JEMSC Journal of Engineering Managment and Soft Computing (JEMSC)

# Multi-Objective Multi-Level Scheduling in Cloud Manufacturing: A Hybrid Approach Integrating Mathematical Modeling and Machine Learning

**Mohammadreza Razdar[1], Mohammad Amin Adibi[2]✉ and Hassan Haleh[3]**

1. Corresponding author: Department of Industrial Engineering, Qa. C., Islamic Azad University, Qazvin, Iran, maadibi@iau.ac.ir,
2. Department of Industrial Engineering, Isfahan University of Technology, Golpayegan Branch, Golpayegan, Iran

**Article Info**

**Article type:**
Research Article

**Article history:**
Received 29 October 2025
Received in revised form 11 February 2026
Accepted 1 March 2026
Published online 1 April 2026

**Keywords:**
Multi-Objective Scheduling, Multi-Level Scheduling, Cloud Manufacturing, Machine Learning, Metaheuristic Algorithms.

**ABSTRACT**

This research investigates multi-objective, multi-level scheduling within a cloud manufacturing environment, employing metaheuristic algorithms. Initially, a comprehensive literature review was conducted, followed by the development of a multi-objective mathematical model and integrated with a machine learning (ML) model. The ML model is designed to estimate the risk associated with production units and sub-activities within the cloud network. Concurrently, the mathematical model aims to optimize production time and cost in the cloud environment. The model's validity was first assessed by solving it for small-scale instances. Given that the exact method was only feasible up to the tenth instance, metaheuristic algorithms were utilized for solving the model in larger dimensions. The results demonstrated the model's solvability in large-scale scenarios using the NSGAII algorithm. Subsequently, the model was solved considering risk input values, revealing that among the temporal parameters, transportation time and setup time exhibit the most significant impacts on overall time. Activity time also indicates a high level of influence in certain cases. Regarding cost parameters, activity time appears to be the most influential factor on cost, followed by setup time. Among the cost-related parameters, preparation cost has the greatest effect on time, with transportation cost being the next. Crucially, activity time emerges as the most impactful cost parameter, with transportation cost following as the second most influential one.

**Cite this article:** Adibi, M. & et al, (2026)., Multi-Objective Multi-Level Scheduling in Cloud Manufacturing: A Hybrid Approach Integrating Mathematical Modeling and Machine Learning. *Journal of Engineering Management and Soft Computing*, 12 (2). 164-183.

**DOI**: https://doi.org/10.22091/jemsc.2026.14343.1319



DOI: https://doi.org/10.22091/jemsc.2026.14343.1319

**Publisher**: University of Qom

## 1) Introduction

In today's competitive world, the needs and requirements of customers and users are rapidly changing, and many customers prefer to receive customized products with the best price and quality in the shortest possible time. Meanwhile, large and small manufacturing companies face challenges in responding to customer needs. The production resources of large companies are usually geographically dispersed, and their integration has always been accompanied by serious challenges (Rana, 2024). On the other hand, small and medium-sized companies are unable to compete with large manufacturing companies and gradually lose their customers. Fortunately, with recent advancements in information technology and the Internet of Things, the aforementioned problems can be appropriately resolved. In fact, the path to achieving the concept of globalization has become smoother than before, such that companies can share various resources and technologies, and also, if necessary, utilize others' resources to compensate for their production shortcomings (Zhang et al., 2024).

Over the past years, trends created by new industrial production inclinations and requirements, such as globalization, individualization, digitalization, cloud, collaboration, and the integration of these activities with new emerging technologies, such as cloud computing, the Internet of Things (IoT), Cyber-Physical Systems (CPS), big data analytics, and Artificial Intelligence (AI), has led to the proposal of a new production paradigm called cloud manufacturing (Hemmati et al., 2025; Zhang et al., 2017; Zhou, 2018). Over the past eight years, cloud manufacturing, as a new production paradigm, has attracted significant research interest worldwide. The goal of cloud manufacturing is to provide on-demand manufacturing services to consumers via the internet, and in this regard, planning is one of the important tools to achieve the goal of cloud manufacturing (Keihani et al., 2026; Pan et al., 2023). Cloud manufacturing, as a service-oriented model, facilitates the production of customized products through the integration of manufacturing resources located in different geographical locations. This model also provides a suitable platform for sharing manufacturing resources.

Information and communication technologies have a significant impact on the development and evolution of advanced manufacturing systems, and the manufacturing sector, including goods and services, has been affected by these advancements, emphasizing innovation, speed, and customization in the delivery of products and services in response to increasing and diverse demands and the competitive business market. The integration and standardization of manufacturing, communication, information, mechanical, and management systems and technologies are essential for the interaction and interoperability of world-class enterprises. In this regard, an integrated supply chain, based on cloud manufacturing and services, utilizing cloud computing technologies and the central role of the customer in all the stages of the product lifecycle services, can bring about a significant transformation in the competitive business market for enterprises, especially for scattered small and medium-sized enterprises.

This research investigates the problem of scheduling activities related to the realization of a product in a cloud manufacturing system. The presented model is based on workshop flow scheduling, implemented based on cloud manufacturing. In this regard, executive activities in service centers, along with the transportation of semi-finished products between service centers, will be considered to best meet the objectives of the cloud manufacturing system. Therefore, a mathematical model and a solution method will be presented, the main objective of which is to minimize the maximum completion time of tasks (Cmax), while also considering the minimization of total cost (TC), and the minimization of maximum risk3 (Rmax) of production.

Previous research has rarely addressed production scheduling in cloud systems considering the above objectives and, from a methodological perspective, the combination of machine learning and metaheuristic algorithms is not observed in the research literature in the studied field, which makes the current research different from previous studies. To solve the model, the metaheuristic method of the multi-objective genetic algorithm 7 (NSGA II) has been used. To generate an initial solution for the proposed metaheuristic algorithm, data analysis techniques are utilized, and finally, the obtained results are analyzed.

## 2) Research Background

This section reviews the research literature in the field of cloud production scheduling. Articles from 2011 to the present have been examined, and finally, based on the reviewed articles, research gaps are extracted. Zhou et al. (2018), while distinguishing cloud manufacturing into two parts: smart manufacturing using cloud computing and cloud manufacturing, determined the structural requirements of cloud computing for three groups: cloud providers, cloud-using enterprises, and cloud users. Liu and Zhang (2017) studied cloud manufacturing as a new method to overcome existing bottlenecks between information development and its applications in manufacturing processes. Ren et al. (2013), investigated the concept of cloud manufacturing and its practical application. Cloud manufacturing operating systems should provide some common features as an intermediary as well as a manager.

Ren et al. (2013) investigated the collaboration of cloud computing in cloud manufacturing, examining the concepts and technologies of advanced manufacturing systems in cloud manufacturing, the key features of cloud manufacturing, and the cloud manufacturing paradigm. Moghaddam et al. (2015) reviewed and analyzed internet manufacturing collaboration alliances and service-oriented architectures to build cloud services that enable manufacturing as a service. He and Xu (2015) reviewed research on the industry status in cloud manufacturing and concluded that approximately 80% of articles published between 2010 and 2012 were from China, where manufacturing is considered for industrial development in the national economy. Bandaru et al. (2017) investigated data mining methods for knowledge discovery in multi-objective optimization problems.

Govindan et al. (2017), in their study, investigated a hybrid method of decision tree and scatter search algorithm for data classification based on the entropy index to generate solutions for determining the sequence of operations for jobs on machines in a process manufacturing system. Zhou et al. (2018), in their paper, modeled and solved the two-level vehicle transportation problem with multiple depots. Helo et al. (2019) studied the concept of cloud manufacturing for scheduling metal sheet production in a process manufacturing system. Yangkui Liu et al. (2018) reviewed research related to scheduling in a cloud manufacturing environment and presented relevant papers in appropriate sections. Halty et al. (2020) conducted a systematic literature review on scheduling in cloud manufacturing systems by searching relevant papers from recent years.

