

Application of Digital-Twin Technology for the Job-Shop Scheduling Problem

Vahid Moradi

A Thesis
in the Department of
Mechanical, Industrial, and Aerospace Engineering (MIAE)

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Applied Science in
Industrial Engineering
at Concordia University
Montreal, Quebec, Canada

March 2022
© Vahid Moradi, 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Vahid Moradi

Entitled: Application of Digital Twin for the Job-Shop Scheduling Problem

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Industrial Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Mingyuan Chen

_____ Examiner
Dr. Anjali Awasthi

_____ Examiner
Dr. Mingyuan Chen

_____ Thesis Supervisor
Dr. Akif Asil Bulgak

Approved by Dr. Sivakumar Narayanswamy
Chair of Department or Graduate Program Director

April 12, 2022

Dr. Mourad Debbabi
Dean, Gina Cody School of Engineering and Computer Science

Abstract

Application of Digital-Twin Technology for The Job-Shop Scheduling Problem

Vahid Moradi

Job Shop Scheduling (JSS) is considered an optimization problem for implementing optimal job scheduling. Different techniques such as mathematical optimization or simulation-optimization have been used for the optimization of the JSS problem. However, Industry 4.0 and new developments in technologies such as the Industrial Internet of Things (IIoT) and Cyber-Physical System (CPS) caused the introduction of the digital twin technology which is considered a new method for optimization of the JSS problem.

The digital twin technology provides a cloud-based simulation model to make use of the real-time data provided by the IIoT and CPS from physical space. Additionally, the digital twin technology applies Machine Learning (ML) techniques to build prediction models and uses powerful cloud computing to do real-time what-if analysis, rescheduling, and response for optimization of the JSS.

In this thesis, we apply the digital twin concepts to a case study which is the stamping shop of AAL company to optimize its JSS problem. We build a JSS simulation model of the stamping shop manufacturing processes, use ML techniques to predict the probability of machine breakdowns through the real-time data captured from the sensors, and consider a Condition-Based Maintenance (CBM) process to optimize the JSS by reducing unexpected machine failures during operation.

Comparing the results of the digital twin technology with the traditional simulation-optimization technique used in the ALL company, the digital twin is a more efficient approach for optimizing the JSS problem by providing an accurate prediction model and implementing CBM scenarios to reduce unplanned machine failures.

Acknowledgment

There are several people that I would like to thank for helping me to write this dissertation.

I would firstly like to thank my supervisor, Dr. Akif Asil Bulgak, for giving great advice, being supportive, and replying to my emails and questions instantly even during the Covid-19 pandemic and lockdown issues.

A big thanks to my wife and my best friend accompanying me during this challenging journey which our marriage life was coincident with it, thank you for your love, support, and patience.

A special thanks to my parents for always giving support and devotion in the whole of my life, thank you for everything you have done for me, words cannot express my gratitude.

I would like to thank my grandmothers who taught me who to love, my paternal grandmother is no longer with us but what she had taught me is in my heart, thanks to my grandfathers who teach me how to work hard, and a great thanks to my grand aunt who has always been supportive without any expectations.

God bless you all!

Table of Contents

List of Figures.....	viii
List of Tables	x
1. Introduction.....	1
• 1.1 Motivation.....	2
• 1.2 Outline.....	3
• 1.3 Contribution of the Thesis.....	4
• 1.4 The Case Study, Data, and Assumptions	5
2. Literature Review	7
• 2.1 Job shop scheduling problem.....	8
• 2.2 JSS Based on Digital Twin	11
• 2.3 Digital Twin-Based Cyber-Physical Production System (DT-CPPS).....	13
• 2.4 DT-CPPS and Condition-Based Maintenance	15
2.4.1 Machine Maintenance Strategies	15
2.4.2 CBM Versus Other Maintenance Strategies for JSS.....	16
2.4.3 DT_CPPS and CBM for JSS problem	17
• 2.5 Machine Learning Technique for Prediction Model	19
2.5.1 Machine learning Methods.....	20
2.5.2 Supervised Learning Methods	21
2.5.3 Decision Tree Method.....	22
2.5.4 Random Forest Decision Tree.....	25
• 2.6 Methodology of Choice.....	27
3. Simulation Model for the JSS.....	29

- 3.1 System Analysis29
- 3.2 Scheduling Strategies31
- 3.3 Creating a Simulation Model for the JSS with Arena32
 - 3.3.1 Scheduling Process33
 - 3.3.2 Operation Process37
- 4. Machine Failure Approach in Traditional Simulation-Optimization Technique.....41
 - 4.1 Distribution Fitting Process.....42
 - 4.2 Descriptive Statistics of Interval HDF problem44
 - 4.3 Outlier Test45
 - 4.4 Histogram.....46
- 5. Machine Failure Approach in Digital Twin Technology.....48
 - 5.1 Importing and Cleaning Dataset.....49
 - 5.2 Data Analyzing and Visualization52
 - 5.3 Machine Learning Technique for Prediction Model59
 - 5.3.1 Defining Dataset and Variables59
 - 5.3.2 Splitting Dataset into Training Set and Test Set60
 - 5.3.3 Training Decision Tree Classification Model on the Training data61
- 6. Analysis of the Results Obtained.....63
 - 6.1 Results of Distribution Fitting Process on Sample Test.....63
 - 6.2 Confusion Matrix and Accuracy Score of ML techniques for Prediction Model ..65
 - 6.3 Applying the Prediction Model to Real-Time Data69
 - 6.4 Implementing CBM in the Simulation model73
- 7. Conclusions, Limitations, and Future Work76

- 7.1 Conclusions76
- 7.2 Limitation of the Research79
- 7.3 Future work80

References82

List of Figures

Figure 1: Digital Twin Based Cyber-Physical Production System (DT-CPPS)	14
Figure 2: ML Technique Process for Prediction Model	20
Figure 3: Supervised ML Technique Process	21
Figure 4: Decision Tree splitting plot	22
Figure 5: Regression Tree Plot	24
Figure 6: Classification Tee Plot	25
Figure 7: Decision Tree Versus Random Forest Methods.....	26
Figure 8: Real-Time Analysis and Timely-Response in Digital Twin	28
Figure 9: AAL Stamping Shop Manufacturing Process	31
Figure 10: Assigning Paint Department Priorities	34
Figure 11: Categorizing Demands in Arena Simulation Software	35
Figure 12: The Decision Model for Job Scheduling in Arena Simulation Software	36
Figure 13: Job Shop Scheduling Model in Arena Simulation Software	37
Figure 14: Transport Section in Arena	38
Figure 15: Decision Module for Comparing two Consecutive Entities in a Queue	39
Figure 16: The Sequence Module in Arena	39
Figure 17: Stamping process model in Arena.....	40
Figure 18: Quality Control and Storage Model in Arena	40
Figure 19: Failure Module in the Arena Simulation Software	41
Figure 20: The Distribution Plot of HDF Problem.....	43
Figure 21: Outlier Test Diagram.....	45
Figure 22: Frequency and Cumulative Diagram of Interval Between Failures	47
Figure 23: Heat Map for Data Visualization	52
Figure 24: Box -Plot diagram for Machine Performance Features.....	54
Figure 25: Box Plot Diagram.....	55
Figure 26: Box-Plot Diagram for HDF problem	56
Figure 27: The Pair-Plot Diagram	58

Figure 28: Splitting the HDF dataset Into Training data and Test data	60
Figure 29: The Decision Tree Classification Model.....	70
Figure 30: CBM in Simulation Model.....	73
Figure 31: Defining the Input File in the Arena Software	74
Figure 32: Read and Write Module in Arena Software	74

List of Tables

Table 1: Requirements and Benefits of Different Maintenance Approaches	17
Table 2: Demands and Stamping Sequences of Body panels	33
Table 3: Descriptive Statistics	44
Table 4: Outlier Test Result.....	45
Table 5: Specification of Histogram.....	46
Table 6: Frequency and Cumulative Table of Interval between failures.....	47
Table 7: Historical Data of Stamping Machine	50
Table 8: The Dataset Information.....	51
Table 9: Checking for Null fields	51
Table 10: Root Causes of Failures.....	53
Table 11: Correlation Between Variables in Machine Normal Condition	57
Table 12: Correlation Between Variables in Machine Failure Condition- HDF problem...57	
Table 13: Results of Distribution Fitting Process.....	64
Table 14: Confusion Matrix.....	66
Table 15: Confusion Matrix of Decision Tree Classifier.....	67
Table 16: Confusion Matrix of Random Forest Classifier.....	68

1. Introduction

This dissertation investigates the job-shop scheduling (JSS) that is considered an optimization problem in operation research (OR) through Digital-Twin technology. In this regard, we work on a case study which is on the stamping shop of American Automobile Limited (AAL) company as a sample of a complex manufacturing system. AAL's stamping shop features a JSS problem categorized as an optimal job scheduling problem because it has multi-stage jobs, while each job has its operation order.

Industry 4.0 advances the simulation-optimization technique also known as “simulation via optimization”, or “simulation-based optimization” toward the digital twin technology (Jian & Henderson, 2016; Söderberg et al., 2017). New developments in technologies caused by Industry 4.0 such as Industrial Information of Things (IIoT) and Cyber-Physical System (CPS) can capture real-time data and connect manufacturing units and resources such as devices, machines, operators, etc. This availability of real-time manufacturing data together with advanced data analysis techniques such as Machine Learning (ML), Artificial Intelligence (AI) enables a simulation model for real-time monitoring, prediction, and optimization which are the concepts of digital twin technology in manufacturing systems (Schroeder et al., 2021; Söderberg et al., 2017). In this thesis, we will apply the digital twin technology concepts for the JSS problem by:

- Working on AAL's stamping shop manufacturing process that is applying a traditional mathematical modeling technique where the machines and production line are digitalized and equipped with sensors recently.
- Developing a JSS simulation model for the stamping shop.
- Developing the distribution fitting method used in the traditional-simulation technique.
- Developing a machine failure prediction model through ML techniques applied for digital twin technology.

- Applying the results of the prediction model to the simulation model and implementing Condition-Based Maintenance (CBM) to optimize the JSS problem by reducing unplanned machine breakdowns.

1.1 Motivation

The motivation for working on optimization of the JSS through the digital twin technology in this dissertation are as follows:

- To move towards a sustainable manufacturing process

It is estimated that the manufacturing sector contains around 52% of the worldwide energy consumption. According to research, 85% of the energy consumption in the manufacturing sector is wasted due to machines and production lines' idle time, failure, and inefficient performance (Xia et al., 2021). Optimizing a JSS through machine failure prediction can be one step toward sustainable manufacturing by reducing machine breakdowns, reducing production line idle time, improving machine performance, increasing the remaining useful life (RUL) of parts and machines, etc. (Jasiulewicz-Kaczmarek et al., 2020; Xia et al., 2021)

- To move towards lean manufacturing

The digital twin technology can optimize a JSS problem and can improve efficiency in lean manufacturing by saving time and cost of resources, reducing downtime, and reducing inventory and overproduction. (Bazaz et al., 2019; Shao & Helu, 2020).

- Industry Fourth Revolution (Industry 4.0) and more access to operation and equipment data

Industry 4.0 that is first introduced at Hannover Fair in 2011, is now expanding due to the reduction in costs of sensors (de Paula Ferreira et al., 2020). A survey held by Price Waterhouse Coopers (PWC) in 2016 with more than 2000 participants from nine major

industries sectors shows that manufacturing companies is going to invest around 5% of their annual revenue in digitalization projects in the next five years (2016 Global Industry 4.0 Survey-Industry Key Findings,). The digitalizing of manufacturing systems make the use of digital twin technology more common in near future.

1.2 Outline

The dissertation is organized as follows:

Chapter 2: literature review

This chapter compares the simulation-optimization technique with mathematical techniques for optimization of the JSS problem. New advancements in the simulation-optimization technique in the industry 4.0 era are discussed. Moreover, the application of digital twin for CBM and the application of ML techniques for building a machine failure prediction model are investigated.

Chapter 3: Simulation Model for the JSS

In this chapter, we work on the JSS problem of the AAL company stamping shop as a case study. We do system analysis, review AAL's scheduling strategies, and develop a simulation model with Arena software for scheduling and sequencing of jobs.

Chapter 4: Machine Failure approach in Traditional Simulation-Optimization Technique

We apply the distribution fitting process used in the traditional simulation-optimization technique to our dataset.

Chapter 5: Machine Failure Approach in Digital Twin Technology

We apply ML techniques to develop a prediction model for the Heat Dissipation Failure (HDF) problem of a stamping machine.

Chapter 6: Results

In this chapter, we compare the results and the accuracy of ML techniques to select one of them in our model. Furthermore, we compare the results of the ML technique for the prediction model in the digital twin technology with the distribution fitting approach in the traditional simulation-optimization technique. we create the concepts of CBM in the simulation model by establishing a connection between the prediction model done by the Python platform and the simulation model created by Arena software.

Chapter 7: Conclusion, Limitation, and Future work

This final chapter concludes this thesis by restating its main contributions and suggesting some areas for future work.

1.3 Contribution of the Thesis

The contributions of this dissertation are as follows:

- We describe the efficiency of digital twin technology for JSS problems and the optimal scheduling of complex manufacturing systems.
- We develop a discrete-event simulation model with Arena software for the JSS problem of the stamping shop.
- We develop a distribution fitting process applied in the traditional simulation-optimization technique by Minitab software and the Input Analyzer tool in Arena software.
- We develop a prediction model for the HDF problem of the stamping machine by comparing the results of the decision tree and the random forest classification techniques with Python programming. In this regard, it is assumed that we have access to historical data of the stamping machine collected by the sensors including air temperature(k), process temperature (k), rotational speed (rpm), torque (Nm), and tool wear (min).

- We present the application of CBM by applying the results of the prediction model in the simulation model to detect the probability of machine failure. It is assumed that we have access to real-time data through CPS and IIoT technologies.
- We compare the results of digital twin technology with the results of the traditional simulation-optimization technique for optimization of the JSS problem.

1.4 The Case Study, Data, and Assumptions

This section describes the case study and data obtained as well as some assumptions that we make to meet the aims and objectives of this thesis:

- The case study is the stamping shop of American Automotive Limited (AAL) company. We obtained this case study from “Ivey Publishing” in January 2021. AAL is a manufacturing company producing different car body panels through stamping, welding, and painting processes. The stamping shop which is the focus of this thesis located in one of the AAL plants (Plant A) supplying body panels for other processes. (Case Study: American Automobiles Limited: Production Planning | Ivey Publishing.). Ivey Publishing is a leading company located in Ontario, Canada, and has published 39,000 business case studies (Ivey Publishing: About | LinkedIn).
- As for the data, we search through some online open data sources providing real datasets such as GitHub, NASA, DataWorld, and Kaggle. Finally, we find a proper dataset on the Kaggle related to maintenance logs of a machine working in a manufacturing system and consider it as a maintenance log of one of AAL’s Stamping machines. The data is collected from sensors including air temperature(k), process temperature (k), rotational speed (rpm), torque (Nm), tool wear(min), and indicated the machine status (normal or failure) in 10,000 records. (*Predictive Maintenance* | Kaggle, n.d.; *UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set*, n.d.; Torcianti & Matzka, 2021).

Regarding the assumptions of the methodology of the study:

1. It is assumed that the data related to the maintenance logs of a stamping machine in the AAL company and collected from sensors every 30 minutes.
2. It is assumed that the AAL stamping shop is digitalized, and we have access to real-time data captured from sensors through CPS technology which makes the concept of digital twin and real-time simulation optimization in this dissertation.
3. It is assumed that have access to advanced cloud computing power for timely data analysis and response.

Regarding the assumption of building the simulation model:

4. There are severe inventory space constraints in the AAL stamping shop.
5. Adequate raw materials are always available.
6. The demands of each job are known and announced by the sales department.
7. Each machine can only process one job at a time.
8. All machines and jobs are available at the beginning.
9. In our simulation model, the only machine failure is considered an interruption in the manufacturing system.
10. Processing time is already statistically analyzed.

2. Literature Review

This chapter reviews some recent literature about the application of the digital twin for the JSS problem in six sections:

Section 1. Job-Shop scheduling problem

We review different optimization techniques for the JSS problem and compare strengths and weaknesses.

Section 2. Digital Twin and The JSS problem

We describe the digital twin concepts and explain their advantages and disadvantages compared to the traditional simulation-optimization technique.

Section 3. Digital Twin-Based Cyber-Physical Production System (DT_CPPS)

We investigate the role of CPPS in digital twin technology for providing real-time data to be applied for timely analysis, prediction, and rescheduling of the JSS problem.

Section 4. DT_CPPS and Condition-Based Maintenance (CBM)

We review maintenance strategies for machine failures, explain the advantages of CBM for the JSS problem, and the process of CBM in the digital twin technology.

Section 5. Machine learning Techniques for Prediction Model

We review ML techniques for creating machine failure prediction models and discuss the decision tree and random forest classification techniques in more detail.

Section 6. Methodology of Choice

We explain our methodology to put the aims and objectives of this dissertation into practice.

2.1 Job shop scheduling problem

The Job shop scheduling (JSS) problem has always been an important Operations Research (OR) subject for efficiency in manufacturing systems (Zhang et al., 2019). JSS is defined as a problem of assigning a set of jobs J_i ($j = 1, 2, \dots, n$) to a set of machines M_j ($j = 1, 2, \dots, m$) in a way to optimize an objective which can be the makespan, energy consumption, total tardiness, etc. (J. Zhang et al., 2019; M. Zhang et al., 2021). JSS is considered a non-deterministic polynomial hard (NP-hard) problem.

