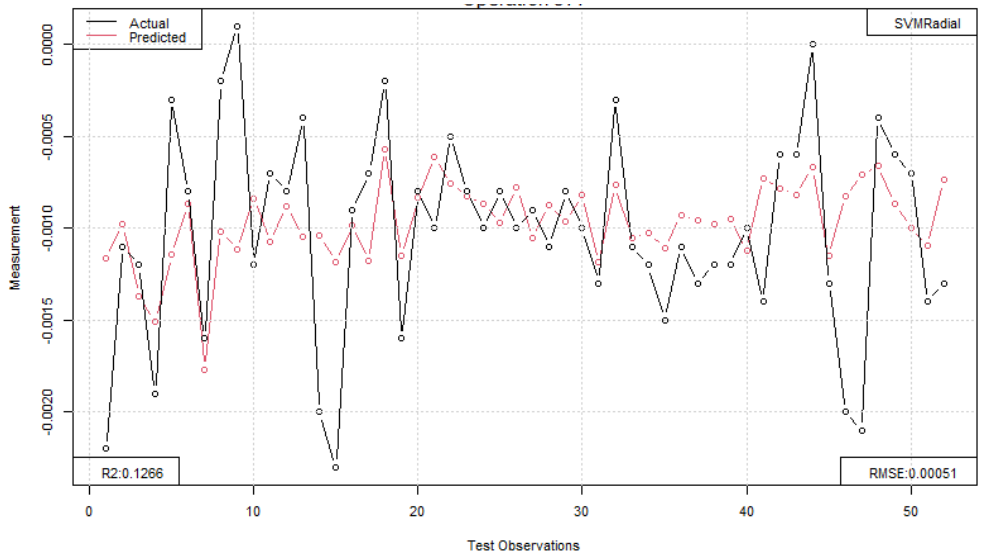

Figure 3.27: Prediction vs Actual values of QC_1

3.4.3 Selected Algorithms

As commented in section 3.3.4, every QC to be predicted at each stage has its own prediction model. That means that we have 168 models (56 QC multiplied by 3 modeling approaches). If a specific algorithm was selected for a given QC at a given stage, it does not necessarily mean that the other algorithms don't perform well at predicting that QC. It just means that the selected algorithm performed better than the rest when predicting the test set.

In this subsection, we present the frequency at which each algorithm was selected as the best predictor for a given quality characteristic (QC) across the three different modeling approaches. Figure 3.28 illustrates the selected algorithms for predicting Y_{q+1}using X_q as independent variables, labeled as $\text{Models}_{q,q+1}$ 1. $\text{Models}_{q+1,Q}$ depicts the algorithm selection for predicting Y_Q using X_{q+1} as independent variables. Finally, $\text{Models}_{q,Q}$ shows the algorithm selection for predicting Y_Q using X_q as independent variables. In the same way table 3.10 shows in detail the frequency of each algorithm selection across the three modeling approaches.

Table 3.10 : Detail of the frequency at which each algorithm was selected as the best predictor for a given QC

Algorithm	**$\text{Models}_{q,q+1}$**	**$\text{Models}_{q+1,Q}$**	**$\text{Models}_{q,Q}$**	**Total**	**%Percentage of the total**
KNN	2	2	6	10	6%
PCR	8	15	10	33	20%
PLS-Regression	16	13	17	46	27%
RF	5	5	2	12	7%
SVMLinear	14	8	14	36	21%
SVMRBF	11	13	7	31	18%

