10-4-2023

Integrated CAPP and Scheduling using a Combined ML and Optimization Approach for Smart Manufacturing

Syeda Marzia
University of Windsor

Integrated CAPP and Scheduling using a Combined ML and Optimization Approach for Smart Manufacturing

By

Syeda Marzia

A Thesis

Submitted to the Faculty of Graduate Studies

through the Industrial Engineering Graduate Program

in Partial Fulfillment of the Requirements for

the Degree of Master of Applied Science

at the University of Windsor

Windsor, Ontario, Canada

2023

Integrated CAPP and Scheduling using a Combined ML and Optimization Approach for Smart Manufacturing

by

Syeda Marzia

APPROVED BY:

A. Rahimi

Department of Mechanical, Automotive and Materials Engineering

---

A. Khan

Department of Mechanical, Automotive and Materials Engineering

---

A. Azab, Advisor

Department of Mechanical, Automotive and Materials Engineering

September 26, 2023

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates materials that are the result of joint research of the author and her supervisor, Prof. Ahmed Azab. Chapter 2,3, and 4 of these theses were co-authored with Prof. Ahmed Azab, and Alejandro Vital-Soto. In all cases, the key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by the author; Prof. Ahmed Azab and Alejandro Vital-Soto. Prof. Ahmed Azab and Alejandro Vital-Soto provided feedback the on refinement of ideas, overall coordination, improvements and editing of the manuscript.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes 1 conference paper that has been previously submitted, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 1 and 2	Marzia, S., Vital-Soto, A., Azab, A., "Automated Process Planning and Dynamic Scheduling for Smart Manufacturing: A Systematic Literature Review"	Accepted

	North American Manufacturing Research Conference (NAMRC) 51	

I certify that I have obtained written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

In the era of digitalization, manufacturing industries are transitioning to Smart Manufacturing (SM) to meet customized customer demands. However, the integration of CAPP and scheduling (ICAPPS) remains a challenge due to conflicting objectives. Most of the literature in existence does not consider the IPPS problem in real-world or dynamic multi-part and multi-machine scenarios and cannot address the sequencing objectives. In this research we propose a machine learning-optimization model to assign and sequence setups in a dynamic flexible job shop environment, considering real-time disruptions like machine breakdowns. The research aims to bridge the gap between process planning and scheduling by treating setups as dispatching units, minimizing makespan, and enhancing manufacturing flexibility. The Dynamic Flexible Job Shop Problem (DFJSP) is solved through a comprehensive methodology that encompasses solving with mathematical programming, heuristics, and creation of a robust dataset for data mining by extracting attributes reflecting priority relationships among setups. The empirical findings demonstrate the effectiveness of the proposed methodology, with the mining model outperforming classical dispatching rules. Furthermore, the model exhibits robust generalization capabilities. This research contributes valuable insights into addressing the complex CAPP and scheduling problems for smart manufacturing environments.

DEDICATION

In the name of Allah, the most gracious, the most merciful

I dedicate this work to my parents, Mohammad Shahid Hossain, Munira Hasnain and my life partner Md Sadman Sakib. Their unwavering backing and unwavering motivation have been the cornerstone of my journey, for which I am profoundly appreciative.

ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Prof. Ahmed Azab, for allowing me to work on this research. Furthermore, I would like to thank Prof. Ahmed Azab and Alejandro Vital-Soto for guidance, suggestions, and continuous support. I would also like to acknowledge my committee members, Afshin Rahimi, and S. Asif Khan, for providing invaluable constructive criticisms.

This work has been funded by NSERC - Natural Sciences and Engineering Research Council, Canada.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

During the era of digitalization, all the manufacturing industries have been experiencing a rapid transition from the traditional manufacturing process to the Smart Manufacturing (SM) process for the past thirty years. Nowadays customer demand has become more customized than before.–Manufacturing industries are experiencing extraordinary challenges to perform production planning and scheduling in a real-time and flexible manner to cope with the customization requirements. In response, the manufacturing system needs to autonomously reconfigure the process plan and adapt the production schedule according to changing manufacturing environments.

Although the two key functions of SM, Process planning and scheduling (PPS) are connected, their objectives usually do not agree. As a result, PPS solutions often cannot cope with dynamic production requirements when these two functions are solved separately, however, the combination of these two functions can suppress the limitation and maximize the strength.

In this research, we aim to tackle a dynamic Flexible Job Shop Scheduling problem (DFJSSP) in conjunction with the Setup Planning to solve the integrated process planning and scheduling problem. This research will allocate and sequence the setups on compatible machines under dynamic setting, such as the random machine breakdown.

1.2 The Computer Aided Process Planning (CAPP) and Scheduling Problem

Process Planning (PP) is an essential decision-making process linking the design and manufacturing of a part. PP can be referred to as the handbook for a part as it

encompasses the selection of raw materials and specific processes, machine and cutting tool, cutting condition, and sequence of operations to transform the designed part into a desirable final product. The quality of the process plan enormously impacts the manufacturing process's efficiency and product final quality. Traditional PP largely depends on the knowledge and experience of human experts, potentially leading to inefficient decision-making and non-optimal solutions [1]. This approach also suffers from being nongeneralizable and cannot fulfill mass customization requirements, which requires manufacturing flexibility [2]. Due to the capability of computers to aid planning activities with increased speed and accuracy, the Computer-Aided Process Planning (CAPP) method has been gaining popularity among researchers [3]. Most CAPP system uses either the variant approach (retrieval of the existing plan and modification) or the generative approach (developing a plan based on part geometry) to generate the process plan [4]. Despite the efforts, few CAPP systems can significantly improve manufacturing due to the high complexity and dynamic aspect of process planning [3].

Although PP and scheduling are two separate activities in manufacturing, both functions are closely related. Scheduling deals with allocating manufacturing processes to manufacturing resources over a specific time interval. The scheduling function depends on the job arrival pattern, operation precedence relation, and the number of available resources and determines the most suitable time to execute an operation on a machine tool. In summary, scheduling is an optimization problem where the objective is to manufacture final products in the shortest possible time considering resource capacity limitations [5]. Although PP can also be considered a manufacturing resource management function, the objective of PP and scheduling are not compatible and are usually in conflict. Where scheduling usually considers manufacturing resources with

time-based objectives, PP mainly focuses on minimizing manufacturing cost and product quality objectives. Traditionally Process Planning and Scheduling (PPS) are done sequentially; scheduling is done after PP. This approach has some significant drawbacks. According to Li et. al. [6], the process planner creates a process plan for individual jobs within the sequential approach. The capacity limitation of resources and uncertain events, such as delays, urgent orders, and machine breakdowns, are not considered in this stage. During the scheduling phase, this fixed process plan often becomes infeasible due to the dynamic changes in the production floor. Thus, it is crucial to study the overlap between the PP and scheduling objectives to handle this kind of disruption of the production floor.

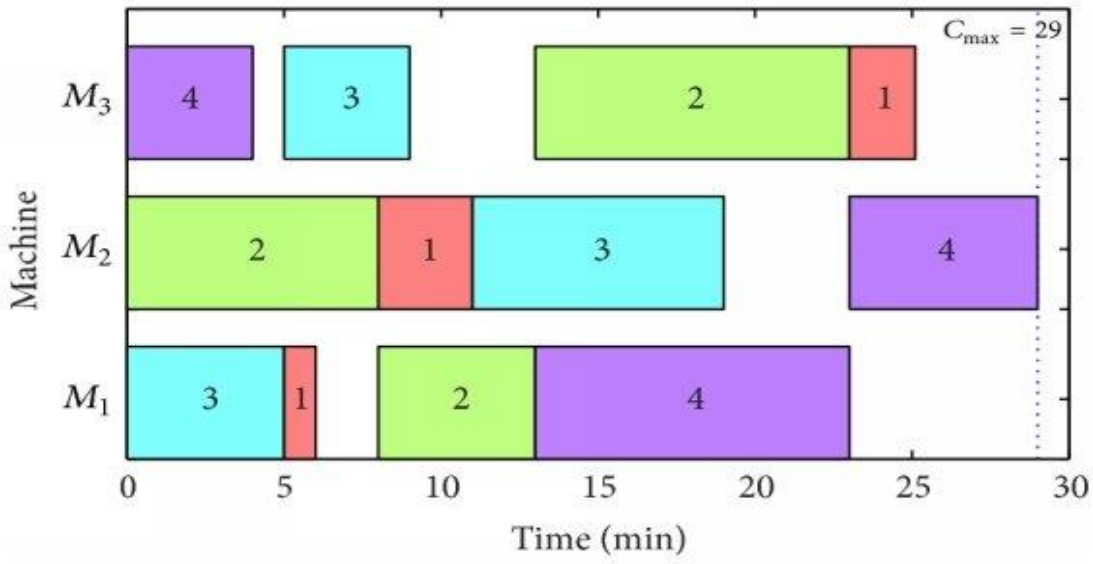

Figure 1. 1 The schedule of 4 × 3 job shop scheduling problem

1.3 Problem Definition

This integrated CAPP and Scheduling problem deals with a situation where there are J jobs that need to be completed using m machines. Each job is composed of a certain number of setups (nj). This problem can be modeled as a Flexible Job Shob Scheduling Problem (FJSP).

Thus, to solve the problem, first each setup needs to be assigned to one specific machine. Secondly, the sequence of the setups needs to be determined following the natural logical sequence of setups within each job. The total number and sequence of setups in each job is determined during the setup planning stage. This problem considers minimum makespan as optimization objective.

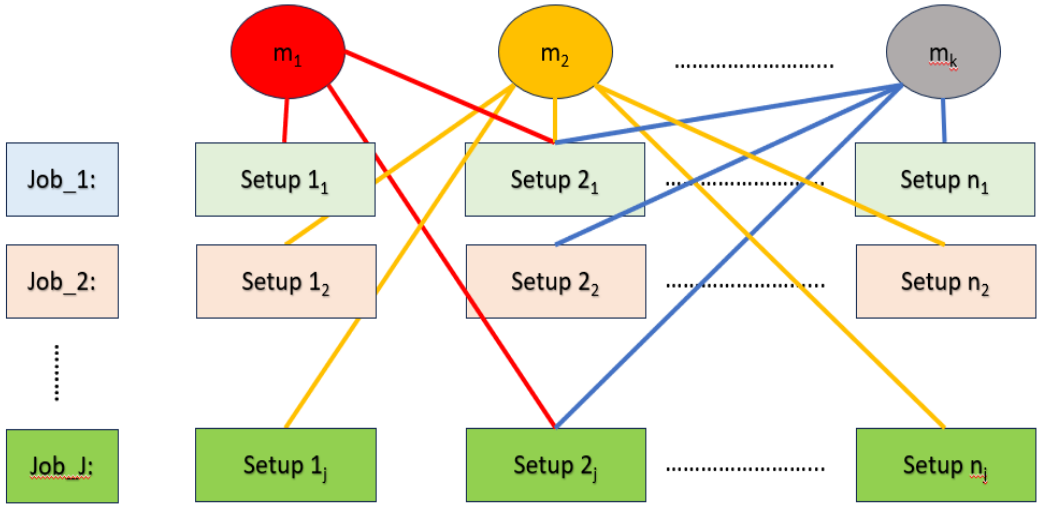

Figure 1. 2 Schematic view of the problem

1.4 Research Question

“How can the Integrated CAPP and Scheduling problem be effectively optimized for SM, by considering setups as dispatching units, allocating setups to machines, and sequencing with the aim of minimizing make span while following the natural logical sequence of setups within the job?"

1.5 Objective of Research

This research aims to develop an ML-Optimization model to solve the integrated CAPP and scheduling problem. The objectives of this research encompass the following key areas:

Literature Review: Conduct an extensive survey of existing literature pertaining to Integrated Process Planning and Scheduling (IPPS) approaches to establish a comprehensive understanding of the field.

Initial Schedule Generation: Create an initial nominal schedule by solving the machine assignment/routing problem and addressing the sequencing problem, laying the foundation for subsequent analyses and optimizations.

Linear Programming Model Development: Formulate a mixed-integer linear programming (MILP) model designed to address the sequencing problem

Machine Learning Model Construction: Develop a machine learning model tailored for the extraction of dispatching rules, leveraging data-driven insights to enhance scheduling efficiency.

Case Study: Undertake case studies employing the developed model to assess its practicality and effectiveness in real-world scenarios, thereby validating its feasibility.

Rescheduling for Uncertain Events: Utilize the developed model to perform rescheduling in response to unpredictable events, showcasing its adaptability and reliability in dynamic manufacturing environments.

1.6 Outline of the thesis

This section highlights the overview of the upcoming chapters of the thesis. In the introductory section, an overview of the research area and its significance has been provided, followed by the presentation of the research problem statement, objectives, and research questions. In Chapter 2, the foundational concepts and theoretical frameworks underpinning the research have been delved into, with a review of relevant literature on the topic, including key theories, methodologies, and previous studies. The research methodology employed in this study has been discussed in Chapter 3,

including the research design, data collection methods, and data analysis techniques, with particular attention given to the rationale behind method selection. Chapter 4 has presented the findings, analyzing the collected ata and providing insights into the research questions and objectives. The final chapter, Chapter 5, has summarized key findings, restated research objectives, and addressed research questions, along with discussing practical implications and offering recommendations for future studies or practical applications. Appendices contain supplementary material, and the references section has provided a comprehensive list of all sources cited throughout the thes

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In this chapter, the foundational concepts and theoretical frameworks that underpin this research is delved into. Relevant literature on the topic, discussing key theories, methodologies, and previous studies is reviewed. This chapter provides the necessary context for understanding the research problem and its significance.

2.2 Setup Planning

Setup planning, a crucial and complex task within CAPP, focuses on defining the guidelines for setting up the workpiece to be machined. This process significantly impacts manufacturability, production time, and costs, while also playing a vital role in the integration of CAD/CAPP/CAM/CNC and contributing to the evolution of intelligent manufacturing (Y. Zhang et al., 2022). ElMaraghy (Elmaraghy et al., 1993) defined Process Planning (PP) divided into two levels: Macro-level and Micro-level planning. The Macro-level focuses on identifying main tasks, their sequence, and suitable manufacturing processes. Micro-level planning provides detailed information about process parameters, tools, setups, time, and resources. Wu [10] defined CAPP as combination of tasks involving translating a part's geometric model into machining features, determining suitable machining resources and operations, and selecting the most cost-effective setup plan and operation sequence considering design and manufacturing constraints. Thus, setup planning can be referred to as a macro-level planning function of process planning.

Setup planning involves repositioning a workpiece on a specific machine's fixture to achieve the machining of highest feasible number of operations in a single setup. Setup

planning can divide into three sub-tasks: setup generation, operation sequencing and setup sequencing (Ming et al. 2000; Joshi et al., 2008) Setup generation involves classifying operations based on their Tool Approach Direction (TAD) and machine tools. The goal is to group operations together in a way that allows them to be machined without the need for setup or machine changes. The second step is operation sequencing which involves arranging operations that use the same cutting tools together to minimize the number of tool changes. In the third step, the overall sequence of is setup is determined by sequencing the setups [13].

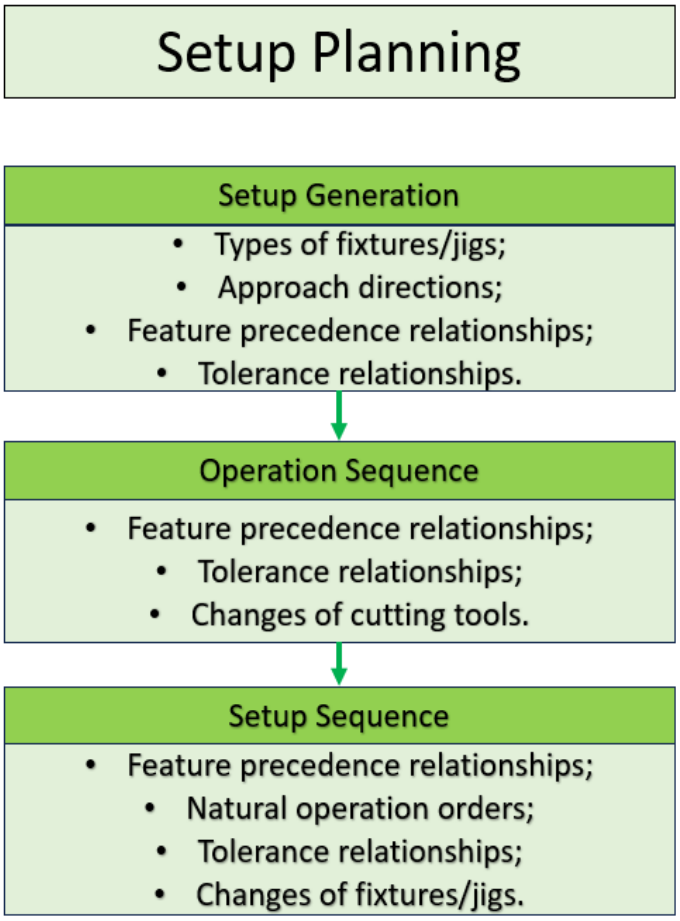

Figure 2. 1 Schematic view of setup planning procedure

2.3 Significance of Using Setup Planning for Integrated Process Planning and Scheduling (IPPS) Problem

Wu [10] defined CAPP as combination of tasks involving translating a part's geometric model into machining features, determining suitable machining resources and operations, and selecting the most cost-effective setup plan and operation sequence considering design and manufacturing constraints. In the majority of research papers addressing the IPPS problem, it is typically dissected into three subproblems: (i) the selection of process plans, (ii) the allocation of machines, and (iii) scheduling [14]. The conventional approach to addressing this problem involves firstly, choosing the process plan, followed by the subsequent allocation and scheduling of operations [14], [15].

All of these approaches consider operations as the dispatching unit. Operation sequencing is a common problem for both process planning and scheduling function. For PP, operations of a job is sequenced with objective such as minimize machining cost [16]. In the case of scheduling, the operations are sequenced to complete the jobs in the shortest possible time [17]–[20]. This creates conflict between the objective of process planning and scheduling. Process planning might involve trade-offs between cost and other factors like quality, production time, or resource utilization. to situations where machines are frequently set up or reconfigured for different jobs, which might not be the fastest manufacturing approach. For example, using a slower machine that consumes less energy and produces might be cost-effective but increase production time. On the contrary Scheduling decisions often prioritize time over cost.

Now, setup planning can play a crucial role in bridging the gap in this conflict. [21]. Setup planning is a pivotal task within CAPP that guides workpiece setup, influencing manufacturability, production efficiency, costs, and the integration of CAD/CAPP/CAM/CNC, thus advancing intelligent manufacturing (Y. Zhang et al.,

2022). It divide into three sub-tasks: setup generation by grouping manufacturing operations, operation sequencing within setup and setup sequencing (Ming et al. 2000; Joshi et al., 2008). Many works dedicated to the IPPS problem acknowledged the importance of setup planning for the integration of CAPP and Scheduling function. For instance, Mohapatra et al. [16], [22]–[24] proposed adaptive setup grouping strategies for minimization of cost and makespan and maximization of machine utilization for alternative machine (3 axis, 4 axis, 5 axis etc.) for a single part. These researchers have focused on grouping operations for a workpiece and assigning each setup on suitable machines, following a cross-machine setup approach. However, they neglected the importance of addressing the true integration of the process planning and scheduling problem, which should involve the consideration of n parts to be processed on m machines. To solve the issue, Haddadzade et al. [21] proposed a cross machine setup planning approach for multiple part and grouped operations simultaneously targeting various objectives. Although, this research does not consider the routing and sequencing task of the problem.

Furthermore, now there is an increase in no of Adaptive Setup Planning (ASP) studies focusing on generating machine specific setups upon request from dynamic schedule [25]. Such ASP approach can adapt to unforeseen events, such as changes in machine availability, fixtures, and tools, and significantly decrease the time required for re-planning and re-scheduling. Thus, it is necessary to consider setups as dispatching unit for scheduling instead of operations. Cai's research reinforced use of setup as the dispatching and scheduling unit of machining.

Table 2. 1 Synthesis of IPPS Research

Ref.	j x m	SP			Routing		Sequencing		uncertainty		Obj.	
		O grouping	O sequencing	S sequencing	O	S	O	S	Static	Dynamic	PP	Scheduling
[26]	1 x m				√					√	tol, SF	MFT, # J_{tk}
[27]	1 x m	√				√				√	P_s, #S	
[25]	1 x m	√				√				√	C, M_{utl}	C_{max}
[28]	1 x m	√				√				√	C, M_{utl}	C_{max}
[29]	1 x m	√				√				√	C, M_{utl}	C_{max}
[22]	1 x m	√			√				√		C, M_{utl}	C_{max}
[23]	1 x m	√			√				√		M_{utl}	C_{max}
[16]	1 x m	√			√				√		C, M_{utl}	C_{max}
[21]	n x m	√			√				√		C, M_{utl}, #S	C_{max}
[14]	n x m				√		√		√			C_{max}
[15]	n x m				√		√		√			C_{max}
[12]		√	√	√							#S	
[11]		√	√	√							#S	
[30]	1 x m	√	√	√	√							
[10]	1 x m	√	√		√						C, E_s	
[13]	1 x 1 (reconfigurable)	√			√						C	
[31]			√								C	
[32]		√	√							√		
[33]		√	√	√							#S	
[34]	1 x m	√	√	√		√			√		C	
[35]		√	√	√								
[36]	1 x m	√	√	√		√					C	
[7]		√		√							C	
Current Work	**n x m**	√	√	√		√		√		√	**#S**	**C_{max}**

(j = job, m = machine, O = operation, S= setup, #S = no of setup, C= cost, M_{utl}= machine utilization, tol= tolerance, SF= surface finish, P_s= Part stability, MFT= mean flow time, # J_{tk} = no of tardy job, C_{max}= makespan, E_s= stacking error)

From the literature review (Table 2.1), it becomes apparent that most of the previous research has primarily concentrated on addressing the process plan selection and

routing problem under static conditions. While there have been some studies that have demonstrated the potential to adapt setup plans to changing shop floor conditions, they have not effectively tackled the sequencing problem within dynamic scenarios.

However, static scheduling becomes outdated when unforeseen events occur on the shop floor due to unrealistic assumptions considered during their creation. Liu et. al. [37] points out in their review that deterministic scheduling assumptions, like known and fixed processing times and absence of machine failures, render these static schedules impractical in real-world situations. As Industry 4.0 continues to evolve, the production system is gaining enhanced flexibility, this progress comes hand in hand with added intricacies in production scheduling. Manufacturing systems inevitably face unpredictable disruptions, causing changes in planned activities due to factors such as resource availability shifts, order arrivals or cancellations, and longer processing times. Consequently, there arises a necessity for scheduling mechanisms to swiftly adapt to these potential disruptions and efficiently re-optimize the operational sequences in real-time [38].

Therefore, this research takes a novel approach by treating the setups for each job or workpiece as the dispatching and scheduling unit. The objective is to encompass the process planning problem within the dynamic scheduling framework. This innovative approach allows for the development of a one-shot solution method for the integrated Computer-Aided Process Planning (CAPP) and scheduling problem. Furthermore, it facilitates the reconfigurability of the process plan, as highlighted by Azab and ElMaraghy in 2007 [39].

2.4 Dynamic Scheduling for Smart Manufacturing

The challenge of managing schedules while accounting for real-time events is referred to as dynamic scheduling. Research has developed into dynamic scheduling to address

real-time disruptions, treating it as a series of static scheduling problems that require periodic revision or updates triggered by real-time events. Methodology of Dynamic scheduling can be grouped into proactive-reactive and predictive-reactive approaches [38], [40]. The aim of the predictive-reactive approach is to develop a preliminary schedule that seeks to mitigate the effects of uncertain events on overall system performance [41]. For adjusting the preliminary schedule or reschedule, we need to answer two questions: when and how to react to uncertain events. Three policies: periodic, event-driven, and hybrid rescheduling, are proposed in different literatures for when to reschedule and two strategies of how to reschedule: schedule repair and complete rescheduling can be found in literature [40].

