**Automatic Scheduling for AGV-based Digital Manufacturing Platforms (DMPs) using Digital Twins**

Submitted by

Zhou Yuanyuan

Sichuan University

College of Electronics and Information Engineering

Supervisor:

ASSOC. PROF. VEERAVALLI, BHARADWAJ

Examiner：

DR ARTHUR TAY

The final year project work was carried out under the 3+1+1 Educational Framework at The National University of Singapore (Suzhou) Research Institute

April 2023

# ABSTRACT

As technology redefines the manufacturing industry, it has become essential to use digital manufacturing platforms (DMP) to maximize operational efficiency and accelerate Industry 4.0. The introduction of AGV has further improved the efficiency of the intelligent logistics system. In practical production, scheduling is an essential part of optimizing the whole chain.

For product processing production, job shop scheduling and flow shop scheduling are mainly used, and the system performance is evaluated through indicators such as blockage rate and machine occupancy rate; the concept of machine learning is introduced on the basis of scheduling, and the digital twin model is used to realize the prediction of the scheduling system and the allocation of resources.

In this essay, the objective is focused on the application of digital twin (DT) models in job shop scheduling problem (JSSP), hybrid machine learning (HML) strategies are implemented to optimize the whole system. Based on the Simpy simulation platform, the discrete event simulations are created. Then the machine failure prediction and DT models are established, and some evaluations are made to fulfill the functions.

**Index Terms:** industry 4.0; job shop scheduling; machine learning; digital twin; discrete event simulation

# ACKNOWLEDGMENTS

Immeasurable appreciation and deepest gratitude for the help and support to make this study possible.

I would firstly like to acknowledge and give my warmest thanks to my supervisor Prof. Bharadwaj Veeravalli who made this work possible. His guidance and advice carried me through all the stages of this project.

I would also like to give special thanks to my parents for giving continuous support and understanding in the whole of my life, thankyou for everything you have done for me.

I am grateful to all of those with whom I have had the pleasure to work during this time, thank you for all the extensive personal and professional guidance and taught me a great deal about both scientific research and life in general. I would also like to thank my teammate Shi Xuan for the meaningful comments and suggestions. And my friends in NUSRI, Zhou Nansi, Zheng Jiachen, Yang Xinyi, Jie Kewei and all my classmates and friends here, who have made my life colorful and meaningful.

# CONTENTS

[ABSTRACT II](#_Toc130996938)

[ACKNOWLEDGMENTS III](#_Toc130996939)

[LIST OF FIGURES V](#_Toc130996941)

[LIST OF TABLES VII](#_Toc130996942)

[LIST OF SYMBOLS AND ABBREVIATIONS VIII](#_Toc130996943)

[CHAPTER 1 INTRODUCTION 1](#_Toc130996944)

[1.1 Settings 1](#_Toc130996945)

[1.2 Objective and Motivation 1](#_Toc130996946)

[1.3 Contributions 2](#_Toc130996947)

[1.4 Thesis Outline 2](#_Toc130996948)

[CHAPTER 2 BACKGROUND 3](#_Toc130996949)

[2.1 AGV-based digital manufacturing platforms 3](#_Toc130996950)

[2.2 Simulation platform –Simpy 4](#_Toc130996951)

[2.3 Scheduling methods 4](#_Toc130996952)

[2.4 Digitalization 5](#_Toc130996953)

[CHAPTER 3 LITERATURE REVIEW 6](#_Toc130996954)

[CHAPTER 4 DESIGN AND PERFORMANCE 7](#_Toc130996955)

[4.1 Job shop scheduling simulation models 7](#_Toc130996956)

[4.1.1 description 7](#_Toc130996957)

[4.1.2 simulation scenarios 7](#_Toc130996958)

[4.1.3 basic models of production line 9](#_Toc130996959)

[4.1.4 simulations in simply environment 11](#_Toc130996960)

[4.1.5 job shop scheduling core strategy 12](#_Toc130996961)

[4.1.6 evaluation and optimization 15](#_Toc130996962)

[4.2 Machine Failure Models 18](#_Toc130996963)

[4.2.1 description 18](#_Toc130996964)

[4.2.2 dataset analysis and visualization 19](#_Toc130996965)

[4.2.3 machine learning based data modeling 22](#_Toc130996966)

[4.2.4 machine failure models in job shop scheduling 29](#_Toc130996967)

[4.3 Digital Twin Machine Learning-assisted Simulation Models 30](#_Toc130996968)

[4.3.1 description 30](#_Toc130996969)

[4.3.2 physical/numerical simulation models 31](#_Toc130996970)

[4.3.3 comparation between experimental data with physical model 32](#_Toc130996971)

[4.3.4 Design of hybrid digital twin 33](#_Toc130996972)

[4.3.5 Compilation of hybrid digital twin and Implementation 36](#_Toc130996973)

[CHAPTER 5 RESULTS 37](#_Toc130996974)

[CHAPTER 6 FUTURE WORK 40](#_Toc130996975)

[CHAPTER 7 CONCLUSION 40](#_Toc130996976)

[REFERENCES 41](#_Toc130996977)

# LIST OF FIGURES

[Figure 1 Job shop Scheduling Mode 4](#_Toc130996852)

[Figure 2 Flow shop Scheduling Modes 5](#_Toc130996853)

[Figure 3 Comparation between Job shop and Flow shop 5](#_Toc130996854)

[Figure 4 Schematic diagram of Scenario A 8](#_Toc130996855)

[Figure 5 Schematic diagram of Scenario B 8](#_Toc130996856)

[Figure 6 Schematic diagram of Scenario C 9](#_Toc130996857)

[Figure 7 Machine sequence 9](#_Toc130996858)

[Figure 8 Product number 9](#_Toc130996859)

[Figure 9 Product’s machine number 9](#_Toc130996860)

[Figure 10 The structure of two-dimensional array 10](#_Toc130996861)

[Figure 11 Product sequence 10](#_Toc130996862)

[Figure 12 Input sequence 11](#_Toc130996863)

[Figure 13 Machine’s processing time 11](#_Toc130996864)

[Figure 14 Resource allocation 12](#_Toc130996865)

[Figure 15 Interaction of functional functions in the simpy environment 12](#_Toc130996866)

[Figure 16 The Core Strategy in Job shop Scheduling 13](#_Toc130996867)

[Figure 17 Machine comparative operation mechanism 13](#_Toc130996868)

[Figure 18 Real simulation example 15](#_Toc130996869)

[Figure 19 Back-up machine number 16](#_Toc130996870)

[Figure 20 Comparison of machine utilization rates 17](#_Toc130996871)

[Figure 21 Occupation Rate of machines 17](#_Toc130996872)

[Figure 22 Blocked products 18](#_Toc130996873)

[Figure 23 Performance of job shop 18](#_Toc130996874)

[Figure 24 Pie chart for the consequences of machine failure 20](#_Toc130996875)

[Figure 25 Pairs Plot of gathered data 21](#_Toc130996876)

[Figure 26 Heatmap of gathered data 22](#_Toc130996877)

[Figure 27 TP vs. FP rate at different classification thresholds 23](#_Toc130996878)

[Figure 28 AUC (Area under the ROC Curve) 24](#_Toc130996879)

[Figure 29 The structure of Decision Tree 25](#_Toc130996880)

[Figure 30 The Structure of Random Forest RF1 26](#_Toc130996881)

[Figure 31 The Structure of Random Forest RF2 26](#_Toc130996882)

[Figure 32 The Structure of Random Forest RF3 27](#_Toc130996883)

[Figure 33 Comparison between different parameters in XGBoost 28](#_Toc130996884)

[Figure 34 The structure of XGBoost (clip) 28](#_Toc130996885)

[Figure 35 Machine failure conditions 29](#_Toc130996886)

[Figure 36 Update results of simulation 29](#_Toc130996887)

[Figure 37 Occupied quantity of the machines 30](#_Toc130996888)

[Figure 38 Physical model of machine occupation rate 32](#_Toc130996889)

[Figure 39 Comparison between initial data and physical model 33](#_Toc130996890)

[Figure 40 Compensate model 34](#_Toc130996891)

[Figure 41 Model Training Process-1 34](#_Toc130996892)

[Figure 42 Model Training Process-2 35](#_Toc130996893)

[Figure 43 Model Evaluation 35](#_Toc130996894)

[Figure 44 Model fitting 36](#_Toc130996895)

[Figure 45 Error value between simulation and initial values 37](#_Toc130996896)

[Figure 46 Implementation of Job Shop Scheduling 38](#_Toc130996897)

# LIST OF TABLES

[Table 1 Description for the dataset 19](#_Toc130996457)

[Table 2 Explanation of Abbreviation 19](#_Toc130996458)

[Table 3 Specific values for machine failure 20](#_Toc130996459)

[Table 4 Measurement regarding Decision Tree 25](#_Toc130996460)

[Table 5 Measurement regarding Random Forest 27](#_Toc130996461)

[Table 6 Evaluation Features 38](#_Toc130996462)

[Table 7 Comparison between 3 ML strategies 39](#_Toc130996463)

[Table 8 New Evaluation Features 39](#_Toc130996464)

[Table 9 Digital Twin Features 40](#_Toc130996465)

# LIST OF SYMBOLS AND ABBREVIATIONS

|  |  |
| --- | --- |
| Abbreviations | Full name |
| DMP | Digital Manufacturing Platforms |
| AGV | Automated Guided Vehicle |
| DT | Digital Twin |
| JSS | Job Shop Scheduling |
| DFJSP | Dynamic Flexible Job Shop Scheduling |
| SMAC | Social, Mobile, Analytics, Cloud |
| ML | Machine Learning |
| DES | Discrete Event Simulation |
| CMU | Cumulative Machine Utilization |
| AUC | Area Under Curve |
| ROC | Receiver Operating Characteristic |
| CPS | Cyber Physical System |
| HML | Hybrid Machine Learning |

# CHAPTER 1 INTRODUCTION

## 1.1 Settings

Job shop scheduling plays an important role in the manufacturing processes. It aims to allocate a set of jobs to the matched machines in the available time to optimize the given objectives, e.g. makespan, energy consumption and robustness, while considering series of constraints(Li, Yibing, et al., 2023). An efficient and high-quality scheduling system can reasonably allocate relevant resources, reduce the overall system scheduling time under the premise of ensuring quality, and be able to flexibly respond to possible accidents in production. The scheduling model can be realized through discrete event simulation, and the simpy package in python provides an effective tool for the process. In the last century, some scholars proposed several job shop scheduling models with the aim to optimize the production process and results. However, these techniques have always been facing a discrepancy between actual production line and ideal input models.

With the advancement of the Industry 4.0 era, job shop scheduling has become a significant link in industrial production. The digital transformation that has been continuously promoted in the past two years is also giving impetus to the further progress of this technology. The introduction of the digital twin model makes real-time information interaction and dynamic scheduling possible, which makes great contribution to the digital manufacturing platforms especially in machine failure prediction, structural optimization design, intelligent management and control, etc.

The digital twin model provides a cloud virtual space to track the production line situation in the actual physical space. The application of the DT can be divided into two aspects: prediction and implementation scheduling, mainly through machine learning and cloud computing methods to analyze and reschedule to achieve the optimization effect of job shop scheduling.

## 1.2 Objective and Motivation

The motivation for working on the automatic scheduling for AGV-based digital manufacturing platforms using digital twin are as follows:

* To solve multi-objective JSS problem

When solving an initial scheduling problem, the researchers consider it as a single-objective optimization task, such as the make span and idle time. The most frequently used single optimization goal is the minimization of the maximum completion time or make span (Zhao, Yangming, et al.). Job shop scheduling problem is a thriving area of scheduling research, which has been concerned and studied widely by scholars in engineering and academic fields(Xiong, Hegen, et al.). The optimization and application of the job scheduling model is of great significance to the industry.

* 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).

* To accelerate into the era of Industry 4.0

The 21st century has seen the development of a fourth industrial revolution, characterized improvements in technology, interconnectivity, and the development of smart automation (Pandey, Vaishnavi, et al., 2023). Under the Industry 4.0, the scheduling should deal with a smart and distributed manufacturing system supported by novel and emerging manufacturing technologies such as mass customization, Cyber-Physics Systems, Digital Twin, and SMAC (Social, Mobile, Analytics, Cloud) (Zhan, Jian, et al., 2019). Digitalization and intelligence will become the development trend in the next few years.

## 1.3 Contributions

The contributions of this essay are as follows:

* I learn and understand the advantages of discrete-event simulation methods in the process of pipeline operation, develop simulation using SimPy to generate the required data and constraints on the simulated AGV platforms.
* I realize the simulation of job shop scheduling with multiple production lines and products, use multidimensional arrays to store the data, simulate the corresponding operation modes of AGV on DMP systems.
* I create several models to compensate the scheduling problems, and develop the evaluation and optimization methods regarding blocking probabilities, efficiency and throughput.
* I develop 3 ML models to predict the possibility of machine failures, and comparing the performances between decision tree, random forests and xgboost. In this regard, xgboost is finally applied to the prediction part. (The dataset is obtained from UCI repository contains the historical data collected by sensors.)
* I apply the ML model to the job shop scheduling sequences and generate the machine failure model.
* I make physical simulation models on the occupation rate of machines using the method of polynomial fitting, and make comparison between experimental data with the physical model.
* I build up hybrid digital twin models, compare different performances under different parameters, and optimize the physical model in this way.
* I develop the digital twin ML-assisted simulation models to implement the job shop scheduling problems. Finally realize the DT-based prediction, detection and evaluation for the machine in JSS.

## 1.4 Thesis Outline

The dissertation is organized as follows:

***Chapter 2: Background***

This chapter introduces the macro concept of AGV-DMP systems, and explaining the direction of the digital manufacturing industry in recent years. New advancements in technologies have greatly prompted the development of industry 4.0, it is meaningful to understand the how the specific methods are applied to the digital era. It also explains the features and advantages of the DES(discrete event simulation), and how to use SimPy simulation platforms model the real world systems.

***Chapter 3: Literature review***

This chapter concentrates on the specific detailed professional field research, such as the development process and current frontier of a certain technology. I will introduce the current work on job shop problems, the path scheduling methods, the application of ML in the prediction area, how to build up a digital twin model to realize the dynamic scheduling, and how digitalization is shaping the manufacturing industry.

***Chapter 4: Design and Performance***

This chapter mainly introduces the design and performance of this project, which mainly includes three categories:

* How to build the job shop models on Simply platform, how data such as sequences and time is planned, how the data metrics are analyzed, how to evaluate and optimize the model at this stage.
* How to establish and introduce a machine failure model, the comparisons between different ML methods, and put the ML prediction model into the sequences of job shop scheduling.
* How to establish and introduce the digital twin model into the scheduling sequences, how to establish a basic physical mode using polynomial curve fitting, evaluate and apply it from the machine’s perspective and finally realize the DT ML-assisted simulation models.

***Chapter 5: Results***

This chapter shows the results of design and performance of each model, and summarize the implementation of JSS, evaluation of the scheduling system, application of ML strategies in machine failure prediction and optimization of JSS on DT-based neural network. I also organize the tables to compare the results using different methods.

***Chapter 6: Future work***

This chapter indicates some technical extensions, and unfinished angles to be optimized.

***Chapter 7: Conclusions***

This chapter concludes the overall structure of the essay, and explains what have been down in the two semesters.

# CHAPTER 2 BACKGROUND

## 2.1 AGV-based digital manufacturing platforms

AGV scheduling is a crucial process in manufacturing systems. It aims at the allocation and timing management of transportation tasks to achieve goals such as reduction in completion time, energy consumption, and resource idle rate (Heger J, Voss T, 2018). As technology redefines the manufacturing industry, it has become essential to use digital manufacturing platforms to maximize operational efficiency and accelerate Industry 4.0. Currently, the rise of digital platforms for manufacturing is a reality as they play a key role in supporting collaborative manufacturing, service, analysis, and forecasting processes in business networks. Moreover, they provide flexibility to enterprises by fast and simple orchestration of services and applications. (Waltz, Thomas J., et al)

This system showcases Industry 4.0 methods, and encompasses both new production systems and legacy equipment within a series of advanced manufacturing scenarios, which is being used for both research and training with a range of industrial partners. The implementation of this research is expected to increase the productivity and flexibility for manufacturing systems by improving shop-floor decision-making efficiency (Yao, Fengjia., et al).

## 2.2 Simulation platform –Simpy

Discrete event simulation (DES) is a method used to model real world systems that can be decomposed into a set of logically separate processes that autonomously progress through time. Each event occurs on a specific process, and is assigned a logical time (a timestamp).

SimPy is a process-based discrete-event simulation framework based on standard Python. Processes in SimPy are defined by Python generator functions and can, for example, be used to model active components like customers, vehicles or agents. SimPy also provides various types of shared resources to model limited capacity congestion points (like servers, checkout counters and tunnels). Simulations can be performed “as fast as possible”, in real time (wall clock time) or by manually stepping through the events.

Though it is theoretically possible to do continuous simulations with SimPy, it has no features that help with that. Also, SimPy is not really required for simulations with a fixed step size and where your processes don’t interact with each other or with shared resources.

## 2.3 Scheduling methods

Scheduling means allocating the shared resources to competing activities over a period of time. The focus is on investigating machine scheduling problems. The jobs and machines represent the activities and the shared resources separately.

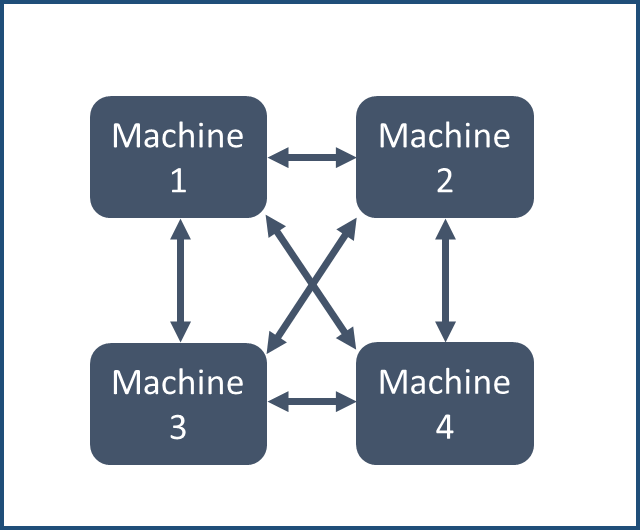


Figure Job shop Scheduling Mode

The job shop scheduling problem (JSSP for short) deals with such a problem, that is, we determine the order or sequence for processing a group of jobs through multiple machines in the best way. More specifically, there is a set of jobs which should be processed by a group of m machines. Every job from has to go through a fixed sequence of machines to be processed. Job consists of an ordered sequence of operations , ,…, . During a given time, a specific machine must process operation . Each machine can process only one operation and one job at most one at a time. The maximum completion time of all the jobs is called the length or makespan of the schedule. The problem is to schedule the jobs to minimize the time .