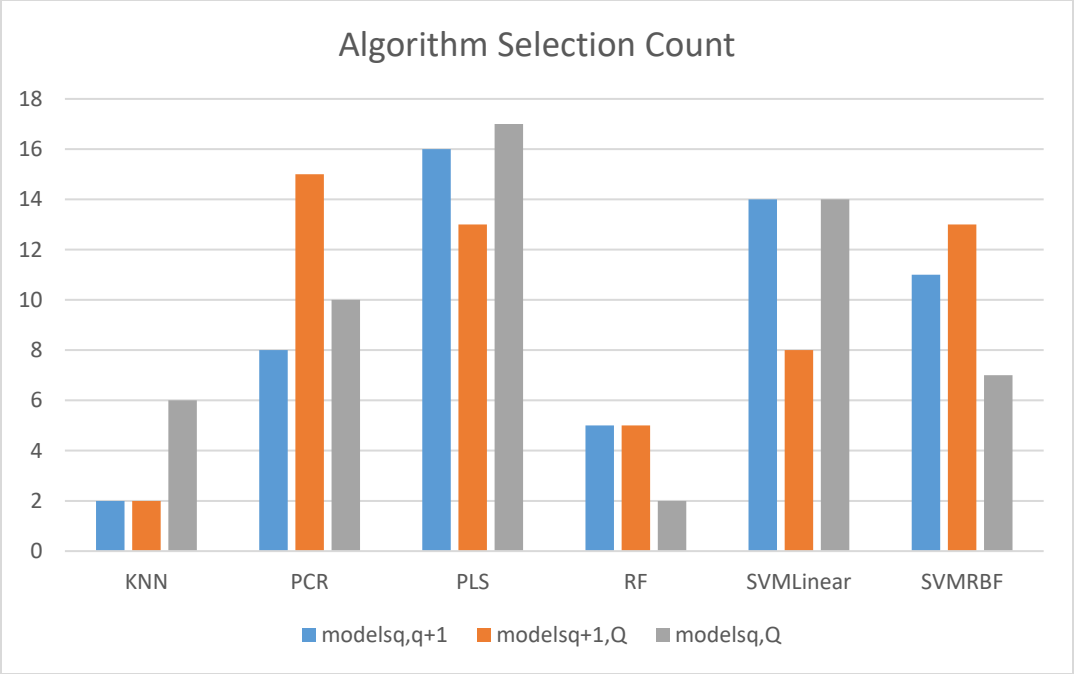

Figure 3.28 : Algorithm selection count

Overall, Partial Least Squares Regression (PLS) was the most predominant algorithm, being selected 46 times out of 168 (27%), followed by Support Vector Machines with linear kernel (SVMLinear), which was selected 36 times out of 168 (21%).

PLS Regression shows high R^2 values and small errors in handling datasets with multicollinearity, where the number of variables exceeds the number of observations, and with limited data, which aligns well with the characteristics of our datasets. According to the literature, PLS is particularly suitable for situations where these conditions are present, making it a strong candidate for our modeling needs.

SVMLinear, the second most predominant algorithm, likely performed well due to its ability to handle linear relationships effectively. This suggests that many of the relationships between the input data (independent variables) and the output data (dependent variables) in our manufacturing process might be linear or nearly linear, making SVMLinear an appropriate choice.

Principal Component Regression (PCR) was selected 33 times (20%). PCR is advantageous in reducing the dimensionality of the data while retaining most of the variability, which helps in handling multicollinearity and improving the stability and interpretability of the regression models.

3.5 Case study's Conclusion

From figures 3.15 to 3.26, we observe that the lower the coefficient of determination (R^2), the less precise the predictions are compared to the actual data.

After reviewing the charts of the predicted versus the real values of all the QCs, models with an R^2 greater than 0.60 visually present predictions close to the actual values. 82%, 84%, and 80% of the predicted characteristics for $models_{q,q+1,}$ $models_{q+1,Q}$ and $models_{q,Q}$ respectively, have a R^2 greater than 0.60. This indicates that most of the models can be used for prediction. However, some exceptions were observed, where models with R^2 below 0.60 demonstrated low *RMSE*, *%EV*, and *%ET* values. Also, their predicted values closely mirrored the real ones in their graphs, meaning that the proportion mentioned earlier could increase, depending on the specific needs and quality requirements. It is always advisable to consider all the metrics and quality requirements to determine if the error generated by any model is acceptable.

Table 3.11 shows a comparison of the metrics' averages between $models_{q,Q}$ and $models_{q+1,Q.}$The data is taken from the previous tables 3.8 and 3.9. We observe that $models_{q+1,Q}$ outperforms models $models_{q,Q}$. This is expected, as stage q+1 is closer to Q, and thus, it is reasonable to anticipate less variation due to the manufacturing process. The measurements from stage q+1 capture more recent information, resulting in improved predictive performance. However, the metrics for $models_{q,Q}$ show potential usefulness in predicting in advance the QCs of stage Q.

Table 3.11 : comparison of the metrics' averages between $models_{q,Q}$ and $models_{q+1,Q.}$

Metric's Average	**$Models_{q+1,Q}$**	**$Models_{q,Q}$**
RMSE	0.00026233	0.00032703
R^2	0.8043	0.7497
MAE	0.00019347	0.00024423
%EV	5.00%	5.80%
%ET	1.80%	2.20%

The characteristics for which prediction performance was unsatisfactory can be explained by several factors. None of the tested models, whether distance-based, linear, or tree-based provided satisfactory results for the characteristics mentioned at the bottom of Tables B.1 to B.3 from

Appendix B. These characteristics may present a complexity that the models failed to capture. Models such as neural networks, XGBoost, and deep learning, in general, have the capacity to capture such complex relationships but require more training data since they have more parameters to be estimated. When data is limited, overfit is likely to happen.

Another cause could be a lack of information on these characteristics in the earlier stages of the production process. A detailed analysis of the characteristics in question may reveal the need to collect additional information from in-process inspection from earlier stages of the process.

Additionally, it is possible that the problematic QC showing low R^2 values could be explained by production process parameters rather than characteristics from previous stages.

Finally, it is important to consider the possibility that the measurement system used to collect data may not be robust enough for certain QCs. Noise from variations in the measurement process can affect data quality and make it difficult to accurately predict the QC concerned.

CHAPTER 4 GENERAL CONCLUSION AND RECOMMENDATIONS

4.1 General Conclusion

This study has demonstrated the feasibility of using AI models to predict quality characteristics (QCs) in a multi-stage manufacturing system (MMS) without relying on controllable and uncontrollable process data, such as sensor readings and process parameters. By leveraging in-process inspection data obtained from Coordinate Measure Machines (CMM) at various intermediate stages, we were able to predict the quality of airplane engine parts across multiple stages of production. Furthermore, algorithms such as PLS-Regression and PCA allowed us to overcome the challenge of high dimensionality and limited quantity of data, in a scenario where we have more independent variables than observations.

One advantage of this approach is the ability to predict QCs for the next consecutive stage ($q+1$) using data from the current stage (q). This enables timely adjustments and proactive actions to ensure that quality specifications are maintained as parts progress through each stage of the manufacturing process. By identifying potential quality deviations early, it is possible to intervene and correct the produced QC's values to save the parts from being scrapped, thus improving overall productivity and reducing waste.

