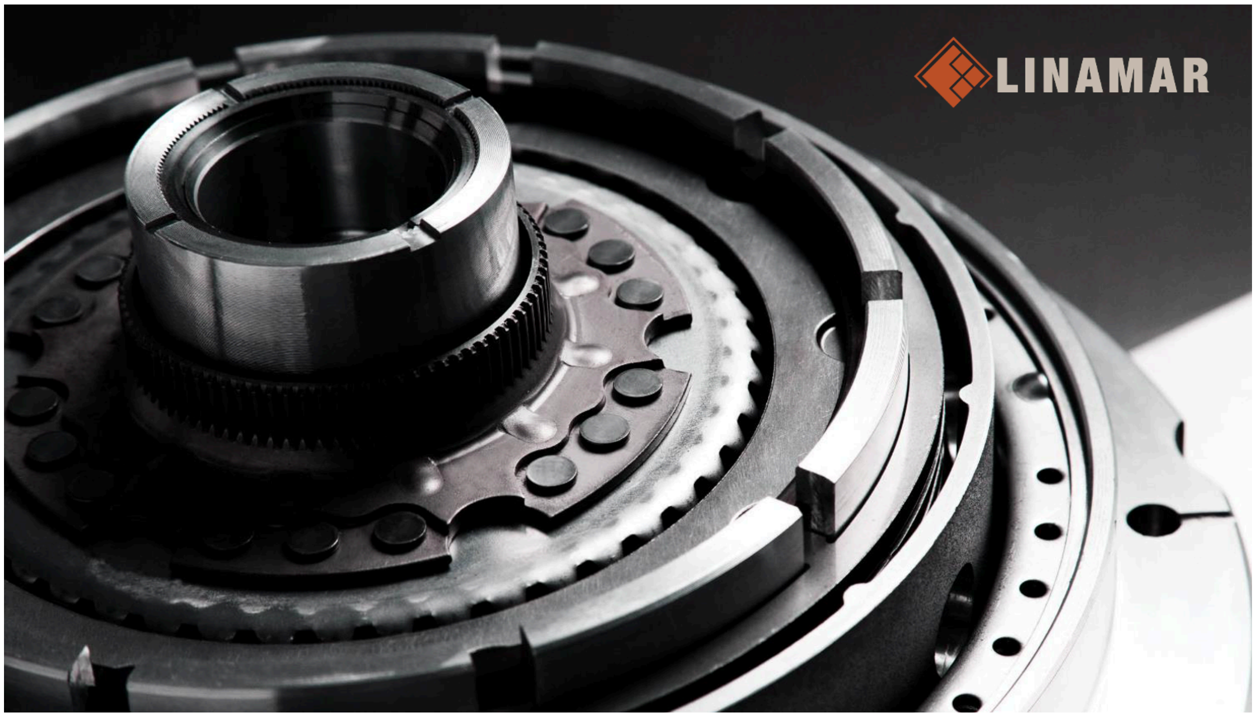


Enhancing Linamar's Big Data Analytics Capability for Continuous Improvement in Production

Student:

Chen Zhang

University Advisor:

Sheng Yang

Linamar Advisor:

Luis Bravo

Brandon Sheldrake

Table of Contents

Abstract

The Linamar Manufacturing Monitoring System (LMMS) serves as a comprehensive data collection infrastructure spanning 11 countries, gathering data from 5600 production machines. This project is in collaboration with LMMS, aiming to harness historical data and develop advanced analytical tools to optimize production line utilization. (Our company, 2023). The core focus lies in enhancing historical data quality through the establishment of a reference table for data consistency across diverse production lines. This endeavor includes devising an adaptive outlier detection framework tailored to different machines and production lines, alongside addressing missing value challenges.

Furthermore, the project entails a comparative analysis of two akin production lines, Hastech and Linergy, engaged in manufacturing identical products. The objective is to discern differentiating attributes that offer enriched insights to Linamar. This analysis leverages techniques such as Linear Discriminant Analysis and T-test to extract meaningful distinctions. Another crucial facet of this project involves devising an alert system for updating designed cycle times, which augments precision in operational utilization for business enhancements. By realizing these objectives, the collaboration with LMMS strives to achieve improved historical data quality, nuanced comparative analysis, and enhanced operational efficiency. The outcomes of this project exemplify the significance of data-driven methodologies in manufacturing contexts and underscore the role of rigorous data maintenance in driving operational excellence.

Introduction

Linamar corporation is a manufacturing company of engineered products powering vehicles, mot ion, work, and lives. They have more than 26,000 employees in 65 manufacturing locations, and headquartered in Guelph, Ontario. This collaborative project partners with the Linamar

Manufacturing Monitoring System (LMMS) team, a machine-centric data collection infrastructure. The initiative aims to capitalize on historical data, devising robust analytical methods to harness the data's potential. Despite an abundance of raw data, the data's low quality has been acknowledged, prompting the LMMS team to embark on enhancing historical data quality. This initiative serves the dual purpose of facilitating deeper analysis and supporting the development of advanced machine learning models.

Structured as a comprehensive, multi-phase endeavor, the project encompasses three pivotal stages:

1. ***Raw Data Quality Improvement***: This phase centers on enhancing the raw data quality. Recognizing the significance of data integrity, the LMMS team endeavors to cleanse, standardize, and enrich the existing data, thereby laying a strong foundation for subsequent analyses.
2. ***Comparison Analysis of Similar Production Lines:*** The second phase delves into a comparative analysis of akin production lines. Focusing on the Hastech and Linergy lines, which manufacture identical products, this stage aims to unearth distinguishing features, offering insights that contribute to informed decision-making and operational optimization.
3. ***Machine Learning Algorithm Development***: Anchored in the third phase is the development of advanced machine learning algorithms. Leveraging the enriched data, this stage aspires to create predictive models, enabling proactive decision-making and process optimization within Linamar's manufacturing operations.

As a long-term venture, this collaboration underscores the commitment to sustained improvements in historical data quality and analytical prowess. By aligning with Linamar's goals, this project exemplifies the transformative potential of data-driven methodologies in the manufacturing sector, paving the way for refined insights and enhanced operational efficiencies.

Main Problems

This progress report will mainly focus on the result and the success of the first two phases, also for the ideally design of the next phase. For the phase one, there are three problems are aimed to solved on Linamar. Which are:

1. ***Inconsistency problem***: As introduced, there are over 50 manufacturing factories all over the world. There might be some inconsistency problem happened during the data collection process, so the project is aimed to create an reference table as a result.
2. ***Missing value problem***: Since the data we used is hourly recorded data from each production line, so there might be some missing value. We aimed to refill the missing value of find the explanation of why those missing value occurs.
3. ***Outlier problem***: There should have some outliers, then we designed an adjusted outlier detection system based on different machines.
4. ***Designed Cycle Time problem***: The designed cycle time sometimes cannot accurately describe the running speed and running conditions of the machine. On the contrary, most of the designed cycle time in the historical data is quite different from the actual cycle time. Therefore, Linamar hopes to establish an early warning system to prompt the factory to recalculate the designed cycle time.

