MD Ragib Hasan

From Statistical Process Control to Smart Process Control

Developing an Explainable AI System for Manufacturing Quality Assurance

School of Technology and Innovation
Major
Industrial Systems Analytics

Vaasa 2026

UNIVERSITY OF VAASA
School of Technology and Innovations
Author: MD Ragib Hasan
Title of the thesis: From Statistical Process Control to Smart Process Control: Developing an Explainable AI System for Manufacturing Quality Assurance
Degree: Master of Sciences
Degree Programme: Industrial Systems Analytics
Supervisor: Petri Helo
Year: 2026 **Pages:** 84

ABSTRACT:

In this thesis an AI supported Statistical Process Control (SPC) system for manufacturing quality assurance is proposed. While such traditional SPC has been adequate for nearly a century in serving the manufacturing, modern day production does not meet its essential prerequisites of normality, independence, and stationarity. Machine learning models identify intricate, nonlinear patterns that control charts are unable to detect, but most ML algorithms are black boxes. The way decisions are made is invisible to operators and it reduces trust and adoption. An explanation enabled AI augmented SPC system is developed in this thesis to bridge the trust gap in SPC. The system combines four modules: conventional SPC control charts, three ML anomaly detection models (Isolation Forest, Autoencoder, One-Class SVM), SHAP-based explainability with Cohen's d as backup, and a natural language interface based on GPT-4.1-mini. The design follows DSR methodology. The artifact is a web application developed in Streamlit. Using the Crimp Force Curves dataset, a public dataset of force measurements obtained from crimping processes, the system was tested. The assessment presented results at the same level as the ML models, both in terms of precision, recall and F1-score. SHAP received mismatch of dimension error when calculating SHAP values and therefore the system switched to Cohen's d effect size evaluation. The LLM interface was evaluated with operator questions about process status. This work makes three contributions. It exhibits the first integrated SPC-ML-XAI-LLM platform for manufacturing quality, provides a quality control education open-source software prototype, and offers a comprehensive experimental evaluation of the LLM-based SPC solution. Thirdly, it presents a quantitative analysis comparing three ML models on real manufacturing data. The results show that ML improves SPC by discovering patterns that control charts do not pick up on. This holistic process closes the trust gap between human operators and AI systems.

KEYWORDS: Statistical Process Control (SPC), Explainable AI (XAI), Machine Leaning (ML), Anomaly Detection, Large Language Model (LLM), Quality 4.0, GPT 4.1 mini

Contents

Figures

Tables

Abbreviations

Abbreviations	**Full Form**
AI	Artificial Intelligence
API	Application Programming Interface
Cpk	Process Capability Index (Upper/lower)
Cp	Process Capability Index (Overall)
DSR	Design Science Research

EWMA	Exponentially Weighted Moving Average
GPT	Generative Pre-trained Transformer
I-MR	Individual Moving Range (Chart)
LCL	Lower Control Limit
LLM	Large Language Model
ML	Machine Learning
R-chart	Range Chart
SHAP	SHapley Additive exPlanations
SPC	Statistical Process Control
SPM	Statistical Process Monitoring
SVM	Support Vector Machine
UCL	Upper Control Limit
XAI	Explainable Artificial Intelligence
X-bar Chart	Mean Chart

1 Introduction

1.1 Background

Statistical Process Control (SPC) has been a key element in manufacturing quality control for about 100 years. In the 1920s Walter Shewhart developed the first complete SPC system that provided a structured way of monitoring a production system and subsequently identified ways to improve it (Shewhart, 1931). According to Juran's Quality Handbook, SPC is a statistical technique that is applied to control processes and maintain product quality. The methods allow the operator to detect the normal process variation as well as events that require their involvement (Juran & Godfrey, 1999).

There are three fundamental requirements in traditional Statistical Process Control: If a process is to be controlled using SPC, it must stay stable and produce a normally distributed process profile throughout its operation, and all observations must be independent from each other. These requirements are not met in the modern manufacturing environment. The data stream generated by current production systems are complex, and self-correlated patterns are present during the life cycle of the production systems. For multiple sensors that are operated simultaneously, multiple measurements are collected that produce time dependent correlations. Raw material variation, tool wear and changes in the environment due to unavoidable reasons give rise to further process variations (Psarakis & Papaleonida, 2007). This is also an operational challenge for the traditional SPC techniques. Woodall & Montgomery, (2009) conducted a study on the validity of traditional control chart assumptions in an advanced manufacturing environment and found significant problems with the traditional control chart assumptions.

Machine learning (ML) models are used as additional tools to traditional statistical process control (SPC). Complex patterns as well as nonlinear anomalies not detectable by conventional control charts are identified by the ML models (Hodge & Austin, 2004). Manufacturing systems use machine learning (ML) for three purposes which include

fault detection and predictive maintenance and quality classification (Wuest et al., 2014). Most machine learning algorithms are black box systems that produce predictions without revealing the steps they take to make their decisions (Lipton, 2018). Operators need to trust their systems as they need to understand how systems work to avoid unsafe situations.

1.2 Problem Statement

There are two imperfect choices available to manufacturers' today: The first approach is to use only traditional SPC, which is transparent, well understood and trusted by operators but it applies poorly to high-dimensional, autocorrelated or non-normal data. The second is using machine learning for anomaly detection. While ML models can be very powerful and detect patterns that are not apparent from SPC, ML models are not transparent. So it is not possible for the operators to know how decisions have been made, and this reduces their trust. This transparency is no simple matter of convenience but a roadblock to real-world production use of AI. Explainable AI (XAI) addresses this problem by enabling AI decisions to be understood by humans (Adadi & Berrada, 2018). A popular XAI method, SHAP, leverages game theory to explain the contribution of each feature to the prediction. SHAP has also been used to explain a model from a global perspective (Lundberg & Lee, 2017). SHAP has additionally been applied to predict maintenance, to explain the causes of maintenance alarms and faults in industrial applications (Marchesano et al., 2025). Consequently, SHAP is usually presented with numerical values and interpreting them requires domain knowledge, preventing their utilization by shop floor engineers that need fast and simple explanations. The large language models (LLMs) such as GPT 4 have shown potential in rewriting technical XAI outputs in layman terms for the operators (OpenAI, 2023). It has been demonstrated that state of the art LLMs can provide human readable explanations for predictions in fields like medicine and finance (Jansen & Pehlke, 2025; López-Pernas et al., 2026). LLMs are also being used in industrial maintenance to provide answers to operators queries and guide them through procedures (Kohl, 2025). If SHAP is used in conjunction with an

LLM, the feature contribution scores can automatically be converted to simple human-readable statements such as: "Anomaly detected at crimp 1050. The force peak was 15% greater than ever before". This could indicate die wear. The thesis will attempt to solve its primary research problem from two perspectives: trust establishment and enhancement of anomaly detection. The system requires two essential components which need to deliver excellent detection capabilities and provide operators with understandable results. There are a few studies that address the applications of AI in process monitoring (Chang & Ghafariasl, 2025), and studies on the use of AI to enhance the explainability of maintenance decisions (Kohl, 2025; Marchesano et al., 2025), but there is currently no single system that offers all four functions: traditional SPC as a trusted baseline, machine learning for advanced anomaly detection, explainability using SHAP, and an LLM interface for natural language explanations. To overcome this issue, the study intends to develop an effective and user friendly AI-driven SPC system.

1.3 Research Question

What are the ways in which traditional Statistical Process Control can be improved by machine learning, and the incorporation of explainable AI, to develop an interpretable intelligent quality monitoring system?

1.4 Research Objectives

Guiding the empirical work of the present thesis are the following objectives:

- Design and develop a Software Platform for integration of traditional SPC with ML based Anomaly Detection.
- To use explainable AI techniques, SHAP and feature contribution based analysis for decision transparency.
- Create a LLM-powered chat bot that the operators can ask questions to the process in their natural language.

- To show the system using public manufacturing data: Crimp Force Curves data, SECOM data as a back up.
- To evaluate the detection ability of the AI-based system in comparison with the conventional SPC alone, the evaluation indices used are precision, re-call and F1-score.

1.5 Scope and Limitations

Some foreseeable constraints are identified in this work.

Firstly, the study does not involve direct interaction with the industry. Publicly accessible datasets, the Crimp Force Curves dataset and, if needed, the SECOM dataset, are the only ones that are used. This enables repeatability and makes it not reliant on proprietary information. But it also means the system doesn't get tested in a live production environment.

Secondly, the study focuses on the raw, single-feature variables, derived directly from force curves. Patterns of simultaneous interactions among several process parameters are not considered, but this constitutes a possibility for further investigations.

Thirdly, the system is described only as a software prototype and not as a commercial product. There is no implementation of physical interface with sensors, controllers, or production units.

Fourthly, explanations produced by LLM should be reviewed by a domain expert instead of being fully automated. In safety-critical systems, these explanations must be inspected by an experienced user before execution.

Lastly, the system has been tested only on historical data, and not on online production data. Training was on known cases; testing on novel, thus unseen, process cases is not yet done.

The limitations of this study outline its boundaries while the study provides paths for upcoming research. The results maintain their validity because the findings do not affect their credibility.

1.6 Relevance of the Study

Three domains are contributed to by this thesis: academia, practice, and methodology.

Contribution to academia: Appropriate ML and XAI approaches are used by the research for enhancing traditional SPC, and their effectiveness is demonstrated. How SHAP and an LLM can work together in manufacturing quality assurance is shown. The framework can be used as a template for future AI-driven quality monitoring systems.

Contribution to practice: A high-quality, open-source software prototype for quality control education is delivered by the study. It is demonstrated that explainable AI can be implemented with high clarity. The code may be utilized by educators to teach contemporary quality control methods, and the system can be adapted by practitioners to specific requirements.

Contribution to methodology: A quantitative comparison of three machine learning models – Isolation Forest, Autoencoder, and One Class SVM – on real manufacturing data is provided by the thesis. Traditional SPC is contrasted with AI-enhanced detection. Practical knowledge of the benefits of integrating AI with classic methods is delivered by the results.

1.7 Thesis Structure

The rest of this thesis is organized into five chapters.

Chapter 2 reviews the literature on traditional SPC, AI in statistical process monitoring, explainable AI in manufacturing, and LLMs for industrial applications. The chapter ends by identifying the research gap that this thesis addresses.

The methodology is described in chapter 3. It discusses why DSR was chosed, introduces the four-layer system architecture, enumerates the employed tools and technologies, introduces the dataset and summarizes the validation strategy.

The implementation is described in Chapter 4. It includes software requirements, use cases, system design, development, testing and a statement of the AI tools employed.

Chapter 5 presents the findings. It contains descriptive statistics, the traditional SPC excellent, performances of the three ML models, comparisons of SPC and the AI-based system, SHAP explainability outputs, examples of explanations generated by LLM.

The discussion, including interpretation of results, limitations, and future directions is in Chapter 6. The chapter ends by providing answers to the research questions and outlining the contributions of the research.

2 Literature Review

As discussed in the Introduction, Statistical Process Control (SPC) has served as the foundation of manufacturing quality for nearly a century. However, the three core assumptions of traditional SPC: normality, independence, and stationarity which are frequently violated in modern production environments (Psarakis & Papaleonida, 2007; Woodall & Montgomery, 2009). This section reviews the existing literature on how researchers have attempted to address these limitations through machine learning, explainable AI, and large language models

Woodall & Montgomery (2014), in their article in the Journal of Quality Technology, show that the underlying assumptions of control charts need to be reconsidered for today's complex information environment. They identify problems such as high-dimensional data, autocorrelation, and process instability where traditional methods fail.

ML has become a powerful supplement to this. As per a draft by Hodge & Austin, (2004), ML models can identify complex and non-linear patterns of anomalies, which traditional statistical charts cannot. ML is applied for fault detection, predictive maintenance, and quality classification in the industry (Wuest et al., 2014). Yet most ML models function as 'black boxes', whose decision rules are not visible to the user. Lipton, (2018) in his well-known article “Mythos of Model Interpretability” criticizes the fact that many models purport to be interpretable, but they are not really understood from within. He also challenges the assumptions that deep neural networks are always less interpretable than linear models and interprets that interpretability is a function of stimulability, decomposability, and algorithmic transparency.

To tackle this issue, Explainable AI (XAI) has been proposed as a promising solution. The aim of XAI, based on the work of Adadi & Berrada, (2018), is to expose and to help human being understand the decision-making of AI. Of all the methods related to XAI, SHAP (SHapley Additive exPlanations) proposed by Lundberg & Lee, (2017) is the most popular. SHAP applies the Shapley value from game theory to quantify each feature’s contribution to the prediction and enables explanations at local and global levels. Mar-

chesano et al., (2025) proves that deep reinforcement learning models can be explained with SHAP to build trust of operators in industrial maintenance. However, the SHAP values are numerical, and technical understanding is needed to interpret them; hence, it is not always beneficial for the operators who want to make quick decisions. Now a look on each part section by section for deeper read has been made.

2.1: Traditional Statistical Process Control

SPC is the basis of quality monitoring in production. In this section, the fundamental concepts of SPC, such as process variation, control charts, process capability indices and their mathematical expressions are briefly introduced.

2.1.1 What is SPC?

SPC is a quality control which employs statistical tools to analyze processes. The Origin of SPC at Bell Telephone Laboratories SPC was formulated in its entirety by Walter Shewhart at Bell Laboratories in the early 1920s (Shewhart, 1931). The purpose of SPC is to make the process to be run at its effective level all the time by analyzing two types of variation, common cause and special cause.

SPC is a statistical technique which controls the process and monitors the product quality according to Juran & Godfrey, (1999). These techniques allow the user to differentiate between normal process variation and situations in which intervention is required, such as when a process goes out of control.

2.1.2 Process Variation

There is always variation in every process of production. Shewhart, (1931) differentiated between two types of variation. Common cause variation is the natural, normal, inherent variability that is present in every process. For example, small differences in

the properties of raw materials, vibrations of the machines, or seasonal influence. A special cause of variation is a factor that can be identified that is responsible for the variation and is not a part of the standard process. Such as tool breakage, operator error, material defects and machine malfunction.

