International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I1P111
Volume 74 Issue 1, 142-151, January 2026

Original Article

A Hybrid Intelligent System for Adaptive Project Scheduling Using Machine Learning, Simulation, and Deep Reinforcement Learning

Issam TALKAM[1], Ibrahim HAMZANE[2], Abdessamad BELANGOUR[3]

[1,3]*Laboratory of Artificial Intelligence and Systems, LAIS, Hassan II University, Faculty of Sciences Ben M'Sik, Casablanca, Morocco.*
[2]*Safi's Graduate School of Technology, Cadi Ayyad University, Morocco*

[1]*Corresponding Author : jtalkam@gmail.com*

Received: 15 August 2025 Revised: 08 December 2025 Accepted: 25 December 2025 Published: 14 January 2026

Abstract - *Project schedule management has long remained one of the unresolved issues, particularly in dynamic and uncertain environments where conventional methods are mostly inadequate for handling disturbances. This article introduces an intelligent and modular scheduling framework that integrates supervised machine learning, metaheuristic optimization, simulation, and deep reinforcement learning. The system employs machine and deep learning models, including Support Vector Machines (SVM), Random Forest, and Long Short-Term Memory (LSTM) networks for task duration prediction as well as delay detection and classification. The optimization components use Genetic Algorithms and Particle Swarm Optimization to produce efficient schedules that are both timely and resource-conscious. In addition, Monte Carlo simulation and fuzzy logic are applied to address uncertainty, while deep reinforcement learning autonomously selects the best rules to keep the system adaptable in real time. The study is validated by implementing the concept within the existing infrastructure using synthetic project data of complex types that include task dependencies, different risk levels, and stochastic disturbances. The experimental outcomes indicate that the proposed technique is not only flexible but also features self-healing capabilities, allowing it to respond to environmental changes without human intervention. The resulting method maintains task prediction accuracy and resilience, representing a promising direction in the field of intelligent scheduling research.*

Keywords - *Project Scheduling, Machine Learning, Reinforcement Learning, Metaheuristic Optimization, Monte Carlo Simulation, Dynamic Environments.*

1. Introduction

Project scheduling is arguably one of the most vital and complicated factors of project management, particularly when stable and certain conditions cannot be assumed. It has been shown that standard project management instruments such as the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) cannot efficiently handle real-time uncertainties and overlapping task relationships [1, 2]. Due to the increasing amount of project data in various sectors, traditional scheduling methods have become increasingly ineffective in terms of adaptability and responsiveness in the construction, computing, and infrastructure sectors [3].

These industries are frequently trapped in a cycle of delays, resource shortages, and contractual penalties resulting from unpredictable risks such as weather events, supply disruptions, and workforce fluctuations. Consequently, the failure to adjust schedules dynamically in real-time may cause exorbitant costs and project inefficiencies of great magnitude. Researchers have sought to address this issue by developing AI-driven scheduling systems, which can be represented as predictive, adaptive, and optimized project scheduling [4, 5]. Machine learning techniques, e.g., Random Forest, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM), have been implemented to estimate the duration of a task and locate delays based on historical or real-time data and thus have gained a lot of popularity recently [6, 7]. Uddin et al. [7], for instance, focus solely on delay prediction through Deep Learning Methods, while they do not discuss adaptive control or resource-constrained optimization. In the same way, Wei and Rana [6] came up with data mining methods for delay identification, but they did not facilitate simulation or reinforcement learning. Recently, Pal et al. [8], for example, have developed a natural-language-based assistant for construction scheduling, thereby making progress in user interaction, but real-time adaptability is still missing. The system proposed in this paper is a step towards connecting these voids by an end-to-end hybrid architecture that integrates forecasting, optimization, uncertainty modeling, and dynamic adaptation within one framework.

Around the same time, metaheuristic algorithms, such as the Genetic Algorithms (GA) or Particle Swarm Optimization (PSO), are famous for considering models of constrained resources. They also have the feature of avoiding scheduling problems, and in this role, they have been utilized for quite a long time [9, 10]. Moreover, the simulation of the project's probabilistic aspects is performed using Monte Carlo and fuzzy logic methods that provide the additional power of robustness and the capability of considering numerous approaches [11, 12]. Deep Reinforcement Learning (DRL), a new sub-area of AI, can grant the system the decision-making power and thus genuine adaptivity through decision-making agents that keep interacting with the project environment to figure out the most efficient scheduling strategies [13].