Fomin et al. (2024) addressed heterogeneous graph networks for scheduling in cloud manufacturing and logistics. Rena et al. (2024) provided a review of open theory models to support production planning, scheduling, cloud manufacturing, and sustainable production systems. Zhang et al. (2024) focused on optimizing resource scheduling by considering the supply and demand sides of services under cloud manufacturing. Akhavan Hariri et al. (2025) presented a sustainable multi-product scheduling model with a focus on logistics service sharing in cloud manufacturing systems.

The section above provides a comprehensive review of the research literature in the studied field; however, based on the literature review, it is observed that no research has been found that performs multi-objective, multi-level scheduling in a cloud manufacturing environment and pursues the objectives of reliability, time, and cost. Therefore, a research gap is evident in this area. Although the existing research in its functional domain has a relatively comprehensive approach, it also has shortcomings that the current research aims to address. Given the weaknesses in existing research, the current study seeks to present a new model in the field of scheduling in a cloud environment that simultaneously aims to minimize and optimize cost and time and maximize reliability.

## 3) Research Method

In cloud manufacturing systems, there are a number of production units that offer various services. Each service consists of several activities, and each activity has several sub-activities that can be performed in different locations. On the other hand, each activity can be performed at different quality levels. In other words, the quality levels of activities may not be uniform and are adjusted based on received orders or overall strategies. In these systems, there are setup, preparation, transfer, and activity times, and consequently, there are costs. This means that each activity incurs costs for setup, transportation, and execution, which are accounted for in the overall system; the lower these costs, the lower the total system cost.

Therefore, it can be said that time and cost are the two main objectives in a cloud manufacturing system, with the goal of reducing or minimizing them. It is noteworthy that risk can also be considered for activities. In fact, the third objective of cloud manufacturing systems can be minimizing risk in the production system. However, risk is not inherently a quantitative concept. Instead, it can be quantified and categorized into different levels, such as low risk, high risk, and medium risk. Determining the risk level can be done using past data. Therefore, utilizing machine learning algorithms for risk classification can be an important and interesting endeavor in here.

Risk can include production unit risk or sub-activity risk. This refers to when a production unit experiences downtime risk, or how many times a sub-activity has been delayed or stopped in the past, which can determine the input variables for measuring the degree of risk or the risk level of activities. The relevant variables are introduced in Table 1.

**Table 1. Input Variables**

| Row | Variable Title | Variable Symbol | Variable Type | Variable Scale |
|---|---|---|---|---|
| 1 | Number of times the sub-activity is delayed | X1 | Determinant of micro-activity risk | Quantitative |
| 2 | Number of times the sub-activity stops | X2 | Determinant of micro-activity risk | Quantitative |
| 3 | The number of times the sub-activity is performed is lower than expected. | X3 | Determinant of micro-activity risk | Quantitative |
| 4 | Frequency of total system failure in the production unit | X4 | Manufacturing unit risk determinant | Quantitative |
| 5 | Frequency of minor system failures in the production unit | X5 | Manufacturing unit risk determinant | Quantitative |
| 6 | Production unit risk level | Y1 | First output variable | Category |
| 7 | Sub-activity risk level | Y2 | Second output variable | Category |

Based on the five variables introduced above, the risk level of sub-activities and the production unit can be determined. These determined values are subsequently entered as parameters into the mathematical model of the present research, and accordingly, the total system risk is minimized. This is done by selecting activities that have lower risk, and also by selecting production units for which lower risk has been considered based on artificial intelligence algorithms. Furthermore, the first objective of the model presented in the current research is to minimize time, and the second objective is to minimize cost. In the following, this model will be introduced along with its indices, parameters, decision variables, and objective functions. Before that, the model assumptions are determined.

- Information processing and exchange may require time.

- Requests are announced by the customer to the cloud space in a customized manner.

- Customer orders are prioritized based on their arrival sequence.

- Each customer's order can include one or more requests.

- There is no mandatory execution priority among the requests of an order from each customer.

- Each request can be broken down into one or more main components for execution.

- For each service provided, there can be one or more service providers (service provider selection).

- The execution of each part of the request is assigned to one of the relevant stakeholders.

- Each request, to complete processing, includes a specific sequence of a series of main work activities.

- The main activity of each part of the requested product can include sub-operational activities (sub-activities).

- The processing sequence of the main and sub-activities of each request is specified.

- At the moment of scheduling, the service capacity (speed, quality level, efficiency, technology, etc.) and the free time of each stakeholder are specified (dynamism).

- The geographical location of service providers is fixed and determined.

- Each machine is capable of performing and processing one type of activity at any given moment.

- An activity cannot be interrupted during processing. Activities are processed continuously and without interruption.

- Machines have no setup time, and setup time is considered part of the operation time.

- There is no transportation time between sub-activities of each station; otherwise, it is negligible or considered part of the processing time.

- There can be a distance (temporal and spatial) between stakeholders.

- The machine speed remains constant after the activity starts and during job processing without the ability to change.

- All parameters are deterministic.

- All machines and activities are available from the beginning of the planning horizon until the completion of job scheduling.

- During the processing of activities, machines do not break down, and there are no breakdowns or unforeseen events for vehicles.

- Machines may experience idle time.

- Storage of semi-finished products is allowed and the capacity of in-process buffers is unlimited.

- The cloud space is responsible for integrated scheduling.

- Activities related to new requests, along with remaining activities from previously scheduled requests, are re-scheduled. If a negative impact (delay) happens in previous schedules, it will incur a penalty.

-The risk of the production unit is determined through machine learning methods.

-The risk of sub-activities is determined through machine learning methods.

-A quality level has been determined for performing sub-activities.

The structure of cloud manufacturing according to the present research model is depicted in the figure below.

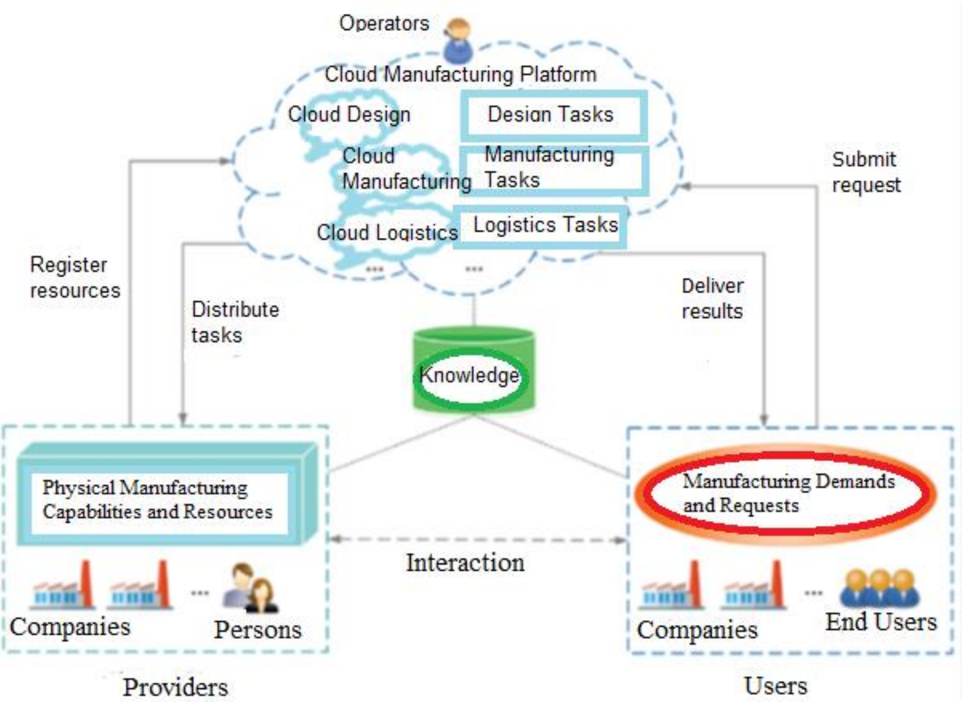


**Figure 1. Cloud Manufacturing Structure**

Figure 1 shows that demands and requests from end-users are sent to the cloud, and then, individuals and companies become responsible for fulfilling these demands. In the cloud, manufacturing logistics and design tasks are executed by the cloud manufacturing platform. After a request is sent, these tasks are performed by service providers in the cloud and then delivered to consumers or customers.