For the comparative analysis, we will conduct an exhaustive examination to contrast Hastech and Linergy production lines which both produce sun gear shaft by similar design of the line. This will allow us to identify differentiating attributes between the two lines, providing valuable insights that will guide our actions in the final project phase.

Background

In Linamar's operational framework, each factory comprises multiple production lines, each equipped with a variety of machines. These machines encompass diverse functions such as drilling, turning, washing, laser marking, and more. To offer a clearer insight into the production line structure, a visual diagram is presented in the *Appendix A*. This diagram, illustrating a segment of the Linergy production line, showcases the interconnected nature of machines. Notably, the sequential arrangement allows for a seamless transfer of parts from one machine to the next.

Within each factory, a distinction is made between two machine types: value-added and non-value added. Value-added machines undertake specific tasks on parts, such as hole drilling, heat treatment, or cutting. Conversely, non-value-added machines serve the role of transferring parts between different stages of the production process. The diagram in the Appendix highlights this demarcation - non-value-added machines are denoted by blue spots, while value-added machines are represented by other symbols. This delineation of machine functions underscores the intricate orchestration required to optimize production processes and enhance operational efficiency within Linamar's manufacturing ecosystem.

Furthermore, each machine is equipped with its own array of sensors and gauges, seamlessly integrated into the Linamar Manufacturing Monitoring System (LMMS). This integration ensures real-time data capture, encompassing a comprehensive spectrum of machine statuses and pertinent information. Predominantly, machines operate within one of the following six distinct states at any given moment:

1. **Running**: The machine is actively engaged in its operation, effectively processing parts in accordance with established parameters.

2. **Blocked**: Upon completion of its task, the machine encounters an impasse where no further capacity exists for output storage, temporarily halting the workflow.
3. **Starved**: Despite its operational readiness, the machine is momentarily idle, awaiting the arrival of parts due to a delay in preceding processes.
4. **Down**: The machine is temporarily offline due to maintenance, repair, or an unforeseen event, rendering it temporarily non-operational.
5. **Offline**: The machine is disconnected from the internet and the central system, causing it to be temporarily isolated from the network.
6. **Over-cycle**: The machine's actual cycle time surpasses the predefined designed cycle time, signifying a deviation from the anticipated operational timeline.

This detailed categorization of machine states underlines the sophisticated monitoring and management enabled by LMMS, contributing to the optimization of Linamar's production processes, and facilitating proactive maintenance and decision-making.

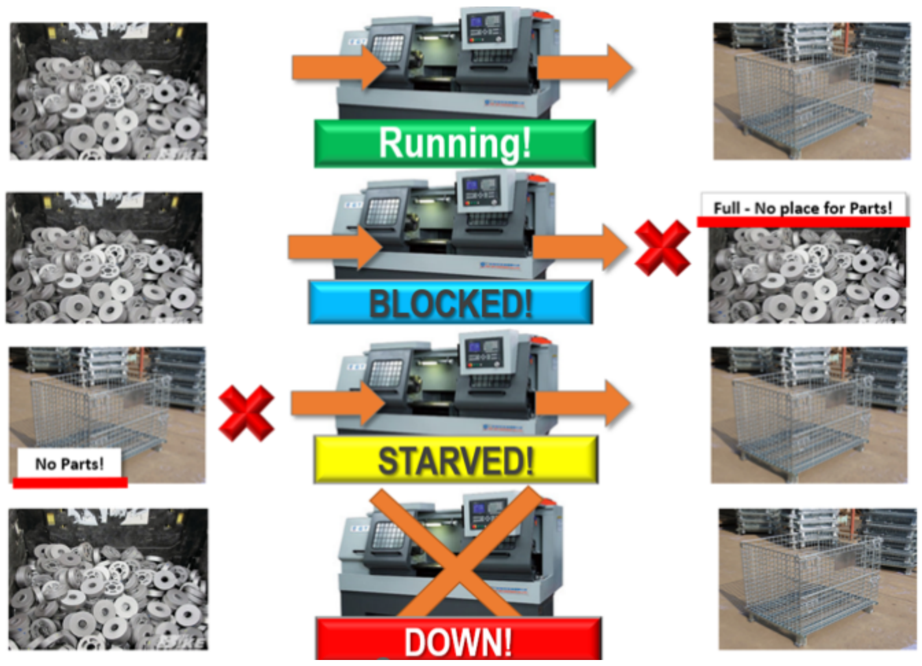


***Figure 1**: Different statues of machines in production line*

Data Summary

To augment the report's comprehensiveness, we have furnished a foundational overview in this section. The dataset at our disposal comprises historical data captured on an hourly basis, encompassing nearly 30 distinct features for each recorded observation. *Table 1*, presented below, showcases a subset of pivotal features extracted from the dataset, accompanied by concise explanations elucidating the significance of each feature. This tabulation provides a preliminary glimpse into the richness of the data and its potential to yield insights that foster informed decision-making and operational optimization.

Feature	**Explanation**	**Example**
PLA_ID	ID represent each plant, each machine, and each operation, first three digit represent the plants, the next two represents the line, the next three represents the operation, and the last two represents the machine.	'0300101001' means 030 plant, 01-line, 010 operation, and 01 machine
Asset_Type	The type of the asset as two labels: machine or automation	
Asset_Make	The company that makes those assets	Doosan, Alfing
Asset_Capability	The description of the capability of the asset model	HLathe, LaserMark
Shift_Name	Different shift name and the shift number, only three types	Shift 3: Night Shift
Date_Time	The time of record each observation	
Running_Time	How long the machine is working well during the observe hour	1365, 2039
Designed_Cycle_Time	Designed cycle time can describe the machine	
Average_Cycle_Time	the good parts produced by that machine during the running time of the recorded hour	

Table 1*: Important features in historical data used in this project for the analysis.*

In this project, our analysis hinges upon a comprehensive set of historical data, where a selection of noteworthy features is presented in *Table 1*. The dataset comprises two distinct types of data: continuous features and categorical features. The continuous features describe machine states over the preceding hour, encompassing parameters such as running time, downtime, starved time, designed cycle time, average cycle time, utilization, and so on. These metrics collectively offer an intricate portrait of machine performance and operational efficiency. *Figure 2*, displayed below, presents a side-by-side bar plot illustrating the distribution of three pivotal categorical features across the two production lines. Notably, Hastech showcases a configuration of 10 diverse machines hailing from 10 distinct manufacturers, whereas Linergy boasts 19 distinct machines originating from 16 unique companies. Intriguingly, a comparison across the two production lines reveals the presence of six identical machines, albeit positioned in separate lines. The forthcoming section will delve into a comparative analysis of these machines' utilization, providing insights into their efficiency and operational impact.