Despite recent advances, current AI-based approaches remain fragmented. To address this limitation, this study introduces a unified, modular framework integrating prediction, optimization, uncertainty modeling, and adaptive control. This research proposes a next-generation scheduling system that aims to consolidate the complementary AI capabilities necessary for robust and adaptive planning. This is the first piece of work that combines machine learning–based prediction, GA–PSO hybrid optimization, Monte Carlo and fuzzy–based robustness simulation, and deep reinforcement learning in a fully modular and end–to–end scheduling system. The suggested system is put to the test through synthetic project data with complex task dependencies, stochastic risks, and real-time disruptions, demonstrating its effectiveness in managing uncertainty and enhancing real-time adaptability.

2. Related Work

An AI-powered project scheduling system has received significant attention in numerous research papers. A common theme among these papers is that the system should be able to provide solutions that are flexible, accurate, and scalable. The different writers have proposed various ideas for the resolution of this complicated issue.

In [1], the Author emphatically points out the importance of project planning and scheduling and, at the same time, states that using traditional methods in a rapidly changing environment is impractical. Bibliometric and scientometric reviews [3, 14, 15] provide a comprehensive overview of scheduling evolution and identify key milestones, particularly in innovative scheduling systems for the construction and IT sectors.

The combination of machine learning and metaheuristics as intelligent systems has been proven to be a good idea in several studies. As an illustration, [6] has advanced a proposal for the creation of AI-run platforms that cover schedule forecasting and buffer management. A good example is [11], which has taken a few steps in that direction and identified the benefits of probabilistic simulation in uncertain situations.

Several studies have reported that deep learning can be a source of accurate delay prediction, especially in the case of LSTM and DNNs. For example, [7] brought out the connection between deep learning and schedule delay prediction. Besides, [2] was instrumental in understanding how artificial intelligence with genetic algorithms and Support Vector Machines (SVM) work together to achieve high-quality scheduling as the end goal. These models help to improve the prediction of task duration and assessment of the risk level. Recent work has delved deeply into the use of AI in construction scheduling. As an example, Pal et al. [8] invented a natural language-based scheduling assistant for construction projects and had very promising results in the way it understood the textual project data. On the other hand, their method is heavily concentrated on language understanding only, without any integrated forecasting, optimization, and adaptive control, which the proposed hybrid framework is designed to do.

Another study [2] compared metaheuristic techniques such as GA and PSO in terms of their practicability for resource-constrained project scheduling. Moreover, the employment of fuzzy logic and Monte Carlo simulation has been revealed as more beneficial in dealing with uncertainty in [16, 17].

One of the conceptual frameworks that has been put forward to merge these approaches is [18], which outlines the architecture for a modular resilient scheduling system consisting of AI agents able to learn from uncertain events. Furthermore, [10, 13] not only provide intelligent rule selection systems using reinforcement learning and AI models that are integrated with corporate data, but also streamline automated adaptation.

As far as the system design is concerned, [19] proposes the use of AI technologies (LSTM + PSO) in the industrial project environment as proof of concept, elaborates on the strategic incorporation of AI technologies into the project management environment, and highlights the enabling role of AI technologies at the enterprise level for coordination, decision-making, and resource optimization. The improvement highlighted in the previous publications still has a major shared weakness, which is that the majority of the research works do not consider the issue of prediction, uncertainty, and adaptation in a single coherent system. The authors [4, 5] of the succeeding papers are of the opinion that the future of the scheduling system would be intelligent, modular, and autonomous software solutions that can collaborate with AI components throughout the lifetime of a project.

3. Materials and Methods

This section presents the design of the proposed hybrid intelligent scheduling system, which integrates Machine Learning (ML), metaheuristic optimization, probabilistic simulation, and Deep Reinforcement Learning (DRL). The proposed architecture is inspired by multiple recent contributions, each addressing different but complementary aspects of intelligent project scheduling. While no single study encompasses all components simultaneously, collectively they form a robust foundation for developing a modular, adaptive, and data-driven framework suitable for complex and uncertain environments [20, 22]. A strong case for combining these methods is that there is now a growing demand for systems that are not only flexible but also able to forecast task durations, handle risks, and adapt to changes immediately. Conventional methods are not sufficient in dealing with these issues when there are changes in constraints and various unexpected events that may occur [23, 24]. To name a few, Deep Reinforcement Learning (DRL)-based scheduling [12, 22] has gained a lot of traction in the field of dynamic manufacturing processes. In contrast, Monte Carlo-based risk models [11, 25] can still be relied upon for having a clear picture in the early stages of forecasting and conducting robustness analysis.