**Indices**

| | |
|---|---|
| $C$ | Production units |
| $S_c$ | Production unit services c |
| $T$ | Activities |
| $O_t$ | Sub-activities of each activity t |
| $P$ | Positions |
| $Q$ | Quality |

**Parameters**

| | |
|---|---|
| $\vartheta_{ctso}$ | Possibility of processing sub-activity o from activity t from service s in production unit c |
| $\tau_{to}$ | The start time of sub-activity o from activity t |
| $\theta_{ctso}$ | Preparation time of sub-activity o from activity t |
| $\sigma_{otscq}$ | Duration of execution of sub-activity o of activity t by service s in production unit c at quality level Q based on machine learning |
| $\alpha_t$ | Transportation time between two production units c |

| | |
|---|---|
| $\boldsymbol{\gamma_{to}}$ | Uptime based on cloud platform |
| $\boldsymbol{\varphi_{to}}$ | The cost of setting up sub-activity o from activity t |
| $\boldsymbol{\omega_{cc'}}$ | Cost of preparing sub-activity o from activity t |
| $\boldsymbol{\rho_{to}}$ | Transportation costs between two production units |
| $\boldsymbol{\vartheta_{csq}}$ | Service cost for performing sub-activity o of activity t at level q |
| $\boldsymbol{\beta_{ciq}}$ | Cost of service s per unit of production c at quality level Q |
| $\boldsymbol{\mu_{c}}$ | Production unit risk level C |
| $\boldsymbol{R_{ot}}$ | Risk level of sub-activity o of activity t based on machine learning techniques |
| $\boldsymbol{\beta}$ | Minimum number of active production units |

## Decision variables

| | |
|---|---|
| $\boldsymbol{V_c}$ | 1 if production unit C is active, otherwise zero |
| $\boldsymbol{X_{ctspo}}$ | If sub-activity o of activity t is executed at location p of service s in production unit C, then 1, otherwise zero |
| $\boldsymbol{Y_{ctso}}$ | If the preparation process is required to execute sub-activity o of activity t in service s of production unit c, then 1, otherwise zero |
| $\boldsymbol{Z_{tcc'o,o+1}}$ | If sub-activity o is performed by a production unit C' other than activity T, then 1, otherwise zero |
| $\boldsymbol{W_{to}}$ | If service s from production unit c is performed in the first position of sub-activity o of activity t, then 1, otherwise zero |
| $\boldsymbol{R_{to}}$ | If two sequences of activity o of activity t are executed in two different production units, 1, otherwise zero |
| $\boldsymbol{G_{to}}$ | Start time of sub-activity o from activity t |
| $\boldsymbol{D_{to}}$ | Completion time of sub-activity o from activity t |
| $\boldsymbol{P_{ct}}$ | Service time of sub-activity o from activity t |
| $\boldsymbol{A_{ct}}$ | The transportation time of sub-activity o from activity t |
| $\boldsymbol{CT_t}$ | Completion time of sub-activity o from activity t |
| $\boldsymbol{RT_c}$ | Start time of sub-activity o from activity t |

$$Min\ Cmax = \max_{t}\{CT_t\} \tag{1}$$

$$\begin{aligned} Min\ TC = &\sum_{t\in T}\sum_{o\in O_t}\sum_{c\in C}\sum_{s\in S_c}\gamma_{to}Y_{ctso} + \sum_{t\in T}\sum_{o\in O_t}\sum_{c\in C}\sum_{s\in S_c}\varphi_{to}N_{ctso} \\ &+ \sum_{t\in T}\sum_{o\in O_t}\sum_{c\in C}\sum_{s\in S_c}\sum_{p\in P}(\rho_{to}+\vartheta_{csq})X_{ctspo} + \sum_{t\in T}\sum_{o\in O_t}\sum_{c\in C}\sum_{c'\in C}\omega_{cc'}Z_{tcc'o,o+1} \end{aligned} \tag{2}$$

$$Min \ = \sum_{c}^{C} \mu_c V_c + \sum_{c}^{C}\sum_{t}^{T}\sum_{s}^{S}\sum_{p}^{P}\sum_{o}^{O} R_{ot} X_{ctspo} \tag{3}$$

$s.t$:

$$\sum_{c\in C}\sum_{s\in S_c}\sum_{p\in P} X_{ctspo} = 1, \quad \forall t \in T; o \in O_t \tag{4}$$

$$\sum_{t\in T}\sum_{o\in O_t} X_{ctspo} \leq 1, \quad \forall c \in C; p \in P; s \in S_c \tag{5}$$

$$\sum_{t\in T}\sum_{o\in O_t} X_{cts,p+1,o} \leq \sum_{t\in T}\sum_{o\in O_t} X_{ctspo}, \quad \forall c \in C; p \in P; s \in S_c \tag{6}$$

$$\sum_{p\in P} X_{ctspo} \leq \vartheta_{csq}, \quad \forall c \in C; t \in T; s \in S_c; o \in O_t \tag{7}$$

$$X_{cts,p+1,o} + X_{ct'spo'} - 1 \leq Y_{ctso}, \quad \forall c \in C; p \in P; t,t' \in T; o,o' \in O_t; s \in S_c \tag{8}$$

$$W_{to} = \sum_{c\in C}\sum_{s\in S_c} \delta_{to} Y_{ctso}, \quad \forall t \in T; o \in O_t \tag{9}$$

$$X_{cts,1,o} \leq U_{ctso}, \quad \forall c \in C; t \in T; s \in S_c; o \in O_t \tag{10}$$

$$F_{to} = \sum_{c\in C}\sum_{s\in S_c} \tau_{to} U_{ctso}, \quad \forall t \in T; o \in O_t \tag{11}$$

$$R_{to} = \sum_{c\in C}\sum_{s\in S_c}\sum_{p\in P} \theta_{ctso} X_{ctspo}, \quad \forall t \in T; o \in O_t \tag{12}$$

$$X_{ctspo} + X_{c'ts'p',o+1} - 1 \leq Z_{tcc'o,o+1}, \quad \forall p,p' \in P; t \in T; c \neq c' \in C; s,s' \in S_c; o \in O_t \tag{13}$$

$$L_{to} = \sum_{c\in C}\sum_{\substack{c'\in C\\ c\neq c'}} \sigma_{otscq} Z_{tcc'o,o+1}, \quad \forall t \in T; o \in O_t \tag{14}$$

$$G_{to} = D_{to} + W_{to} + F_{to} + R_{to}, \quad \forall t \in T; o \in O_t \tag{15}$$

$$G_{t,o-1} + L_{t,o-1} \leq D_{to}, \quad \forall t \in T; o \in O_t \tag{16}$$

$$\alpha_t \leq G_{t,1}, \quad \forall t \in T \tag{17}$$

$$CT_t \geq G_{to}, \quad \forall t \in T; o = O_t \tag{18}$$

$$X_{ctspo} \leq V_c \tag{19}$$

$$\sum_{c}^{C} V_c \geq \boldsymbol{\beta} \tag{20}$$

$$X_{ctspo}, Y_{ctso}, U_{ctso}, Z_{tcc'o,o+1} \in \{0,1\} \tag{21}$$

$$W_{to}, F_{to}, R_{to}, L_{to}, G_{to}, D_{to}, P_{ct}, A_{ct}, CT_t, RT_c \geq 0 \tag{22}$$

Equation 1 aims to minimize the completion time of each activity across all production units. This is achieved by calculating the maximum manufacturing time for the last activity, based on scheduling patterns.