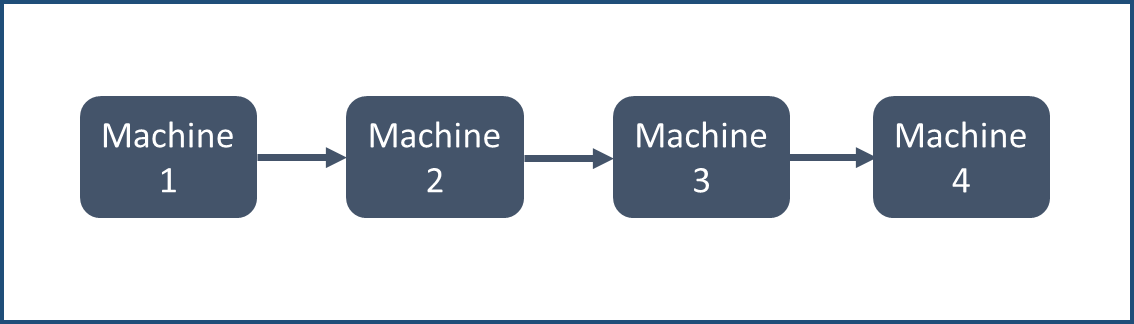


Figure  Flow shop Scheduling Mode

Flow shop scheduling is a special case of job shop scheduling. Different from the job shop scheduling, when dealing with flow shop scheduling problem, operations within one job must be processed in a given order. This means the first operation gets executed on the first machine, and when the first operation has been finished, the second operation starts to be processed on the second machine, and so on until all the given operations have been done. The problem is also to schedule the jobs to minimize the makespan.

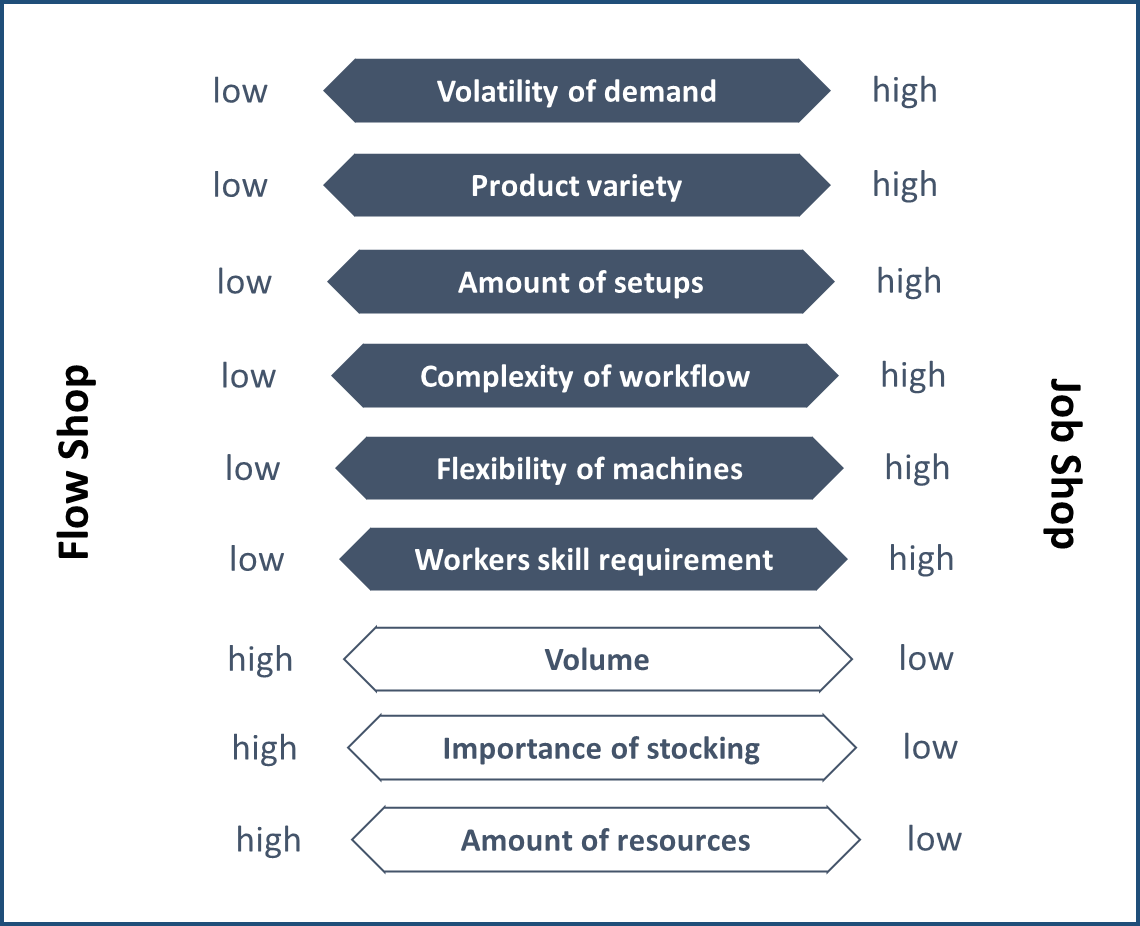


Figure Comparation between Job shop and Flow shop

In terms of a flow shop, this would be a manufacturing site with 100% standardization operated in an assembly production line. On the other hand, a job shop would be a business with 100% customization with a typical batch size of 1, which implies that every finished product is unique.

## 2.4 Digitalization

Digitization is the process of converting information into a digital (i.e. computer-readable) format. The result is the representation of an object, image, sound, document, or signal (usually an analog signal) obtained by generating a series of numbers that describe a discrete set of points or samples. The result is called digital representation or, more specifically, a digital image, for the object, and digital form, for the signal. In modern practice, the digitized data is in the form of binary numbers, which facilitates processing by digital computers and other operations, but digitizing simply means "the conversion of analog source material into a numerical format"; the decimal or any other number system can be used instead (Bloomberg, Jason, 2019).

The increasing production intricacy in a more demanding market is gathering momentum for the integration of the physical and digital world. At the same time, human’s escalating practical needs for industrial products are challenging the digital model’s capability to interact with the physical object. The digital twin was conceived in this context and has sparked a far-reaching industry revolution. In the beginning, digital twins were employed primarily in the military and aerospace (Mantel, Ronald J., and Henri RA Landeweerd). Currently, the digital twin is in a period of rapid development, which has already progressed from theoretical research to pragmatic implementation and has been used in various fields.

Digital Twin is now considered as one of the enabling technologies of Industry 4.0, which has played a significant role in in many different industries. Through using model, sensors, data and software, it can couple the physical system with the virtual representation to monitor and analyze data. Even if there are changes in its working environment such as weather and the input products, digital twin can quickly adapt to it and take these new data into account with the help of the support enabling technologies like machine learning. So in order to supervise the production system of the industry in real time, deal with the problems in time, and obtain the optimal production line, it is of great importance to apply the digital world and the digital twin technology.

# CHAPTER 3 LITERATURE REVIEW

The job shop scheduling problem (JSSP) is one of the most classical and important combinatorial optimization problems in the field of operational research and management science (Xiong, Hegen, et al., 2022).

Johnson firstly proposed the optimal scheduling of the production in a three-machine problem and solved for a restricted case (Johnson, Delmer Martin. 1954), scheduling has been a popular research topic in manufacturing industry. A classical JSSP can be described as follows: in a job shop environment containing several machines , there are a number of jobs , each job, say , contains a serial of operations which need to be processed in a predefined technological sequence in a way to minimize the makespan (Geyik, Faruk, and Ismail Hakki Cedimoglu, 2004). When assigning one job on one machine, it must meet some constraints. Firstly, each job assigned on a machine is associated with a given order and a machining (or performing) time (Zhang, Jian, et al., 2019). Secondly, each machine can perform only one job at any moment (Chen et al. 2012). Lastly, the performing (machining) time of a job is fixed, and once the job is started, it cannot be interrupted (Ju 2007).

The static and dynamic scheduling was distinguished by Jackson. In the actual production scheduling process, there will be many unexpected situations, such as machine failure, blocking, etc. o, according to the changes in the actual conditions, we must constantly adjust the scheduling plan, which is called dynamic scheduling (Wan-Liang et al.,2003). The earliest assignment rule is introduced in the dynamic scheduling system (Jackson, James R,1955) and carried out with the priority rule and genetic algorithm which shows the effectiveness (Fox M S, et al., 1984). Yu et al. proposed to use the ant colony optimization algorithm after treatment of mutant so as not to fall into local optimum easily, and to solve this urgent order inserting in dynamic scheduling problem (Yan-hai, Hu, et al., 2005). An immune algorithm was proposed for dynamic scheduling of flexible manufacturing cell, which also brought in sliding window technique and scheduling task pool concept (Yu, Jian-jun, et al., 2008). Recently it is becoming a hotspot to combine various algorithms to reach a comprehensive result. Genetic algorithms are used to minimize the operation time of the parallel machine scheduling problem, combining with different scheduling rules to improve the algorithm performance in different situations (Liu Min, et al., 2000). Zhang et al. (2017) took into account the shortest processing time and the balanced use of machines, and put forward the multi-population genetic algorithm based on the multi-objective scheduling of flexible job-shop.

Planning and scheduling within the confine of Industry 4.0 have gained much traction over the last few years. Erol et al. (2012), next to only scheduling machines, also look at the simultaneously scheduling of AGVs within their MAS. Recently, DT has been widely applied in the manufacturing field, especially in prognostics, assembly, resource virtualization, information integration, asset simulation, etc. By introducing DT, further convergence between physical and virtual spaces of the job-shop can be achieved, which greatly enables dynamic scheduling (Zhang, Meng, et al., 2021). In addition, some scholars paid attention on the distributed scheduling. Due to the improvement of the accuracy control of the cyber-physical production system (K. Zhu and Y.Zhang, 2018). The applications of DT in smart workshops bring a chance to reduce the gap, which makes it an efficient and accessible way to drive the production lines dynamically (R. Rosen, et al., 2015). Based on literature review, my article researches the application of job shop scheduling in the current industry 4.0 era, combining ML methods and path planning algorithms for scheduling optimization.

# CHAPTER 4 DESIGN AND PERFORMANCE

## 4.1 Job shop scheduling simulation models

### 4.1.1 description

Job shop is a common scheduling problem, in which multiple jobs are processed on several machines. Each job consists of a sequence of tasks, which must be performed in a given order, and each task must be processed on a specific machine.

There are several constraints for the job shop problem:

* No task for a job can be started until the previous task for that job is completed.
* A machine can only work on one task at a time.
* A task, once started, must run to completion.

Based on these conditions, I build up a job shop scheduling simulation models using simply package. The details regarding this system are explained in the following parts.

### 4.1.2 simulation scenarios

In the CA1 semester, we completed the simulation of the following three scenarios, whose requirements are as follows.

1. Scenario A

There are three different types of products in one line, labelled 0, 1 and 2. The corresponding products are fed into the corresponding machines 0, 1 and 2 for processing, and upon completion the three types of products are output in the three lines.

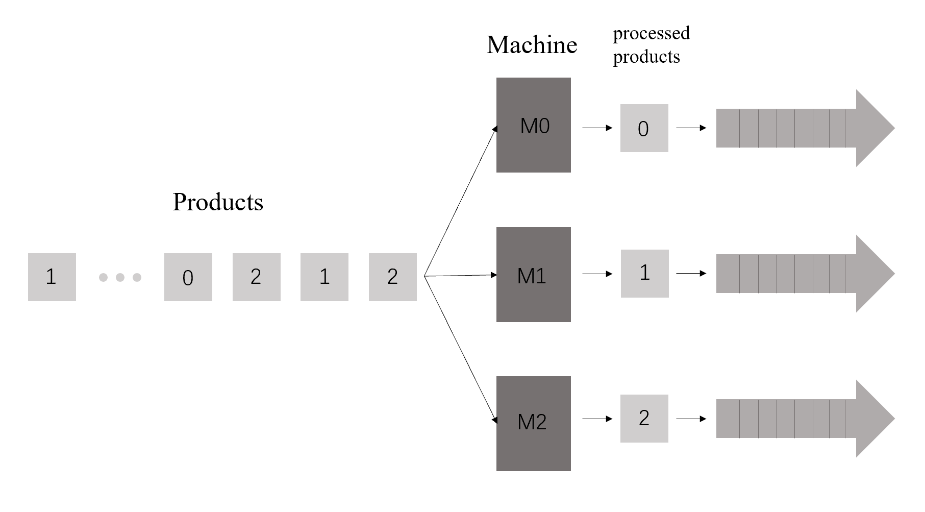


Figure Schematic diagram of Scenario A

1. Scenario B

There are three different types of products in one line, labelled 0, 1 and 2. Each workpiece needs to go through several machines to complete the corresponding processing steps. e.g. product 0 needs machine 0 and machine 1 to complete the processing, product 1 needs machines 2, 3 and 4 to process, product 2 needs machines 5, 6, 7 and 8 to process. And three types of products are output in the three lines respectively.

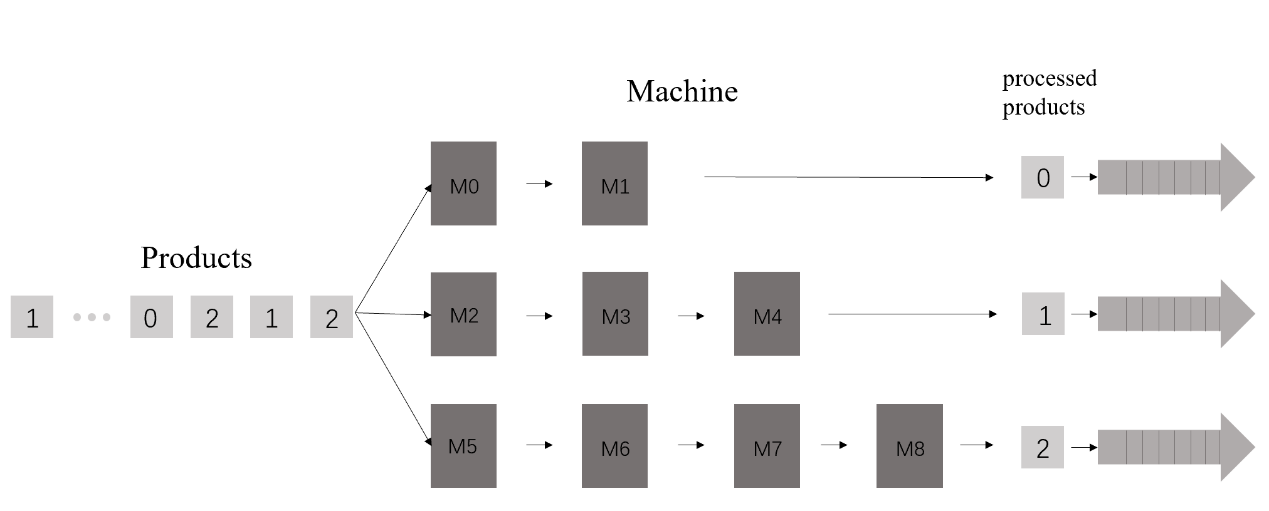


Figure Schematic diagram of Scenario B

1. Scenario C

In this scenario there are several products, each with a number of processing steps, and the sequences of products are all randomly generated in terms of their labels, the labels and the number of machines performing the processing, so that some scheduling algorithms are required during the production process, which is the basis for the scheduling of digital manufacturing platforms.

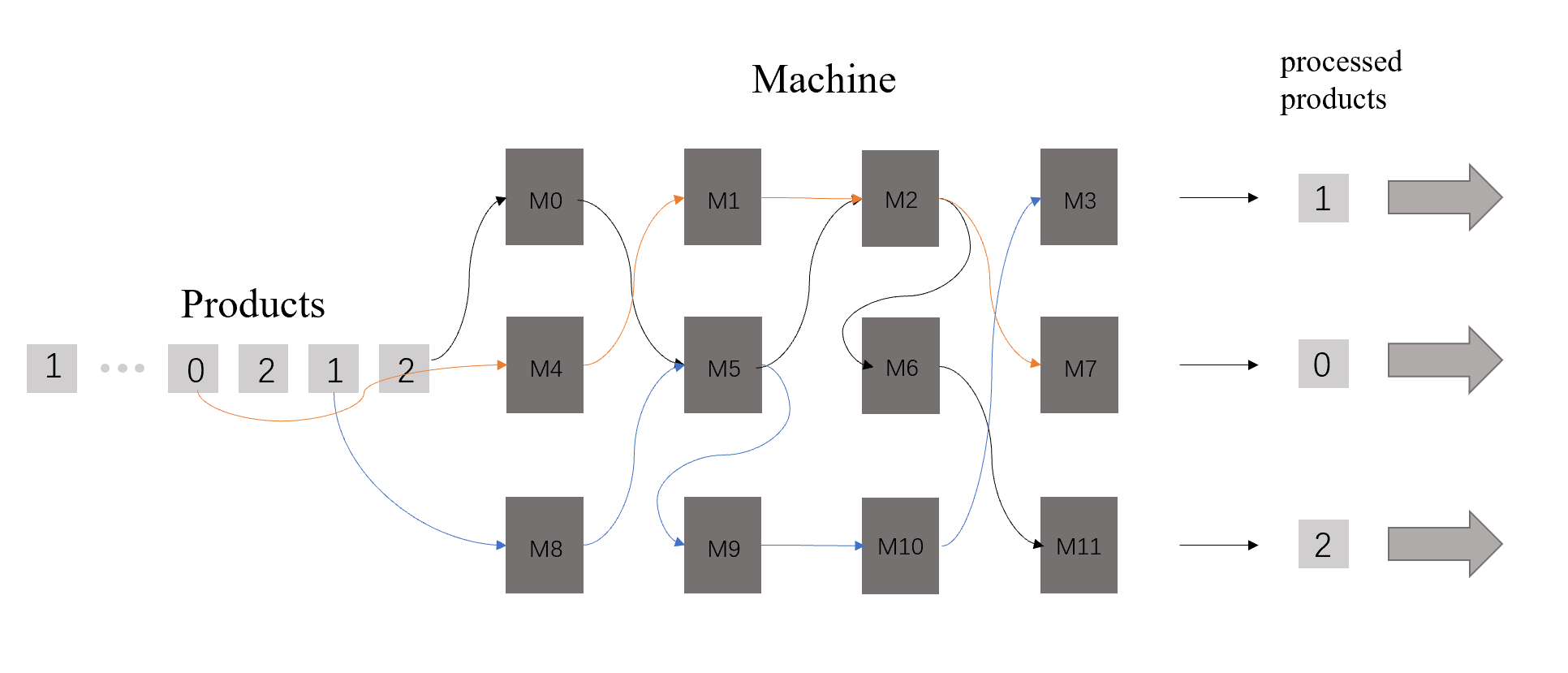


Figure Schematic diagram of Scenario C

In practice, two implementations of this scenario are considered, with small batch production usually using job shop scheduling and large-scale production using flow shop scheduling.