Simulation-based optimization and mathematical optimization are the two main OR methods for finding optimal or near-optimal solutions for decision-making problems such as JSS (Law, 2015). With regards to mathematical optimization methods, JSS has been traditionally solved by heuristic optimization techniques such as artificial neural networks, or by meta-heuristic optimization techniques such as tabu search, genetic algorithm, simulated annealing, etc. (Geyik & Cedimoglu, 2004; J. Zhang et al., 2019).

Simulation-based optimization or simulation-optimization is another method for solving JSS problems which is the focus of this thesis as well. This dissertation chooses the simulation-optimization technique for the AAL JSS problem. First, because of some advantages that the simulation modeling method has over mathematical modeling. Second, because of new developments caused by industry 4.0 shifting simulation model technique to digital twin concepts.

Regarding the advantages of modeling a JSS problem based on the simulation rather than a mathematical method:

1. (Jian & Henderson, 2016; Kelton et al., 2015, p. 7) state that the mathematical optimization techniques require some simplifications in comparison with simulation models to be applicable for complex systems. These simplifications can be included in reducing the number of decision variables, considering a stochastic system as a deterministic system, assuming a dynamic system as a statistic system, etc.

2. (Kelton et al., n.d.; Law, 2015; Lin & Chen, 2015; P et al., 2016) assert that the simplifications in the mathematical optimization method lose the accuracy of the model by problems such as Flaws of Averages; however, the simulation technique is not only flexible in defining more variables but also capable of creating stochastic and dynamic models.
3. As regards machine failure considerations which is the focus of this dissertation for the JSS optimization, (M. Zhang et al., 2021) state that for the simplification in mathematical modeling machines are assumed to be always available whereas we are encountered unplanned machine breakdowns during the operation from time to time. This is while we can easily assign a distribution function to consider machine failures in the simulation software packages like Arena through a failure module.
4. (Caggiano et al., 2015; de Paula Ferreira et al., 2020; Fang et al., 2019) add the possibility of doing system analysis and what-if analysis to the advantages of the simulation-optimization over the mathematical optimization method. The what-if analysis provides an opportunity to evaluate different uncertainties scenarios such as failures or delays in the model.

On the other hand, some literature cases elaborate challenges for modeling a JSS problem based on the simulation-optimization technique:

- (R. Liu et al., 2018) declare that dynamic variables and constant changes in real-time manufacturing operations make the simulation-optimization technique a time-consuming approach to detect and adjust changes and uncertainties in a JSS modeling.
- (Monostori et al., 2016; Z. Zhang et al., 2020) mention that the simulation-optimization technique is not equipped with powerful statistical analysis to deal with big data collected from sensors and manufacturing information systems.

(Fang et al., 2019; He & Bai, 2021; J. Zhang et al., 2019) address these issues by the digital twin technology as below:

- 1) Real-time data accessibility: Digital Twin technology can get access to real-time data through sensors, IIoT, and CPS technologies exchanging data between physical and virtual space. Therefore, it is possible to monitor real-time in the physical space to optimize estimations in the simulation model.
- 2) Dynamic analysis method: it happens by monitoring and analyzing the real-time data through ML or AI platforms like Python that can be integrated by the digital twin technology to detect and predict any changes or unseen events in scheduling.
- 3) Cloud-based simulation model: the digital twin technology can implement cloud-based simulation models with powerful cloud computing to do timely analysis and respond to any changes and uncertainties in the JSS model.

Therefore, in case of facing an unplanned machine failure (which is the focus of this dissertation), or absence of a worker, or any other unprecedented events in the manufacturing operation, the simulation model in the digital twin can immediately detect, analyze, and respond. This occurs through real-time data monitoring, predicting, what-if analyzing, rescheduling, and timely response.

In a nutshell, the simulation-optimization technique in Digital twin technology for the JSS problem is evolved by the features of real-time data and dynamic scheduling which can be named a job shop scheduling based on Digital Twin (Fang et al., 2019).

2.2 JSS Based on Digital Twin

(Z. Zhang et al., 2020) defines Digital Twin as a combination of physical space and virtual space of a system connected by exchanging data through advanced technologies to perform real-time optimization. The main advanced technologies that help the digital twin for data gathering and fusing include CPS, Industrial Internet of Things (IIoT), cloud technology, and computing power technologies.

The digital twin can perform planning (scheduling strategies), scheduling, controlling, and rescheduling of a manufacturing system in virtual space by a simulation-optimization technique through the real-time data obtained from manufacturing units such as machines, operators, and information systems in the physical space (Shao & Helu, 2020; Z. Zhang et al., 2020). In other words, (Zhang et al., 2019; Kalita et al., 2019) point out that the digital twin application can implement the concept of smart devices, smart machines, and smart manufacturing systems.

(Fang et al., 2019; Z. Zhang et al., 2020) declare that the interaction between physical and virtual space is the main advantage of the digital twin simulation-optimization over traditional simulation-optimization techniques for the JSS problem as below:

1. Accurate processing time estimation

Fitting a distribution to a processing time of decision variables in traditional simulation optimization is based on the statistical analysis of a random size of n samples. This estimation is always considered a fix and constant for job type i on machine j in the simulation model. However, with the help of more detailed data acquired from manufacturing units, it is possible to fit a more accurate distribution function by considering the relationship between jobs, workers, and machines. For example, t_{ijkl} is a distribution of job i , operation j , machine k , and worker s (Fang et al., 2019)

2. Dynamic scheduling

As matter of fact, a scheduling plan is not always implemented in actual operation due to a dynamic environment and uncertain decision variables. Many disturbances such as machine failure, new demands, and delays lead to deviation of plan and rescheduling (Fang et al., 2019; M. Zhang et al., 2021). In comparison with the traditional simulation optimization technique, the digital twin technique has real-time data sensing. Therefore, any abnormal data such as the probability of machine failure can be detected and then implemented in the simulation model to evaluate effects and response for rescheduling (Fang et al., 2019)

In 2017 and for the two years after, Gartner, which is a Stanford technology research and consulting company announced Digital Twin as a top 10 strategic technology trend (Gartners Top 10 Technology Trends 2017; Liu et al., 2021). Many modern companies such as Siemens, General Electric, and Brilliant have already implemented digital twin technologies in their manufacturing systems for applications such as system planning, maintenance, optimization, and asset management. These companies mostly use popular simulation software such as AnyLogic, Simulink, Arena, Simul8, Autodesk, etc. (de Paula Ferreira et al., 2020). These simulation platforms are mainly equipped with optimization tools for example Simulink which is a product of MathWork company is equipped with MATLAB (*Simulink - Simulation and Model-Based Design - MATLAB & Simulink*, n.d.), Anylogic, Arena, and Simulink have the OptQuest tool for optimization purpose (*Arena (Software) - Wikipedia*, n.d.; *How Many Simulation Trials Should I Run? Factors That Impact OptQuest Search Performance – AnyLogic Simulation Software*, n.d.).

2.3 Digital Twin-Based Cyber-Physical Production System (DT-CPPS)

(Rho et al., 2016) described CPS as “systems that integrate computing and communication capabilities with dynamics of physical and engineered systems”. As it is shown in figure 1 the interaction between the real physical operation and virtual space is done by CPS, and the digital twin technology implements dynamic and real-time analyzing and response algorithms through a simulation-optimization technique ((Ding et al., 2019; Fang et al., 2019; Schroeder et al., 2021).

(Ding et al., 2019; Monostori et al., 2016; Z. Zhang et al., 2020) introduce the integration of digital Twin and CPS technology for a production system as Digital Twin-Based Cyber-Physical Production System (DT-CPPS) establishing the concept of smart manufacturing. In short, they categorize the DT-CPPS responsibilities:

1. Data collection from manufacturing units
data are gathered from manufacturing units and information systems. Manufacturing units such as machines and devices are equipped with sensors and Radio Frequency (RF) for data transmission
2. Data synchronizing and fusing
A high-speed protocol of CPS establishes a system for mutual interaction between physical and virtual space by connecting sensors, controllers, equipment, and information systems to provide real-time data gathering, fusing, transmission, feedback, and response (Ding et al., 2019; Monostori et al., 2016).
3. Simulation optimization in virtual space
Based on real-time data acquired from physical space, the digital twin technology optimizes the JSS problem by monitoring, predicting rescheduling, and timely response to any unprecedented conditions and abnormal data. (Fang et al., 2019).

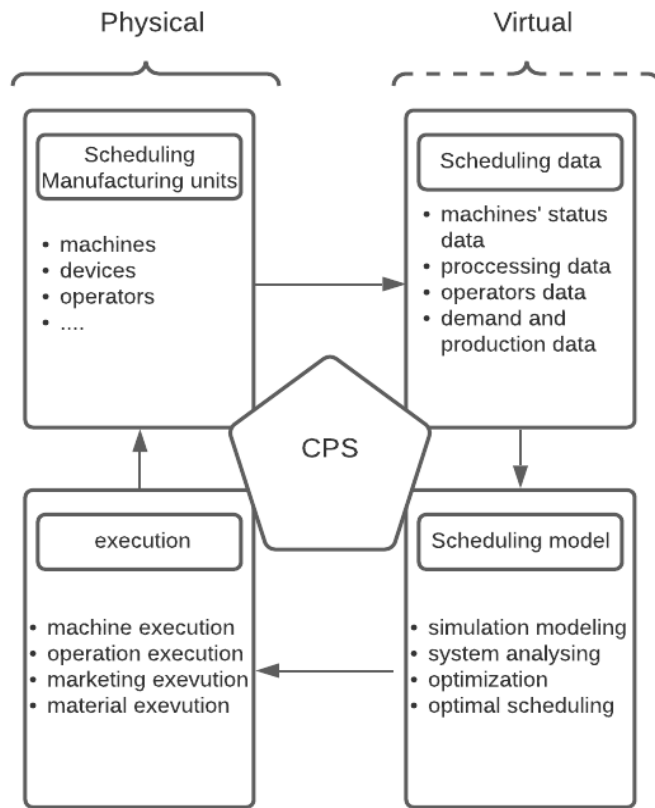


Figure 1: Digital Twin Based Cyber-Physical Production System (DT-CPPS)

Source: (Fang et al., 2019)

2.4 DT-CPPS and Condition-Based Maintenance

Unexpected machine breakdown is one of the uncertainties that cause catastrophic consequences on the JSS. An unexpected machine failure not only causes direct costs such as waste in energy consumption, raw material, etc. but also it causes indirect costs such as inefficient use of equipment and resources, or reputation risks. It is estimated that machine breakdown is the cause of US\$ 450 billion in revenue lost in a year. (Ayvaz & Alpay, 2021; Roosefert Mohan et al., 2021a).

Therefore, deploying a mechanism to detect machines failure with DT_CPPS technology at early stages can provide an opportunity for the maintenance team to do corrective actions before happening a catastrophic machine breakdown.

2.4.1 Machine Maintenance Strategies

Cakir et al., 2021 categorize machine maintenance plans to:

1. corrective maintenance (CM)

This maintenance happens after machine breakdown. It affects the scheduling, and today's maintenance strategy is based on minimizing corrective maintenance.

2. preventive maintenance (PM)

a set of periodic actions done on machines to increase the machine's function and prevent any further possible breakdown.

3. predictive maintenance or condition-based maintenance (CBM)

This maintenance policy is based on monitoring machines' functions and acting in advance to prevent a probable failure.

4. proactive maintenance (PaM)

Proactive maintenance is to find the roots causes of failure such as the wrong lubricant, unbalance, and operator error (Fitch, n.d.).

2.4.2 CBM Versus Other Maintenance Strategies for JSS

CM approach is used after a machine breakdown happens which is not a proper approach to reduce the number of unplanned machine failures to optimize a JSS. On the other hand, implementing an effective PM plan can reduce the number of unplanned machine failures; however, PM is not an efficient and sustainable approach for reducing machine failure because of the trade-off in its decision model (Traini et al., 2019):

1. maximizing the Remaining Useful Life (RUL) of parts by considering the acceptance of the risk of machine breakdown
2. maximizing the RUL of a machine by accepting the cost of replacing parts earlier

Relying just on CM and PM is considered a traditional maintenance strategy these days, while CBM and PaM are becoming more and more popular approaches. CBM strategy which is the focus of this dissertation can make use of advanced tools like ML and AI to improve decision-making progress and optimize the trade-off mentioned above. The optimization of the trade-off is done by applying prediction models to find the likelihood of a machine failure or estimate the RUL of parts (Cakir et al., 2021; Kaparathi & Bumblauskas, 2020; Roosefert Mohan et al., 2021b).

Statistics show that factories that have switched to CBM have around a 25 to 30 percent decrease in maintenance activities, 35 to 45 percent decrease in machines breakdown, and a 20 to 25 percent increase in production and investment return (Cakir et al., 2021)

The two main key Performance Indicators (KPIs) are:

- Mean time between failure (MTBF)
- Mean time to repair (MTTR)

Table 1 presents the requirements and benefits of each maintenance approach, and as it is shown the CBM and PaM reduce MTBF and MTTR making scheduling more optimized compared to CM and PM strategies (Cakir et al., 2021; Kučera et al., 2020). And as discussed earlier in the introduction section of this dissertation, having an optimal JSS will lead to a reduction of waste in energy consumption and raw material, an increase in efficiency of the manufacturing system, etc.

Table 1: Requirements and Benefits of Different Maintenance Approaches

Strategies	Requirements			Benefits			
	Application	Needed technology	Cost of implementation	Machine Downtime	MTBF	MTTR	Optimal scheduling
Corrective	Low impact on failure	Low	Low	Highest	Lowest	Highest	Lowest
Preventive	Average impact on failure	Low	Average	Average	Average	Average	Average
Predictive	Simple prediction failure	High	High	Low	High	Low	High
Proactive	Multiple predictions of failure	High	High	Lowest	Highest	Lowest	Highest

2.4.3 DT_CPPS and CBM for JSS problem

CBM is categorized to:

1. offline monitoring
2. online monitoring

Digital Twin and CPS provide a real-time data-driven algorithm for an online monitoring machine condition which is called continuous condition monitoring, to continuously capture, transfer, synchronize, and apply data collected from sensors to a prediction model (Cakir et al., 2021).

To build a prediction model, the ML techniques work on the historical data of a machine's performance. The dataset features the machine's performance factors such as temperature, vibration, rotation, etc which can indicate the normal or failure machine status. (Roosefert Mohan et al., 2021a). The supervised ML techniques are used for labeled data (the dataset indicating the machine statuses) and unsupervised learning ML techniques are used for unlabeled data. Thereafter, the prediction model is applied to real-time data to predict the possibility of unexpected machines failure (Ayvaz & Alpay, 2021; Roosefert Mohan et al., 2021a). Therefore, the prediction model gives an in-advance notice to decision-makers of a machine failure and provides an opportunity for maintenance teams to do corrective actions ahead of time and try to avoid a machine breakdown or any further disastrous issues (Ayvaz & Alpay, 2021). This process helps to make the scheduling of complex manufacturing systems or JSS optimal, smart, agile, and sustainable (Ayvaz & Alpay, 2021; Jasiulewicz-Kaczmarek et al..).

2.5 Machine Learning Technique for Prediction Model

As it is shown in figure 2 the ML process for deploying a prediction model consists of 1. Data acquisition, 2. Data cleaning 3. Training Data, 4. Test data.4. Modeling, 5. Deploying which are described with the following steps (Roosefert Mohan et al., 2021a):

- Data Acquisition: to deploy a prediction model it is necessary to have access to the historical data of a machine. The historical data usually shows the dates, times, machine conditions (normal or failure), machine performance features, and root causes of failures.
- Data Cleaning: once the data is acquired, it needs to be cleaned. To do this, statistical analysis is applied to find for example outliers or missing to make a dataset ready for further steps.
- Splitting the training and test data: the cleaned data is split into test data and train data. The proportion of this split is usually 25% and 75% for train data and test data respectively.
- Training the ML techniques on the train data: ML techniques are applied to the training data to create a prediction model
- Testing the prediction model on the test data: the prediction model is applied to the test data to test its accuracy by comparing the result of the model with accrual data
- Deploying: finally, once the desired accuracy is reached, the model is applied to real-time data for prediction purposes.
- Evaluating a model: the process of adjusting the prediction model overtime to make it more accurate.

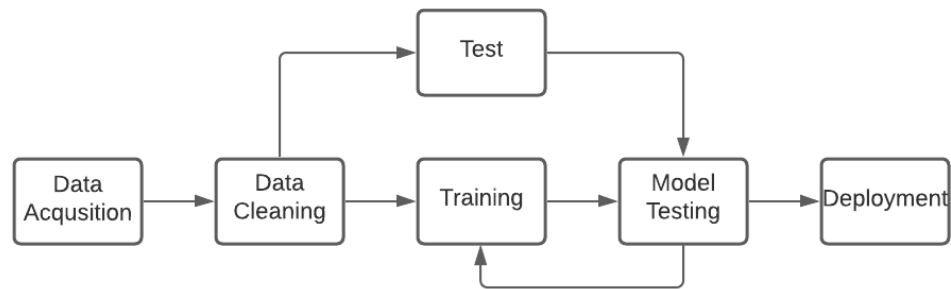


Figure 2: ML Technique Process for Prediction Model

Source: (Roosefert Mohan et al., 2021b)

2.5.1 Machine learning Methods

Machine learning techniques that are used for prediction models are categorized into:

- Supervised
- Unsupervised
- Reinforce learning

The supervised machine learning method is used when a set of data are tagged with a label or labels, this kind of data is named labeled data. Supervised machine learning technique tries to fit a function or model for the labeled data. In other words, as it is illustrated in figure 3, it tries to fit $h(x)$ while $h: x \rightarrow y$ for on training set with labeled data (G. James et al., 2021; Marzec et al., 2016).