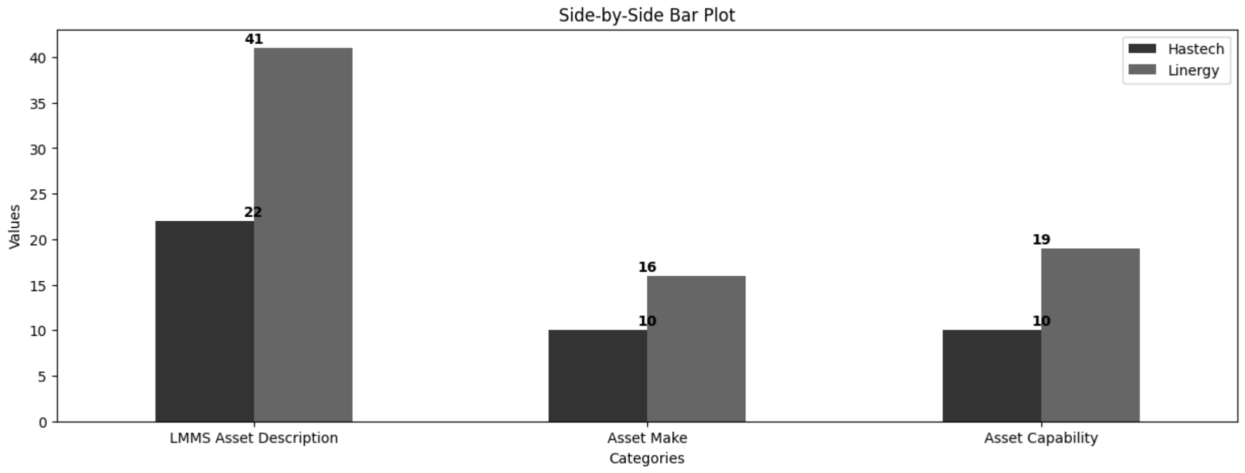


Figure 2: *Side-by-side bar plot to summary three categorical features over Hastech and Linergy*

Methodology

For data extraction, I used SQLAlchemy library in python to pull the last three months' data both in Hastech and Linergy. SQLAlchemy provides a nice way of interacting with the databases, we can leverage the Python framework of SQLAlchemy to streamline the workflow and more efficiently query the data. Furthermore, the comprehensive data cleaning process was executed within the VsCode IDE environment, integrated with the power of Python scripting. This enabled us to effectively navigate and transform the dataset, ensuring its readiness for subsequent analysis.

To better detect the outlier, we combined the designed cycle time and regular z-score interval as an adjusted z-score interval based on different machines. Basically, this idea is to see where the designed cycle time is compared to z-score interval. So as a result, we come up with three possible cases that may have for this adjusted z-score interval. More detailed explanation will cover in the later section.

For the subsequent phases of analysis, I employed linear discriminant analysis (LDA) as a tool to discern distinctive features between the two production lines (P. M., & T. B. 2013). In essence, LDA is a statistical approach used to identify a linear combination of attributes that effectively differentiate multiple classes or groups. By implementing LDA, I sought to extract the most influential features that substantially contribute to the differentiation of the two production lines. Building on the insights gained from LDA, my focus shifted to comparing the utilization of identical machines across the two production lines. This comparative evaluation aimed to determine the relative efficiency of machines situated within each line. Underlying this analysis is the assumption of data independence between the two production lines. To facilitate this comparison, I harnessed the independent one-sided T-test (Kim, T. K., 2015). This statistical method involves assessing whether a given dataset significantly differs from an expected value, considering only one direction (greater than or less than). This pragmatic approach enabled me to ascertain evidence of disparity in machine efficiency and draw meaningful conclusions based on observed differences.

For better compare the production line and for further alter system design, I used root mean square distance as a measurement, the formula of the distance is showing below:

$$\sqrt{\sum_{i=1}^{n} (A_i - D_i)^2}$$

Where A_i represent the average cycle time of each observation, and D_i represents the designed cycle time. By computing the distance could give us more information about the operations and provide more accurate support to the commercial sector. By employing these practical analytical techniques, I aimed to derive actionable insights that could inform operational decisions and drive optimization within Linamar's manufacturing processes.

Data cleaning results

Reference Table

As mentioned earlier, certain categorical features in the database displayed disparities, leading to data inconsistencies between the production lines. To promote consistent analysis, we developed a reference table concentrating on two main categorical factors: asset capability and asset make. This reference table, as presented in *Table 2* below, acts as a cohesive structure aligning the asset capabilities across Hastech and Linergy. From table, we can see some typical examples, such as MarkLaser used in Hastech and LaserMark used in Linergy, Wash in Hastech and Washer in Linergy, or SplineRoll in Hastech and RollSpline. From those examples, we can see most of the cases describes the exact same operations but using different names. This pivotal measure simplifies subsequent analyses by creating a common ground for comparison, streamlining the overall process. Also, I created a whole version of the reference table based on almost all the factories from the database to provide convenience to Linamar.

Hastech	**Linergy**	**Final Name**
MarkLaser	LaserMark	LaserMark
SplineRoll	RollSpline	RollSpline
Wash	Washer	Washer
PressBush	PressBushing	PressBushing
HtTreat	HtTreatInduct	HtTreatInduct
Grind	Grinder	Grinder

Table 2*: Reference table for different asset capability name in both lines*

The *Table 3* below, showing the result reference table of asset make companies between two production lines. As we can see the mismatched problem just due to some typo or different format, for example, in Hastech they used DOOSAN, but in Linergy they used Doosan in the regular format. So, keep its consistency can improve the historical data quality.

Hastech	**Linergy**	**Final Name**
DOOSAN	Doosan	Doosan
Anderson-Cook	Anderson_Cook	Anderson_Cook
ANDEC	Andec	Andec
OKUMA	Okuma	Okuma
IMPCO	Impco	Impco
KEYENCE	Keyence	Keyence
MARPOSS	Marposs	Marposs
KOYO	Koyo	Koyo
THE CENTRE	The Centre	The Centre
SIC	SIC AUTOMATION	SIC AUTOMATION

Table 3*: Reference table for different asset make companies' name in both lines.*

Missing Value Explanation

After I solved the mismatched problem, I found that there are still lots of missing values happening for the average cycle time feature. In my original thought, once I found out why those missing values occur, then maybe I can recalculate the average cycle time. So, the first step I did is a basic summary, in Hastech there are 13763 missing values for average cycle time during the last 3 months, and there are 46048 missing values in Linergy. After a thorough analysis of the missing values, a recurring pattern emerged concerning two key features: average cycle time and average load time. Notably, these gaps in data were observed across both factories and could be attributed to five distinct scenarios. These scenarios encompass instances where the machine is inoperative (down), encounters a bottleneck (blocked), loses online connectivity (offline), awaits input (starved), or is actively running but still registers missing values. *Table 4* shown below represent the counts and percentage of each case during the last 3 months.