The four layers of the suggested architecture are functionally linked to each other, working as a cohesive pipeline. The Prediction Layer (ML) initially goes through historical or simulated task data to perform a classification and prediction of the possible delays. Such outputs become input features for the Optimization Layer, which uses them to develop the initial schedules that consider resource constraints and sequencing. Subsequently, the schedule is reviewed in the Simulation Layer, where random variations (e.g., Delays, Resource Availability) are generated by Monte Carlo and fuzzy logic for robustness evaluation. The DRL Layer, in contrast, is always aware of the changes in the environment and learns the most effective rescheduling strategies by interacting with the simulation output. This continual feedback loop not only ensures that the system can react to unforeseen events in real-time but also that the overall performance is maintained.

3.1. General System Architecture

The proposed architecture consists of four interconnected layers:

- Layer 1 – Prediction and Classification (ML)
- Layer 2 – Scheduling Optimization (Metaheuristics)
- Layer 3 – Uncertainty Modeling (Simulation)
- Layer 4 – Continuous Adaptation (DRL)

Each module communicates through a shared project repository and operates on a structured synthetic dataset reflecting realistic project features: resource limitations, task dependencies, and random disruptions [23, 25].

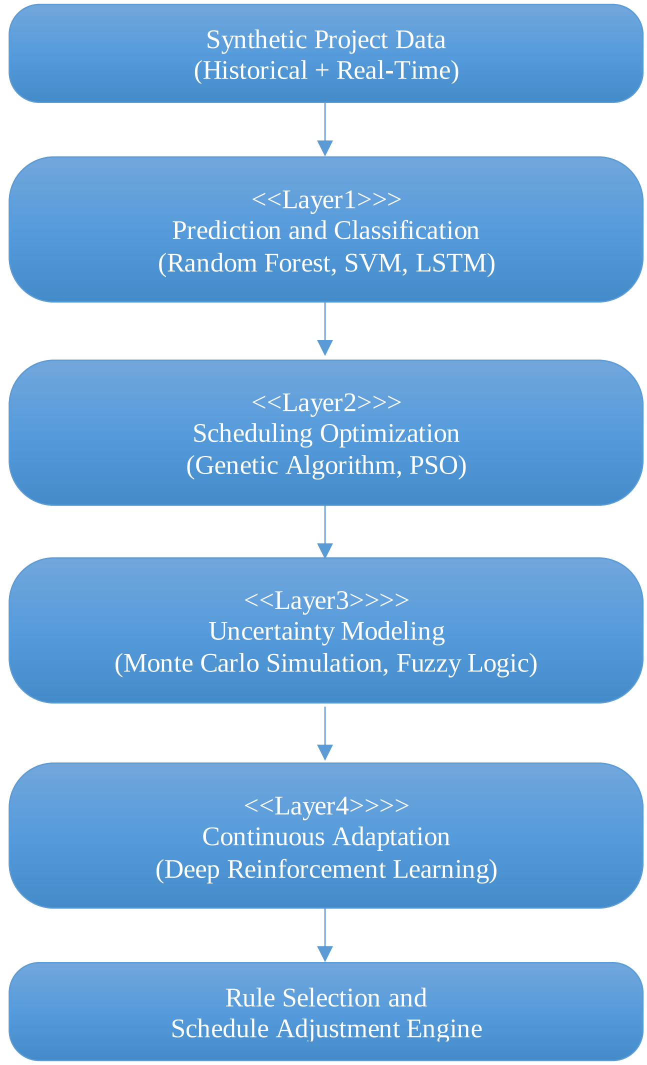

Fig. 1 Hybrid intelligent scheduling architecture

3.1.1. Mathematical Formalization of the Hybrid Architecture

To clarify the mode of the hybrid intelligent scheduling system that was proposed, the whole process is treated as a flow through four functional modules: a machine learning predictor, a metaheuristic optimizer, a robustness evaluator, and an adaptive reinforcement learning agent. The system works in a modular pipeline, where the output of each stage becomes the input of the subsequent one. The overall process is modeled as a composite function:

The functional composition represents the process:

$$S_{final} = \mathcal{F}^{\mathrm{DRL}} \circ \mathcal{F}^{\mathrm{UNc}} \circ \mathcal{F}^{\mathrm{OPT}} \circ \mathcal{F}^{\mathrm{ML}}(X) \qquad (1)$$

Where:

- X: represents the input data (historical and real-time project parameters),
- $\mathcal{F}^{\mathrm{ML}}$: is the delay prediction function based on Machine Learning Models (SVM, RF, LSTM),

- $\mathcal{F}^{\mathrm{OPT}}$: generates an initial schedule using Metaheuristics (GA, PSO),
- $\mathcal{F}^{\mathrm{UNc}}$: uncertainty modeling (Fuzzy logic and Monte Carlo),
- $\mathcal{F}^{\mathrm{DRL}}$: is the adaptive policy learned by a DRL agent to respond to real-time disruptions,
- $Sfinal$: is the adaptive and robust schedule delivered to the user.