Furthermore, the ability to predict QCs for the final stage Q using data from earlier stages (such as q or $q+1$) provides an even greater strategic advantage. This long-term forecasting capability allows manufacturers to anticipate the final quality of parts well in advance, facilitating better planning and resource allocation. It enables a more comprehensive understanding of how early-stage processes impact final parts' quality, thus allowing for more informed decision-making and optimization of the entire manufacturing workflow.

Overall, these predictive capabilities enhance the robustness and reliability of the manufacturing process, ensuring high-quality outcomes and greater operational efficiency without the need for extensive process data. This makes it more applicable to other industries where sensor implementation is limited, costly, or not feasible, such as traditional manufacturing, legacy production systems, or small-scale industries. Additionally, the approach is scalable across industries with multi-stage manufacturing processes such as automotive, electronics,

pharmaceuticals and any other sector where quality characteristics can be measured at various checkpoints of the MMS.

The study also highlighted the performance of various algorithms in prediction models. Partial Least Squares Regression (PLS) emerged as the most predominant algorithm, being selected frequently due to its ability to handle multicollinearity and linear relationships effectively, which are common in high-dimensional manufacturing data. This aligns with findings in literature, where PLS and kernel-based methods are often preferred for their robustness in similar contexts. Additionally, Support Vector Machines (SVM) with radial base functions were noted for their capacity to manage nonlinear problems, reinforcing their suitability for complex manufacturing environments.

These results suggest that leveraging PLS and SVM, along with exploring advanced deep learning techniques, may enhance the predictive accuracy and reliability in multi-stage manufacturing processes.

The proposed method in this thesis has strong potential for real-time quality monitoring. In multi-stage manufacturing systems (MMS), where CMM inspections are time-consuming and parts often wait hours between stages, predictions can be generated during these delays without the need to be strictly instantaneous. This allows for timely interventions to prevent defects before parts move to the next stage. While managing 168 models requires computational efficiency and maintenance, the approach aligns with production timelines, leveraging waiting periods for processing.

Integration with control systems ensures accurate and proactive data driven decision-making, avoiding relying solely on workers' experience. By implementing our approach, the decision of intervention is assessed based on the prediction models' outputs, rather than subjective judgment. This approach ensures more consistent and reliable decisions, minimizes variability introduced by unnecessary interventions, and improves overall process stability and product quality. Ultimately, predictive modeling provides a data-driven framework that reduces the risks associated with worker turnover or retirement, workers' training, and inconsistent judgments.

However, this study is not without limitations. The primary constraint was the availability of data from only four controlled batches for model training, potentially limiting the representation of the true variability inherent in the production process. Additionally, the absence of process data, while beneficial in reducing the need for additional investments in sensors and other resources, may have

limited the models' ability to capture certain complex relationships within the production process. As demonstrated in the results section, model accuracy decreases as the prediction target moves further along with the production stages. This decline occurs because variability accumulates in the QCs as they pass through each stage. When stages are far apart, the prediction task becomes challenging, and in some cases, nearly impossible such as predicting final inspection QCs using data from the very first stage. However, certain QCs that undergo minimal changes during the production process may still be predicted with reasonable accuracy. Finally, our approach relies on the presence of quality inspections at intermediate stages. For MMS with continuous processes, such as the extrusion process described in [13], where there is no clear separation between stages, this approach would either be infeasible or complex to implement.

4.2 Recommendations for Future Work

- Expand Data Collection: Future work should prioritize obtaining more data from additional batches to improve the models' ability to generalize and accurately represent the variability in the production process. This could involve increasing the number of controlled runs and ensuring a diverse set of data points across different production conditions.
- Explore Alternative Algorithms: While the study focused on specific algorithms, future research should explore alternative deep learning algorithms. It is important to clarify that the exploration of deep learning is not based on an assumption that it will automatically outperform the current methodology. Rather**,** with a larger dataset, it would be worthwhile investigating deep learning approaches because of their ability to model complex, non-linear relationships. Additionally, deep learning has the capability to predict multiple responses simultaneously, which would facilitate the prediction of all quality characteristics at different stages of the multi-stage manufacturing system with just 3 models instead of 168.
- An alternative modeling approach could incorporate QCs from both stage q and stage q+1 as input variables to predict the QCs of stage Q. By combining information from two stages, it may be possible to improve the performance of the regression models when predicting the QCs at stage Q.
- Integrate Process Data: Although this study successfully predicted QCs without using process data, integrating such data into future models could enhance prediction accuracy and provide a more comprehensive understanding of the production process. Process parameters and sensor readings could offer additional insights into the factors influencing QC outcomes.
- Apply Ensemble Modeling: Ensemble modeling involves combining predictions from multiple models to improve accuracy and robustness. By leveraging various algorithms and combining their outputs, ensemble models can often outperform single models, especially when different algorithms capture different aspects of the data. Future work should investigate ensemble methods, such as bagging, boosting, or stacking, to improve the

prediction accuracy of quality characteristics in the multi-stage manufacturing process. These models could also help mitigate the weaknesses of individual models and better capture the variability in the production process.