### 4.1.3 basic models of production line

#### 1. Sequences Model

1) Machine sequence:

Randomly generates the number of machine types between (10, 20) and label all machines generated starting from 0.

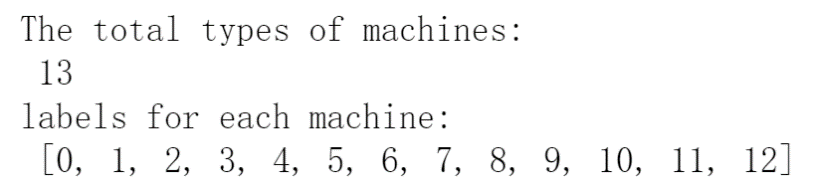


Figure Machine sequence

2) Product sequence:

Randomly generates the number of product types between (15, 25) and label all products generated starting from 0.

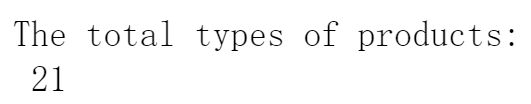


Figure Product number

Randomly generates the number of steps required to produce each product, and the corresponding machine number for each step.

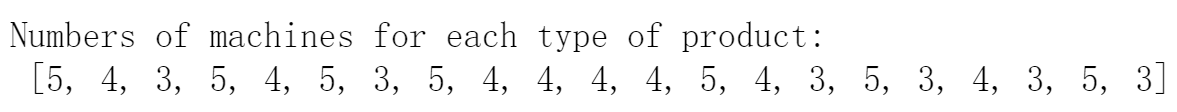


Figure Product’s machine number

Using a two-dimensional array to record the length of the sequence, the class of machines through which each product passes.

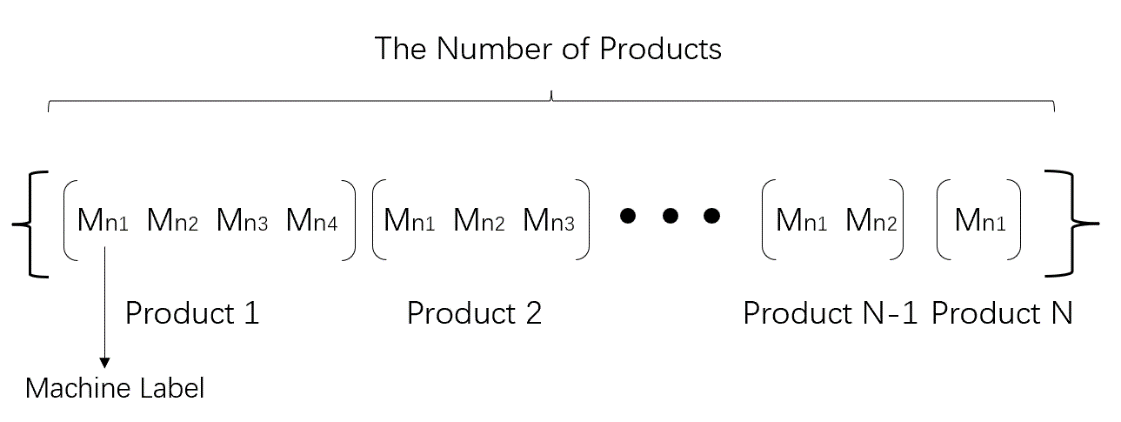


Figure The structure of two-dimensional array

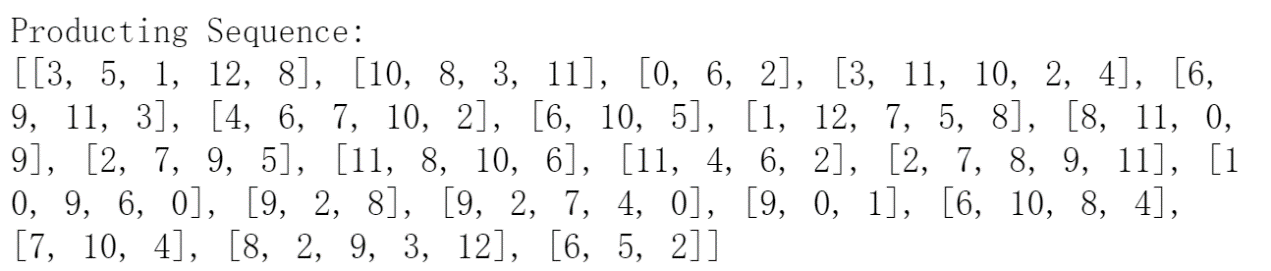


Figure Product sequence

The figure above shows that the procedures of processing each product, eg. the first array shows the production process of product1, it needs to go through the machine3, machine4, machine1 and machine9 to finish the production processing.

3) Input sequence:

Set up five production lines and randomly generate the length and the type of the product sequence for each line.

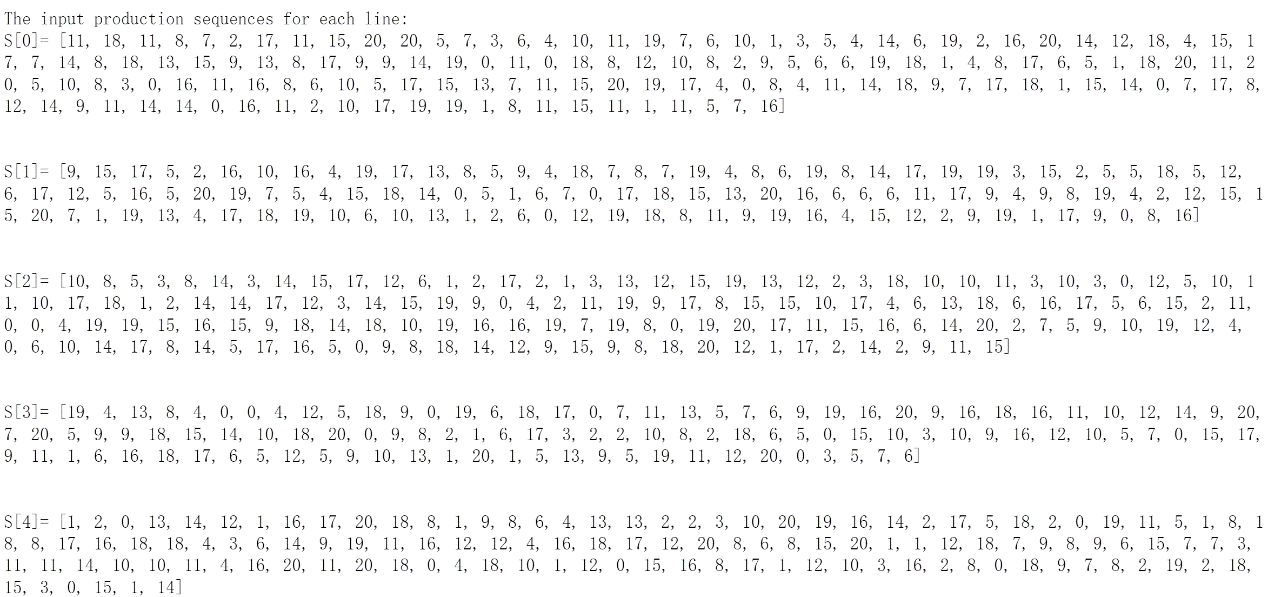


Figure Input sequence

**2. Parameter Assignment**

1. Processing Time for Each Machine:

Random generation of processing times for each machine between (10, 30), which means the nth machine spend on one product.



Figure Machine’s processing time

1. Block Time:

It takes a certain amount of time for the machine to process each product, so when the machine is occupied as the part enters the line, blocking occurs. We use the blocking parameter to keep track of the current machine blocking situation.

block1: track the total amount of products that have been blocked by other products by a certain time

block2: track how many products are blocked by other products at a certain time

1. Current Machine Status:

When the machine is processing a product, then the status of the machine is occupied. This is essential when determining the operational status of the product.

m\_occupied: The number of machines that are occupied

m\_vacant: The number of machines that are vacant

### 4.1.4 simulations in simpy environment

#### 1. Simpy Simulation

Create a simpy environment, then allocate relevant resources for the use, here set the number of each type as resources.

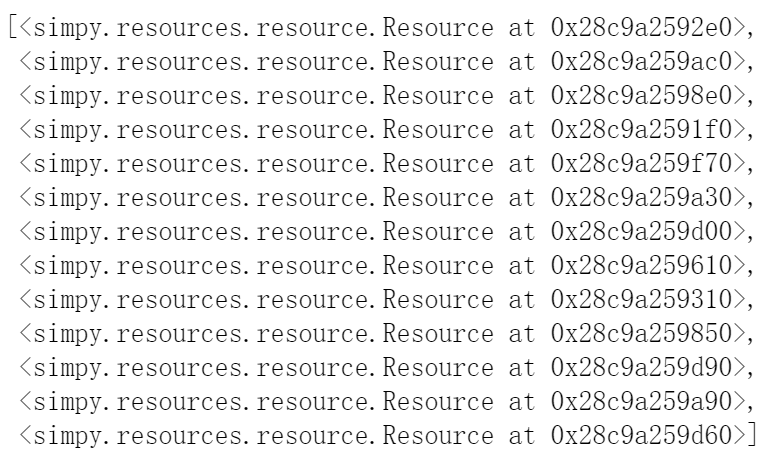


Figure Resource allocation

#### 2. Implementing Functions

The first function (input\_sequence( )) is the action to inject the arrival product into the system.

The mutual function (system\_process( )) is to receive these input sequences, inject the arriving products into the waiting sequence and feeds them to the appropriate machine for processing when the machine is idle.

The current time slice of the simulation space can be obtained during the run with env.now().

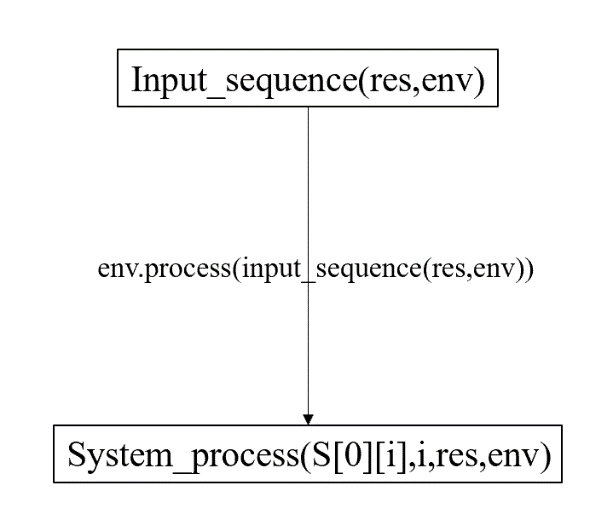


Figure Interaction of functional functions in the simpy environment

*Statement: All simulations below are based on data from pipeline S[0] in Scenario C*

### 4.1.5 job shop scheduling core strategy

#### 1. strategy principle

A job is characterized by its route, its processing requirements, and its priority. In a job shop the mix of products is a key issue in deciding how and when to schedule jobs. Jobs may not be completed based on their arrival pattern in order to minimize costly machine set-ups and change-overs. Work may also be scheduled based on processing time, from shortest to longest.

A product is processed by several machines and in the machine data stored for the product, the core strategy is to prioritize the machines according to the length of its waiting list.

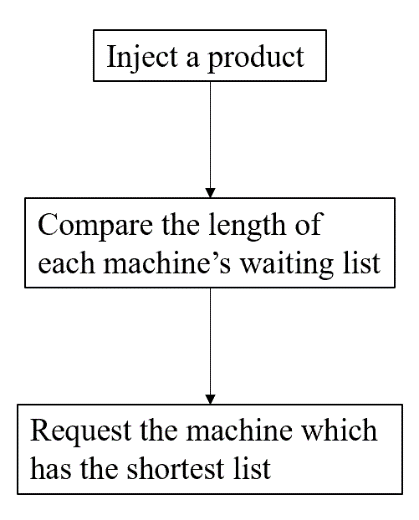


Figure The Core Strategy in Job shop Scheduling

The following diagram provides a concrete illustration of the arrangement of machines running in job shop scheduling

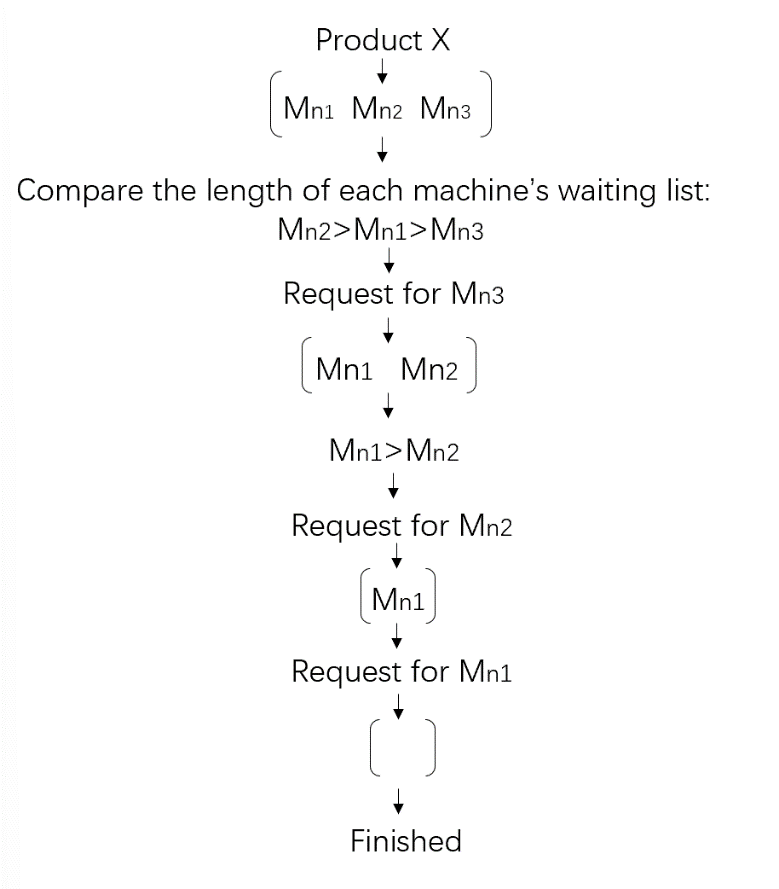


Figure Machine comparative operation mechanism

#### 2. Real simulation

In the real simulation, when a type 11 product goes in to the system at a certain moment, to complete the processing of this product it requires to go through machine11, machine4, machine 6 and machine2.

a. At this point machine2 has the shortest waiting sequence;

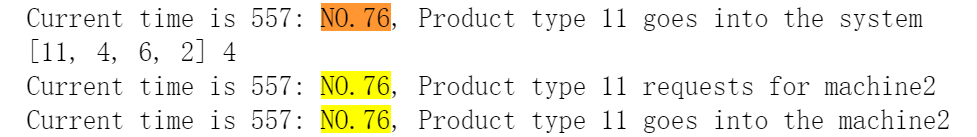
b. After the machining of machine 2 is completed, machine 4 has the shortest waiting sequence of the three remaining machines.

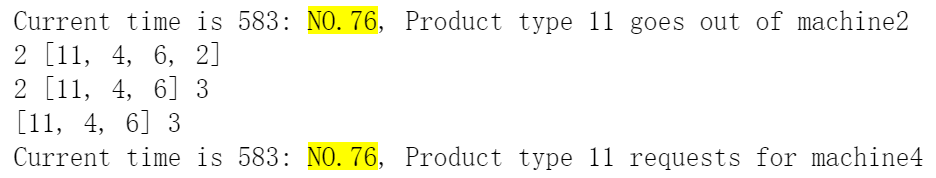
c. After the machining of machine 4 is completed, machine 11 has the shortest waiting sequence of the three remaining machines.

d. The remaining machine 6 carries out the final processing of the product.

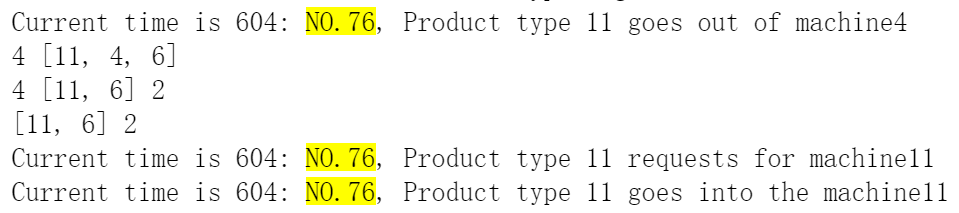
This is a simulation of the actual situation.

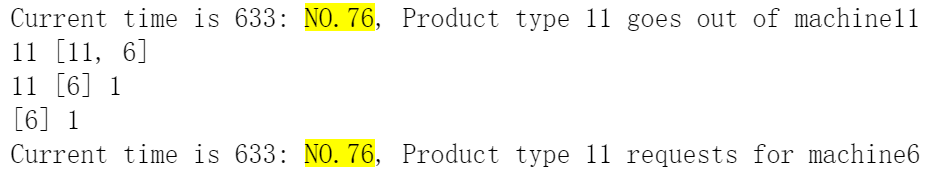
X Y: X represents the “Minimum waiting sequence machines”, Y represents the number of steps remaining. e.g. 2 [11, 4, 6] 3: The processing of the product by machine2 is carried out/completed and needs to be completed by machine11, machine4 and machine6, where existing 3 machines need to be passed.













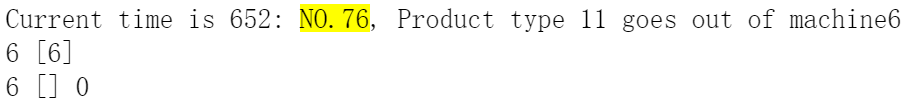


Figure Real simulation example

### 4.1.6 evaluation and optimization

1. Cumulative Machine Utilization (CMU)

To evaluate the performance of this scheduling system, cumulative machine utilization (CMU) can be used from a machine perspective.

The average Response Time for each machine:

[42.96666666666667, 28.894736842105264, 279.1515151515151,

34.357142857142854, 134.875, 16.393939393939394,

56.61290322580645, 122.58974358974359, 316.0769230769231,

322.03508771929825, 240.6122448979592, 273.265306122449,

27.047619047619047]

Each machine's CMU is:

[2.5274509803921568, 1.2039473684210527, 10.736596736596736,

2.0210084033613445, 6.4226190476190474, 1.3661616161616161,

4.354838709677419, 5.3299888517279825, 11.288461538461538,

11.10465819721718, 8.91156462585034, 9.422941590429275,

1.3523809523809525]

From the results of the calculation we can see that the value of CMU varies going to a large interval. If a machine is used frequently it can lead to a long waiting interval for that machine, which reduces the efficiency of the whole scheduling system. This means that the smaller the value of the CMU (always greater than 1), the more efficiently the machine is used.