SPC aims at identifying special cause variation and experienced action. Common cause variation is for the most part acceptable variation, but can be improved upon (i.e. reduced) as a result of process improvement activity such as 6-Sigma or Lean manufacturing..

2.1.3 Control Charts

A control chart is a statistical graph that is used to track how a process performs over time. The control chart was proposed by Shewhart, (1931) as means of discriminating between common cause and special cause variation. There are three horizontal lines on the chart. The process mean or average is represented by the centerline. The upper control limit (UCL) and lower control limit (LCL) are the statistical control limits positioned at 3 sigma (3 standard deviations) from the process mean. Data are plotted sequentially over time and compared to the boundaries.

Any points within the limits of the control that corresponds common cause variation, meaning that the process is in control. Any points are out of control limits for which special cause variation is indicated, meaning the process is not in control and the source of that variation warrants investigation. Trends, cycles and runs are patterns that, even if they do not violate control limits, are indicators of process trouble. These pattern recognition tests, the so called Western Electric Rules (1956), increase the sensitivity of detecting changes in processes.

There are several ways in which control charts facilitate quality. They identify process shifts and notify operators prior to creating wasteful products. They make a not so obvious difference in common and special cause variation, thereby avoiding making un needed changes to an stable process. They prevent scrap and rework by identifying

problems sooner. They increase the knowledge of processes by disclosing features and tendencies. They enable continuous improvement by providing a baseline to assess improvement activities.

2.1.4 Types of Control Charts

Certain types of control charts are better suited to specific data types and goals of monitoring. Control charts can be classified into two major types, namely variable control charts and attribute control charts.

Variable control charts track continuous data that are measurable, such as length, weight or force. In order to monitor the process mean, the X-bar chart displays the average of the sub groups. The R chart is used to monitor the variability of the process by calculating the range within the subgroups. Subgroups' standard deviations are plotted by the S chart to track variability. The I-MR chart for individual measurements When the data cannot be subgrouped by a natural rational subgroup, the I-MR chart can be used for individual observations and successive differences (moving ranges). The EWMA chart is sensitive to small changes in the process mean, by putting more weight on recent observations.

Attribute control charts observe count data such as the number of bad items or defects per unit. The p-chart tracks the proportion of bad items. The np-chart tracks the number of bad items. The c-chart tracks the number of defects per unit. The u chart is used to monitor the number of defects per unit where the sample size are not constant.

Which chart to use is based on the data type and the monitoring purpose. And, for good reason; X-bar and R charts are typically paired to monitor both the central tendency and the dispersion of a process. EWMA charts are appropriate when the detection of small shifts is essential. Attribute charts Are utilized when the measurement is binary or counted.

2.1.5 Control Chart Formulas and Interpretations

The formulas used to calculate control limits depend on the chart type.

X-bar Chart

The center line of an X-bar chart is the grand mean of the subgroup means:

$CL = \bar{\bar{X}} = (1/m) \Sigma \bar{X}_i$, where m is the total number of the subgroups and $\bar{X}_i$ is the mean of the ith subgroup.

The control limits are:

$UCL = \bar{\bar{X}} + A_2\bar{R}$, $LCL = \bar{\bar{X}} - A_2\bar{R}$, where A_2 is constant, which depends on the size of the subgroup n, and $\bar{R}$ is the mean of the subgroup range.

Points which exceed the UCL or LCL suggest special cause variation. The process mean has shifted, and something needs doing about it. Sequences of points such as seven points in a row above or below the center line may indicate a concern (Montgomery, 2009).

R Chart

In an R chart, the center line represents the mean of the ranges:

$CL = \bar{R} = 1/m \Sigma R_i$, where R_i is the range of the subgroup i.

The upper and lower limits of the control are:

$UCL = D_4 \bar{R}$, $LCL = D_3 \bar{R}$ where D_3 and D_4 are constants which depend on the size of the subgroup n.

Points above UCL also indicate higher process variability, which means the process is less stable. Points below LCL (at least for such subgroup sizes n that $D_3 > 0$) also suggest a decrease in the process variance, which can be considered unusual as well.

Montgomery, (2009) states that R chart signals should prompt an investigation into the cause for the change in variability.

S Chart

For an S chart, the center line is the average standard deviation:

$CL = \bar{S} = (1/m) \Sigma S_i$, where S_i is the standard deviation of subgroup i.

The upper and lower control limits are:

$UCL = B_4\bar{S}$, $LCL = B_3\bar{S}$, where B_3 and B_4 are constants that depend on subgroup size n.

The S chart provides a more accurate measure of variability than the R chart for larger subgroup sizes. Signals above UCL indicate increased variability; signals below LCL indicate reduced variability.

I-MR Chart (Individuals and Moving Range)

For the individuals chart, the center line is the process mean:

$CL = \bar{x} = (1/n) \Sigma x_i$

The upper and lower control limits are:

$UCL = \bar{x} + 3(M\bar{R}/1.128)$, $LCL = \bar{x} - 3(M\bar{R}/1.128)$, where $M\bar{R}$ is the average of the moving ranges calculated as:

$M\bar{R} = (1/(n-1)) \Sigma |x_i - x_{i-1}|$

For the moving range chart, the center line is:

$CL = M\bar{R}$

The upper control limit is:

$UCL = 3.267 \times M\bar{R}$

The I-MR chart is used when subgrouping is not possible. The individuals chart detects shifts in the process mean. The moving range chart detects changes in variability. Both charts should be interpreted together, as a signal on either chart indicates special cause variation (Montgomery, 2009).

EWMA Chart

Monitoring Exponentially Weighted Moving Average (EWMA) charts track with the exponentially weighted moving average z̄ t z t z¯t:

$z_i = \lambda x_i + (1 - \lambda)\, z_{i-1}$, where λ is a smoothing constant (typically in the range [0.05,0.25]), and z_o is either the target value or the process mean.

The process mean is the center line in figure 1:

$CL = \mu_o$

The control limits are:

$UCL = \mu_o + 3\sigma \sqrt{(\lambda/(2-\lambda)[1 - (1-\lambda)^{2i}])}$, $LCL = \mu_o - 3\sigma \sqrt{(\lambda/(2-\lambda)[1 - (1-\lambda)^{2i}])}$, where σ is the standard deviation of the process.

The EWMA chart is more sensitive to small shifts than X-bar charts, because it assigns larger weight to recent data points. λ is the smoothing parameter, and it determines how much sensitivity the chart has. For example: λ =0.1 (small value of λ): more sensitive to small shifts of process mean, λ =0.2 (larger value of λ): can detect larger shifts more rapidly Woodall & Montgomery, (2014). The factors $[1 - (1-\lambda)^{2i}]$ symbolize the control limits reaching steady-state sides as i increases

Constants for Control Charts

The constants A_2, D_3, D_4, B_3, and B_4 depend on subgroup size n. Table 2.1 provides selected values.

Table 2-1. Control Chart Constants

Subgroup Size	A_2	D_3	D_4	B_3	B_4
2	1.880	0	3.267	0	3.267
3	1.023	0	2.575	0	2.568
4	0.729	0	2.282	0	2.266
5	0.577	0	2.115	0	2.089
6	0.483	0	2.004	0.030	1.970
7	0.419	0.076	1.924	0.118	1.882
8	0.373	0.136	1.864	0.185	1.815
9	0.337	0.184	1.816	0.239	1.761
10	0.308	0.223	1.777	0.284	1.716

The constants A_2, D_3, D_4, B_3, and B_4 are used to calculate control limits. For a given subgroup size n, these constants adjust the control limits to account for the distribution of subgroup statistics. For example, when n = 5, $A_2 = 0.577$ is used to calculate X-bar chart limits, and $D_4 = 2.115$ is used to calculate R chart limits.

2.1.6 Process Capability Indices and Interpretations

Process capability indices measure how well a process produces output within specification limits. These indices compare the natural variation of the process to the allowable variation defined by customer requirements.

Cp (Process Capability Index)

Cp measures the potential capability of a process, assuming the process mean is centered between the specification limits.

Formula:

$Cp = (USL - LSL) / 6\sigma$,

where:

USL = Upper Specification Limit,

LSL = Lower Specification Limit,

σ = Process standard deviation (estimated from control chart data),

$Cp > 1.33$: Highly capable. The process produces output well within specification limits.

$1.0 < Cp \leq 1.33$: Capable. The process generally meets specifications.

$0.67 < Cp \leq 1.0$: Marginally capable. The process produces some defects.

$Cp \leq 0.67$: Not capable. The process produces many defects.

Cp does not account for process centering. A process can have a high Cp value but still produce defects if the mean is off-target.

Cpk (Process Capability Index with Centering)

Cpk measures the actual capability of a process, accounting for both variation and centering relative to specification limits.

Formula:

$Cpk = \min((USL - \mu)/3\sigma, (\mu - LSL)/3\sigma)$

where:

μ = Process mean

σ = Process standard deviation

USL = Upper Specification Limit, LSL = Lower Specification Limit

Cpk ≥ 1.33: Capable. The process consistently meets specifications.

1.0 ≤ Cpk < 1.33: Acceptable. The process meets most specifications but requires monitoring.

Cpk < 1.0: Needs improvement. The process produces excessive defects.

Relationship Between Cp and Cpk:

If the process mean is perfectly centered, Cp = Cpk. If the process mean is off-center, Cpk < Cp. The difference between Cp and Cpk indicates how much the process needs centering.

2.1.7 Limitations of Traditional SPC

Although traditional SPC is widely used, the growing complexity of processes has led to an interest in integrating machine learning techniques. The normality condition states that the data should be approximately normally distributed to use control limits. Many traditional activities are now generating non-normal data. The assumption of independence states that the observations are independent. Modern processes often yield autocorrelated data, i.e. data in which successive observations are correlated. The stationarityassumption states that the process mean and variance do not change over time. Processes many drift result of tool wear or material changes. Univariate SPC Although it could be generalized, classical SPC is mainly univariate, focusing on the monitoring of a single characteristic at a time. Nowadays, more processes have multiple correlated variables that depend on each other in complicated manners. It is said that SPC is reactive, and it identifies problems once they happen. It does not anticipate

future failures. Common control charts such as X-bar and R charts are not very sensitive to small shifts. EWMA chart overcome this problem but need more experience to be applied and understood. Although SPC charts are transparent, patterns such as trends, cycles, and runs require training of the operator in interpretating the charts. This limits the usefulness of SPC in situations of high turnover or limited training. These drawbacks have encouraged the research of machine learning techniques for anomaly detection, which are surveyed in the following sections.

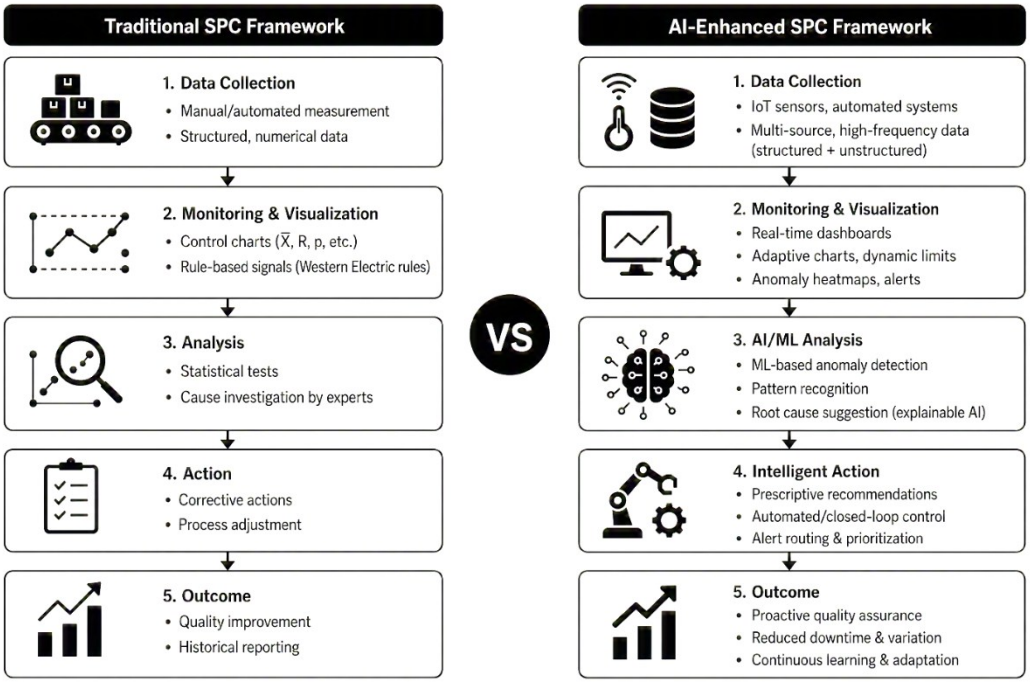

Figure 2-1. Traditional SPC Framework vs AI enhanced SPC Framework (based on the system architecture described in section 3.2)

2.2 Machine Learning for Anomaly Detection

According to a detailed review by Chang & Ghafariasl, (2025), SPC has been primarily statistical in the last hundreds of years, but recent AI advancements have opened new horizons in SPM. They have shown that neural networks such as ANN, CNN, RNN and GAN can monitor not only horizontal and multidimensional qualitative features but also

profile and image data. According to them, in the future, AI-driven Smart Process Control (SMPC) will be able to automatically prevent errors and restore processes.

Similarly, a systematic review has been described by Qiao et. al. (following PRISMA guidelines) (Entropy journal). Applying a problem-driven classification, they addressed three core problems: 1. High-dimensional and repetitive data (Dimensionality reduction, feature selection), 2. Automatically co-occurred and dynamic processes (Time-series, state-space models), 3. Scarcity and imbalance of data (Cost-sensitive learning, generative modelling, transfer learning). We review the theoretical basis and applications in the power industry of the ML-based remedy to the problem, and identify open issues concerning model explanation, selection of threshold, and real-time implementation.

Mayer & Jochem, (2024) presented a review on the Process Capability Index. They show that in the era of digitalization, the process capability index is moving from being not only descriptive but also predictive. It is possible to predict the quality of the process in advance using ML algorithms to overcome the reactive nature of traditional SPC.