- Investigate Measurement System Robustness: Analyzing the robustness of the measurement systems used in inspections can help identify potential sources of noise and variability in the data. Ensuring high-quality measurement systems can improve the accuracy and reliability of the data used for predictions.
- Conduct Detailed Feature Analysis: A more in-depth analysis of the specific features and characteristics that contribute to prediction performance could reveal additional areas for improvement. This might involve identifying and collecting new features that could enhance model accuracy.
- Finally, any quality characteristics (QC) that were not satisfactorily predicted could be prioritized for applying these recommendations and future work to further improve predictive capabilities.

By addressing these recommendations, future research can build on the findings of this study to further improve the prediction of quality characteristics in multi-stage manufacturing systems, ultimately contributing to more efficient and effective quality control processes.

REFERENCES

[1] A. Mitra, *Fundamentals of Quality Control and Improvement*. John Wiley & Sons, 2016.

[2] R. Srinivasu, G. S. Reddy, and S. R. Rikkula, "UTILITY OF QUALITY CONTROL TOOLS AND STATISTICAL PROCESS CONTROL TO IMPROVE THE PRODUCTIVITY AND QUALITY IN AN INDUSTRY," 2009.

[3] J. Shi and S. Zhou, "Quality control and improvement for multistage systems: A survey," *IIE Transactions*, vol. 41, no. 9, pp. 744–753, Jul. 2009, doi: 10.1080/07408170902966344.

[4] R. S. Peres, J. Barata, P. Leitao, and G. Garcia, "Multistage Quality Control Using Machine Learning in the Automotive Industry," *IEEE Access*, vol. 7, pp. 79908–79916, 2019, doi: 10.1109/ACCESS.2019.2923405.

[5] M. Ismail, N. A. Mostafa, and A. El-assal, "Quality monitoring in multistage manufacturing systems by using machine learning techniques," *J Intell Manuf*, vol. 33, no. 8, pp. 2471–2486, Dec. 2022, doi: 10.1007/s10845-021-01792-1.

[6] Q. Liu, V. S. Vassiliadis, Y. Wu, J. Zhang, C. Cheng, and Y. Yuan, "A Quality Prediction and Parameter Optimization Approach for Turbine Blade Multistage Manufacturing," *IEEE/ASME Transactions on Mechatronics*, pp. 1–13, 2023, doi: 10.1109/TMECH.2023.3260884.

[7] F. Eger *et al.*, "Part Variation Modeling to Avoid Scrap Parts in Multi-stage Production Systems," *Procedia CIRP*, vol. 107, pp. 851–856, Jan. 2022, doi: 10.1016/j.procir.2022.05.074.

[8] D. Zhang, Z. Liu, W. Jia, H. Liu, and J. Tan, "Path Enhanced Bidirectional Graph Attention Network for Quality Prediction in Multistage Manufacturing Process," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1018–1027, Feb. 2022, doi: 10.1109/TII.2021.3076803.

[9] D. Zhang, Z. Liu, W. Jia, H. Liu, and J. Tan, "Contrastive Decoder Generator for Few-shot Learning in Product Quality Prediction," *IEEE Transactions on Industrial Informatics*, pp. 1–12, 2022, doi: 10.1109/TII.2022.3190554.

[10] M. Hodnett, J. F. Wiley, Y. (Hayden) Liu, and P. Maldonado, *Deep Learning with R for Beginners: Design Neural Network Models in R 3. 5 Using TensorFlow, Keras, and MXNet*. Birmingham, UNITED KINGDOM: Packt Publishing, Limited, 2019. Accessed: Jul. 05, 2023. [Online]. Available: http://ebookcentral.proquest.com/lib/polymtl-ebooks/detail.action?docID=5778833

[11] G. Köksal, İ. Batmaz, and M. C. Testik, "A review of data mining applications for quality improvement in manufacturing industry," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13448–13467, Sep. 2011, doi: 10.1016/j.eswa.2011.04.063.

[12] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *WIREs Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015, doi: 10.1002/widm.1157.

[13] V. García, J. S. Sánchez, L. A. Rodríguez-Picón, L. C. Méndez-González, and H. de J. Ochoa-Domínguez, "Using regression models for predicting the product quality in a tubing extrusion process," *J Intell Manuf*, vol. 30, no. 6, pp. 2535–2544, Aug. 2019, doi: 10.1007/s10845-018-1418-7.

[14] S. Guha and A. Alaeddini, "Predictive Model for Multi-Stage Manufacturing Using Nonlinear Partial Least Square Methods," *IIE Annual Conference. Proceedings*, pp. 662–669, 2015, Accessed: Apr. 26, 2023. [Online]. Available: https://www.proquest.com/docview/1791989214/abstract/63A587EAE93D44FAPQ/1

[15] S. Wold, N. Kettaneh-Wold, and B. Skagerberg, "Nonlinear PLS modeling," *Chemometrics and Intelligent Laboratory Systems*, vol. 7, no. 1, pp. 53–65, Dec. 1989, doi: 10.1016/0169-7439(89)80111-X.

[16] X. Yin, Z. He, Z. Niu, and Z. (Steven) Li, "A hybrid intelligent optimization approach to improving quality for serial multistage and multi-response coal preparation production systems," *Journal of Manufacturing Systems*, vol. 47, pp. 199–216, Apr. 2018, doi: 10.1016/j.jmsy.2018.05.006.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[18] B.-H. Mevik and R. Wehrens, “Introduction to the pls Package”.