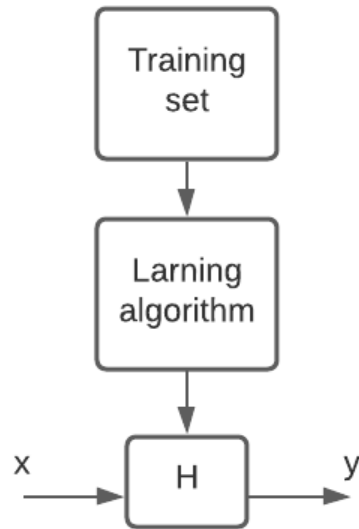


Figure 3: Supervised ML Technique Process

Source:(G. James et al., 2021)

In contrast to supervised learning, unsupervised learning methods are used for unlabeled to find clusters for a training set (G. (Gareth M. James et al., 2013.)

The reinforced learning method is considered an advanced ML technique that is being applied to self-driving cars and robotics subjects. In this method, a model is developed through learning from feedback received from the environment through the trial-and-error method (Allah Bukhsh et al., 2019)

2.5.2 Supervised Learning Methods

Supervised learning methods are categorized into two regression and classification methods. Regression models are generally used for quantitative variables while classification models

are used for qualitative variables (G. (Gareth M. James et al., 2013). Classification techniques or classifiers include logistic regression, naive Bayes, K-nearest neighborhood, and some more computer-intensive such as decision tree, random forest, boosting, and support vector machine(G. (Gareth M. James et al., 2013.).

Since in our case study (AAL stamping shop), it is assumed that we have access to historical labeled data of machines, the classification technique is chosen for the prediction model. Further in chapter 5, we will work on two decision tree and random forest classification techniques to build a prediction model for a machine failure, compare their accuracies, and finally apply one of them in the simulation model.

2.5.3 Decision Tree Method

The decision tree methods implement different algorithms which are based on recursive binary splitting of a training set. These algorithms try to optimize the homogeneity of each node and find the optimal thresholds for splitting a dataset. (G. James et al., 2021) (see figure 4).

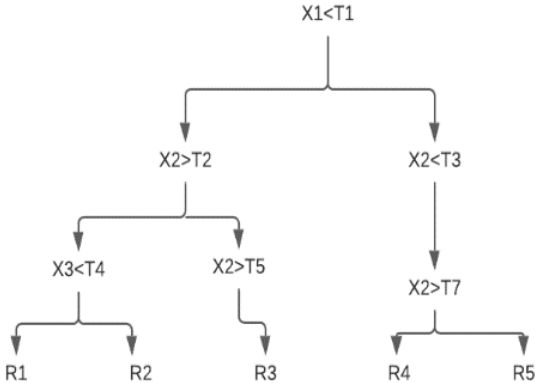


Figure 4: Decision Tree splitting plot

Source: (G. James et al., 2021)

The Decision Tree method is categorized to:

- Regression tree

In this method, a feature (x_j) is selected, and then at the feature's space is divided by a threshold t_m minimizing the residual sum of square (RSS) (see figure 5) (Allah Bukhsh et al., 2019; G. (Gareth M. James et al., 2013.):

$$\phi_{left}(j, t_m) = \{x | x_j \leq t_m\}$$

$$\phi_{right}(\theta) = \theta / \phi_{left}(\theta)$$

$$MIN \left(\sum_{i: x_i \in \phi_{left}} (y_i - y_{\phi_{left}})^2 + \sum_{i: x_i \in \phi_{right}} (y_i - y_{\phi_{right}})^2 \right)$$

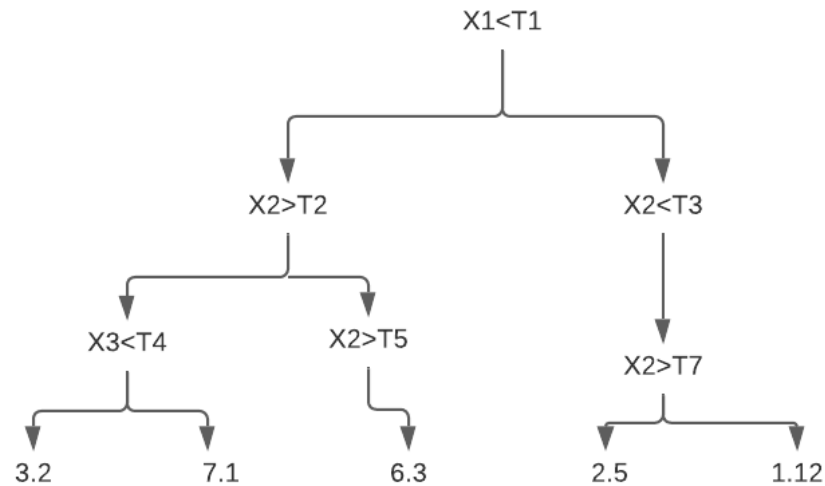


Figure 5: Regression Tree Plot

Source: (G. James et al., 2021)

- Classification Tree

The classification tree and regression tree algorithm are quite alike. The differences are in (Allah Bukhsh et al., 2019; G. (Gareth M. James et al., 2013.):

1. The classification tree is used for qualitative responses, whereas the regression tree is for quantitative data (see figure 6).
2. In analyzing the result of a classification tree, not only the partitioning of the training set but also the proportions of observation placed in each partition is important.
3. Considering the qualitative nature of responses data in the classification tree, impurity measures are used instead of RSS for this method such as Gini index, entropy (information gain), etc.

$$\text{Gini index: } G = \sum_{k=1}^K p_{mk}(1 - p_{mk})$$

$$\text{Entropy: } - \sum_{k=1}^K p_{mk} \log p_{mk}$$

- p_{mk} represents the proportion of data observations in the m th region that are from the k th class.

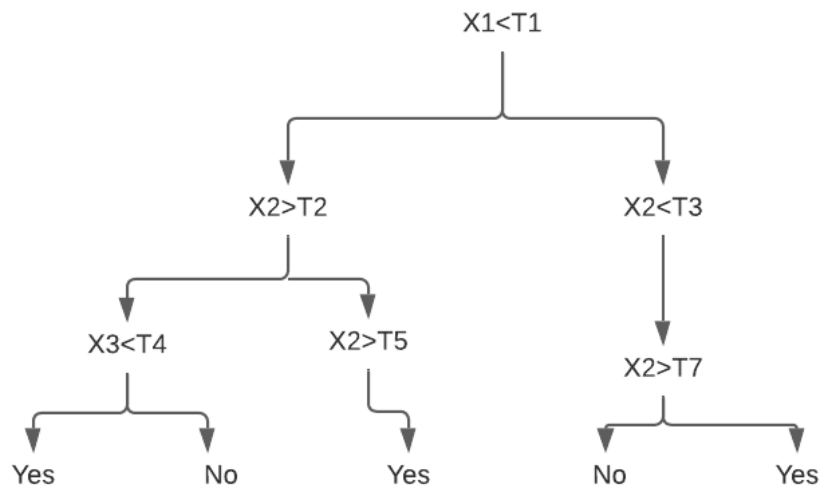


Figure 6: Classification Tree Plot

Source: (G. James et al., 2021)

2.5.4 Random Forest Decision Tree

Random forest is an ensemble learning built on the decision tree and is considered an upgrade version of the bagging method. The bagging method is used to reduce the high variance problem of the decision tree by working on n bootstraps training samples instead of a training

set and averaging the results. Random forest improves the problem of high correlation in the bagging method by forcing each split to work on a random subset of the feature instead of the entire feature set (see figure 7). (Allah Bukhsh et al., 2019; G. (Gareth M. James et al., 2013.; Random Forest - Wikipedia).

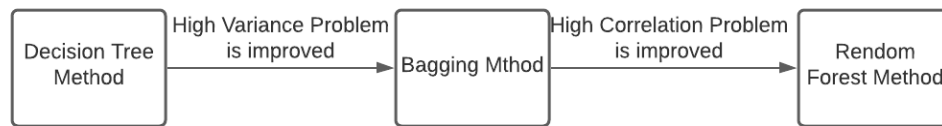


Figure 7: Decision Tree Versus Random Forest Methods

2.6 Methodology of Choice

As regards machine failure considerations in the JSS problem, which is the focus of this dissertation, the traditional simulation-optimization technique with simulation software platforms like Arena provides the possibility of assigning a distribution function based on the time or count.

The distribution function is obtained by statistical analysis to fit a distribution to machine historical data. The distribution function is defined in a failure module of the Arena to consider machine failure frequency and patterns for optimization of the JSS.

On the other hand, the digital twin technology provides a real-time data-driven simulation-optimization technique for machine failures as follows:

- Integrating ML and simulation models to apply prediction models for machine failures. Applying a prediction model for machine failures helps in the optimization of JSS by giving notices in advance for any unplanned machine failures. (Cavalcante et al., 2019).
- Enabling timely analyzing, rescheduling, and response. Therefore, in case of detecting the possibility of machine failures, the simulation model can apply CBM, do a what-if analysis, and timely response (see figure 8).

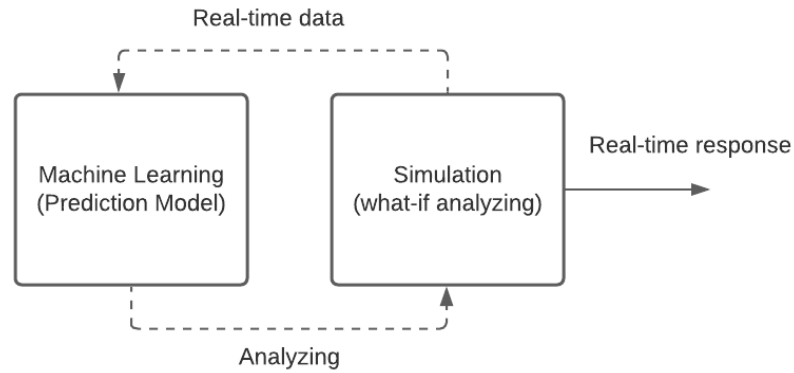


Figure 8: Real-Time Analysis and Timely-Response in Digital Twin

source: (Cavalcante et al., 2019)

Our methodology in this dissertation aims to show the application of the digital twin for optimization of the JSS problem by working on machine failures and comparing its results with the traditional simulation-optimization approach. To do so, we work on a case study which is the AAL company, and we go through the following processes in chapters 3, 4, and 5:

- Creating a simulation model for the JSS of the AAL stamping shop.
- Implementing machine failure approach in traditional simulation-optimization technique
- Implementing the machine failure approach in digital twin technology

3. Simulation Model for the JSS

To Build the simulation model for the JSS of AAL's stamping shop we go through the following steps:

1. Doing the system analysis of the stamping process
2. Defining the scheduling objectives and strategies at ALL company
3. Creating the simulation model for the JSS in Arena simulation software

3.1 System Analysis

The sheet metal stamping process is used in many manufacturing sectors from automotive to aircraft industries. It also supports many other downstream manufacturing processes in the production line such as welding, assembling, and painting. Therefore, implementing an optimal JSS for the stamping process leads to the optimization of the whole manufacturing system (Roychowdhury et al., 2017). important challenges in this regard include:

1. production/ delay
2. Setups
3. Die-set constrains
4. Inventory capacity

AAL company's final product is the car body panel manufactured through stamping, welding, and painting processes. The stamping shop which is the focus of this dissertation is in one of the AAL plants (Plant A) supplying body panels for other processes. The Production Planning and Controlling (PPC) department of AAL is responsible for the scheduling and production planning of car body panels. The PPC uses the traditional simulation-optimization method for JSS problems, but since the digitalization of the

manufacturing line is done, the company is willing to apply digital twin concepts by implementing a real-time simulation-optimization technique.

The stamping shop processes at ALL as it is shown in figure 9 are as follow:

1. The sheet metals are moved by a forklift in a batch of twelve from sheet metal storage to the conveyor station
2. The sheet metals are hand loaded from the forklift, visually checked, and placed on the conveyor by two workers.
3. The sheet metals are conveyed toward the station of stamping machines.
4. The type of body panel is checked in case of needing to change dies for the stamping process. Since five different body panels are manufactured at the AAL stamping shop, each type of body panel needs specific dies for the stamping process
5. The sheet metals stamping process is done based on the sequence of assigning jobs J1, ..., J5 (type of body panels) to machines M1, ..., and M6.
6. The outputs of the stamping process named body panels are picked up from the conveyor and placed in a pallet for a quality check.
7. The quality check of each body panel is done in the highlight room.
8. And the pallets of body panels are moved to the storage to be transported to other factories for welding and painting processes.

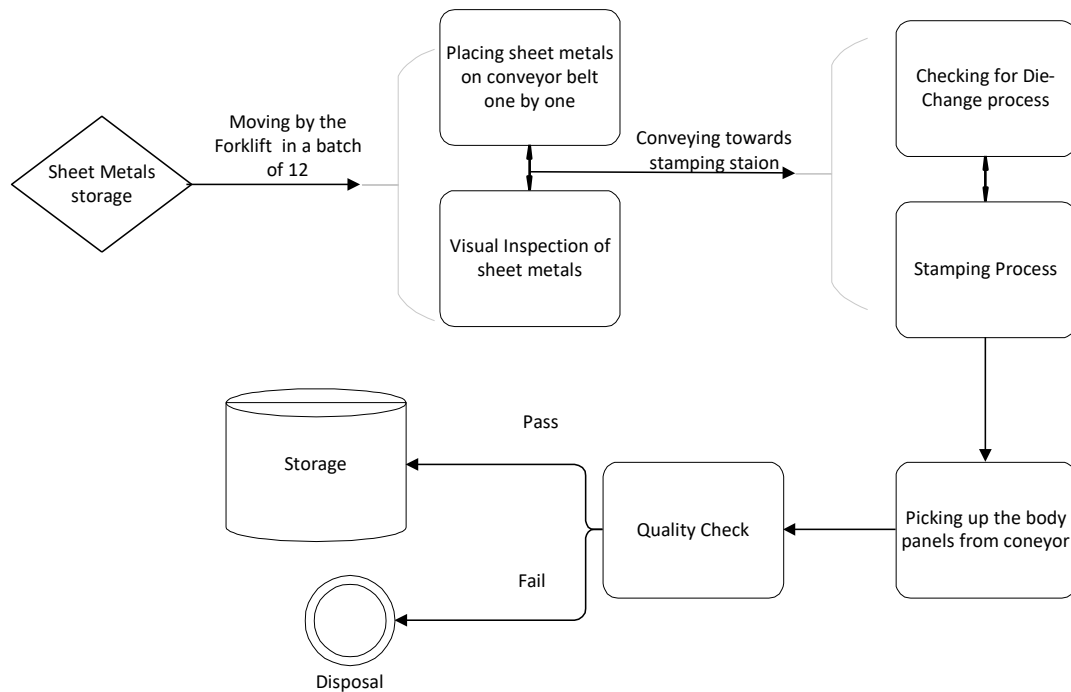


Figure 9: AAL Stamping Shop Manufacturing Process

3.2 Scheduling Strategies

To efficiently model and optimize the JSS problem, it is necessary to create the simulation model based on the scheduling strategies. These strategies are defined by policies, aims, agreements, and operational constraints of AAL company which are as follows:

- To meet the daily demands committed to being delivered to each customer.
- To minimize the total tardiness costs in case of not being able to deliver the products on time.
- To meet the painting department priorities.

- To consider the constraints of the die-change process
- To consider the sequence of stamping process defined by assigning jobs to machines for each type of body panel.
- To consider the stamping shop storage capacity for final products (body panels)

3.3 Creating a Simulation Model for the JSS with Arena

Considering the system analysis and scheduling strategies of AAL company, we create a JSS simulation model containing scheduling and manufacturing processes with the following sections:

- Scheduling process
 - Categorizing and batching demands based on the painting department's priorities
 - Assigning jobs to the machines based on manufacturing requirements for each job
 - Searching and selecting through batches in the order of the priorities defined by the painting department to find batches with minimum tardiness cost
 - Dispatching and sequencing jobs by placing them in a queue. The queue is considered the JSS for the manufacturing processes.
- Manufacturing process
 - Transporting and conveying process
 - Die-change process

- Stamping process
- Quality control and storage process

3.3.1 Scheduling Process

3.3.1.1 Categorizing and Batching Demands Based on Their Priorities

In this section, the daily demands announced by the sales department (see table 2) are categorized based on their painting priority. Additionally, the sequence of assigning jobs (VM1101, ..., VM1333) to machines (M1, ..., M6) for the stamping process is done based on table 2.

Table 2: Demands and Stamping Sequences of Body panels

Type	Daily demands	M1	M2	M3	M4	M5	M6
VM1002	53	1	1	1	1	0	1
VM1011	40	1	1	1	1	1	1
VM1030	24	1	1	1	1	0	1
VM2011	35	1	1	1	1	1	1
VM1333	18	1	1	1	0	1	1

To categorize our demands in Arena simulation software we use five Create Modules to create the entities of each job (see figure 11) Additionally, we assign the sequence of jobs to machines in the sequence module of the Arena software (find more information in section 3.3.1.3) Moreover, a distribution function (DISC) is used to define the color of body panels (white, black, and silver) based on the percentage announced by the sales department for the daily production. For example, as is shown in figure 10 the DISC (0.4, 1, 0.7, 2, 1.0, 3) is assigned to divide the VM1011 daily demand to %40 white, %30 black, and %30 silver. Finally, according to paint department priorities (the pint department prefers to receive the products based on the similarity of their color due to its batch painting process), the entities

are prioritized and categorized in three separate queues (the priority is defined first for the white, second for the black, and last for the silver jobs).