1. Prediction Layer (ML):

$$\hat{Y} = \mathcal{F}^{\mathrm{ML}}(X) = \{\hat{d}_{\mathrm{i}}, \hat{r}_{\mathrm{i}}\}^{\mathrm{n}}{}_{\mathrm{i}=1} \tag{2}$$

With $\hat{d}_{\mathrm{i}}$ and $\hat{r}_{\mathrm{i}}$ being the estimated task duration and delay risk for task iii, respectively.

2. Optimization Layer:

$$S_{(init)} = \mathcal{F}^{\mathrm{OPT}}(\hat{Y}) \tag{3}$$

Where $S_{(init)}$ is the initial feasible schedule.

3. Robustness Evaluation:

$$S_{ro\beta ust} = \mathcal{F}^{\mathrm{UNc}}(S_{(init)}) \tag{4}$$

Representing a robustified version of the schedule through stochastic simulation and fuzzy evaluation.

4. Adaptive Adjustment:

$$S_{final} = \mathcal{F}^{\mathrm{DRL}}(S_{ro\beta ust}, E_t) \tag{5}$$

Where E_t is the system's current execution state at time t, and the DRL agent updates the schedule accordingly.

3.2. Prediction via Machine Learning

Inspired by approaches in [24, 26, 27], the ML module applies supervised learning to perform two key functions:

- Classification: Support Vector Machines (SVM) and Random Forest algorithms identify critical tasks with high delay probability.
- Forecasting: LSTM (Long Short-Term Memory) neural networks use performance histories and contextual features to predict task durations.

These models are trained on synthetic datasets that are designed to mirror interdependent project structures and dynamic performance patterns [23].

3.2.1. Training Configuration

Supervised Machine Learning Models (SVM, Random Forest, and LSTM) were trained with an 80/20 train-test split, and five-fold cross-validation was used to check generalization and robustness. LSTM model training went through 50 epochs with a batch size of 32, and the Adam optimizer was used along with a learning rate of 0.001. Mean Squared Error (MSE) was used as the loss function for regression tasks. To avoid overfitting, the procedure of early stopping was carried out on the basis of the validation loss.

The model hyperparameters (for example, SVM kernel type, Random Forest tree depth, and LSTM hidden units) were determined by performing a grid search on the training set.

3.3. Optimization Using Metaheuristic Algorithms

Based on the approaches proposed in [20, 21, 28], the optimization layer uses two complementary metaheuristic algorithms:

- Genetic Algorithms (GA): Generate initial candidate schedules by encoding task-resource mappings under precedence and resource constraints.
- Particle Swarm Optimization (PSO): Refines these candidates to minimize makespan and balance resource load while improving resilience against disturbances.

The combined GA-PSO strategy allows the system to explore large solution spaces effectively while adjusting to multi-objective requirements.

3.4. Uncertainty Modelling: Monte Carlo Simulation and Fuzzy Logic

Project environments are regularly uncertain as a result of random events that cannot be predicted. The system features the following elements to meet this challenge:

- Monte Carlo Simulation: Simulates hundreds of different situations by changing the time a task takes, the availability of resources, and the sequence of events [19, 25].
- Fuzzy Logic: Based on the work of [11, 12], fuzzy rules recreate the vagueness of the input, such as the experience of the team, the probability of the risk, and the responsiveness of the client, by adding the features of flexibility and interpretability.

These features ensure that the system is dealing with the randomness and vagueness of project planning.

3.5. Dynamic Adaptation via Deep Reinforcement Learning

Based on recent changes in DRL-based scheduling [13, 16, 22], the final stage employs a Deep Q-Network (DQN) agent:

- The agent learns strategies by experimenting with a project simulator and selecting the best outcomes for long-term performance.
- It changes the schedule promptly in the event of operational setbacks like task delay, resource conflict, or environmental changes.

Such a feature of the adaptation system, being strong and proactive, is constantly maintaining the schedule stability amidst the changing and unpredictable environment.

3.6. Experimental Dataset

In order to test the effectiveness of the proposed hybrid intelligent system, a well-structured synthetic dataset was developed to simulate project management scenarios in the real world. This synthetic dataset models the dynamic task interactions, resource limitations, risk factors, and execution uncertainties, which are common in industrial environments. The dataset consists of 100 interdependent tasks that are divided into four project categories: analysis, development, testing, and deployment. Each task record in the dataset is characterized by the main features necessary for training, optimization, simulation, and adaptation stages. The dataset schema is shown in Table 1.