[19] H. Abdi, “Partial Least Squares (PLS) Regression.”.

[20] C.-C. Yang and M.-D. Shieh, “A support vector regression based prediction model of affective responses for product form design,” *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 682–689, Nov. 2010, doi: 10.1016/j.cie.2010.07.019.

[21] S. Lee, P. Kang, and S. Cho, “Probabilistic local reconstruction for *k*-NN regression and its application to virtual metrology in semiconductor manufacturing,” *Neurocomputing*, vol. 131, pp. 427–439, May 2014, doi: 10.1016/j.neucom.2013.10.001.

[22] M. R. Segal, “Machine Learning Benchmarks and Random Forest Regression,” Apr. 2004, Accessed: May 16, 2024. [Online]. Available: https://escholarship.org/uc/item/35x3v9t4

[23] U. Talukdar, S. M. Hazarika, and J. Q. Gan, “A Kernel Partial least square based feature selection method,” *Pattern Recognition*, vol. 83, pp. 91–106, Nov. 2018, doi: 10.1016/j.patcog.2018.05.012.

[24] D. C. Montgomery, *Introduction to statistical quality control*, 6th ed. Hoboken, N.J.: John Wiley & Sons, Inc., 2009.

[25] R Core Team, *R: A Language and Environment for Statistical Computing*. (2022). R Foundation for Statistical Computing, Vienna, Austria. [Online]. Available: https://www.R-project.org/

[26] Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan, *dplyr: A Grammar of Data Manipulation*. (2023). [Online]. Available: https://CRAN.R-project.org/package=dplyr

[27] Hadley Wickham, *ggplot2: Elegant Graphics for Data Analysis*. (2016). New York. [Online]. Available: https://ggplot2.tidyverse.org

[28] R. J. Perla, L. P. Provost, and S. K. Murray, “The run chart: a simple analytical tool for learning from variation in healthcare processes,” *BMJ Quality & Safety*, vol. 20, no. 1, pp. 46–51, Jan. 2011, doi: 10.1136/bmjqs.2009.037895.

[29] A. V. Tatachar, “Comparative Assessment of Regression Models Based On Model Evaluation Metrics,” vol. 08, no. 09, 2021.

[30] Kuhn and Max, “Building Predictive Models in R Using the caret Package,” *Journal of Statistical Software*, pp. 1–26, 2008, doi: 10.18637/jss.v028.i05.

[31] E. Pekel, “Estimation of soil moisture using decision tree regression,” *Theor Appl Climatol*, vol. 139, no. 3, pp. 1111–1119, Feb. 2020, doi: 10.1007/s00704-019-03048-8.

APPENDIX A DECISION TREE REGRESSION

Decision Tree Regression is a non-parametric supervised learning method used for predicting continuous outcomes. It models the data by recursively splitting the feature space of independent variables X into distinct regions based on certain criteria, resulting in a tree-like structure. The goal is to create a model that predicts the dependent variable $\mathcal{Y}$ by learning simple decision rules inferred from the data features. X is the matrix of independent variables with p variables and n observations and y is the dependent variable with dimension $1 \times n$.

A decision tree T consists of nodes and edges. Each internal node represents a decision based on a feature X_p, and each leaf node represents a predicted value of the dependent variable $\mathcal{Y}$.

The tree is built by recursively splitting the data at each node based on the feature that results in the best split. The quality of a split is typically measured using metrics such as Mean Squared Error (*MSE*), which aims to minimize the variance within each split.

The algorithm starts with the entire dataset at the root node and partitions it into subsets that contain similar values of the dependent variable. This process is repeated recursively for each derived subset.

To determine the best split at each node, the algorithm evaluates all possible splits across all features $\mathcal{X}_p$. The goal is to minimize the *MSE* of the dependent variable $\mathcal{Y}$ within the partitions and the optimal split is found by choosing the feature and threshold that minimizes the weighted average of the *MSE* for the resulting subsets:

$$MSE_{split} = \frac{n_{left}}{n} MSE_{left} + \frac{n_{right}}{n} MSE_{right}$$

Where n_{left} and n_{right} are the numbers of samples in the left and right subsets after the split, respectively.

The construction of a decision tree involves 4 main steps:

1. Initialization: Start with the root node containing the entire dataset.
2. Splitting: For each node, find the best feature X_p and threshold l_j that minimize the *MSE* for the split. Then, split the node into two child nodes: left (samples with $X_p \leq l_j$) and right (samples with $X_p > l_j$). j represents the j-th node. p= 1, .., P and j= 1, …, J.
3. Recursion: Repeat the splitting process recursively for each child node until a stopping criterion is met.

4. Prediction: For a new observation, traverse the tree from the root to a leaf node by following the decision rules. The prediction for the observation is the mean value of *y* in the leaf node.

The growth of the tree can be controlled and stop to prevent overfitting by using the following parameters:

Maximum Depth: Limits the depth of the tree. A shallower tree reduces overfitting but may underfit the data.

Minimum Samples per Split: The minimum number of samples required to split an internal node. Higher values prevent splits that do not contribute significantly to model improvement.

Minimum Samples per Leaf: The minimum number of samples required to be at a leaf node. Ensures that leaves have enough samples to provide reliable predictions.

Minimum Impurity Decrease: A node will be split if the split induces a decrease of the impurity greater than or equal to this value[31].