In short, the outputs of this section are:

- Batching the jobs based on their types and daily demands
- categorizing and placing batches in three different queues based on the priority of the painting department
- Assigning the body panel types to stamp machines for the stamping process. Referring to table 2, for example, the VM1001 body panel is assigned to stamp machine 1 (M1), machine 2 (M2), machine 3 (M3), machine 4 (M4), and machine 5 (M5).

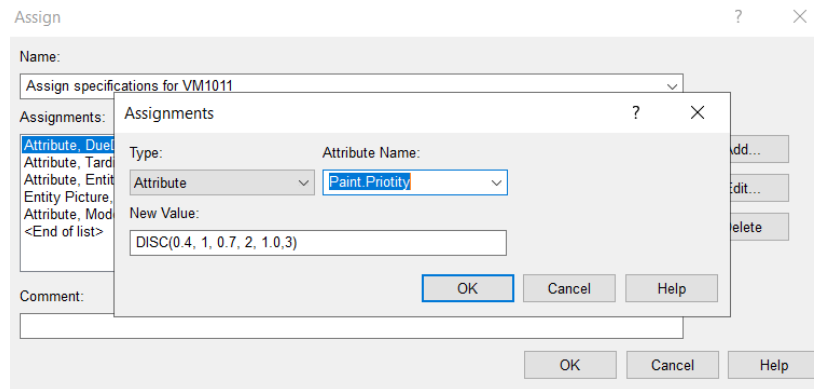


Figure 10: Assigning Paint Department Priorities

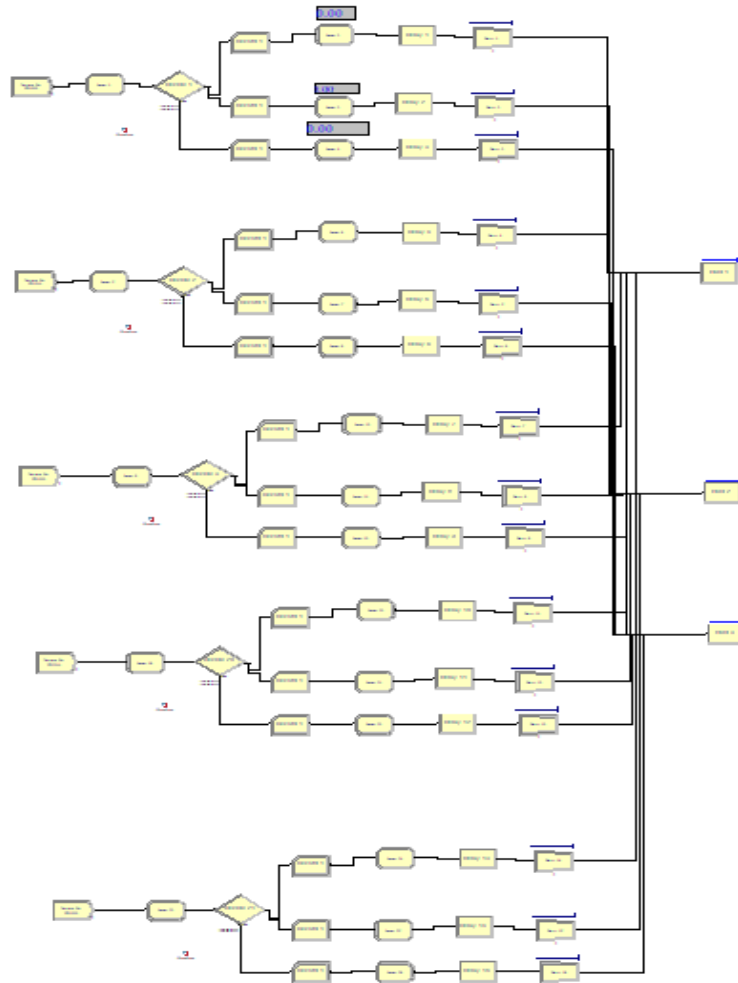


Figure 11: Categorizing Demands in Arena Simulation Software

3.3.1.2 A Decision Model for Job Scheduling

In this part of the simulation, we build a decision model algorithm to determine the sequences of jobs for the stamping process. Since the scheduling strategy of AAL is to minimize the total tardiness costs, we create an algorithm to search in the three queues established in the previous section based on the painting department's priorities to find and remove batches with minimum tardiness.

As it is demonstrated in figure 12, we build the decision model algorithm mostly by using search, remove, and hold modules in the Arena simulation software to search through the queues and pick those batches with minimum tardiness

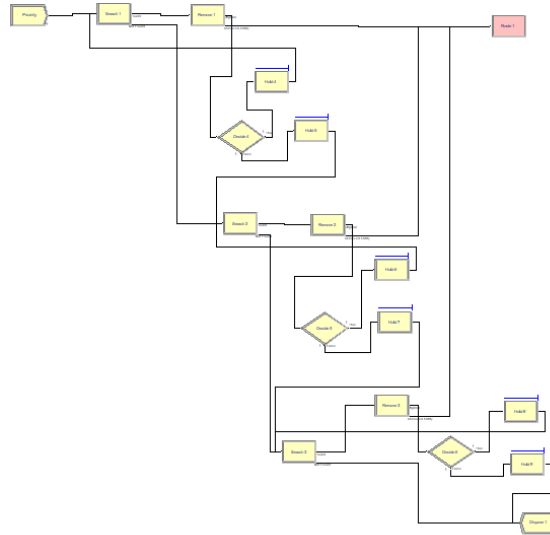


Figure 12: The Decision Model for Job Scheduling in Arena Simulation Software

3.3.1.3 Job Shop Sequencing

This section aims to finalize the scheduling process by sequencing jobs for the manufacturing operation. This part of the simulation is in connection with the decision model created in the previous section by signal modules. Those batches selected at the decision model section are sent to this section one after another, dispatched, and placed in order in a queue (see figure 13).

Therefore, the queue represents jobs sequenced based on the AAL scheduling strategies to go through the manufacturing process.

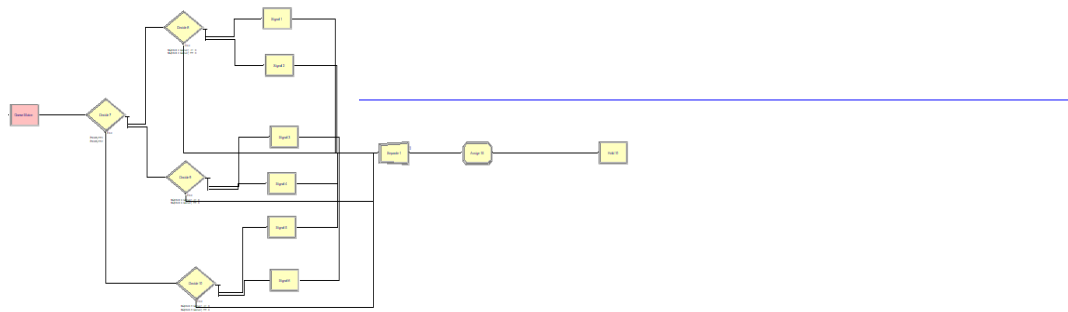


Figure 13: Job Shop Scheduling Model in Arena Simulation Software

3.3.2 Operation Process

3.3.2.1 Transport Section

In this section of simulation, we model transporting processes in the stamping shop as follows (see figure 14):

- Transporting sheet metals by a forklift from stock in a batch of twelve toward the conveyor belt.
- lifting the sheet metals and placing them on the conveyor. This is done by assigning two workers to the resource module of the Arena software.
- Conveying sheet metals toward the stamping machines station with a conveyor. We use the conveyor module in the Arena in this regard.

Moreover, we duplicate the entity of each job picked up for the transport and send it to the die-change section in the simulation model to be checked for the die change process. We mostly use transport and conveyor modules in the Arena software to simulate the process of transport in the stamping shop.

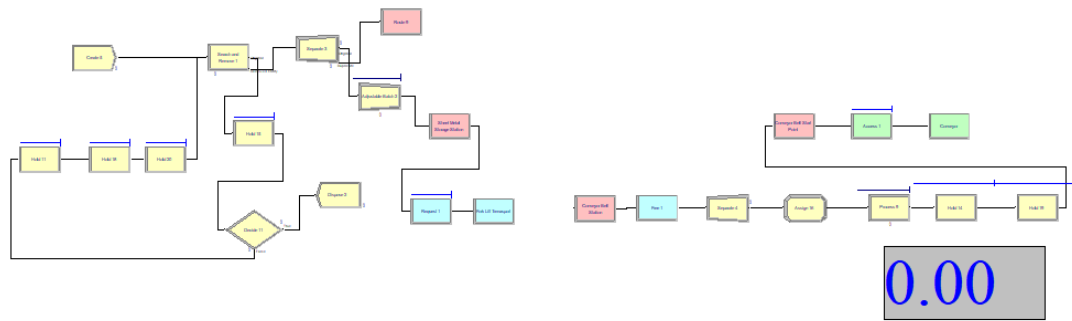


Figure 14: Transport Section in Arena

3.3.2.2 Die-Change Process Algorithm

This part of the situation aims to compare the type of body panel picked up for the stamping process by the next order scheduled in the job sequencing queue (see figure 15). Therefore, if their types are different, the stamping process is held for the next order and the die-change process is done.

An algorithm is defined in the simulation model to compare the duplicated entity sent by the transport section with the type of the next entity scheduled for the stamping process (see figure 15). We use the code below in the diction module used in Arena to check the two consecutive orders:

```
Model<>AQUE (Hold 271.Queue,1, NSYM(Model))
```

In this code:

- Hold 27 is represented the job sequencing queue in this for
- The model represents the type of body pane.

If their types of body panels are alike, the duplicated entity will be disposed of; otherwise, we define an algorithm in the simulation model to first wait for stamping machines to finish

the in-progress order, second to hold the stamping process, and finally to do the die-change process for the next order.

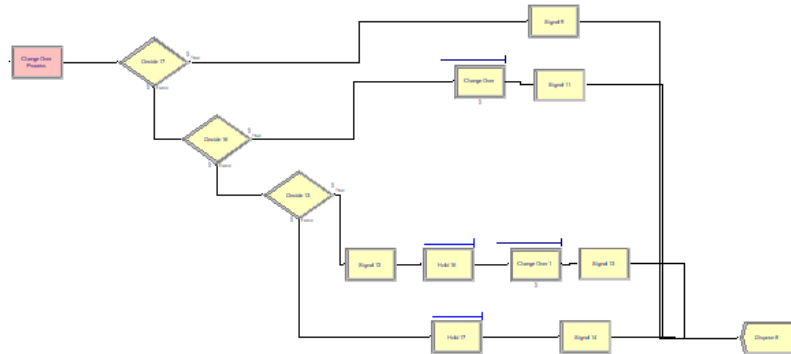


Figure 15: Decision Module for Comparing two Consecutive Entities in a Queue

3.3.2.3 Stamping process

The sheet metals that are already conveyed by the conveyor to the stamping machines station go through the stamping process based on the sequences defined for each type of body panel (referring to table 2). The assigning of jobs to machines in Arena is done in the sequence module as it is demonstrated in figure 16.

		Steps	
		Station Name	St
1		M1	
2		M2	
3		M3	
4		M4	
5		M6	

		Name	Steps
1	▶	VM1002	5 rows
2		VM1011	6 rows
3		VM1030	5 rows
4		VM2011	6 rows
5		VM1333	5 rows

Figure 16: The Sequence Module in Arena

We use the station and process modules in the Arena to guide jobs from one machine to another based on the sequences defined for each body panel and to simulate the stamping process for each machine (see figure 17).

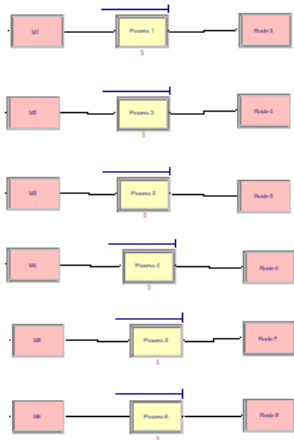


Figure 17: Stamping process model in Arena

3.3.2.4 Quality Control and Storage

In this section of the manufacturing process, the body panels that are stamped in the previous section are picked up from the conveyor, batched in the pallet of six, checked for quality, and stored temporarily to be transported to the welding and painting departments later (see figure 18).

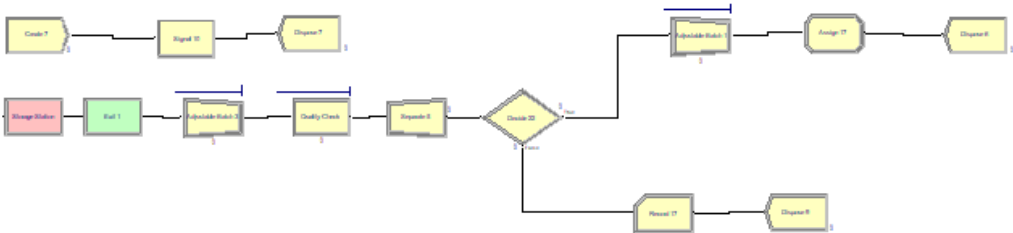
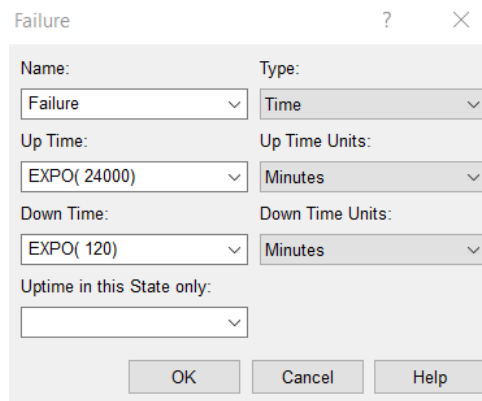


Figure 18: Quality Control and Storage Model in Arena

4. Machine Failure Approach in Traditional Simulation-Optimization Technique

As described before, one of the advantages of the simulation-optimization technique over the mathematical technique for optimal scheduling problems is the possibility of considering machine failures in the model. In the traditional-simulation optimization technique for the JSS is possible to assign a distribution to resources such as machines to consider machine failures in the simulation model. As it is shown in figure 19, Arena software provides a failure module to define a distribution function for resource failures with the following options (Kelton et al., 2015):

- Type: define a time-based or count-based type of distribution
- Up Time: a distribution function used to show the up-time pattern of a resource
- Down Time: a distribution function used to show the downtime pattern of a resource. For example, in case of occurring a machine failure, proper distribution is used to simulate the pattern of time needed to solve the failure.



The screenshot shows a dialog box titled "Failure" with a question mark and a close button (X). The dialog contains the following fields:

Name:	Type:
Failure	Time
Up Time:	Up Time Units:
EXPO(24000)	Minutes
Down Time:	Down Time Units:
EXPO(120)	Minutes
Uptime in this State only:	

At the bottom of the dialog are three buttons: OK, Cancel, and Help.

Figure 19: Failure Module in the Arena Simulation Software

4.1 Distribution Fitting Process

In this method, we apply statistical analysis to fit a probability distribution function to the historical data representing the machine conditions including the HDF problem. For the fitting process, it is important to select proper distribution functions based on the type of process and data. If we assumed that each record of data is captured every 30 minutes. It means the dataset presents 300,000 working minutes or 5,000 working hours of the machine (the dataset contains 10,000 records).

As it is shown in the distribution plot (see figure 20) the 115 times the HDF problem occurred between 97,710th and 145,560th working minutes of the machine. Therefore, we can make two assumptions about the HDF failure reasons:

1. It is triggered by external factors which are not directly related to the machine's function, factors can be related to the working environment of human errors.
2. It is triggered because of the wear and tear of the machine's parts which is caused after a specific amount of working time or use.

```
Time_HDFdataset = pd.read_csv('Time.csv')  
  
sns.displot (data =Time_HDFdataset, x="Time (min)", y= "HDF")
```

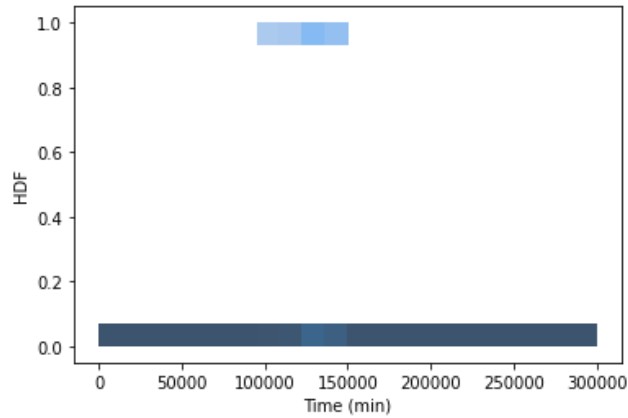


Figure 20: The Distribution Plot of the HDF Problem

Regarding the first assumption, the distribution fitting process in the traditional simulation-optimization approach is not able to evaluate the external factors on machine function. The cause-and-effect evaluations in complex systems need complicated methods such as ML techniques which will be applied in chapter 5.