Alwan et al. (2025) have proposed a next-generation hybrid control system. It combines an adaptive SPC chart (Max-mixed EWMA, Bayesian SPC) with a transformer, a graph neural network (GNN), and a reinforcement learning agent. Evaluations on real production data have shown that this hybrid method is much more effective than neural networks alone or traditional SPC: detection delay, false alarm rate, and recovery time are significantly reduced, while interpretability is increased.

2.3 Explainable AI and Industrial Applications

Explainability in industrial applications of AI models is increasingly becoming necessary. Anusha et al., (2025), they introduce a multi-level framework for explainability enhancement in predictive SPC charting: XAI-SPCNET. They introduce an architecture involving (1) a temporal causal decision tree (adaptive causal split using Granger

causality), (2) a dynamic rule-embedding network (temporal embedding and Bayesian rule learning), (3) a contrastive explanation generator, (4) sparse symbolic regression (mathematical constraint setting), and (5) a multi-objective reinforcement learner (trading off accuracy, reliability, and stability). The accuracy results show that the prediction accuracy of XAI-SPCNET is 91%, the explanation coverage is beyond 95% and the rule-form is very simple which causes transparency to process control.

Marchesano et al., (2025) introduced the Explainable Deep Reinforcement Learning (XDRL) framework in an IFAC conference paper. They utilized the PPO algorithm to obtain the optimal maintenance strategy for a three-machine flow shop and subsequently analyzed the characteristics importance in each decision with SHAP. They report that their DRL with the XAI reduces the makespan by 5.1% and the number of breakdowns by 62% in their experiments. Failure probability and ratio corrective maintenance are the largest influencing factors as can be seen in the SHAP summary plot. Hsieh et al., (2024) in the book A Comprehensive Guide to Explainable AI goes through all explainability of all models from traditional decision trees and linear regression to LLM. It gives a hands-on example for methodologies such as SHAP, LIME, integrated gradients, Grad-CAM etc. which are very useful for the researchers in XAI.

The paper by Lipton, (2018) is considered a foundational document in the XAI field. He shows that explainability is not a single concept; rather, it involves distinct roles for stimulability (understanding the model as a whole), decomposability (separate explanation of each part), and algorithmic transparency (the consistency of the training algorithm). He argues that linear models are not always more explainable than deep networks, and post-hoc explanations can also lead astray.

Miller, (2019) analyses the process of explanation in the light of psychology and social science. According to him, explanation is not just cause and effect; it is a social interaction where the explainer considers the expectations and level of knowledge of the recipient. To make XAI truly effective, this social dimension must be understood.

2.4 LLM's in Manufacturing and Quality

With the advent of large language models, the explainability and applicability of AI in the industry has been brought to new extents.

GPT-4 is a transformer-based multimodal model (accepting image and text inputs, emitting text outputs) (OpenAI, 2023) according to the GPT-4 Technical Report released by OpenAI. It has earned a place in the top 10% of test takers for the Uniform Bar Examination. The alignment after training prompted a more accurate and safer model.The model architecture and scaling methods are crucial for the thesis to understand.

Ali & Kostakos, (2023) aimed to a specialized intrusion detection dashboard named HuntGPT. They employ a Random Forest classifier, which is trained on the KDD99 dataset, and then they utilize XAI with SHAP and LIME. With the integration of GPT-3.5 Turbo into this, the explanation of the prediction is given in NL. Evaluation on the CISM certification test HuntGPT obtained 82.5% accuracy, and the readability analysis that enunciations can be given in graduate level language.

Ojuolape & Hu, (2024) introduced a causal-LM model for fault diagnosis of control systems. They applied PCA and Q-statistics on Tennessee Eastman process data to reduce the variables to the top five and then posed the question to the LLM (ChatGPT-4) which variable is the leading cause. With good prompt engineering they achieved the root cause identification with 96% accuracy using just meta data (variable names). This shows that LLM can do causal analysis without tangible numerical data.

The application of AI and LLM is expanding not only in fault diagnosis but also in the entire quality management system. Gentil & Merle, (2026) have introduced a conceptual XAI tool, QUALIBOT, that supports the quality management in the evolution towards Enterprise 4.0 and Quality 4.0- supports the quality management in its transformation. It offers clear and efficient monitoring at the strategic, organizational and operational levels of an ISO 9001-based QMS.

Megahed et al., (2024) constructed the ChatSQC chatbot by fine-tuning the GPT-3.5 and GPT-4 models with the NIST/ SEMATECH e-Handbook of Statistical Methods as the knowledge source. The results demonstrate that the grounded model (ChatSQC) offers significantly more accurate responses than the baseline (GPT-3.5 and GPT-4) and the model can answer "I don't know", which mitigates hallucinations. They demonstrate that the inclusion of domain-specific reference material makes the answers of the LLM much more trustworthy.

2.5 Hybrid and Multi-Modal Approaches

Hybrid and multi-modal methods are proving to be more effective than single methods.

In Hu et al., (2026), DML and LLM hybrid architecture is proposed to detect faults among complex sensor rich industrial systems. DML provides a causal and temporal fuzzy logic environment, LLM processes unstructured data (logs, notes, documents) in that environment. In a semiconductor fab case study, this combined approach reduced the detection time by 1.2 h from 7.4 h of traditional SPC and FDC workflows, by 3.2 h from 4 h, increased the F1 score from 0.42 to 0.59. improved the F1 score from 0.59 to 0.83 and the triage time for engineers from 72 min to 18 min. This confirms that combining LLM with structured reasoning can make the results more dependable and auditable.

Jansen & Pehlke, (2025) suggested a novel way to improve explainability by integrating LLM standard workflows. They developed layered architecture by embedding LLM logic within standardized processes like Question-Option-Criterion (QOC), sensitivity analysis, game theory and risk management. This design isolates the undisclosed reasoning space of the LLM from the explainable process space. Assessments of DAO investment suggestions have indicated that the application of GPT-5 reduces expenses and approaches human-level decision-making.

López-Pernas et al., (2026) gave a tutorial on enabling XAI outputs and LLM to be expressed in user-friendly natural language for analysis of learning. Nothing was

manually converted, but they fashioned feature importance, PDP plots and local SHAP explanations into narratives easily comprehended by teachers and students using open-source models (e.g., Gemma) via LM Studio and into the Python package Elmer. This demonstrates that XAI + LLM can be applied successfully in industry as well as education.

Kohl, (2025) is focused on a knowledge-based maintenance scheme in smart production. ARCHIE: a cognitive maintenance system architecture that integrates physical with virtual (i.e., text logs, shift reports) sensors. 20 per cent reduction in MTTR through its application to the semiconductor industry. His work illustrates how an LLM-based cognitive assistant simplifies maintenance by responding to operators' inquiries and walking methodically through the procedure.

2.6 Research Gap

It is clear from the above review that there is no integrated solution in the current literature that combines the following four elements:

1. Traditional SPC as a reliable foundation

2. Machine learning and LLM-based anomaly detection

3. SHAP-based explainability

4. LLM interface for discussion with operators in natural language

Chang & Ghafariasl, (2025) provided future directions for AI-based SPC, but they did not discuss the combination of SHAP and LLM. Qiao et al. (2026) presented various challenges and solutions in machine learning but left the issues of real-time explainability and thresholding unresolved. (Anusha et al., 2025) brought explainability to the XAI-SPCNET framework, but there is no LLM interface. (Ali & Kostakos, 2023) integrated LLM and XAI in HuntGPT, but it is for cybersecurity, not for production quality control. Marchesano et al., (2025) used SHAP but did not add LLM. Hu et al., (2026)

combined DML and LLM but did not use traditional SPC as a baseline. Jansen & Pehlke, (2025) increased explainability by keeping LLM within the standard process, but not in the context of SPC or manufacturing quality.

Table 2-2. Positioning the thesis against prior work

Study	Traditional SPC as Baseline	Ml-based Anomaly detection	SHAP/XAI	LLM interface for operator	Domain/Application
(Chang & Ghafariasl, 2025)	Yes (review)	Yes	No	No	Statistical Process Monitoring (Conceptual)
Qiao et al. (2026)	Yes (review)	Yes	No	No	Statistical Process Monitoring (Taxonomy)
(Anusha et al., 2025)	Yes	Yes	Yes	No	Predictive SPC Charting (XAI-SPCNET)
(Ali & Kostakos, 2023)	No	Yes (random forest)	Yes (SHAP+LIME)	Yes (GPT-3.5 Turbo)	Cybersecurity (HuntGPT)
(Marchesano et al., 2025)	No	Yes (deep RL)	Yes (SHAP)	No	Industrial Maintenance (XDRL)
(Hu et al., 2026)	No	Yes (DML)	No	Yes	Semiconductor Fault Detection
(Jansen & Pehlke, 2025)	No	No (Conceptual)	Yes (via standard processes)	Yes (GPT 5)	General AI Explainability (DAO investments)
This Thesis	Yes	Yes (Isolation forest, SVM, autoencoder)	Yes (SHAP)	Yes (LLM powered chat interface)	Manufacturing Quality Assurance (Crimp Force Curves & SECOM datasets)

Filling this gap is the main contribution of this thesis: a software platform where SPC, ML anomaly detection, SHAP explainability and LLM chat interface will work together.

2.7 Summary of the Literature Review

- Traditional SPC is robust, but it is not without its challenges when applied to high-dimensional data.
- Machine learning can uncover strong anomalies but cannot be trusted by the operators as it is a black box.
- Explainable AI (SHAP) can expose the decisions of the model which improves the confidence of the users.
- Large language models (GPT-4, HuntGPT, QUALIBOT, ChatSQC) translate XAI output into natural language that can be understood by the users of all educational levels.
- The hybrid approach (DML-LLM, hybrid control system) improves both the detection power and the explainability.
- Currently, there is no platform that can seamlessly integrate these four modules (SPC + ML + XAI + LLM).

This research will fill that gap and create an open-source, explainable, robust quality control system.

3 Methodology

3.1 Research Method: Design Science Research

This study has followed the Design Science Research (DSR) methodology. DSR was chosen as the research focuses on creating an artifact, rather than simply analyzing an existing situation. This thesis is about an AI-enabled SPC dashboard that can be used in manufacturing quality control. Designed to provide clear and structured approach to the design, construction, and testing of a solution to be applied in the field.

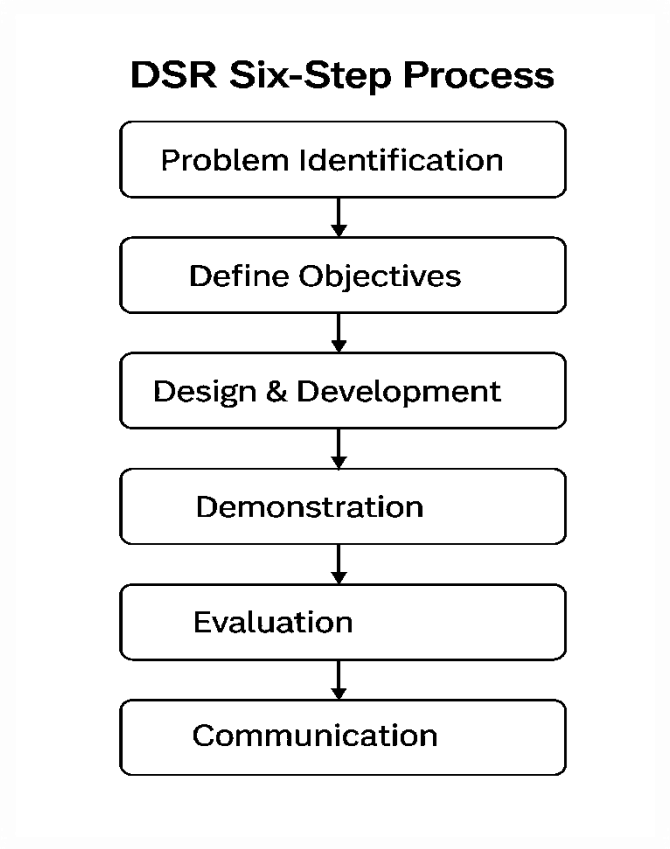

Figure 3-1. Six-steps of DSR Processes

The practical issue addressed in this thesis is the trust gap between machine learning models and manufacturing operators. Traditional SPC is transparent and widely accepted

in manufacturing settings because operators can review control charts, limits, and any violations directly. Nevertheless, SPC becomes less effective when the data is elaborate, non linear, or biased with hidden patterns. On the other hand, machine learning can identify complex anomalies, but its results are often hard to interpret which is the main reason for ultimately lowers operator trust.

The proposed system brings together SPC, machine learning, explainability methods, and a natural language interface. SPC supports familiar monitoring processes wheras machine learning is used for anomaly detection. SHAP and Cohen's d are applied to provide explanations at the feature level and finally, the LLM interface enables users to ask questions about the process status. DSR is well suited for this work because the system is developed, demonstrated, and evaluated as a practical artifact.

The DSR process used in this thesis follows the six steps outlined by Peffers et al., (2007). The first step was problem identification and motivation. The research problem was defined through a literature review and an analysis of challenges in manufacturing quality monitoring. The core issue is that manufacturers often collect process data but cannot use AI-based monitoring effectively because the model outputs are not explained clearly enough.

The second step was defining the objectives of the solution. The research questions were translated into system requirements. The system needed to extend SPC with ML-based anomaly detection, provide explainability using SHAP (or a fallback method), and support natural language interaction through an LLM interface.

This was followed by design and development phase in third step. The software product developed by the 4-layer architecture. Data layer: This layer is responsible for loading, pre-processing, and feature extraction from the data sets. The analytics engine will perform SPC calculations, and train the machine learning models. The explanation engine generates SHAP explanations and Cohen's d analysis. The system is accessed via four tabs on the user interface created using Streamlit (a python library). As implementation was carried out, each layer was developed, tested and refined.

The next step was demonstration.The fourth step was demonstration. It has been tested with the dataset of Crimp Force Curves and shown the following processes: Data loading, generation of SPC charts, training of ML model, generation of explanations and interaction via AI Chat tab.

The fifth step was the evaluation. Precision, recall, F-1 score, Cpk and control limit violation were used to evaluate the quantitative evaluation. This step was aimed at evaluating the clarity, accuracy and usefulness of LLM explanations. The usability of dashboard interface was also evaluated.