Equation 2 seeks to minimize the total cost of the cloud production system. These costs include startup, setup, service, and transportation costs between production units. Startup cost includes the cost of initiation. The setup cost covers the cost of machine commissioning. Transportation cost includes the cost of transferring products between production units.

Equation 3 aims to create information disclosure within production units. In this constraint, the risk level is calculated using machine learning algorithms, and thus, the risk level of each activity is determined based on prediction rather than general opinions. This parameter is multiplied by the binary variable for activity execution, thereby determining the risk level.

Constraint 4 indicates that each sub-activity must be assigned to exactly one service. In other words, each sub-activity is naturally performed by one service.

Constraint 5 shows that a sub-activity can be assigned to a maximum of one service position. In the above constraint, each sub-activity is performed at one position.

Constraint 6 indicates that a sub-activity is assigned to the next position only when all active positions of that service have been filled. Naturally, the active positions of the sub-activity must be completed before moving to the next position.

Constraint 7 ensures that a sub-activity is only assigned to services that are technically capable of processing it.

Constraint 8 indicates the processing location of activity sequences based on preparation time by different services.

Constraint 9 indicates the preparation time for each sub-activity.

Constraints 10 and 11 indicate the start time of the first activity and subsequent sub-activities

Constraint 12 calculates the service time for each sub-activity.

Constraint 13 provides the probability of processing another sub-activity from an activity by another production unit.

Constraint 14 indicates the transportation time between two different production units.

Constraint 15 calculates the completion time of all activities and their corresponding sub-activities.

Constraints 16 and 17 ensure that the start time of an activity depends on the completion of previous activities.

Constraint 18 states that the start time of each activity must be greater than the client's request time from the cloud system.

Constraint 19 calculates the completion time of each activity by all production units.

Constraint 20 shows that if a production unit is active, it is possible to perform sub-activities within it.

Constraint 21 indicates the calculation of the minimum number of active production units.

Constraints 22 indicate the type of decision variables.

In this research, machine learning algorithms are used to predict risk. These algorithms include the Random Forest algorithm, Support Vector Machine, and Decision Tree. Using these algorithms, the risk level of each production unit and each sub-activity is determined. Then, the presented mathematical model is solved using the metaheuristic algorithms NSGAII and Grey Wolf Optimizer.

Random Forest is considered a supervised learning algorithm. As its name suggests, this algorithm randomly builds a forest. The constructed forest is, in fact, a group of decision trees. The process of building a forest using trees is often done using the bagging method. The main idea of the bagging method is that a combination of learning models enhances the overall results of the model. Simply put, Random Forest builds several decision trees and merges them to achieve more accurate and stable predictions.

One of the advantages of Random Forest is its applicability to both classification and regression problems, which constitute the majority of current machine learning systems. Here, the performance of Random Forest for classification will be described, as classification is sometimes considered the building block of machine learning.

A decision tree is a flowchart-like model that helps predict outcomes by mapping observations about an item to conclusions about its target variable. The tree consists of nodes that represent data features, branches that represent decision rules, and leaves that represent outcomes or class labels.

In other words, a decision tree algorithm creates a tree structure from training data that represents a set of decisions leading to the prediction of the target variable. This algorithm works by recursively splitting the data into subsets based on features that provide the most information until a final prediction or decision is made.

Support Vector Machine (SVM) is one of the supervised learning methods used for classification and regression. The working principle of the SVM classifier is the linear classification of data, and in linear data separation, we try to choose a hyperplane with a larger margin of confidence. Solving the equation for finding the optimal line for data is done by nonlinear programming methods, which are well-known methods for solving constrained problems. Before linear separation, in order for the machine to be able to classify highly complex data, we map the data to a much higher-dimensional space using the phi function.

The performance of these algorithms is determined based on the following four criteria:

- Accuracy
- Precision
- Recall
- F1score

The higher the score obtained from the algorithms for these criteria, the higher the efficiency of the corresponding algorithm. Accuracy indicates the number of correctly classified samples relative to the total sample data. Its calculation formula is as follows:

$$accuracy = \frac{TN + TP}{TN + FP + TP + FN} \tag{23}$$

In the above constraint, TN is the total number of true negatives. TP represents the total number of true positives. FP denotes the total number of false positives, and FN stands for the total number of false negatives.

Precision indicates the positive predictive value in data sample classification. Its formula is as follows:

$$Precision = \frac{TP}{FP + TP} \tag{24}$$

The next metric, recall, is defined as sensitivity or true positive rate. Its formula is as follows:

$$Recall = \frac{TP}{FN + TP} \tag{25}$$

And finally, the ultimate criterion for evaluating the efficiency of machine learning algorithms in classification is the F1-Score, which simultaneously calculates both precision and recall, and is as follows:

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{26}$$

In the above formula, precision is multiplied by recall in the numerator; however, in the denominator, these two metrics are summed and multiplied by 2, the result of which is the f1score, and the higher it is, the better the performance of an algorithm.

## 4) Research Findings

This section discusses the implementation of the hybrid machine learning and mathematical modeling approach presented in the methodology section. First, machine learning algorithms are used to predict the risk level of the production unit and sub-activity. After determining these two parameters, their values are input into the mathematical model, and the mathematical model is optimized to minimize both total time and cost in the cloud environment. The NSGAII algorithm in MATLAB software is used to solve the mathematical model. Furthermore, Python is used to implement the machine learning algorithms, and the results are presented below.

This section focuses on implementing activity risk and activity unit risk. Three algorithms—Random Forest, Decision Tree, and Support Vector Machine—are implemented, and these algorithms are compared based on four criteria: accuracy, precision, recall, and f1score, to select the best algorithm. The input variables are introduced in the methodology and model sections, and these variables form the basis of measuring both the risk of the production unit and sub-activity risk. The results are presented in the following tables and figures. It should be noted that an algorithm is considered superior if it achieves a higher value than other algorithms for all the four examined criteria.

**Table 2. Comparison of Machine Learning Algorithms for Predicting Production Unit Risk**

| Algorithms | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Support Vector Machine | 0.873 | 0.89 | 0.855 | 0.872 |
| Decision Tree | 0.9 | 0.91 | 0.89 | 0.9 |
| Random Forest | 0.9 | 0.88 | 0.92 | 0.9 |

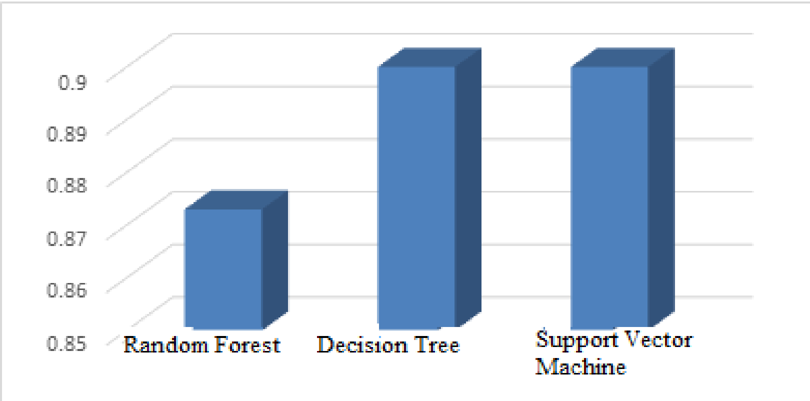


**Figure 1. Comparison of Machine Learning Algorithms for Predicting Production Unit Risk in Terms of Accuracy**

As shown in Figure 1, the Random Forest and Decision Tree algorithms have the highest level of accuracy in predicting production unit risk, with a value of 0.9, indicating up to 90% accuracy in classifying production units by risk using these two algorithms.