Table 1. Schema and description of attributes in the synthetic project dataset

Column Name	Description
Task_ID	Unique identifier for each task
Project_ID	Project identifier
Task_Name	Descriptive task name
Duration_ Estimated	Estimated duration according to the initial plan
Start_ Date_ Planned	Planned task start date
End_Date_ Planned	Planned task end date
Dependencies	Predecessor task(s) required to start this task
Complexity_ Level	Estimated task complexity
Required_ Resources	Number of person-days required
Risk_Score	Assigned risk probability (0 = low, 1 = high)
Delay_ Observed	Actual delay recorded (used for supervised ML training)
Disruption_ Flag	True if a disruption occurred during task execution
Progress_%	Real-time execution progress percentage
MonteCarlo_ Variation	Simulated variance under stochastic conditions
Label_Delay_ Class	Delay classification
Simulation_ ScenarioID	Identifier of the stochastic simulation scenario applied

The dataset acts as the system's four core elements and accordingly determines how each module works with the specific data features. The Machine Learning (ML) module utilizes features such as Delay_Observed, Dependencies, Risk_Score, and Label_Delay_Class to train predictive models and perform classification tasks. The optimization module uses Required_Resources, Dependencies, and Complexity_Level not only to generate but also, through iterations, refine feasible schedules under resource and precedence constraints.

The simulation module uses MonteCarlo_Variation and Disruption_Flag in order to assess schedule robustness over a wide range of stochastic scenarios. Meanwhile, the DRL agent changes its scheduling policy regularly by using real-time inputs like Progress_%, disruption indicators, and simulation feedback.

The synthetic dataset was constructed on the basis of typical patterns seen in large-scale construction and IT projects so that it is as close to the real as possible. The experimental results utilizing this dataset demonstrate significant advances in the prediction accuracy, schedule robustness, and adaptability over the conventional baseline methods [13, 23, 25, 26].

3.7. Sequence Diagram of System Execution

To understand the hybrid intelligent scheduling system better, the authors present a UML sequence diagram that is depicted in Figure 2. The diagram visually represents the workflow of the system components, starting from the input provided by the user and ending with the final adaptive schedule output. Moreover, it schematically shows how the different system components work together during their on-the-spot decision-making process. In the end, the user receives an adaptive and robust schedule from the system, which completes the intelligent scheduling cycle. First, the user loads both historical and real-time project data into the Shared Data Repository, which is the primary source of the structure input. The data is fed into the Prediction & Classification layer, using SVM, Random Forest, and LSTM models, is the unit that not only predicts delays but also estimates the durations of tasks. The Scheduling Optimization component receives these predictions and uses metaheuristic methods (GA and PSO) to draft a first schedule.

The Uncertainty Modelling module is the place where Monte Carlo simulations and fuzzy logic are applied to assess the schedule's trustworthiness under various uncertainty scenarios. A robustness check leads to the schedule being re-optimized if a re-optimization request occurs; thus, the feedback loop with the optimization module is reopened to draw up another more reliable plan.

The schedule is then sent to the Continuous Adaptation layer, where the Deep Reinforcement Learning (DRL) agent supervising the execution progress can make real-time adjustments to the schedule, provided that the changes are due to disruptions or real-time execution conditions.