The final step was communication. The results are shared in a special way in the open-source code repository. This helps others to validate and elaborate on the efforts.

As a whole, the DSR process provides a comprehensive way of designing the artifact based on a problem and evaluating via measurable criteria. Therefore, research contribution covers not only the software system itself, but also an evaluation of the software system's contribution in enabling the support of manufacturing quality monitoring.

3.2 System Architecture

The dashboard was developed using four layers design, with each layer responsible for carrying out a specific functionality. This setup aids testing, continuous upkeep, and future updates. These layers include: Data layer, Analytics engine, Explanation engine, User interface (as seen below).

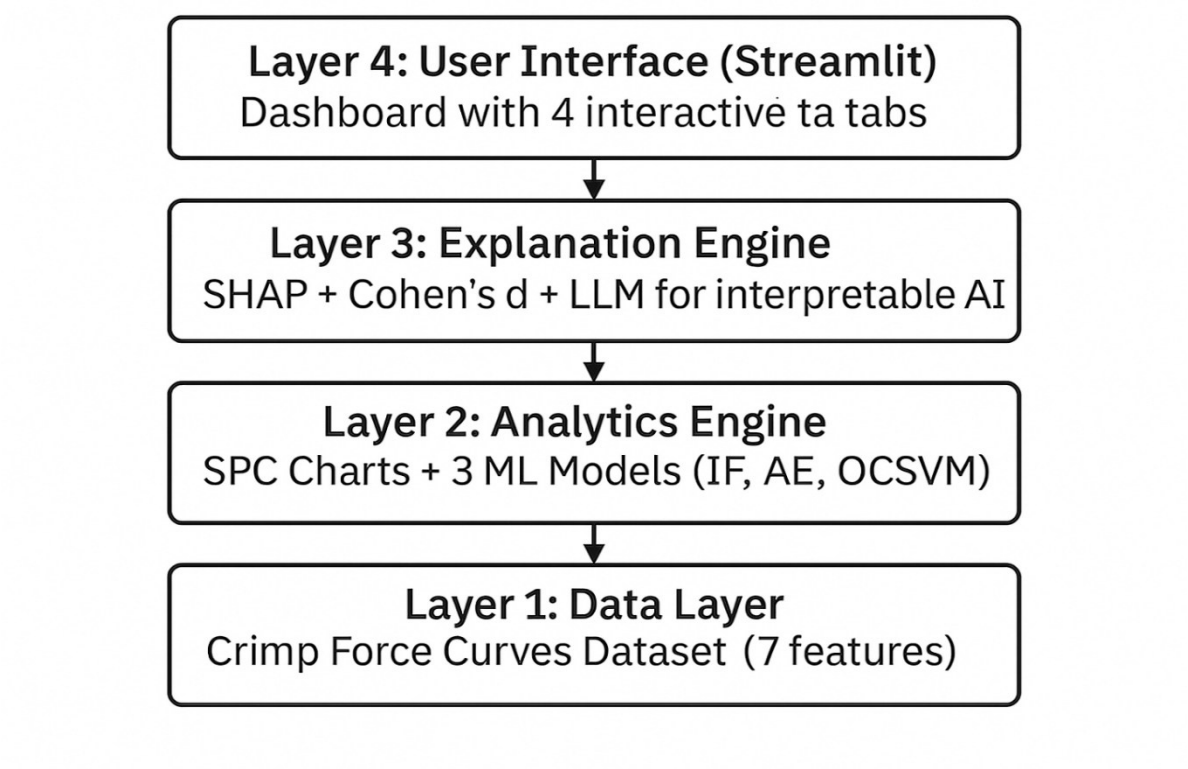

Figure 3-2. 4 layers of system architecture

The data layer is for acquiring data, preprocessing and feature extraction. It will load the dataset from pickle files, provided they are named "crimp_force_curves.pkl", and a preprocessed version of this file. When the expected dataset file is not found the system will automatically use fake demo data to be able to run the dashboard. This is a backup which is intended for test and demonstration purposes.

Mean imputation is used for managing missing values. Seven features are extracted from the raw force curves after being preprocessed: curve_min, curve_max, curve_mean, curve_std, curve_median, curve_slope and curve_rms.

SPC and ML based anomaly detection is then applied via the analytics engine. SPC module generates X-bar, R, EWMA and I-MR control charts and calculates Cp and Cpk. These control charts are used for assessing whether the data of the process is within the upper and lower control limits. Cpk indicates the fit of the process within the specification limits.

For machine learning anomaly detection, three unsupervised models are chosen which are Isolation Forest, Autoencoder, and One-Class SVM. Anomalies are identified by using a random partitioning approach at Isolation Forest. The Autoencoder learns what are the normal reconstruction behaviors and what are those that create a high reconstruction error. One-Class SVM learns a boundary around normal data and looks upon data outside of the boundary as an anomaly.

The data set is divided into training and test data (70:30)%. The models are trained on training set and tested on test set. The models' performances are compared using precision, recall, and F-1 score.

To enhance the model output interpretation, an explanation engine is used. SHAP for the first time is utilized to produce feature contributions. A dimension mismatch happened during development between training and test data and Cohen's d was added as a fall-back. Cohen's d is used to provide information about how different the two types of samples are, thereby indicating which measurements best distinguish between the normal and abnormal samples.

For this case, all the features extracted were smaller when Cohen's d was checked, showing that all were anomalous samples. This gives a clear focus on what to investigate about the process. Potential cause is examined for Force Level, Tool Condition, Material Variation or Machine Calibration.

Finally, the engine for explanation is linked with the LLM. GPT-4.1-mini is provided with the dashboard context and asked to provide natural language responses for the users' questions. Anomaly counts, model outcomes, SPC values and results generated by the explanation engine are part of the context.

The user interface is created using Streamlit and divided into 4 tabs: SPC Charts, ML Anomaly Detection, SHAP Explanations, and AI Chat.

Users can view control charts on the SPC Charts tab and Cpk values. The ML Anomaly Detection tab displays the training results, statistics and anomaly plots of the model. The

SHAP Explanations tab gives sample level explanations and feature importance. The AI Chat tab enables users to pose questions in a natural language about the process.

Overall, the dashboard is for operators and quality managers and aggregates all SPC, machine learning results, explanations and chat answers in one place.

3.3 Technologies and Tools

The entire development process of the dashboard – data processing, machine learning, visualization, explainability, and web application development – is done using Python and various other Python libraries.

Python 3.11 was used as the main programming language. Pandas was used for loading the dataset, processing missing values, and extracting features. Numerical operations, array manipulation, and more statistical calculations were performed using NumPy. All these libraries combine to be the backbone of the entire data processing pipeline.

The Isolation Forest and One-Class SVM models were developed using scikit-learn which handled the task of data splitting, training and testing and the calculation of the performance measures. After that, TensorFlow and Keras were used to construct the Autoencoder (AEC) in which the layers were defined and dropout and early stopping were used to help mitigate the risk of overfitting. By means of these programs three different anomaly detection models could be implemented into one tool.

Table 3-1. Technology and Tools

Tool/Library	**Purpose**	**Version**
Python	Programming language	3.11
Pandas	Data manipulation	2.0.0

NumPy	Numerical operations	1.24.0
Scikit-learn	ML models	1.3.0
TensorFlow	Autoencoder	2.13.0
SHAP	Explainability	0.42.0
Stremlit	Dashboard	1.28.0
Plotly	Interactive charts and visualizations	5.14.0
Matplotlib	Static plots for SHAP visualizations	3.7.0
OpenAI API	GPT 4.1-mini-integration	1.0.0
Git	Version Control	Latest
Github	Code repository hosting	-

Feature importance analysis was done using SHAP. If SHAP fails, due to a dimension mismatch, then SciPy was used as the explanation method fallback. Cohen's d effect size and percentile ranks were also calculated using SciPy, and explanations were still provided when SHAP didn't.

The web dashboard was created using Streamlit. The interactive charts—e.g., model comparison figures and control plots—were created using Plotly, and Matplotlib was used for static charts, primarily in the explanation section. The tools selected are compatible and easy to use with Python data structures and model outputs.

Lastly, integration of GPT-4.1-mini into the AI Chat tab was accomplished using the OpenAI API. The use of the context within the dashboard to generate natural language

responses. Also implemented was a rule-based fallback in event that no API key is given, and the dashboard functions even in the absence of an external API.Version control was by git and the source code repository was on GitHub. This approach allows to review, re-use and make it accessible for public.

3.4 Dataset

The main data in this study is the Crimp Force curves dataset that is publicly accessible on Kaggle. It also contains measurements of forces during crimping operations. The records are associated with time-varying force curves. The data is suitable in this research due to a number of reasons.

First, the data is based on actual production processes and thus the work becomes practically applicable. Moreover, the dataset is expert-labeled with anomalies hence reliable ground truth to be used in validation. It is also publicly available thus making it reproducible. In addition, it includes typical features of modern manufacturing data (such as non-normal distributions, and autocorrelation). These features precondition that this benchmark can be particularly helpful when it comes to assessing the hybrid technique suggested earlier.

Overall, this dataset contains more than 2,000 entries. Each of the entries is marked as either normal or belongs to one of the three anomaly classes i.e. anomaly 1, anomaly 2 or anomaly 3. Based on the raw force curves, the study was able to engineer seven features of the description of various properties of the crimping process. In particular, they are the following features:

Table 3-2. Feature Description

Feature	**Description**	**Why Important**
Curve_min	Minimum force during crimping	Flags insufficient crimping force
Curve_max	Maximum force during crimping	Flags excessive force
Curve_mean	Average force over the crimp	Indicates overall process level
Curve_std	Standard deviation of force	Measures process consistency
Curve_median	Median force value	Robust central tendency measure
Curve_slope	Rate of force change	Can indicate tool wear
Curve_rms	Root mean square force	Represents energy of the crimp

The domain knowledge and prior experience in studying crimp quality monitoring have been used to select these features. Being a set, they provide several dimensions of the crimping behavior, including its peak force properties and variability.

One of the questions that can be raised here is why the analysis uses different sample sizes with SPC and ML since the entire dataset contains over 2,000 samples. This option is not based on any restrictions but it is by virtue of how each approach is formed to work.

For traditional SPC, the X-bar chart analysis was carried out using 200 samples. Control charts are intended as visual tools, and if too many points are included, the chart becomes cluttered and harder to read. In such cases, signals of process instability may be harder to recognize. In standard SPC practice for Phase I, it is common to use a manageable number of subgroups typically 20 to 25 subgroups of size 4 or 5, which yields approximately 100 to 125 observations. In this study, 200 samples were selected because they produced a control chart that remained clear and interpretable, while still being sufficient to detect meaningful violations. The 38 violations previously found in these 200 samples were sufficient to demonstrate that the process was on the verge of collapse, and that conclusion would not have been changed by adding more samples.

For the MLbased detection, we split the dataset into training and testing set by 70/30. There were 732 samples in the test set. This setup is to test the models on data they have never seen before in training, so as to have a more realistic measure of their generalizability. Among the methods, Isolation Forest was the best and it does internal subsampling as part of its design. By default, each tree is constructed from a random sample of the training data (when sampling is enabled). This is embedded in an algorithm and should not be interpreted as some kind of restriction. In practice, it contributes for making Isolation Forest both efficient and more robust to noise. It also mitigates wasting work, by training models on, potentially, billions of data points and then finding them underfitting. With that in mind, the 732 test samples were sufficient for evaluating the model's performance with statistical confidence. The 96 anomalies identified from these samples also demonstrated that the model could capture subtle patterns missed by SPC.

The varying sample sizes come from the different needs of each method. SPC prioritizes visual clarity and real world usefulness. In oppose, ML focuses on algorithmic efficiency and strong generalization. Both approaches were applied to representative portions of the dataset and produced results that were both informative and meaningful. As a result, the findings from each analysis are valid and also complement one another.

The data set was again divided into a training set (70) and a test set (30) for the ML models and the test set contained 732 samples. These test samples were withheld in order to measure performance on data not used for training, to ensure a fair test of the models on data that they had never previously encountered.This enables them to be accurately assessed in terms of their generalisation. Moreover, if the Crimp Force Curves dataset is not supplied, the system generates the synthetic demo data with the help of the random number generator of the NumPy library. This allows the dashboard to still show it's work without the real data set making the system suitable for educational and demonstration purposes.

3.5 Validation Strategy

The validation method involves mixed approach of quantitative and qualitative methods. The system should provide correct outputs in addition to being clear and understandable to the manufacturing users.

Quantitative Validation involves comparing predictions of the ML model with labels that are point labels provided by experts. It depends on precision, recall and the F-1 score. Precision is the ratio of true anomalies divided by the number of anomalies which are flagged by the system: high precision would have low false positive. The "recall" is the percentage of all positive anomalies that the model is able to correctly identify and detect, and the higher the recall ratio, the fewer number of anomalies were in danger of being missed. The score F-1 is a combination of precision and recall.

Table 3-3. Validation metrics and purposes

Metric	**Purpose**	**Interpretation**
Precision	Measures false alarm rate	Higher= fewer false alarm

Recall	Measure missed anomalies	Higher= fewer missed anomalies
F-1 score	Harmonic mean of precision and recall	Balanced performance measure
Cpk	Process Capability Index	>=1.33= capable, 1.0-1.33= acceptable, <1= needs improvement
Control Limit Violations	Process stability indicator	High violations= out-of-control process

For comparison, traditional statistical process control (SPC) is used. Control limit violations and Cpks are used for measuring SPC performance. A very popular process capability index used in manufacturing is Cpk. If Cpk of 1.33 or higher, then the process is capable. If the number is at or above 1, but below 1.33, it is still in the "weak" range. If it is less than one, it's a sign that the process needs improvement.

It is expected that the ML models would perform at least as well as SPC and would be better by 15-20 percent. The comparison is performed to determine if there is any added useful detection capability with ML over the control charts. SPC remains in the evaluation since it is clear and known to operators.

The focus of qualitative validation is on LLM explanations. The review concentrates on clarity, accuracy and realism of content. The explanation should be easy to follow, should accurately refer to the data in the dashboard and should contain recommendations for the user. For instance, it could suggest checking tool wear, checking material variance or checking machine calibration.