Existing scheduling methodologies can be grouped into three categories: exact approaches, meta-heuristic algorithms, and heuristic approaches (Priore et al., 2014; L. Zhang et al., 2022). Exact approaches based on mathematical modelling have been used to ensure better performance than other heuristic methods in terms of finding optimal solutions. Approaches such as mixed-integer linear programming, branch and bound can find the optimal solutions for small or mid-size scheduling problems [43]. However, they are computationally inefficient for large-scale problems because they cannot solve the problems in polynomial times [43]. Metaheuristics [e.g., simulated annealing (SA), tabu search, genetic algorithms (GAs)] are widely applied to solve large scheduling problems [40]. However, Meta-heuristic algorithms are time-consuming, and their performance can even vary dramatically among different problems, especially for solving dynamic or online scheduling problems. Shahzad and Mebarki stated in their work that, although metaheuristics have an advantage over heuristics such as dispatching rules in terms of solution quality and robustness; nevertheless, these are usually more difficult to implement and tune, and are

computationally too complex to be applied in a real-time system (Shahzad & Mebarki, 2012). Ouelhadj and Petrovic [41], reported in their work that there is hardly any research work has addressed the use of metaheuristics in dynamic scheduling.

Now, in literature a common and popular way of dynamically schedule jobs is by implementing dispatching rules. Dispatching rules are efficient, simple, and capable of instantly solving scheduling problems by assigning a priority for every job in the waiting queue and are frequently used in practice due to their ease of implementation and quick computation time [37], [44]–[46]. However, as dispatching rules are traditionally derived by empirical or analytical studies, the performance of these rules depends on the state the system is in at each moment [40]. To resolve this limitation and boost their effectiveness/performance, machine learning algorithms appearing as a promising solution [38], [40]. Among the two approaches of dynamic scheduling, knowledge based system is capable of extracting implicit knowledge from earlier system simulation to determine best dispatching rule for each possible system state.

The main algorithm types in the field of dispatching rule development are case-based reasoning (CBR), neural networks, inductive learning, and reinforcement learning. The Inductive Learning Algorithm (ILA) is an iterative and inductive machine learning approach employed to generate a set of classification rules, typically presented in the "IF-THEN" format, based on a given set of examples. This algorithm progressively refines its rule set through successive iterations, appending newly generated rules to the existing set. Shahzad et. al. [47] proposed an hybrid simulation-optimization-data mining approach to generate JSP solutions by tabu search and identified dominance relationship between competing jobs with predefined attributes. A decision tree is subsequently employed to efficiently dispatch jobs in a real-time. Anran et. al. [48]constructed a data mining dynamic scheduling model to assign DR’s from DR

library to scheduling subproblem in real time. Gokhan et. al. [49]also developed a decision tree learning model to select dispatch job in real time. Mohammad [50] developed GA-datamining approach to automatically assign different dispatching rules to machines based on the jobs in the queues. This work tried to address the dominance or priority of different jobs. Li and Olfasan [51] is one of the pioneers for developing data mining-based approach for discovering new dispatching rule for operation sequencing of multiple jobs. They used a decision tree to discover key scheduling decisions from production data. Liping et. al. [42]Investigated new dispatching rule for operation sequencing development through the optimization of scheduling, as well as data transformation and data mining through hybrid GA-random forest algorithm. Sungbum et. al. [43] also took a similar approach for developing operation assignment rule and sequencing rule with random forest. From this it becomes evident that, developing a dispatching rule mining system for dynamic setup sequencing can be beneficial for addressing the current gap in the integrated CAPP and Scheduling problem. Thus, This study adopts a predictive-reactive approach to effectively sequence setups on the shop floor. Through the integration of machine learning and optimization within a unified framework, the schedule can be dynamically adjusted in response to these disruptions, all while ensuring that the fundamental objectives of the Integrated Computer-Aided Process Planning (CAPP) and Scheduling problem remain unviolated.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter outlines the research methodology employed in this study. The research design, data collection methods, and data analysis techniques are described. Special attention is given to the rationale behind method selection.

3.2 Framework

We introduce a novel approach that combines machine learning (data mining) and optimization techniques for addressing the integrated CAPP and Scheduling problem. The primary objective of this approach is to create a set of rules for guiding dispatching decisions to sequence setups within a flexible job shop scheduling environment. Thus, initial nominal solutions for small problem instances are generated as sources of learning rules for scheduling. Once the solutions have been obtained, they are transformed into learning data by constructing new attributes. In this research, the term ‘attributes’ refers to the set of all data related to the scheduling decisions. In the proposed approach firstly, setups are assigned to available machines on the shop floor. Secondly, setups are sequenced on assigned machine by learning the best dispatching rule through a ML-Optimization model. Finally, considering an event of random machine breakdown, the initial schedule is adjusted by re-assigning disrupted setups on the new available machine and sequenced utilizing the mined dispatching rule.

In the proposed approach,

- Initially, a simulation module generates a series of problem instances that are relevant to real-world scheduling systems. Alternatively, historical data from the

manufacturing system can be used in place of this. These problem instances are then stored in an instance database.

- Subsequently, the optimization module generates solutions for a subset of these instances, from which the initial training dataset is created. These solutions represent a collection of well-informed scheduling decisions that could potentially benefit the manufacturing system. These scheduling decisions form valuable scheduling knowledge, which is stored in a scheduling database and utilized by a learning process to construct a decision tree. This decision tree is then used for generating dispatching rule of the setups. Importantly, it is a dynamic sequencing model which can be updated with the change in resource.

Figure 3.1 illustrates the framework of the dispatching rule mining approach for sequencing the setups. Later, the generated rule can also be used to dynamically sequence disrupted setups as needed.

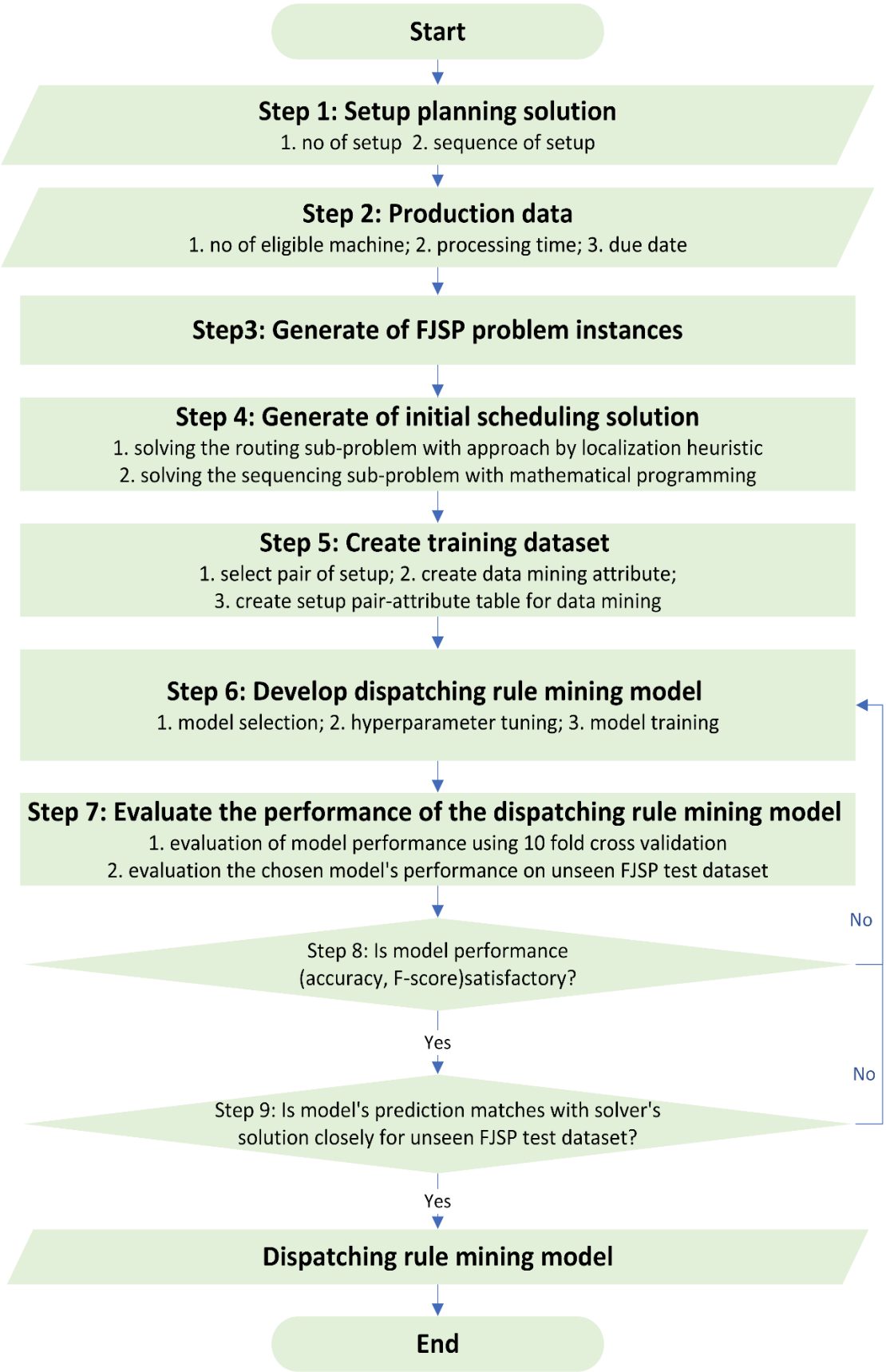

Figure 3. 1 Rule mining procedure for initial nominal schedule

3.3 Generation of Problem Instances

A simulation module is used to generate the relevant scheduling problem instances. In our experiments, we created 3 sets of similarly sized static FJSP instances: FJSP_5, which consists of 5 jobs and 3 machines. These specific problem instances were generated randomly, following the parameters outlined in the methodology introduced by Jun, 2019.[43]. All jobs are assumed to be available simultaneously at time zero. Discrete uniform distribution between 10 and 50 is used to generate the operation processing times.

The due date of each job was specified by a date tightness parameter as in Tay and Ho (2008). The due date formula is stated below:

$$di = c * \sum_{j=1}^{ni} P\_bar\_ij$$

where,

c = tightness factor of the due date

ni = number of operations of job i

Table 3. 1 Considered parameters for case study

Parameters	FJSP_5
no of jobs	5
range setups per job	2-3
no of machines	3
min no of equivalent machine per setup (flexibility:f)	2
range of processing time per setup (hours)	10-50
Tightness factor of due date	0.8-1.2

Table 3. 2 FJSP problem instance with 5 job and 3 machines

Job_id	Setup_id	M0	M1	M2	Job_due
0	0	23	12		49.2
0	1		21	28	49.2
0	2		27	12	49.2
1	0	31	28		44.4
1	1		23	29	44.4
2	0	16		18	45.2
2	1	49		30	45.2
3	0	41	18	14	48
3	1	19	20	26	48
3	2	19	12	11	48
4	0	38		18	42.8
4	1		21	30	42.8

3.4 Solving the FJSP

An FJSP can be divided into two sub-problems, a routing problem, and a sequencing problem. The routing sub-problem involves assigning each operation to a suitable machine, while the scheduling sub-problem focuses on determining the order in which operations should be performed while considering precedence constraints. The sequencing problem is for sequencing assigned operations to machines and is equivalent to the classical job shop scheduling problem. These two sub-problems have been shown to be NP-hard [43].

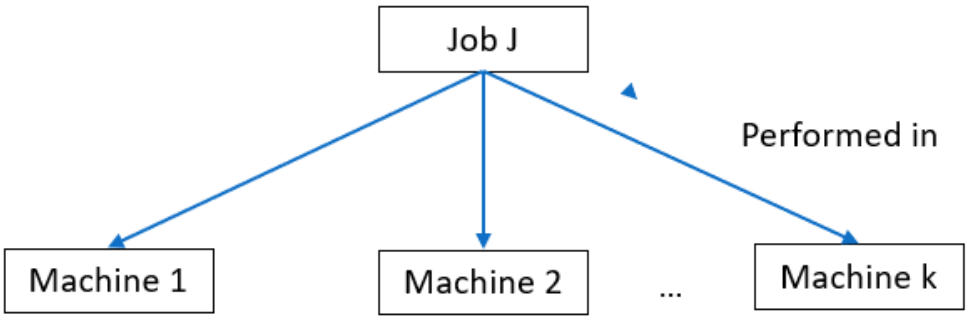

Figure 3. 2 Routing flexibility

The Flexible Job Shop Problem (FJSP) can be approached using two main strategies: concurrent approaches and hierarchical approaches. Hierarchical approaches provide a structured method by handling assignment and sequencing decisions independently, thus reducing the complexity of the problem.

In this research, a hierarchical methodology is employed to address the research problem. Specifically, a rule-based algorithm is adopted to tackle the routing problem, thereby transforming the initial problem into a form that can be effectively analysed and compared with a classical job shop sequencing problem.

3.4.1 Solving the Routing Sub-Problem / Machine Assignment

The routing sub-problem is a crucial aspect of production scheduling and involves the assignment of each operation or task to a suitable machine or workstation. This is a fundamental step in optimizing the production process, as it determines the sequence in which tasks are executed and the allocation of resources.

The goal of solving the routing sub-problem is to minimize production costs, maximize efficiency and utilization, reduce makespan or achieve other specific objectives depending on the manufacturing environment and requirements. Various algorithms and techniques, such as mathematical optimization, heuristics, and simulation, can be

used to address the routing sub-problem and find an optimal or near-optimal assignment of operations to machines.

In this study, we have employed the approach by localization (AL), which enables us to address the resource allocation challenge and construct an ideal assignment model [52], [53]. This method considers both the time it takes to complete tasks and the load on each machine, which is the total processing time of the operations assigned to it. The process involves identifying, for each operation, the machine with the shortest processing time, locking in that assignment, and subsequently adding this time to all the following entries in the same column (updating the machine's workload), as shown in Table 3.3, where bold values correspond to workload updates.

Table 3. 3 Algorithm for solving routing subproblem.

Input:	FJSP problem instance	
Output:	Route of Jobs	
	For index in range(length_input):	
	row = random_select	# Get the current row by random selection
	get row_min get min_column_index	# assign setup in machine with min_pt
	for i in range(index+1, len(length_input)):	
	row_val += row_min	#Add the minimum value to the subsequent rows in the same column

Table 3. 4 Approach by localization (machine workload updates in bold)

	M1	M2	M3		M1	M2	M3		M1	M2	M3		M1	M2	M3
s11	7	6	4		7	6	**5**		**11**	6	5		7	6	**4**
s12	4	8	5		4	8	**6**		**4**	8	6		**4**	8	5
s13	9	5	4		9	5	**5**		**13**	5	5	…	9	5	4
s21	2	5	1		2	5	**1**		**6**	5	**1**		2	5	1
s22	4	6	8		4	6	**9**		**8**	6	9		4	6	8

3.4.2 Solving the Sequencing Sub-Problem/Job Shop Scheduling (JSP)

Once the assignments are settled, the problem becomes like a classical JSP problem. We just need to determine the sequence of the setups on the machines. The sequencing is feasible if it respects the natural precedence relationship among the setups of the same job, i.e., setup $S_{i,j}$ cannot be processed before setup $S_{i,j+1}$. In this research, the sequencing of the initial assignments is obtained by solving the following Mixed Integer Linear Programming (MILP) model:

Mathematical Formulation

The problem considers n jobs that must be processed in m machines. Each job consists of a total of n_j setup. Each setup S_{ij} must be assigned to a machine k and find the sequence of the job j. Precedence between the setups of a job is given by the setup planning solution. The objective is to minimize maximum make span.

The following assumptions are proposed for the FJSP:

- All the jobs and machines are available at time zero.
- Each machine can perform at most one operation at any time.
- Transportation time is not considered.

- procession time includes setup time.
- Job preemption is not allowed.
- The setup numbers are indicative of their natural logical sequence within a job.

The notations used in this paper are defined as follow:

Index:

J: Number of jobs

j: The index of jobs of {1,2,..,J}

m: Number of machines

k: The index of machine {1,2,..,m}

n_j: Number of setup in a job j

i: The index of setup {1,2,..,n_j

Parameter:

$P_{i,j,k}$: Processing Time of setup i of job j on machine k

M: a very large positive number

$$x_{i,j,k} = \begin{cases} 1, if\ the\ setup\ i\ of\ job\ j\ is\ processed\ on\ machine\ k \\ 0\ otherwise \end{cases}$$

Decision variables:

$s_{i,j,k}$: start time of the setup i of job j on machine k

$$Z_{i,i',j,j',k} = \begin{cases} 1, if\ the\ setup\ i\ of\ job\ j\ precedes\ setup\ i'of\ job\ j'\ on\ machine\ k \\ 0\ otherwise \end{cases}$$

C_{max}: Makespan

Objective Function:

$$\text{Min } C_{max} \tag{1}$$

s.t.,

$$s_{i,j,k} \leq M * x_{i,j,k} \; \forall \; \text{i, j, k} \tag{2}$$

$$s_{i+1,j,k} \geq s_{i,j,k} + P_{i,j,k} * x_{i,j,k} \tag{3}$$

$$\forall \text{i} = 1..(\text{n}_\text{j}-1), \text{j}, \text{k}, x_{i+1,j,k} = 1, x_{i,j,k} = 1$$

$$s_{i+1,j,k'} \geq s_{i,j,k} + P_{i,j,k} * x_{i,j,k} \tag{4}$$

$$\forall \; \text{i} = 1..(\text{n}_\text{j}-1), \text{j}, \text{k}' : \text{k} \neq \text{k}', \text{k}, x_{i+1,j,k'} = 1, x_{i,j,k} = 1$$

$$s_{i'j'k} \geq s_{ijk} + P_{ijk} * x_{ijk} - M\left(3 - Z_{ii'jj'k} - x_{i'j'k} - x_{ijk}\right) \tag{5}$$

$$\forall \; \text{i}, \text{i}' : \text{i} \neq \text{i}', \text{j}, \text{j}' : \text{j} \neq \text{j}', \text{k}, x_{i'j'k} = 1, x_{ijk} = 1$$

$$s_{ijk} \geq s_{i'j'k} + P_{i'j'k} * x_{i'j'k} - M\left(3 - Z_{ii'jj'k} - x_{i'j'k} - x_{ijk}\right) \tag{6}$$

$$\forall \; \text{i}, \text{i}' : \text{i} \neq \text{i}', \text{j}, \text{j}' : \text{j} \neq \text{j}', \text{k}, x_{i'j'k} = 1, x_{ijk} = 1$$

$$C_{max} \geq \sum_{k=1}^{K} s_{ijk} + \sum_{k=1}^{K} x_{ijk} * P_{ijk} \; \forall \; \text{i, j, k} \tag{7}$$

$$s_{ijk}, C_{max} \geq 0 \; \forall \; \text{i, j, k} \tag{8}$$

$$Z_{ii'jj'k} \in \{0,1\} \; \text{i}, \text{i}' : \text{i} \neq \text{i}', \text{j}, \text{j}' : \text{j} \neq \text{j}', \text{k} \tag{9}$$

The objective function is defined by Eq. (1), which minimizes the makespan. Constraint set (Eq. (2)) defines the start time for each setup on the assigned machine. The disjunctive sets (Eqs. (3) and (4)) are feasibility constraints that ensure that only one setup of a job processed on a machine at a time and precedence relationship is followed.

The disjunctive constraint sets (Eqs. (5) and (6)) avoid the overlapping of setup on same machines of different job at a time. Constraint sets (Eq. (7)) define the maximum make span. Constraint set (Eq. (8)) ensures that the starting time and make span should be either positive or zero. Constraint sets (Eqs. (9) define the types of variables.

Solver:

The goal of the experiment is to solve the problem instance to generate quality solutions (makespan). OR-Tools2 (ORT), an open-source solver developed by Google, won the gold medal in all categories in 2018 [54]. In this research, we employed Google's OR-Tools to find the sequence of the initial assignment. Concerning the solvers' version, we use version 9.6 for OR-Tools. We decided to use the CP-SAT solver, because CP-SAT proved to be better on average as reported in the literature, fairly easy to implement and compatible with other necessary Python libraries and packages [54], [55].The experiment is conducted on a system equipped with a 3 GHz Intel Core i7 4-Core (11th Gen) 16GB of DDR4 RAM | 256GB M.2 SSD.

3.5 Construction of Data Mining Dataset from Initial Solution

Creating an appropriate training dataset is a pivotal aspect of the entire rule mining procedure. When viewed from the perspective of setup sequencing, the primary objective is to identify the preferred order in which setups should be prioritized for dispatching among a collection of schedulable setups, regardless of whether they belong to the same or different jobs and are intended for the same machine at a specific moment. By extracting this knowledge from the training dataset, we enable the ability to determine the sequence for dispatching the next setup at any given time. Subsequently, this knowledge can be used to generate dispatching lists for any combination of jobs and machines, provided that the assignment or routing for each setup is known.

3.5.1 Attributes Selection

Attribute selection is the task of identifying the most appropriate set of attributes for a classifier, with the aim of reducing the number of attributes while maximizing the separation between classes [47]. This process is crucial for the effectiveness of subsequent model induction since it helps eliminate redundant and irrelevant attributes. However, it is also important to note that the attributes recorded as part of the available data may not always be the most relevant or useful for the data mining process, making the creation of new attributes a necessary consideration.

Priority relationship can be formed between the jobs while the sequencing based on their processing time, due date etc. [47], [51], [56]. This priority relationship can be reduced by only considering two setups on the same machine, among schedulable jobs, at any given instance for comparison. However, proper attribute selection is essential for capturing this relationship.

Furthermore, both the selection of raw attributes from production data and creation of new attributes are closely tied to the objectives of the scheduling problem. Objectives related to making span require different attributes to be considered compared to objectives related to flow time or tardiness. For example, attributes related to processing time, precedence relationship and associated statistics are more suitable for makespan or completion time-based objectives. Similarly, attributes related to deadlines and associated statistics are more suitable for tardiness-based objectives.

Additionally, the attributes that are recorded as part of the raw production data may not be the attributes that are the most useful for the data mining itself. Thus, new attributes creation must be considered. [47], [51], [56]. Combining raw attributes through arithmetic operations can lead to the creation of new valuable attributes as pointed out in Olafsson and Li [51]. However, it is important to avoid having a large set of attributes,

as they are often not independent of each other, which can make the process computationally impractical [42].

In this research, 11 attributes belonging to two types, raw and constructed are considered. The 4 raw attributes based on are the setup processing time (p_{ijk}) and due date of job (dj). These are considered directly from production data. Constructed attributes can further be divided into two types. Composite attributes and categorical attributes. 2 composite attributes are constructed with basic arithmetic operations following the methodology proposed by Li and Olafsson, [51], [56]. The categorical attributes represent binary variables used to indicate a direct comparison between two setups, A and B. When the raw attribute value of A exceeds that of B, the categorical value is set to 1. Conversely, when the raw attribute value of A is less than that of B, the categorical value is set to -1. For all other situations, the categorical value is set to 0. In this research, 5 categorical attributes are also constructed to capture the priority, delay, and precedence relationship among setups. Details of the attributes are shown in table 3.4.

Table 3. 5 Considered attributes for rule mining.