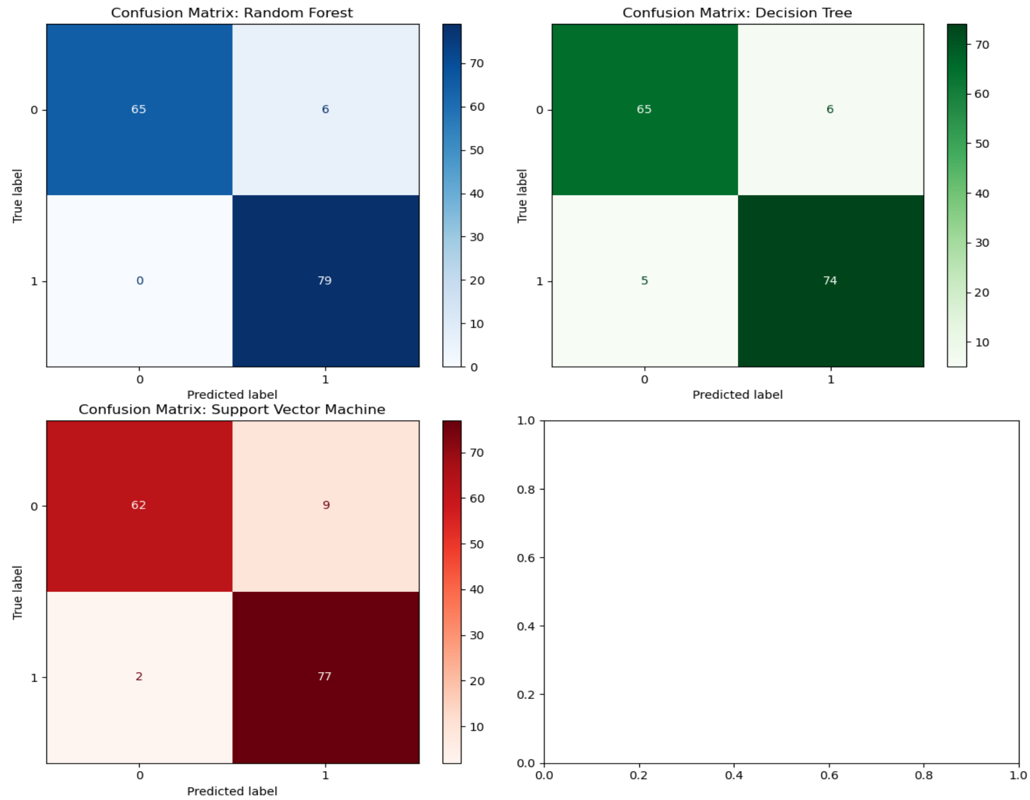


**Figure 2. Confusion Matrix for Predicting the Risk of Production Units**

In the confusion matrix below, it is observed that the numbers on the main diagonal for the random forest are higher than the other two algorithms, and the values on the off-diagonal are closer to zero than the other algorithms. In a confusion matrix, an algorithm is superior if its off-diagonal is closer to zero and its main diagonal has larger values than other algorithms. Therefore, in terms of the confusion matrix, it can be said that the random forest is considered the superior algorithm. In fact, it should be emphasized that the higher the values on the main diagonal, the more accurate the estimation of categories and classification, with the so-called correct true category detection performed, while the opposite is true for the off-diagonal, indicating that incorrect estimation is reflected in the off-diagonal.

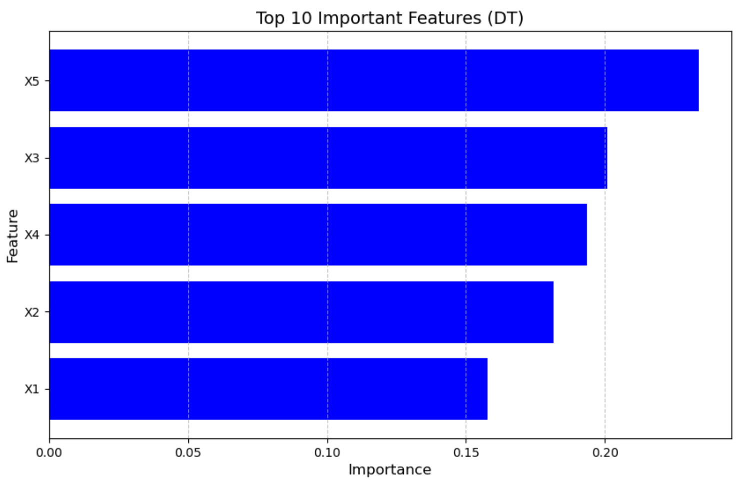


**Figure 3. Comparison of Influential Factors Based on the Decision Tree Matrix**

Figure 3 compares the influential input variables, showing that variable X5, which is the number of partial system failures in the production unit, has the greatest impact on the risk of production units. Following this are the number of times sub-activities are performed below expectations, the number of total system failures in the production unit, the number of sub-activity stoppages, and the number of sub-activity delays.

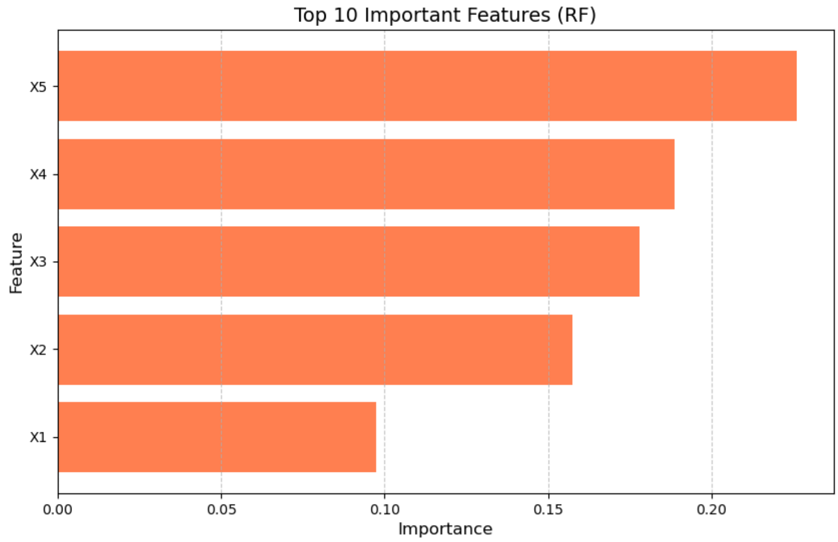


**Figure 4. Comparison of Influential Factors According to the Random Forest Algorithm**

According to the Random Forest algorithm, the most influential variable is still the number of partial system failures in the production unit; however, the second rank, belonging to the number of total system failures in the production unit, and the third rank, belonging to the number of times sub-activities, were performed below expectations, indicating a slight difference between the Random Forest and Decision Tree algorithms.