The second assumption is applicable in the traditional simulation optimization by fitting a proper distribution resembling the frequency and pattern of the failures. Referring to figure 20 the frequency of the HDF can be categorized into three sections:

- No HDF machine failure before 97,710th minutes
- 115 times of HDF machine failure between 97,710th and 145,560th minutes
- No HDF machine failure after 145,560th minutes

Therefore, since we do not have access to data after 300,000th minutes or some other samples with the same working time of the machine to evaluate the pattern of failures, to increase the reliability of the system we work on the worse condition which is between 97,710th and 145,560th working minutes of the machine.

To fit a probability distribution for the interval between failures, we go through the following steps:

- Providing the descriptive statistics of the interval between HDF failures to have a better understanding of data distribution (see table 3)
- Performing Grubbs test to find outliers and cleaning the data for HDF interval between failures
- Defining the number of bins and the length of each bin to plot the histogram and evaluate the frequency of failures
- Fitting most probable distributions by using the input analyzer in Arena software and comparing the results

4.2 Descriptive Statistics of Interval HDF problem

Descriptive statistics provide an overview of the distribution of data including the range, standard deviation, mean, minimum, maximum, etc. (see table 3) (*Overview for Descriptive Statistics - Minitab Express*, n.d.). By looking at the descriptive statistics, we can find the possibility of having outliers due to the wide gap between the mean and the maximum numbers (*Interpret the Key Results for Display Descriptive Statistics - Minitab*, n.d.). Additionally, since the mean is greater than the median, the possible distribution functions should have positive skewness, the characteristic that is shown in the Exponential, Gamma, and Weibull distribution functions.

Table 3: Descriptive Statistics

Variable	N	N*	CumPct	Mean	SE Mean	StDev	Minimum	Q1	Median	Q3	Maximum
Time-Interval (min)	115	0	100	1266	854	9158	30	60	180	300	97110

Variable	Range	Skewness
Time-Interval (min)	97080	10.27

4.3 Outlier Test

We use Minitab software to apply the Grubbs and Dixon's Q (Adikaram et al., 2015) for the outlier tests with the below hypothesis and %5 significance level ($\alpha = 0.05$):

- H0: if all the data come from the same normal distribution
- H1: if one of the values in the dataset does not come to form the same normal distribution

Both outlier tests consider the 115th row of the database as an outlier (see table 4 and figure 21).

Table 4: Outlier Test Result

Variable	Row	Outlier
Time Interval	115	97110

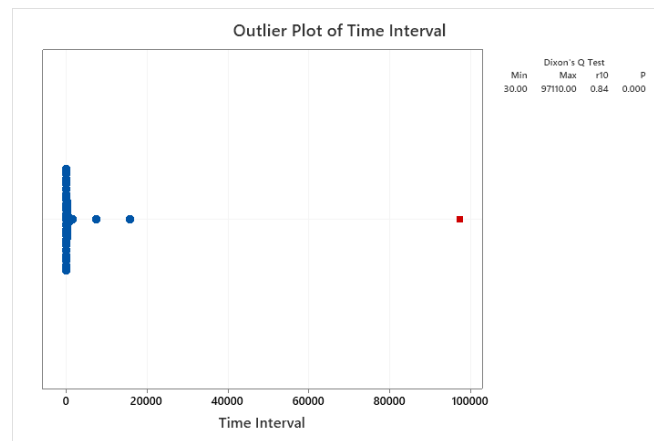


Figure 21: Outlier Test Diagram

4.4 Histogram

After removing the outlier from the dataset, we plot the histogram diagram to check and interpret the frequency and cumulative percentages of data based on the intervals (or bins). To define the number of bins and length of the bin for the histogram, we use Sturge's rule as follows(Scott, 2010):

$$K = [1 + 3.322 \log_{10} n]$$

K = the number of bins

n = 115 (number of data)

As is shown in table 5, Sturge's rule indicates 7 bines with a length of 13868 approximately.

Table 5: Specification of Histogram

Time Interval	Data
Average(min)	1265.74
Min	30.00
Max	97110.00
Median	180.00
Mode	60
Variance	83863119.41
Standard Deviation	9157.68
Count	115.00
Skewness	10.27
Kurtosis	107.92
Range	97080.00
Number of bins	7
Length of bins	13868.571

Therefore, we use the Minitab software to plot the histogram and the results are shown in table 6 and figure 22.

Table 6: Frequency and Cumulative Table of Interval between failures

<i>Bin</i>	<i>Frequency</i>	<i>Cumulative %</i>
2241.43	112	98.25%
4482.86	0	98.25%
6724.29	0	98.25%
8965.71	1	99.12%
11207.14	0	99.12%
13448.57	0	99.12%
15690.00	0	99.12%
More	1	100.00%

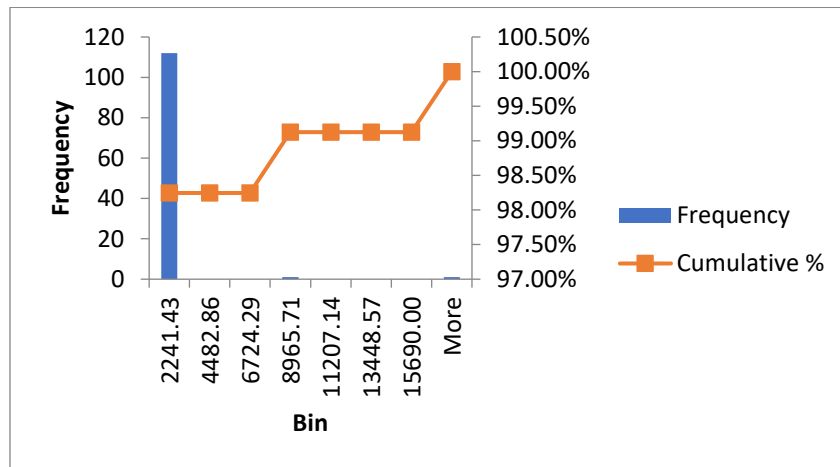


Figure 22: Frequency and Cumulative Diagram of Interval Between Failures

Considering the information that we have obtained through the descriptive statistics and the histogram, we select the Exponential, Gamma, and Weibull probability distributions for the fitting process. Later in the result chapter, we use the Input Analyzer tool in Arena simulation software to fit these distributions and compare the results of square error and chi-square error tests for each one.

5. Machine Failure Approach in Digital Twin Technology

As it is described in section 2.3, the digital twin provides a real-time data-driven simulation-optimization technique that can use the ML platforms for timely analysis and prediction. In this chapter, we are going to use ML techniques to build a prediction model for one of the stamping machines in our case study. Then, we apply the result of the prediction in the simulation model to optimize the JSS for AAL's stamping shop.

To implement ML techniques and work on historical data of the machine, we go through the following steps:

- Data cleaning
- Data visualization and analyzing
- Training prediction model on the training set
- Testing prediction model on the test set

We use the python programming language to implement ML techniques and import the following libraries in python (Pedregosa et al., 2011; VanderPlas, 2016):

- Numpy library for loading, storing and preprocessing data
- Pandas for working on data frames and arrays
- Matplotlib for data visualization
- Seaborn for high-level statistical plot type
- Scikit-learn for ML techniques
- Graphviz for showing decision tree diagrams

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import graphviz
```

5.1 Importing and Cleaning Dataset

The dataset shown in table 7 is the historical data of a stamping machine containing features as follows:

Features related to machine performance collected from sensors:

- air temperature [K}
- process temperature [K]
- rotational speed [rpm]
- torque [Nm]
- tool wear [min]

Features that are considered as labeled data indicating the condition of machines (failure or normal):

- UID: unique identifier numbering the machine data records and maintenance logs from 1 to 10000
- ProductID: a specific number related to products
- Machine failure: labeling whether the machine was at the failure “1” or normal “0”.

Features represent the root causes (or mechanical causes) of failures:

- tool wear failure (TWF)
- heat dissipation failure (HDF)
- power failure (PWF)
- overstrain failure (OSF)

The historical data is read and named as “dataset” in our programming. The dataset has 10,000 records and 14 columns with the information described in table 8.

```
dataset = pd.read_csv('/content/ai4i2020.csv')
dataset.info()
```

Table 7: Historical Data of Stamping Machine

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

Table 8: The Dataset Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column                               Non-Null Count  Dtype
---  -
0   UDI                                   10000 non-null  int64
1   Product ID                           10000 non-null  object
2   Type                                  10000 non-null  object
3   Air temperature [K]                   10000 non-null  float64
4   Process temperature [K]               10000 non-null  float64
5   Rotational speed [rpm]                10000 non-null  int64
6   Torque [Nm]                           10000 non-null  float64
7   Tool wear [min]                       10000 non-null  int64
8   Machine failure                       10000 non-null  int64
9   TWF                                    10000 non-null  int64
10  HDF                                    10000 non-null  int64
11  PWF                                    10000 non-null  int64
12  OSF                                    10000 non-null  int64
13  RNF                                    10000 non-null  int64
dtypes: float64(3), int64(9), object(2)
memory usage: 1.1+ MB
```

We also check each column to find if there is any missing data (see table 9). Furthermore, the heatmap diagram is used for data visualization (see figure 23) indicating that there is no null field in the dataset.

```
dataset.isnull()
sns.heatmap (dataset.isnull(),yticklabels=False, cbar=False, cm
ap='viridis')
```

Table 9: Checking for Null fields

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
9995	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9996	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9997	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9998	False	False	False	False	False	False	False	False	False	False	False	False	False	False
9999	False	False	False	False	False	False	False	False	False	False	False	False	False	False

10000 rows x 14 columns

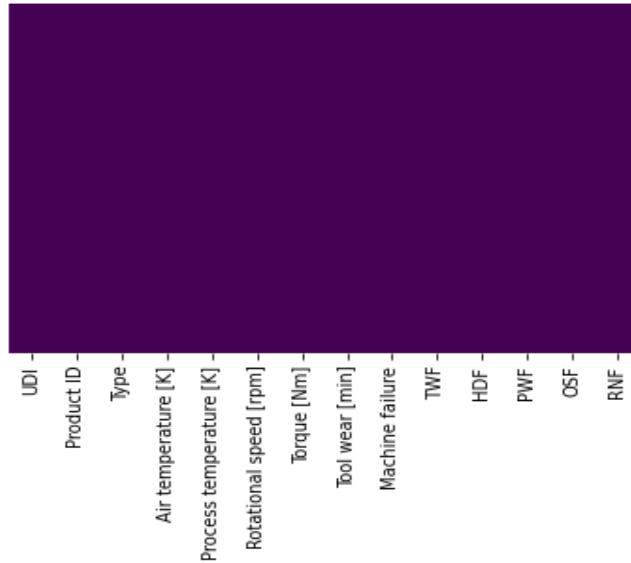


Figure 23: Heat Map for Data Visualization

5.2 Data Analyzing and Visualization

If we assume that the data is captured every 30 minutes, it means that the 10,000 records of data are collected in 5,000 working hours of the stamping machine. Of the 10,000 records, the machine experienced 339 times of failures which is 3.39% of the total records.

As is illustrated in table 10 the root causes of the 339 failures are detected for:

- 46 times of TWF
- 115 times for HDF
- 95 times for PWF
- 98 times for OSF
- And one time for RNF

Some of the failures were blamed to have more than one causes.

```
dataset[dataset['Machine failure']==1][['TWF', 'HDF', 'PWF', 'OSF', 'RNF']].apply(pd.value_counts)
```

Table 10: Root Causes of Failures

	TWF	HDF	PWF	OSF	RNF
0	293	224	244	241	338
1	46	115	95	98	1

Furthermore, we use the boxplot diagram to find the data ranges and outliers for each performance feature (see figure 24). However, Since the dataset contains both normal and failure conditions, we do not remove the outlier in our model. In fact, we aim to fit a function to show a relationship between failures and performance and the outliers may reflect the relationship.

```
plt.figure(figsize=(10,7))
sns.boxplot(y='air temperature [K]', data=dataset)
sns.boxplot(y='process temperature [K]', data=dataset)
sns.boxplot(y='rotational speed [rpm]', data=dataset)
sns.boxplot(y='torque [Nm]', data=dataset)
sns.boxplot(y='tool wear [min]', data=dataset)
```

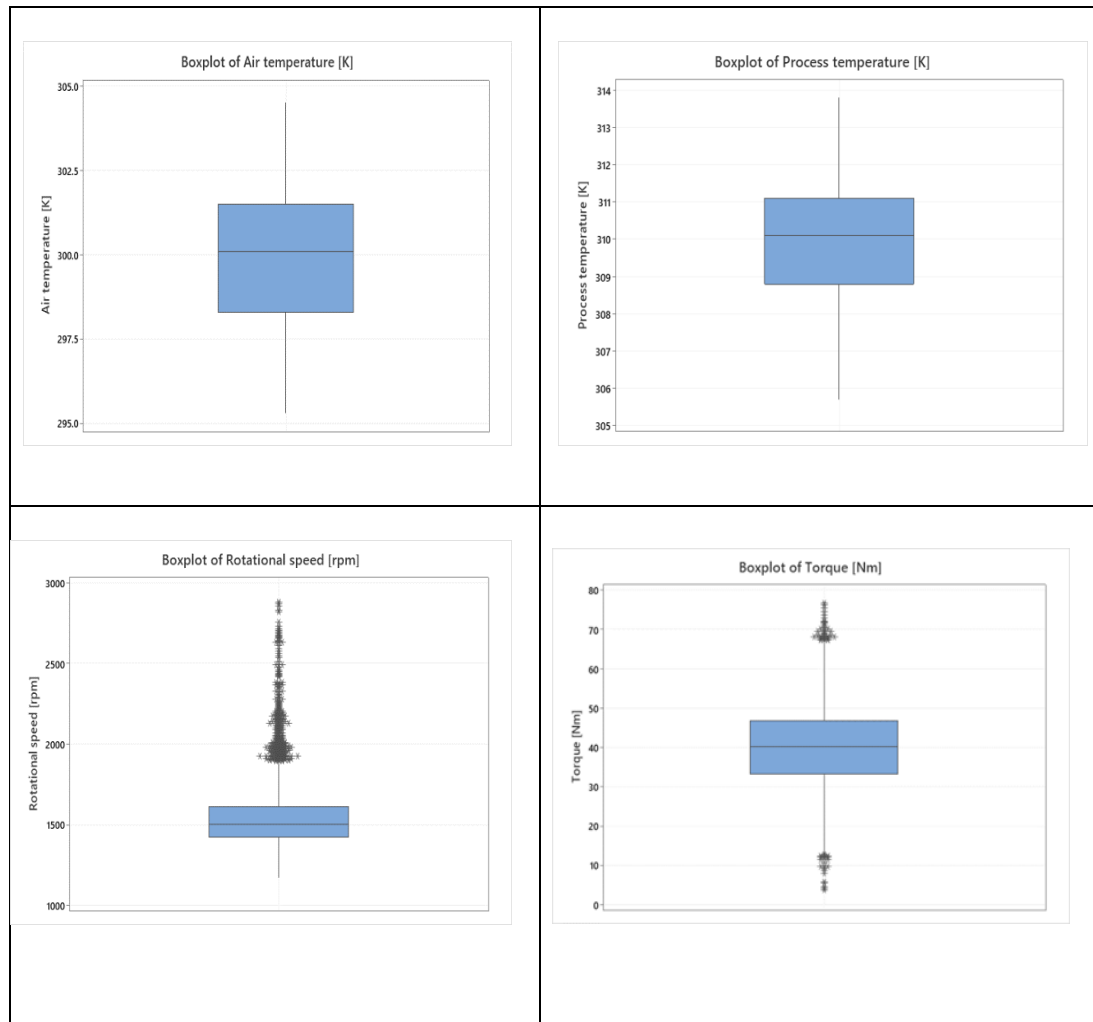


Figure 24: Box -Plot diagram for Machine Performance Features

In this dissertation we narrow down the scope of the case study and apply ML techniques to fit a prediction model for just those failures caused by the HDF problem; however, the same method can be applied to other failure problems as well.

To compare data distribution in failure with normal conditions, we use the boxplot diagram to indicate the median, minimum, maximum, first quarter (Q1), and third quarter (Q3) (see figure 25).

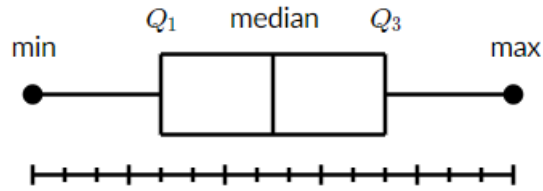


Figure 25: Box Plot Diagram

Source(*Box Plot Review (Article) | Khan Academy*, n.d.)