Machine Statues	**Hastech**	**Linergy**
Down	11283 (82.0%)	25037 (54.4%)
Offline	1070 (7.8%)	9739 (21.1%)
Starved	743 (5.4%)	6436 (14.0%)
Blocked	740 (5.4%)	4878 (10.6%)
Running	188 (1.4%)	612 (1.3%)

***Table 4**: The summary of each machine's status of those missing values occurs.*

From *Table 4*, we can see that for both production lines, most of the missing values occur when the machine is down, there are 82% in Hastech and 54.4% in Linergy. The summary I found is totally reasonable, since once the machine is down then no parts will be produced by that machine, then there is no way to calculate the average cycle time of that machine. To provide a more detailed summary, I also created a summary for the same machine of different production lines to see what happened between those machines. The Table in *Appendix B* can provide more detailed summary based on different machine, most of the machine is just like we mentioned before, which is due to the machine is down then they may have missing value for average cycle time.

Outlier Detection System

As we mentioned before, there could be some outliers for the average cycle time maybe due to the typo or some wrong computation. It is important to develop an outlier detection system for this kind of problem. Due to differences between the machines, we should design an outlier detection system based on each machine. Basically, I combined the designed cycle time of each machine and the z-score interval of each machine. For the z-score interval, I used the [μ-3σ,μ+3σ] for each machine. Generally, there are only three different cases for the outlier detection system. For convenience, I will use D to represent the designed cycle time.

1. *Case I: Designed cycle time less than the lower bound of the interval*

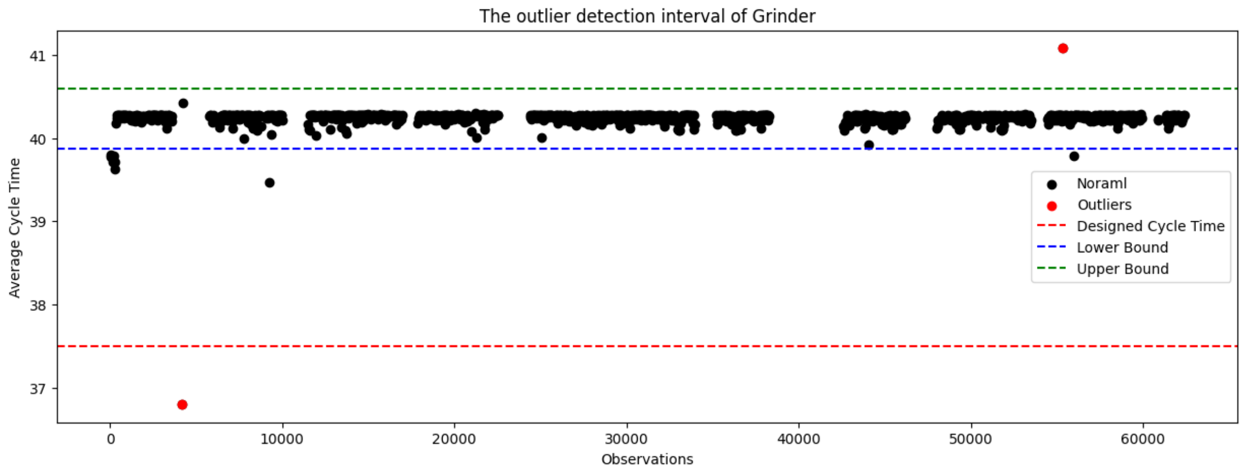


Figure 3*: Scatter plot with adjusted z-score interval example of case I*

Figure 3 above is a typical example of case I, the red dashed line represents the designed cycle time, blue dashed line represents the lower bound, the green dashed line is the upper bound. As we can see that the red dashed line is lower than the lower bound line, then we made the original z-score interval wider. Then for this example, the new outlier detection interval is [D, μ+3σ], and we treat every point outside the interval as an outlier, which is the red point shown in Figure 3.

2. *Case II: Designed cycle time between the lower bound and the upper bound*

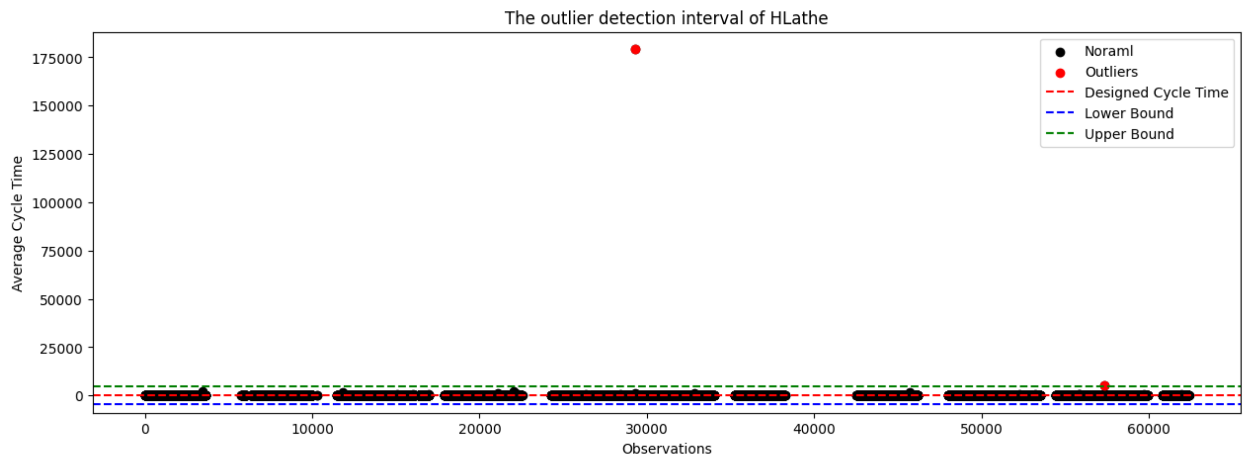


***Figure 4**: Scatter plot with adjusted z-score interval example of case II*

There is an example of case II, we were using the same notation and the same label from the last example. As we can see from *Figure 4*, the red dashed line is lower than the upper bound but higher than the lower bound. Then for this example, the outlier detection interval will not change is still $[\mu - 3\sigma, \mu + 3\sigma]$.