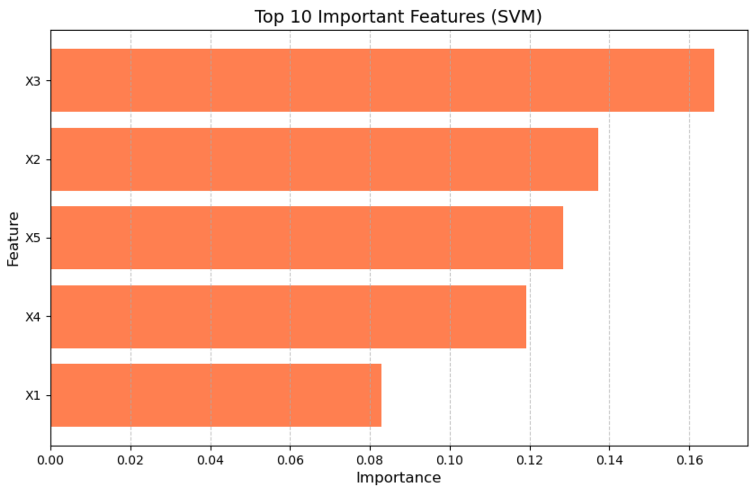


**Figure 5. Comparison of Influential Factors According to the Support Vector Machine Algorithm**

In Figure 5, it can be observed that the variable with the most influence is the number of times the sub-activity is performed below expectations, followed by the number of times the sub-activity is stopped. Given that the Support Vector Machine algorithm revealed a relative superiority over other algorithms, its results should be considered as the standard for predicting the risk of production units.

**Table 3. Comparison of Machine Learning Algorithms for Sub-Activity Risk Prediction**

| Algorithms | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Random Forest | 0.946 | 0.927 | 0.974 | 0.95 |
| Decision Tree | 0.926 | 0.925 | 0.936 | 0.93 |
| Support Vector Machine | 0.926 | 0.9 | 0.96 | 0.932 |

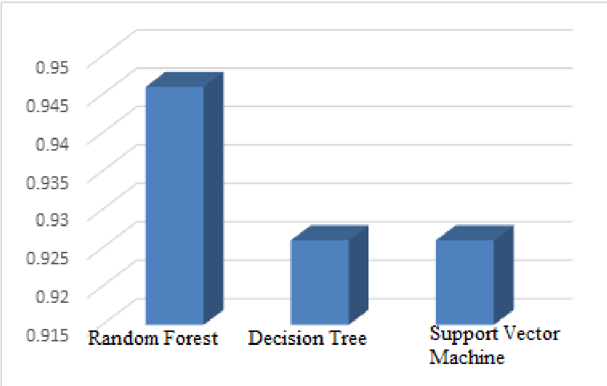


**Figure 6. Comparison of Machine Learning Algorithms for Predicting Micro-Activity Risk in Terms of Accuracy**

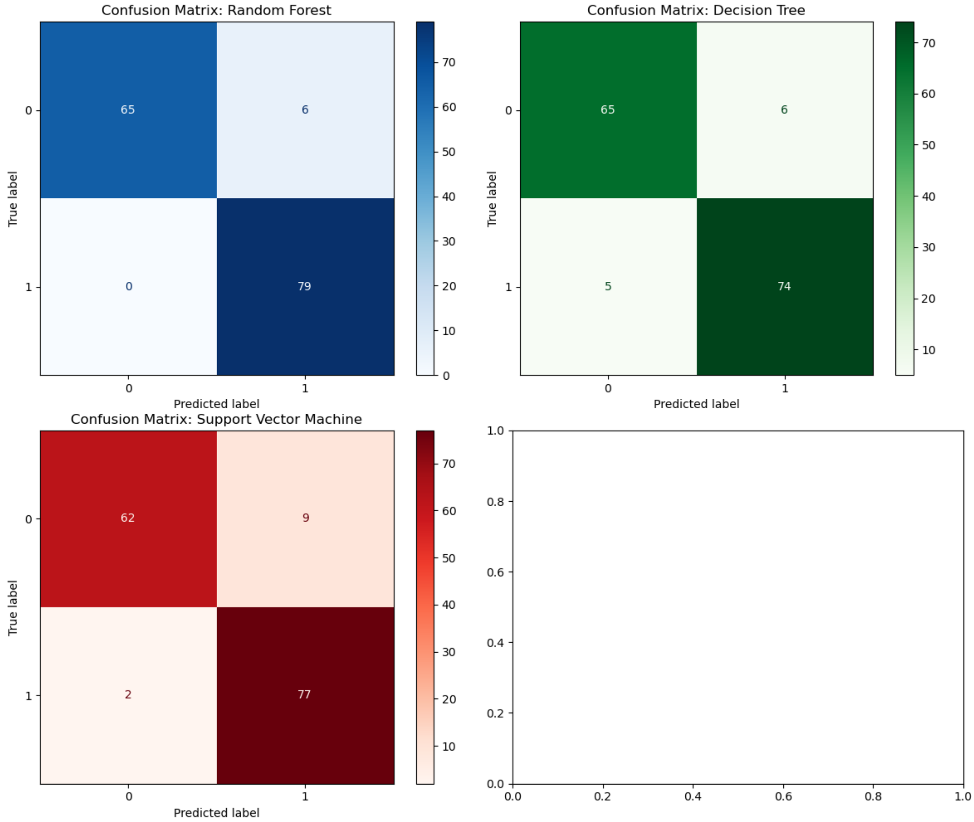


**Figure 7. Confusion Matrix for Predicting Sub-Activity Risk**

In Figure 7, regarding the compared algorithms, it can be observed that the Random Forest algorithm performs better in predicting the risk of sub-activities. Therefore, the results obtained from this algorithm are considered as the criterion for predicting this risk.

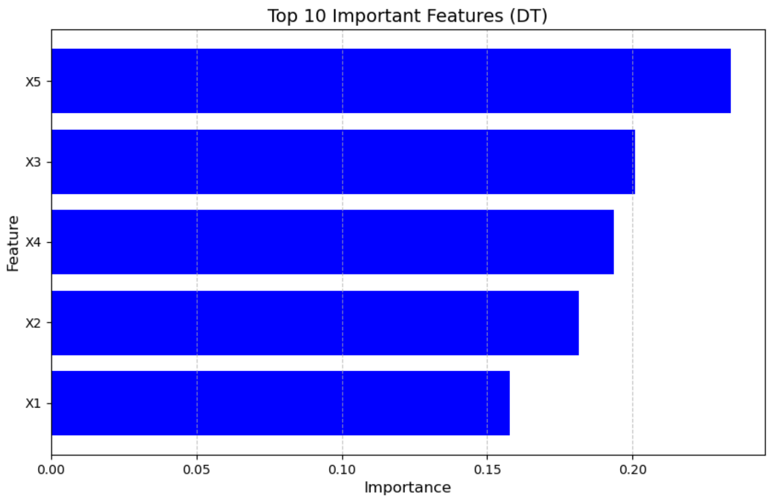


**Figure 8. Comparison of Factors Influencing Sub-Activity Risk According to the Decision Tree Algorithm**

Based on Figure 8, the frequency of partial system failures in the production unit, followed by the variables of the frequency of sub-activities performed below expectations, and the frequency of total system failures in the production unit are ranked.

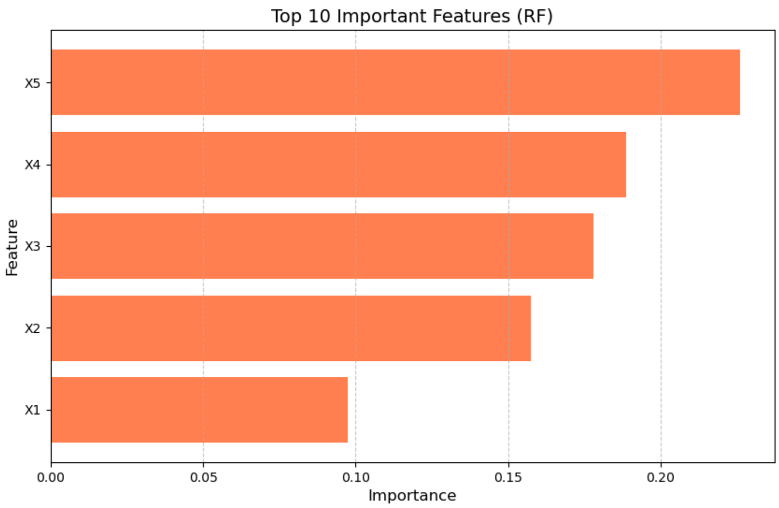


**Figure 9. Comparison of Factors Affecting the Risk of Sub-Activities in Terms of the Random Forest Algorithm**

Since the random forest algorithm is the superior algorithm in the present study, the ranking results by this algorithm are the benchmark. The results show that the variable of the number of partial system failures in the production unit is considered as the most important factor of the risk of the sub-activity, followed by the number of total system failures in the production unit, the number of times the sub-activity is performed below the expected level, the number of times the sub-activity is stopped, and the number of times the sub-activity is delayed.