Type	Feature/attributes	Notation
Raw	processing time of setup A	p_A
	processing time of setup B	p_B
	due data of the A	d_A
	due data of the B	d_B
Constructed	if processing time of A is higher than B (categorical)	p_A>B
	if due date of A is higher than B (categorical)	d_A>B

	processing time difference	p_A-B
	Due date difference	d_A-B
	if A & B has precedence relationship (categorical)	$Z_{ii'j}$
	if A precedes B (categorical)	$Z_{ij}>Z_{i'j}$
	if A and B processed on same machine (categorical)	$x_{ii'k}$

3.5.2 Creation of Training Dataset

The goal of this step is to convert the initial nominal scheduling solution into training data. From the previous steps, nominal solutions for each problem instance are saved as a flat data file. The columns represent separate data attributes, and each row of the file represents the schedule of a setup.

Then the training dataset for sequencing setups is generated by following 2 steps, as shown in Figure 3.4.

- First, the first setup in the schedule list is selected and all setups that can be processed at the start time are taken. Subsequently all possible combinations of setup pairs are selected. Thus, for a problem instance with j job each having i setups, there will be 2 x C_2^{ixj} possible setup pair.
- Then, rows for all possible pairs of setups are appended to a dataset with their attributes.

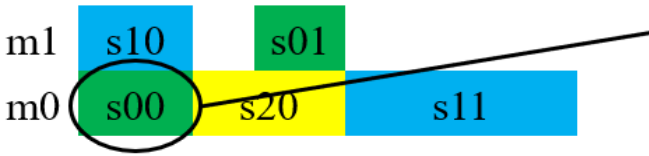

Figure: Initial nominal solution

Table: Flat data format

S_{ij}	m	P_{ijk}	d_j	seq	s_{ijk}	e_{ijk}
s00	0	5	8	0	0	5
s01	1	2	8	1	5	7
s10	1	4	10	0	0	4
s11	0	6	10	1	7	13
s20	0	3	5	2	4	7

AB		P_{Ajk}	d_j	$P_{Bj'k'}$	$d_{j'}$	…	A_go first
s00	s01	5	8	2	8	…	1
s01	s00	4	10	5	8	…	0
s00	s11	5	8	6	10	…	1
:	:	:	:	:	:		
:	:	:	:	:	:		
s11	s20	6	10	3	5	…	0

(j, j’ ∈ set if all job (1…J); k,k’ ∈ set if all machine (1…m))

Figure 3. 3 Process of training dataset generation

3.6 Development of Dispatching Rule Mining Model

Setup sequencing rule or dispatching rule are mind by the following methodology of supervised learning. The implementation details are described in the following sections.

3.6.1 Preprocessing of the data

Preprocessing the data, including feature selection and data cleaning, such as handling missing, outliers, inconsistent, skew values, removing duplicates, ensuring data format consistency, correcting typos, errors, dealing with irrelevant or redundant information etc. In the present scenario, case studies have been meticulously crafted through simulation. Nevertheless, it is imperative to emphasize the significance of this step, particularly when dealing with datasets derived from real-world manufacturing systems.

3.6.2 Model Selection

The choice of potential classifiers suitable for the problem depends on the problem's complexity, dataset size, interpretability needs, and available algorithms. For this research, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes, and Logistic Regression is chosen which represent a mix of ensemble, instance-based, linear, and probabilistic algorithms. These classifiers offer a range of strengths and weaknesses, and they are widely recognized and applied in various classification scenarios. Given the relatively small dataset size and the need to understand the behavior of different algorithm families, these choices provided a comprehensive baseline for assessment.

Random Forest (RF)

Random Forest classification combines the output of multiple decision trees to reach a single classification decision. Decision trees evaluate each node and decide which leads to another node. This process is repeated until a final decision is reached. Decision trees

seek to find the best split to subset the data via classification and regression tree algorithm. Gini impurity, mean square error or information gain may be used for performance evaluation. Decision trees are prone to overfitting and bias problems; however, prediction can be tuned toward greater accuracy when an ensemble is formed by multiple decision trees.

Random forest algorithm takes advantage of feature randomness to create an uncorrelated forest of decision trees, by generating a random subset of features which ensures low correlation among decision trees. While decision trees by themselves consider all the possible feature splits, random forest only uses a subset of those features. Three main hyperparameters that must be specified for random forest algorithms are node size, number of trees, and number of sample features. making it compatible with both regression and classification problems [57].

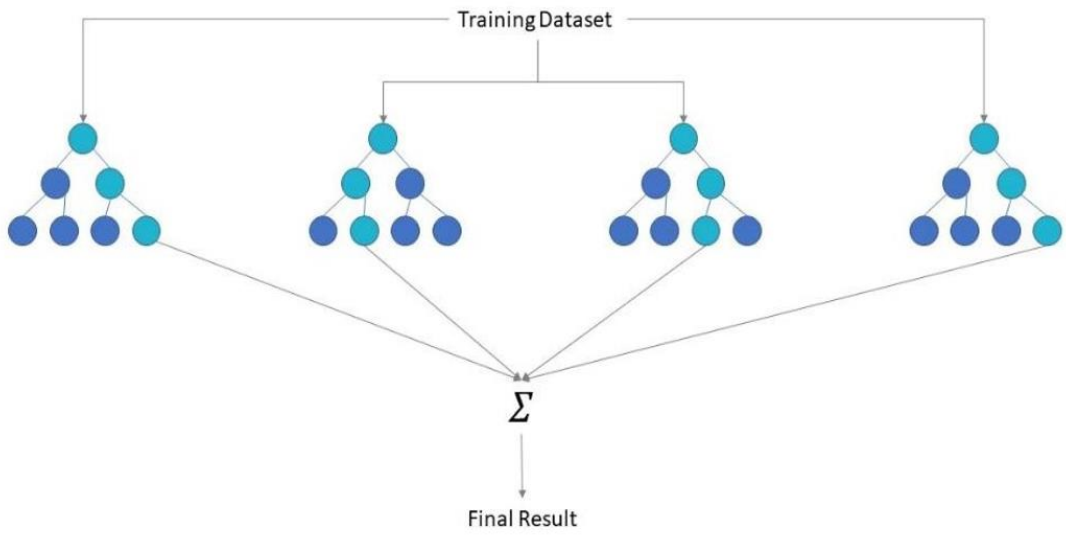

Figure 3. 4 Random Forest Classifier

Strengths:

- Robust to overfitting due to ensemble nature (combining multiple trees).
- Handles both categorical and continuous features well.

- Provides feature important rankings for better interpretability.
- Tends to handle noisy data well.

Weaknesses:

- Can be computationally expensive with large datasets and many trees.
- May not perform well on datasets with highly correlated features.

K-Nearest Neighbors (K-NN)

K-NN classifier is a lazy type of classification technique which looks at the neighbors of the data point being classified. It then conducts a vote amongst the k of those neighbors closest to the data point and classify according to the majority vote. There is no defined formula for choosing the number k. K is chosen by trial and validation checks to assess the best choice. It can be seen below that if k is set to 3, the 'red' point being classified will be labeled 'class B' however, if k is set to 6, that same point will be labeled 'class A' [57].

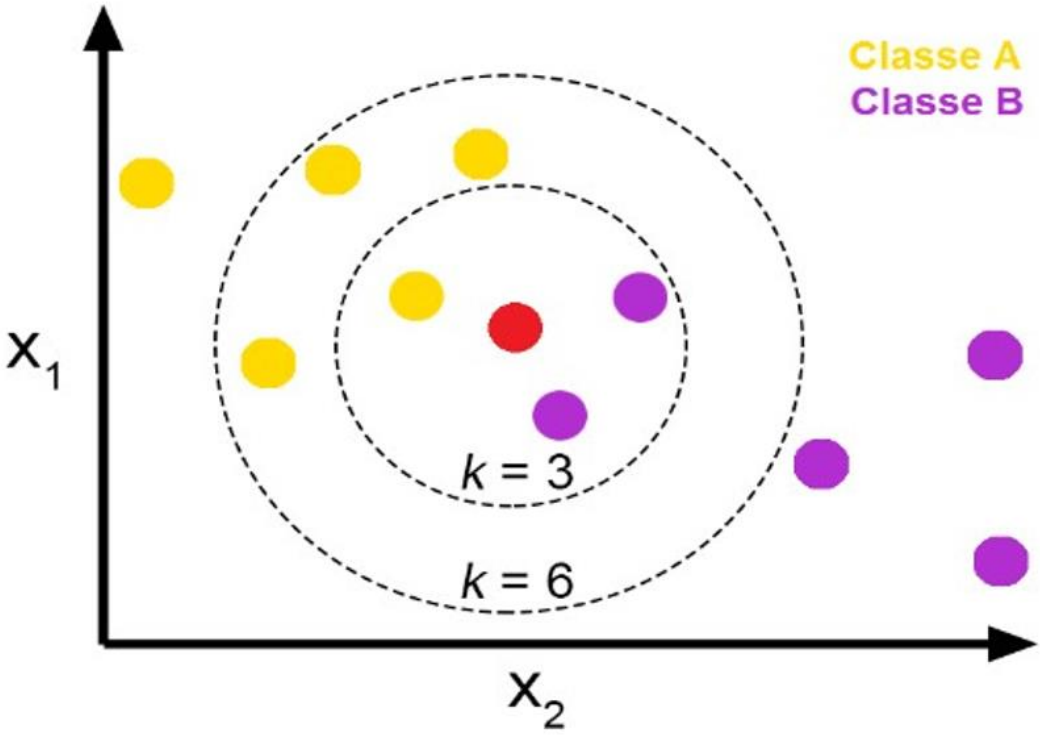

Figure 3. 5 K-NN Classification

Strengths:

- Simple and intuitive concept.
- Can capture non-linear relationships in data.
- Works well when classes are well-separated.

Weaknesses:

- Sensitive to the choice of distance metric and the value of K.
- Can be computationally expensive during prediction, especially with large datasets.
- Prone to overfitting if the dataset has noisy or irrelevant features.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful and versatile classification algorithm widely used in machine learning and pattern recognition. It is fundamentally a geometric classification approach. Instead of directly modeling probabilities like some other classifiers, it focuses on finding the optimal decision boundary (hyperplane) that best separates different classes in the feature space. The key idea behind SVM is to maximize the margin between the two classes. The margin is the distance between the decision boundary and the nearest data points (support vectors) of each class. By maximizing this margin, SVM aims to create a robust and generalizable model that can handle new, unseen data effectively. SVM can perform both linear and non-linear classification. In its basic form, it constructs a linear hyperplane to separate classes. However, using kernel functions (e.g., polynomial, radial basis function), SVM can transform the feature space to find non-linear decision boundaries, making it adaptable to complex data distributions. Support vectors are the data points that are closest to the decision boundary. These are the most influential points in determining the optimal hyperplane. SVM focuses on these critical examples to ensure robustness and efficiency. SVM introduces a regularization parameter, often denoted as 'C,' which

balances the trade-off between maximizing the margin and minimizing the classification errors. Smaller values of 'C' prioritize a wider margin but might allow some misclassifications, while larger values of 'C' aim to minimize errors even if it means a narrower margin [57].

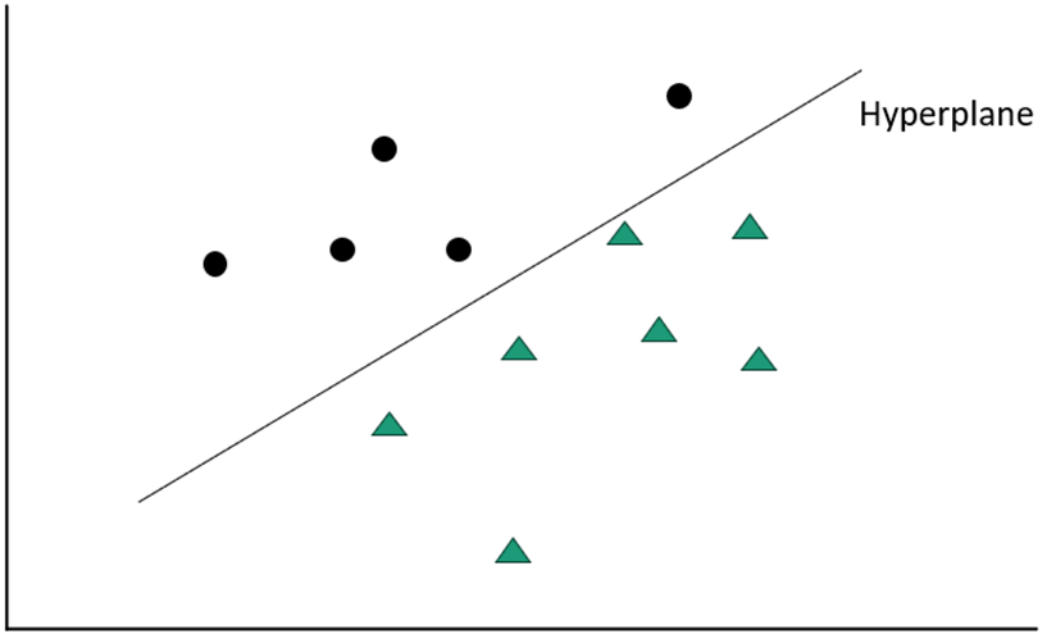

Figure 3. 6 SVM Classification

Strengths:

- Effective in high-dimensional spaces.
- Good for datasets with clear margin of separation.
- Can handle non-linear decision boundaries using kernel functions.

Weaknesses:

- Can be sensitive to the choice of kernel and hyperparameters.
- May not perform well on datasets with significant overlap between classes.
- Computationally demanding, especially with large datasets.

Naive Bayes (NV)

Naive Bayes is a probabilistic algorithm that models the likelihood of a data point belonging to a particular class. It calculates the probability of each class given a set of features and selects the class with the highest probability as the predicted class. One key characteristic of Naive Bayes is the assumption of feature independence. It assumes that all features are conditionally independent given the class label. The algorithm estimates the probability distributions of features for each class during the training phase. It calculates the prior probability of each class and the likelihood of each feature given the class. To make a prediction for a new data point, Naive Bayes calculates the posterior probability of each class given the observed features using Bayes' theorem. It then selects the class with the highest posterior probability as the predicted class [57].

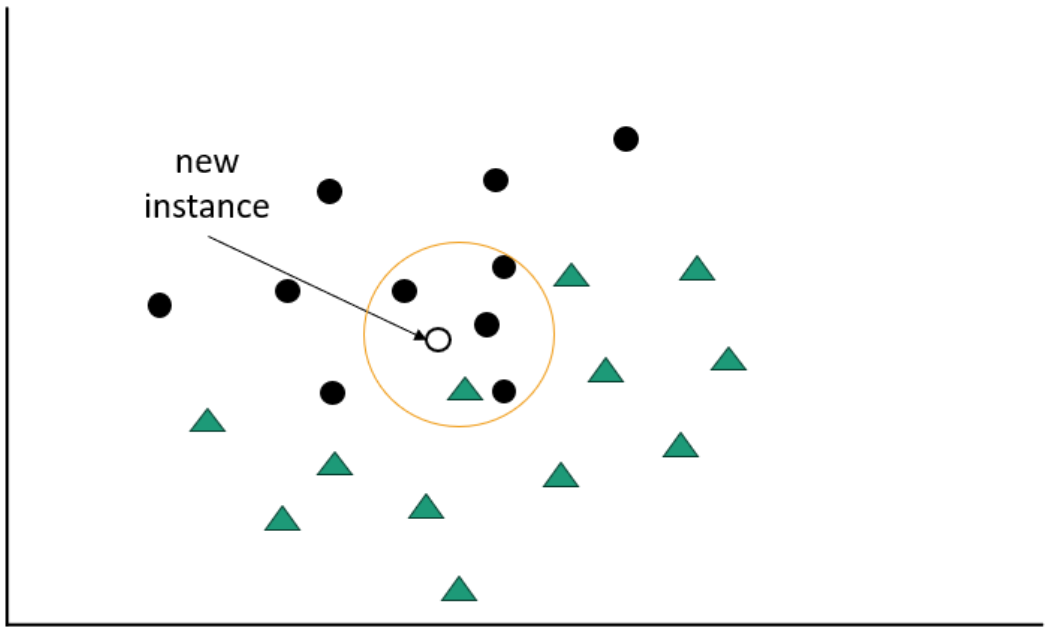

Figure 3. 7 NV Classification

Strengths:

- Simple and computationally efficient.
- Works well with high-dimensional data.
- Suitable for text classification and other probabilistic tasks.

Weaknesses:

- Assumes feature independence (naive assumption), which might not hold true.
- May not perform well when strong dependencies exist between features.

Logistic Regression (LR)

Logistic regression (LR) is a robust and widely adopted approach in supervised classification. It can be viewed as an extension of ordinary regression, primarily used for modeling binary outcomes that signify the presence or absence of an event. LR's primary function is to estimate the probability that a new data point belongs to a particular class. As probabilities inherently range between 0 and 1, LR serves as a probabilistic model. Consequently, to employ LR as a binary classifier, we assign a threshold to distinguish between the two classes. For instance, if the probability assigned to an input instance exceeds 0.50, it is categorized as 'class A'; otherwise, it is categorized as 'class B.' LR can also be adapted to handle categorical variables with more than two categories. This extended form of LR is known as multinomial logistic regression [57].

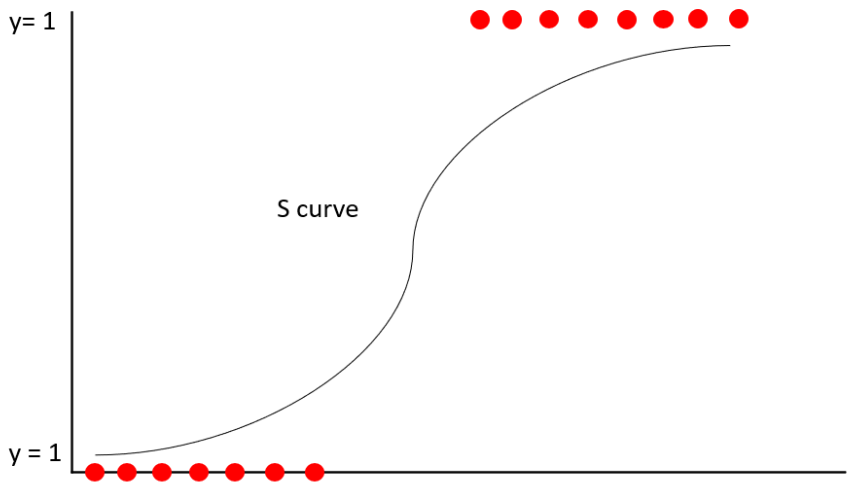

Figure 3. 8 Logistic Regression

Strengths:

- Simple and interpretable.
- Works well when classes are linearly separable.
- Provides probabilities for class membership.

Weaknesses:

- Assumes a linear relationship between features and the log-odds of the response variable.
- Can struggle with complex relationships in the data.

3.6.3 Parameter Tuning

Identify hyperparameters specific to the chosen models (e.g., learning rate, number of trees, regularization strength) that affect model performance. We investigated the typical variation of parameters for each learning algorithm. This section provides a summary of the parameters employed for each learning algorithm (Caruana, 2006).

Random Forest (RF)

The number of trees in the forest varies between 50 to 500. The number of features to consider when looking for the best split was 1, 2, 4, 6, 8 and 11.

KNN

We used 10 values of k ranging from k = 1 to (number of sample). The standard Euclidean distance was used as distance computation matric.

SVM

The following kernels were used: linear, polynomial degree 3 and radial with kernel varying coefficient (1 / (n_features * X.var()), 1 / n_features, 0.001, 0.01, 0.5, and 1)

Naive Bayes (NB)

We employed Gaussian Naive Bayes.

Logistic Regression (LR)

Regularized logistic regression is employed. Tolerance was varied by a factor of 10 from 10^{-5} to 10^{5}.

3.6.4 Cross Validation

Cross validation involves splitting the data set into equal folds, testing our model on one-fold after training is performed on the remaining folds. The number of folds used for cross validation is determined by the user. The folds above are given the label of the split on which the test is performed. It should be understood however that each fold is truly a tenth of the entire data points. After testing the classifier on each fold, the cross validation is completed. This process is naturally followed by obtaining a performance measure of the classifier on all the tests conducted. Prediction statistics are calculated based on the results of testing on all folds.

Figure 3. 9 10 Folds Cross Validation

In this research, stratified K fold CV was used on the dataset to perform 10-fold cross validation. Dataset was shuffled to have representative folds.

3.6.5 Model Evaluation Metrics

In this research, the best-performing model based on its performance on the cross-validation set is selected and assessed against the test set, which it has never seen before. This gives an estimate of its generalization ability. For evaluating the performance, we followed the approach proposed by (Caruana, 2006).

In this research, we have considered seven performance metrics:

- **Accuracy:**

Mathematical Formula:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Range: [0, 1]

Interpretation: Accuracy measures the proportion of correctly classified instances out of the total number of instances. It ranges from 0 (completely inaccurate) to 1 (perfect accuracy).

- **F-score (F1 Score):**

Mathematical Formula:

$$F1 = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

Range: [0, 1]

Interpretation: The F1 score is the harmonic mean of precision and recall. It balances precision and recall and is useful when dealing with imbalanced datasets.

- **ROC Score (Receiver Operating Characteristic AUC):**

Mathematical Formula:

ROC Score measures the area under the Receiver Operating Characteristic (ROC) curve, which plots True Positive Rate (TPR) against False Positive Rate (FPR) at different thresholds.

Range: [0, 1]

Interpretation: ROC Score quantifies the ability of a classification model to discriminate between positive and negative classes. A higher ROC AUC indicates better model performance.

- **Precision:**

Mathematical Formula:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Range: [0, 1]

Interpretation: Precision measures the proportion of true positive predictions out of all positive predictions. It reflects the accuracy of positive predictions.

- **Recall (Sensitivity or True Positive Rate):**

Mathematical Formula:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Range: [0, 1]

Interpretation: Recall measures the proportion of true positive predictions out of all actual positive instances. It quantifies the model's ability to identify all positive instances.

- **Mean Squared Error (MSE):**

Mathematical Formula:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Range: [0, ∞)

Interpretation: MSE measures the average squared difference between actual values (y_i) and predicted values ($\hat{y}_i$). Smaller MSE values indicate better model fit.

- **Cross-Entropy (Log Loss):**

Mathematical Formula (for binary classification):

$$Cross\ Entropy = -\frac{1}{N}\sum_{i=1}^{N}[\,y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Range: [0, ∞)

Interpretation: Cross-Entropy measures the dissimilarity between predicted probabilities ($\hat{y}_i$) and actual binary labels (y_i). Lower values indicate better model calibration.

Comparing Across Performance Metrics

Performance metrics such as accuracy and F-score have range [0,1]. On the other hand, mean square error or cross entropy ranges from [0, ∞). For accuracy, F-score, precision, recall and ROC score, the higher the value the better the model fit. However, for mean square error or cross entropy, lower value indicates better model performance. Thus, to measure model performance on comparable scale and to find average across metrics, each performance metric is normalized between 0 to 1.