3. *Case III: Designed cycle time greater than the upper bound of interval*

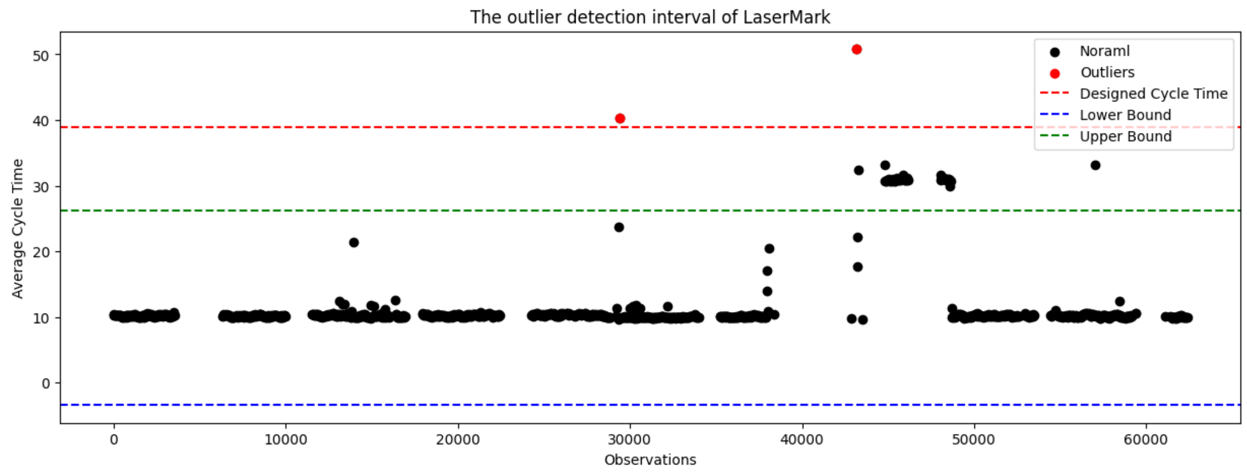


***Figure 5**: Scatter plot with adjusted z-score interval example of case III*

Figure 5 is a typical example, we can see that the red dashed line is higher than the upper bound. For this example, the original outlier interval become wider, which is $[\mu - 3\sigma, D]$. Then we still treated every point beyond the interval as an outlier. By applying this method to the last three months' data both in Hastech and Linergy, we got 86 outliers in Hastech and 196 outliers in Linergy as a result.

Comparison analysis results

Linear Discriminant Analysis

After simply applying the two-tailed T-test between two productions, I found that there is evidence to show that the overall utilization between the two production lines is not the same. To find the most distinguishing features between the two lines, I used linear discriminant analysis (LDA). To reduce the analysis time, I calculated the overall average cycle time of each machine based on machine and shift combinations. Then over the last three months, there are 75 observations in the new dataset, I used those data to fit a simple linear discriminant model. The result coefficient table of some important features shown below,

Feature	**Coefficient**
Shift 1: Day Shift	0.014
Shift 2: Afternoon Shift	0.041
Shift 3: Night Shift	0.054
Asset Capability: VMill	30.29
Asset Capability: VMC	28.05

***Table 5**: Important features' coefficient of the final linear discriminant model*

From *Table 5*, we can see that through different shifts, the most distinguishing feature is the night shift. The contribution of the night shift to the separation of two production lines is 0.054, which is totally reasonable. Probably due to the night shift, the worker may have unexpected behaviors. Finally, two machines with significant differences are VMill and VMC with coefficients of 30.29 and 28.05 respectively. Through these coefficients, we verified the difference between the two production lines. The analysis of different shifts will be further explained in the next section. *Figure 6* below is a confusion matrix heat map fitted on the sample, as we can see there are 28 true positive results of the test sample and there are 34 negative negative results of the test sample, which proves the success of previous linear discriminant analysis.

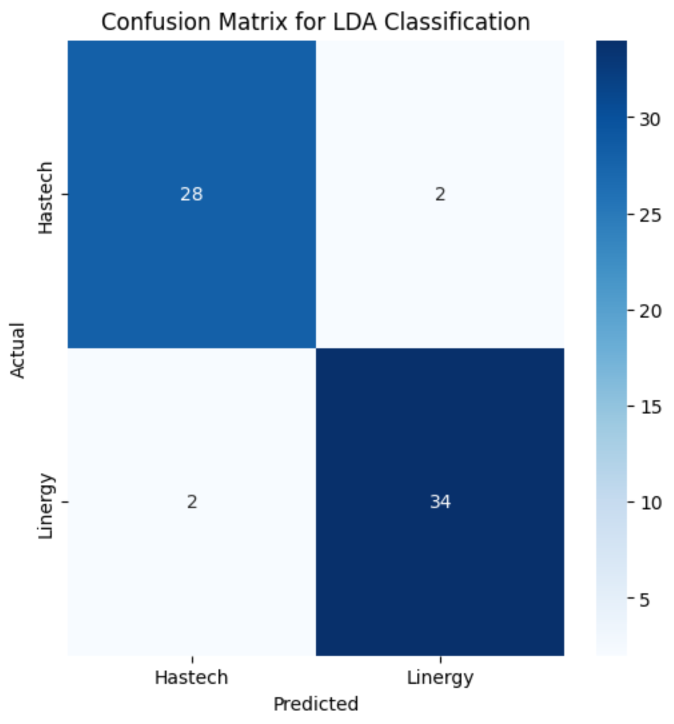


Figure 6*: Confusion matrix heat map of Linear Discriminant Analysis*

T-Test Results

By using the linear discriminant analysis, I had already shown there exists a difference between the two production lines. To analyze why those differences happened, I also applied one-tailed T-tests on these 6 exact same machines through two lines, which are LaserMark, HLathe, PressBushing, Grinder, Polish, and RollSpline. The null hypothesis of those tests is the utilization of each machine in Hastech is equal to the utilization of each machine in Linergy. Then the alternative hypothesis is the utilization of each machine in Hastech is greater than the utilization of each machine in Linergy. *Figure 7* on the next page shows the significant level of each test result. We can see that for those six machines, all of the machine operations are better in Hastech than in Linergy. I would include the limitation of my hypothesis analysis.