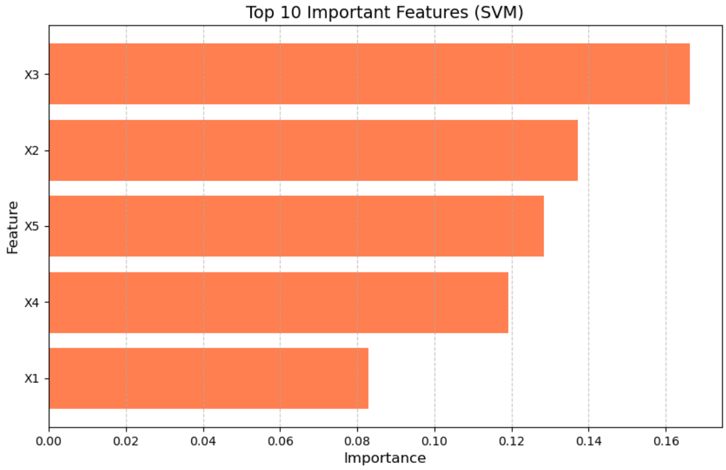


**Figure 10. Comparison of Factors Affecting the Risk of Sub-Activities in Terms of the Support Vector Machine Algorithm**

The most important factor affecting the risk of sub-activities is the number of times the sub-activity is performed lower than expected. The mathematical model is subsequently validated and solved. After the risk values of sub-activities and production units are obtained, these parameters are entered into the mathematical model and the model is solved based on their values. The results of solving the model are presented in this section. Before that, the model is solved using small dimensions to check the validity of the model. First, the dimensions of the model are introduced, as presented in Table 4. It should be noted that the examples are collected from sample data.

**Table 4. Dimensions of the Model**

| Examples | Production Units | Services | Activities | Sub-Activities | Situations | Quality |
|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 3 | 6 | 1 | 1 |
| 2 | 4 | 1 | 6 | 12 | 1 | 1 |
| 3 | 6 | 2 | 9 | 18 | 2 | 1 |
| 4 | 8 | 2 | 12 | 24 | 2 | 2 |
| 5 | 10 | 3 | 15 | 30 | 3 | 2 |
| 6 | 12 | 3 | 18 | 36 | 3 | 2 |
| 7 | 14 | 4 | 21 | 42 | 4 | 2 |
| 8 | 16 | 4 | 24 | 48 | 4 | 3 |
| 9 | 18 | 5 | 27 | 54 | 5 | 3 |
| 10 | 20 | 5 | 30 | 60 | 5 | 3 |
| 11 | 22 | 6 | 33 | 66 | 6 | 3 |
| 12 | 24 | 6 | 36 | 72 | 6 | 4 |
| 13 | 26 | 7 | 39 | 78 | 7 | 4 |
| 14 | 28 | 7 | 42 | 84 | 7 | 4 |
| 15 | 30 | 8 | 45 | 90 | 8 | 4 |
| 16 | 32 | 8 | 48 | 96 | 8 | 4 |
| 17 | 34 | 9 | 51 | 102 | 9 | 5 |

| Examples | Production Units | Services | Activities | Sub-Activities | Situations | Quality |
|---|---|---|---|---|---|---|
| 18 | 36 | 9 | 54 | 108 | 9 | 5 |
| 19 | 38 | 10 | 57 | 114 | 10 | 5 |
| 20 | 40 | 10 | 90 | 120 | 10 | 5 |

The model is then solved in these dimensions to verify its validity. The results of solving the model in different dimensions are presented in the table below.

**Table 5. Solving the Model in Different Dimensions**

| Examples | Completion Time | Cost | Calculation Time |
|---|---|---|---|
| 1 | 239 | 18486 | 10 |
| 2 | 249 | 20947 | 17 |
| 3 | 259 | 23461 | 24 |
| 4 | 269 | 27450 | 32 |
| 5 | 288 | 29588 | 42 |
| 6 | 300 | 32970 | 48 |
| 7 | 315 | 36419 | 54 |
| 8 | 325 | 38824 | 62 |
| 9 | 335 | 41422 | 72 |
| 10 | 350 | 45000 | 81 |
| 11 | | | low memory |
| 12 | | | low memory |
| 13 | | | low memory |
| 14 | | | low memory |
| 15 | | | low memory |
| 16 | | | low memory |
| 17 | | | low memory |
| 18 | | | low memory |
| 19 | | | low memory |
| 20 | | | low memory |

As can be seen, the objective function values increase with increasing dimensions and cannot be solved further up to the tenth example. Therefore, the validity of the model can be confirmed according to the solution results and the response of the objective function values to it. It is worth mentioning that the exact method can only be solved up to the tenth example, and from this example onwards, metaheuristic algorithms should be used to solve the model to obtain an almost optimal solution.

To adjust the parameters, the metaheuristic algorithm uses the sensitivity analysis method in such a way that other parameters are kept constant, and by changing some important parameters, the best solution is obtained. The optimal solution is considered as the criterion for the results of parameter adjustment; that is, the more a parameter achieves optimal solutions, the more positive its performance is. The summary of the parameter adjustment results is as follows:

**Table 6. Parameter Setting of NSGAII Algorithm**

| Parameter | Value |
|---|---|
| Initial population | 75 |

| Parameter | Value |
|---|---|
| Number of replications | 100 |
| Crossover rate | 0.7 |
| Mutation rate | 0.03 |

Next, the model is solved in real dimensions. The exact method is carried out using the epsilon constraint method in the GEMS software. It should be noted that the implementation of the exact method with the epsilon constraint method was carried out using a CorI5 CPU, with a RAM of 8 GB, and using the CPlex solver, as observed. It was only able to solve the model up to the tenth example. Therefore, the model is completely solved using the NSGAII algorithm, the results of which are presented below. Additionaly, the results are compared with and without machine learning algorithms. The results are presented in Table 7.