2. Back-up Machines

In order to solve the problem of reducing the overall efficiency of the scheduling system due to different usage frequency, we add some back-up machines to improve the overall efficiency of the system and reduce the CMU value of each type of machine according to the usage time and frequency of different types of machines.

Rules for adding back-up machines:

1. Count the number of times each machine is used in the S[0] queue.
2. Obtain the total running time of each machine, i.e. the frequency of machine use multiplied by the running time of the machine.
3. Get the shortest runtime of all machines, divide the total runtime of each machine by the shortest runtime and round up to the nearest integer to the number of additional spare machines.

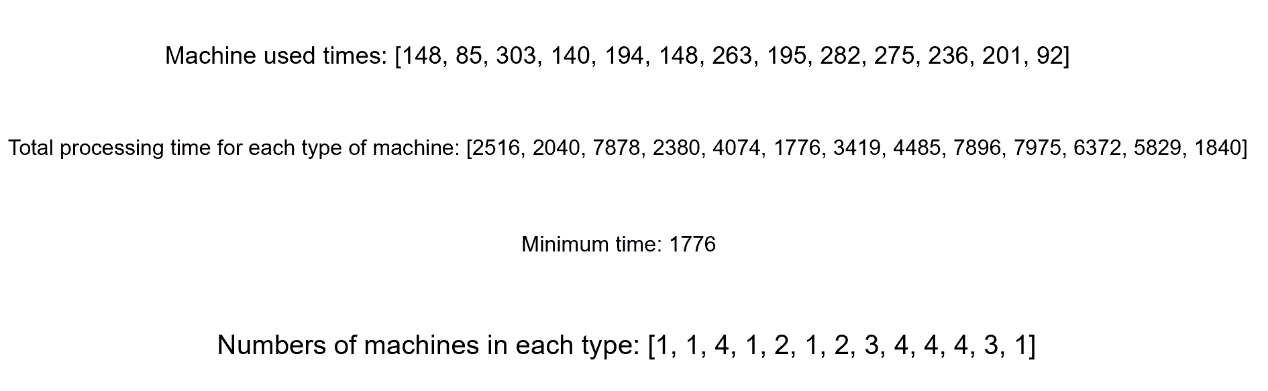


Figure Back-up machine number

After adding the back-up machines, we calculate the Response Time and CMU again.

The average Response Time for each machine:

[30.966666666666665, 34.63157894736842, 27.318181818181817, 24.821428571428573, 25.083333333333332, 15.121212121212121, 14.983870967741936, 23.76923076923077, 29.76923076923077, 32.70175438596491, 27.653061224489797, 33.93877551020408, 27.285714285714285]

Each type of machine's CMU is:

[1.8215686274509804, 1.4429824561403508, 1.0506993006993006, 1.4600840336134455, 1.1944444444444444, 1.2601010101010102, 1.152605459057072, 1.0334448160535117, 1.0631868131868132, 1.1276467029643074, 1.0241874527588815, 1.1703026038001407, 1.3642857142857143]

We compare the efficiency of the original machine with the efficiency after adding the back-up machine (CMU is the comparison value).

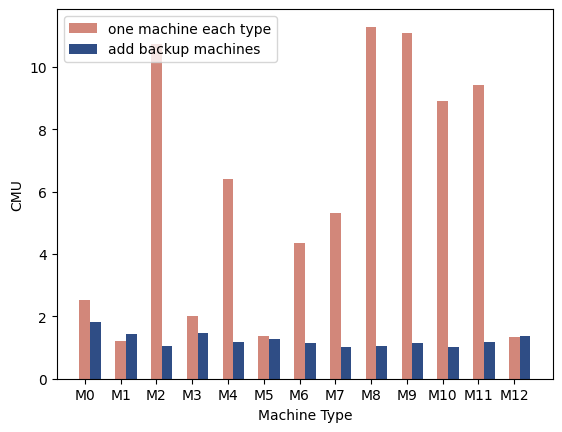


Figure Comparison of machine utilization rates

From the bar chart we can see that the average CMU values have been optimized to a large extent and that the introduction of the back machine has had a huge effect on the efficiency of the system.

3. Occupation Rate

Machine efficiency can also be evaluated by the machine’s occupation rate, which we discretize in time, using time slices to record the number of machines that are occupied by products.

The number of occupied machines and machine occupation ratios are plotted on a scatter plot as shown below.

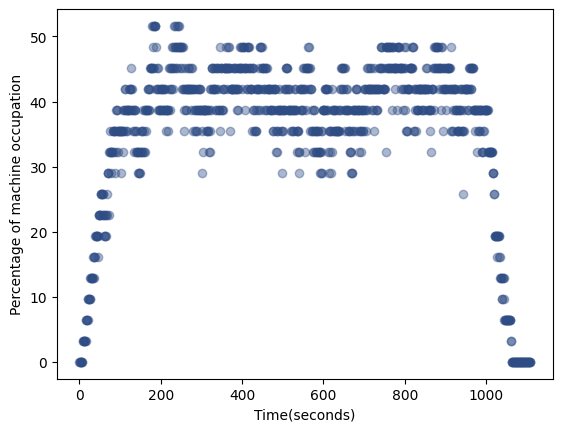
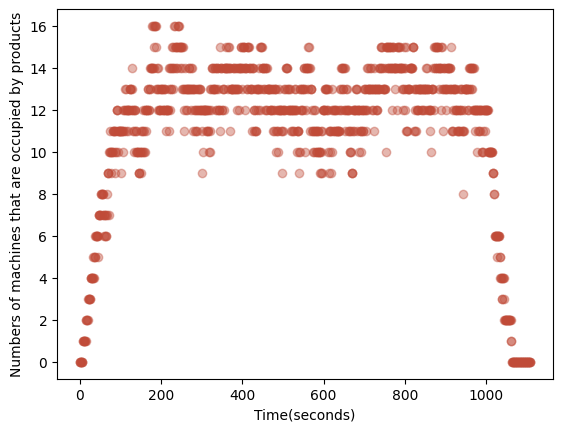


Figure Occupation Rate of machines

4. Blocking Rate

The performance of the system can also be assessed by products by tracking the number of blocked products. We use two variables to track product blocking, block1 to record all product blocking throughout the process and block2 to record the number of product blocking at a given time.

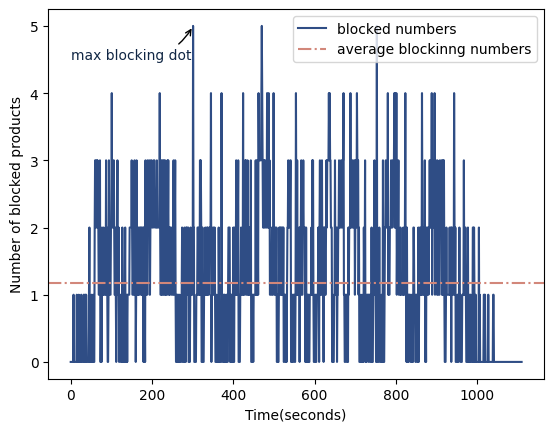


Figure Blocked products

The following data on the entire blocking process was obtained by recording and calculating.

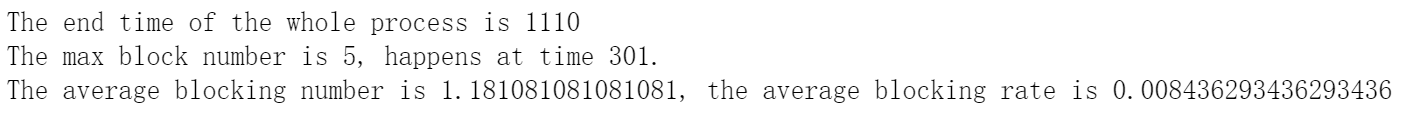


Figure Performance of job shop

## 4.2 Machine Failure Models

### 4.2.1 description

In the actual production process, the factory has numerous assembly lines, and a large amount of machines are put into use at the same time. During the operation, unexpected machine breakdown happens which should be taken into account in the simulation process.

As the machine failure occurrence is a significant factor that cause catastrophic consequences on the JSS, I am going to use ML techniques to build a prediction model then apply it to the simulation part, which increases the realism and usability and lays a foundation for the final real-time data-driven optimization in job shop scheduling.

I got the dataset from UCI Machine Learning Repository, which reflects real predictive maintenance data encountered in industry. It consists of 10,000 data points stored as rows with features like product type, air temperature, process temperature, rotational speed, torque wear, machine failure. The machine failures are grouped into 5 subcategories.

In this section, I go through data cleaning, data visualization and analyzing, training and testing prediction respectively. I used three machine learning methods, namely decision tree, random forest and XGBoost, compared and analyzed the results of the three algorithms. Finally implementing ML techniques and merge the model into the job shop scheduling sequences.

### 4.2.2 dataset analysis and visualization

#### 1. Data description

I use pandas for working on data frames and arrays, converting the csv format dataset into dataframe which is accessible to be used. The dataset shown in the table1 is the historical data of machine conditions containing several features.

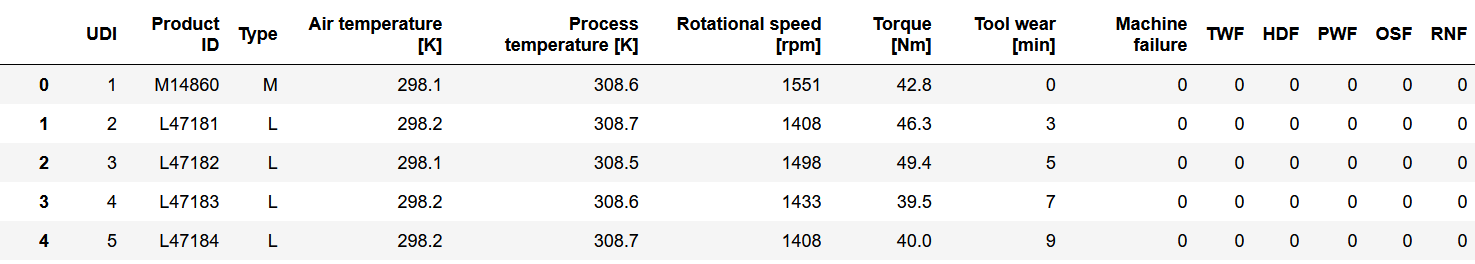


Table Description for the dataset

UDI, Product ID and Type are the basic information about each machine

Air temperature, process temperature, rotational speed torque and tool wear are the features about the machine conditions, which are the consequences to the machine failure.

The last 5 columns show the failure conditions of the machines, the value is 1 means the machine encounters failure, the consequences can be seen in the last 5 columns.

|  |  |
| --- | --- |
| Machine failure abbreviation | Full Name |
| TWF | Tool Wear Failure |
| HDF | Heat Dissipation Failure |
| PWF | Power Failure |
| OSF | Overstrain Failure |
| RNF | Random Failure |

Table Explanation of Abbreviation

#### 2. Data Visualization

The objective of our model is to predict the possibility of machine failure, then I visualize the consequences in the pie chart shown below.

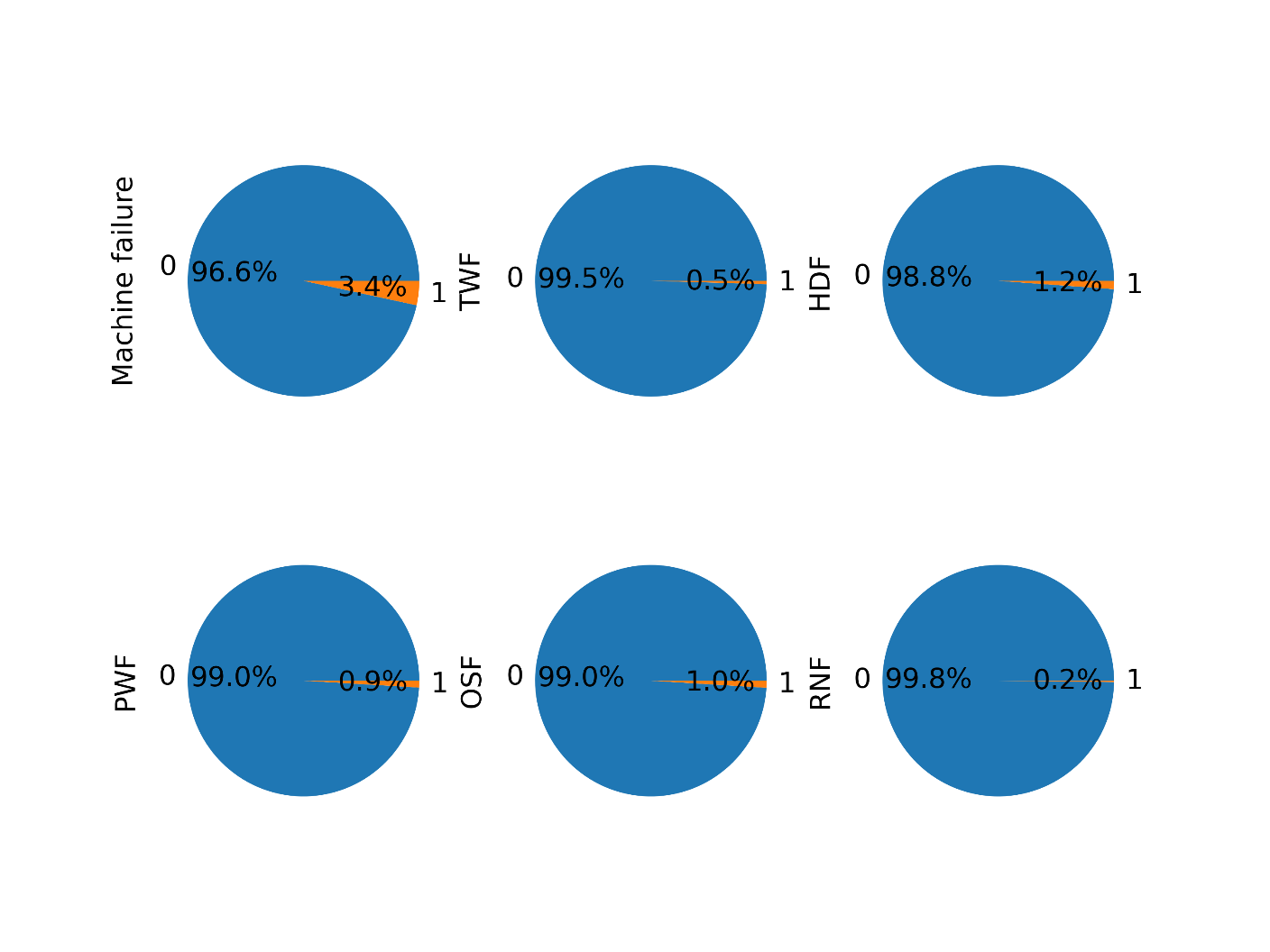


Figure Pie chart for the consequences of machine failure

From the pie chart, we can figure out that of all machines in operation, the probability of a machine failure is 3.4%. HDF is the most important factor causing machine damage, which accounts for 1.2%.

The specific values are shown in the table below

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

Table Specific values for machine failure

Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters. It also helps to form some simple classification models by drawing some simple lines or make linear separation in our dataset.

In this part, I load the data in the seaborn library and call the pairplot function which can explicitly show the correlations between the features, red points when hue=1, blue points when hue=0.

The pair figure is shown below.

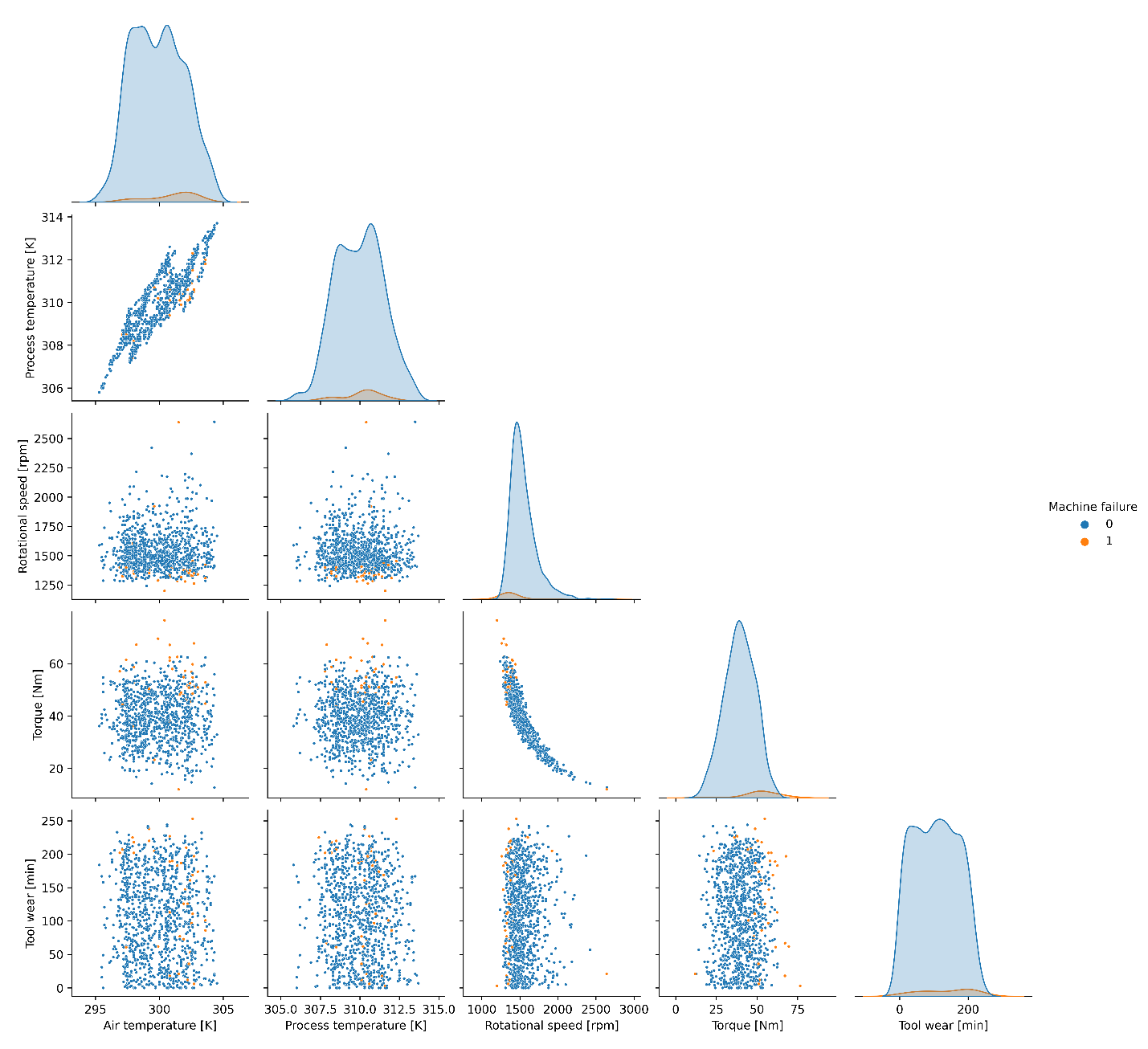


Figure Pairs Plot of gathered data

A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.