APPENDIX B PREDICTIVE MODELS' DETAIL

Tables B.1, B.2, and B.3 present the prediction results in the test data for the three modeling approaches. The first column indicates the selected algorithm, the second the name of the QC, and the third, fourth, and fifth columns show the *RMSE* values, the coefficient of determination R^2, *MAE*, the *%EV*, and the *%ET* respectively, generated by the model in the test data. The content of the tables is sorted in descending order by R^2. Each table has 56 models, one for each QC to predict.

Table B.1: Details of $Models_{q,q+1}$

Algorithm	QC	*RMSE*	R^2	*MAE*	*%EV*	*%ET*
PCR	QC_53	0.00015386	0.9730143	0.00012292	4%	2%
PCR	QC_42	0.00025893	0.95257834	0.00017877	4%	2%
PCR	QC_40	0.00019834	0.94786368	0.00015184	4%	2%
PCR	QC_52	0.00011072	0.93164042	8.98E-05	6%	1%
SVMLinear	QC_49	0.00022546	0.93128364	0.00017194	5%	2%
SVMLinear	QC_43	0.00019533	0.92850544	0.00013527	4%	2%
PLS	QC_45	0.00019217	0.9276364	0.00013424	4%	2%
SVMLinear	QC_44	0.00018579	0.91746122	0.00014034	4%	2%
SVMLinear	QC_23	0.00015467	0.9163629	0.00012356	4%	2%
RF	QC_50	0.00028317	0.916335	0.00020069	5%	3%
PCR	QC_46	0.00017739	0.91614591	0.00013149	5%	2%
KNN	QC_54	0.00014856	0.89337845	0.00011716	5%	2%
PLS	QC_15	0.00040963	0.89207895	0.00027147	4%	1%
PCR	QC_14	0.00036382	0.8889337	0.00028267	6%	1%
SVMLinear	QC_9	0.00063705	0.88672941	0.00049209	5%	4%
SVMRBF	QC_26	0.00021191	0.88295326	0.00017438	6%	2%
PLS	QC_24	0.00023531	0.87839407	0.0001767	5%	2%
PLS	QC_16	0.00039038	0.87749104	0.00028331	4%	1%
PLS	QC_13	0.00033762	0.87613536	0.00023321	4%	1%
SVMLinear	QC_7	0.00072504	0.87501594	0.00055619	6%	5%
PLS	QC_12	0.00032382	0.85616336	0.00025695	5%	1%
SVMRBF	QC_22	0.00019514	0.85156945	0.00014352	5%	2%
PLS	QC_30	0.00021008	0.85118747	0.00015571	6%	2%
SVMLinear	QC_29	0.00022919	0.84655141	0.00017394	5%	2%
SVMRBF	QC_27	0.00023509	0.82940231	0.00017635	6%	2%
SVMLinear	QC_6	0.00065583	0.82111457	0.00051221	7%	4%
SVMLinear	QC_10	0.00084244	0.81966537	0.00067847	7%	6%
SVMRBF	QC_47	0.00018438	0.8175903	0.00013369	5%	2%

Table B.1: Details of $Models_{q,q+1}$ (continuation and end)

Algorithm	**QC**	***RMSE***	***R^2***	***MAE***	***%EV***	***%ET***
SVMLinear	QC_8	0.00072525	0.81126381	0.00055727	6%	5%
PLS	QC_37	0.00017017	0.80441322	0.00013106	6%	2%
SVMRBF	QC_25	0.00025329	0.78522994	0.00017239	6%	2%
PLS	QC_17	0.00035226	0.78233684	0.00027976	4%	1%
SVMLinear	QC_19	0.00029428	0.77774459	0.0002346	5%	1%
PLS	QC_55	0.00013356	0.77684709	0.0001039	6%	2%
SVMRBF	QC_48	0.00021085	0.77589546	0.00015873	6%	2%
PLS	QC_11	0.00033352	0.74396296	0.00023303	3%	1%
PCR	QC_31	0.00025883	0.74177338	0.00020708	5%	3%
SVMRBF	QC_28	0.00024106	0.73570991	0.00019107	6%	2%
RF	QC_32	0.00032885	0.73428629	0.00019597	7%	2%
PLS	QC_20	0.00038892	0.7091867	0.00027848	7%	1%
SVMLinear	QC_34	0.00021463	0.69182684	0.00016707	7%	2%
SVMLinear	QC_35	0.00020903	0.67975857	0.00014975	6%	2%
SVMRBF	QC_51	0.00014211	0.66788628	0.00010068	8%	2%
RF	QC_33	0.00028242	0.66015189	0.00016873	6%	2%
PCR	QC_18	0.00034136	0.6506566	0.0002864	5%	1%
PLS	QC_39	0.00027462	0.61650477	0.00019862	7%	2%
PLS	QC_56	0.00022749	0.57672924	0.00017015	8%	3%
RF	QC_4	0.00046096	0.56657929	0.00036887	8%	3%
SVMRBF	QC_41	0.00031872	0.50929757	0.00022982	7%	3%
PLS	QC_38	0.0002837	0.5048471	0.00019533	8%	2%
PLS	QC_36	0.00028444	0.42472657	0.0001915	8%	2%
SVMRBF	QC_3	0.00056224	0.16051985	0.00043022	10%	4%
RF	QC_21	0.00077819	0.0973473	0.0004367	7%	5%
SVMLinear	QC_1	0.000514	0.0514	0.000369	13%	3%
SVMRBF	QC_2	0.0003619	0.042	0.0002759	8%	2%
KNN	QC_5	0.00047389	0.00176	0.0003659	12%	3%