The usability of the Dashboard is evaluated too.This includes the interface design, user-friendly navigation, chart readability, and the users' ability to draw valid interpretations

based on the results. Switching between the SPC view, ML outputs, the explanation, and the chat function should be seamless and intuitive for users.

Reproducibility is tackled in a number of ways. All random operations (train-test split, model initialization) are done using a fixed random seed of 42. The data set is publicly available. The source is open so that people can look at it, and reuse it. The documentation explains the operation of the system. In addition, there is a fallback option for when there is no data to run the dashboard (this is synthetic data).

4 Implementation

4.1 Software Requirements

The dashboard has been developed based on some functional and non-functional requirements. These needs were the implementation criteria and were applied to evaluate the system that was completed.

The functional requirements specify the functionality of the system. Data will have to be loaded into the system using pickle files, namely, crimp_force curves.pkl (or a processed one). In case of the unavailability of the file, it will have to create fabricated demo data. The cases of missing values have to be filled with mean imputation. The force curve data need to be mined to extract seven features.

To analyze via SPC, the system should produce X-bar, R, EWMA and I-MR charts. The calculation of control limits should be based on the standard SPC formulas. Control limit (violation) should be determined and reported. Cp and Cpk will have to be determined and indicated in the dashboard.

For ML anomaly detection, the system must include Isolation Forest, Autoencoder, and One-Class SVM. The models must be trainable on a subset of the data. Their parameters must be configurable where needed. Precision, recall and f-1 measure should be used to assess the performance. The system should also compare model results and select the best model.

For explainability, SHAP must be used for feature importance analysis. If SHAP fails, Cohen's d must be used as a fallback. Sample-level explanations must be available for individual anomalies. This allows users to inspect why a sample was flagged.

For LLM integration, the system must accept natural language questions. It must return context-aware responses based on dashboard results. If no API key is provided, a rule-

based fallback must respond using available data. The system must also allow comparison between SPC and ML-based detection.

The non-functional requirements define system quality. The interface must be web-based and accessible through a browser. It should work on different screen sizes. SPC calculations should finish within five seconds. ML training should finish within 60 seconds under normal operating conditions. The code must be open source and documented. The system must handle errors and missing data without crashing.

Usability requirements were also defined. The interface must be easy to navigate. Visualizations must be readable. Results must be accompanied by enough explanation for manufacturing users to understand the output.

Table 4-1. Functional vs non-functional requirements

Requirement	Description	Type
Data Loading	Load from pickle files or generate synthetic data	Functional
Data Preprocessing	Handling missing values, extract 7 features	Functional
SPC Charts	Generate X-bar, R, EWMA, I-MR Charts	Functional
Capability Analysis	Calculate Cpk	Functional
ML Models	Implement Isolation Forest, autoencoder, one-class SVM	Functional
Model Evaluation	Calculate precision, recall, f-1 score	Functional

Explainability	SHAP with cohen's d fallback	Functional
LLM Chat	Natural language Q and A with GPT 4.1-mini	Functional
SPC vs ML comparison	Compare detection rates	Functional
Web-based	Accessible through browser	Non-functional
Performance	SPC<5s, ML training<60s	Non-functional
Open Source	Code publicly available	Non-functional
Error handling	Graceful fallbacks for errors	Non-functional
Usability	Intuitive interface, clear visualizations	Non-functional

4.2 Use Cases

Four primary use cases were specified for the dashboard.

In the first use case, an operator investigates an anomaly. The operator opens the ML Anomaly Detection tab once a sample has been flagged. The dashboard reports that Isolation Forest found 96 anomalies, representing 13.1% of the test dataset. The operator clicks on an anomaly and looks at its feature values. The explanation tab indicates that all extracted features are decreased in the anomalous samples. The operator then tests for insufficient crimping force.

Quality manager inspects process capability in the second use case. The manager opens the SPC Charts tab and click on curve_min. The dashboard shows an X-bar chart with 38 violations. The Cpk is 0.061. This says the process is not capable. The manager then decides to take corrective action.

In a third example a quality manager analyzes the patterns of the anomalies. The manager then clicks on the ML Anomaly Detection tab to see the anomaly rate. The dashboard indicates 96 anomalies from 732 test samples, or 13.1 percent. It also reports that 19 samples were predicted by all 3 ML models. These samples are the highest confidence anomaly group . The manager reviews them in the explanation tab. The general trend is all features are smaller in anomalous group. The investigation is focused on force reduction, tool condition, material variation, or calibration.

In a fourth example, an operator queries process status from the AI Chat tab. The operator inquires if there are any anomalies in the process. The LLM answers that Isolation Forest identified 96 anomalies which is 13.1 percent of the test set. It suggests looking for tooling wear, material inconsistencies, and machine calibration. The operator also asks how one can reduce variation. The LLM returns recommendations on special cause detection, process standardization, equipment maintenance, and utilization of ML results.

These use cases show how the dashboard supports both operators and quality managers. Operators can inspect individual samples. Quality managers can review process capability, anomaly rates, and model agreement.

4.3 System Design

The architecture of the system is the four-layers architecture as discussed in Chapter 3. The work of each layer is determined individually.

The Crimp Force Curves dataset has been inserted into the data layer or synthetic data has been generated in case of the non-availability of the file. It handles missing values and has seven features of the raw force curves. The resultant is a cleaned dataset to be analyzed.

The analytics engine is capable of computing SPC and training ML models. It produces X-bar, R, EWMA as well as I-MR charts. It computes the values of Cp and Cpk. It trains Isolation Forest, Autoencoders and a single class SVMs models on 70 percent of the data. The rest 30 percent is allocated to testing. There are 732 samples in the test set. Each model is computed in terms of precision, recall and F1-score of the system.

SPC results and ML predictions are fed to the explanation engine. Where possible, it implements SHAP. In case SHAP is not successful, it uses Cohen d to compare normal and abnormal samples. This determines what features are most different in the two groups. The context preparation is also involved in preparing the context to the LLM through the explanation engine.

Streamlit is used to create the user interface. It provides four tabs that can be interacted with. The users can choose charts, adjust the model parameters, view the outcome of anomalies, view the explanations, and ask process related questions.

The flow of data is in a predefined order. The data layer either loads or creates data. It preprocesses the data and obtains features. The result of the analytics engine is SPC and ML. The explanation engine generates both feature-level explanations and context in LLM. The outputs are displayed in the user interface.

There is minimal inter-layer intercourse, which is determined by data transfers. The data layer sends the features that are contrived to the analytics engine. The SPC output and ML predictor are sent to the analytics engine to the explained engine. The explanation engine drives the explanations, visualizations and chat context to the interface. This kind of disconnection reduces the aspect of component dependency.

4.4 Development Process

The process of development was agile and had two-week sprints. Construction of the system was done in phases. Each stage involved the creation of a part of the dashboard.

Phase 1, entailed the establishment of SPC and this was done in the first two sprints. Pickle files data loading functions were added. I-MR chart, EWMA, X-bar and R chart were made. Calculation of Cpk was introduced. The most basic Streamlit dashboard was created. The final product was the working SPC Charts tab and the control charts and ability analysis.

The identification of ML anomalies was carried out in sprint three and four and is known as phase 2. Isolation Forest and One-Class SVM and autoencoders were used. The model training pipeline and train test split were built. Precision, recall and F1-score were presented. The results of ML were brought on the dashboard. The outcome was the ML Anomaly Detection tab.

Phase 3, discuss explainability and was accomplished in sprint five and six. SHAP was added to measure the feature importance, cohen d was added as the second line of defence in case of a dimension mismatch issue. Explanations at sample level were used. The outcome was SHAP Explanations tab.

Phase 4 entailed incorporating LLM and it occurred in sprints seven and eight. GPT-4.1-mini was linked to the OpenAI API. Context of dashboard was transferred to the model. An API key-less fallback was established and based on rules. The result of this was the AI Chat tab.

Next phase is on validation and thesis writing. This will be to run the system using the Crimp Force Curves data, evaluate the result and report all the findings, and the final report.

The sequential approach enabled the testing and development of one of these modules then the other modules were incorporated.

4.5 Testing and Experimentation

The units were tested in unit testing. Checking the computation of SPC limits was done by use of known data. The feature extraction functions were experimented to make sure that the seven features were all extracted correctly. The model training functions were tested using small datasets. Missing or incomplete data was used to test the error handling. The base tests above showed that the base functions were operating as they were supposed to.

The presence of inter-layer data flow was checked by integration testing. The analytics engine flow to data layer was tested to verify that there was the appropriate transfer of clean data. Flow The analytics engine to SPC flow was verified, to make sure that the predictions and SPC results were sent to one another. The flow between the user interface and the explanation engine was also tested in order to make sure that there were no issues with displaying the charts and explanations. The SHAP to the d fallback of Cohen was tested as well.

Testing validation was done based on the differences between system outputs and expected values. The predictions of the ML were compared to the predictions of the experts (precision, recall and F1-score). The calculation of SPC had been compared with the manual calculations. The standard formula was used to check Cpk. These tests were done to make sure that the outputs were congruent with what was expected.

The two methods were compared (SPC vs. ML) to examine the performance of the two methods. The 200 samples of SPC which constituted 19 percent had 38 violations. ML detected 96 abnormalities against the 732 samples on 732 test samples with a percentage of 13.1. A group of 19 samples was flagged by all the three ML models. The highest-confidence anomalies were given these samples.

As it is shown in the comparison, SPC and ML ascertain different conditions of processes. SPC puts occurrences of control limits in violation, and helps analyze ability. ML recognizes patterns through learnt patterns among features. The two approaches

combined would provide holistic process monitoring in comparison with that of either of the approaches.

An ultimate test was done to test the entire workflow. Data loading, SPC analysis, ML detection, explanation generation, and LLM response were used to run the system. All the four tabs were displayed. Interaction with the operator was demonstrated by the dashboard. Screen-shots and presentation slides were used as the means to record the findings.

4.6 Openness during the use of AI Tools

GitHub Copilot was used to develop the code. It provided the completion of codes and suggestions about functions. The author edited, tested and edited the code and then incorporated it in the whole system. Unproven code was not tolerated.

The AI Chat was utilized in the dashboard with the help of the OpenAI API. This model was GPT-4.1-mini. The query allows the users to make natural language queries on the process status. The model is given dashboard context and responses to results provided. A fall back of cases where the API key was not given on a rule-based one was experienced.

Streamlit was used to build the web dashboard. It enabled the interface to be written in Python and be directly linked to the data processing, SPC, ML and explanation modules.

The language editing and rewriting were also done with the aid of AI when preparing the thesis. The ultimate content, technical accuracy, interpretation of results and academic decisions remained the tasks of the author. Texts with the help of AI were checked and corrected prior to inclusion.

The code of the instructions in the thesis can be found on GitHub as a source code. This facilitates an inspection, re-use and re-implementation. The use of AI tools should be disclosed to promote transparency in academics.

5 Results

5.1 Descriptive Statistics

This study has been empirically based on the Crimp Force Curves data that consisted of more than 2,000 individual records of the force curves, one of which represented each crimping operation. The domain experts annotated each record and the identified three pieces of different anomaly types along with a normal operational class were identified. In order to enable the use of machine learning (ML) algorithms, a feature engineering process was performed, and seven most important quantitative features were obtained based on the raw curve data.

The predictive models were tested on a special test set of 732 samples (30-percent holdout of the original dataset) and the rest 70 percent on model training. The test set consisted of 96 cases of anomalies and 636 cases of normality, a class ratio that was intentionally kept such that it approximated the desired ratio of anomalies which had been specified in the first phase of the model configuration.

To conduct subsequent statistical process control (SPC) analysis, 200 samples from the particular feature of the curve, that is, curve_min, were filtered out. This characteristic was chosen because it is a quality parameter that is critical due to its measurement of the minimum force during the crimping cycle. The lowest force is a straight and significant measure of the mechanical stability and structural resilience of the crimp. It was also decided that the subset size of 200 was the best to offer a transparent and readable visualization of the behavior of the process to prevent clutter of charts and at the same time represent the process.

5.2 SPC Performance

Conventional SPC was applied to the curve_min feature by constructing a X-bar control chart. The chart was built with a subgroup size equals 5. The UCL was computed to be 0.555. Process Mean is 0.411. The LCL was shown in the output of the dashboard as it was computed with a value of 0.267.

The control chart showed 38 violations (out of 200) samples. This corresponds to a 19% violation rate. The violations were points that fell outside the control limits. These points indicated special cause variation that means the process is not stable. It requires investigation and corrective action.

The process capability indices Cpk was computed. The value is 0.061. That's an incredibly low number. It represents a so out of control process. The mean of 0.411 is alarmingly near to the upper control limit of 0.555. The difference between the mean and the upper bound is only 0.144. Based on the Cpk calculation, the process standard deviation is approximately 0.7920. This is because the process variation is so large, compared to the specification width.

A Cpk below 1.0 indicates the process needs improvement. A Cpk of 0.061 is well below this threshold. The process is producing many out-of-specification products. This finding alone justified the need for better monitoring methods.

Table 5-1. SPC Results summary

Metric	value	Interpretation
Feature analyzed	Curve_min	Minimun force during crimping
UCL	0.555	Upper control limit

Process mean	0.411	Central line
LCL	0.267	Lower control limit
Violations detected	38 out of 200	19% violation rate
Process standard deviations	0.792	Estimated from Cpk
Cpk	0.061	Critically out of control

The SPC results showed that the process was already in a critical state. The 38 violations were not subtle. They were obvious deviations from the control limits. Traditional SPC successfully identified this problem. However, it could only detect the most extreme deviations.