3.7 Implementation Detail

- **Device:**

CPU: Intel i5 11 the Gen

RAM: 8 GB

OS: Windows 10 64-bit

Store: 256 GB SSD

- **Environment:**

Anaconda version: conda 23.5.2 (Link: https://www.anaconda.com/download)

Python version: 3.11.3

- **Libraries:**

Pandas (Link - https://pandas.pydata.org/)

Numpy (Link - https://numpy.org/)

IPython (Link - https://pypi.org/project/ipython/)

Collections (Link- https://docs.python.org/3/library/collections.html)

io (Link- https://docs.python.org/3/library/io.html)

ortools (Link- https://pypi.org/project/ortools/)

sklearn (Link- https://scikit-learn.org/stable/)

matplotlib (Link- https://matplotlib.org/)

3.8 Reconfiguration of Initial Nominal Schedule Under Disruption

This section explains the rescheduling strategy. The rescheduling strategy employs dynamic adjustments to the existing schedule, prioritizing the reassignment of affected jobs to alternative available machines. This ensures that production can resume as swiftly as possible following a breakdown event. In this research, we have considered a FJSP with machine breakdown problem based on the following definitions and assumptions:

Index:

J: Number of jobs

j: The index of jobs of {1,2,..,J}

m: Number of machines

k: The index of machine {1,2,..,m}

n_j: Number of setup in a job j

i: The index of setup {1,2,..,n_j}

$m_{i,j,k}$: A subset of machines for setup i

$m_{i,j,k} \subseteq (m_1, m_2, .., m_k)$

Parameter:

$P_{i,j,k}$: Processing Time of setup i of job j on machine k

$x_{i,j,k} = \begin{cases} 1, if\ the\ setup\ i\ of\ job\ j\ is\ processed\ on\ machine\ k \\ 0\ otherwise \end{cases}$

T^k_{MTBD}= Mean time between breakdown of machine k

TH_k: Breakdown probability threshold of machine k

Decision variables:

$s_{i,j,k}$: start time of the setup i of job j on machine k

$e_{i,j,k}$: end time of the setup i of job j on machine k

t^k_{BD}: Breakdown time of machine k, $t^k_{BD} = f(t)$

Assumptions:

- The occurrence of machine failures is modeled as following an exponential distribution
- During a production cycle, only one machine will experience breakdown

3.8.1 Machine Breakdown Distribution

According to the assumption of He and Sun [59], breakdown probability follows the exponential distribution.

$$P_k = \begin{cases} 0\,, & when\ t \leq 0\ or\ t = r_{bk} \\ 1 - e^{-\lambda t}, & when\ 0 < t < r_{bk} \end{cases}$$

Here, P_k = Probability of machine failure

r_{bk} = Estimated repair time

λ = 1/Mean time between two successive breakdowns

Following this assumption, this thesis introduces a Monte Carlo simulation-based approach to model the probability of breakdowns occurring over a production cycle. The simulation model is implemented using Python, leveraging the random and matplotlib libraries.

Simulation Model for Machine Breakdown:

- **Setting the mean time between two successive breakdowns (lambda):** A key parameter that influences the simulation's behavior is the mean time between two successive breakdowns (lambda). This parameter is defined as user-adjustable, allowing for different real-time scenarios and system characteristics to be explored.
- **Generating Random Breakdown Times:** The simulation generates random breakdown times for each machine independently, using exponential distribution. We conduct 1000 simulations for each machine to collect data on breakdown times.
- **Calculating Breakdown Probability:** We compute breakdown probabilities at various time points for each machine. This allows us to construct cumulative probability curves specific to each machine.

In this research, If the probability function exceeds a specified threshold, the machine will experience a breakdown. To simply the problem, multiple breakdowns are not considered.

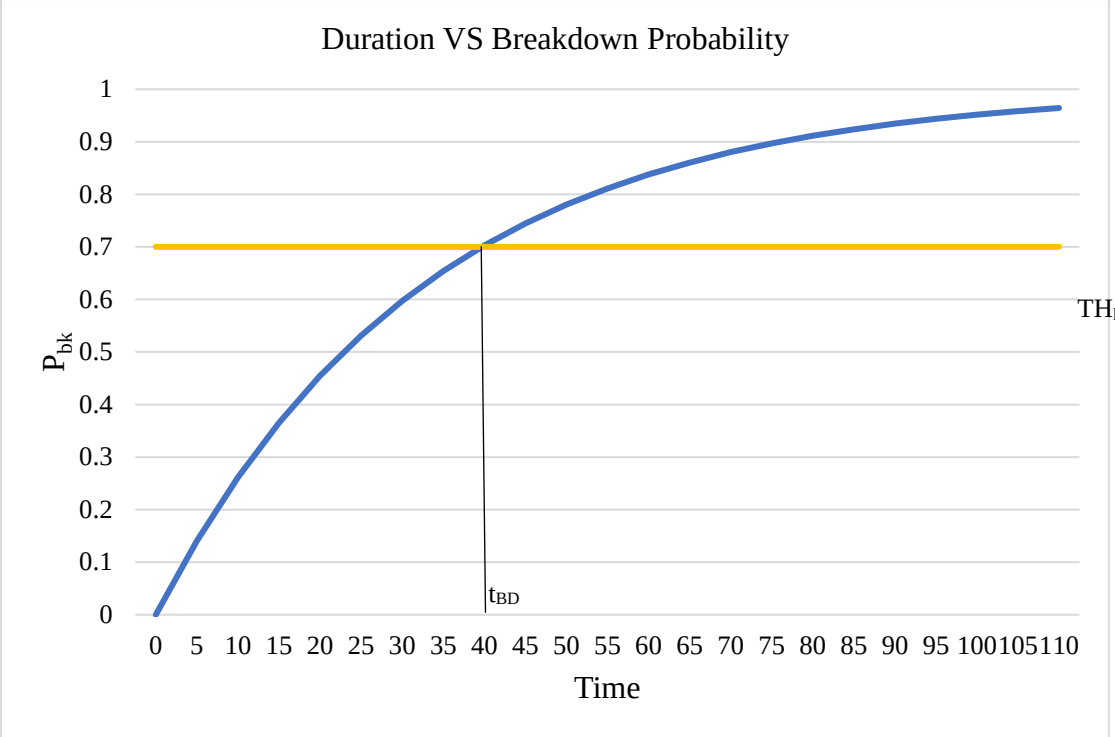

Figure 3. 10 Breakdown Probability VS Duration

3.8.2 Rescheduling Framework

To address rescheduling in response to machine breakdowns, a comprehensive strategy is proposed, which is outlined in the following framework (Figure 3.7):

Assuming an initial state at t = 0, where the probability of machine breakdown is zero, the prescheduling process is initiated on the job floor, and setups are executed in accordance with the initial nominal scheduling solution. In instances where no machine breakdown occurs, this schedule becomes the realized schedule. As the probability of machine breakdown surpasses a predefined threshold, machine failures are anticipated. Subsequently, the following decision criteria must be evaluated:

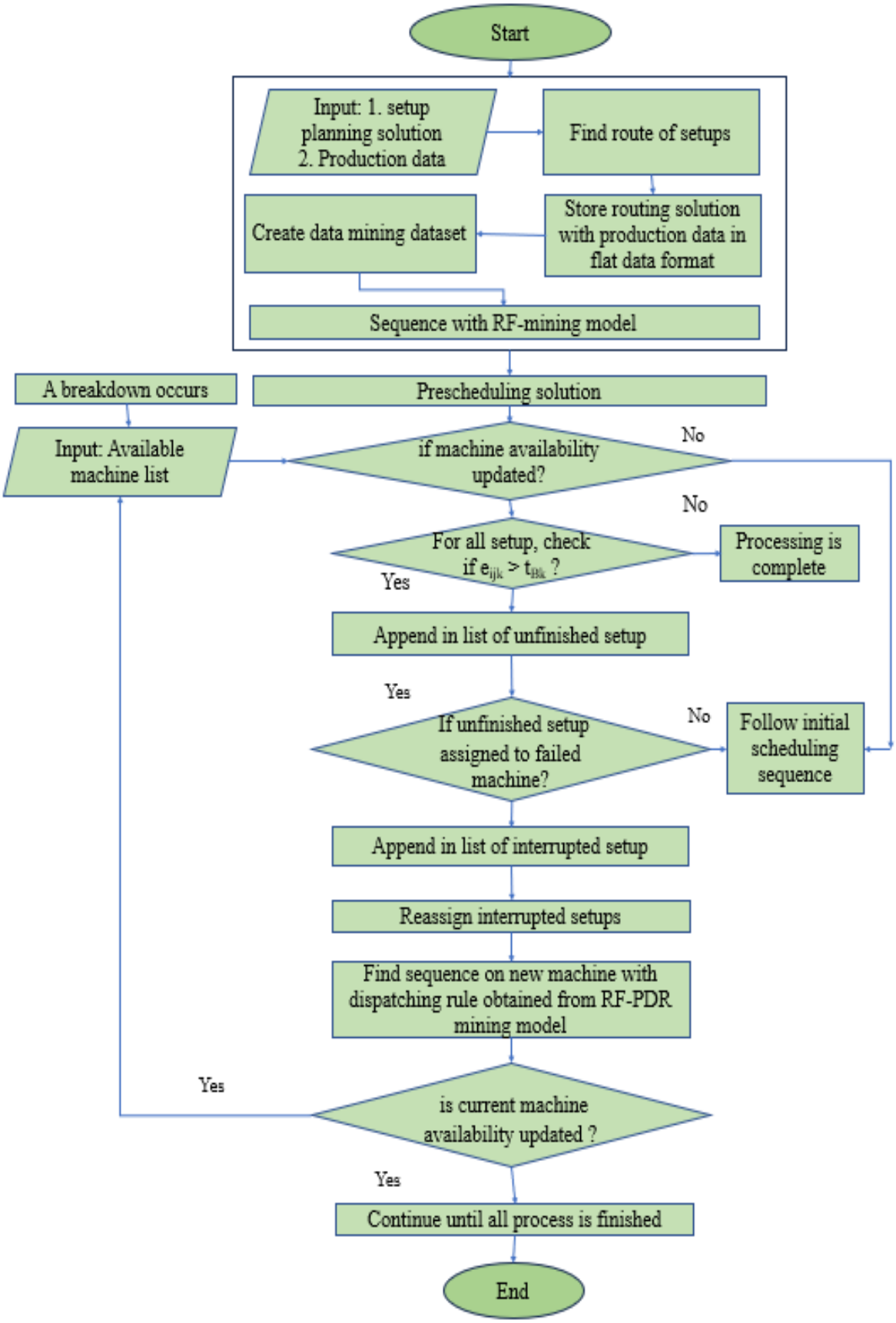

Figure 3. 11 Rescheduling Framework

- **Identification of Interrupted Setups:** For all setups that were in progress on the broken machine at the time of breakdown, a critical assessment is conducted. These setups are categorized as "interrupted setups" if their scheduled end time exceeds the breakdown time.
- **Reassignment of Interrupted Setups:** To resume production without any delay, these interrupted setups must be reassigned to currently available eligible machines. This reassignment is executed following a localization heuristic approach.
- **Sequencing of Interrupted Setups:** Once the setups have been reassigned to new machines, their sequence is determined using a dispatching rule derived from the RF-PDR mining model.
- **Continuation of the Rescheduling Process:** The rescheduling process is iteratively executed until the machine is repaired and brought back into operational condition. Throughout this process, the current availability of resources is continuously taken into consideration to ensure optimal scheduling decisions are made.

This rescheduling framework is designed to effectively address machine breakdowns, minimizing disruption to production processes, and optimizing resource utilization in a systematic and adaptive manner.

3.8.3 Robust and Stability Measures of Rescheduling

The rescheduling implemented on the job floor is characterized by two crucial attributes: robustness and stability. The development of a rescheduling system that embodies both robustness and stability is imperative to mitigate the impact of unforeseen disruptions. In this study, the robustness and stability metric are adopted from He and Sun [59]and defined as follows:

$$\text{Robustness, RM} = \frac{C_{maxR} - C_{maxp}}{C_{maxp}} \; x \; 100\%$$

Here, C_{maxR} = makespan after rescheduling

C_{maxp} = makespan of prescheduling

The stable measures can be articulated as follows:

$$\text{SM} = \min \frac{\sum_{j=1}^{n'} \sum_{i=1}^{q'} |C_{ijp} - C_{ijR}|}{\sum_{j=1}^{n} n_j}$$

Here, n' = no of unfinished and currently in-progress jobs,

n = total number of jobs,

q' = no of unfinished and currently in-progress setup of job i.

C_{ijp} = predicted completion time for setup i of job j in the prescheduling phase

C_{ijR} = completion time for setup i of job j in the rescheduling process

CHAPTER 4

RESULT AND DISCUSSION

4.1 Overview

This chapter presents the empirical findings of the research. The proposed methodology is evaluated by analyzing simulated data, providing insights into the research questions and objectives. To bolster the clarity and comprehensibility of the results, a variety of tables and graphs are also employed.

4.2 Initial Nominal Solution

In the context of our simulation experiments, we worked with three distinct datasets of comparable sizes. These datasets were generated through random processes, as elaborated in Appendix 1. Following the methodology outlined in Section 3.4, we initially obtained nominal solutions, encompassing routing and sequencing decisions. For visual reference, please refer to Figure 4.1, which illustrates the Gantt chart derived from these obtained solutions.

Subsequently, we persisted these solutions in a flat file format to assemble the dataset required for rule mining, as detailed in Table 4.1. Each row within the flat data file corresponds to a specific setup, while the columns encapsulate the following parameters:

S_{ij} = Setup ID

k = Assigned machine

P_{ijk} = Processing time of S_{ij} on k

d_j = Due date of job j

seq = Sequence of S_{ij} on k

s_{ijk} = Start time of S_{ij} on k

e_{ijk} = End time of S_{ij} on k

The next step involved crafting a training dataset from these flat files by aggregating all feasible setup pairs alongside their corresponding attributes for each of the three case studies (please see Appendix 2). In total, we generated 313 setup-pairs from these three case studies.

Table 4. 1 Initial nominal solution in flat data format. (a) Case study 1, (b) Case study 2, (c) Case study 3.

(a)

S_{ij}	k	P_{ijk}	d_j	seq	s_{ijk}	e_{ijk}
s00	0	23	49.2	1	16	39
s01	1	21	49.2	2	39	60
s02	1	27	49.2	3	60	87
s10	0	31	44.4	2	39	70
s11	2	29	44.4	2	70	99
s20	0	16	45.2	0	0	16
s21	2	30	45.2	1	18	48
s30	1	18	48	0	0	18
s31	0	19	48	3	70	89
s32	0	19	48	4	89	108
s40	2	18	42.8	0	0	18
s41	1	21	42.8	1	18	39

(b)

S_{ij}	k	P_{ijk}	d_j	seq	s_{ijk}	e_{ijk}
s00	1	17	71	1	15	32
s01	2	16	71	3	44	60
s02	1	19	71	3	60	79
s10	2	30	36.8	4	60	90
s11	0	13	36.8	2	90	103
s20	2	10	48.5	0	0	10

s21	1	18	48.5	2	32	50
s30	1	15	73	0	0	15
s31	2	22	73	2	22	44
s32	0	31	73	1	48	79
s40	2	12	35	1	10	22
s41	0	26	35	0	22	48

(c)

S_{ij}	k	P_{ijk}	d_j	seq	s_{ijk}	e_{ijk}
s00	0	10	36	0	0	10
s01	0	20	36	3	75	95
s10	2	11	82	0	0	11
s11	0	27	82	2	48	75
s12	2	15	82	3	75	90
s20	1	30	101	0	0	30
s21	1	18	101	1	30	48
s22	1	25	101	2	48	73
s30	2	29	49	1	11	40
s31	2	18	49	2	40	58
s40	0	38	75	1	10	48
s41	1	28	75	3	73	101

These datasets are meticulously prepared to facilitate comprehensive analysis and model development for subsequent rule mining process.

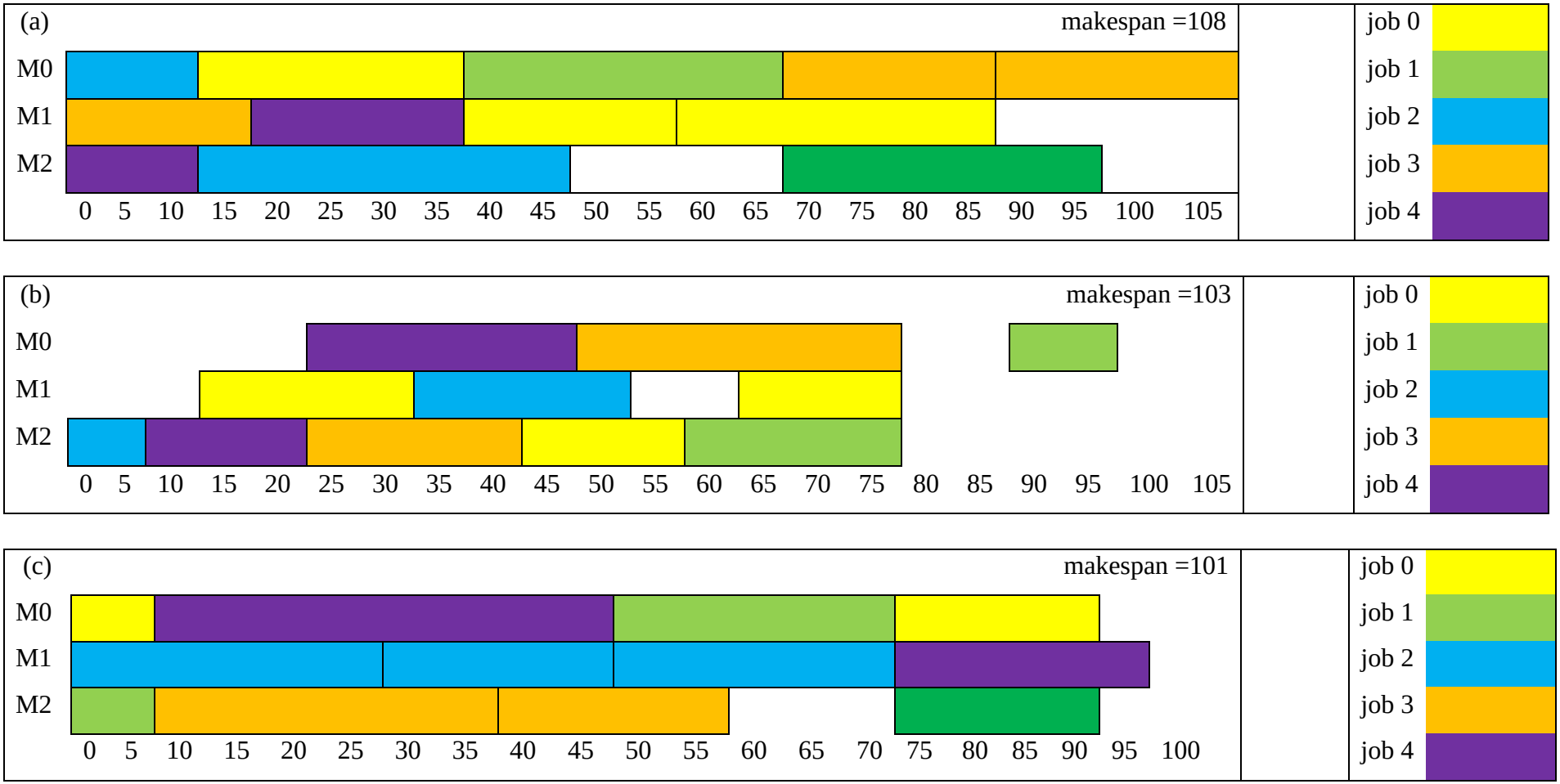

Figure 4. 1 Gantt chart (a) Case study 1, (b) Case study 2, (c) Case study 3

4.3 Parameter Tuning and Model Selection

To rigorously evaluate the performance of our model, we employed a systematic approach. We began by selecting 250 setup-pair instances at random from a comprehensive dataset compiled from three distinct case studies. These instances were divided into training and testing sets, with 5-fold cross-validation applied to each trial to ensure robustness and reduce bias.

The experimentation involved training models and selecting optimal parameters for the prediction of sequences between two setups. This evaluation process is illustrated in Figures 4.2, 4.3, 4.4, and 4.5, where we present the performance metrics based on seven evaluation parameters: Accuracy (ACC), F-score (FSC), Receiver Operating Characteristic (ROC) score, Precision (APR), Recall (REC), Root Mean Square Error (RMS), and Cross-Entropy (MXE), as well as the execution time (TIME). Five classifiers were considered: Random Forest (RF), k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naïve Bayse (NB) and Logistic Regression (LR), each with varying parameters.

4.3.1 Key Findings and Model Selection

- **Random Forest (RF) Classifier:** The RF classifier with 500 trees and 11 features consistently outperformed other configurations across all evaluation metrics. However, it is important to note that the computational time increased significantly, from 3 seconds for 50 trees to 16 seconds for 500 trees. Interestingly, beyond 300 trees, the performance metrics exhibited minimal change. Hence, for the RF classifier, a balance between computational efficiency and performance led us to select the model with 300 trees and 11 features for building the rule mining model, referred to as the RF-PDR mining model.

- **k-Nearest Neighbors (KNN) Classifier:** In the case of KNN, a k-value of 1 yielded the best metrics. However, the computational time was minimal for all k-values, making it a computationally efficient choice.
- **Support Vector Machine (SVM) Classifier:** SVM exhibited similar performance across various parameter combinations. Models with a linear kernel and a scale coefficient consistently outperformed other. SVM models were also relatively efficient in terms of execution time.
- **Logistic Regression (LR) Classifier:** LR showed the weakest performance across all metrics, with limited variation based on parameter selection. The best results were obtained with a tolerance value of 0.001.

4.3.2 Normalized of Performance Metrics

To facilitate a fair and comprehensive comparison across different algorithms, performance metrics were scaled using z normalization. This enabled us to objectively evaluate and select the best model for learning dispatching rules. Table 4.2 presents the normalized values for each algorithm on each of the seven metrics and execution time, calculated as the average over 5-fold cross-validation across different parameter combinations.

In the table, the algorithm with the best performance on each metric is **boldfaced**. Upon aggregating the results across all seven metrics, RF emerged as the superior model. Following RF, KNN exhibited the next best performance, while LR consistently performed the poorest across all metrics.

Selected Model for Dispatching Rule Mining: Taking into consideration both performance and computational efficiency, we opted for the RF classifier with 300 trees and 11 features to build the RF-PDR mining model. This decision strikes a balance

between robust predictive capabilities and manageable computational demands, making it an ideal choice for learning dispatching rules in our context.

This selection ensures that the RF-PDR mining model can provide effective sequencing recommendations for setups in a flexible job shop scheduling environment, thereby optimizing manufacturing operations. The comprehensive evaluation process presented in this section underpins our confidence in the chosen model's ability to deliver real-world value.