The display of the correlation matrix uses Pearson’s Product-Moment Correlation, and the pearson heat map is shown below.

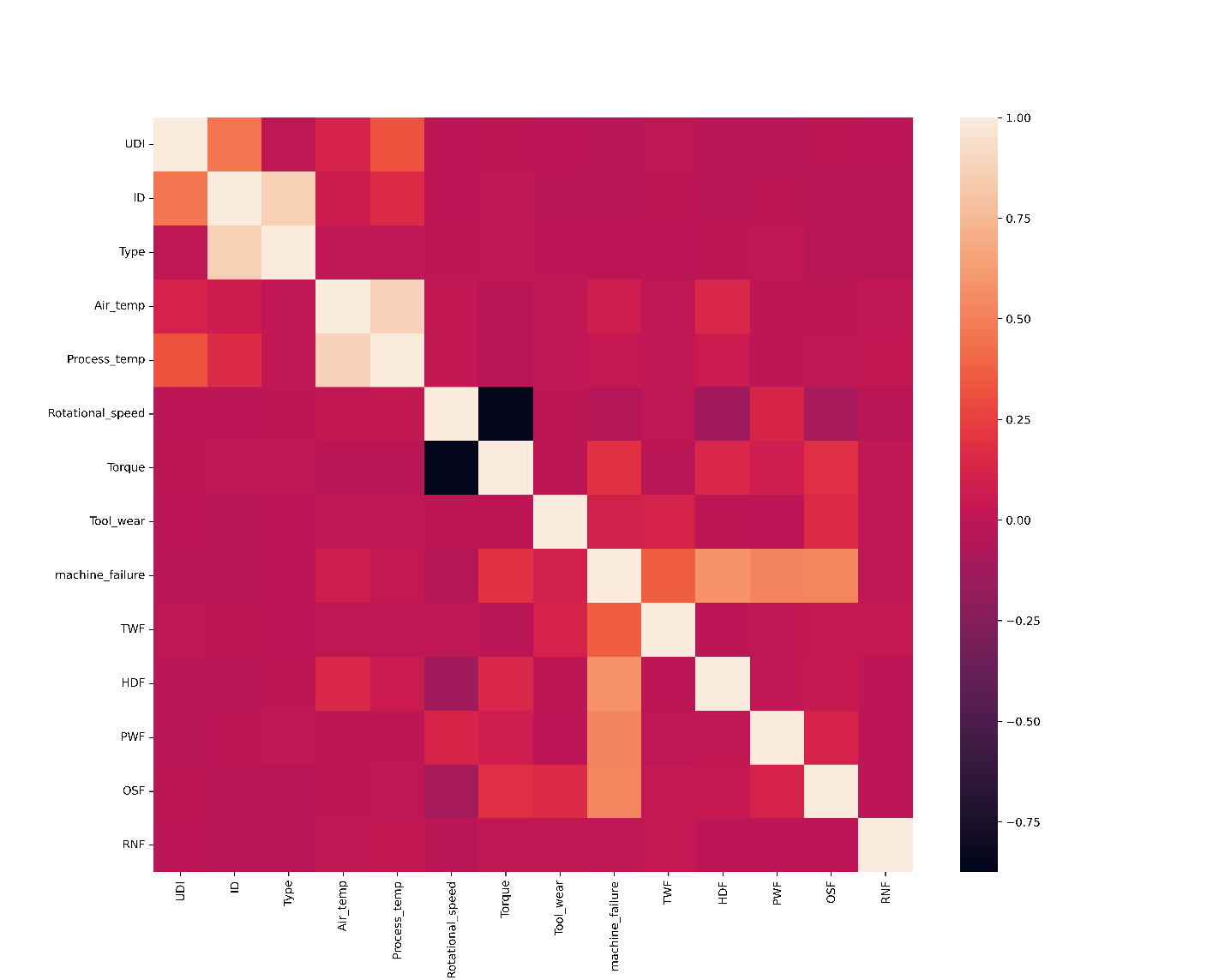


Figure Heatmap of gathered data

### 4.2.3 machine learning based data modeling

#### 1. Basic preparation

In order to predict the damage frequency of the machines, I use three machine learning methods for modeling prediction, namely decision tree, random forest and xgboost. Through the comparison of the accuracy of the results, we chose xgboost for the final application. In the following, we describe the establishment and results of the three models in detail.

Firstly, First we need to do the initial processing of the dataset, I prepare the training set and the training set, and divide the data set with a ratio of 8.5:1.5 as the basis for machine learning model training. Then I drop the additional information stored in the dataframe, which lays the foundation for the modeling part.

There are two significant features in the classification session, ROC Curve and AUC.

1. ROC curve

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate.

True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

False Positive Rate (FPR) is defined as follows:

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

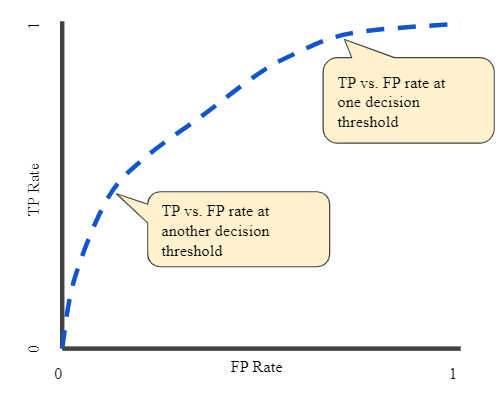


Figure TP vs. FP rate at different classification thresholds

To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called AUC.

1. AUC: Area Under the ROC Curve

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

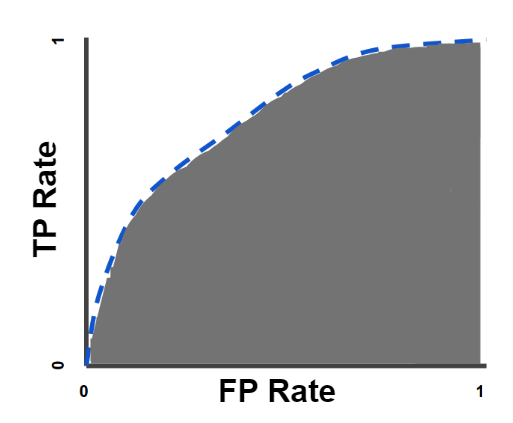


Figure AUC (Area under the ROC Curve)

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.

AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

AUC is desirable for the following two reasons:

* AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values.
* AUC is classification-threshold-invariant. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

#### 2. Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

The following graph shows the structure of the decision tree, the dots of each not explains how the classification is built up.

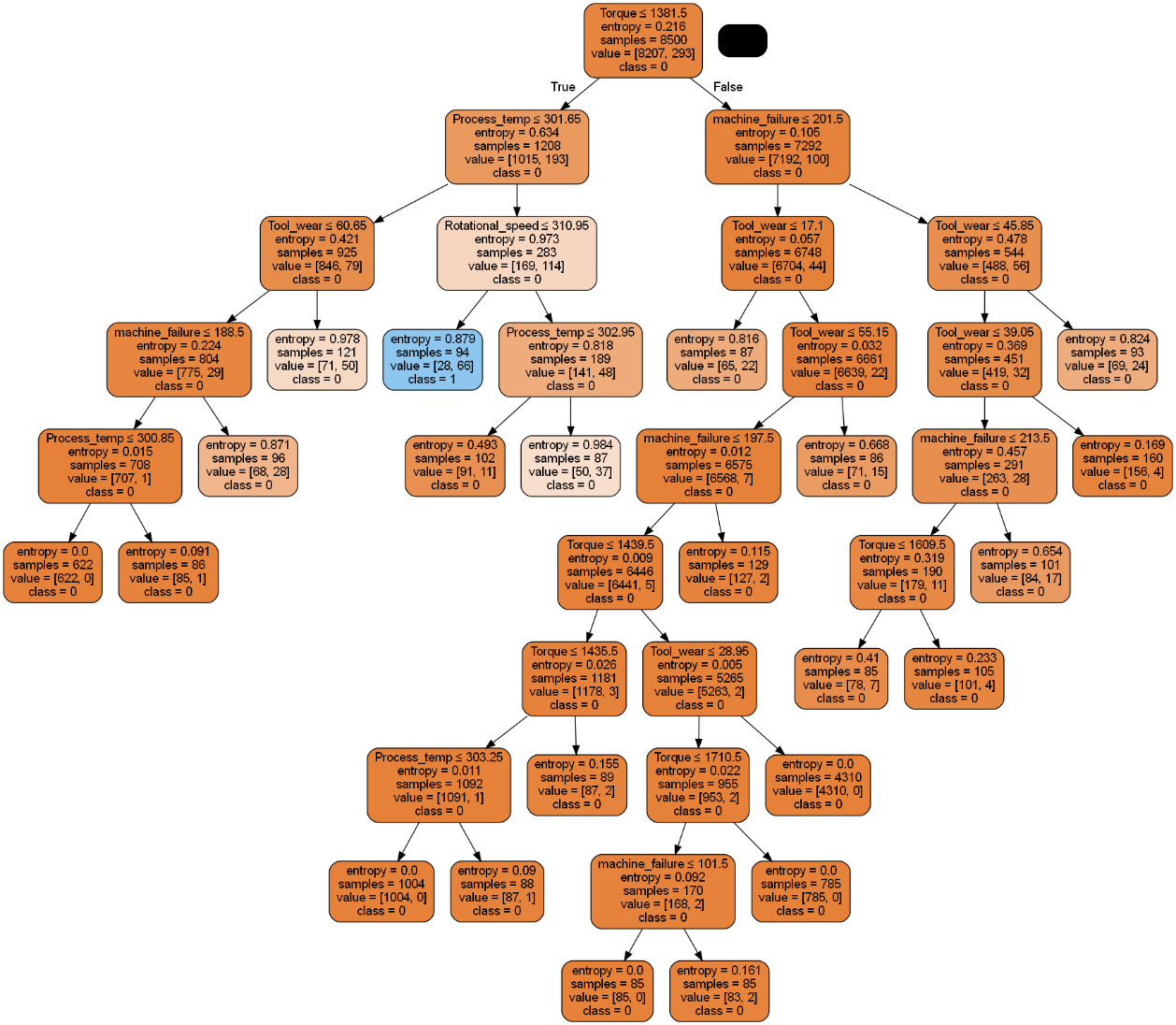


Figure The structure of Decision Tree

In this algorithm structure, the value of AUC reaches 0.58. It can be seen that the accuracy rate of the model is 0.7, but the recall rate for the damaged machine is only 0.15, which shows that the accuracy rate of this model for this data set is very low and has great limitations. Therefore, a better algorithm needs to be found.

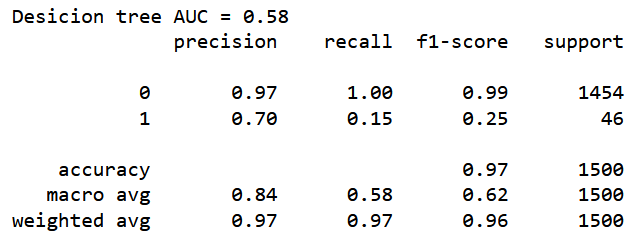


Table Measurement regarding Decision Tree

#### 3. Random Forest

Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

This is an optimization algorithm based on decision trees, with higher complexity and higher accuracy. The figure below shows how the three trees of the random forest are organized.

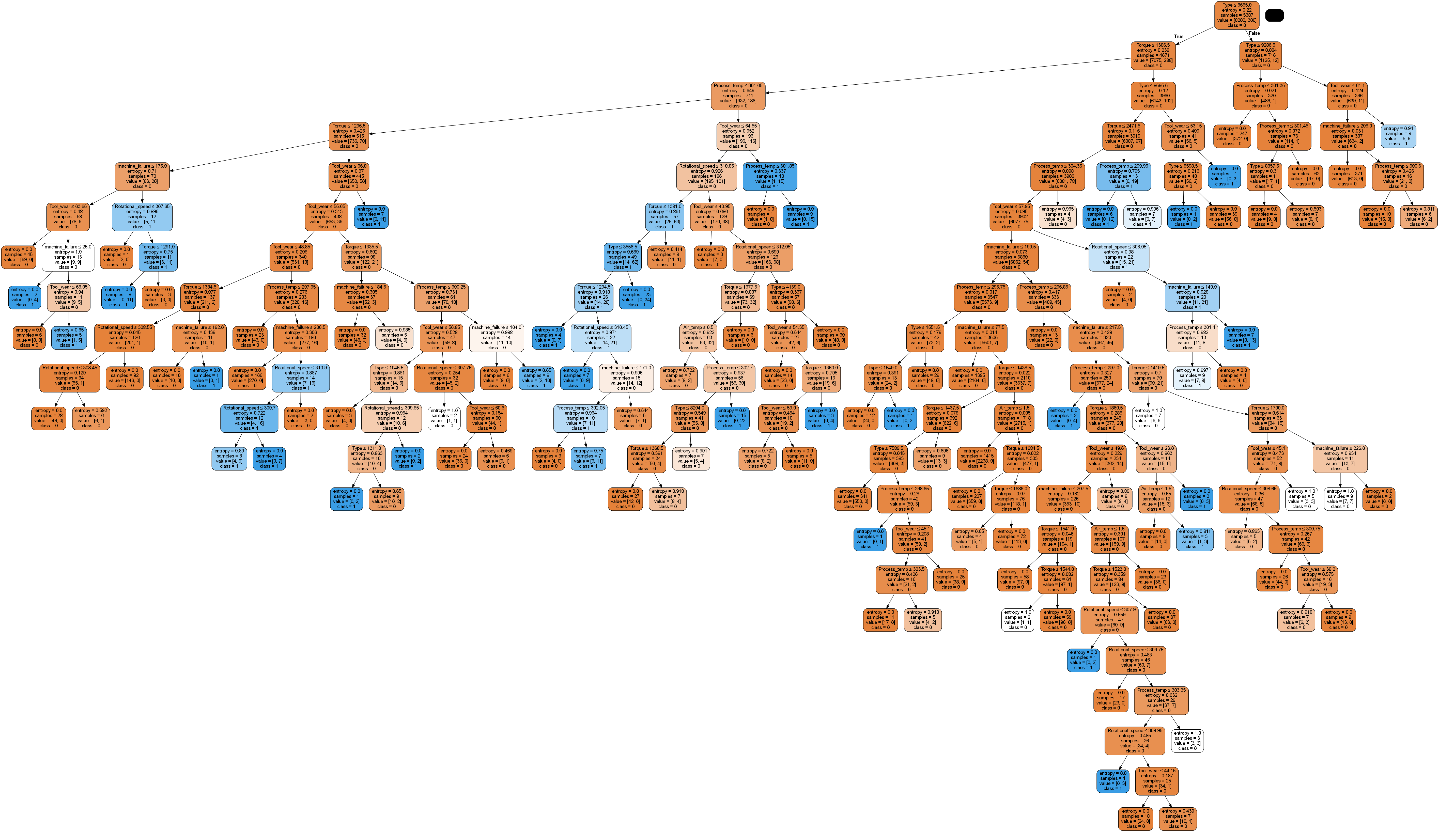


Figure The Structure of Random Forest RF1

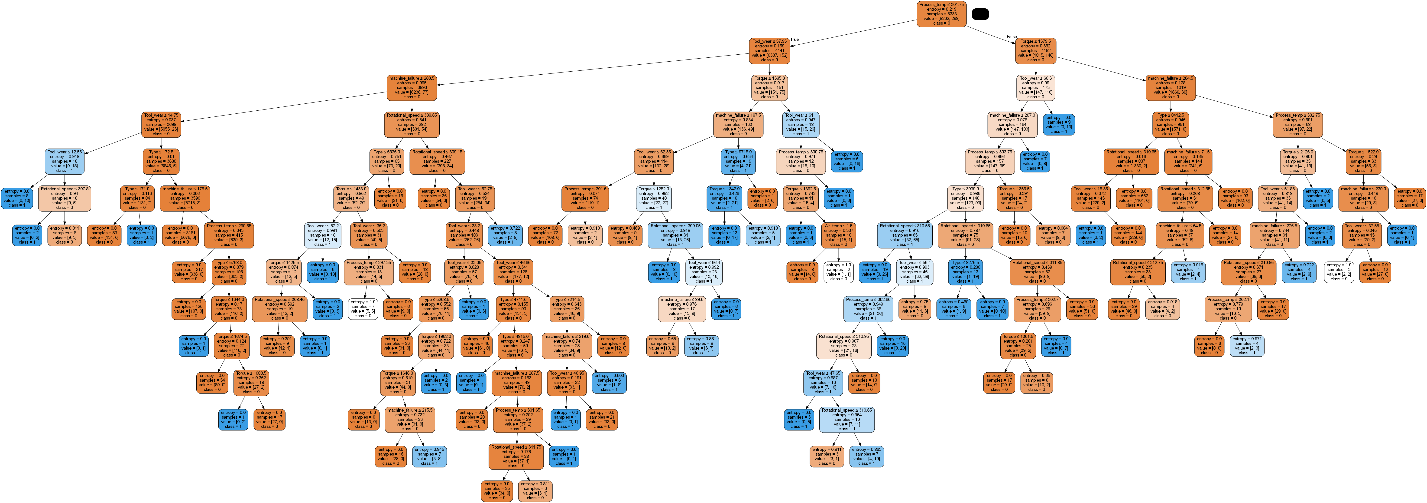


Figure The Structure of Random Forest RF2

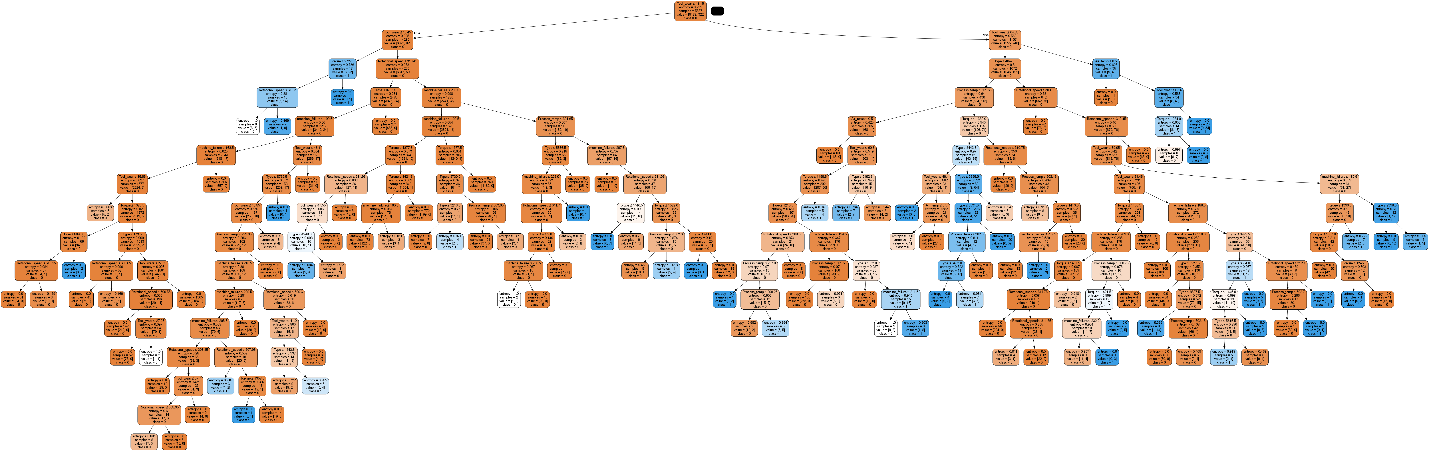


Figure The Structure of Random Forest RF3

In this structure, the AUC value reaches 0.79 which has been largely improved comparing to the first algorithm. The machine failure recall rate is 0.59.