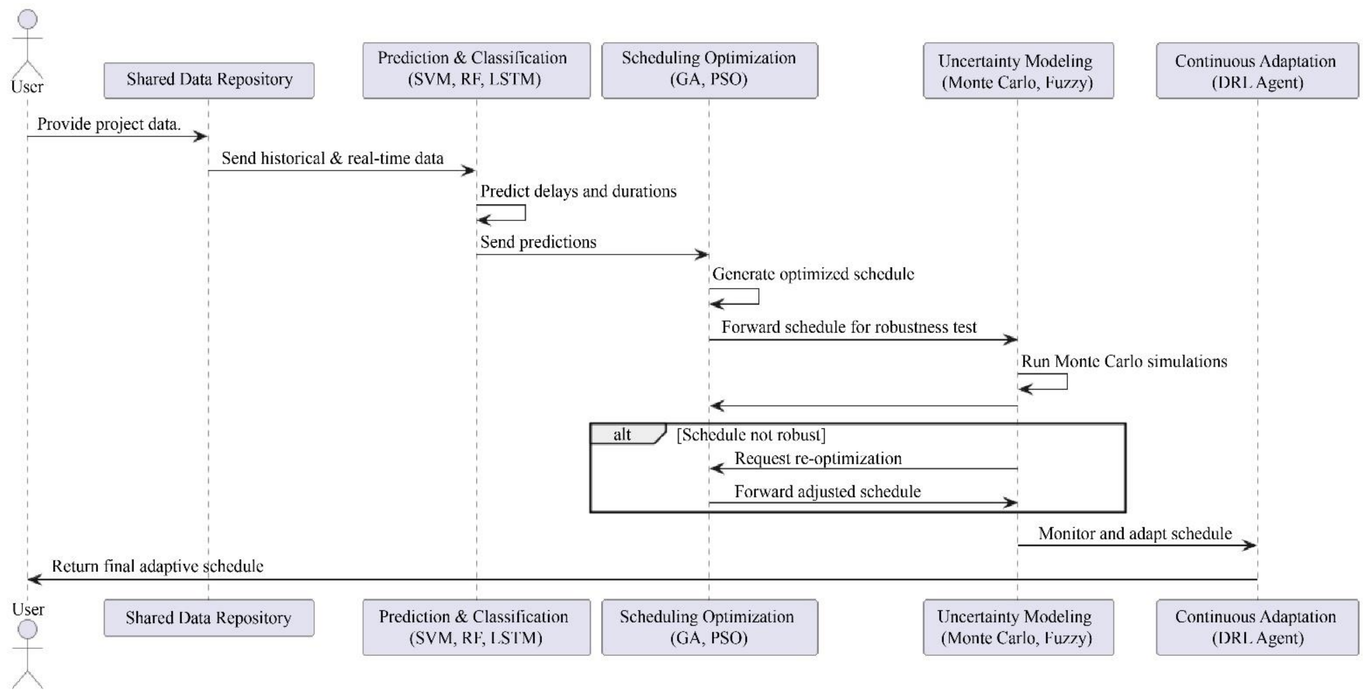

Fig. 2 System workflow from data input to adaptive scheduling

4. Results and Discussion

This section presents the testing process used to verify the effectiveness of the hybrid intelligent scheduling system proposed. It was stated in Section 3.6 that the system's evaluation was conducted through a synthetic dataset specifically designed. The criteria of the system are based on the four main features of the system, i.e., the precision of prediction, the value of the schedule optimization, the robustness of the system in the case of uncertainty, and the system's adaptive reaction to changes in the environment.

4.1. Prediction Performance (ML Module)

The Random Forest, Support Vector Machines (SVM), and Long Short-Term Memory (LSTM), three supervised learning models, were trained with features like Risk_Score, Dependencies, Delay_Observed, and Label_Delay_Class. Their performance in predicting task delay risk was measured using the metrics of classification accuracy and F1 score.

Table 2. Performance metrics of supervised learning models for project delay risk prediction

Model	Accuracy (%)	F1-Score	Key Strength
Random Forest	89.6	0.87	Best overall accuracy in delay classification
SVM	85.2	0.83	High precision for short-term delay categories
LSTM	83.4	0.89	Superior detection of medium and significant delays

Figure 3 presents a visual comparison of the three models based on their accuracy and F1-score. As shown, Random Forest achieved the highest accuracy, while LSTM led in F1-score, indicating its strength in detecting more complex delay patterns.

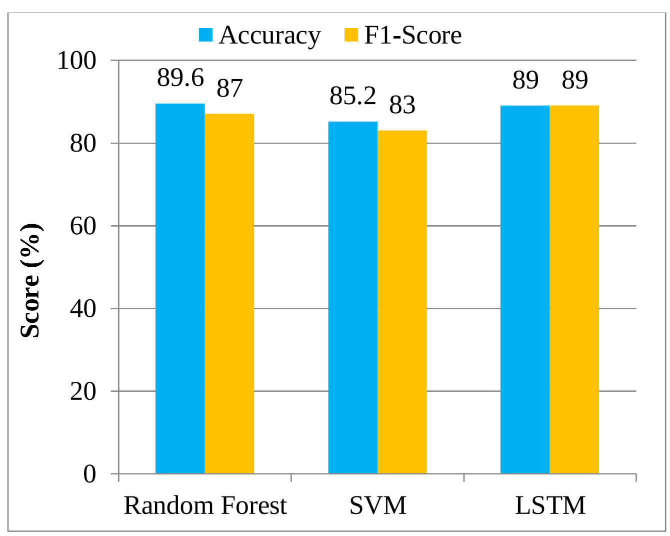

Fig. 3 Performance metrics of supervised learning models for project delay risk prediction

These results illustrate a compromise between accuracy and recall performance. Random Forest, for example, when used as a single classifier, achieved the best overall accuracy (89.6%). However, the LSTM model's F1-score (89.0%) reveals the capability of the model for recognition of not only the popular but also the rare delay classes. On the other hand, SVM was a little bit lower in both metrics but fairly demonstrated the classification without mistakes with fewer false positives. LSTM is a suitable method for temporal risk modeling because delay propagation depends on sequential patterns, where such a model can recognize them easily.