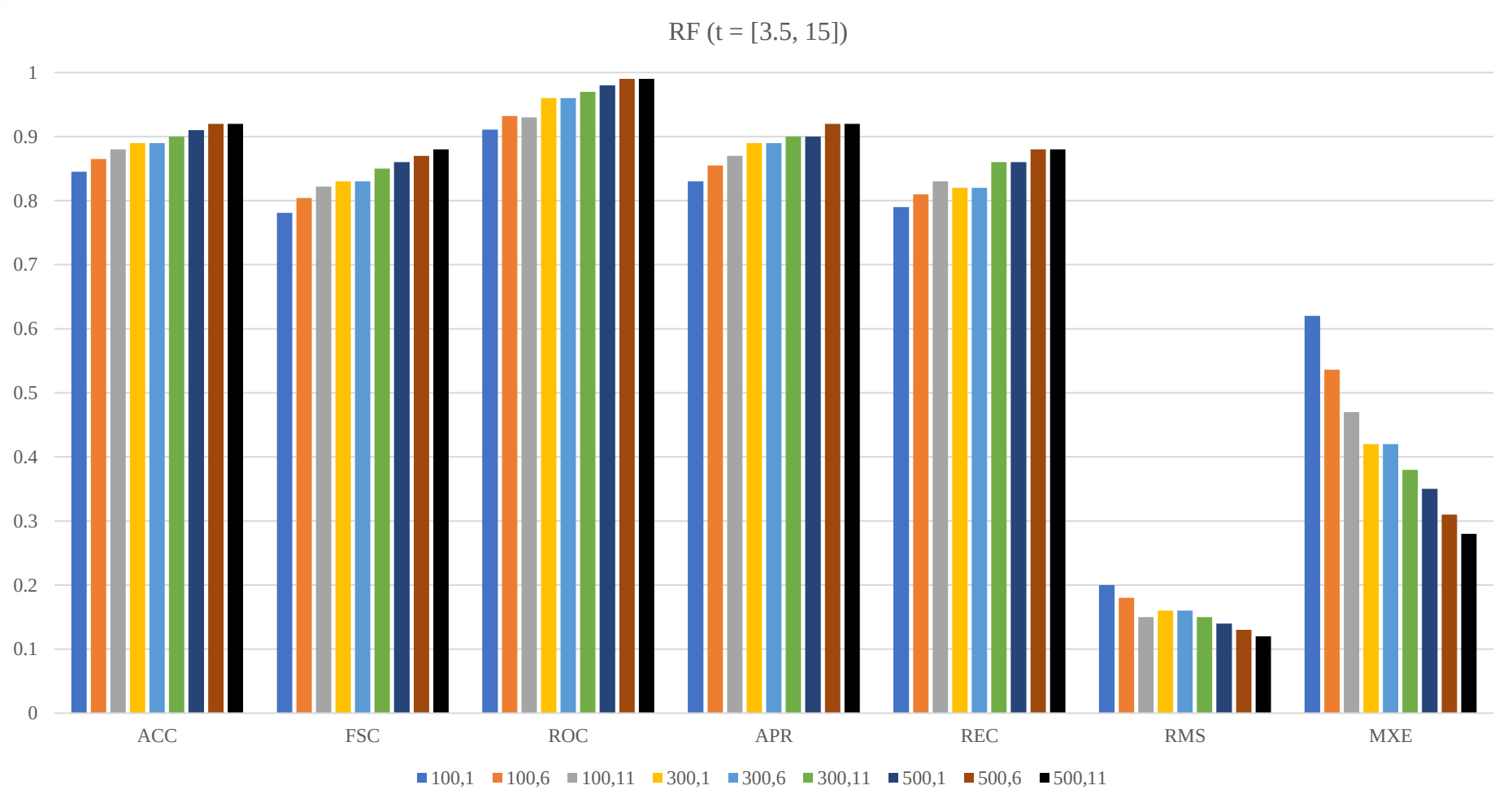

Figure 4. 2 Performance metrics for RF classifier

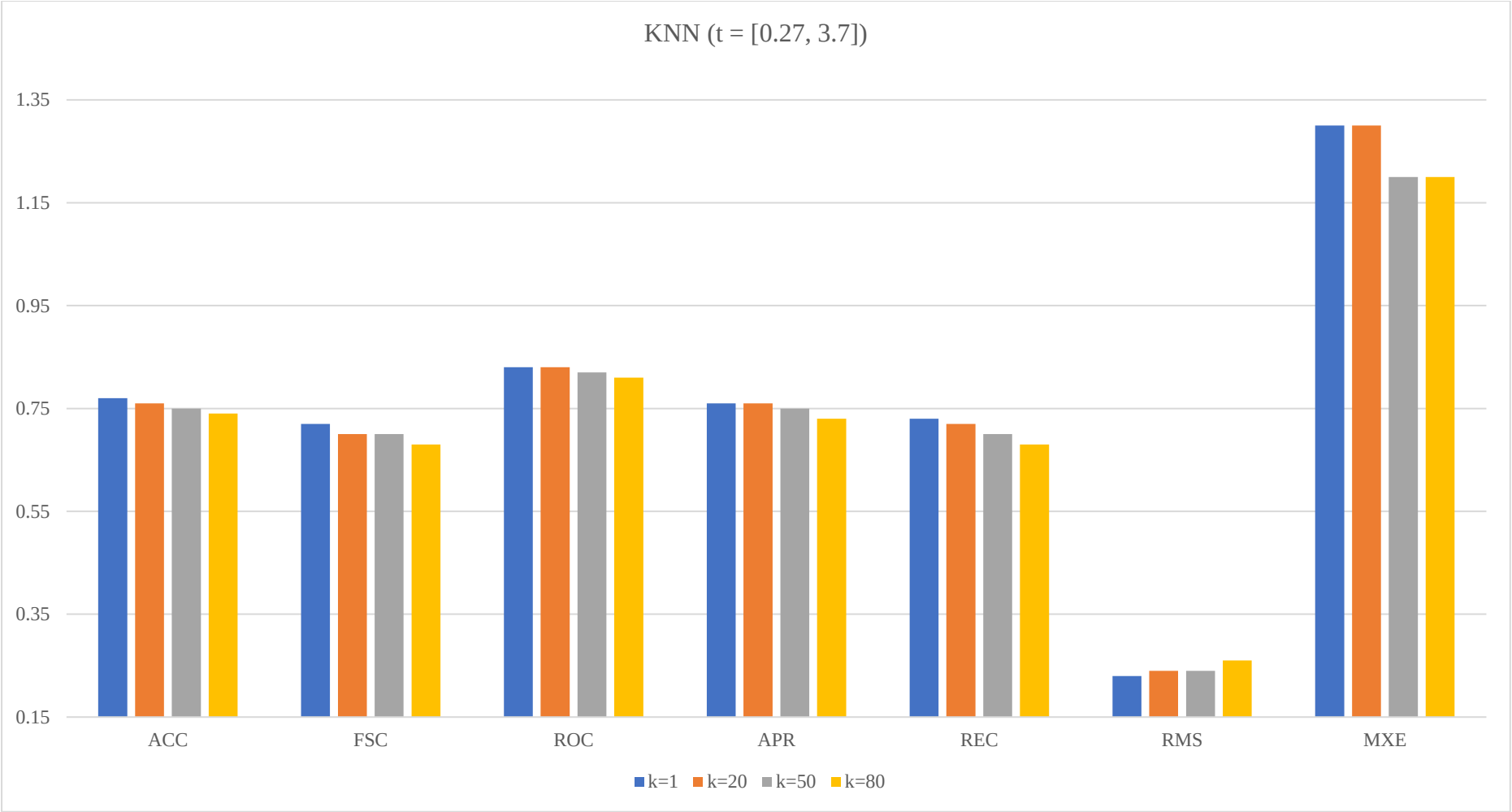

Figure 4. 3 Performance metrics for KNN classifier

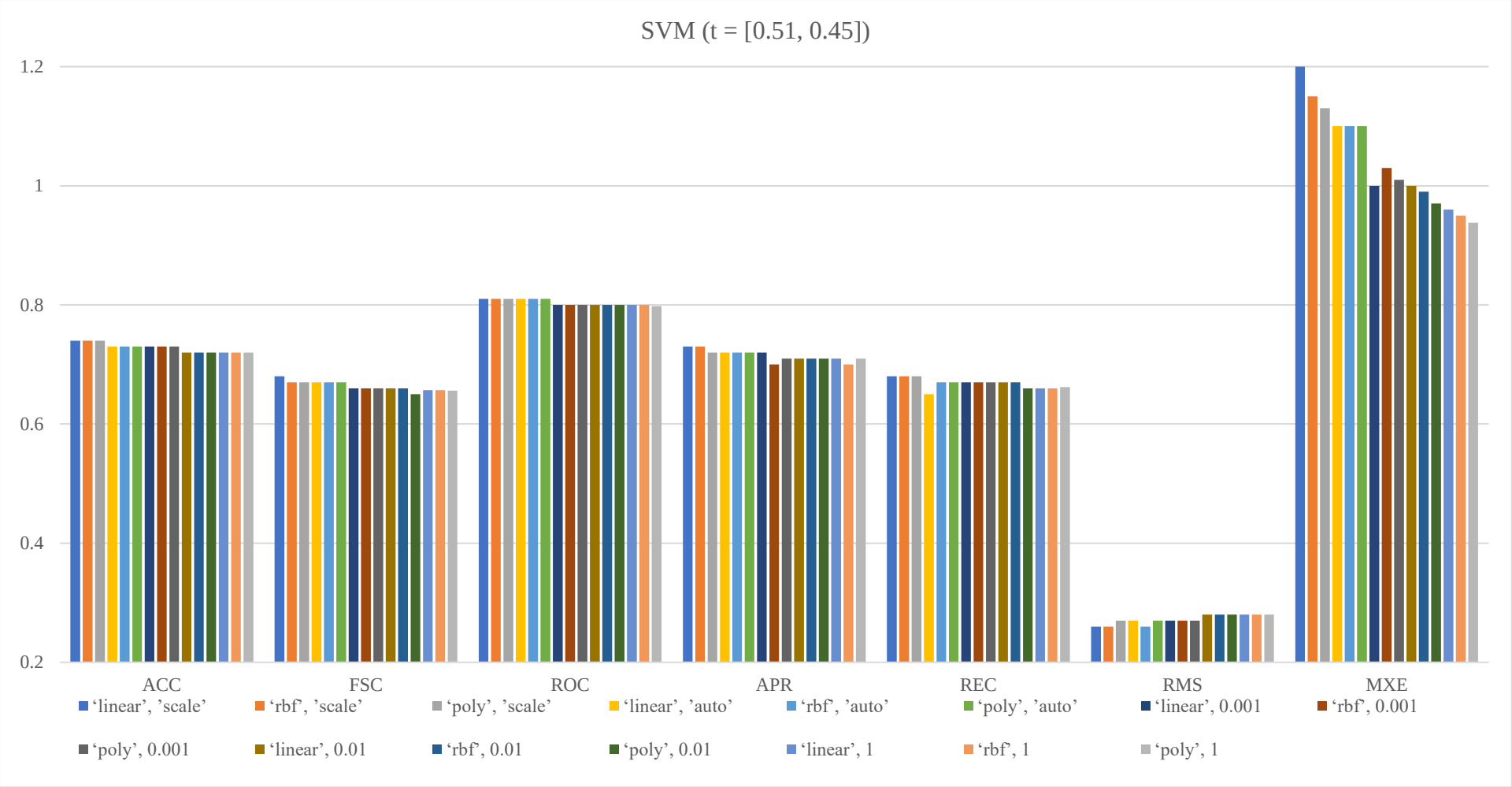

Figure 4. 4 Performance metrics for SVM classifier

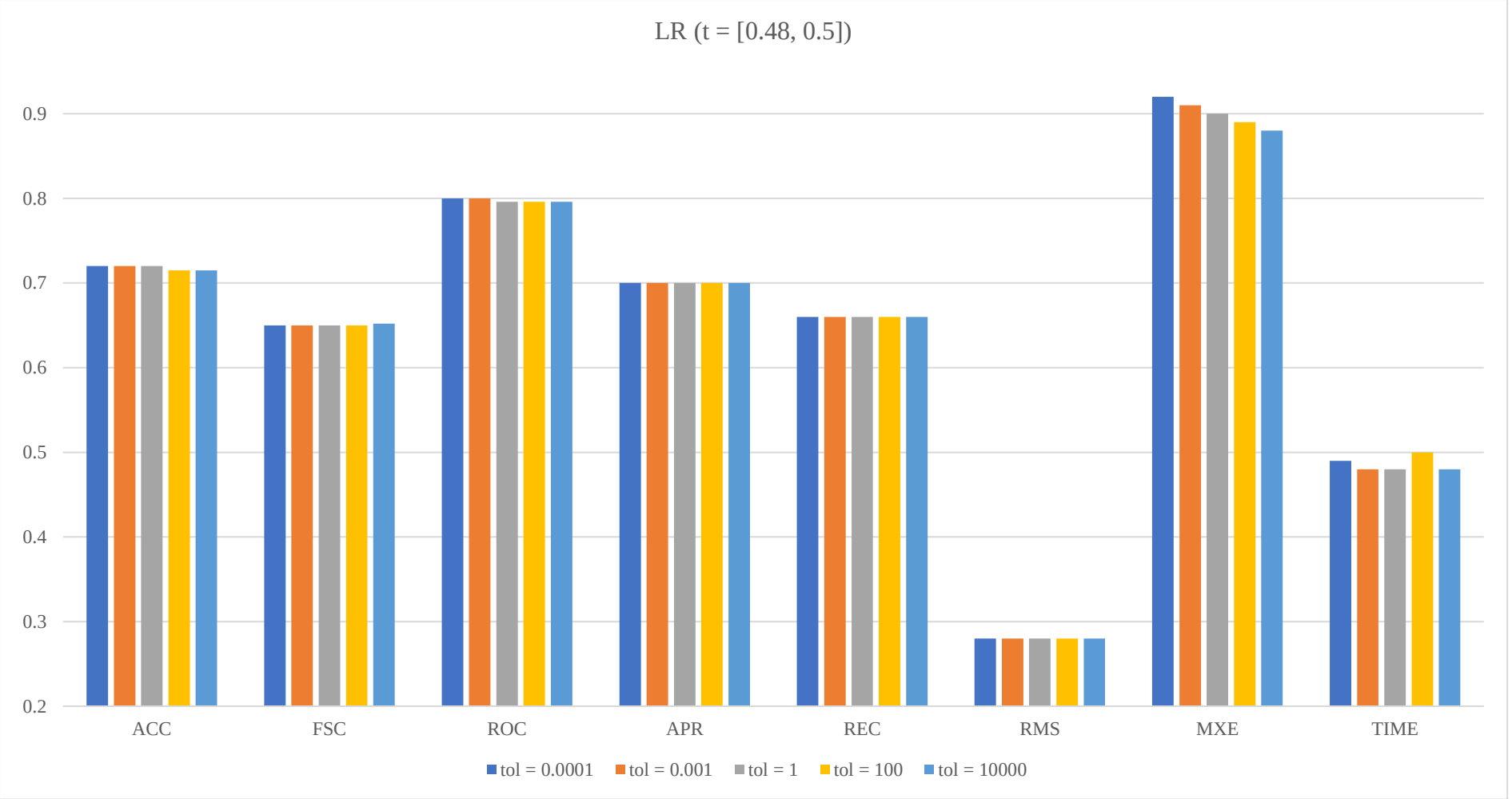

Figure 4. 5 Performance metrics for LR classifier

Table 4. 2 Normalized scores for each learning algorithm by metrics (average over 5-folds)

Model	Combinations	ACC	FSC	ROC	APR	REC	RMS	MXE	MEAN	TIME
RF (num_tree, num_Features)	100,1	0.643	0.570	0.593	0.591	0.609	0.500	0.333	0.548	0.980
	100,6	0.738	0.670	0.701	0.705	0.696	0.375	0.251	0.591	0.927
	100,11	0.810	0.748	0.691	0.773	0.783	0.188	0.186	0.597	0.950
	300,1	0.857	0.783	0.845	0.864	0.739	0.250	0.137	0.639	2.735
	300,6	0.857	0.783	0.845	0.864	0.739	0.250	0.137	0.639	2.735
	300,11	0.925	0.890	0.907	0.919	0.917	0.108	0.078	0.643	2.353
	500,1	0.952	0.913	0.948	0.909	0.913	0.125	0.069	0.690	4.324
	500,6	**1.000**	0.957	**1.000**	**1.000**	**1.000**	0.063	0.029	0.721	4.586
	500,11	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.000**	**0.000**	0.714	4.703
KNN	k=1	0.833	0.714	**1.000**	**1.000**	0.875	0.750	1.000	0.882	**0.000**
	k=20	0.667	0.714	0.706	0.833	0.625	0.750	0.902	0.742	0.006
	k=50	0.500	0.429	0.412	0.500	0.375	0.875	0.902	0.570	0.009
	k=80	0.500	0.429	0.412	0.500	0.375	0.875	0.902	0.570	1.000
SVM (kernel, coeff)	‘linear’, ’scale’	0.500	0.286	0.412	0.500	0.375	0.875	0.853	0.543	0.067
	‘rbf’, ’scale’	0.500	0.286	0.412	0.333	0.375	0.938	0.833	0.525	0.067
	‘poly’, ’scale’	0.333	0.286	0.412	0.333	0.000	0.938	0.804	0.444	0.079
	‘linear’, ’auto’	0.333	0.286	0.412	0.333	0.250	0.875	0.804	0.470	0.070
	‘rbf’, ’auto’	0.333	0.286	0.412	0.333	0.250	0.938	0.804	0.479	0.061
	‘poly’, ’auto’	0.333	0.143	0.118	0.333	0.250	0.938	0.706	0.403	0.055
	‘linear’, 0.001	0.333	0.143	0.118	0.000	0.250	0.938	0.735	0.360	0.070
	‘rbf’, 0.001	0.333	0.143	0.118	0.167	0.250	0.938	0.716	0.381	0.070
	‘poly’, 0.001	0.167	0.143	0.118	0.167	0.250	1.000	0.706	0.364	0.076
	‘linear’, 0.01	0.167	0.143	0.118	0.167	0.250	1.000	0.696	0.363	0.073
	‘rbf’, 0.01	0.167	0.000	0.118	0.167	0.125	1.000	0.676	0.322	0.070
	‘poly’, 0.01	0.167	0.100	0.118	0.167	0.125	1.000	0.667	0.335	0.070
	‘linear’, 1	0.167	0.100	0.118	0.000	0.125	1.000	0.657	0.309	0.055
	‘rbf’, 1	0.167	0.086	0.059	0.167	0.150	1.000	0.645	0.325	0.052
	‘poly’, 1	0.000	0.143	0.059	0.000	0.125	1.000	0.637	0.281	0.052

NB		0.167	0.000	0.118	0.000	0.125	1.000	0.627	0.291	0.061
LR (tol)	tol = 0.0001	0.167	0.000	0.118	0.000	0.125	1.000	0.618	0.290	0.064
	tol = 0.001	0.167	0.000	0.000	0.000	0.125	1.000	0.608	0.271	0.061
	tol = 1	0.083	0.000	0.000	0.000	0.125	1.000	0.598	0.258	0.061
	tol = 100	0.083	0.029	0.000	0.000	0.125	1.000	0.588	0.261	0.067
	tol = 10000	0.083	0.026	0.000	0.000	0.125	1.000	0.582	0.267	0.061

4.4 Evaluation of Generalization Capability of the RF-PDR Mining Model

The effectiveness and generalization capability of our Random Forest (RF)-based dispatching rule mining model were rigorously assessed through extensive testing on new, unseen problem instances. In this section, we present the results of these tests, highlighting the model's ability to predict sequencing schedules for setups within a flexible job shop scheduling environment. To assess the model's generalization prowess, we conducted experiments where we excluded instances generated from one specific problem instance and utilized instances generated from the remaining two problem training datasets.

4.4.1 Observations:

- **Instance FJSP5_C1 (**Perfect Prediction): Remarkably, the RF-Dispatching Rule Mining Model demonstrated outstanding performance on the first instance. It flawlessly predicted the sequencing schedule for all setups, achieving a perfect match with the solutions generated by the optimization solver (Figure 4.6).
- **Instance FJSP5_C** 2 and 3 (Near-Perfect Prediction): In the second instance, the model continued to exhibit a high degree of accuracy. It successfully predicted the sequencing schedule for most setups, aligning perfectly with the solutions obtained

from the solver. However, it is worth noting that there was one sequence (s41 and s32) where the model's prediction diverged slightly from the solver's output. Despite this minor discrepancy, the model's performance remained impressive (Figure 4.7).

In the third new instance, the RF-Dispatching Rule Mining Model once again showcased its robustness and generalization capabilities. It accurately predicted the sequencing schedule for most setups, aligning with the solver-generated solutions. Like the second instance, there was a single sequence (s31 and s12) where the model's prediction deviated slightly from the solver's output (Figure 4.8).

It is noteworthy to emphasize that, in all both cases (C2 and C3), prediction deviation did not violate the natural sequence of setups within the jobs. The results of our testing on new problem instances reaffirm the robustness and generalization capability of the RF-Dispatching Rule Mining Model. These findings underscore the model's adaptability to diverse scheduling scenarios and its ability to consistently provide reliable sequencing recommendations.

The RF-based dispatching rule mining model demonstrates its effectiveness and generalization potential, making it a valuable tool for improving scheduling efficiency in real-world manufacturing environments. Further refinements and ongoing testing with a broader range of instances will continue to enhance its performance and applicability.