Therefore, we create the boxplot for each feature and compare the data distribution. As it is illustrated in figure 26, the difference in data distribution is obvious in normal conditions compared to failure conditions specifically in some features such as the air temperature, process temperature, rotational speed, and torque.

```
plt.figure(figsize=(10, 7))
sns.boxplot(x='HDF', y='Air temperature [K]', data=dataset)
sns.boxplot(x='HDF', y='process temperature [K]', data=dataset)
sns.boxplot(x='HDF', y='rotational speed [rpm]', data=dataset)
sns.boxplot(x='HDF', y='torque [Nm]', data=dataset)
sns.boxplot(x='HDF', y='tool wear [min]', data=dataset)
```

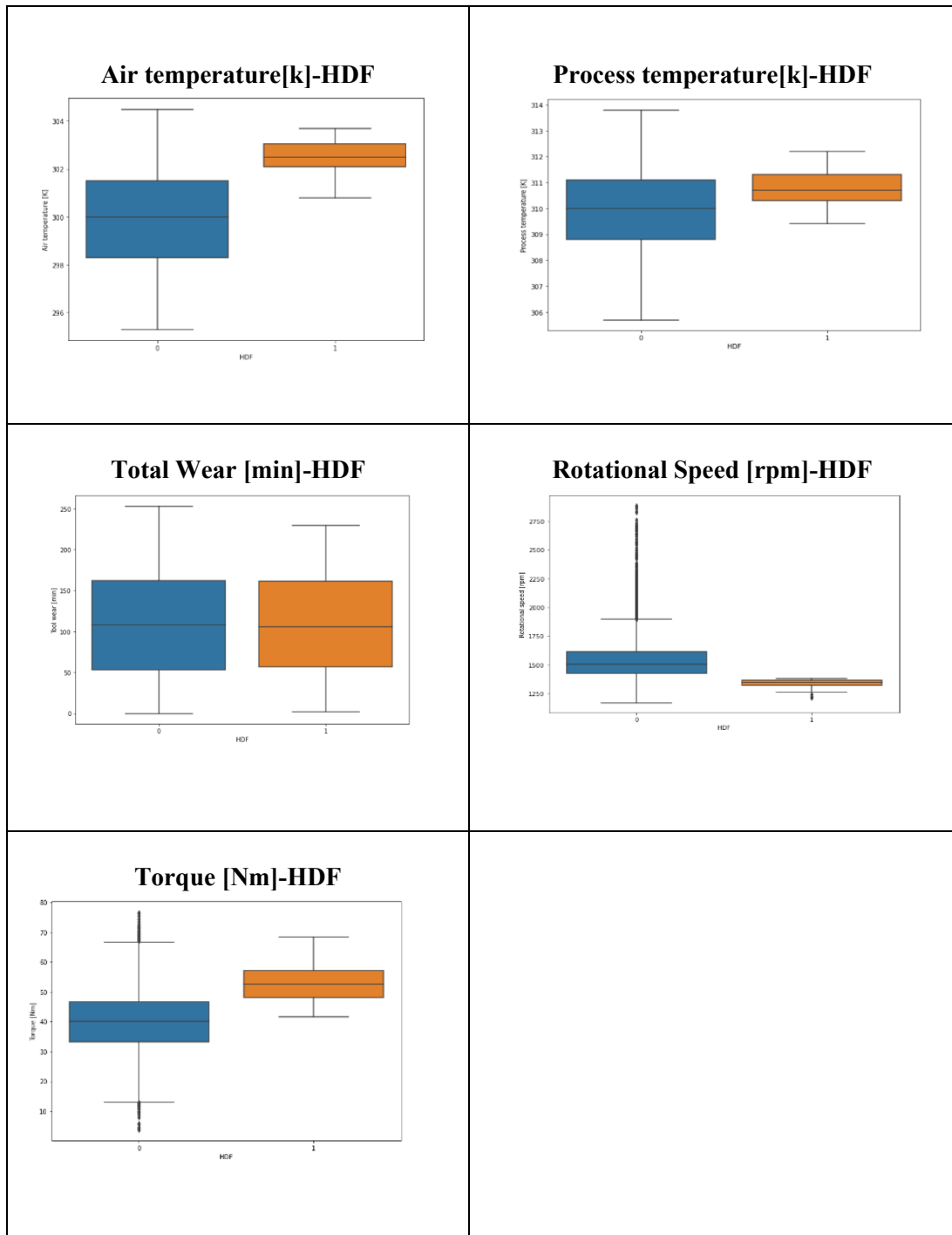


Figure 26: Box-Plot Diagram for HDF problem

Moreover, to have a better understanding of the relationship between variables (features), we compare the correlation among features in the normal conditions (see table 11) and the failure conditions (see table 12) caused by the HDF problem.

```
dataset[dataset['Machine failure'] == 1][['TWF', 'HDF', 'PWF',
'OSF', 'RNF']].apply(pd.value_counts)
HDFdatasetF = dataset[dataset['HDF'] == 1].drop(['UDI', 'Product ID', 'Type', 'Machine failure', 'TWF', 'PWF', 'OSF', 'RNF'], axis =1 )
HDFdatasetN = dataset[dataset['HDF'] == 0].drop(['UDI', 'Product ID', 'Type', 'Machine failure', 'TWF', 'PWF', 'OSF', 'RNF'], axis =1 )
HDFdatasetF.corr()
HDFdatasetN.corr()
```

Table 11: Correlation Between Variables in Machine Normal Condition

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	HDF
Air temperature [K]	1.000000	0.878105	0.040086	-0.033911	0.014401	NaN
Process temperature [K]	0.878105	1.000000	0.026446	-0.022063	0.013849	NaN
Rotational speed [rpm]	0.040086	0.026446	1.000000	-0.874446	0.000163	NaN
Torque [Nm]	-0.033911	-0.022063	-0.874446	1.000000	-0.003025	NaN
Tool wear [min]	0.014401	0.013849	0.000163	-0.003025	1.000000	NaN
HDF	NaN	NaN	NaN	NaN	NaN	NaN

Table 12: Correlation Between Variables in Machine Failure Condition- HDF problem

	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	HDF
Air temperature [K]	1.000000	0.899555	-0.000011	-0.132221	-0.042087	NaN
Process temperature [K]	0.899555	1.000000	0.004677	-0.145829	-0.034769	NaN
Rotational speed [rpm]	-0.000011	0.004677	1.000000	-0.543797	-0.042580	NaN
Torque [Nm]	-0.132221	-0.145829	-0.543797	1.000000	0.008511	NaN
Tool wear [min]	-0.042087	-0.034769	-0.042580	0.008511	1.000000	NaN
HDF	NaN	NaN	NaN	NaN	NaN	NaN

Finally, we use the pair plot diagram for data visualization to see correlations between variables by categorizing data into red points (when HDF=0) and blue points (when HDF=1) (see figure 27).

```
sns.pairplot(HDFdataset, hue='HDF', palette = 'Set1')
```

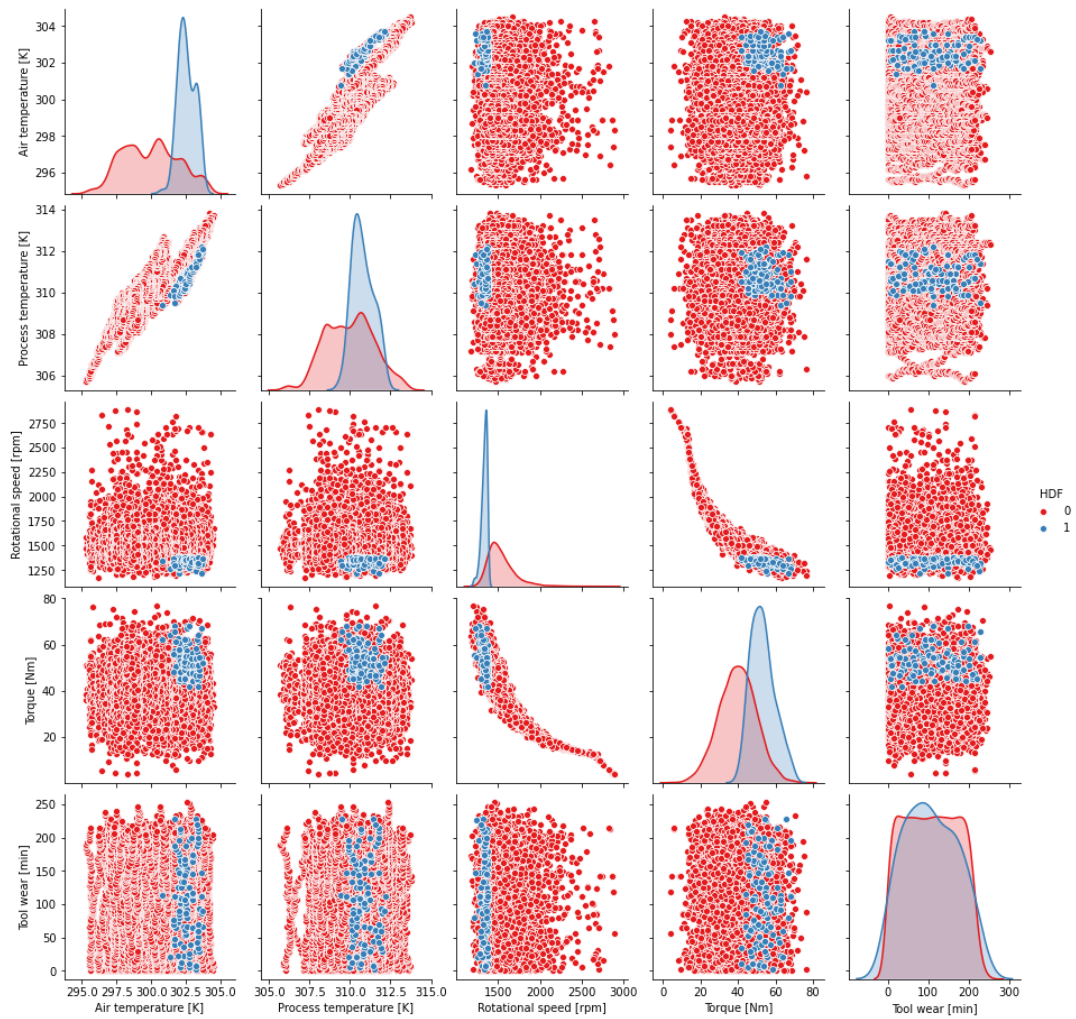


Figure 27: The Pair-Plot Diagram

5.3 Machine Learning Technique for Prediction Model

To propose a prediction model for the stamping machine HDF failure, we use ML techniques to work on the machine's historical data with the following processes:

- Defining dataset and variables
- Splitting the HDF dataset into a training set and test set
- Training decision tree and random forest ML techniques on the HDF dataset
- Test the results of the ML techniques on the test set (which will be discussed in the next chapter)

5.3.1 Defining Dataset and Variables

To define the dataset, first, we drop those columns of the dataset that are not required in training the prediction model including:

'UDI', 'Product ID', 'Type', 'Machine failure', 'TWF', 'PWF', 'OSF', 'RNF' .

Second, we name the new format of the dataset “HDFdataset”.

Finally, we define the independent variables (features) as ‘X’ contained all the columns except the last one and the dependent variables (responses) as ‘y’ contained just the last column.

Therefore, HDFdataset contains air temperature [K], process temperature [K], rotational speed [rpm], torque [Nm], tool wear [min], and HDF columns.

```
HDFdataset = dataset.drop (['UDI', 'Product ID', 'Type', 'Machine failure', 'TWF', 'PWF', 'OSF', 'RNF'], axis =1 )
X = HDFdataset.iloc[:, :-1].values
y = HDFdataset.iloc[:, -1].values
print(X_train)
```

5.3.2 Splitting Dataset into Training Set and Test Set

To avoid overfitting problems and to be able to validate the prediction model, we split the HDF dataset into a training set and a test set (see figure 28). This approach helps us to train ML techniques on the training set and then validate the model on the test set.

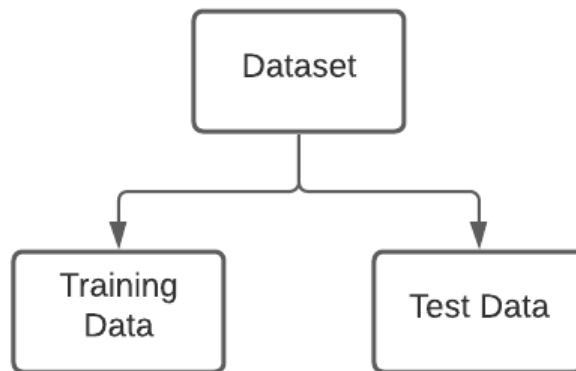


Figure 28: Splitting the HDF dataset Into Training data and Test data

The portion of the split is usually ranged as high as around 70 to 80 % for the training set and the rest for the test set. (3.1. Cross-Validation: Evaluating Estimator Performance — *Scikit-Learn 1.0.2 Documentation*, n.d.).

Since our dataset has a quite large number of records (10,000), we choose 80% and 20% portions of the HDFdataset for training and test data respectively. Then, we define `X_train` and `y_train` for the training data and `X_test` and `y_test` for the test data representing the feature and responses for each split.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

```

print(X_train)

[[ 299.8  310.6 1707.    32.5  124. ]
 [ 298.2  308.7 1605.    29.4   47. ]
 [ 300.5  309.8 1550.    37.4  148. ]
 ...
 [ 301.3  310.1 1455.    44.1  188. ]
 [ 298.3  309.1 1421.    47.4   33. ]
 [ 299.7  309.2 1346.    57.4  138. ]]

print(y_train)
[0 0 0 ... 0 0 0]

```

5.3.3 Training Decision Tree Classification Model on the Training data

Considering that the response variables (y_{test} and y_{train}) have the qualitative data type (failure “1” or normal “0”) we need to use ML classification techniques to fit a prediction model for the HDF failure problem. Therefore, we fit the decision tree and random forest classification techniques to HDFdataset by going through the following process:

- The decision tree and random forest decision tree classification methods are fitted to X_{train} and y_{train} .
- The fitted model (or prediction model) named as ‘classifier’ in our programming is applied to the X_{test} .
- We named the results of the prediction model on X_{test} as “ y_{pred} ”
- Then, we compared y_{pred} with y_{test} to validate the model and find its accuracy (this part is discussed in the next chapter)

Accordingly, we use the Scikit-Learn library package in Python to fit a prediction model on the X_{train} and y_{train} . To check the impurity of each split which is necessary for the decision tree and random forest techniques, Scikit-Learn library suggests the Gini index and

entropy methods discussed in section 2.5.3 we choose the entropy algorithm (information gain) to classify data into the most commonly occurring class (Pedregosa et al., 2011).

Later in the result chapter of the dissertation, we will compare the accuracy of decision tree and random forest techniques, apply the results of the prediction model “y_pred” in the simulation model, and consider CBM concepts in the simulation model to optimize the JSS problem in AAL stamping shop.

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit (X_train,y_train)
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```


6. Analysis of the Results Obtained

This chapter focuses on the results and compares the efficiency of digital twin technology with the traditional simulation-optimization technique for optimization of JSS through the following steps:

- Comparing the accuracy of the ML techniques (decision tree and random forest) applied in the previous chapter for the machine failure prediction to choose one of them in our model.
- Implementing the process of fitting distribution for machine failures HDF problem applied for the traditional simulation-optimization technique.
- Comparing the results of the digital twin technology with the traditional simulation-optimization technique for the JSS optimization in AAL's stamping shop.
- Completing the simulation model by adding the CBM scenario and applying the ML prediction model results for the HDF problem.
-

6.1 Results of Distribution Fitting Process on Sample Test

Considering the descriptive statistics and the histogram in section 4.4, in this section we apply the Arena Input Analyzer tool to the exponential, gamma, and Weibull probability distributions and compare their Square Error, Chi-Square test, and Kolmogorov-Smirnov tests.

As it is shown in table 13, the exponential distribution function ($30 + EXPO(395)$) with the minimum square error and chi-square test results is the best option to resemble the interval between HDF machine breakdowns in our model.

Table 13: Results of Distribution Fitting Process

Distribution Function	30 + EXPO (395)	30 + GAMM (1.23e+03, 0.32)	30 + WEIB (180, 0.488)
Square Error	0.000497	0.158754	0.00327
Chi Square Test Corresponding p-value < 0.005	0.000193	51.9	0.184
Kolmogorov-Smirnov Test p- value < 0.01	0.259	4.11	0.244

To compare the results of the fitted distribution function (30 + *EXPO*(395)) with the results of the sample test (*y_test*):

- First, we obtain the probability of density function based on the fitted distribution function:

$$30 + EXPO(395) \rightarrow \lambda = 395$$

$$f(x) = 30 + \lambda e^{-\lambda x} \rightarrow f(x) = 30 + 395 e^{-395 x}$$

- Second, we calculate the mean of the probability of density

$$\theta = 30 + \frac{1}{\lambda} \rightarrow = 30 + \frac{1}{395} = 30.0025 \text{ min}$$

The θ is the mean of the interval between failures. Therefore, Since the sample test represents the machine working for 60,000 minutes ($T = 2,000^{records} \times 30^{min}$), we expect to have the HDF machine failures on average (\bar{N}) as follow:

$$\bar{N} = T \div \theta = 60,000 \div 30,0025 \cong 2,000 \text{ times}$$

It means to increase the reliability of the system; we need to have on average 2,000 times preventive maintenance during the sample duration

6.2 Confusion Matrix and Accuracy Score of ML techniques for Prediction Model

In this section, we compare the accuracy of the two ML techniques (decision tree and random forest classification) implemented in the previous chapter for predicting the machine's HDF problem. The results of the decision tree classifier (y_pred_dte) are compared with the results of the random forest classifier (y_pred_rfc) by analyzing their confusion matrixes and accuracy scores.