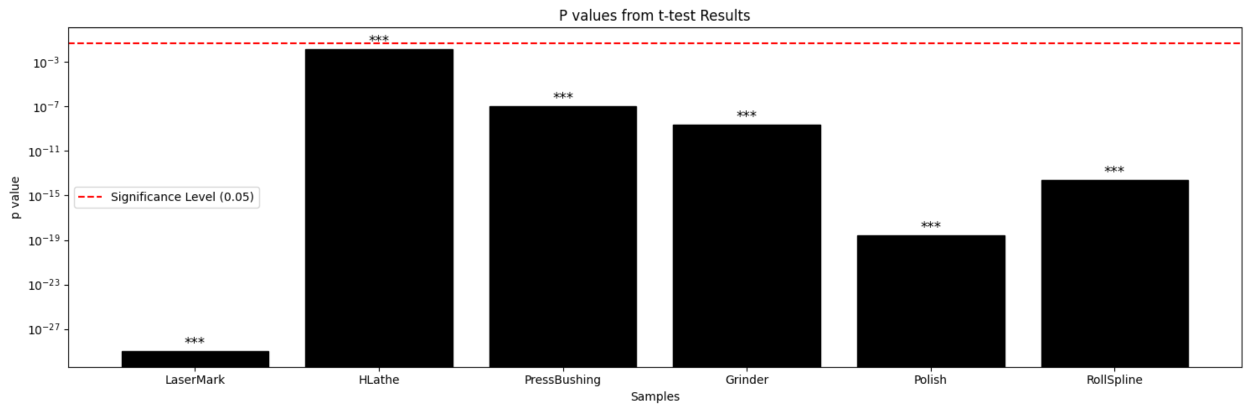


***Figure** 7: the significant level of each test result over 6 same machines*

Shift Analysis Result

As I mentioned in the previous section, the night shift is the most contributing' feature to distinguish Hastech and Linergy. But what if we only look at each machine in each line, which shift is more efficient? That is the main problem I tried to solve by the shift analysis. Then I visualized the data by creating three scatter plots of averaged cycle time over different shifts. From *Figure 8* on the next page, the scatter plot on the top shows the distance between each observation and the designed cycle time of the HLathe machine in the day shift. The middle scatter plot is in the afternoon shift and the bottom scatter plot is in the night shift. By calculating the root mean square of distance which I mentioned in the methodology part, the distance of the day shift is 1335, the distance of the afternoon shift is 2513, but the distance of the night shift is 4380. This means the utilization of HLathe is more stable in the day shift and afternoon shift but is not that stable during the night shift. The results I got are not surprising at all, probably because workers are more likely to be negligent on night shifts and workers are less productive. Also, the scatter plot in

Figure 8 can verify this finding from the position of those scatter, we can see the scatter plot of night shift spread more widely than the other two shifts.

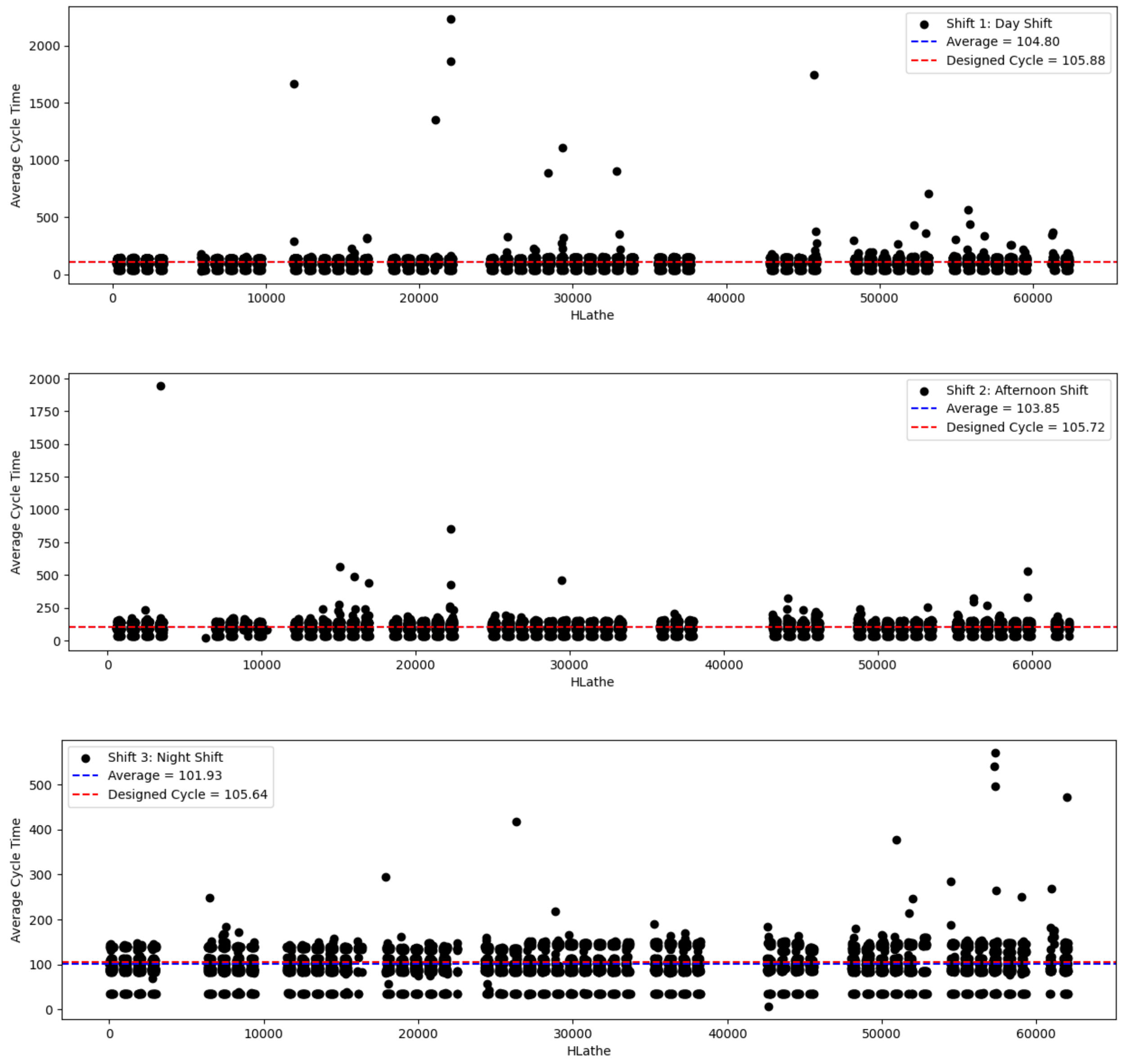


***Figure 8**: top scatter plot shows the averaged cycle time of HLathe in day shift; middle scatter plot shows the average cycle time of HLathe in afternoon shift; bottom scatter plot shows the averaged cycle time of HLathe in night shift.*

Alert System of updating the designed cycle time

We devised our alert system by integrating both distance and the fundamental z-score calculation. Initially, we computed the disparity between the average cycle time and the intended cycle time. Subsequently, we applied a 96% z-score threshold to identify potential outliers. In our methodology, if there are three consecutive dates flagged as outliers, we consider this an indication that these particular machines warrant closer inspection.

Deliverable

Reference List for Mismatch Problem

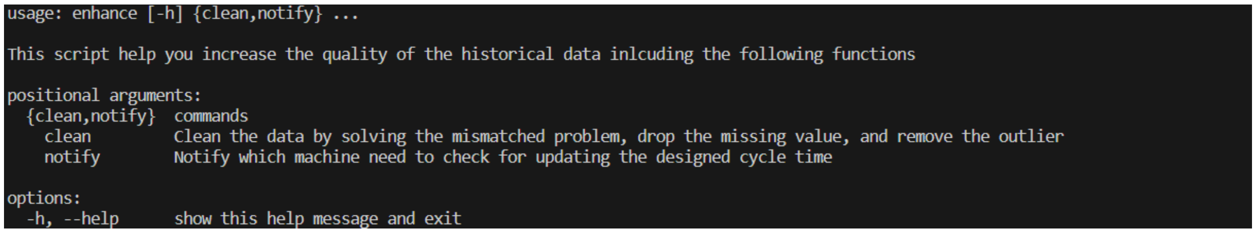
This component takes the form of a comprehensive CSV file. It serves as a valuable resource, cataloging the prevalent mismatch issues encountered across various production lines. Additionally, it provides corresponding replacement terminologies for each identified problem. This list is a valuable asset for simplifying the information input process, potentially transforming it into a user-friendly scroll-down list.