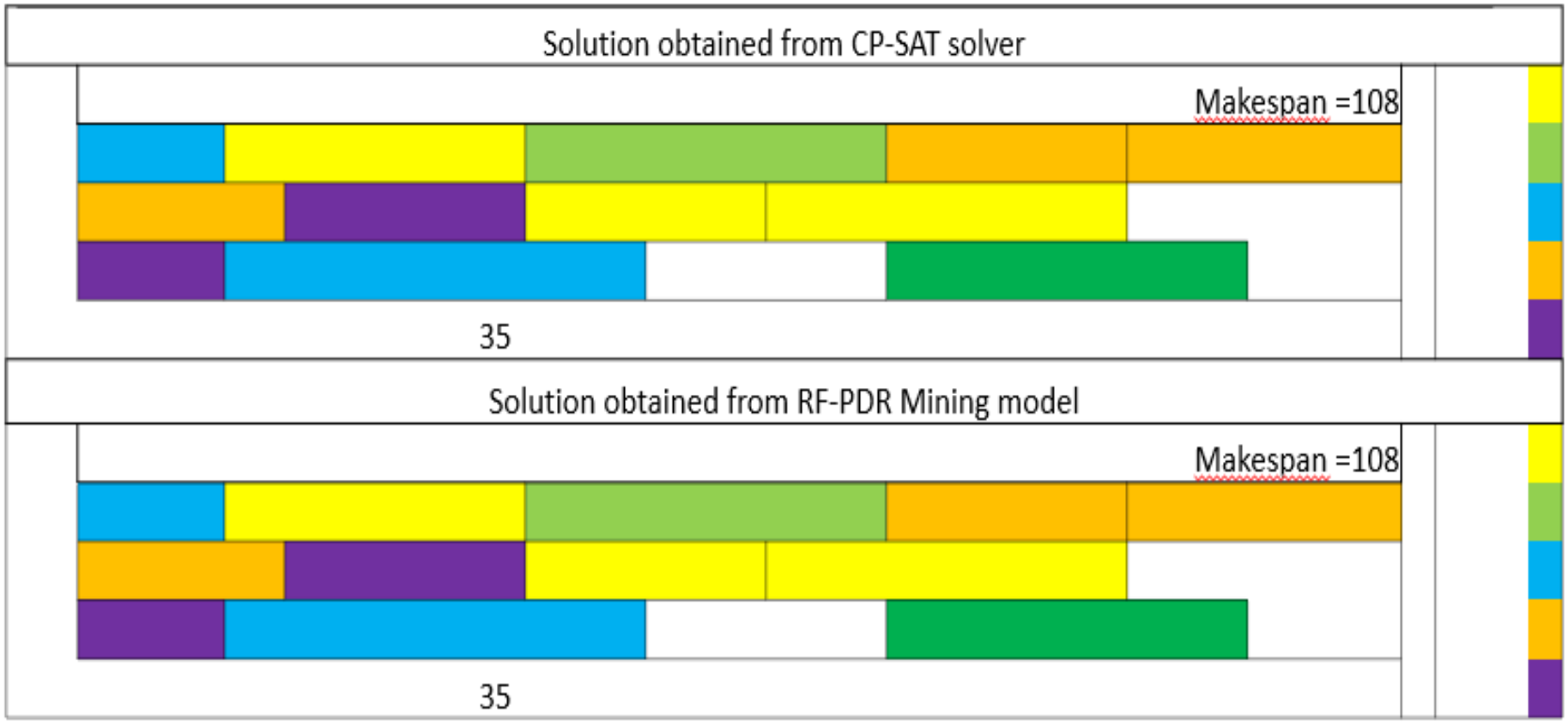

Figure 4. 6 Predicted sequence of Case study 1

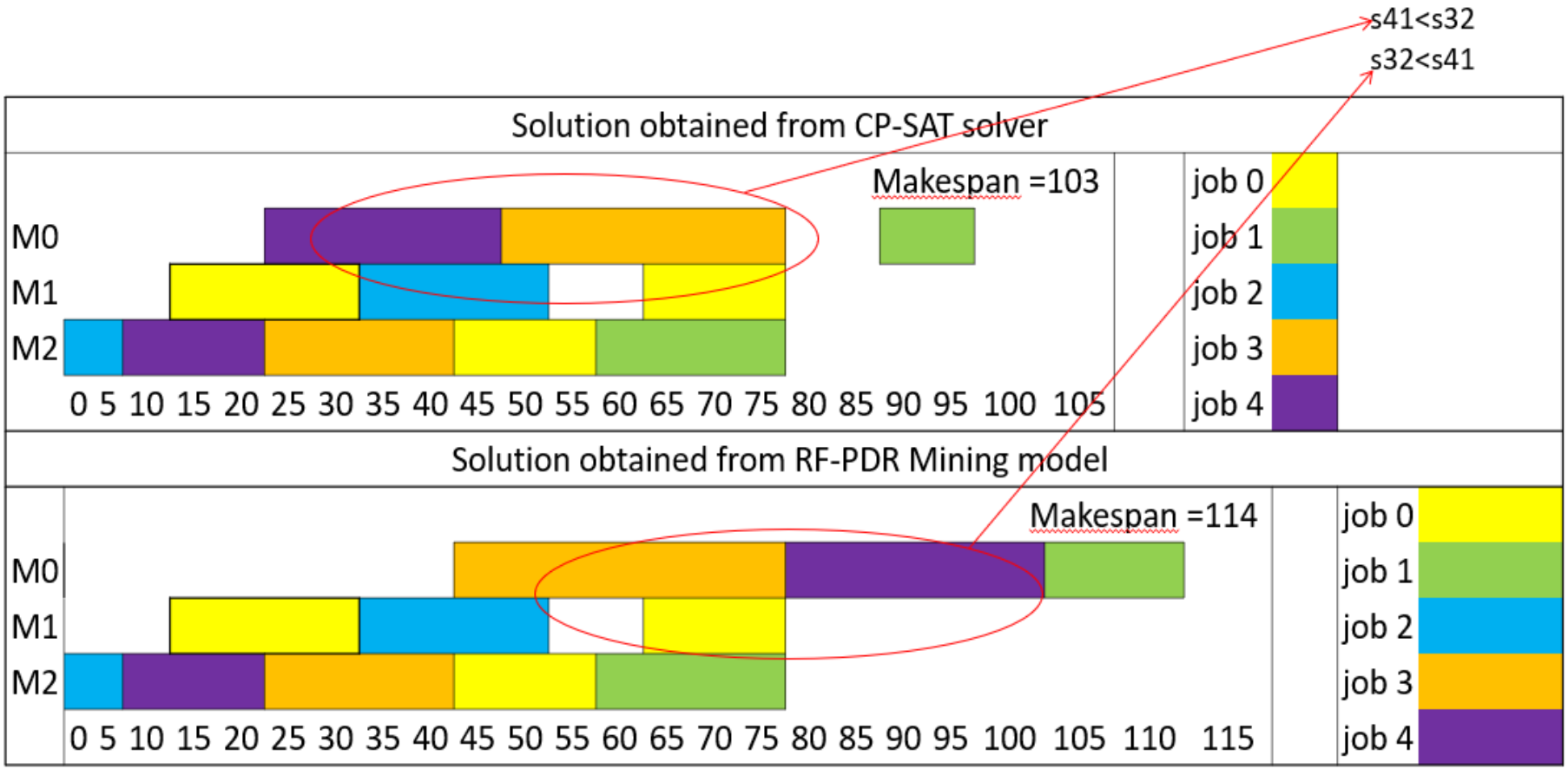

Figure 4. 7 Predicted sequence of Case study 2

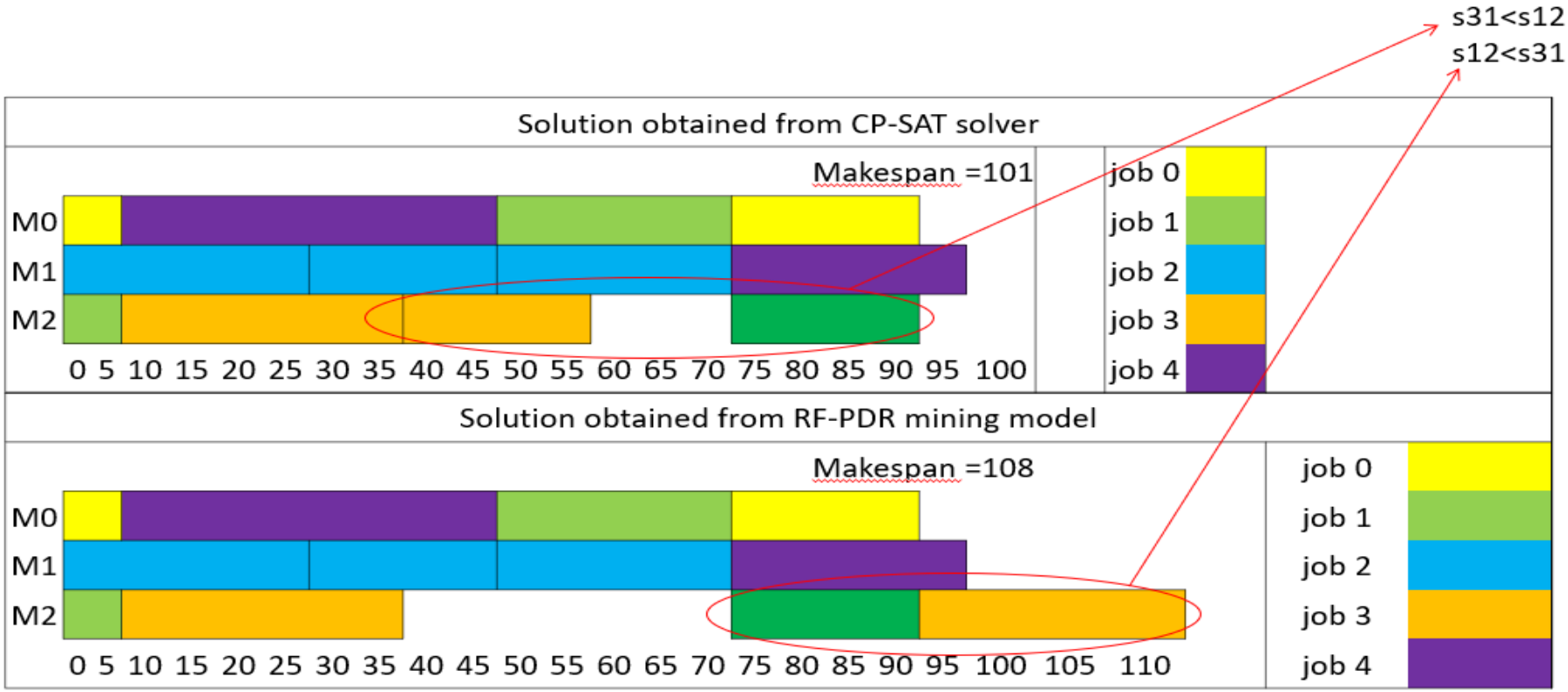

Figure 4. 8 Predicted sequence of Case study 3

4.5 Comparison with Classical Dispatching Rule

To assess the effectiveness of the dispatching rules derived from the RF-PDR (Random Forest-Dispatching Rule) mining model, we conducted a comparison with two well-established classical dispatching rules: Earliest Due Date (EDD) and Shortest Processing Time (SPT). The SPT rule prioritizes tasks or jobs based on their processing times, with the shortest processing time jobs being scheduled first. The rationale behind SPT is to minimize the average waiting time or flow time of jobs in a production system, which can help improve efficiency and reduce lead times. The EDD rule prioritizes tasks or jobs based on their due dates, with jobs having the earliest due dates scheduled first. The EDD rule is particularly useful when the timely delivery of projects or tasks is critical, as it aims to minimize the lateness or tardiness of jobs. The objective of this comparison was to evaluate the performance of the RF-PDR mining model in generating sequencing recommendations for setups within a flexible job shop scheduling environment.

4.5.1 Experimental Setup

In our experiment, we randomly divided the problem instances into training and testing sets, with 60% of the instances used for training and the remaining instances reserved for testing. This partitioning ensured an unbiased evaluation of the dispatching rules on unseen data.

4.5.2 Performance Evaluation

Table 4.3 provides a detailed overview of the makespan (Cmax) for three testing instances, each characterized by the number of jobs (j), the number of machines (k), and the number of setups within each job (i). The table presents the makespan results for the RF-PDR mining model, SPT, and EDD dispatching rules.

RF-PDR vs. SPT:

In the comparison between the RF-PDR mining model and the SPT dispatching rule, it is evident that the RF-PDR model consistently outperforms SPT in terms of makespan. For each of the testing instances, RF-PDR achieves a lower makespan, indicating more efficient scheduling. The percentage deviation between RF-PDR and SPT is also presented, highlighting the significant improvement achieved by the RF-PDR model.

Instance FJSP5_C1: RF-PDR achieves a makespan of 108, while SPT results in a considerably higher makespan of 169, representing a 36% improvement.

Instance FJSP5_C2: RF-PDR once again demonstrates superior performance with a makespan of 114, compared to SPT's makespan of 166, resulting in a 31% improvement.

Instance FJSP5_C3: In this instance, RF-PDR achieves a makespan of 108, whereas SPT yields a makespan of 141, indicating a 23% improvement.

RF-PDR vs. EDD:

Similarly, when comparing the RF-PDR mining model with the EDD dispatching rule, RF-PDR consistently delivers better makespan results. The percentage deviation highlights the superior performance of the RF-PDR model.

Instance FJSP5_C1: RF-PDR achieves a makespan of 108, while EDD results in a makespan of 166, marking a 35% improvement.

Instance FJSP5_C2: RF-PDR's makespan of 114 outperforms EDD's makespan of 198 by 42%.

Instance FJSP5_C3: In this instance, RF-PDR's makespan of 108 is substantially better than EDD's makespan of 169, indicating a 36% improvement.

The RF-PDR mining model exhibits clear superiority in terms of makespan when compared to the classical dispatching rules, SPT and EDD. This demonstrates the potential of data-driven dispatching rules in enhancing scheduling efficiency and optimizing manufacturing operations. Further research can explore the model's performance on a wider range of problem instances and its applicability to real-world manufacturing environments. The superior performance of the dispatching rule obtained from the RF-PDR mining model can be attributed to its adaptability and ability to discover implicit knowledge from production data. Unlike classical dispatching rules, which are often designed for specific manufacturing systems with fixed sequencing criteria, the RF-PDR model leverages attributes derived from real production data. As a result, the RF-PDR model can dynamically adjust its sequencing recommendations based on the unique characteristics of each problem instance, leading to more efficient scheduling. It harnesses the power of machine learning to uncover hidden patterns and correlations within the data, ultimately outperforming traditional dispatching rules.

Table 4. 3 Comparison of mined dispatching rule with SPT and EDD dispatching rule

instance	jxk	i	C_{max}				
			RF-PDR	SPT	% dev	EDD	% dev
FJSP5_C1	5x3	2-3	**108**	169	36%	166	35%
FJSP5_C2	5x3	2-3	**114**	166	31%	198	42%
FJSP5_C3	5x3	2-3	**108**	141	23%	169	36%

4.6 Rescheduling with RF-PDR Mining Model

In the experimental setup designed to evaluate the efficacy of the rescheduling framework, we consider the predicted solution for FJSP_C3 as the initial nominal solution. Table 4.3 represents the solution in a flat data format.

Table 4. 4 Initial nominal solution for case study 3 in a flat data format

setup_id	Eligible machine			Solution		
	m0	m1	m2	Assigned mid	start	end
s00	10	13		0	0	10
s01	20	12	23	0	75	95
s10	25		11	2	0	11
s11	27	18	20	0	48	75
s12	44	26	15	2	75	90
s20	47	30	27	1	0	30
s21	27	18		1	30	48
s22		25	29	1	48	73
s30	35	22	18	2	11	40
s31	19	27	0	2	40	58
s40	38		23	0	10	48
s41	38	28	29	1	73	101

4.6.1 Machine Breakdown Simulation

The simulation model focuses on predicting breakdown times for three machines, parameter for each machine is considered as followed:

Input parameters:

- Mean time between two successive breakdowns:
  - λ_{m1} = 30 hours,
  - λ_{m2} = 80 hours,
  - λ_{m3} = 120 hours
- Threshold, TH_{bk} = 0.7 [59]

Output:

- Breakdown time (Refer to Figure 4.9):

- t_{bd}^{mo} = 45 hours
- t_{bd}^{m1} >120 hours
- t_{bd}^{m2} > 120 hours

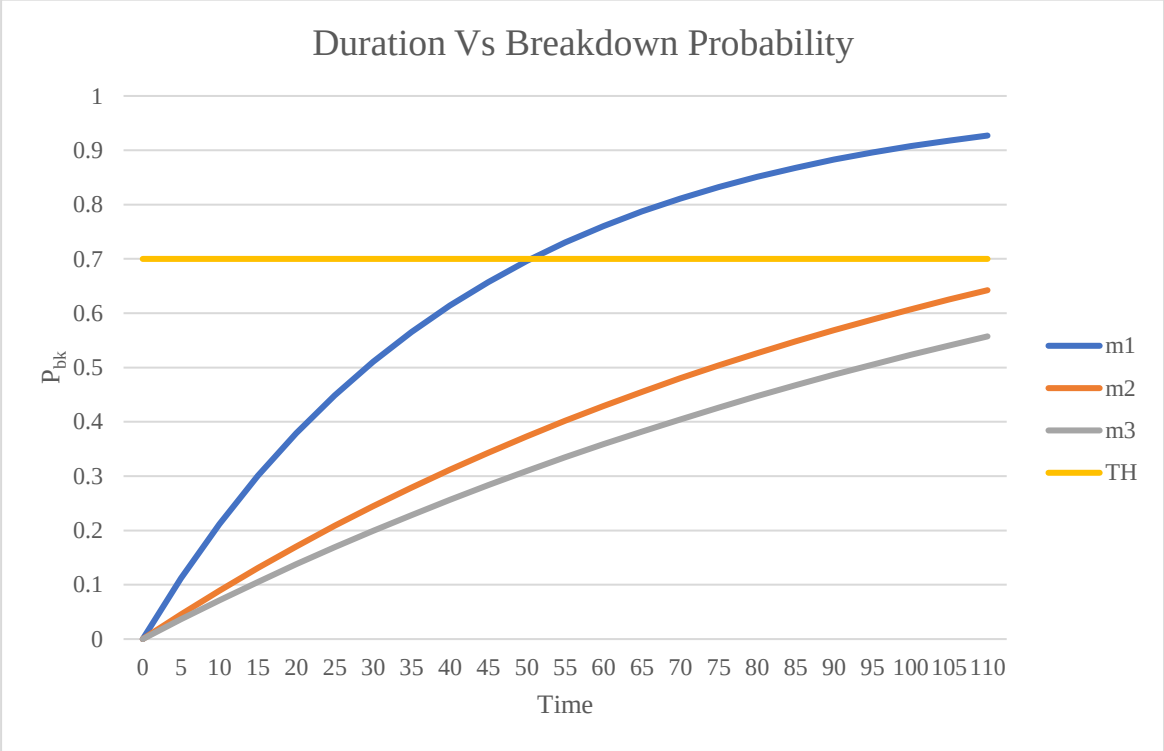

Figure 4. 9 Time of breakdown

4.6.2 Identification of Disrupted Setups

Following the evaluation of the critical criteria, which considers eij > t_{bd}^{m}, we have compiled a list of disrupted setups in conjunction with the presently available machines. This compilation is presented in Table 4.5, wherein a "status" column has been included to categorize the setups into two distinct classifications: "Interrupted" and "Unfinished."

In the context of this table, "Interrupted" setups necessitate reassignment and resequencing on currently eligible machines, while "Unfinished" setups indicate those that have already been assigned and sequenced on the available machines.

Table 4. 5 Setup status after machine breakdown

setup_id	current eligible machine		solution			status
	m1	m2	assigned m_id	start	end	
s01	12	23	-	-	-	Interrupted
s11	18	20	-	-	-	Interrupted
s40		23	-	-	-	Interrupted
s21	18		1	30	48	Unfinished
s22	25	29	1	48	73	Unfinished
s41	28	29	1	73	101	Unfinished
s12	26	15	2	75	90	Unfinished
s31	27	18	2	40	58	Unfinished

4.6.3 Re-scheduling of the Interrupted Setups

In accordance with the localization heuristics approach, the interrupted jobs have been subjected to reassignment. In Table 4.6, the cells that are highlighted denote the updated routing assignments.

Table 4. 6 Updated routing of interrupted setups

setup_id	m1	m2
s01	12	23
s11	18	20
s40		23

Subsequently, a revised sequence for the interrupted setups on eligible machines has been derived utilizing the RF-PDR mining model. This rescheduling solution is visually depicted in Figure 4.10. the shadow block on failed machine stands for idle time interval (time length is equal to repair time). As a result of this rescheduling effort, the makespan has been reduced to 116 hours.When the now broken machine will become operational, unfinished setups then again can be scheduled considering updated machine availability following the same approach.

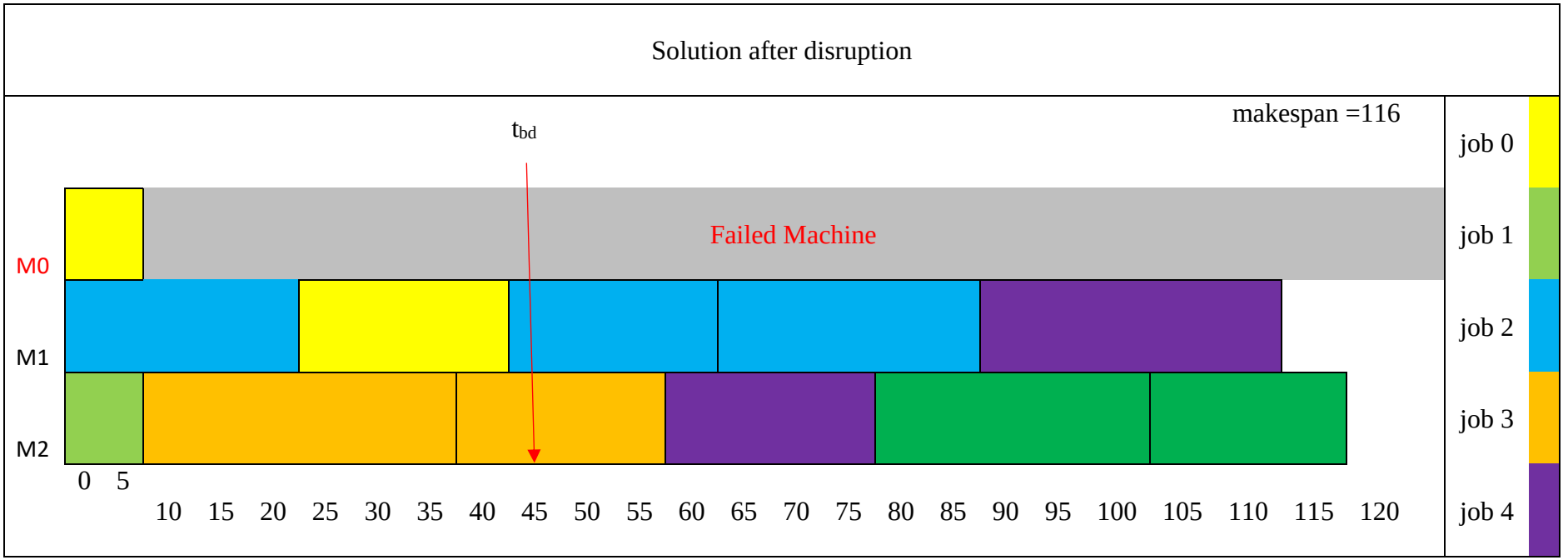

Figure 4. 10 Rescheduling solution

4.6.4 Re-scheduling Robustness & Stability Measure

In order to assess the efficacy of the proposed re-scheduling approach, a comparative evaluation was conducted, juxtaposing the sequenced results obtained through this approach with those derived from two widely adopted classical dispatching rules, namely SPT (Shortest Processing Time) and EDD (Earliest Due Date). Table 4.7 provides a comprehensive depiction of the performance metrics associated with robustness and stability.

Table 4. 7 Comparative Analysis with Classical Dispatching Rules

	C_{max}	RM %	SM
RF-PDR	116	12.93	25.8
SPT	144	33.33	21.4
EDD	152	40.7	52.6

The comparison illustrates that the RF-PDR approach yields the lowest C_{max} value of 116 hours, indicating the shortest completion time among the considered approaches. Additionally, it exhibits the lowest RM%, signifying robustness in minimizing deviations from the optimal solution. Furthermore, the RF-PDR approach boasts a substantial SM value of 25.8, signifying its capability to maintain stability in scheduling operations.

In contrast, the classical dispatching rules, SPT and EDD, exhibit higher C_{max} values, greater RM% deviations, and SM values, suggesting comparatively inferior performance. These findings underscore the superior performance of the RF-PDR model in achieving efficient and stable re-scheduling outcome.

CHAPTER 5

CONCLUSION AND FUTURE RESEARCH DIRECTION

5.1 Conclusion

This research is undertaken with the aim of tackling the integrated Computer-Aided Process Planning (CAPP) and Scheduling problem for Smart Manufacturing (SM). Considering the increasing demand for customization in response to customer needs, there has arisen a pressing requirement for real-time and adaptable production planning and scheduling strategies in the manufacturing sector. Traditional sequential approaches of handling Process Planning (PP) and Scheduling, have often resulted in conflicting objectives, leading to inefficiencies within the production environment.

To confront these challenges, we proposed an innovative approach combining machine learning and optimization techniques. The key accomplishments of this research can be summarized as follows:

- This study delves into the Integrated CAPP and Scheduling problem within a multipart-multimachine setting, bridging a notable gap in the existing literature. It offers a comprehensive one-shot solution to the complex CAPP and dynamic scheduling problem.
- To the best of our knowledge, this research marks the pioneering effort to treat setups as the fundamental dispatching units for scheduling to resolve the conflict between process planning and scheduling objectives.
- The introduced dispatching rule mining model exhibits the capacity to acquire sequencing knowledge from optimized solutions and implicit insights from production data. It emerges as a robust and dependable solution provider for both scheduling and re-scheduling.

In conclusion, this research endeavors to pave the way for more efficient, responsive, and holistic manufacturing processes by integrating process planning and scheduling within the context of Smart Manufacturing. The fusion of machine learning and optimization techniques offers promising prospects for addressing the complexities of modern manufacturing environments and meeting the ever-evolving demands of customers.

5.2 Future Research Direction

The research on " Integrated CAPP and Scheduling using a Combined ML and Optimization Approach for Smart Manufacturing" lays a solid foundation for addressing complex challenges of the process planning and scheduling function. However, several avenues for future work can further enhance the understanding, application, and impact of the proposed approach:

- In our proposed approach, it is important to note that the generation of an optimal routing has not been explicitly addressed within the scope of this research. Instead, we have adopted a heuristic approach for the assignment of setups, where the attainment of optimality in the initial nominal solution is not guaranteed. Consequently, this heuristic assignment process can impact the quality of the sequencing solution. These observations underscore the need for future research endeavors to investigate and assess the influence of the initial optimal schedule's quality on the subsequent stages of the integrated process.
- Another promising avenue for future research lies in addressing the routing sub-problem through the utilization of unsupervised learning techniques. This could potentially enhance the efficiency and effectiveness of the overall approach by autonomously discovering optimal routing strategies.