**Table 7. Comparison of Results Using Machine Learning Algorithms and Not Using Them**

| Examples | With Machine Learning | | Without Machine Learning | | Percentage Improvement | |
|---|---|---|---|---|---|---|
| | Completion Time | Cost | Completion Time | Cost | Completion Time | Cost |
| 1 | 223 | 19655 | 226 | 19830 | 0.013 | 0.009 |
| 2 | 238 | 22419 | 243 | 22589 | 0.021 | 0.008 |
| 3 | 256 | 24833 | 258 | 24944 | 0.008 | 0.004 |
| 4 | 272 | 27661 | 275 | 27772 | 0.011 | 0.004 |
| 5 | 288 | 31072 | 290 | 31179 | 0.007 | 0.003 |
| 6 | 298 | 33921 | 302 | 34094 | 0.013 | 0.005 |
| 7 | 311 | 36219 | 314 | 36414 | 0.010 | 0.005 |
| 8 | 323 | 38360 | 324 | 38516 | 0.003 | 0.004 |
| 9 | 338 | 41317 | 340 | 41480 | 0.006 | 0.004 |
| 10 | 350 | 45000 | 354 | 45126 | 0.011 | 0.003 |
| 11 | 360 | 47719 | 361 | 47844 | 0.003 | 0.003 |
| 12 | 378 | 51227 | 381 | 51377 | 0.008 | 0.003 |
| 13 | 394 | 54090 | 397 | 54240 | 0.008 | 0.003 |
| 14 | 408 | 57110 | 411 | 57221 | 0.007 | 0.002 |
| 15 | 423 | 59135 | 424 | 59266 | 0.002 | 0.002 |
| 16 | 443 | 61590 | 448 | 61747 | 0.011 | 0.003 |
| 17 | 463 | 64384 | 468 | 64565 | 0.011 | 0.003 |
| 18 | 476 | 68276 | 481 | 68474 | 0.011 | 0.003 |
| 19 | 495 | 70721 | 497 | 70859 | 0.004 | 0.002 |
| 20 | 513 | 73045 | 515 | 73190 | 0.004 | 0.002 |

As can be observed in Table 7, the values of the objective functions, i.e., completion time and cost, are in a more optimal state and are lower in the case of using machine learning. Considering that the completion time and cost achieve a more optimal solution if they are lower, in the case of using machine learning and using it, the results of the objective functions can be in a better state in all examples, indicating the superiority and optimality of using the machine learning algorithm. The percentage of improvement is presented separately for each example, which mainly indicates less than 1% to slightly more than 1% improvement in time and cost, which overall highlights the positive performance of machine learning and its use in the multi-objective multi-level scheduling model.

## 5) Conclusion and Suggestions

The present study used a combined machine learning and meta-heuristic approach to implement a cloud production model. Machine learning algorithms were used to predict the execution time of sub-activities and also to determine the risk level of sub-activities. The results show that the random forest algorithm has a significant advantage over other algorithms in terms of classification criteria, and therefore, it is considered the superior algorithm of the present study; the results obtained are the criteria for classification and prediction of existing parameters. Subsequently, the results of solving the optimization model show that the NSGAII algorithm is able to solve the cloud production scheduling model for large and real-world examples. On the other hand, it can be seen that using the machine learning algorithm leads to more optimal solutions than not using it. Therefore, machine learning algorithms have led to improved solutions in the present research problem.

Among the time parameters, the parameter that affects the cost more than other parameters is the activity time parameter, with the startup time parameter being in the second place. Among the cost parameters, preparation cost has the greatest impact on time. Transportation cost then shows the greatest impact. Among the cost parameters, the most important one is the activity cost. Additionally, transportation cost has the greatest impact on cost.

Based on management concepts, although the result of using machine learning algorithms in the present study has led to a 1% improvement in results, this improvement will be very significant in high-volume production. Therefore, it can be said that the result of the present study justifies the performance of large, high-performance organizations, and perhaps the results may not seem very suitable and appropriate for small and medium-sized enterprises. However, the performance of machine learning algorithms in improving results is considered positive overall.

Considering the results of the present study, the future research model can operate as a scenario and consider some parameters such as time and cost in the form of different scenarios. The objective function of the future problem can be expanded beyond the objectives of the present study, and items such as quality and risk can be considered as objective functions. Employing deep learning models, such as graph neural networks (GNNs), to model complex interactions between production units as well as predicting disruption risks at the system level, which is a natural extension of the networked structure of cloud production, are also suggested. Future research could add other time and cost parameters to the model, thereby expanding the current model.

## References

Akhavan Hariri, M., Khashkhashimoghadam, S., & Fatahi Valilai, O. (2025). A multi-product sustainable scheduling model focusing on logistic service sharing in cloud manufacturing systems. *Process Integration and Optimization for Sustainability*, 1-17. https://doi.org/10.1007/s41660-025-00532-9

Bandaru, S., Ng, A. H., & Deb, K. (2017). Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey. *Expert Systems with Applications*, *70*, 139-159. https://doi.org/10.1016/j.eswa.2016.10.015

Fomin, D., Makarov, I., Voronina, M., Strimovskaya, A., & Pozdnyakov, V. (2024). Heterogeneous graph attention networks for scheduling in cloud manufacturing and logistics. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3522020

Govindan, K., Balasundaram, R., Baskar, N., & Asokan, P. (2017). A hybrid approach for minimizing makespan in permutation flowshop scheduling. *Journal of Systems Science and Systems Engineering*, *26*(1), 50-76. https://doi.org/10.1007/s11518-016-5297-1

Halty, A., Sánchez, R., Vázquez, V., Viana, V., Piñeyro, P., & Rossit, D. A. (2020). Scheduling in cloud manufacturing systems: Recent systematic literature review. https://doi.org/10.3934/mbe.2020377

He, W., & Xu, L. (2015). A state-of-the-art survey of cloud manufacturing. *International Journal of Computer Integrated Manufacturing*, *28*(3), 239-250. https://doi.org/10.1016/j.procir.2018.03.055

Helo, P., Phuong, D., & Hao, Y. (2019). Cloud manufacturing–scheduling as a service for sheet metal manufacturing. *Computers & Operations Research*, *110*, 208-219. https://doi.org/10.1016/j.cor.2018.06.002

Hemmati, A., Motevalli, S. H., Pourghader Chobar, A., Akhlaghpour, A., & Nazari, L. (2025). Analyzing customer sentiment with AI to improve the smart supply chain. *Engineering Management and Soft Computing*, *11*(1), 306-286. https://doi.org/10.22091/jemsc.2025.3654.1260

Keihani, H., Kaveh, F., Talebi, S., Shirazi, H. K., & Chobar, A. P. (2026). A multi-objective optimization framework for enhancing the resilience and self-sufficiency of photovoltaic systems in smart cities. *Solar Compass*, 100156. https://doi.org/10.1016/j.solcom.2026.100156

Liu, B., & Zhang, Z. (2017). QoS-aware service composition for cloud manufacturing based on the optimal construction of synergistic elementary service groups. *The International Journal of Advanced Manufacturing Technology*, *88*(9), 2757-2771. https://doi.org/10.1007/s00170-016-8992-7

Moghaddam, M., Silva, J. R., & Nof, S. Y. (2015). Manufacturing-as-a-service—From e-work and service-oriented architecture to the cloud manufacturing paradigm. *IFAC-PapersOnLine*, *48*(3), 828-833. https://doi.org/10.1016/j.ifacol.2015.06.186

Pan, H., Bayanati, M., Vaseei, M., & Chobar, A. P. (2023). Empowering solar photovoltaic logistic operations through cloud-enabled blockchain technology: a sustainable approach. *Frontiers in Energy Research*, *11*, 1293449. https://doi.org/10.3389/fenrg.2023.1293449

Ren, L., Zhang, L., Zhao, C., & Chai, X. (2013, June). Cloud manufacturing platform: operating paradigm, functional requirements, and architecture design. In *International Manufacturing Science and Engineering Conference* (Vol. 55461, p. V002T02A009). American Society of Mechanical Engineers. https://doi.org/10.1115/MSEC2013-1185

Zhang, Q., Li, N., Duan, J., Qin, J., & Zhou, Y. (2024). Resource scheduling optimisation study considering both supply and demand sides of services under cloud manufacturing. *Systems*, *12*(4), 133. https://doi.org/10.3390/systems12040133

Zhang, Y., Wang, J., Liu, S., & Qian, C. (2017). Game theory based real-time shop floor scheduling strategy and method for cloud manufacturing. *International Journal of Intelligent Systems*, *32*(4), 437-463. https://doi.org/10.1002/int.21868

Zhou, L., Baldacci, R., Vigo, D., & Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research*, *265*(2), 765-778. https://doi.org/10.1016/j.ejor.2017.08.011