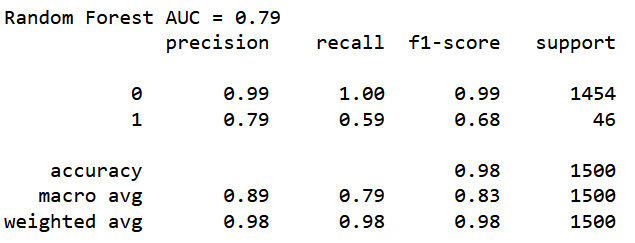


Table Measurement regarding Random Forest

#### 4. XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance which can be used to solve regression, classification, ranking, and user-defined prediction problems. Due to the nature of an ensemble model, you have to be careful about overfitting. The eta parameter is used to prevent this overfitting. The eta can be intuitively thought of as a learning rate. Rather than simply adding the predictions of new trees to the ensemble with full weight, the eta will be multiplied by the residuals being added to reduce their weight. This reduces the complexity of the overall model and the likelihood of overfitting.

The appropriate parameter is extremely essential in the application of XGBoost, I adjusts the values while the eta rate equals 0.26, num\_class equals 3 reaches the best performance. The comparison between different parameters is shown below.

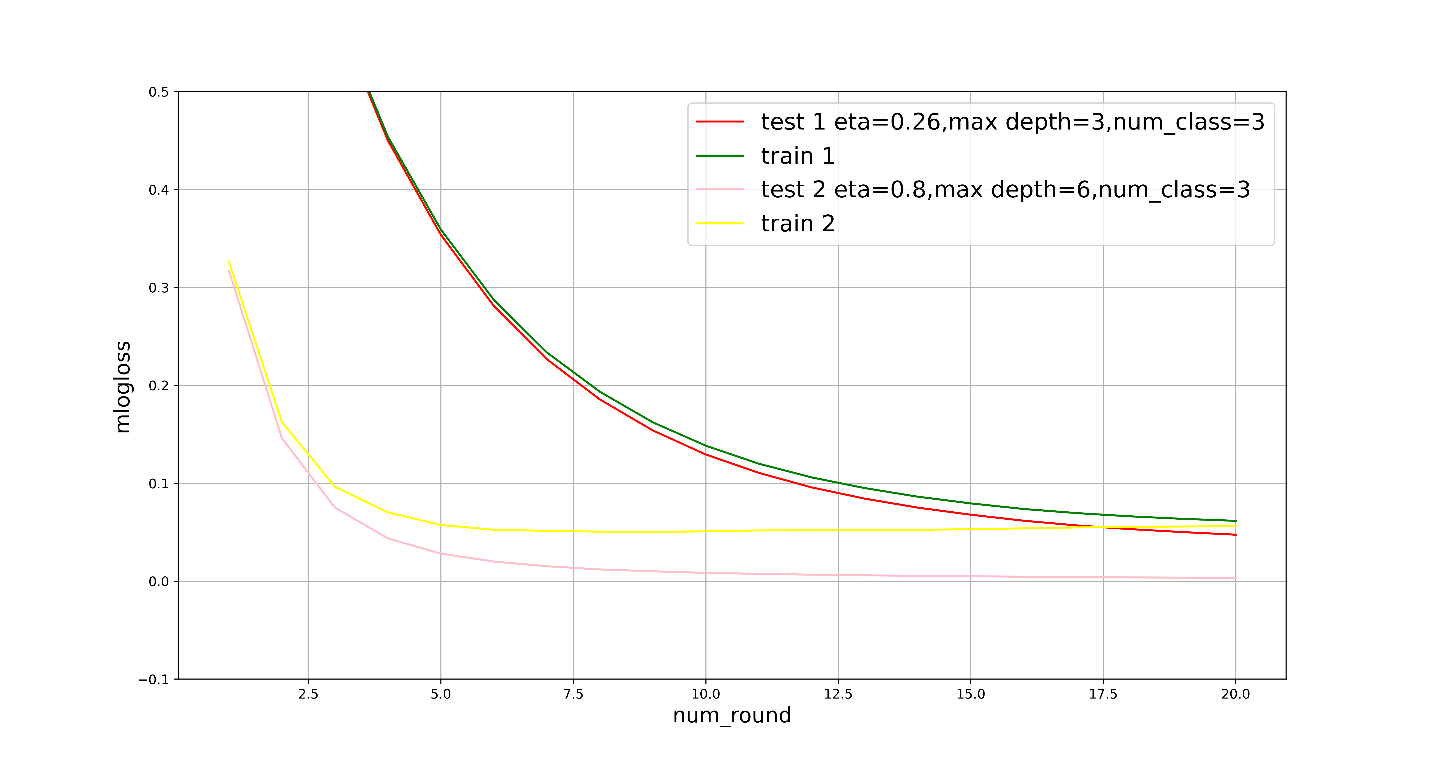


Figure Comparison between different parameters in XGBoost

The structure of this algorithm is too complicated, I store the outcome into a txt file, the basic layout is shown below.

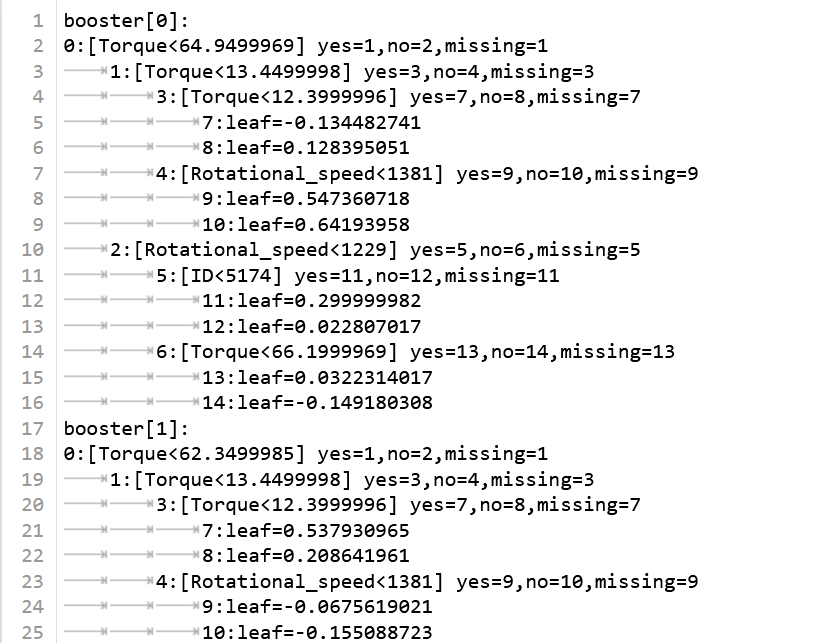


Figure The structure of XGBoost (clip)

Under parameter1 and after the verification, we calculated the final results:

Precision = 0.915068493150685

Recall = 0.767753587169036

Accuracy = 0.9766666666666667

The three main measurement features are reaching a relatively high score, and we use this method to do the prediction in the sequences generated before.

### 4.2.4 machine failure models in job shop scheduling

#### 1. The machine failure-repair model

Based on the historical dataset, we can predict the failure rate of machines. Firstly, we randomly generate the machine conditions in rational ranges, like the air temperature, process temperature, rotational speed, torque and tool wear. Then establishing the machine failure model into the job shop scheduling system, and setting the repair time of each machine. Then we can figure the machine conditions shown below.

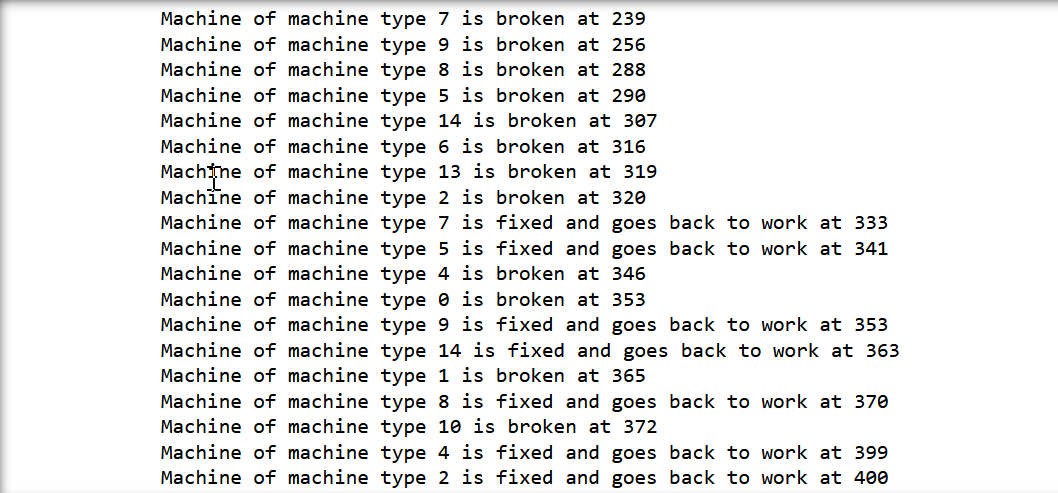


Figure Machine failure conditions

#### 2. The operating state of the simulation system

With the introduction of the machine failure-repair model, the whole operating state of the simulation is then changed. The new model metrics is shown below.

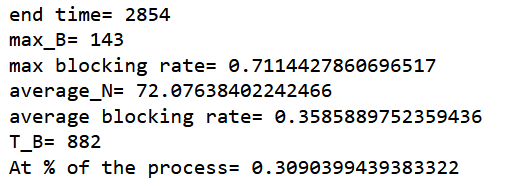


Figure Update results of simulation

The end time of the whole system is prolonged to 2854 while the max blocking numbers have reached 143. In order to describe the operating status of the system in a more intuitive form, I use the following line chart to express the occupied quantity of the machines.

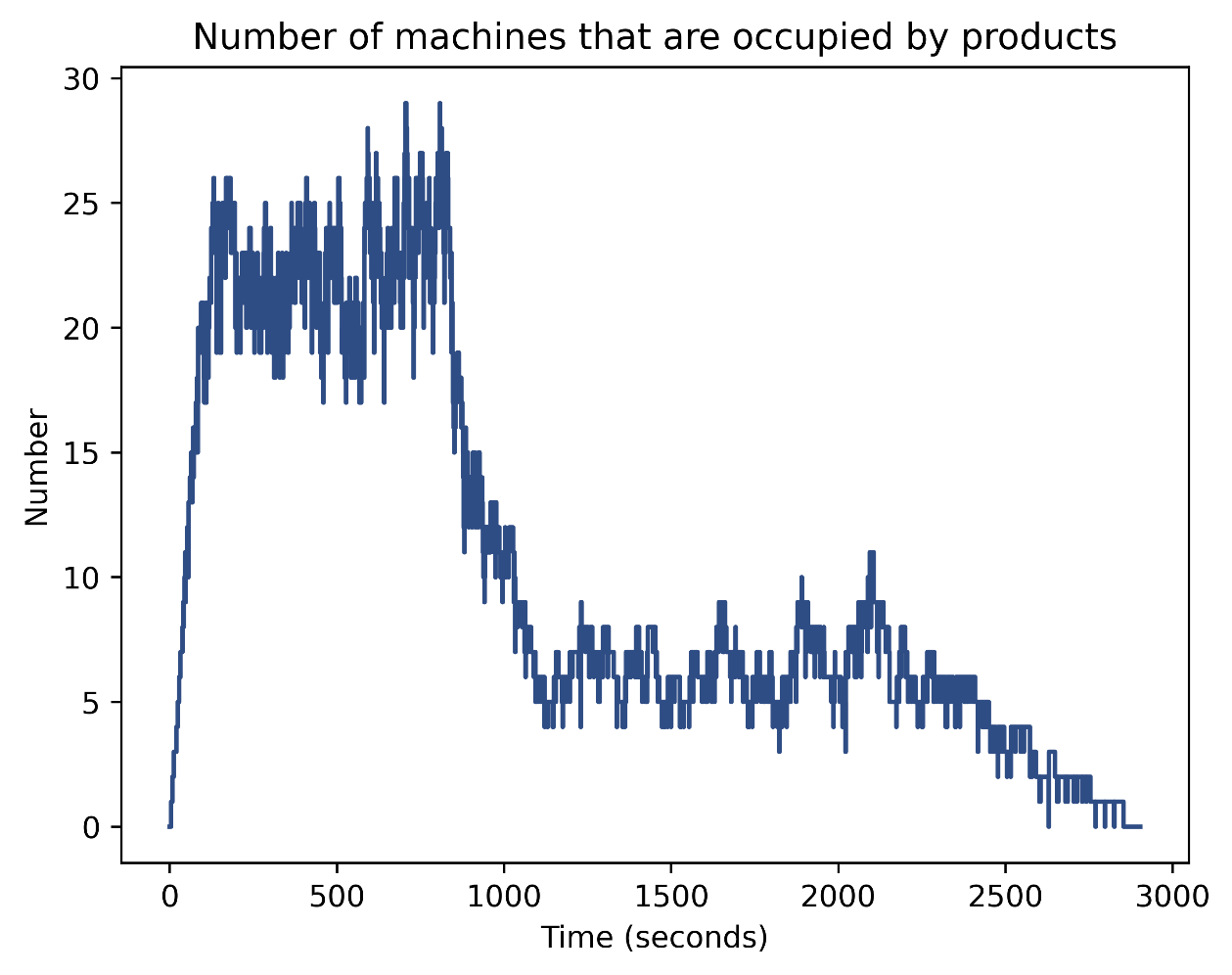


Figure Occupied quantity of the machines

The data is the basis for us to introduce the digital twin machine learning-assisted simulation model. We will use the occupancy ratio of the machines to measure and optimize the efficiency of the system. The specific details will be elaborated in the next section.

## 4.3 Digital Twin Machine Learning-assisted Simulation Models

### 4.3.1 description

Based on the job scheduling simulation models and the machine failure-repair models for the AGV-based Digital Manufacturing Platforms, this section will introduce a digital twin model to optimize the whole system and realize automatic scheduling.

Digital twin models can be divided into two parts — prediction and online scheduling, which are based on machine learning (ML) methods. The basic dataset is from the discrete-event simulation results from section 4.1 regarding the machine’s occupation conditions, and I normalized the data for further use. This section explains how to build up a ML-assisted simulation model, including how to generate the physical simulation model, computing the recurrent neural network based on a hybrid method to do the vibration prediction and creating the data-driven ML model. The results show that the accuracy is greatly improved comparing to the traditional technique.

The ultimate goal is to realize the prediction of the machine condition, and to invest and reduce relevant resources according to the actual situation of machine operation and production, so as to realize the efficiency of automated production.

### 4.3.2 physical/numerical simulation models

#### 1. Polynomial Curve Fitting - Least Squares

The task of polynomial function fitting is to assume that the given data is generated by M polynomial functions and to select the M polynomial function that is most likely to generate these data, i.e. to choose a function among the M polynomial functions that has good predictive power for known data as well as unknown data.

The method of least squares is a mathematical optimization technique. It finds the best function fit to the data by minimizing the sum of squared errors. The unknown data can be easily obtained by using the least square method, and the sum of squares of the errors between the obtained data and the actual data can be minimized. Least squares can also be used for curve fitting. Some other optimization problems can also be formulated by least squares by minimizing energy or maximizing entropy.

Let the training data set be:



Among them, in is the observation value of the input X, in R is the observation value of the corresponding output , 

Let the polynomial of degree be



Among them, X is a single variable input  are parameters.

Use the square loss as the loss function (that is, the least squares method), and the coefficient 1/2 is for the convenience of calculation, and the model and training data are substituted, as follows:



Find the partial derivative of  and make it 0





Calculate

 and 

The values are then brought into the above linear equations to solve.

#### 2.Define the physical model

Based on the polynomial curve fitting function(polyfit) in numpy, I generate the polynomial fitting formula with a degree of 5. The parameters generated in this way better fit the data of the machine occupancy ratio in a specific time with a curve, while avoiding the risk of overfitting.

These are the model parameters:

[

]

The physical equation is:



The physical simulation model is shown below:

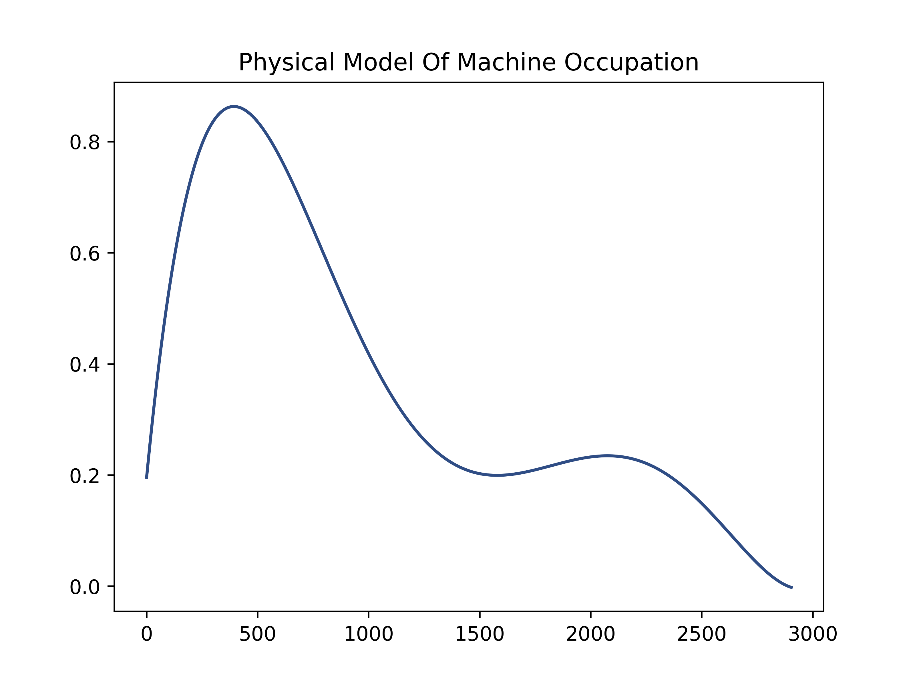


Figure Physical model of machine occupation rate

### 4.3.3 comparation between experimental data with physical model

#### 1. Load measurement data and physical model

Firstly I load the measurement data and physical model created from last section, and now compare the model prediction to the initial values.