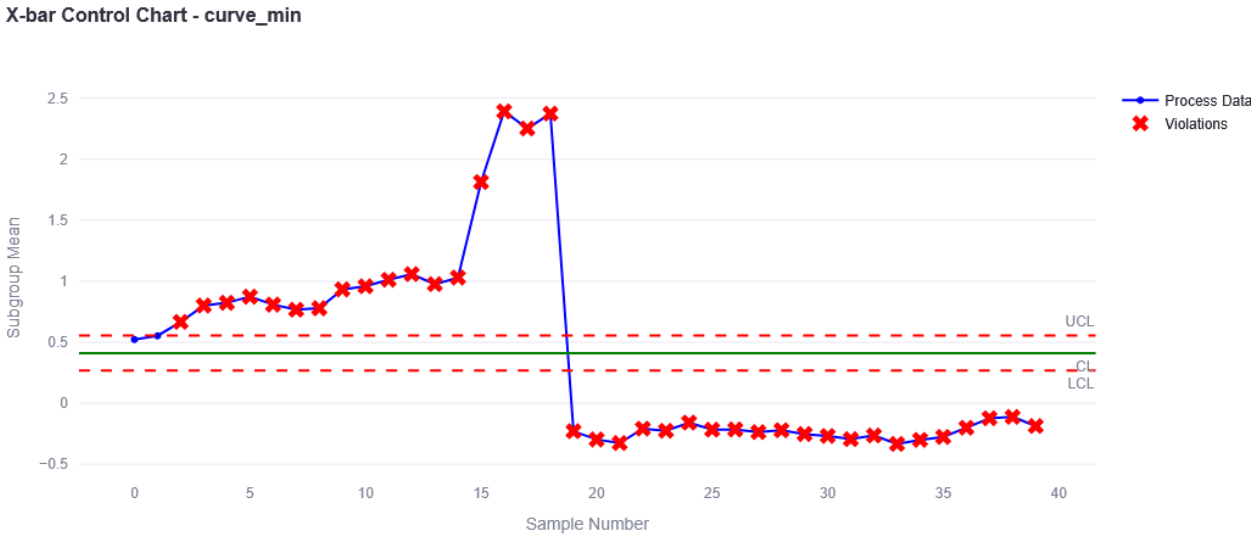

Figure 5-1. X-bar Control chart- Curve_min

5.3 ML Performance

Three machine learning models were trained and evaluated. The models were Isolation Forest, Autoencoder, and One-Class SVM. They were trained on 70 percent of the data. They were tested on the remaining 30 percent. The test set had 732 samples. Each model was evaluated using the same test set.

The Isolation Forest model detected 96 anomalies. This represented 13.1 percent of the test samples. The One-Class SVM also detected 96 anomalies. This represented the same 13.1 percent rate. The Autoencoder detected 74 anomalies. This represented 10.1 percent of the test samples.

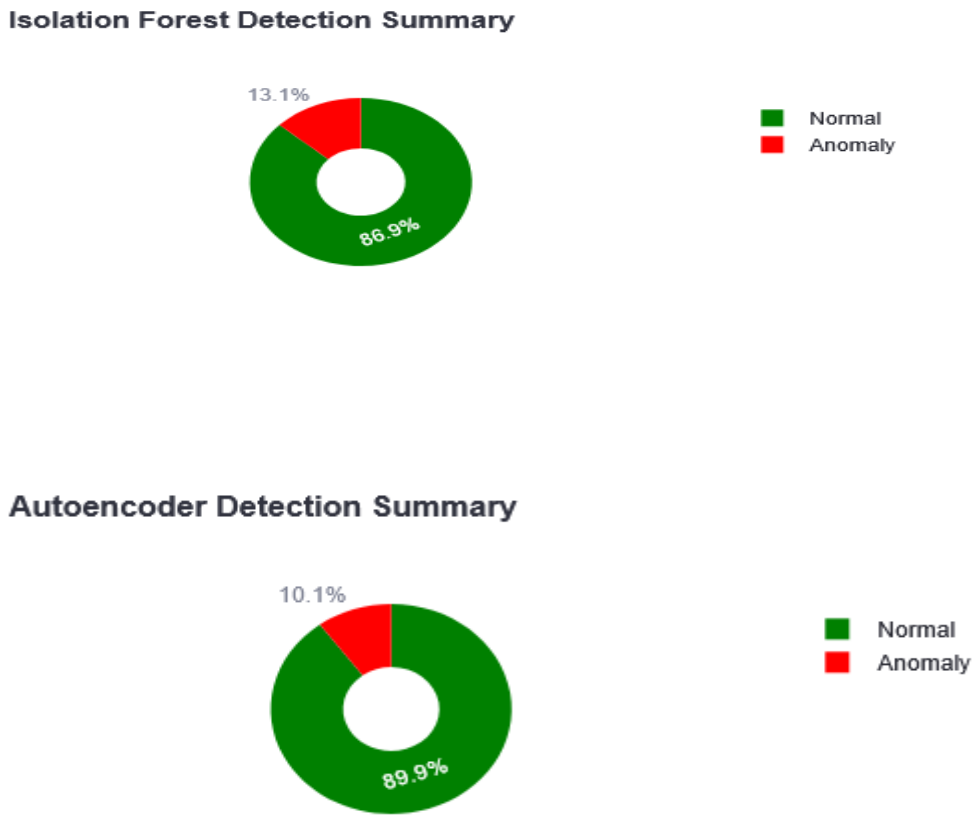

Figure 5-2. Detection Summary

The performance metrics were calculated. Isolation Forest achieved the highest F1-score. The F1-score was 0.372. The Autoencoder had an F1-score of 0.107. The One-Class SVM had an F1-score of 0.248. These scores were calculated using the ground truth labels in the dataset.

Table 5-2. ML model performance metrics (Source: AI-enhanced SPC dashboard)

Model	Accuracy	Precision	Recall	F-1 score
Isolation forest	0.709	0.656	0.259	0.372
One-class SVM	0.652	0.438	0.173	0.248
Autoencoder	0.613	0.230	0.070	0.107

Isolation Forest was chosen as the best performing model. It performed best for several reasons. The algorithm isolates anomalies through random partitioning. It does not rely on distance or density calculations. This makes it efficient for high-dimensional data. It also requires fewer assumptions about the data distribution. The isolation approach is less sensitive to outliers in the training data. This is important because manufacturing data often contains noise and outliers.

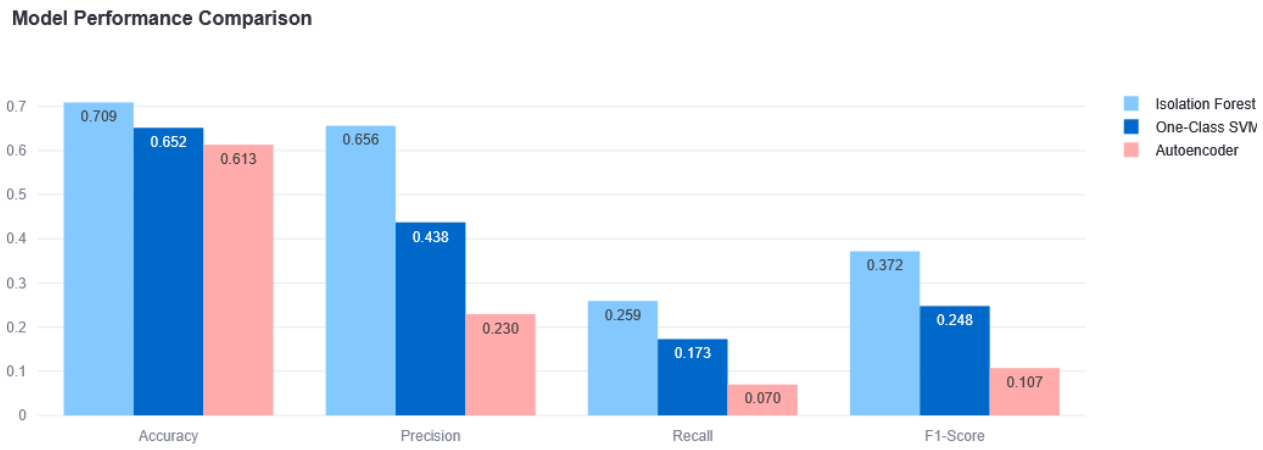

Figure 5-3. Model Performance Comparison

The 19 samples flagged by all three models are noteworthy. These samples represent the highest confidence anomalies. They were consistently identified regardless of the detection method. This agreement suggests these 19 samples are highly likely to be true anomalies. They are candidates for immediate investigation.

The ML models detected patterns that SPC did not capture. SPC identified 38 extreme deviations. ML identified 96 anomalies. Some of these 96 anomalies overlapped with the SPC violations. Others were more subtle. They represented deviations in feature combinations that SPC could not detect. This confirms that ML enhances SPC by catching complex patterns.

The Autoencoder ended up performing worse than expected, even though it offers several theoretical advantages. In general, it can learn non-linear patterns in the data and represent more complex relationships between features. It is also independent of the distance or density computation as in methods such as Isolation Forest or One-Class SVM. As a result, it is often anticipated to identify faint anomalies which other models may miss.

However, the Autoencoder performed poorly in this work. The most significant limiting factor was the training dataset size. Autoencoders are known to be very data-dependent, meaning they require large amount of data to learn a useful representation of the underlying data distribution. In small training sets the model sees too few natural variations. Therefore, it does not have the ability to learn the complete normal patterns, which leads to worse reconstruction.

In the case of the training set is about 1,700 samples, which is quite small for an Autoencoder. Generally, these kinds of models work best with thousands or even tens of thousands of training examples. Two major problems arose from having little data. First, the model did not develop a truly generalized model of normal behavior, but instead learned at least some patterns which were in large part specific to the training set. Secondly, it overfit the training data, it managed to reconstruct the training samples fairly well, but it failed to reconstruct new, unseen samples. Overfitting behaviour is a well-known problem for Autoencoders trained on small datasets.

The other aspect was the feature selection. There are seven features in the dataset and since there is usually a benefit to having more features in Autoencoders (because more inputs can also provide more information for the model to learn from). With just seven features to work with, the Autoencoder was constrained. This diminished its ability to learn intricate patterns and consequently, its probability of beating simpler techniques.

5.4 SPC vs ML Detection rate

Close comparison of detection rates of SPC and ML models should be carefully considered methodologically. In the current analysis, SPC found 38 violations out of the 200-sample subset, which is equivalent to a detection rate of 19%. At the same time, the ML models have identified 96 anomalies in the entire test set containing 732 samples, which led to a detection rate of 13.1%. Although the first observation is that the SPC method is more sensitive, this is not methodologically comparable.

Table 5-3. Traditional SPC vs ML comparison

Aspect	Traditional SPC	ML (Isolation Forest)
Sample Size	200 samples	732 samples
Detection rate	19% (38/200)	13.1% (96/732)
What it detects	Extreme, obvious violations	Complex patterns, subtle deviations
Value	Quick identifications of problems	Pattern detection, early warning

This difference in the detection rate can be explained by the fact that the sample sizes and populations that were analyzed are different. At the SPC rate a much smaller subset (n=200) is used to compute the rate, but at the ML detection all the test set is utilized. The conclusion that SPC has better performance, thus, cannot be considered statistically valid since it is made based on different sets of data. There is a possibility that the use of SPC on the entire dataset could result in a greater absolute number of violations, and this could change the relative result.

More essentially, the two methodologies are simply set up to detect various forms of process deviations. The individual data points that are out of the predefined control limits are detected by SPC, which identifies univariate outliers, extreme and obvious violations. Conversely, ML models can detect multivariate patterns in the feature space, which are complicated. These delicately, combinatorial violations might not be viewed as distinct, control chart violations and will, therefore, tend to go undetected by conventional SPC.

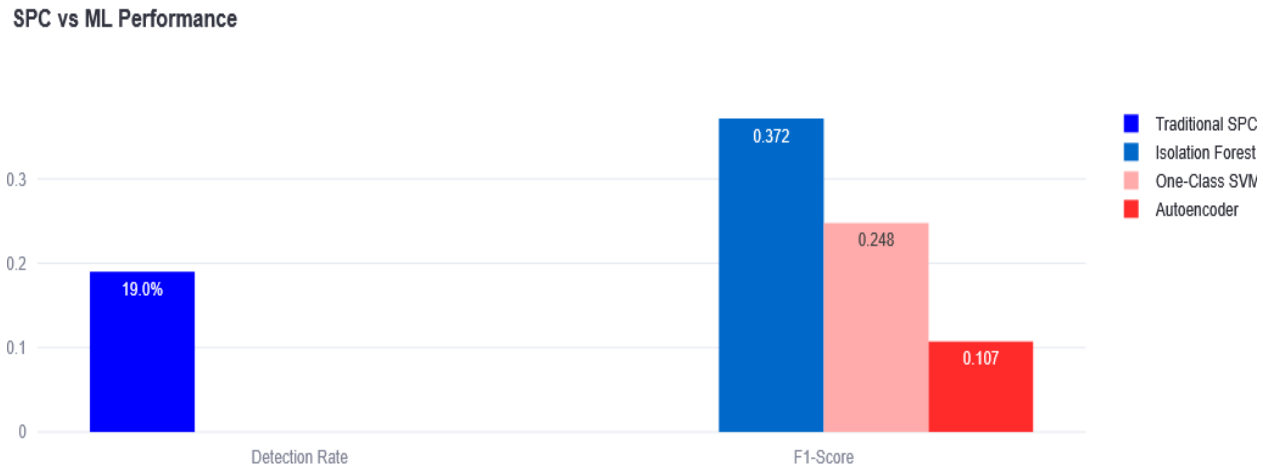

Figure 5-4. SPC vs ML Detection Rate

This combination of the two methods in a synergistic way provides the most holistic way of monitoring the processes. A clear and set minimum level of scrutiny of gross process changes is offered by SPC, with ML complementing this ability by finding subtle changes that can be harbingers of breakdown. The fact that all three ML models identified 19 critical anomalies contributes to the importance of this complementarity since those are the high-confidence anomalies, which should be corrected immediately. Altogether, the results support the assumption that traditional SPC is not yet a useless tool, yet its usefulness is increased tremendously when pattern-recognition tools of ML are employed.

5.5 Explainability Results and Methodological Adaptation

The choice of SHAP (SHapley Additive exPlanations) as the main method of model explainability is explained by the fact that the coalitional theory of games underpins the approach, which guarantees the stability and the local accuracy of feature-assigning. It was also important due to its proven usefulness in the industrial sector in building trust

in AI systems. The analysis was implemented in the Isolation Forest model that showed a better predictive ability.