- In this research, we have employed resampling method (k-fold cross validation) for direct comparison. Thus, future research can include statistical significance test to quantify the likelihood of the performance metrics being drawn from same samples.
- Furthermore, it is imperative to acknowledge that in the context of rescheduling, our research has not taken into account certain scenarios such as uncertain repair times, instances involving multiple breakdowns, and setup pre-emption. Future research initiatives could extend their focus to include these complex scenarios, thereby enriching the applicability and robustness of the proposed rescheduling strategy.

REFERENCES/BIBLIOGRAPHY

[1] H. Besharati-Foumani, M. Lohtander, and J. Varis, "Intelligent process planning for smart manufacturing systems: A state-of-the-art review," *Procedia Manuf*, vol. 38, no. 2019, pp. 156–162, 2019, doi: 10.1016/j.promfg.2020.01.021.

[2] M. Trstenjak and P. Cosic, "Process Planning in Industry 4.0 Environment," *Procedia Manuf*, vol. 11, no. June, pp. 1744–1750, 2017, doi: 10.1016/j.promfg.2017.07.303.

[3] M. Al-wswasi, A. Ivanov, and H. Makatsoris, "A survey on smart automated computer-aided process planning (ACAPP) techniques," *International Journal of Advanced Manufacturing Technology*, vol. 97, no. 1–4, pp. 809–832, 2018, doi: 10.1007/s00170-018-1966-1.

[4] W. J. Zhang and S. Q. Xie, "Agent technology for collaborative process planning: A review," *International Journal of Advanced Manufacturing Technology*, vol. 32, no. 3–4, pp. 315–325, 2007, doi: 10.1007/s00170-005-0345-x.

[5] W. Shen, L. Wang, and Q. Hao, "Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 36, no. 4, pp. 563–577, 2006, doi: 10.1109/TSMCC.2006.874022.

[6] X. Li and L. Gao, "Review for Integrated Process Planning and Scheduling," *Engineering Applications of Computational Methods*, vol. 2, no. 2, pp. 47–59, 2020, doi: 10.1007/978-3-662-55305-3_3.

[7] Y. Zhang, X. Yu, J. Sun, Y. Zhang, X. Xu, and Y. Gong, “Intelligent STEP-NC-compliant setup planning method,” *J Manuf Syst*, vol. 62, pp. 62–75, Jan. 2022, doi: 10.1016/j.jmsy.2021.11.002.

[8] H. A. Elmaraghy *et al.*, “Evolution and Future Perspectives of CAPP With contributions from: E. A g e r m a n.”

[9] A. Azab, G. Perusi, H. Elmaraghy, and J. Urbanic, “SEMI-GENERATIVE MACRO-PROCESS PLANNING FOR RECONFIGURABLE MANUFACTURING.”

[10] W. Wu, Z. Huang, K. Wu, and Y. Chen, “An optimization approach for setup planning and operation sequencing with tolerance constraints,” *International Journal of Advanced Manufacturing Technology*, vol. 106, no. 11–12, pp. 4965–4985, Feb. 2020, doi: 10.1007/s00170-019-04791-y.

[11] R. S. Joshi, N. Kumar, and A. Sharma, “Setup planning and operation sequencing using neural network and genetic algorithm,” in *Proceedings - International Conference on Information Technology: New Generations, ITNG 2008*, 2008, pp. 396–401. doi: 10.1109/ITNG.2008.94.

[12] “Intelligent setup planning in manufacturing by neural networks based approach,” 2000.

[13] M. Ameer and M. Dahane, “Reconfigurability improvement in Industry 4.0: a hybrid genetic algorithm-based heuristic approach for a co-generation of setup and process plans in a reconfigurable environment,” *J Intell Manuf*, vol. 34, no. 3, pp. 1445–1467, Mar. 2023, doi: 10.1007/s10845-021-01869-x.

[14] R. Barzanji, B. Naderi, and M. A. Begen, “Decomposition algorithms for the integrated process planning and scheduling problem,” *Omega (United Kingdom)*, vol. 93, Jun. 2020, doi: 10.1016/j.omega.2019.01.003.

[15] X. Wu and J. Li, “Two layered approaches integrating harmony search with genetic algorithm for the integrated process planning and scheduling problem,” *Comput Ind Eng*, vol. 155, May 2021, doi: 10.1016/j.cie.2021.107194.

[16] P. Mohapatra, S. Nanda, and S. Maji, “DNA based approach: Integration of process planning and scheduling,” in *Proceedings of 2015 IEEE 9th International Conference on Intelligent Systems and Control, ISCO 2015*, Institute of Electrical and Electronics Engineers Inc., Sep. 2015. doi: 10.1109/ISCO.2015.7282253.

[17] D. Alemão, A. D. Rocha, and J. Barata, “Smart manufacturing scheduling approaches—systematic review and future directions,” *Applied Sciences (Switzerland)*, vol. 11, no. 5, pp. 1–20, 2021, doi: 10.3390/app11052186.

[18] M. Parente, G. Figueira, P. Amorim, and A. Marques, “Production scheduling in the context of Industry 4.0: review and trends,” *Int J Prod Res*, vol. 58, no. 17, pp. 5401–5431, 2020, doi: 10.1080/00207543.2020.1718794.

[19] P. Wenzelburger and F. Allgöwer, “Model predictive control for flexible job shop scheduling in industry 4.0†,” *Applied Sciences (Switzerland)*, vol. 11, no. 17, 2021, doi: 10.3390/app11178145.

[20] Y. Liu, L. Wang, X. V. Wang, X. Xu, and L. Zhang, “Scheduling in cloud manufacturing: state-of-the-art and research challenges,” *Int J Prod Res*, vol. 57, no. 15–16, pp. 4854–4879, 2019, doi: 10.1080/00207543.2018.1449978.

[21] M. Haddadzade, M. R. Razfar, and M. H. F. Zarandi, “Multipart setup planning through integration of process planning and scheduling,” *Proc Inst Mech Eng B J Eng Manuf*, vol. 230, no. 6, pp. 1097–1113, Jun. 2016, doi: 10.1177/0954405414565138.

[22] P. Mohapatra, L. Benyoucef, and M. K. Tiwari, “Integration of process planning and scheduling through adaptive setup planning: A multi-objective approach,” *Int J Prod Res*, vol. 51, no. 23–24, pp. 7190–7208, Nov. 2013, doi: 10.1080/00207543.2013.853890.

[23] P. Mohapatra, L. Benyoucef, and M. K. Tiwari, “Realising process planning and scheduling integration through adaptive setup planning,” *Int J Prod Res*, vol. 51, no. 8, pp. 2301–2323, 2013, doi: 10.1080/00207543.2012.715770.

[24] P. Mohapatra, N. Kumar, A. Matta, and M. K. Tiwari, “A nested partitioning-based approach to integrate process planning and scheduling in flexible manufacturing environment,” *Int J Comput Integr Manuf*, vol. 28, no. 10, pp. 1077–1091, Oct. 2015, doi: 10.1080/0951192X.2014.961548.

[25] N. Cai, L. Wang, and H. Y. Feng, “GA-based adaptive setup planning toward process planning and scheduling integration,” *Int J Prod Res*, vol. 47, no. 10, pp. 2745–2766, Jan. 2009, doi: 10.1080/00207540701663516.

[26] M. Kumar and S. Rajotia, “Integration of scheduling with computer aided process planning,” in *Journal of Materials Processing Technology*, Jul. 2003, pp. 297–300. doi: 10.1016/S0924-0136(03)00088-8.

[27] N. Cai, L. Wang, and H. Y. Feng, “Adaptive setup planning of prismatic parts for machine tools with varying configurations,” *Int J Prod Res*, vol. 46, no. 3, pp. 571–594, Feb. 2008, doi: 10.1080/00207540600849125.

[28] L. Wang, H.-Y. Feng, N. Cai, and J. Ma, “8 Adaptive Setup Planning for Job Shop Operations under Uncertainty.”

[29] L. Wang, N. Cai, H. Y. Feng, and J. Ma, “ASP: An adaptive setup planning approach for dynamic machine assignments,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 2–14, Jan. 2010, doi: 10.1109/TASE.2008.2011919.

[30] O. Hua-Bing, “A STEP-Compliant Intelligent Process Planning System for Milling,” 2015.

[31] L. X. Phung, D. Van Tran, S. V. Hoang, and S. H. Truong, “Effective method of operation sequence optimization in CAPP based on modified clustering algorithm,” *Journal of Advanced Mechanical Design, Systems and Manufacturing*, vol. 11, no. 1, 2017, doi: 10.1299/jamdsm.2017jamdsm0001.

[32] M. Hazarika, S. Deb, U. S. Dixit, and J. P. Davim, “Fuzzy set-based set-up planning system with the ability for online learningg,” *Proc Inst Mech Eng B J Eng Manuf*, vol. 225, no. 2, pp. 247–263, Feb. 2011, doi: 10.1243/09544054JEM1867.

[33] H. Hajimiri, M. H. Siahmargouei, H. Ghorbani, and M. Shakeri, “A simple and robust setup planning scheme for prismatic workpieces,” *CIRP J Manuf Sci Technol*, vol. 19, pp. 164–175, Nov. 2017, doi: 10.1016/j.cirpj.2017.07.002.

[34] D. Manafi and M. J. Nategh, “Reducing search space of optimization algorithms for determination of machining sequences by consolidating decisive agents,” *Proc Inst Mech Eng B J Eng Manuf*, vol. 234, no. 6–7, pp. 1057–1068, May 2020, doi: 10.1177/0954405419896118.

[35] D. Manafi and M. J. Nategh, “Integrating the setup planning with fixture design practice by concurrent consideration of machining and fixture design principles,” *Int J Prod Res*, vol. 59, no. 9, pp. 2647–2666, 2021, doi: 10.1080/00207543.2020.1736357.

[36] D. Manafi and M. J. Nategh, “Optimization of Setup Planning by Combined Permutation-Based and Simulated Annealing Algorithms,” *Arab J Sci Eng*, vol. 48, no. 3, pp. 3697–3708, Mar. 2023, doi: 10.1007/s13369-022-07209-2.

[37] L. Renke, R. Piplani, and C. Toro, “A Review of Dynamic Scheduling: Context, Techniques and Prospects,” in *Intelligent Systems Reference Library*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 229–258. doi: 10.1007/978-3-030-67270-6_9.

[38] C. Ferreira, G. Figueira, and P. Amorim, “Effective and interpretable dispatching rules for dynamic job shops via guided empirical learning,” *Omega (United Kingdom)*, vol. 111, p. 102643, 2022, doi: 10.1016/j.omega.2022.102643.

[39] A. Azab and H. A. ElMaraghy, “Mathematical Modeling for Reconfigurable Process Planning,” *CIRP Ann Manuf Technol*, vol. 56, no. 1, pp. 467–472, 2007, doi: 10.1016/j.cirp.2007.05.112.

[40] P. Priore, A. Gómez, R. Pino, and R. Rosillo, “Dynamic scheduling of manufacturing systems using machine learning: An updated review,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, vol. 28, no. 1, pp. 83–97, 2014, doi: 10.1017/S0890060413000516.

[41] D. Ouelhadj and S. Petrovic, “A survey of dynamic scheduling in manufacturing systems,” *Journal of Scheduling*, vol. 12, no. 4, pp. 417–431, 2009, doi: 10.1007/s10951-008-0090-8.

[42] L. Zhang, Y. Hu, C. Wang, Q. Tang, and X. Li, “Effective dispatching rules mining based on near-optimal schedules in intelligent job shop environment,” *J Manuf Syst*, vol. 63, pp. 424–438, Apr. 2022, doi: 10.1016/j.jmsy.2022.04.019.

[43] S. Jun, S. Lee, and H. Chun, “Learning dispatching rules using random forest in flexible job shop scheduling problems,” *Int J Prod Res*, vol. 57, no. 10, pp. 3290–3310, May 2019, doi: 10.1080/00207543.2019.1581954.

[44] S. Jun and S. Lee, “Learning dispatching rules for single machine scheduling with dynamic arrivals based on decision trees and feature construction,” *Int J Prod Res*, vol. 59, no. 9, pp. 2838–2856, 2021, doi: 10.1080/00207543.2020.1741716.

[45] P. Kianpour, D. Gupta, K. K. Krishnan, and B. Gopalakrishnan, “Automated job shop scheduling with dynamic processing times and due dates using project management and industry 4.0,” *Journal of Industrial and Production Engineering*, vol. 38, no. 7, pp. 485–498, 2021, doi: 10.1080/21681015.2021.1937725.

[46] S. Zhang, F. Tang, X. Li, J. Liu, and B. Zhang, “A hybrid multi-objective approach for real-time flexible production scheduling and rescheduling under dynamic environment in Industry 4.0 context,” *Comput Oper Res*, vol. 132, no. March, p. 105267, 2021, doi: 10.1016/j.cor.2021.105267.

[47] A. Shahzad and N. Mebarki, “Data mining based job dispatching using hybrid simulation-optimization approach for shop scheduling problem,” *Eng Appl Artif Intell*, vol. 25, no. 6, pp. 1173–1181, Sep. 2012, doi: 10.1016/j.engappai.2012.04.001.

[48] A. Zhao *et al.*, “Data-Mining-Based Real-Time Optimization of the Job Shop Scheduling Problem,” *Mathematics*, vol. 10, no. 23, Dec. 2022, doi: 10.3390/math10234608.

[49] G. Metan, I. Sabuncuoglu, and H. Pierreval, “Real time selection of scheduling rules and knowledge extraction via dynamically controlled data mining,” *Int J Prod Res*, vol. 48, no. 23, pp. 6909–6938, Dec. 2010, doi: 10.1080/00207540903307581.

[50] M. Habib Zahmani and B. Atmani, “Multiple dispatching rules allocation in real time using data mining, genetic algorithms, and simulation,” *Journal of Scheduling*, vol. 24, no. 2, pp. 175–196, Apr. 2021, doi: 10.1007/s10951-020-00664-5.

[51] S. Olafsson and X. Li, “Learning effective new single machine dispatching rules from optimal scheduling data,” in *International Journal of Production Economics*, Nov. 2010, pp. 118–126. doi: 10.1016/j.ijpe.2010.06.004.

[52] F. Pezzella, G. Morganti, and G. Ciaschetti, “A genetic algorithm for the Flexible Job-shop Scheduling Problem,” *Comput Oper Res*, vol. 35, no. 10, pp. 3202–3212, Oct. 2008, doi: 10.1016/j.cor.2007.02.014.

[53] A. Vital-Soto, A. Azab, and M. F. Baki, “Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility,” *J Manuf Syst*, vol. 54, pp. 74–93, Jan. 2020, doi: 10.1016/j.jmsy.2019.11.010.

[54] G. Da Col and E. C. Teppan, “Google vs IBM: A constraint solving challenge on the job-shop scheduling problem,” in *Electronic Proceedings in Theoretical*

Computer Science, EPTCS, Open Publishing Association, Sep. 2019, pp. 259–265. doi: 10.4204/EPTCS.306.30.

[55] G. Da Col and E. C. Teppan, "Industrial Size Job Shop Scheduling Tackled by Present Day CP Solvers," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2019, pp. 144–160. doi: 10.1007/978-3-030-30048-7_9.

[56] X. Li and S. Olafsson, "DISCOVERING DISPATCHING RULES USING DATA MINING," 2005.

[57] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC Med Inform Decis Mak*, vol. 19, no. 1, Dec. 2019, doi: 10.1186/s12911-019-1004-8.

[58] R. Caruana, "An Empirical Comparison of Supervised Learning Algorithms." [Online]. Available: www.cs.cornell.edu

[59] W. He and D. H. Sun, "Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies," *International Journal of Advanced Manufacturing Technology*, vol. 66, no. 1–4, pp. 501–514, Apr. 2013, doi: 10.1007/s00170-012-4344-4.

APPENDIX

Appendix 1. Generated Problem Instances for Case Studies

Case study 1					
job_id	setup_id	m0	m1	m2	job_due
0	0	23	12		49.2
0	1		21	28	49.2
0	2		27	12	49.2
1	0	31	28		44.4
1	1		23	29	44.4
2	0	16		18	45.2
2	1	49		30	45.2
3	0	41	18	14	48
3	1	19	20	26	48
3	2	19	12	11	48
4	0	38		18	42.8
4	1		21	30	42.8

Case study 2

job_id	setup_id	m0	m1	m2	job_due
0	0	27	17	21	71
0	1		26	16	71
0	2	48	19	18	71
1	0		15	30	36.83333
1	1	13	20	10	36.83333
2	0	25		10	48.5
2	1	47	18	28	48.5
3	0		15	19	73
3	1		28	22	73
3	2	31			73
4	0	21		12	35
4	1	26	11		35

Case study 3

job_id	setup_id	m0	m1	m2	job_due
0	0	10	13		36
0	1	20	12	23	36
1	0	25		11	82
1	1	27	18	20	82

1	2	44	26	15	82
2	0	47	30	27	101
2	1	27	18		101
2	2		25	29	101
3	0	35	22	18	49
3	1	19	27	0	49
4	0	38		23	75
4	1	38	28	29	75

Appendix 2 Created training dataset for rule mining

Case study 1

A	B	p_A	d_A	p_B	d_B	Xii'k	Zii'j	Zij>Zi'j	p_A>B	d_A>B	p_A-B	d_A-B	class
s00	s01	23	49.2	21	49.2	0	1	1	1	0	2	0	1
s00	s02	23	49.2	27	49.2	0	1	1	0	0	-4	0	1
s00	s10	23	49.2	31	44.4	1	0	0	0	1	-8	4.8	1
s00	s11	23	49.2	29	44.4	0	0	0	0	1	-6	4.8	1
s00	s20	23	49.2	16	45.2	1	0	0	1	1	7	4	0
s00	s30	23	49.2	18	48	0	0	0	1	1	5	1.2	0
s00	s31	23	49.2	19	48	1	0	0	1	1	4	1.2	1
s00	s32	23	49.2	19	48	1	0	0	1	1	4	1.2	1
s00	s40	23	49.2	18	42.8	0	0	0	1	1	5	6.4	0
s00	s41	23	49.2	21	42.8	0	0	0	1	1	2	6.4	1
s01	s02	21	49.2	27	49.2	1	1	1	0	0	-6	0	1
s01	s10	21	49.2	31	44.4	0	0	0	0	1	-10	4.8	0
s01	s11	21	49.2	29	44.4	0	0	0	0	1	-8	4.8	1
s01	s20	21	49.2	16	45.2	0	0	0	1	1	5	4	0
s01	s30	21	49.2	18	48	1	0	0	1	1	3	1.2	0
s01	s31	21	49.2	19	48	0	0	0	1	1	2	1.2	1
s01	s32	21	49.2	19	48	0	0	0	1	1	2	1.2	1
s01	s40	21	49.2	18	42.8	0	0	0	1	1	3	6.4	0
s01	s41	21	49.2	21	42.8	1	0	0	0	1	0	6.4	0
s02	s10	27	49.2	31	44.4	0	0	0	0	1	-4	4.8	0
s02	s11	27	49.2	29	44.4	0	0	0	0	1	-2	4.8	1
s02	s20	27	49.2	16	45.2	0	0	0	1	1	11	4	0
s02	s30	27	49.2	18	48	1	0	0	1	1	9	1.2	0
s02	s31	27	49.2	19	48	0	0	0	1	1	8	1.2	1
s02	s32	27	49.2	19	48	0	0	0	1	1	8	1.2	1
s02	s40	27	49.2	18	42.8	0	0	0	1	1	9	6.4	0
s02	s41	27	49.2	21	42.8	1	0	0	1	1	6	6.4	0
s10	s11	31	44.4	29	44.4	0	1	1	1	0	2	0	1

s10	s20	31	44.4	16	45.2	1	0	0	1	0	15	-0.8	0
s10	s30	31	44.4	18	48	0	0	0	1	0	13	-3.6	0
s10	s31	31	44.4	19	48	1	0	0	1	0	12	-3.6	1
s10	s32	31	44.4	19	48	1	0	0	1	0	12	-3.6	1
s10	s40	31	44.4	18	42.8	0	0	0	1	1	13	1.6	0
s10	s41	31	44.4	21	42.8	0	0	0	1	1	10	1.6	0
s11	s20	29	44.4	16	45.2	0	0	0	1	0	13	-0.8	0
s11	s30	29	44.4	18	48	0	0	0	1	0	11	-3.6	0
s11	s31	29	44.4	19	48	0	0	0	1	0	10	-3.6	0
s11	s32	29	44.4	19	48	0	0	0	1	0	10	-3.6	1
s11	s40	29	44.4	18	42.8	1	0	0	1	1	11	1.6	0
s11	s41	29	44.4	21	42.8	0	0	0	1	1	8	1.6	0
s20	s30	16	45.2	18	48	0	0	0	0	0	-2	-2.8	0
s20	s31	16	45.2	19	48	1	0	0	0	0	-3	-2.8	1
s20	s32	16	45.2	19	48	1	0	0	0	0	-3	-2.8	1
s20	s40	16	45.2	18	42.8	0	0	0	0	1	-2	2.4	0
s20	s41	16	45.2	21	42.8	0	0	0	0	1	-5	2.4	1
s30	s31	18	48	19	48	0	1	1	0	0	-1	0	1
s30	s32	18	48	19	48	0	1	1	0	0	-1	0	1
s30	s40	18	48	18	42.8	0	0	0	0	1	0	5.2	0
s30	s41	18	48	21	42.8	1	0	0	0	1	-3	5.2	1
s31	s32	19	48	19	48	1	1	1	0	0	0	0	1
s31	s40	19	48	18	42.8	0	0	0	1	1	1	5.2	0
s31	s41	19	48	21	42.8	0	0	0	0	1	-2	5.2	0
s32	s40	19	48	18	42.8	0	0	0	1	1	1	5.2	0
s32	s41	19	48	21	42.8	0	0	0	0	1	-2	5.2	0
s40	s41	18	42.8	21	42.8	0	1	1	0	0	-3	0	1
s01	s00	21	49.2	23	49.2	0	1	-1	0	0	-2	0	0
s02	s00	27	49.2	23	49.2	0	1	-1	1	0	4	0	0
s10	s00	31	44.4	23	49.2	1	0	0	1	0	8	-4.8	0
s11	s00	29	44.4	23	49.2	0	0	0	1	0	6	-4.8	0
s20	s00	16	45.2	23	49.2	1	0	0	0	0	-7	-4	1
s30	s00	18	48	23	49.2	0	0	0	0	0	-5	-1.2	1
s31	s00	19	48	23	49.2	1	0	0	0	0	-4	-1.2	0
s32	s00	19	48	23	49.2	1	0	0	0	0	-4	-1.2	0
s40	s00	18	42.8	23	49.2	0	0	0	0	0	-5	-6.4	1
s41	s00	21	42.8	23	49.2	0	0	0	0	0	-2	-6.4	0
s02	s01	27	49.2	21	49.2	1	1	-1	1	0	6	0	0
s10	s01	31	44.4	21	49.2	0	0	0	1	0	10	-4.8	0
s11	s01	29	44.4	21	49.2	0	0	0	1	0	8	-4.8	0
s20	s01	16	45.2	21	49.2	0	0	0	0	0	-5	-4	1
s30	s01	18	48	21	49.2	1	0	0	0	0	-3	-1.2	1
s31	s01	19	48	21	49.2	0	0	0	0	0	-2	-1.2	0

s32	s01	19	48	21	49.2	0	0	0	0	0	-2	-1.2	0
s40	s01	18	42.8	21	49.2	0	0	0	0	0	-3	-6.4	1
s41	s01	21	42.8	21	49.2	1	0	0	0	0	0	-6.4	1
s10	s02	31	44.4	27	49.2	0	0	0	1	0	4	-4.8	1
s11	s02	29	44.4	27	49.2	0	0	0	1	0	2	-4.8	0
s20	s02	16	45.2	27	49.2	0	0	0	0	0	-11	-4	1
s30	s02	18	48	27	49.2	1	0	0	0	0	-9	-1.2	1
s31	s02	19	48	27	49.2	0	0	0	0	0	-8	-1.2	0
s32	s02	19	48	27	49.2	0	0	0	0	0	-8	-1.2	0
s40	s02	18	42.8	27	49.2	0	0	0	0	0	-9	-6.4	1
s41	s02	21	42.8	27	49.2	1	0	0	0	0	-6	-6.4	1
s11	s10	29	44.4	31	44.4	0	1	-1	0	0	-2	0	0
s20	s10	16	45.2	31	44.4	1	0	0	0	1	-15	0.8	1
s30	s10	18	48	31	44.4	0	0	0	0	1	-13	3.6	1
s31	s10	19	48	31	44.4	1	0	0	0	1	-12	3.6	0
s32	s10	19	48	31	44.4	1	0	0	0	1	-12	3.6	0
s40	s10	18	42.8	31	44.4	0	0	0	0	0	-13	-1.6	1
s41	s10	21	42.8	31	44.4	0	0	0	0	0	-10	-1.6	1
s20	s11	16	45.2	29	44.4	0	0	0	0	1	-13	0.8	1
s30	s11	18	48	29	44.4	0	0	0	0	1	-11	3.6	1
s31	s11	19	48	29	44.4	0	0	0	0	1	-10	3.6	0
s32	s11	19	48	29	44.4	0	0	0	0	1	-10	3.6	0
s40	s11	18	42.8	29	44.4	1	0	0	0	0	-11	-1.6	1
s41	s11	21	42.8	29	44.4	0	0	0	0	0	-8	-1.6	1
s30	s20	18	48	16	45.2	0	0	0	1	1	2	2.8	0
s31	s20	19	48	16	45.2	1	0	0	1	1	3	2.8	0
s32	s20	19	48	16	45.2	1	0	0	1	1	3	2.8	0
s40	s20	18	42.8	16	45.2	0	0	0	1	0	2	-2.4	0
s41	s20	21	42.8	16	45.2	0	0	0	1	0	5	-2.4	0
s31	s30	19	48	18	48	0	1	-1	1	0	1	0	0
s32	s30	19	48	18	48	0	1	-1	1	0	1	0	0
s40	s30	18	42.8	18	48	0	0	0	0	0	0	-5.2	0
s41	s30	21	42.8	18	48	1	0	0	1	0	3	-5.2	0
s32	s31	19	48	19	48	1	1	-1	0	0	0	0	0
s40	s31	18	42.8	19	48	0	0	0	0	0	-1	-5.2	1
s41	s31	21	42.8	19	48	0	0	0	1	0	2	-5.2	1
s40	s32	18	42.8	19	48	0	0	0	0	0	-1	-5.2	1
s41	s32	21	42.8	19	48	0	0	0	1	0	2	-5.2	1
s41	s40	21	42.8	18	42.8	0	1	-1	1	0	3	0	0