The figure is shown below:

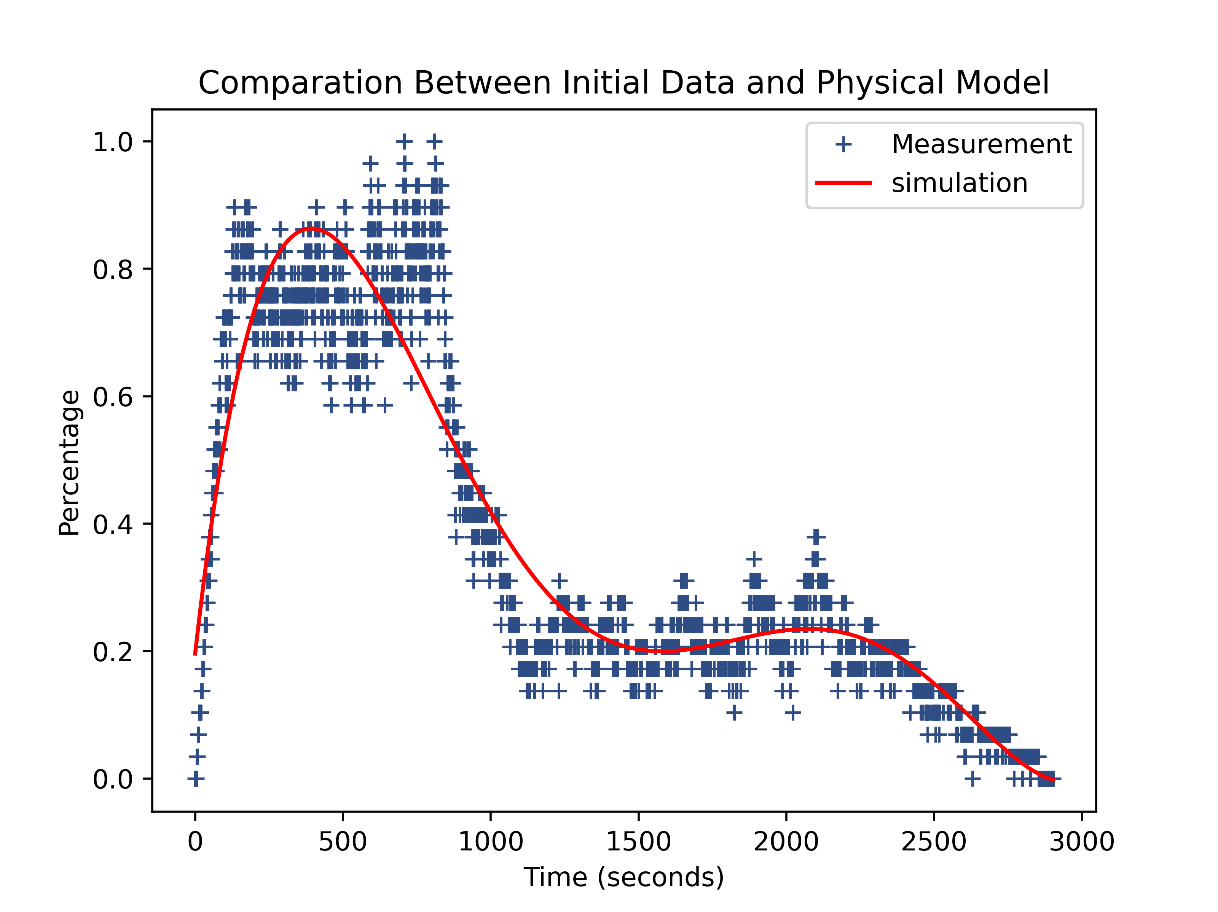


Figure Comparison between initial data and physical model

#### 2. Numerical Comparation

Mean Absolute Error (MAE) is a measure of the average size of the mistakes in a collection of predictions, without taking their direction into account. It is measured as the average absolute difference between the predicted values and the actual values and is used to assess the effectiveness of a regression model.

MAE is calculated as the sum of absolute errors divided by the sample size, the formula is shown below.



Compute the MAE between simulations and measurement:



The mean absolute error is 0.060340457176829725

### 4.3.4 Design of hybrid digital twin

#### 1. The principle of improved physical model

Digital twin refers to the interaction in two spaces, and the data in the virtual space can be corrected through the feedback and modification of the physical space data detected by the sensors, so as to realize the closed-loop automatic control.

Here I build a recurrent neural network model that can compensate for the simulation errors to come up with a better prediction given a set of initial values. The following figure is a simple logical concept map.

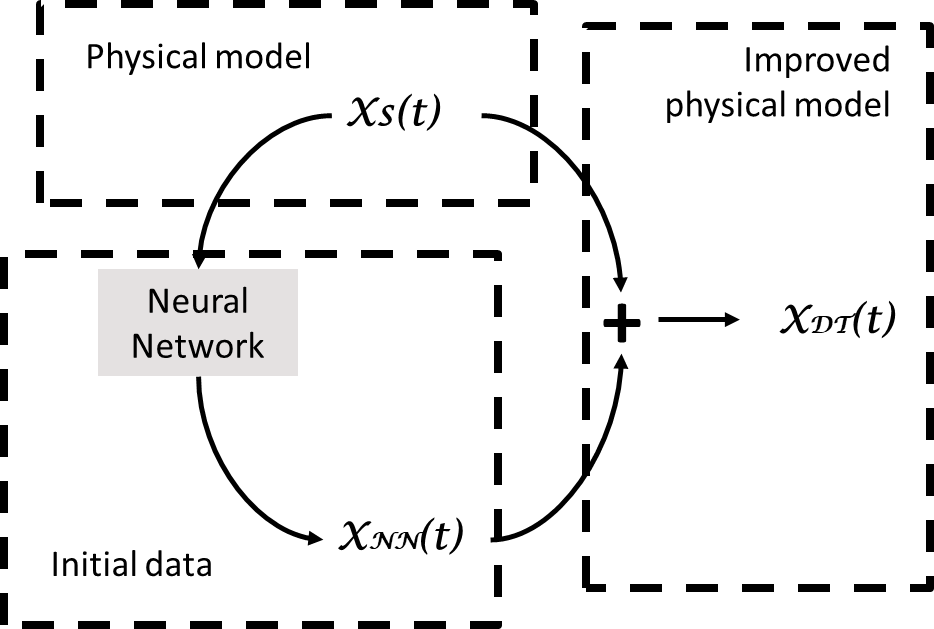


Figure Compensate model

#### 2. Build recurrent neural network

Firstly, we need to define the inputs and outputs, the input data refer to the simulation time series (physical model) while the output data indicate the differences between measurement and simulation. Then split the training set and test set using a ratio of 8:2, for the establishment of the neural network.

Before the model training procedure, check the shape of the training and testing dataset. And we can apply expand dims function to make it matched with the neural network.

Now construct a recurrent neural network using the Keras API and Dense layer, training the model on the training set and observing the performance on the test set to spot overfitting. In this part, we set the epochs as 150 and batch size to be 20.

The process of neural network training is as follows:

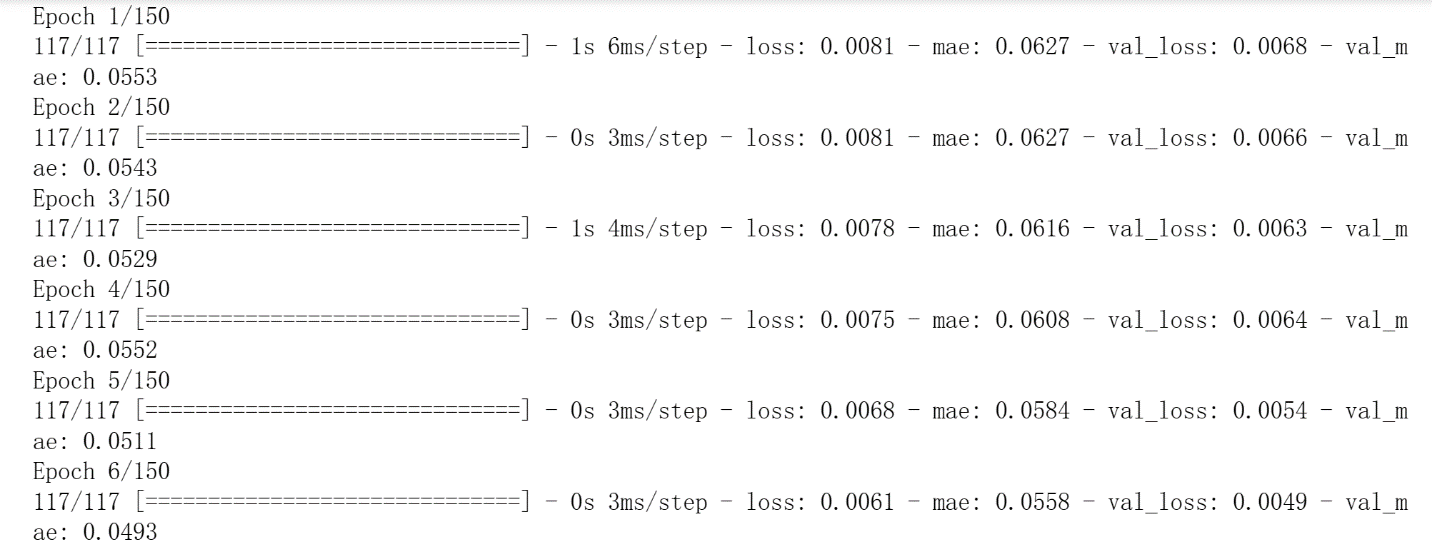


Figure Model Training Process-1

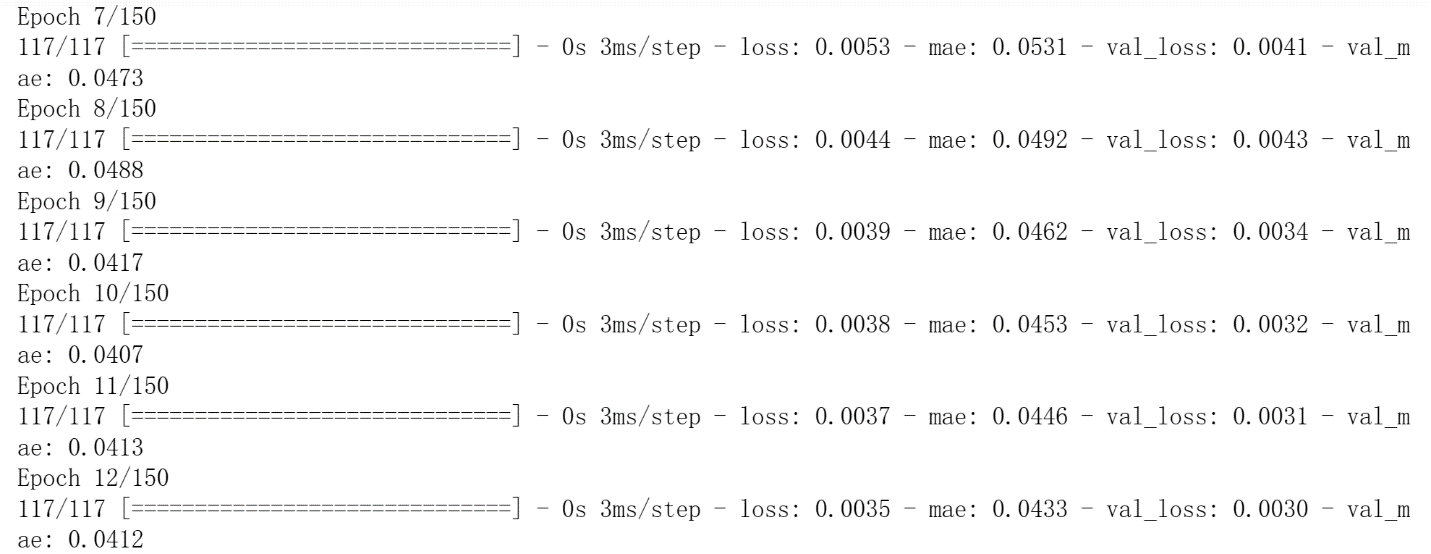


Figure Model Training Process-2

We can see from this process that the loss and mae are significantly reduced, which means that the effectiveness of physical model optimization is constantly improving.

#### 3. Model Evaluation

The next step would be evaluating the training process and the final accuracy. Overfitting may occur in this part, which can be observed through increasing training scores and decreasing test scores (diverging loss curves). The optimization effect can be seen more clearly from the figure below, which is the comparison between the loss and mae values as each epoch is carried out.

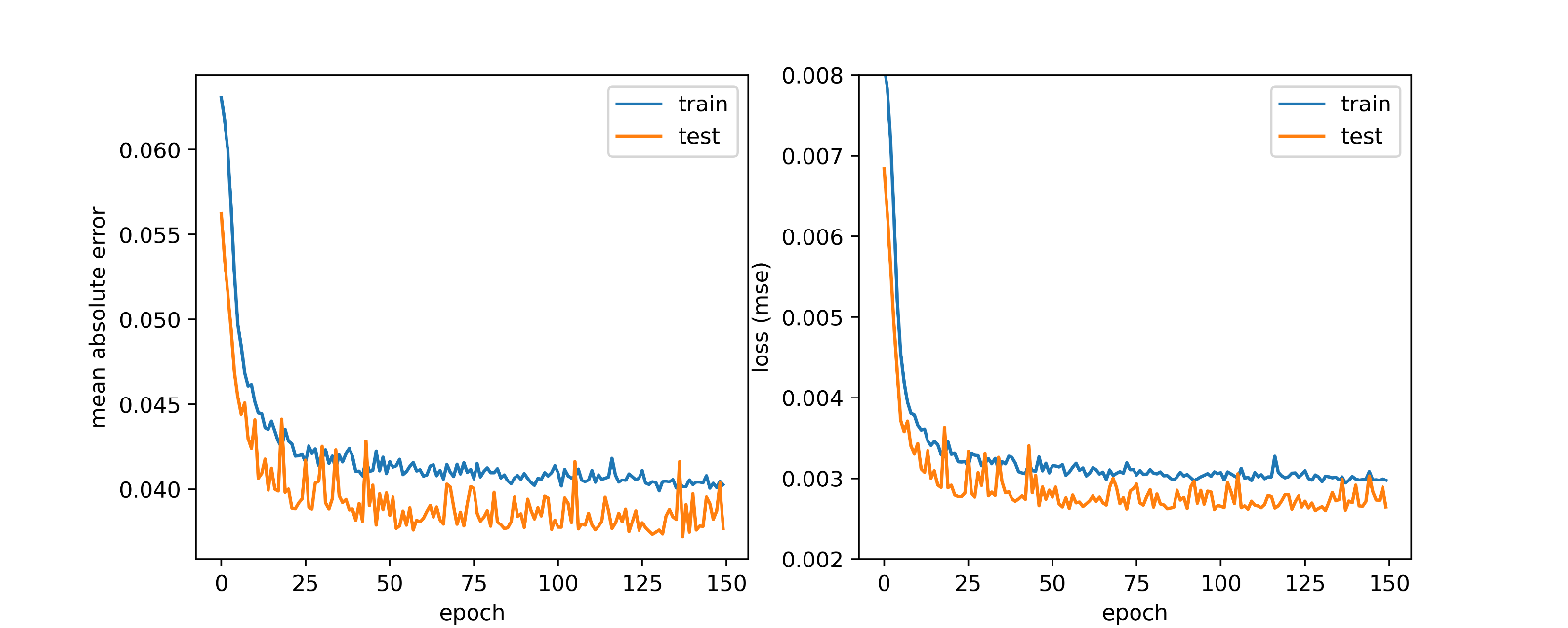


Figure Model Evaluation

At the same time, the mae value of the neural network is also obtained, which is 0.04. Till now, the final physical model is completed which is also called data-driven machine learning model.

### 4.3.5 Compilation of hybrid digital twin and Implementation

#### 1. Preconditions

Now we have all the ingredients ready for compiling the digital twin:

* physics-based simulation model
* data-driven machine learning model

The behavior prediction for a given initial condition X0 will be a three-step process:

run a time integration using the simulation model to obtain X𝑆

provide XS as input to the trained ML model to obtain the correction values X𝑁𝑁

obtain the final prediction by adding the previous results: X𝐷𝑇=X𝑆+X𝑁𝑁

#### 2. Digital Twin Model

So far, we have completed the establishment of the digital twin model, and the results are as follows.

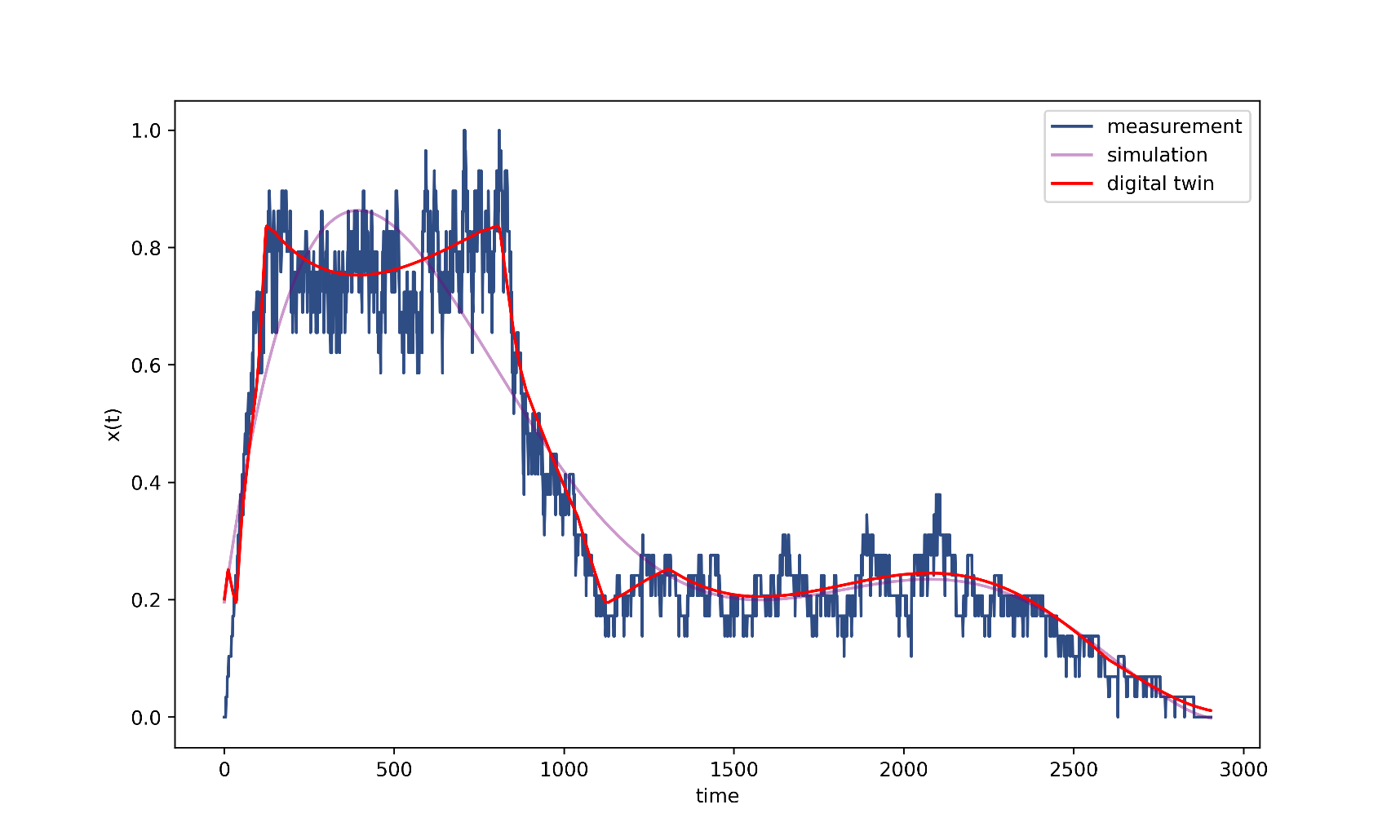


Figure Model fitting

From the figure, we can observe the fitting degree of the two models to the data, among which the digital twin model greatly improves the compatibility of the system prediction.