Nevertheless, there was a technical hitch in the SHAP analysis and it was not possible to implement it. The error message, Found input variables of different numbers of samples: [1707, 732], showed that the training dataset (n=1707) and the test dataset (n=732) have different dimensions, which is a requirement of SHAP explainer. Although several corrective measures such as verification of data dimensions, re-training the model with different splits, and re-configuring the explainer parameters, the error still remained. There was a need to switch to a pragmatic methodological pivot considering the time constraints of the study.

As a result, the d effect size of Cohen has been used as a backup measure of explainability. The d of Cohen is a strong statistical statistic that measures the difference in two group means in a standardized manner that is independent of equal sample sizes and thereby does not fall victim to the data shape interference that hampered the SHAP analysis. It is a good, interpretable metric of feature importance, and is a viable and practical alternative.

The Cohen analysis of d showed a steady and understandable trend whereby all the seven features that were engineered had a lower mean value in the anomalous samples than it was in normal samples. The largest differences were found in the case of curve_min, curve_mean as well as curve_rms. In particular, the standard deviation of anomalies was less by 2.670 standard deviations in curve min which is a relative reduction of 1152.2%. On the same note, the standard deviations of curves indicated a decrease in the plot standard deviations of 2.157 (1448.7) and 2.154 (1436.1) by the plot mean and rms standard deviations, respectively. These large effect sizes are impressive and suggest that anomalies are typified by a considerably reduced force on a variety of quantitative characteristics.

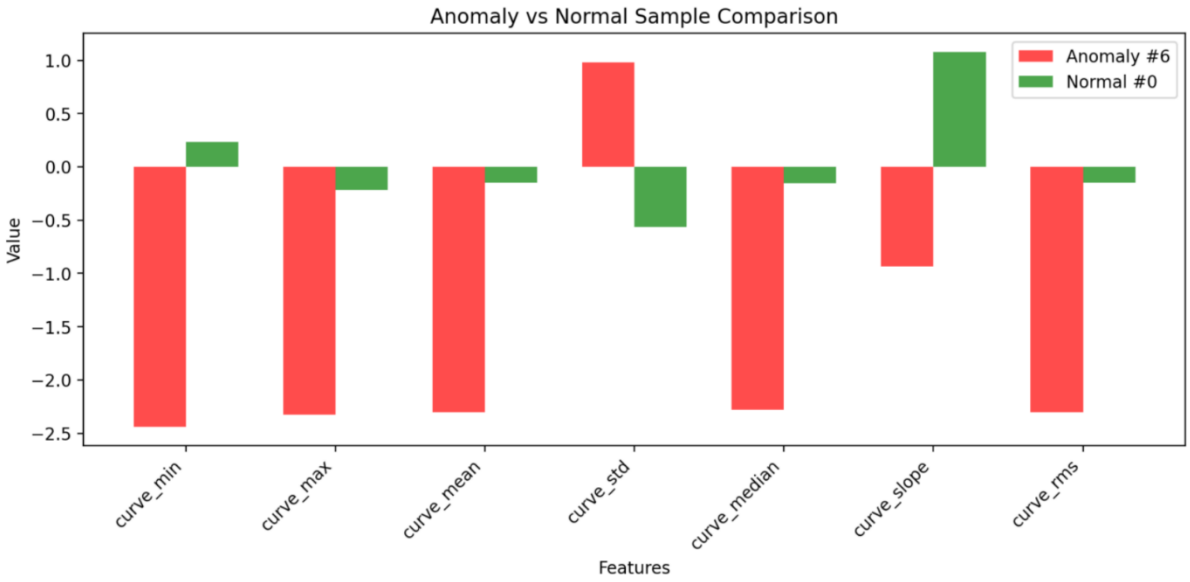

Figure 5-5. Anomaly vs Normal Sample Comparison.

A pattern analysis proved this result and produced a heatmap of unusual samples. It was the prevailing pattern: "Consistently Low: curve_min, curve_max, curve_mean, curve_median, curve_rms, curve". This steady, as opposed to random, overall flattening of all the features indicates that there is a breakdown in the functioning, indicating an underlying problem of inadequate crimping force.

The significance here is that anomalies are motivated by an entire lack of force application, which can be explained by systematic problems of incongruence in the material feeding, tools, or machine calibration inaccuracies. The rationale of changing to Cohen d was eventually warranted since it enabled the analysis to continue giving meaningful and actionable explanations and system functionality. Although the analysis of SHAP would have been more desirable due to its fine-grained local explanations, the Cohen d analysis was adequate to address the immediate research goals. Subsequent versions of this project need to focus on correcting this dimensional mismatch to allow a complete SHAP to be implemented.

5.6 Chat Assistant based on LLM Evaluation

It was tested by a series of questions asked to the hypothetical user in accordance with the dashboard context which assessed the performance of the integrated LLM chat assistant. All the answers were created with the help of the GPT-4.1-mini model that was fed with the contextual information such as the number of anomalies, the most successful model, and the SPC outcomes.

The assistant was first queried: "Are there any anomalies in my manufacturing process?" The answer was Yes, 96 anomalies detected (13.1) by Isolation Forest. Investigate tooling wear, material variability or machine set-up. This response was a precise representation of the dashboard data, properly identifying the number of anomalies and the detection rate, and complemented this data with recommendations that are actionable and relevant to the domain.

The second question, explain my X-bar chart. what does it tell me about my process quality had received the following answer: no data drawn, yet. take sample data to determine control limits. integrating SPC and ML to understand everything. This answer appropriately realized that plotted data was not provided and made a rational, situation-appropriate follow-up, indicating the need to employ an integrated monitoring strategy.

The assistant, when questioned on the topic, How to reduce variation?, replied with a multi-step, step-by-step strategy: 1. Identify special causes 2. Standardize processes 3. Maintain equipment 4. use ML insights. This output was practical and well formatted and was a combination of the old concepts of SPC as well as the new ideas of ML.

Lastly, answering the question, What specific actions can I take to improve the quality of the processes and the number of defects, the assistant answered: 1. Investigate anomalies 2. Implement SPC 3. Root cause analysis 4. Process adjustment and standardization. This reaction was uniform in its act-oriented framing with its emphasis on investigation and control of the process.

Generally, the responses of the LLM were very contextually aware, accurate, and useful. The system properly addressed the exact data points that were provided on the dashboard and provided sensible and practical guidance. This analysis proves the ability of this assistant to be a useful decision-support tool, as it can encode complicated analytical outputs into understandable, functional guidance.

6 Discussions and Conclusions

6.1 Interpretation of Findings

The findings indicate that it is possible to combine the use of traditional SPC with machine learning and explainable AI. The four-layer architecture was effectively used in practice. The data layer dealt with loading of data sets and preprocessing. The analytics engine has been used to do SPC calculations and ML model training. Interpretability outputs were produced by the explanation engine. All results were displayed in a user-friendly way.

The results of the SPC were impressive. The Cpk of 0.061 was a sign of a critically out-of-control process which is far less than the minimum of 1.0 to control than minimally acceptable capability. This was proved by the 38 violated control limits. The process mean of 0.411 is too close to the upper control limit of 0.555 with a difference of only 0.144. The process standard deviation of 0.792 and is very large compared to the specification width. These results support the necessity to improve monitoring practices.

A new dimension was provided by the ML findings. Isolation Forest identified 96 anomalies that consist of both blatant and subtle anomalies. Of special interest are the 19 critical anomalies that are all raised by all the three models. This consensus among techniques implies that these samples are, of course, almost sure anomalies which need quick research. Unless it is accompanied by ML, these 19 anomalies would have probably been not detected by the SPC alone. This result is significant since it proves that ML could detect the problems with processes, which could not be observed in traditional control charts.

The Autoencoder was not as good as thought with an F1-score of 0.107. This lousy performance can be attributed to several factors. Being roughly 1,700 samples, the training dataset is small enough that it would be better suited to an Autoencoder which in most cases needs thousands of examples to learn a reliable model of normal behavior.

This model was an overfitting to the training data, where it reproduces training examples well but collapses on test examples not seen. Also, the seven enabled features could have only given the Autoencoder the information needed to learn meaningful patterns insufficiently. This implies that Autoencoders might not be able to work with smaller manufacturing datasets or those with fewer sets of features.

The results of explainability gave an insight into the nature of the anomalies. The analysis of d between Cohen showed that the seven features were always low in the anomalies in comparison with normal samples. The most significant effect sizes were found in the case of: curve_min (2.670), curve_mean (2.157) and curve_rms (2.154) thus these are the most powerful predictors of normal behavior and anomalous behavior. The similarity of the low pattern in all features is an indication of a systematic issue, and does not imply that it happened by chance. The underlying cause is probably lack of force during crimping which could be as a result of material feeding, misalignment of tools or machine calibration. This logical arrangement contracts the scope of investigations, wasting less time and money.

The chat assistant of LLM proved to be useful. It gave context sensitive answers which used dashboard information correctly and gave actionable insights. This feature demonstrates that LLMs are able to convert technical outputs into a plain language, which can be highly advantageous in the case of technical operators who are not trained extensively. The fallback which was built on the basis of rules was a basic functionality, in case of the unavailability of an API key, so that the dashboard could be used.

Complementation of SPC and ML is interesting. SPC does a good job at identifying apparent control limit violations. ML is more appropriate in detecting the obscure patterns in combinations of features. This complementary is important as the two methods offer more comprehensive monitoring than each method. This is supported by the results.

6.2 Comparison with the Prior Work

These data are consistent with the related literature on the subject of ML in SPC. Chang & Ghafariasl, (2025) addressed the opportunities of the AI in the monitoring of the statistical process. The research showed a feasible execution of their suggestions. Isolation Forest, Autoencoder and One-Class SVM were used as per their recommendation. The findings supported the fact that ML has the potential to improve SPC capabilities.

The results are also consistent with the studies on explainable AI. Kohl, (2025) addressed the issue of trust between the operators and AI systems. The gap in this study was filled with SHAP and Cohen d. The explainability approaches ensured that the ML decisions were explainable to the operators, which is significant to acceptance. The Cohen d fallback was especially handy, as it helped to explain when SHAP had errors. This illustrates the importance of the existence of backup mechanisms in the event that main explainability tools go down.

The integration of LLDs and the studies on cognitive assistants are related. Kohl, (2025) wrote about maintenance assistants based on LLM. This paper applied that notion to quality monitoring. The chat assistant of the LLM gave natural language explanations and as such, the system could be easily accessed by the operators along with the necessity of technical training. The results indicate that LLMs are applicable to fill the communication gap between the technical systems and the operators in the shop-floor.

The literature review revealed a research gap that was addressed in the research. None of the studies combined all the four elements: traditional SPC, ML anomaly detection, XAI explainability, and LLM interface. This research bridged this gap. A combination of these components was achieved in the four-layer architecture. It was tested on actual manufacturing data and the code is open source, which means that others can build upon the work.

Comparison with the existing approaches demonstrates the importance of the integration. There are limitations of each component. The assumptions of SPC are

limited by their data assumptions. ML is opacitinuous. XAI provides transparency. LLM provides accessibility. The limitations of each of the components are taken care of by the integrated system.

6.3 Limitations

There are a number of constraints that should be recognised. The data came out of a publicly available source as opposed to a live production environment. The system was run on historical data and not in real time. This restricts the relationship of the results to actual manufacturing places.

The reason was that only univariate features have been analyzed. Multivariate trends were not explored. Other features could have been determined; the seven features were designed based on the force curves. We analyzed only one of the curves, curve min using SPC. This is a narrow area of focus which constrains the findings. Multivariate trends and more sets of features should be investigated in the future.

In SHAP method, there was a mismatch error in the dimension. The mechanism regressed to the d of Cohen which is a less-powerful and easier approach. The fallback was effective but not the best. The mismatch in dimensions came about due to the fact that Isolation Forest requires subsampling internally, and as such, the dimension expectations of SHAP have been altered. This problem should be addressed in future work to allow the whole SHAP analysis.

The LLM chat assistant will need an API key, and the key will have to be paid. This limits accessibility. The rule-based fallback will give rudimentary functionality, and all the advantages of the LLM are only offered with the API key. Some users may find this as a barrier especially in resource-constrained environment.

The system is a software prototype and not a commercial product. It is not put in a factory and not tried by actual operators. Feedback of the user is not taken. The

practicality and usability is yet to be tested in a real-life scenario. Users studies should also be incorporated in future to determine the effectiveness of the system in practice.

The explanations of LLM were not verified separately. The subjective evaluation of the explanations was done. It would be beneficial to have a more formal assessment, which could include questionnaires or operator interviews. This would give a more stringent test of the effectiveness of the LLM.

6.4 Future Research

There are a number of future research directions that are promising. The mismatch between the SHAP dimension needs to be addressed to facilitate SHAP analysis. This would involve knowing the reason behind the mismatch and putting in place a workaround. The dimension problem could be prevented by the use of a sampled background dataset. SHAP offers a more in-depth account of the importance of features and would enhance the explainability of the system.

The system can be implemented in actual production set up. This would put its performance into real-time and will give the feedback to the users. The feedback might be utilized to enhance the system and the system might be evaluated in terms of usability by surveying the operators. An early trial run in a regulated manufacturing environment would be a good next step.

It is possible to test additional ML models. Possible candidates that can be more effective on certain datasets include XGBoost, CNN and LSTM. A comparison to the existing models would be an insight to the most appropriate models to manufacture data. The optimization of hyperparameters may also be considered as a way of enhancing performance of the models.

Integrated multi-modal data might be provided. Images, audio, or vibration data could be considered as force curves. This would give a more comprehensive picture of the

process and subject the system to more complex data. It would make the system more versatile since it would offer the ability to deal with various types of data.

The fault-tree analysis would be completely incorporated. This could be an optional goal that was partially discussed by using LLM root cause suggestions. Some root cause analysis would be much more structured in a full fault-tree, which would be beneficial with a complicated manufacturing process. This would help the system to supply more detailed diagnostic information. Transfer learning might be considered. The models were also able to fit into other manufacturing processes and the system was more universally applicable as its retraining would not be required. This would involve research on the degree of generalization of the models to other production settings.

6.5 Conclusion

In this thesis, a SPC dashboard that uses AI to enhance the quality of manufacturing was developed. The system incorporated the conventional SPC with ML anomaly detection, explainable AI, and an LLM interface. The four layered architecture was implemented successfully. Crimp Force Curves data set was used to validate the system.