The confusion matrix presents the following information as it is shown in table 14 (*Confusion Matrix — Scikit-Learn 1.0.2 Documentation*, n.d.):

- True negative (TN) = C (0,0)
- False positive (FP) = C (0,1)
- False negative (FN) = C (1,0)
- True positive (TP) = C (1,1)

Table 14: Confusion Matrix

Confusion Matrix (CM)	Prediction		
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

And the accuracy score is obtained by (3.3. Metrics and Scoring: Quantifying the Quality of Predictions — Scikit-Learn 1.0.2 Documentation, n.d.):

$$\text{Accuracy Score} = \frac{\text{TP} + \text{TN}}{\text{Total (TP + FN + FP + TN)}}$$

Considering table15, the confusion matrix and the accuracy score of the decision tree classifier (y_pred_dtc) shows:

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
cm = confusion_matrix (y_test,y_pred_dtc)
print(cm)
print ('\n')
ac= accuracy_score(y_test, y_pred_dtc)
print ('Accuracy=', ac)
print ('\n')
print(classification_report(y_test,y_pred_dtc))
```

Table 15: Confusion Matrix of Decision Tree Classifier

Confusion Matrix	Prediction		
	Negative	Positive	
Actual	Negative	1976	2
	Positive	0	22

- 1976 times correct prediction for machine none-failure (or normal) conditions (C (0,0) = True negative (TP)).
- Two times incorrect prediction for machine non-failure conditions while the actual data shows the failure condition False positive (C (0,1) = False positive (FP))
- No incorrect prediction while the actual data shows machine normal conditions (C (1,0) = False negative (FN))
- 22 times correct prediction for machine failure conditions (C (1,1) = True Positive (TP))
- And the accuracy score of the decision tree classification technique is %99.9 which is quite significant.

On the other hand, the confusion matrix (see table 16) and the accuracy score for the random forest classifier technique (y_pred_rfc) shows:

```
y_pred_rfc = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
cm = confusion_matrix (y_test,y_pred_rfc)
print(cm)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred_rfc)
```

Table 16: Confusion Matrix of Random Forest Classifier

Confusion Matrix	Prediction		
		Negative	Positive
Actual	Negative	1978	0
	Positive	4	18

- 1978 times correct prediction for machine none-failure (or normal) conditions (C (0,0) = True negative (TP))
- No incorrect prediction for machine failure conditions (C (0,1) = False positive (FP))
- 4 times incorrect prediction for machine normal conditions (C (1,0) = False negative (FN))
- 18 times correct prediction for machine failure conditions True positive (C (1,1) = True Positive (TP))
- and the accuracy score of the random forest classification technique is %99.8.

6.3 Applying the Prediction Model to Real-Time Data

To implement the prediction model on the real-time data, it is assumed that the real-time data is transmitted through CPS technology from the sensors every 30 minutes. Thereafter, we apply the prediction model trained based on the decision tree classification method to the transmitted real-time data (which is `X_test` in our model) in the Python platform. The results of the prediction named `y_pred_dtc` in our coding are exported to a CSV file as follows indicating the possibility of the HDF problem by “1”, or a by “0” if not:

```
y_pred_dtc = classifier.predict(X_test)
df = pd.DataFrame(y_pred_rfc.reshape(len(y_pred), 1))
df.to_csv(r'C:\Users\vahid\OneDrive - Concordia University - Canada\HP\Concordia\M.A.SC\Case Study\output.csv', header=False)
```

To dip into what the decision tree classifier is done in our model, we use Graphviz imported from the sci-kit-learn library into the Python package (*1.10. Decision Trees — Scikit-Learn 1.0.2 Documentation*, n.d.). Referring to the explanations of section 2.5.3, we can see from figure 29 that the decision tree classification technique has built an optimization model to classify any new set of real-time data into a tree binary structure with 43 nodes.

```
import graphviz
from sklearn.tree import export_graphviz
dot_data = export_graphviz(classifier, out_file=None, filled=True,
rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

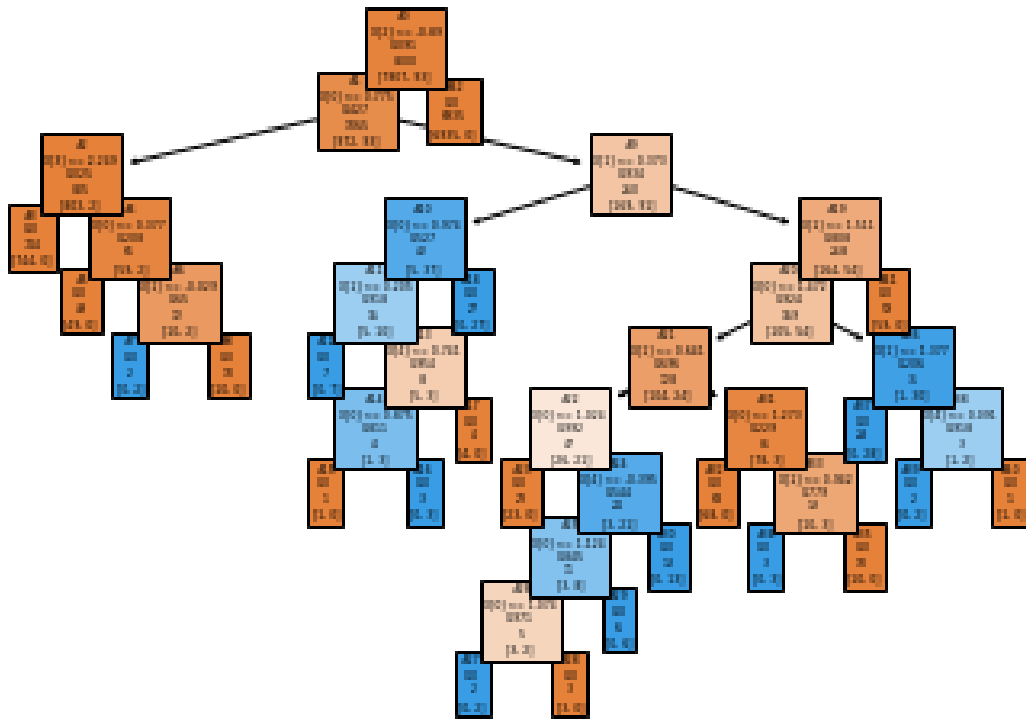


Figure 29: The Decision Tree Classification Model

node=0 is a split node: go to node 1 if $X[:, 2] \leq 1379.5$ else to node 42.

node=1 is a split node: go to node 2 if $X[:, 0] \leq 301.5500030517578$ else to node 9.

node=2 is a split node: go to node 3 if $X[:, 3] \leq 62.35000038146973$ else to node 4.

node=3 is a leaf node.

node=4 is a split node: go to node 5 if $X[:, 0] \leq 300.75$ else to node 6.

node=5 is a leaf node.

node=6 is a split node: go to node 7 if $X[:, 1] \leq 309.9499969482422$ else to node 8.

node=7 is a leaf node.

node=8 is a leaf node.

node=9 is a split node: go to node 10 if $X[:, 1] \leq 310.5500030517578$ else to node 19.

node=10 is a split node: go to node 11 if $X[:, 0] \leq 301.9499969482422$ else to node 18.

node=11 is a split node: go to node
12 if $X[:, 1] \leq 310.3000030517578$ else to node 13.
node=12 is a leaf node.
node=13 is a split node: go
to node 14 if $X[:, 4] \leq 154.5$ else to node 17.
node=14 is a split n
ode: go to node 15 if $X[:, 0] \leq 301.75$ else to node 16.
node=15 is a
leaf node.
node=16 is a
leaf node.
node=17 is a leaf no
de.
node=18 is a leaf node.
node=19 is a split node: go to node 20 if
 $X[:, 1] \leq 312.25$ else to node 41.
node=20 is a split node: go to node
21 if $X[:, 0] \leq 302.9499969482422$ else to node 36.
node=21 is a split node: go
to node 22 if $X[:, 1] \leq 310.9499969482422$ else to node 31.
node=22 is a split n
ode: go to node 23 if $X[:, 0] \leq 302.0500030517578$ else to node
24.
node=23 is a
leaf node.
node=24 is a
split node: go to node 25 if $X[:, 4] \leq 82.0$ else to node 30.
node=
25 is a split node: go to node 26 if $X[:, 0] \leq 302.25$ else to n
ode 29.
node=26 is a split node: go to node 27 if $X[:, 0] \leq 302.
15000915527344$ else to node 28.
node=27 is a leaf node.
node=28 is a leaf node.
node=29 is a leaf node.
node=
30 is a leaf node.
node=31 is a split n
ode: go to node 32 if $X[:, 0] \leq 302.5500030517578$ else to node
33.
node=32 is a
leaf node.

```

node=33 is a
split node: go to node 34 if X[:, 1] <= 311.40000915527344 else
to node 35.
node=
34 is a leaf node.
node=
35 is a leaf node.
node=36 is a split node: go
to node 37 if X[:, 1] <= 312.0500030517578 else to node 38.
node=37 is a leaf no
de.
node=38 is a split n
ode: go to node 39 if X[:, 4] <= 113.0 else to node 40.
node=39 is a
leaf node.
node=40 is a
leaf node.
node=41 is a leaf node.

```

The script above explains how the decision tree classification technique classifies the new set of real-time data captured from sensors to optimize the prediction for the HDF problem in our model. As it is shown in figure 29, from the top to the down of the decision tree, each node classifies the data by a threshold and is split into another two nodes, and the sequence is continued to the end nodes called leaves to the leaves indicating the possibility of the HDF problem by “1” or “0” if not.

For example, as it is described in the script, node 1 is split into nodes 2 and 3 by the condition “if X [: 2] <= 1379.5” meaning if X [: 2] which is the data captured from the rotational speed [rpm] sensors is less than 1379.5. Then the same process is done for nodes 2 and 3 assigned to “X [: 0]” which is the air temperature [K] data and “X[: 3]” which is the torque [Nm] by assigned thresholds.

6.4 Implementing CBM in the Simulation model

To apply the prediction results and implement the CBM in the simulation model, as it is shown in figure 30, we go through the following steps in the Arena simulation software:

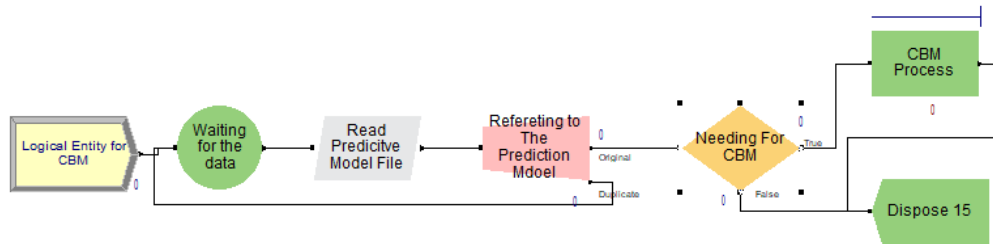


Figure 30: CBM in Simulation Model

- Creating a logical entity for the CBM process in the simulation model to initiate the process of CBM.
- Waiting for 30 minutes is the time assumed in our model for capturing the real-time data from sensors.
- Defining the CSV file as the input file in the Arena software (see figure 31). The CSV file is the result of the prediction exported from the Python (see section 6.3)
- Reading the result of the prediction model (`y_pred_dtc`) from the CSV file (see figure 32)
- Checking the results of the prediction to see if we need to implement CBM. If the result shows “1”, it means there is a possibility of an HDF problem for the machine.

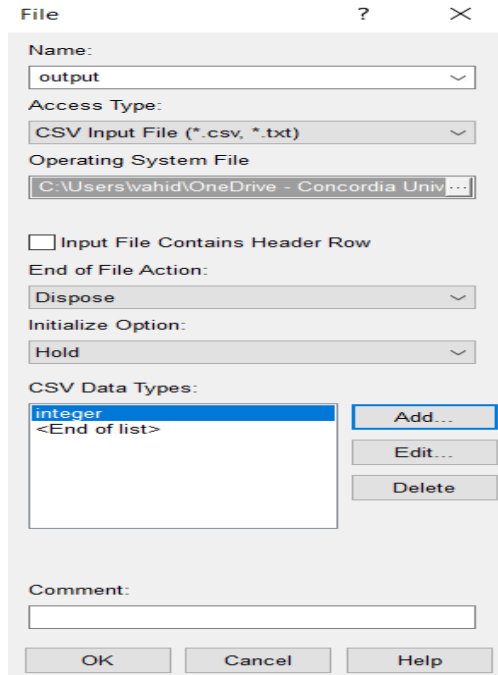


Figure 31: Defining the Input File in the Arena Software

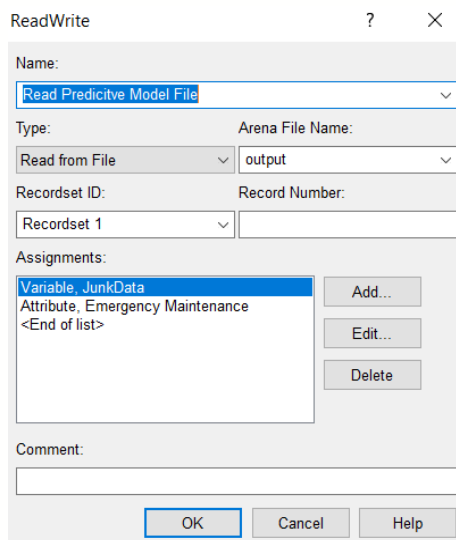


Figure 32: Read and Write Module in Arena Software

In short, to make use of the results of the prediction model built by the decision tree classification technique on real-time data captured from the sensors, we define the CSV file exported from Python as an input file in the Arena simulation software by the Input File module in Arena (see figure 31). Then to run the CBM scantron in the simulation model (see figure 30), we use the Read/Write module in Arena to read the CSV file every 30 minutes (see figure 32).

In implementing the CBM, if the system captures real-time data at a higher rate, for example, every 30 seconds, it will make our model more accurate by applying the prediction model and CBM more frequently to real-time data. We think that this condition even makes the role of applying data science techniques such as ML used in digital twin more important in terms of speed in comparison with statistical analysis techniques which is used in simulation optimization technique.

Additionally, we believe that in case of capturing the real-time every 30 seconds, we need to create a system dynamics model instead of the discrete event simulation model. In this case, we can also define a confidence interval for the threshold values obtained by the decision tree classification technique to implement the CBM process at confidence interval rates.

7. Conclusions, Limitations, and Future Work

7.1 Conclusions

We conclude this thesis by comparing the results of the digital twin technology with the traditional simulation-optimization technique for optimization of the JSS problem and reviewing the main contributions as follows:

- Comparing the results of the decision tree classification method (y_{pred}) and random forest classification model trained on the x_{test} with the actual data (y_{test}) of the sample test.
- Comparing the results of the distribution fitting method proceeded by statistical analysis of the historical data with the actual data (y_{test}) of the sample test.
- Comparing the accuracy of the digital twin technology with the traditional simulation-optimization technique for the JSS problem.
- Restating the main contributions of the dissertation

Considering the results of the confusion matrix and accuracy score calculated for decision tree and random forest classification techniques in section 6.2, we select the decision tree classifier (y_{pred_dtc}) with a higher accuracy score which is %99.9 compared to %99.8 for the random forest method to use it in our prediction model. Although the random forest is considered a more advanced technique than the decision tree classification technique as it is described in section 2.5.4, it does not always provide a better result especially when the accuracy score of the decision tree classification technique is high enough.

As it is explained in section 6.2, we know that the y_{test} contains 22 actual machine failures for the HDF problem during the 60,000 working minutes of the machine in the sample test. The distribution fitting technique which is applied to the traditional simulation-optimization technique tries to imitate the frequency and pattern of a machine's failures in the simulation model by statistical analysis. The distribution fitting method applied in the traditional simulation-optimization technique (see section 6.1) shows that if we want the most reliable

condition for the HDF problems of the stamping machine, it is needed to implement PM instructions every 30 minutes. This means performing 2,000 PMs for the HDF problem during the sample test (60,000 minutes).

The 2,000 PMs are much more than the 22 times of actual failures in our sample test which is not an efficient way to be implemented. Therefore, we need to consider the trade-off for implementing PMs as we discussed in section 2.4.2 with the following strategies:

1. Maximizing the RUL of the parts by accepting the risk of machine breakdown
2. Maximizing the RUL of the machine by accepting the cost of replacing parts earlier

If we choose the first strategy by reducing the number of PMs, we will accept the risk of facing unplanned machine failures for HDF problems during the stamping process. The JSS based on this strategy:

- Firstly, increasing the total tardiness costs, while as is defined in section 3.2 one of the main objectives of AAL company is to minimize the total tardiness costs.
- Secondly, it is against the sustainable manufacturing approach defined as the motivation of this dissertation (see section 1.1). Because accepting the risks of unexpected machine failures during the manufacturing process leads to a waste of resources, energy, and time.

If we choose the second strategy by increasing PMs and replacing machine parts earlier, we probably have fewer unplanned machine failures for the HDF problem, but the JSS based on this strategy:

- Firstly, it increases the costs of maintenance which is not reasonable for the AAL company with several stamping machines.
- Secondly, it is against the lean manufacturing approach defined as the motivation of this dissertation (see section 1.1). Replacing parts sooner than is

needed, is not an efficient way of using resources and increases the costs of purchasing and inventory of spare parts.

On the other side, as is described in section 2.5 the digital twin technology can apply ML techniques to predict machine failures, and as is shown in section 6.2 the accuracy of the decision tree classification model is %99.9. This accuracy is obtained by evaluating the correlations not only among features related to machine functions such as rotational speed or torque but also features related to the working environment like the air temperature. Referring to section 5.2 and figure 26, we can see from the box plot that there is a high correlation between the air temperature and the HDF problem.

Therefore, the ML learning techniques used in digital twin technology is a developed approach with higher accuracy compared to the distribution fitting method in the traditional simulation-optimization technique for HDF problem prediction. Additionally, the possibility of getting access to real-time data captured from sensors and implementing CBM in digital twin technology optimizes the JSS problem in our case by reducing unplanned machine failures.