Table B.2: Details of Models$_{q+1,Q}$

Algorithm	**QC**	***RMSE***	***R^2***	***MAE***	***%EV***	***%ET***
PCR	QC_9	0.0002187	0.98965855	0.00017114	2%	1%
PLS	QC_8	0.00019391	0.98941872	0.00015824	2%	1%
PCR	QC_7	0.00024594	0.98785187	0.0001839	2%	1%
PLS	QC_10	0.00024258	0.98523364	0.00020038	2%	2%
PLS	QC_6	0.00022987	0.97484198	0.00017552	2%	1%
PCR	QC_42	0.00019033	0.97254843	0.00015579	4%	2%
PLS	QC_53	0.00015293	0.96393954	0.00012132	4%	2%
SVMRBF	QC_50	0.00019107	0.95700786	0.00013477	4%	2%
PCR	QC_45	0.0001661	0.94848923	0.00013502	4%	2%
SVMRBF	QC_43	0.00015482	0.94392178	0.00012192	4%	2%
SVMRBF	QC_49	0.00018415	0.94258392	0.0001436	3%	2%
SVMLinear	QC_44	0.00016616	0.93251336	0.00012807	4%	2%
PCR	QC_14	0.00044164	0.92662409	0.00035745	5%	2%
PCR	QC_3	0.00020355	0.92138767	0.00016695	4%	1%
PCR	QC_56	0.00016101	0.91482344	0.00012619	5%	2%
PLS	QC_29	0.00016851	0.91320588	0.00014542	5%	2%
PLS	QC_23	0.00017298	0.90816801	0.00013919	5%	2%
PLS	QC_25	0.00014749	0.90806026	0.00011419	5%	1%
PCR	QC_15	0.00052711	0.90354506	0.00037762	4%	2%
KNN	QC_40	0.00028931	0.90181784	0.00021933	5%	3%
PCR	QC_30	0.0001907	0.89654063	0.00015947	6%	2%
SVMRBF	QC_24	0.00017299	0.88648443	0.00013311	5%	2%
SVMLinear	QC_5	0.00017241	0.87973669	0.00011082	3%	1%
SVMRBF	QC_17	0.000431	0.87830318	0.00033245	4%	2%
PCR	QC_28	0.00018971	0.87680314	0.00015211	6%	2%
PLS	QC_16	0.00057603	0.87634326	0.00036947	4%	2%
SVMRBF	QC_46	0.00018951	0.86866028	0.0001533	5%	2%
SVMRBF	QC_1	0.00020003	0.86726848	0.00014894	5%	1%
PLS	QC_12	0.00050992	0.86485348	0.00040159	6%	2%
SVMRBF	QC_54	0.00014381	0.85027347	0.00011438	5%	2%
PCR	QC_32	0.00019779	0.8492111	0.00014755	4%	2%
RF	QC_55	0.00012681	0.83699532	0.00010369	4%	2%
PLS	QC_22	0.00018743	0.83633486	0.00013994	5%	2%
SVMLinear	QC_2	0.00020286	0.8293524	0.00015044	4%	1%
PCR	QC_27	0.00018859	0.82768392	0.00015391	6%	2%
SVMLinear	QC_39	0.00018653	0.79446831	0.00014767	5%	2%
SVMLinear	QC_4	0.00035975	0.79214828	0.00018483	4%	1%
SVMRBF	QC_11	0.00053018	0.79007871	0.00039949	5%	2%

Table B.2: Details of Models$_{q+1,Q}$ (continuation and end)

Algorithm	QC	*RMSE*	R^2	*MAE*	*%EV*	*%ET*
PCR	QC_47	0.00015012	0.78020166	0.00012123	5%	2%
PLS	QC_21	0.00025288	0.75503536	0.00016716	3%	2%
SVMLinear	QC_13	0.00073259	0.73951212	0.00049922	7%	2%
KNN	QC_26	0.00020885	0.72312168	0.00016204	6%	2%
RF	QC_33	0.00025003	0.66996336	0.00018114	7%	2%
PCR	QC_41	0.00023003	0.64254408	0.00018291	8%	2%
RF	QC_52	0.00019422	0.64202544	0.00011487	7%	2%
RF	QC_48	0.00023395	0.63553845	0.0001835	8%	2%
SVMRBF	QC_18	0.00048758	0.61876086	0.00035026	4%	2%
SVMRBF	QC_34	0.00025088	0.59624722	0.00019698	7%	2%
SVMRBF	QC_31	0.00025483	0.57860093	0.00018071	6%	2%
RF	QC_51	0.00013566	0.55537964	0.00010675	9%	2%
PLS	QC_20	0.00040734	0.55289649	0.00029872	6%	1%
SVMLinear	QC_37	0.00020613	0.55185573	0.00014426	7%	2%
PLS	QC_19	0.00057596	0.54695715	0.0003845	7%	2%
SVMRBF	QC_35	0.00026764	0.47693429	0.00018544	7%	2%
PCR	QC_38	0.00030166	0.46783013	0.00020473	9%	3%
SVMLinear	QC_36	0.00034595	0.32098906	0.00019065	7%	2%