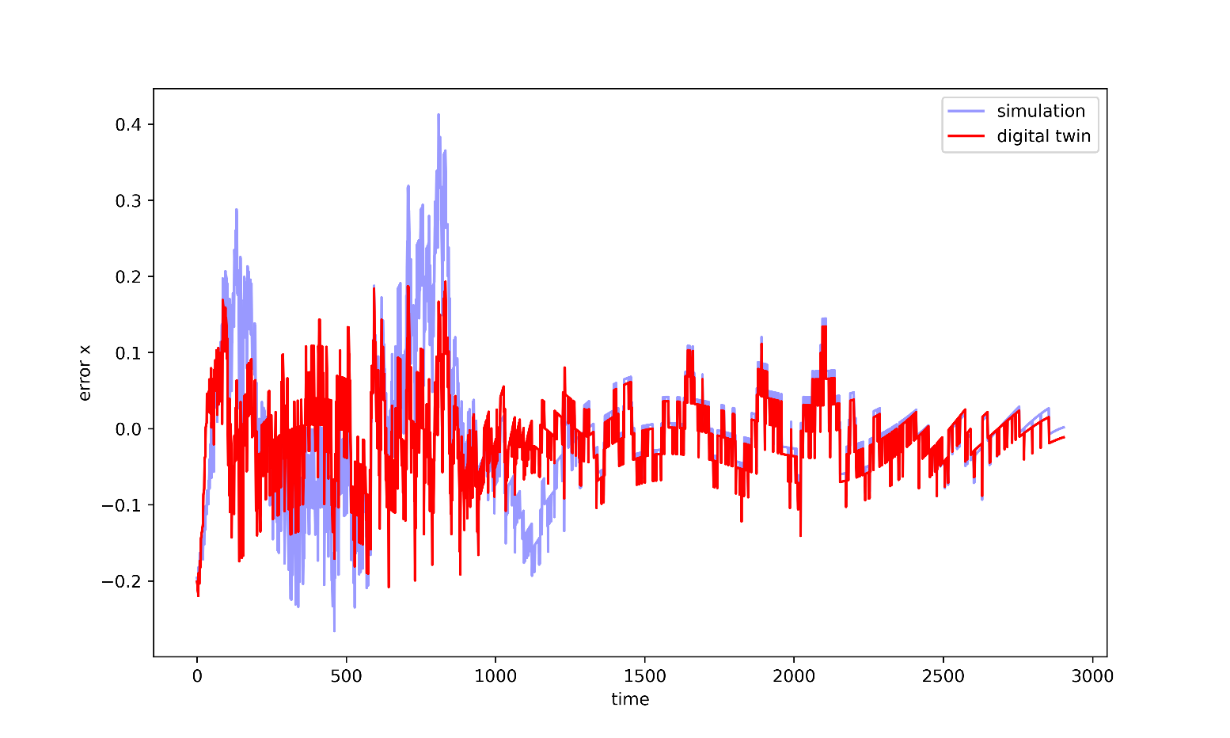


Figure Error value between simulation and initial values

If the system requires higher accuracy, it can be optimized by increasing the order of polynomial fitting and increasing the depth of the neural network. But this will increase the risk of overfitting, so compromise parameters should be selected to ensure both accuracy and fit.

#### 3. Application

This chapter introduces a digital twin scheduling model predicted from the perspective of the machine, which predicts the possible usage status of the machine. From this perspective, the production and operation status of the product can be monitored in real time, so as to realize the overall optimization of the system and improve the use efficiency of the machine. On the contrary, the digital twin model can also be optimized from the perspective of product injection system, and the degree of system blockage can be predicted through the time and quantity of product arrival, combined with the damage rate of the machine, so as to invest the required resources to avoid possible problems.

The application of the job shop scheduling model improves the production efficiency and diversity of small-batch products. The digital twin models provide a continuously optimized simulation model in the cloud area for dynamic adjustment. The machine learning-based digital twin technology further provides job shop scheduling models with analysis, re-scheduling and optimization for real-time response. The application of this technology further realizes the possibility of intelligent system production.

# CHAPTER 5 RESULTS

#### 1. Implementation of JSS

In this part, from 3 different scenarios I mainly build up sequences models to simulate the conditions of job shop scheduling. Based on the SimPy platform, the discrete event simulations can clearly show the operating processes. For example, when the number of products on this production line enter the system, when to start applying for machines, when to process them, the sequence of machines that have passed, and the time slice at each moment.

All data are randomly generated within a reasonable data range after investigation, thus they are highly representative.

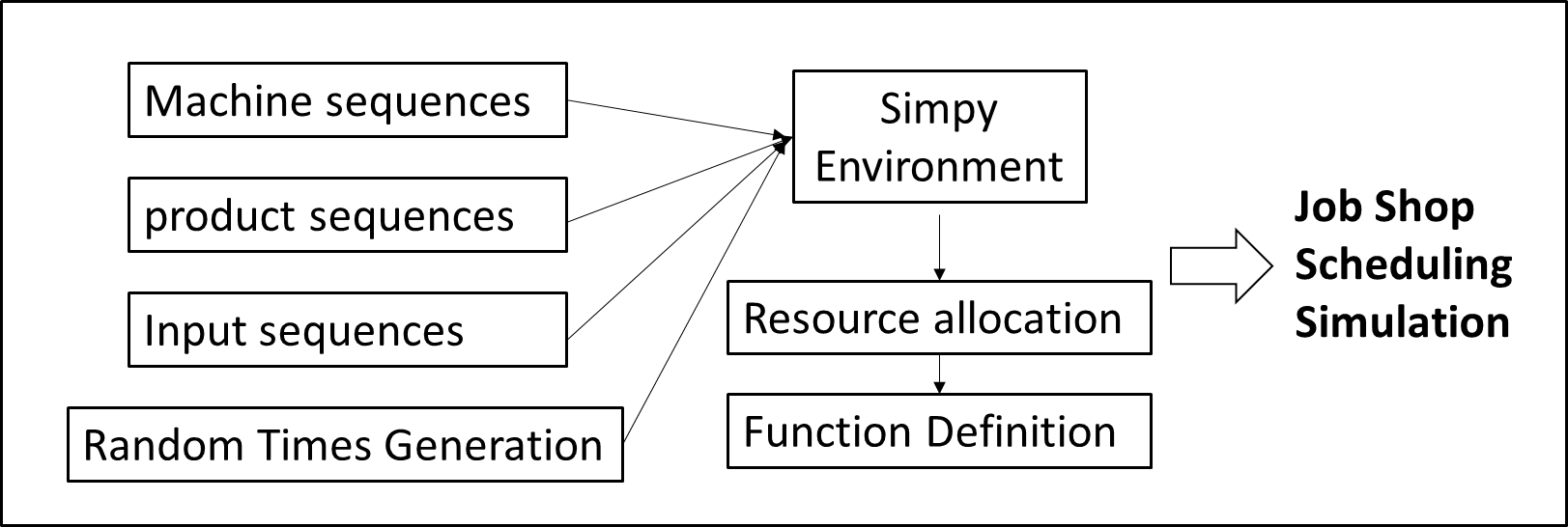


Figure Implementation of Job Shop Scheduling

In the whole process, I generate 13 types of machines, 21 types of products and 5 pipelines in total. The end time is 1071, the injection of high-dense products is the main reason for the blocking. m\_vacant and m\_occupied are the parameters to track the utilization of machines. I also define several arrays to track the status of the whole system at each time clip. The model of job shop scheduling is done in this section.

#### 2. Evaluation of the scheduling system

The blocking rate of machines and products are the main features to signify the efficiency of the system, and also help to do the optimization.

I plot figures about the numbers of machines that are occupied by products, the percentage of machine occupation and the number of blocked products at each time slice.

To promote the utilization of each machine, I add back-up machines based on the frequency of the usage of machines and end up with the histogram graph to make the comparison.

|  |  |
| --- | --- |
| Feature | Value |
| End Time | 1071 |
| Block Number | 83 |
| Max Block Number | 5 |
| Max Block Time | 203 |
| Average Block Number | 1.1805 |
| Average Block Rate | 0.0084 |

Table Evaluation Features

#### 3. Application of ML strategies in machine failure prediction

In order to make the simulated scenes more realistic, considering the actual operating state of the factory is necessary, in which the failure of the machines due to various reasons is one of the key factors affecting production efficiency.

I found a data set containing 10,000 data, and used machine learning methods to predict machine damage, decision tree, random forest and xgboost are compared, and finally applied xgboost to the simulation sequences.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision | Recall | Accuracy |
| Decision Tree | 0.84 | 0.58 | 0.62 |
| Random Forest | 0.89 | 0.79 | 0.83 |
| Xgboost | 0.92 | 0.77 | 0.98 |

Table Comparison between 3 ML strategies

Based on the data we obtained, the xgboost shows the most accurate results, so I use this method to do the predictions about machine failure-repair model.

After introducing the machine failure-repair model, the length of the pipeline for product processing has prolonged to 2854. The data of the production line is shown in the table below:

|  |  |
| --- | --- |
| Feature | Value |
| End Time | 2837 |
| Block Number | 803 |
| Max Block Number | 143 |
| Max Block Time | 713 |
| Average Block Number | 63 |
| Average Block Rate | 0.35 |

Table New Evaluation Features

#### 4. Optimization of JSS on DT-based Neural Network

After introducing the machine failure-repair model, the congestion of the system is becoming serious. In order to alleviate this situation in actual production, I introduced the digital twin model and realized dynamic scheduling with the help of the neural network model in ML planning.

Firstly, the polynomial curve is used to fit the utilization rate of the machines, then generating the basic physical model, and a hybrid digital twin model was established on the basis of the physical model by using a neural network to achieve efficient system optimization.

The details of the parameters are shown in the table below.

|  |  |
| --- | --- |
| Feature | Value |
| Polynomial Order | 5 |
| Epochs | 200 |
| Batch Size | 50 |
| MAE of the Physical Model | 0.0603 |
| MAE of Digital Twin Model | 0.0376 |

Table Digital Twin Features

The realization of the digital twin model provides another effective method for realizing production on the digital manufacturing platforms.

# CHAPTER 6 FUTURE WORK

#### 1. Digital twin model forecasting from a product perspective

At this stage, I have implemented the application of the digital twin model from the perspective of the machine. In the next stage, I will try to build a digital twin model from the view of product injection frequency and blocking conditions, and integrate the models from the two perspectives to achieve the best optimization results.

#### 2. Application of Muti-production lines

I simulated the models with a total of five production lines, and selected a specific line for the simulation. In the following work, all five assembly lines could be put into the simulation to achieve a digital twin application for a parallel system with multiple production lines. Finally, the number of production lines can be selected automatically for simulation, providing more flexibility to the simulation platform.

#### 3. Path planning with more complex methods

The job scheduling model was proposed to optimize the variety and uniqueness of production, and many algorithms for job scheduling have now been derived, of which I have selected the basic job scheduling model. Improving production efficiency can start with path planning itself, and the next stage can be the integration of more path planning algorithms, such as genetic and grey wolf optimizer algorithms. This can improve the efficiency of the system from the product itself.

# CHAPTER 7 CONCLUSION

The simulation of CA1 started with a system understanding of AGV and DMP, extensive literature reading and discrete time simulation using the simpy package under python.

In the first phase I focused on discrete-time simulation for scheduling problem. In the simpy environment, I simulated three different scenarios of the product production process, ranging from simple sequential arrangement to more complex out-of-order scheduling problems.

From scenario C, I simulated the system using job shop scheduling for product production. Then the performance of the scheduling system was evaluated using Cumulative Machine Utilization, Machine Occupancy and Product Blockage.

In the next phase, I introduce ML algorithms into the previously simulated sequences, firstly by finding the dataset for machine failure training, using decision trees, random forests and xgboost algorithms. By comparing the three methods, the xgboost algorithm is applied into the final machine failure model, and is merged into job shop scheduling system.

The final prediction of the machine's occupancy was based on a polynomial fit and hybird neural network, which produced very small errors by comparison. This highly accurate model interacts with the actual data and ultimately enables the application of a digital twin model, which plays a significant role in the allocation of productivity resources.

During this complete project, I gained a cutting-edge understanding of the macro concepts of industrial automation and profound insights into discrete event simulation, path scheduling, data processing of matrices and learned how ML methods can be applied to traditional techniques and how AGV-based scheduling can be innovatively optimized in these ways.

# REFERENCES

Bazaz, S. M., Lohtander, M., & Varis, J. (2019). 5-dimensional definition for a manufacturing digital twin. *Procedia Manufacturing*, *38*, 1705–1712.

Bloomberg, Jason. "Digitization, digitalization, and digital transformation: confuse them at your peril." *Forbes. Retrieved on August* 28 (2018): 2019.

Cheng, Runwei, Mitsuo Gen, and Yasuhiro Tsujimura. "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies." *Computers & Industrial Engineering* 36.2 (1999): 343-364.

Erol, Rizvan, et al. "A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems." *Applied soft computing* 12.6 (2012): 1720-1732.

Fox, Mark S., and Stephen F. Smith. "ISIS—*a knowledge‐based system for factory scheduling*." Expert systems 1.1 (1984): 25-49.

Geyik, Faruk, and Ismail Hakki Cedimoglu. "The strategies and parameters of tabu search for job-shop scheduling." *Journal of intelligent manufacturing* 15 (2004): 439-448.

Heger, J.; Voss, T. *Dynamic priority based dispatching of AGVs in flexible job shops.* In Proceedings of the 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gulf of Naples, Italy, 18–20 July 2018.

Jackson, James R. *Scheduling a production line to minimize maximum tardiness*. CALIFORNIA UNIV LOS ANGELES NUMERICAL ANALYSIS RESEARCH, 1955.

Johnson, Selmer Martin. "Optimal two‐and three‐stage production schedules with setup times included." *Naval research logistics quarterly* 1.1 (1954): 61-68.

Ju, Q. Y. (2007). *Planning and scheduling optimization of job-shop in intelligent manufacturing system* (pp. 1–97). Doctoral dissertation of Nanjing University of Aeronautics and Astronautics, Nanjing, China.

K. Zhu and Y. Zhang, "A cyber-physical production system framework of smart CNC machining monitoring system", *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 6, pp. 2579-2586, Dec. 2018.

Li, Yibing, et al. "Digital twin-based job shop anomaly detection and dynamic scheduling." *Robotics and Computer-Integrated Manufacturing* 79 (2023): 102443.

LIU, Min, Cheng WU, and Ying-jie YANG. "Genetic algorithm method based on combinatorial rules in identaical parallel machine scheduling problem." *ACTA ELECTONICA SINICA* 28.5 (2000): 52.

Mantel, Ronald J., and Henri RA Landeweerd. "Design and operational control of an AGV system." *International Journal of Production Economics* 41.1-3 (1995): 257-266.

Pandey, Vaishnavi, et al. "Accelerating the renewable energy sector through Industry 4.0: Optimization opportunities in the digital revolution." *International Journal of Innovation Studies* (2023).

R. Rosen, G. von Wichert, G. Lo and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing", *IFAC-PapersOnLine*, vol. 48, pp. 567-572, 2015.

Shao, G., & Helu, M. (2020). Framework for a digital twin in manufacturing: Scope and requirements. *Manufacturing Letters*, *24*, 105–107.

Waltz, Thomas J., et al. "Use of concept mapping to characterize relationships among implementation strategies and assess their feasibility and importance: results from the Expert Recommendations for Implementing Change (ERIC) study." Implementation Science 10 (2015): 1-8.

Wan-Liang, Wang, Xu Xin-Li, and Wu Qi-Di. "Hopfield neural networks approach for job shop scheduling problems." *Proceedings of the 2003 IEEE International Symposium on Intelligent Control.* IEEE, 2003.

Xiong, Hegen, et al. "A survey of job shop scheduling problem: The types and models." *Computers & Operations Research* (2022): 105731.

Yan-hai, Hu, et al. "Flow shop rescheduling problem under rush orders." *Journal of Zhejiang University-SCIENCE* A 6.10 (2005): 1040-1046.

Yao, Fengjia, et al. “Improving Just-in-Time Delivery Performance of IoT-Enabled Flexible Manufacturing Systems with AGV Based Material Transportation.” *Sensors*, vol. 20, no. 21, Nov. 2020, p. 6333.

Yu, Jian-jun, Xu-jun Xu, and Fei Ye. "Study on multi-objective flexible production scheduling based on improved immune algorithm." *2008 International Conference on Management Science and Engineering 15th Annual Conference Proceedings*. IEEE, 2008.

Zhang, Jian, et al. "Review of job shop scheduling research and its new perspectives under Industry 4.0." *Journal of Intelligent Manufacturing 30* (2019): 1809-1830.

Zhang, Meng, Fei Tao, and A. Y. C. Nee. "Digital twin enhanced dynamic job-shop scheduling*." Journal of Manufacturing Systems* 58 (2021): 146-156.

Zhang, W., et al. "Multi-objective scheduling simulation of flexible job-shop based on multi-population genetic algorithm." *International Journal of Simulation Modelling* 16.2 (2017): 313-321.

Zhao, Yangming, et al. "Minimize the make-span of batched requests for FPGA pooling in cloud computing." *IEEE Transactions on Parallel and Distributed Systems* 29.11 (2018): 2514-2527.