Complete Python Notebook for the Analysis Process

The project includes a meticulously crafted Python Notebook that documents the entire analysis process, including data cleaning, data visualization, outlier detection, T-test for average cycle time, Linear Discriminant Analysis, and alert system for updating designed cycle time. This notebook acts as a detailed guide, offering insights into the methodologies and techniques employed during the analysis. It serves as a valuable reference for project stakeholders and future users.

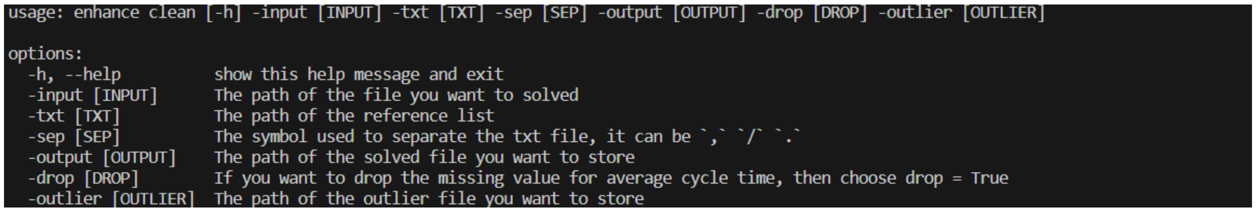
Python Script for Continuous Support

A dynamic Python script is a key element of the project. This script is designed to provide ongoing support, which ensures the project's sustainability. It serves as a reliable tool to address and mitigate mismatch problems as they arise, contributing to the long-term success of the project. This script has two main functions shown in *Figure 9* which are data cleaning and notify, and to make convenience it can be used by simple command line. So, the following functions and examples can help us to understand more about this script.



```
usage: enhance [-h] {clean,notify} ...

This script help you increase the quality of the historical data inlcuding the following functions

positional arguments:
  {clean,notify}  commands
    clean         Clean the data by solving the mismatched problem, drop the missing value, and remove the outlier
    notify        Notify which machine need to check for updating the designed cycle time

options:
  -h, --help      show this help message and exit
```

Figure 9: *Help function about python script called enhanced.py.*



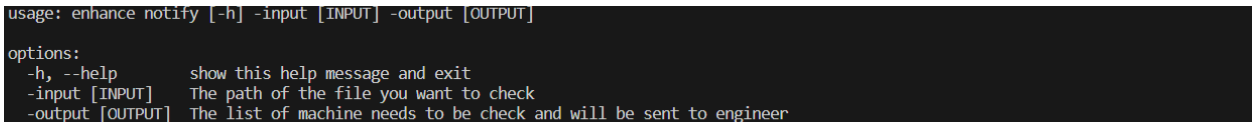
```
usage: enhance clean [-h] -input [INPUT] -txt [TXT] -sep [SEP] -output [OUTPUT] -drop [DROP] -outlier [OUTLIER]

options:
  -h, --help         show this help message and exit
  -input [INPUT]     The path of the file you want to solved
  -txt [TXT]         The path of the reference list
  -sep [SEP]         The symbol used to separate the txt file, it can be `,` `/` `.`
  -output [OUTPUT]   The path of the solved file you want to store
  -drop [DROP]       If you want to drop the missing value for average cycle time, then choose drop = True
  -outlier [OUTLIER] The path of the outlier file you want to store
```

Figure 10: *Help function about the clean function of that python script.*

Based on Figure 10, it becomes evident that the 'clean' function necessitates the specification of six primary arguments when invoked. These include:

1. Input: This argument denotes the file path to the target file you intend to address.
2. Txt: Here, 'Txt' signifies the path leading to the reference list.
3. Sep: The 'Sep' parameter signifies the delimiter symbol employed within the text file, which can take various forms such as ',', '/', or '.'.
4. Output: This argument is utilized to denote the file path where you wish to store the resolved file after processing.
5. Drop: To handle missing values within the average cycle time, the 'Drop' parameter comes into play. When set to 'True', it facilitates the removal of these missing values.
6. Outlier: Finally, 'Outlier' pertains to the file path where you intend to store any identified outliers.

These six arguments collectively enable the 'clean' function to effectively process and manage data, making it a versatile and essential tool for your project.



```
usage: enhance notify [-h] -input [INPUT] -output [OUTPUT]

options:
  -h, --help        show this help message and exit
  -input [INPUT]    The path of the file you want to check
  -output [OUTPUT]  The list of machine needs to be check and will be sent to engineer
```

Figure 11: *Help function about the notify function of that python script.*

For the notify function, the arguments are really simple which including ‘input’ represent the path of the file you want to check and the ‘output’ means the list of the machines need to be checked.

Discussion and Suggestion

Scroll-down List

For this project, I manually created a reference table to solve the inconsistency problem through the different lines. But if the data amount is huge, for example, we want to clean the last 5 years’ data, then there will be more than 100 million observations. So, it still will be a time waste. Then my suggestion is for each factory if Linamar can provide a scroll-down list for asset capability and asset make. Then they can just collect the consistency data into the database. For example, when the workers try to fill in the information of that machine, they must fill in options to choose from, then the data will be cleaner.

Machine Status Reason

For those missing values, it may happen all the time. But if we know the reason for the machine's status, it will provide more chances to help to analysis the production line. If it is possible, Linamar can collect the reason for that machine status from each factory and store it as a new feature in the database. For example, if the machine is down due to maintenance or due to some other reason, they can label it as its own reason which can provide their data analyst with more information.

Alter System for Updating Designed Cycle Time

After the analysis of those two production lines, I found that there are some obvious differences. Also, we can see the distance between the average cycle time and the designed cycle time is not as small enough. This means there exists the same lap, which also means that the designed cycle time is not updating lately. If there is an alert system for updating the designed cycle time, it will bring more efficiency to the production line and Linamar.

Warning System for older machine

Generally speaking, the operating efficiency of the machine will decrease as the age of the machine increases, but for Linamar's machine, I am still verifying this conjecture. Currently, I have only looked at the data for three months. It will not bring a significant efficiency impact to the machine. But I think if we look at the time of the past few years, the age will definitely affect the operation of the machine, so I think that if an early warning system is established, when the machine is too old and the operating efficiency is greatly reduced, the early warning system will give A certain amount of early warning by workers will greatly improve the efficiency of the production line.