Table B.3: Details of Models$_{q,Q}$

Algorithm	**QC**	***RMSE***	$\boldsymbol{R^2}$	***MAE***	***%EV***	***%ET***
PLS	QC_42	0.00016378	0.97940568	0.00012723	3%	2%
PCR	QC_45	0.00013136	0.96991022	9.75E-05	3%	1%
PLS	QC_53	0.0001596	0.96052763	0.00012507	4%	2%
SVMLinear	QC_49	0.00014687	0.95987538	0.00011574	3%	1%
SVMLinear	QC_43	0.00015009	0.94622374	0.00010779	3%	1%
SVMLinear	QC_50	0.00022391	0.93611749	0.00015664	4%	2%
PCR	QC_56	0.00014524	0.92970258	0.00011599	5%	2%
KNN	QC_44	0.00016729	0.9271967	0.00014052	5%	2%
PLS	QC_16	0.00045025	0.9239764	0.00032142	3%	2%
SVMLinear	QC_40	0.00026882	0.92372026	0.00021902	5%	3%
SVMLinear	QC_46	0.00015415	0.91670676	0.00011548	4%	1%
PCR	QC_30	0.00017457	0.9159376	0.00013696	5%	2%
PLS	QC_17	0.00038869	0.91225395	0.00029292	3%	1%
PLS	QC_23	0.00017188	0.91074769	0.00013986	5%	2%

Table B.3: Details of Models$_{q,Q}$ (continuation)

Algorithm	**QC**	***RMSE***	***R^2***	***MAE***	***%EV***	***%ET***
PLS	QC_14	0.00048953	0.90899192	0.00040629	6%	2%
PCR	QC_24	0.00016054	0.90855938	0.00012339	5%	2%
SVMRBF	QC_29	0.00018993	0.88779393	0.00015268	5%	2%
PCR	QC_15	0.00057169	0.88619687	0.00041393	5%	2%
SVMLinear	QC_7	0.00075809	0.88584881	0.00054723	5%	4%
PLS	QC_9	0.00077698	0.86739458	0.00058328	6%	4%
SVMLinear	QC_8	0.00068939	0.86662625	0.00052452	6%	4%
KNN	QC_25	0.0001819	0.86600035	0.000145	6%	2%
PLS	QC_28	0.00020705	0.85280987	0.00016873	6%	2%
SVMLinear	QC_6	0.00056733	0.84933589	0.00043569	5%	3%
RF	QC_32	0.00019461	0.84070518	0.00015452	5%	2%
PLS	QC_10	0.00081625	0.83765437	0.00063217	7%	5%
SVMLinear	QC_55	0.00012811	0.83386806	0.00010162	4%	2%
RF	QC_27	0.00020234	0.82680035	0.0001575	7%	2%
SVMLinear	QC_34	0.00016782	0.81803612	0.00014179	5%	2%
SVMRBF	QC_31	0.00017692	0.80948854	0.00014137	5%	2%
SVMRBF	QC_13	0.00062804	0.8061767	0.00049877	7%	2%
SVMLinear	QC_39	0.00018545	0.80235196	0.00014454	5%	2%
PLS	QC_41	0.00018014	0.79429883	0.00013422	6%	2%
PLS	QC_22	0.00021176	0.79124713	0.00016587	6%	2%
SVMRBF	QC_48	0.00017967	0.78838652	0.00011965	5%	1%
KNN	QC_12	0.00065169	0.77769014	0.00044856	6%	2%
PLS	QC_21	0.00024175	0.77757479	0.00016808	4%	2%
PCR	QC_26	0.0001889	0.77489691	0.00014946	6%	2%
PLS	QC_11	0.00063369	0.7476983	0.0004795	7%	2%
SVMLinear	QC_47	0.00016585	0.73981515	0.00012086	5%	2%
PCR	QC_54	0.00019516	0.7187828	0.00014373	6%	2%
PLS	QC_19	0.00049432	0.63597734	0.00037275	6%	2%
PCR	QC_18	0.00048605	0.63116249	0.00039504	5%	2%
KNN	QC_52	0.0002042	0.62708834	0.00012941	8%	2%
PLS	QC_35	0.00022931	0.60390579	0.00014968	6%	2%
PCR	QC_33	0.00027691	0.58841587	0.00016886	7%	2%
PCR	QC_37	0.00019711	0.57855906	0.0001472	7%	2%
PLS	QC_20	0.0004118	0.54417932	0.00030222	6%	2%
SVMRBF	QC_51	0.0001348	0.50161363	0.00010715	9%	2%
KNN	QC_4	0.00055851	0.49224959	0.00041005	9%	3%

Table B.3: Details of $Models_{q,Q}$ (continuation and end)

Algorithm	**QC**	***RMSE***	R^2	***MAE***	***%EV***	***%ET***
PLS	QC_38	0.00030533	0.45799088	0.0001851	8%	2%
SVMLinear	QC_36	0.00033155	0.37689455	0.00018021	6%	2%
KNN	QC_3	0.00056912	0.37125798	0.00041723	9%	3%
SVMRBF	QC_1	0.00051323	0.1266054	0.00038763	12%	3%
SVMLinear	QC_5	0.00047946	0.06988174	0.00035684	11%	3%
SVMRBF	QC_2	0.00048469	5.24E-05	0.00035239	10%	3%