4.2. Optimization Quality (GA–PSO Scheduling Engine)

The hybrid GA–PSO optimization module was assessed based on three key scheduling metrics:

- Makespan reduction
- Resource load balancing
- Precedence constraint compliance

Table 3. Comparison of scheduling performance metrics between baseline heuristics and proposed GA–PSO hybrid optimization

Metric	Baseline (Heuristics)	GA Only	PSO Only	GA + PSO (Proposed)
Average Makespan (in days)	138	124	122	114
Resource Overload Incidents	18	13	12	9
Precedence Constraint Compliance	84.6%	91.2%	93.7%	96.3%

Both GA and PSO independently contributed to the enhancement of the performance metrics in all the cases when they were compared to the baseline heuristics. GA alone managed a 10.1% makespan reduction (from 138 to 124 days) and a 27.8% reduction of the overload incidents (from 18 to 13), while PSO resulted in 11.6% and 33.3% decreases in Makespan and overloads, respectively. Constraint compliance has been significantly increased to 91.2% with GA and 93.7% with PSO, compared to 84.6% in the baseline. The hybrid GA-PSO method was more advantageous than any of the two alone, with a 17.5% reduction in Makespan, only a few overloads, and a 96.3% compliance rate. These improvements are evidence that the hybrid approach effectively combines the GA's capability for deep exploration with the PSO's ability for fast convergence. GA is the one that ensures an exhaustive search of the scheduling space, whereas PSO is the one that can strictly adjust candidate solutions to make them more accurate and feasible.

Multi-objective optimization is especially important in dynamic project environments where the availability of resources and the dependencies of tasks may suddenly change. Through the combination of two algorithms, the system becomes more adaptable to the complexity of the problem than the methods using either one of the heuristics or single-heuristic methods. Basically, it results in quicker turnaround of projects, fewer scheduling conflicts, and higher operational stability.

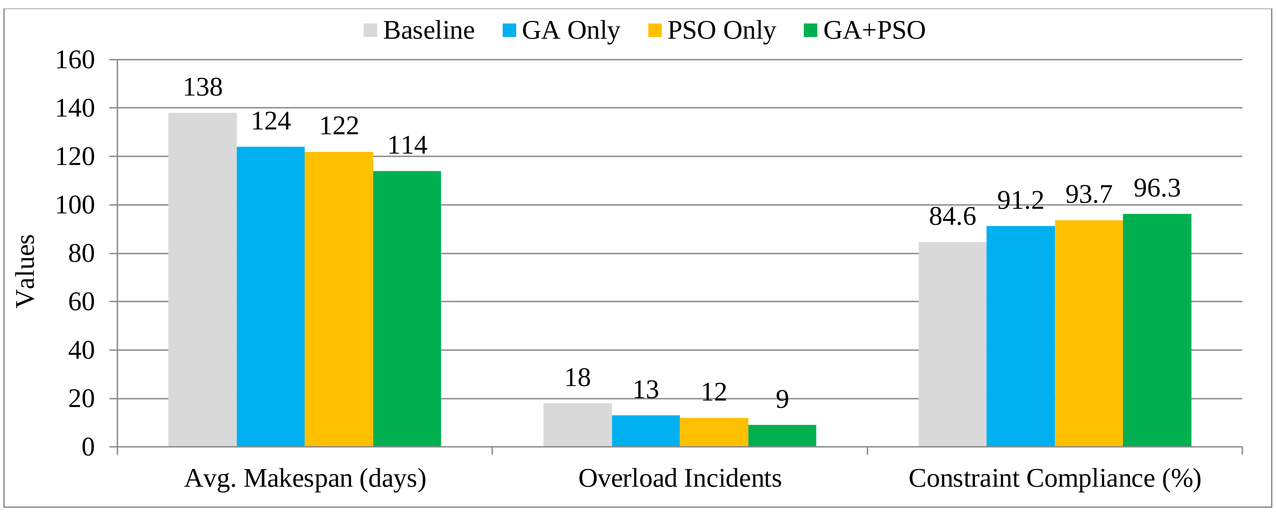

Fig. 4 Comparative performance of optimization strategies across key scheduling metrics

Table 4. Robustness evaluation metrics comparing non-optimized schedules and GA–PSO with simulation

Robustness Metric	Without Optimization	With GA+PSO + Simulation
Scenario Feasibility Rate (%)	49.3	70.1
Critical Path Sensitivity (Δ Days)	14.8	11.6
Fuzzy Risk Zone Classification Accuracy	–	91.4%

4.3. Robustness Evaluation under Simulation

Three hundred Monte Carlo simulations were run per schedule to check schedule robustness under uncertainty. Task durations were randomly changed within ±10% to ±30% using MonteCarlo_Variation, and fuzzy logic was used considering Complexity_Level and other qualitative attributes.