In summary, to restate the contribution of his dissertation:

- We build a discrete-event simulation model with Arena software for the JSS of AAL's stamping shop (see chapter 3).
- We describe the application and advantages of the digital twin technology for optimization of the JSS problem (See section 2.2)
- We develop a distribution fitting method for the HDF problem of the stamping machine used in the traditional simulation optimization technique through Minitab software and Arena Input Analyzer tool for AAL's JSS problem (see chapter 4 and section 6.1).
- We develop the decision tree and random forest method as two ML classification techniques in a Python Scikit-Learn package (see section 5.3.3) to build a

prediction model for the HDF problem of a stamping machine applied to digital twin technology.

- We show that the ML approach is more efficient with higher accuracy for the prediction of the HDF problem in the stamping machine compared to the distribution fitting approach in the traditional simulation-optimization technique (see sections 6.1 and 6.2).
- We create the CBM scenario in the simulation model by reading the results of the prediction model (decision tree classification technique) exported from Python that was applied to the real-time data for the HDF problem of the machine (see sections 6.3 and 6.4).
- We explain the concept of the digital twin in making use of IIoT, CPS, cloud computing to receive real-time data and do timely analysis and respond (see sections 2.3 and 2.4)
- We compare the results of the traditional simulation-optimization technique with the digital twin technology for the AAL stamping shop, and we conclude that the digital twin is a more sophisticated approach by employing real-time data, ML techniques, and CBM to reduce unplanned machine breakdowns for optimization of the JSS problem (see section 7.1).
- We present that the digital twin technology can conduct the sustainable and lean manufacturing system concepts and can be one step toward the implementation of smart units, smart machines, and smart manufacturing systems.

7.2 Limitation of the Research

In this dissertation we had the following limitations:

- Not being able to implement our case study on more advanced simulation software like Anylogic which is a commercial simulation-optimization package, because its provider does not offer a free full version for the research work and the Concordia university does not have it installed in the computer labs. The

Anylogic software can ease the implementation of the digital twin technology by offering cloud-based simulation models. The cloud-based simulation can apply machine learning platforms such as Python or Java, and synchronize the real-time data captured from sensors faster by powerful cloud computing resources(*Cloud Computing Simulation Tool – AnyLogic Simulation Software*, n.d.). Therefore, instead of applying machine learning in a platform such as Python, exporting the results, and then importing and reading the results in the simulation model, the same process that is done in this thesis with Arena software, Anylogic cloud-based models can integrate the process.

- We assume that the data is captured and transferred from the machine's sensors every 30 minutes. Due to the lack of time, we were stopped to do deeper research to evaluate a simulation model in the condition of receiving the data with a higher frequency like every 30 seconds which is more efficient for CBM.

7.3 Future work

As for recommendations and future work we suggest:

- Implementing the same concepts defined in this thesis with Anylogic software to investigate how a cloud-based simulation model is integrated with ML platforms and real-time data in this software to optimize the JSS problem.
- Supposing the real-time data is transferred from the machine's sensor every 30 seconds instead of the 30 minutes that it is assumed in this dissertation. We believe, we need to develop system dynamics instead of a discrete-event simulation model with this high rate of data flow. Working on this subject and building a system dynamics simulation model for optimization of the JSS problem of the case study.
- In this dissertation, we assume that we have access to the historical labeled data of machine failures. Considering we do not have access to labeled data, which

is common in industrial cases, working on unsupervised ML techniques to build a machine failure prediction model, and comparing its results with the results of this dissertation.

- Working on transport and logistic optimization problem of the AAL company. As it is described in the case study, the AAL company has three factories located in three different cities. Supposing the AAL company wants to relocate sales agencies based on the demands across the U.S to minimize the cost of logistics. Working on an agent-based simulation model to minimize the transport and logistic costs.

References

1.10. *Decision Trees — scikit-learn 1.0.2 documentation*. (n.d.). Retrieved March 2, 2022, from <https://scikit-learn.org/stable/modules/tree.html?highlight=graphviz>

3.1. *Cross-validation: evaluating estimator performance — scikit-learn 1.0.2 documentation*. (n.d.). Retrieved January 9, 2022, from https://scikit-learn.org/stable/modules/cross_validation.html

3.3. *Metrics and scoring: quantifying the quality of predictions — scikit-learn 1.0.2 documentation*. (n.d.). Retrieved March 2, 2022, from https://scikit-learn.org/stable/modules/model_evaluation.html

2016 *Global Industry 4.0 Survey-Industry key findings*. (n.d.). Retrieved November 19, 2021, from www.pwc.com/industry40

Adikaram, K. K. L. B., Hussein, M. A., Effenberger, M., & Becker, T. (2015). Data transformation technique to improve the outlier detection power of grubbs' test for data expected to follow linear relation. *Journal of Applied Mathematics*, 2015. <https://doi.org/10.1155/2015/708948>

Allah Bukhsh, Z., Saeed, A., Stipanovic, I., & Doree, A. G. (2019). Predictive maintenance using tree-based classification techniques: A case of railway switches. *Transportation Research Part C: Emerging Technologies*, 101, 35–54. <https://doi.org/10.1016/J.TRC.2019.02.001>

Arena (software) - Wikipedia. (n.d.). Retrieved November 30, 2021, from [https://en.wikipedia.org/wiki/Arena_\(software\)](https://en.wikipedia.org/wiki/Arena_(software))

Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert*

Systems with Applications, 173, 114598.
<https://doi.org/10.1016/J.ESWA.2021.114598>

Bazaz, S. M., Lohtander, M., & Varis, J. (2019). 5-dimensional definition for a manufacturing digital twin. *Procedia Manufacturing*, 38, 1705–1712.
<https://doi.org/10.1016/J.PROMFG.2020.01.107>

Box plot review (article) | Khan Academy. (n.d.). Retrieved January 24, 2022, from <https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/box-whisker-plots/a/box-plot-review>

Caggiano, A., Bruno, G., & Teti, R. (2015). Integrating Optimisation and Simulation to Solve Manufacturing Scheduling Problems. *Procedia CIRP*, 28, 131–136.
<https://doi.org/10.1016/J.PROCIR.2015.04.022>

Cakir, M., Guvenc, M. A., & Mistikoglu, S. (2021). The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system. *Computers and Industrial Engineering*, 151.
<https://doi.org/10.1016/J.CIE.2020.106948>

Case Study: American Automobiles Limited: Production Planning | Ivey Publishing. (n.d.). Retrieved December 24, 2021, from <https://www.iveypublishing.ca/s/product/american-automobiles-limited-production-planning/01t5c00000CwqywAAB>

Cavalcante, I. M., Frazzon, E. M., Forcellini, F. A., & Ivanov, D. (2019). A supervised machine learning approach to data-driven simulation of resilient supplier selection in digital manufacturing. *International Journal of Information Management*, 49, 86–97.
<https://doi.org/10.1016/J.IJINFOMGT.2019.03.004>

Cloud Computing Simulation Tool – AnyLogic Simulation Software. (n.d.). Retrieved February 19, 2022, from <https://www.anylogic.com/features/cloud/>

- Confusion matrix* — *scikit-learn 1.0.2 documentation*. (n.d.). Retrieved March 2, 2022, from https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html?highlight=confusion%20matrix
- de Paula Ferreira, W., Armellini, F., & de Santa-Eulalia, L. A. (2020). Simulation in industry 4.0: A state-of-the-art review. *Computers & Industrial Engineering*, *149*, 106868. <https://doi.org/10.1016/J.CIE.2020.106868>
- Ding, K., Chan, F. T. S., Zhang, X., Zhou, G., & Zhang, F. (2019). Defining a Digital Twin-based Cyber-Physical Production System for autonomous manufacturing in smart shop floors. *International Journal of Production Research*, *57*(20), 6315–6334. <https://doi.org/10.1080/00207543.2019.1566661>
- Fang, Y., Peng, C., Lou, P., Zhou, Z., Hu, J., & Yan, J. (2019). Digital-Twin-Based Job Shop Scheduling Toward Smart Manufacturing. *IEEE Transactions on Industrial Informatics*, *15*(12), 6425–6435. <https://doi.org/10.1109/TII.2019.2938572>
- Fitch, E. C. (n.d.). *Proactive maintenance for mechanical systems : an activity conducted to detect and correct root cause aberrations of failure*. 339.
- Gartners Top 10 Technology Trends 2017*. (n.d.). Retrieved December 7, 2021, from <https://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017>
- Geyik, F., & Cedimoglu, I. H. (2004). The strategies and parameters of tabu search for job-shop scheduling. *Journal of Intelligent Manufacturing* *2004 15:4*, *15*(4), 439–448. <https://doi.org/10.1023/B:JIMS.0000034106.86434.46>
- He, B., & Bai, K. J. (2021). Digital twin-based sustainable intelligent manufacturing: a review. *Advances in Manufacturing*, *9*(1), 1–21. <https://doi.org/10.1007/S40436-020-00302-5/FIGURES/4>

How Many Simulation Trials Should I Run? Factors that Impact OptQuest Search Performance – AnyLogic Simulation Software. (n.d.). Retrieved November 30, 2021, from <https://www.anylogic.com/blog/how-many-simulation-trials-should-i-run-optquest/>

Interpret the key results for Display Descriptive Statistics - Minitab. (n.d.). Retrieved March 2, 2022, from <https://support.minitab.com/en-us/minitab/20/help-and-how-to/statistics/basic-statistics/how-to/display-descriptive-statistics/interpret-the-results/key-results/>

Ivey Publishing: About | LinkedIn. (n.d.). Retrieved December 25, 2021, from <https://www.linkedin.com/company/ivey-publishing/about/>

James, G. (Gareth M., Witten, D., Hastie, T., & Tibshirani, R. (n.d.). *An introduction to statistical learning : with applications in R.*

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning.* <https://doi.org/10.1007/978-1-0716-1418-1>

Jasiulewicz-Kaczmarek, M., ... S. L.-M. and, & 2020, undefined. (n.d.). Maintenance 4.0 technologies–new opportunities for sustainability driven maintenance. *Yadda.Icm.Edu.Pl.* Retrieved December 14, 2021, from <https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-bb1a5e3b-e61c-4dee-9409-3dde410ff421>

Jasiulewicz-Kaczmarek, M., Legutko, S., & Kluk, P. (2020). Maintenance 4.0 technologies – new opportunities for sustainability driven maintenance. *Management and Production Engineering Review, Vol. 11, No. 2(2), 74–87.* <https://doi.org/10.24425/MPER.2020.133730>

- Jian, N., & Henderson, S. G. (2016). An introduction to simulation optimization. *Proceedings - Winter Simulation Conference, 2016-February*, 1780–1794. <https://doi.org/10.1109/WSC.2015.7408295>
- Kaparthi, S., & Bumblauskas, D. (2020). Designing predictive maintenance systems using decision tree-based machine learning techniques. *International Journal of Quality and Reliability Management*, 37(4), 659–686. <https://doi.org/10.1108/IJQRM-04-2019-0131/FULL/PDF>
- Kelton, W. David., Sadowski, R. P., & Zupick, N. B. (n.d.). *Simulation with Arena*.
- Kučera, M., Kopčanová, S., & Sejkorová, M. (2020). Lubricant analysis as the most useful tool in the proactive maintenance philosophies of machinery and its components. *Management Systems in Production Engineering, nr 3 (28)(3)*, 196–201. <https://doi.org/10.2478/MSPE-2020-0029>
- Law, A. M. (2015). *Simulation Modeling and Analysis, FIFTH EDITION*. www.averill-law.com
- Lin, J. T., & Chen, C. M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory*, 51, 100–114. <https://doi.org/10.1016/J.SIMPAT.2014.10.008>
- Liu, M., Fang, S., Dong, H., & Xu, C. (2021). Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 58, 346–361. <https://doi.org/10.1016/J.JMSY.2020.06.017>
- Liu, R., Xie, X., Yu, K., & Hu, Q. (2018). A survey on simulation optimization for the manufacturing system operation. *International Journal of Modelling and Simulation*, 38(2), 116–127. <https://doi.org/10.1080/02286203.2017.1401418>

Marzec, M., Morkisz, P., Wojdyła, J., & Uhl, T. (2016). Intelligent Predictive Maintenance System. *Lecture Notes in Networks and Systems*, 15, 794–804.

https://doi.org/10.1007/978-3-319-56994-9_55

Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., & Ueda, K. (2016). Cyber-physical systems in manufacturing. *CIRP Annals*, 65(2), 621–641.

<https://doi.org/10.1016/J.CIRP.2016.06.005>

Operations research - Wikipedia. (n.d.). Retrieved December 3, 2021, from

https://en.wikipedia.org/wiki/Operations_research

Overview for Descriptive Statistics - Minitab Express. (n.d.). Retrieved March 2, 2022, from <https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/basic-statistics/summary-statistics/descriptive-statistics/before-you-start/overview/>

P, K., J, V., & 2016 International Conference on Industrial Engineering and Engineering Management, I. 2016. (2016). Simulation and optimisation based approach for job shop scheduling problems. *IEEE International Conference on Industrial Engineering and Engineering Management, 2016-December*, 360–364.

<https://doi.org/10.1109/IEEM.2016.7797897>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85), 2825–2830.

<http://jmlr.org/papers/v12/pedregosa11a.html>

Predictive Maintenance | Kaggle. (n.d.). Retrieved January 25, 2022, from

<https://www.kaggle.com/tolgadincer/predictive-maintenance/metadata>

- Random forest - Wikipedia*. (n.d.). Retrieved December 17, 2021, from https://en.wikipedia.org/wiki/Random_forest
- Rho, S., Vasilakos, A. v., & Chen, W. (2016). Cyber physical systems technologies and applications. *Future Generation Computer Systems*, *56*, 436–437. <https://doi.org/10.1016/J.FUTURE.2015.10.019>
- Roosefert Mohan, T., Preetha Roselyn, J., Annie Uthra, R., Devaraj, D., & Umachandran, K. (2021a). Intelligent machine learning based total productive maintenance approach for achieving zero downtime in industrial machinery. *Computers & Industrial Engineering*, *157*, 107267. <https://doi.org/10.1016/J.CIE.2021.107267>
- Roosefert Mohan, T., Preetha Roselyn, J., Annie Uthra, R., Devaraj, D., & Umachandran, K. (2021b). Intelligent machine learning based total productive maintenance approach for achieving zero downtime in industrial machinery. *Computers & Industrial Engineering*, *157*. <https://doi.org/10.1016/j.cie.2021.107267>
- Roychowdhury, S., Allen, T. T., & Allen, N. B. (2017). A genetic algorithm with an earliest due date encoding for scheduling automotive stamping operations. *Computers & Industrial Engineering*, *105*, 201–209. <https://doi.org/10.1016/J.CIE.2017.01.007>
- Schroeder, G. N., Steinmetz, C., Rodrigues, R. N., Henriques, R. V. B., Rettberg, A., & Pereira, C. E. (2021). A Methodology for Digital Twin Modeling and Deployment for Industry 4.0. *Proceedings of the IEEE*, *109*(4), 556–567. <https://doi.org/10.1109/JPROC.2020.3032444>
- Scott, D. W. (2010). Scott's rule. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 497–502. <https://doi.org/10.1002/WICS.103>
- Shao, G., & Helu, M. (2020). Framework for a digital twin in manufacturing: Scope and requirements. *Manufacturing Letters*, *24*, 105–107. <https://doi.org/10.1016/J.MFGLET.2020.04.004>

Simulink - Simulation and Model-Based Design - MATLAB & Simulink. (n.d.). Retrieved November 30, 2021, from <https://www.mathworks.com/products/simulink.html>

Söderberg, R., Wärmefjord, K., Carlson, J. S., & Lindkvist, L. (2017). Toward a Digital Twin for real-time geometry assurance in individualized production. *CIRP Annals - Manufacturing Technology*, *66*(1), 137–140. <https://doi.org/10.1016/J.CIRP.2017.04.038>

Torcianti, A., & Matzka, S. (2021). *Explainable Artificial Intelligence for Predictive Maintenance Applications using a Local Surrogate Model*. 86–88. <https://doi.org/10.1109/AI4I51902.2021.00029>

Traini, E., Bruno, G., D’Antonio, G., & Lombardi, F. (2019). Machine Learning Framework for Predictive Maintenance in Milling. *IFAC-PapersOnLine*, *52*(13), 177–182. <https://doi.org/10.1016/J.IFACOL.2019.11.172>

UCI Machine Learning Repository: AI4I 2020 Predictive Maintenance Dataset Data Set. (n.d.). Retrieved March 2, 2022, from <https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset#>

VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. *O’Reilly*, 541. <https://www.oreilly.com/library/view/python-data-science/9781491912126/>

Xia, T., Shi, G., Si, G., Du, S., & Xi, L. (2021). Energy-oriented joint optimization of machine maintenance and tool replacement in sustainable manufacturing. *Journal of Manufacturing Systems*, *59*, 261–271. <https://doi.org/10.1016/J.JMSY.2021.01.015>

Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, *30*(4), 1809–1830. <https://doi.org/10.1007/S10845-017-1350-2/TABLES/3>

Zhang, M., Tao, F., & Nee, A. Y. C. (2021). Digital Twin Enhanced Dynamic Job-Shop Scheduling. *Journal of Manufacturing Systems*, 58, 146–156.

<https://doi.org/10.1016/J.JMSY.2020.04.008>

Zhang, Z., Guan, Z., Gong, Y., Luo, D., & Yue, L. (2020). Improved multi-fidelity simulation-based optimisation: application in a digital twin shop floor.

<https://doi-org.lib-ezproxy.concordia.ca/10.1080/00207543.2020.1849846>.

<https://doi.org/10.1080/00207543.2020.1849846>