Case study 2

A	B	p_A	d_A	p_B	d_B	Xii'k	Zii'j	Zij>Zi'j	p_A>B	d_A>B	p_A-B	d_A-B	class
s00	s01	17	71	16	71	0	1	1	1	0	1	0	1
s00	s02	17	71	19	71	1	1	1	0	0	-2	0	1
s00	s10	17	71	30	36.8	0	0	0	0	1	-13	34.2	1
s00	s11	17	71	13	36.8	0	0	0	1	1	4	34.2	1
s00	s20	17	71	10	48.5	0	0	0	1	1	7	22.5	0
s00	s30	17	71	15	73	1	0	0	1	0	2	-2	0
s00	s31	17	71	22	73	0	0	0	0	0	-5	-2	1
s00	s40	17	71	12	35	0	0	0	1	1	5	36	0
s00	s41	17	71	26	35	0	0	0	0	1	-9	36	1
s01	s02	16	71	19	71	0	1	1	0	0	-3	0	1
s01	s10	16	71	30	36.8	1	0	0	0	1	-14	34.2	1
s01	s11	16	71	13	36.8	0	0	0	1	1	3	34.2	1
s01	s20	16	71	10	48.5	1	0	0	1	1	6	22.5	0
s01	s30	16	71	15	73	0	0	0	1	0	1	-2	0
s01	s31	16	71	22	73	1	0	0	0	0	-6	-2	0
s01	s40	16	71	12	35	1	0	0	1	1	4	36	0
s01	s41	16	71	26	35	0	0	0	0	1	-10	36	0
s02	s10	19	71	30	36.8	0	0	0	0	1	-11	34.2	0
s02	s11	19	71	13	36.8	0	0	0	1	1	6	34.2	1
s02	s20	19	71	10	48.5	0	0	0	1	1	9	22.5	0
s02	s30	19	71	15	73	1	0	0	1	0	4	-2	0
s02	s31	19	71	22	73	0	0	0	0	0	-3	-2	0
s02	s40	19	71	12	35	0	0	0	1	1	7	36	0
s02	s41	19	71	26	35	0	0	0	0	1	-7	36	0
s10	s11	30	36.8	13	36.8	0	1	1	1	0	17	0	1
s10	s20	30	36.8	10	48.5	1	0	0	1	0	20	-11.7	0
s10	s30	30	36.8	15	73	0	0	0	1	0	15	-36.2	0
s10	s31	30	36.8	22	73	1	0	0	1	0	8	-36.2	0
s10	s40	30	36.8	12	35	1	0	0	1	1	18	1.8	0
s10	s41	30	36.8	26	35	0	0	0	1	1	4	1.8	0
s11	s20	13	36.8	10	48.5	0	0	0	1	0	3	-11.7	0
s11	s30	13	36.8	15	73	0	0	0	0	0	-2	-36.2	0
s11	s31	13	36.8	22	73	0	0	0	0	0	-9	-36.2	0
s11	s40	13	36.8	12	35	0	0	0	1	1	1	1.8	0
s11	s41	13	36.8	26	35	1	0	0	0	1	-13	1.8	0
s20	s30	10	48.5	15	73	0	0	0	0	0	-5	-24.5	0
s20	s31	10	48.5	22	73	1	0	0	0	0	-12	-24.5	1
s20	s40	10	48.5	12	35	1	0	0	0	1	-2	13.5	1
s20	s41	10	48.5	26	35	0	0	0	0	1	-16	13.5	1
s30	s31	15	73	22	73	0	1	1	0	0	-7	0	1
s30	s40	15	73	12	35	0	0	0	1	1	3	38	1
s30	s41	15	73	26	35	0	0	0	0	1	-11	38	1
s31	s40	22	73	12	35	1	0	0	1	1	10	38	0

s31	s41	22	73	26	35	0	0	0	0	1	-4	38	0
s40	s41	12	35	26	35	0	1	1	0	0	-14	0	1
s01	s00	16	71	17	71	0	1	-1	0	0	-1	0	0
s02	s00	19	71	17	71	1	1	-1	1	0	2	0	0
s10	s00	30	36.8	17	71	0	0	0	1	0	13	-34.2	0
s11	s00	13	36.8	17	71	0	0	0	0	0	-4	-34.2	0
s20	s00	10	48.5	17	71	0	0	0	0	0	-7	-22.5	1
s30	s00	15	73	17	71	1	0	0	0	1	-2	2	1
s31	s00	22	73	17	71	0	0	0	1	1	5	2	0
s40	s00	12	35	17	71	0	0	0	0	0	-5	-36	1
s41	s00	26	35	17	71	0	0	0	1	0	9	-36	0
s02	s01	19	71	16	71	0	1	-1	1	0	3	0	0
s10	s01	30	36.8	16	71	1	0	0	1	0	14	-34.2	0
s11	s01	13	36.8	16	71	0	0	0	0	0	-3	-34.2	0
s20	s01	10	48.5	16	71	1	0	0	0	0	-6	-22.5	1
s30	s01	15	73	16	71	0	0	0	0	1	-1	2	1
s31	s01	22	73	16	71	1	0	0	1	1	6	2	1
s40	s01	12	35	16	71	1	0	0	0	0	-4	-36	1
s41	s01	26	35	16	71	0	0	0	1	0	10	-36	1
s10	s02	30	36.8	19	71	0	0	0	1	0	11	-34.2	0
s11	s02	13	36.8	19	71	0	0	0	0	0	-6	-34.2	0
s20	s02	10	48.5	19	71	0	0	0	0	0	-9	-22.5	1
s30	s02	15	73	19	71	1	0	0	0	1	-4	2	1
s31	s02	22	73	19	71	0	0	0	1	1	3	2	1
s40	s02	12	35	19	71	0	0	0	0	0	-7	-36	1
s41	s02	26	35	19	71	0	0	0	1	0	7	-36	1
s11	s10	13	36.8	30	36.8	0	1	-1	0	0	-17	0	0
s20	s10	10	48.5	30	36.8	1	0	0	0	1	-20	11.7	1
s30	s10	15	73	30	36.8	0	0	0	0	1	-15	36.2	1
s31	s10	22	73	30	36.8	1	0	0	0	1	-8	36.2	1
s40	s10	12	35	30	36.8	1	0	0	0	0	-18	-1.8	1
s41	s10	26	35	30	36.8	0	0	0	0	0	-4	-1.8	1
s20	s11	10	48.5	13	36.8	0	0	0	0	1	-3	11.7	1
s30	s11	15	73	13	36.8	0	0	0	1	1	2	36.2	1
s31	s11	22	73	13	36.8	0	0	0	1	1	9	36.2	1
s40	s11	12	35	13	36.8	0	0	0	0	0	-1	-1.8	1
s41	s11	26	35	13	36.8	1	0	0	1	0	13	-1.8	1
s30	s20	15	73	10	48.5	0	0	0	1	1	5	24.5	0
s31	s20	22	73	10	48.5	1	0	0	1	1	12	24.5	0
s40	s20	12	35	10	48.5	1	0	0	1	0	2	-13.5	0
s41	s20	26	35	10	48.5	0	0	0	1	0	16	-13.5	0
s31	s30	22	73	15	73	0	1	-1	1	0	7	0	0
s40	s30	12	35	15	73	0	0	0	0	0	-3	-38	0
s41	s30	26	35	15	73	0	0	0	1	0	11	-38	0
s40	s31	12	35	22	73	1	0	0	0	0	-10	-38	1
s41	s31	26	35	22	73	0	0	0	1	0	4	-38	0

s41	s40	26	35	12	35	0	1	-1	1	0	14	0	0

Case study 3

A	B	p_A	d_A	p_B	d_B	Xii'k	Zii'j	Zij>Zi'j	p_A>B	d_A>B	p_A-B	d_A-B	class
s00	s01	10	36	20	36	1	1	1	0	0	-10	0	1
s00	s10	10	36	11	82	0	0	0	0	0	-1	-46	0
s00	s11	10	36	27	82	1	0	0	0	0	-17	-46	1
s00	s12	10	36	15	82	0	0	0	0	0	-5	-46	1
s00	s20	10	36	30	101	0	0	0	0	0	-20	-65	0
s00	s21	10	36	18	101	0	0	0	0	0	-8	-65	1
s00	s22	10	36	25	101	0	0	0	0	0	-15	-65	1
s00	s30	10	36	29	49	0	0	0	0	0	-19	-13	1
s00	s40	10	36	38	75	1	0	0	0	0	-28	-39	1
s00	s41	10	36	28	75	0	0	0	0	0	-18	-39	1
s01	s10	20	36	11	82	0	0	0	1	0	9	-46	0
s01	s11	20	36	27	82	1	0	0	0	0	-7	-46	0
s01	s12	20	36	15	82	0	0	0	1	0	5	-46	0
s01	s20	20	36	30	101	0	0	0	0	0	-10	-65	0
s01	s21	20	36	18	101	0	0	0	1	0	2	-65	0
s01	s22	20	36	25	101	0	0	0	0	0	-5	-65	0
s01	s30	20	36	29	49	0	0	0	0	0	-9	-13	0
s01	s40	20	36	38	75	1	0	0	0	0	-18	-39	0
s01	s41	20	36	28	75	0	0	0	0	0	-8	-39	0
s10	s11	11	82	27	82	0	1	1	0	0	-16	0	1
s10	s12	11	82	15	82	1	1	1	0	0	-4	0	1
s10	s20	11	82	30	101	0	0	0	0	0	-19	-19	0
s10	s21	11	82	18	101	0	0	0	0	0	-7	-19	1
s10	s22	11	82	25	101	0	0	0	0	0	-14	-19	1
s10	s30	11	82	29	49	1	0	0	0	1	-18	33	1
s10	s40	11	82	38	75	0	0	0	0	1	-27	7	1
s10	s41	11	82	28	75	0	0	0	0	1	-17	7	1
s11	s12	27	82	15	82	0	1	1	1	0	12	0	1
s11	s20	27	82	30	101	0	0	0	0	0	-3	-19	0
s11	s21	27	82	18	101	0	0	0	1	0	9	-19	0
s11	s22	27	82	25	101	0	0	0	1	0	2	-19	0
s11	s30	27	82	29	49	0	0	0	0	1	-2	33	0
s11	s40	27	82	38	75	1	0	0	0	1	-11	7	0
s11	s41	27	82	28	75	0	0	0	0	1	-1	7	1
s12	s20	15	82	30	101	0	0	0	0	0	-15	-19	0
s12	s21	15	82	18	101	0	0	0	0	0	-3	-19	0
s12	s22	15	82	25	101	0	0	0	0	0	-10	-19	0
s12	s30	15	82	29	49	1	0	0	0	1	-14	33	0
s12	s40	15	82	38	75	0	0	0	0	1	-23	7	0
s12	s41	15	82	28	75	0	0	0	0	1	-13	7	0

s20	s21	30	101	18	101	1	1	1	1	0	12	0	1
s20	s22	30	101	25	101	1	1	1	1	0	5	0	1
s20	s30	30	101	29	49	0	0	0	1	1	1	52	1
s20	s40	30	101	38	75	0	0	0	0	1	-8	26	1
s20	s41	30	101	28	75	1	0	0	1	1	2	26	1
s21	s22	18	101	25	101	1	1	1	0	0	-7	0	1
s21	s30	18	101	29	49	0	0	0	0	1	-11	52	0
s21	s40	18	101	38	75	0	0	0	0	1	-20	26	0
s21	s41	18	101	28	75	1	0	0	0	1	-10	26	1
s22	s30	25	101	29	49	0	0	0	0	1	-4	52	0
s22	s40	25	101	38	75	0	0	0	0	1	-13	26	0
s22	s41	25	101	28	75	1	0	0	0	1	-3	26	1
s30	s40	29	49	38	75	0	0	0	0	0	-9	-26	0
s30	s41	29	49	28	75	0	0	0	1	0	1	-26	1
s40	s41	38	75	28	75	0	1	1	1	0	10	0	1
s01	s00	20	36	10	36	1	1	-1	1	0	10	0	0
s10	s00	11	82	10	36	0	0	0	1	1	1	46	0
s11	s00	27	82	10	36	1	0	0	1	1	17	46	0
s12	s00	15	82	10	36	0	0	0	1	1	5	46	0
s20	s00	30	101	10	36	0	0	0	1	1	20	65	0
s21	s00	18	101	10	36	0	0	0	1	1	8	65	0
s22	s00	25	101	10	36	0	0	0	1	1	15	65	0
s30	s00	29	49	10	36	0	0	0	1	1	19	13	0
s40	s00	38	75	10	36	1	0	0	1	1	28	39	0
s41	s00	28	75	10	36	0	0	0	1	1	18	39	0
s10	s01	11	82	20	36	0	0	0	0	1	-9	46	1
s11	s01	27	82	20	36	1	0	0	1	1	7	46	1
s12	s01	15	82	20	36	0	0	0	0	1	-5	46	0
s20	s01	30	101	20	36	0	0	0	1	1	10	65	1
s21	s01	18	101	20	36	0	0	0	0	1	-2	65	1
s22	s01	25	101	20	36	0	0	0	1	1	5	65	1
s30	s01	29	49	20	36	0	0	0	1	1	9	13	1
s40	s01	38	75	20	36	1	0	0	1	1	18	39	1
s41	s01	28	75	20	36	0	0	0	1	1	8	39	1
s11	s10	27	82	11	82	0	1	-1	1	0	16	0	0
s12	s10	15	82	11	82	1	1	-1	1	0	4	0	0
s20	s10	30	101	11	82	0	0	0	1	1	19	19	0
s21	s10	18	101	11	82	0	0	0	1	1	7	19	0
s22	s10	25	101	11	82	0	0	0	1	1	14	19	0
s30	s10	29	49	11	82	1	0	0	1	0	18	-33	0
s40	s10	38	75	11	82	0	0	0	1	0	27	-7	0
s41	s10	28	75	11	82	0	0	0	1	0	17	-7	0
s12	s11	15	82	27	82	0	1	-1	0	0	-12	0	0
s20	s11	30	101	27	82	0	0	0	1	1	3	19	1
s21	s11	18	101	27	82	0	0	0	0	1	-9	19	1
s22	s11	25	101	27	82	0	0	0	0	1	-2	19	0

s30	s11	29	49	27	82	0	0	0	1	0	2	-33	1
s40	s11	38	75	27	82	1	0	0	1	0	11	-7	1
s41	s11	28	75	27	82	0	0	0	1	0	1	-7	0
s20	s12	30	101	15	82	0	0	0	1	1	15	19	1
s21	s12	18	101	15	82	0	0	0	1	1	3	19	1
s22	s12	25	101	15	82	0	0	0	1	1	10	19	1
s30	s12	29	49	15	82	1	0	0	1	0	14	-33	1
s40	s12	38	75	15	82	0	0	0	1	0	23	-7	1
s41	s12	28	75	15	82	0	0	0	1	0	13	-7	1
s21	s20	18	101	30	101	1	1	-1	0	0	-12	0	0
s22	s20	25	101	30	101	1	1	-1	0	0	-5	0	0
s30	s20	29	49	30	101	0	0	0	0	0	-1	-52	0
s40	s20	38	75	30	101	0	0	0	1	0	8	-26	0
s41	s20	28	75	30	101	1	0	0	0	0	-2	-26	0
s22	s21	25	101	18	101	1	1	-1	1	0	7	0	0
s30	s21	29	49	18	101	0	0	0	1	0	11	-52	1
s40	s21	38	75	18	101	0	0	0	1	0	20	-26	1
s41	s21	28	75	18	101	1	0	0	1	0	10	-26	0
s30	s22	29	49	25	101	0	0	0	1	0	4	-52	1
s40	s22	38	75	25	101	0	0	0	1	0	13	-26	1
s41	s22	28	75	25	101	1	0	0	1	0	3	-26	0
s40	s30	38	75	29	49	0	0	0	1	1	9	26	1
s41	s30	28	75	29	49	0	0	0	0	1	-1	26	0
s41	s40	28	75	38	75	0	1	-1	0	0	-10	0	0

Appendix 3 Performance metrics

Model	Parameter	Value	ACC	FSC	ROC	APR	REC	RMS	MXE	TIME
RF	num_tree, num_feature	100,1	0.85	0.781	0.91	0.83	0.79	0.2	0.62	3.63
	num_tree, num_feature	100,6	0.87	0.804	0.93	0.86	0.81	0.18	0.54	3.45
	num_tree, num_feature	100,11	0.88	0.822	0.93	0.87	0.83	0.15	0.47	3.53
	num_tree, num_feature	300,1	0.89	0.83	0.96	0.89	0.82	0.16	0.42	9.65
	num_tree, num_feature	300,6	0.89	0.83	0.96	0.89	0.82	0.16	0.42	9.65
	num_tree, num_feature	300,11	0.9	0.85	0.97	0.9	0.86	0.15	0.38	10.4
	num_tree, num_feature	500,1	0.91	0.86	0.98	0.9	0.86	0.14	0.35	15.1

	num_tree, num_feature	500,6	0.92	0.87	0.99	0.92	0.88	0.13	0.31	16
	num_tree, num_feature	500,11	0.92	0.88	0.99	0.92	0.88	0.12	0.28	16.4
KNN	k	k=1	0.77	0.72	0.83	0.76	0.73	0.23	1.3	0.27
	k	k=20	0.76	0.7	0.83	0.76	0.72	0.24	1.3	0.29
	k	k=50	0.75	0.7	0.82	0.75	0.7	0.24	1.2	0.3
	k	k=80	0.74	0.68	0.81	0.73	0.68	0.26	1.2	3.7
SVM	kernel, gamma	‘linear’, ’scale’	0.74	0.68	0.81	0.73	0.68	0.26	1.2	0.5
	kernel, gamma	‘rbf’, ’scale’	0.74	0.67	0.81	0.73	0.68	0.26	1.15	0.5
	kernel, gamma	‘poly’, ’scale’	0.74	0.67	0.81	0.72	0.68	0.27	1.13	0.54
	kernel, gamma	‘linear’, ’auto’	0.73	0.67	0.81	0.72	0.65	0.27	1.1	0.51
	kernel, gamma	‘rbf’, ’auto’	0.73	0.67	0.81	0.72	0.67	0.26	1.1	0.48
	kernel, gamma	‘poly’, ’auto’	0.73	0.67	0.81	0.72	0.67	0.27	1.1	0.46
	kernel, gamma	‘linear’, 0.001	0.73	0.66	0.8	0.72	0.67	0.27	1	0.51
	kernel, gamma	‘rbf’, 0.001	0.73	0.66	0.8	0.7	0.67	0.27	1.03	0.51
	kernel, gamma	‘poly’, 0.001	0.73	0.66	0.8	0.71	0.67	0.27	1.01	0.53
	kernel, gamma	‘linear’, 0.01	0.72	0.66	0.8	0.71	0.67	0.28	1	0.52
	kernel, gamma	‘rbf’, 0.01	0.72	0.66	0.8	0.71	0.67	0.28	0.99	0.51
	kernel, gamma	‘poly’, 0.01	0.72	0.65	0.8	0.71	0.66	0.28	0.97	0.51
	kernel, gamma	‘linear’, 1	0.72	0.657	0.8	0.71	0.66	0.28	0.96	0.46
	kernel, gamma	‘rbf’, 1	0.72	0.657	0.8	0.7	0.66	0.28	0.95	0.45
	kernel, gamma	‘poly’, 1	0.72	0.656	0.8	0.71	0.66	0.28	0.94	0.45
NB			0.71	0.66	0.8	0.7	0.66	0.28	0.93	0.48

LR	tolerance	tol = 0.0001	0.72	0.65	0.8	0.7	0.66	0.28	0.92	0.49
	tolerance	tol = 0.001	0.72	0.65	0.8	0.7	0.66	0.28	0.91	0.48
	tolerance	tol = 1	0.72	0.65	0.8	0.7	0.66	0.28	0.9	0.48
	tolerance	tol = 100	0.72	0.65	0.8	0.7	0.66	0.28	0.89	0.5
	tolerance	tol = 10000	0.72	0.652	0.8	0.7	0.66	0.28	0.88	0.48

VITA AUCTORIS

NAME:	Syeda Marzia
PLACE OF BIRTH:	Dinajpur, Bangladesh
YEAR OF BIRTH:	1997
EDUCATION:	Holy Cross College, Bangladesh, 2014
	Bangladesh University of Engineering and Technology, B.Sc., Dhaka, Bangladesh, 2019
	University of Windsor, M.Sc., Windsor, ON, 2023