Their findings indicated that the system is effective in detecting anomalies. The 38 violations that SPC identified had a Cpk of 0.061 which means that the process was critically out of control. ML identified 96 anomalies, and Isolation Forest was the best. The three models flagged nineteen samples, which are the most confident anomalies. The hybridization of SPC and ML was proved. SPC identifies clear differences, whereas ML detects subtle patterns controlling charts that are not detected by control charts.

The methods of explainability gave an insight into the anomalies. SHAP experienced a problem with dimension mismatch and thereby also used the d of Cohen as a default. Cohen d analysis indicated that all the features had a constant lower value in anomalies indicating a systematic issue. This was probably caused by a lack of force when crimping. Such an event proves the usefulness of explainability in detecting root causes.

Natural language explanations were offered by the LLM chat assistant. It has rightly cited the dashboard information and provided practical suggestions. This made the system operator-friendly and minimized the technical training. Technical precision and easy communication are a combination that is significant to adopt in practice.

The thesis makes three contributions. It shows the initial combined SPC-ML-XAI-LLM quality manufacturing platform. It offers an open-source software prototype of quality control education. It uses three machine learning models to compare on actual manufacturing data. These works contribute to the development of the sphere of smart process control.

The results substantiate that ML complements SPC with the ability to identify intricate trends. The unified system fills the gap of trust between the human operators and AI systems. The future of quality manufacturing is in human expertise and AI power, which necessitates explainable and accessible intelligent systems. The open source code can be used in the future research and education. The system can be reconfigured to other manufacturing processes. This has been the groundwork in making more developments of smart process control.

References

Adadi, A., & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., … Zoph, B. (2024). *Open AI (2023). GPT-4 Technical Report* (arXiv:2303.08774). arXiv. https://doi.org/10.48550/arXiv.2303.08774

Ali, T., & Kostakos, P. (2023). *HuntGPT: Integrating Machine Learning-Based Anomaly Detection and Explainable AI with Large Language Models (LLMs)* (arXiv:2309.16021). arXiv. https://doi.org/10.48550/arXiv.2309.16021

Alwan, S. J., Khadim, R. J., & Alsaedi, H. (n.d.). Next-Generation Hybrid Control System: Integrating Modern Statistical Process Charts and Advanced AI for Autonomous Manufacturing. *JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES*.

Anusha, V., George, L., Harish, K. S., B, Y., & Manimaran, A. (2025). XAI-SPCNET: A MULTI-LEVEL EXPLAINABLE AI FRAMEWORK FOR PREDICTIVE CONTROL CHART AUTOMATION IN MANUFACTURING QUALITY MONITORING PROCESS. *Reliability: Theory & Applications*, *20*(4 (89)), 381–398.

Chang, S. I., & Ghafariasl, P. (2025). *A Review of Artificial Intelligence Impacting Statistical Process Monitoring and Future Directions*.

Gentil, M.-H., & Merle, C. (2026). QUALIBOT: AI assistant for implementing quality management systems in the transition to Enterprise 4.0 and Quality 4.0. *International Journal of Quality & Reliability Management*.

Hodge, V. J., & Austin, J. (2004). *A Survey of Outlier Detection Methodologies*.

Hsieh, W., Bi, Z., Jiang, C., Liu, J., Peng, B., Zhang, S., Pan, X., Xu, J., Wang, J., Chen, K., Feng, P., Wen, Y., Song, X., Wang, T., Liu, M., Yang, J., Li, M., Jing, B., Ren, J., … Liang, C. X. (2024). *A Comprehensive Guide to Explainable AI: From Classical Models to LLMs* (arXiv:2412.00800). arXiv. https://doi.org/10.48550/arXiv.2412.00800

Hu, Y.-S., Marandi, S., & Modarres, M. (2026). DML–LLM Hybrid Architecture for Fault Detection and Diagnosis in Sensor-Rich Industrial Systems. *Sensors*, *26*(6), 2008. https://doi.org/10.3390/s26062008

Jansen, M., & Pehlke, M. (2025). *Increasing AI Explainability by LLM Driven Standard Processes* (arXiv:2511.07083). arXiv. https://doi.org/10.48550/arXiv.2511.07083

Juran, J. M., & Godfrey, A. B. (1999). *Juran's Quality Handbook* (5th ed.). McGraw-Hill. https://www.academia.edu/3048123/Jurans_Quality_Handbook

Kohl, L. (2025). *Knowledge-based maintenance framework for smart manufacturing: Advancing efficient planning and cognitive assistance to enhance system availability* [Thesis, Technische Universität Wien]. https://doi.org/10.34726/hss.2025.117579

Lipton, Z. C. (2018). The Mythos of Model Interpretability. *Machine Learning*.

López-Pernas, S., Song, Y., Oliveira, E., & Saqr, M. (2026). LLMs for Explainable Artificial Intelligence: Automating Natural Language Explanations of Predictive Analytics

Models. In M. Saqr & S. López-Pernas (Eds.), *Advanced Learning Analytics Methods: AI, Precision and Complexity* (pp. 261–286). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-95365-1_11

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions* (arXiv:1705.07874). arXiv. https://doi.org/10.48550/arXiv.1705.07874

Marchesano, M. G., Mattera, G., Guizzi, G., Santillo, L. C., & Converso, G. (2025). Explainable Deep Reinforcement Learning Enhancing Industrial Maintenance. *IFAC-PapersOnLine, 11th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2025*, *59*(10), 524–529. https://doi.org/10.1016/j.ifacol.2025.09.090

Mayer, J., & Jochem, R. (2024). Capability Indices for Digitized Industries: A Review and Outlook of Machine Learning Applications for Predictive Process Control. *Processes*, *12*(8), 1730. https://doi.org/10.3390/pr12081730

Megahed, F. M., Chen, Y.-J., Zwetsloot, I., Knoth, S., Montgomery, D. C., & Jones-Farmer, L. A. (2024). Introducing ChatSQC: Enhancing Statistical Quality Control with Augmented AI. *Journal of Quality Technology*, *56*(5), 474–497. https://doi.org/10.1080/00224065.2024.2372328

Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, *267*, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

Ojuolape, A. E., & Hu, S. (2024). Explainable Fault Diagnosis of Control Systems Using Large Language Models. *2024 IEEE Conference on Control Technology and Applications (CCTA)*, 491–498. https://doi.org/10.1109/CCTA60707.2024.10666634

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Psarakis, S., & Papaleonida, G. E. A. (2007). SPC Procedures for Monitoring Autocorrelated Processes. *Quality Technology & Quantitative Management*, *4*(4), 501–540. https://doi.org/10.1080/16843703.2007.11673168

Shewhart, W. A. (1931). *Economic control of quality of manufactured product*. Van Nostrand Company.

Woodall, W. H., & Montgomery, D. C. (2009). Some Current Directions in the Theory and Application of Statistical Process Monitoring. *Journal of Quality Technology*, *46*(1), 78–94. https://doi.org/10.1080/00224065.2014.11917955

Wuest, T., Irgens, C., & Thoben, K.-D. (2014). An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *Journal of Intelligent Manufacturing*, *25*(5), 1167–1180. https://doi.org/10.1007/s10845-013-0761-y

Appendices

Appendix 1. System Source Code and Installation guidelines

GitHub Access

The full implementation of the AI SPC dashboard is available at:

Repository URL: *https://github.com/ragibaustbd-hub/AI-SPC-Dashboard*

System Specification

Requirement	Minimum
Python	3.9, 3.10, 3.11, or 3.12
Ram	8 GB
Storage	2GB free space
Operating System	Windows 10/11, macOS, or Linux
Browser	Chrome, Firefox, or Microsoft Edge etc.

Installation Guide

Step 1: Copying the Repository

```bash
git clone https://github.com/ragibaustbd-hub/ai-spc-dashboard.git
cd ai-spc-dashboard
```

Step 2: Creating Virtual Environment (Suggested)

```bash
Python m venv
```

```
Activate on Windows:
venv\ Scripts\ activate
Activate on macOS/Linux:
source venv/ bin/ activate
```

Step 3: Installing Dependencies

```bash
pip install r requirements.txt
```
If requirements.txt is not available, install manually:
```bash
pip install  streamlit pandas numpy plotly scikit learn tensorflow
shap openai matplotlib scipy
```

Step 4: Placing Dataset

Get the Crimp Force Curves dataset and put it in the project folder as "crimp_force_curves.pkl". When the dataset is not available, the dashboard generates synthetic data.

Step 5: Run Dashboard

```bash
streamlit run app.py
```

Step 6: Accessing Dashboard

It will automatically direct or start browser and navigate to: *http://localhost:8501*

Dashboard Navigation

The built dashboard has four tabs:

SPC Charts (Control charts (X bar, R, EWMA, I MR)) →

ML Anomaly Detection (Isolation Forest, Autoencoder, One Class SVM) →

SHAP Explanations (Feature importance analysis (SHAP or Cohen's d fallback)) →

AI Chat (Natural language questions about process status)

OpenAI API Key Setup (Optional)

To use GPT enabled chat answers:

1. Sign Up at https://platform.openai.com

2. Add at least 5$ funds

3. Generate the API code at platform.openai.com/api keys

4. Paste the code in the dashboard sidebar (Tab 4)

The dashboard has a rule based fallback when no API key is specified.

Troubleshooting

Issue	**Solution**
ModuleNotFoundError	Run pip install requirements.txt
Dataset not found	Select Sample Data (Demo) in sidebar
Port 8501 already in use	Run streamlit run app.py and server.port 8502`
SHAP error	System auto falls back to Cohen's d
OpenAI API error	Check key validity or use rule-based mode

Files Submitted

File	**Purpose**
app.py	Main Streamlit dashboard
crimp_force_curves.pkl	Dataset (optional)
requirements.txt	Python dependencies
README.md	Project description

Appendix 2. Hyperparameters and Model Configurations

Isolation Forest Configuration

Parameter	Value	Justification
n_estimators	100	Standard ensemble size
max_samples	‘auto’	Uses all training samples
contamination	0.10	Matches expected anomaly ratio
random_state	42	Fixed seed for reproducibility
n_jobs	-1	Uses all CPU cores
bootstrap	False	Deterministic results

Code Implementation

```
```python
                    if use isolation forest:
                        iso_forest=       Isolation       Forest
(contamination= contamination, random state= 42, estimators= 100)
                        iso_forest.fit(X_train)
                        y_pred_if = iso_forest.predict(X_test)
                        results [Isolation Forest] = (y_pred_if
== -1).as type(int)
                        model objects [Isolation Forest] =
iso_forest
                        anomaly scores [Isolation Forest] = -
iso_forest.score_samples(X_test)
```
```

One-Class SVM Configuration

Parameter	Value	Justification
Kernel	‘rbf’	Captures non-linear boundaries
Nu	0.05	Upper bound on training errors
Gamma	‘auto’	Automatic kernel coefficient
Random_state	42	Fixed seed for reproducibility

Code Implementation

```
```python
if use one class svm:
                        oc_svm = One-Class SVM (nu=contamination,
kernel=rbf, gamma=auto)
                        oc_svm.fit(X_train scaled)
                        y_pred_svm = oc_svm.predict(X_test scaled)
                        results[One-Class SVM] = (y_pred svm == -
1).astype(int)
                        model objects[One-Class SVM] = oc_svm
                        anomaly scores[One-Class SVM] = -
oc_svm.decision_function(X_test_scaled)
```
```

Autoencoder Architecture

Layer	Type	Neurons	Activation	Dropout
Input	Dense	7	-	-
Hidden 1	Dense	32	ReLU	0.2
Hidden2	Dense	16	ReLU	0.2
Hidden 3	Dense	8	ReLU	-

Hidden 4	Dense	16	ReLU	0.2
Hidden 5	Dense	32	ReLU	0.2
Output	Dense	7	Linear	-

Code Implementation

```
```python
if use autoencoder:
                        input dim = x_train scaled. Shape [1]
                        encoding dim = max (2, input dim // 2)

                        autoencoder = Sequential ([
                            Dense (encoding dim * 2, activation=
relu, input_shape= (input dim,)),
                            Dropout (0.2),
                            Dense (encoding dim, activation=relu),
                            Dense    (encoding    dim    *    2,
activation=relu),
                            Dense (input dim, activation=linear)
                            Dense (input dim, activation=linear)
])
```
```

Autoencoder Training Parameters

Parameter	**Value**	**Justification**
Learning rate	0.001	Adam default
Batch size	32	Balance of speed and stability
Epochs	50	Maximum (early stopping halts earlier)
Loss function	MSE	Reconstruction error for anomaly detection
Optimizer	Adam	Adaptive learning rate

Early shopping	Patiene=5	Prevents overfitting

Code Implementation

```
'''python

autoencoder. Compile (optimizer= adam, loss= mse)

early stop = Early stopping (patience= 5, restore best weights=True)

autoencoder. Fit(X_train scaled, x_train scaled, epochs= 50, batch size= 32,

validation split= 0.2, callbacks=[early stop], verbose= 0)

reconstructions = autoencoder. Predict(x_test_scaled, verbose= 0)

mse = np.mean((x_test_scaled - reconstructions) * 2, axis= 1)

threshold = percentile(mse, (1 - contamination) * 100)

results [Autoencoder] = (mse > threshold). as type(int)

model objects [Autoencoder] = autoencoder

anomaly scores [Autoencoder] = mse

'''
```

Training Parameters

Parameter	**Value**
Train-Test Split	70%30%

Random seed	42
Test set size	732 samples
Training set size	1707 samples
Expected Anomaly ratio	10% (configurable via slider)
Feature Scaling	StandardScaler (mean=0, std=1)

Code Implementation

```
'''python
stratify = y_true binary if ground truth available else None
                    x_train,    x_test,    y_train,    y_test    =
train_test_split(
                        x,    y_true_binary    if    ground    truth
available else None,
                        test_size=test  size,  random  state=42,
stratify=stratify
                    )
                    scaler = StandardAero ()
                    x train_scaled = scaler.fit transform(x_train)
                    x test_scaled = scaler.transform(x_test)

                    results = {}
                    model objects = {}
                    anomaly scores = {}
'''
```