Future Plan

Right now, I have almost finished the first phase of this long-term project. And for the one last month of my co-op, I will continue to work on the alter system for updating the designed cycle time and warning system for older machines. So below is my idea about how to solve those two problems.

For the machine age analysis, I will look back to the last three years' data. Then maybe I can find some trend of age against the utilization, and also I decided to use the monthly age, like how many months passed after the machine is produced. If I can find the trend, maybe I will fit some simple linear regression model or polynomial model to predict the average machine utilization ahead. Then if the average machine utilization decreases, the worker will know ahead, then they may check the machine, repair it, or maintenance to keep the efficiency of the lines. But once the machine is too old, they may consider buying a new one.

Finally, for the machine learning algorithm, I think the valuable algorithm is to predict down time of each machine or predict the utilization of each machine. If I can use the existing historical data of existing production lines to predict the new utilization of the new lines, it will help Linamar to decide whether to open that new line or not.

Reference

Kim, T. K. (2015). T test as a parametric statistic. Korean Journal of Anesthesiology, 68(6), 540. https://doi.org/10.4097/kjae.2015.68.6.540

Kłopotek, M., Wierzchoń, S. T., &amp; Trojanowski, K. (2005). Confusion Matrix Visualization. In Intelligent Information Processing and Web Mining: Proceedings of the international IIS: IIPWM'05 Conference held in Gdansk, Poland, June 13-16, 2005. essay, Springer.

Our company. Linamar. (2023, August 11). https://www.linamar.com/our-company/

Pardalos, P. M., & Trafalis, T. B. (2013). Linear Discriminant Analysis. In Robust Data Mining. essay, Springer.

Appendix A

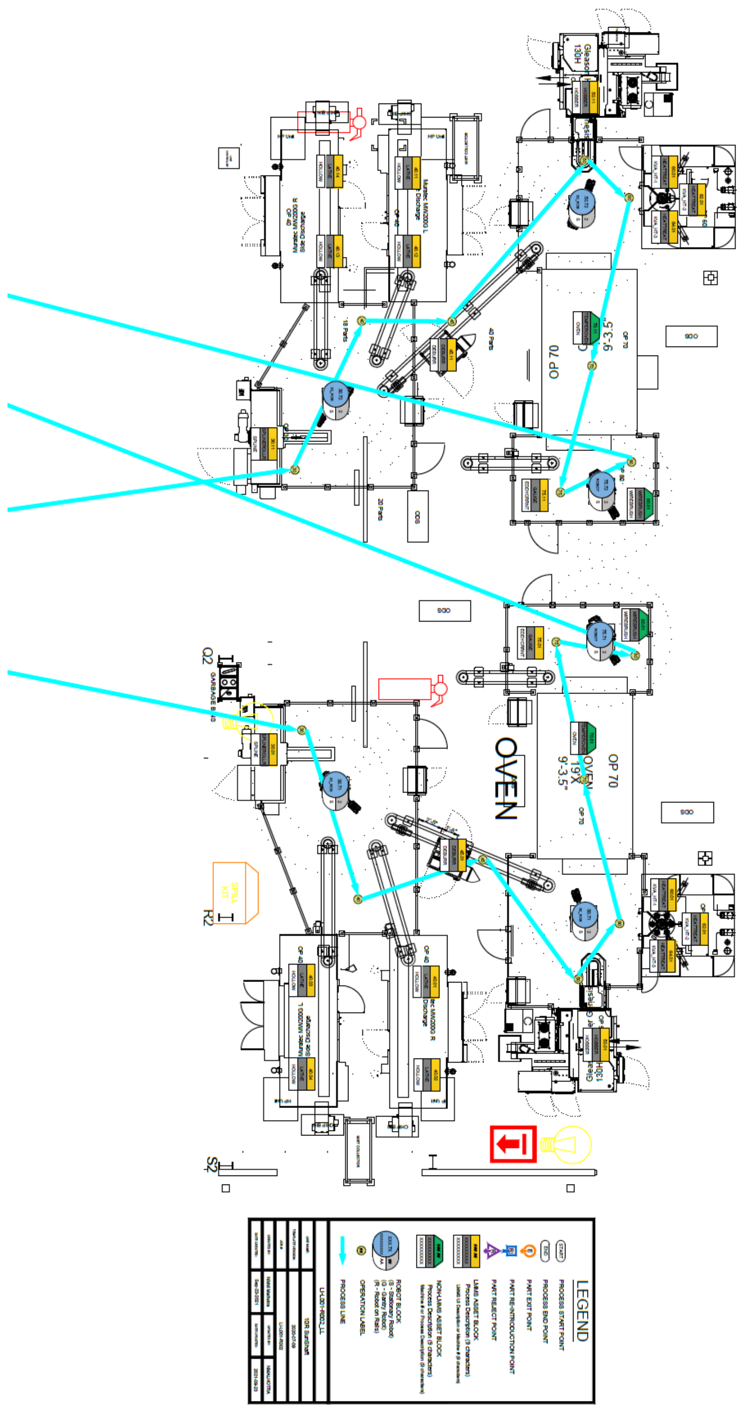


Figure 9: Part of production line design diagram in Linergy

Appendix B

Asset Capability	**Machine Status**	**Hastech**	**Linergy**
HLathe	Down	4637 (79.3%)	8433 (55.6%)
	Offline	784 (13.4%)	2577 (17.0%)
	Blocked	284 (4.9%)	1775 (11.7%)
	Starved	205 (3.5%)	2377(15.7%)
LaserMark	Down	251 (43.7%)	18 (1.9%)
	Offline	14 (2.4%)	9 (0.9%)
	Blocked	176 (30.6%)	0 (0%)
	Starved	102 (17.7%)	910 (98.8%)
PressBushing	Down	852 (90.0%)	1023 (52.0%)
	Offline	23 (2.4%)	304 (15.5%)
	Blocked	90 (2.4%)	115 (5.8%)
	Starved	23 (2.4%)	571 (29%)
Grinder	Down	411 (91.3%)	1163 (59.1%)
	Offline	2 (0.4%)	7 (3.5%)
	Blocked	1 (0.2%)	0 (0%)
	Starved	0 (0%)	675 (34.3%)
Polish	Down	423 (90.8%)	180 (18.8%)
	Offline	39 (8.4%)	7 (0.7%)
	Blocked	1 (0.2%)	372 (38.9%)
	Starved	16 (3.4%)	395 (41.3%)
RollSpline	Down	428 (84.9%)	115 (10.6%)

Asset Capability	Machine Status	Hastech	Linergy
	Offline	**16 (3.2%)**	**6 (0.6%)**
	Blocked	0 (0%)	673 (62.2%)
	Starved	69 (13.7%)	137 (12.7%)

Table 6: *The summary of machine's status of those missing values occurs for six same machines.*