Figure 5 illustrates the distribution of scenario feasibility rates across the 500 Monte Carlo simulations. Without optimization, the results show a wider spread and a lower average feasibility of around 49%, indicating greater instability. In contrast, the GA–PSO optimized schedules are more concentrated around 70%, with less variation, demonstrating greater reliability under uncertainty. In practical terms, 70.1% of the optimized schedules remained feasible in over 90% of simulated scenarios, compared to just 49.3% without optimization. Figure 6 presents a comparison of the two methods in terms of the critical path sensitivity. The critical path fluctuated more in the case of no optimization, as the differences could reach even 25 days in some scenarios. By means of the GA–PSO method, this fluctuation was drastically reduced, as the majority of the values were within the 10–14-day range. Such a decrease in the range of values is a clear indication that the optimized schedules have become less reactive to uncertainty and thus planning reliability has been enhanced.

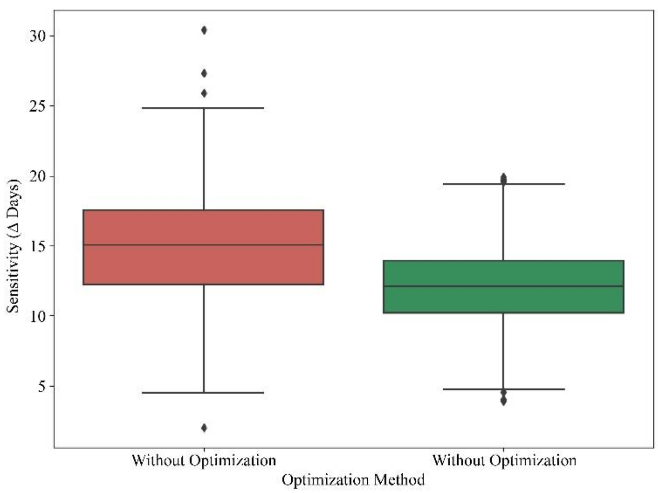

Fig. 5 Statistical distribution of scenario feasibility rates across 500 monte carlo simulations

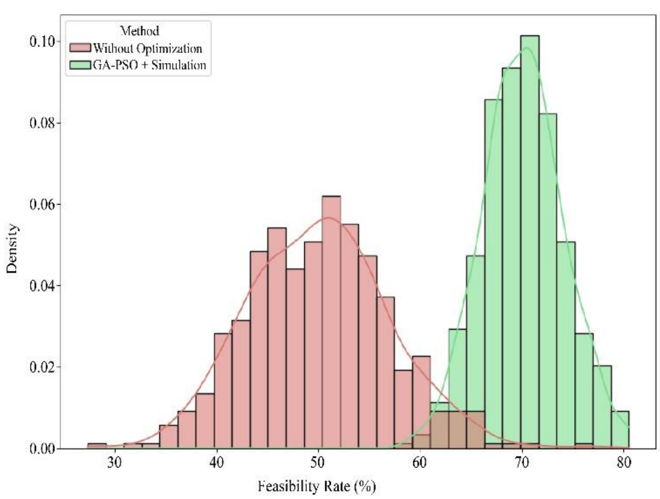

Fig. 6 Box plot of critical path sensitivity under monte carlo simulation

The GA–PSO + Simulation pipeline achieved a 91.4% classification accuracy in fuzzy risk zone assignment, based on qualitative and probabilistic attributes. This helped support human interpretability in risk-prone conditions, especially for ambiguous project phases.

4.4. Adaptive Behaviour of the DRL Agent

The Deep Q-Network (DQN) reinforcement learning agent was trained over 100 rescheduling episodes using state inputs, including Progress%, Disruption Flag, and Task Dependencies.

Table 5. Comparison of performance metrics between static re-planning and the proposed DRL-based adaptive scheduler

Performance Metric	Static Re-Planning	DRL Agent (Proposed)
Average Recovery Time	–	4.2
Recovery Success Rate (%)	72.8	90.3
Reward Convergence	–	After 60 Episodes
Average Delay Reduction (%)	–	14.2%

Based on the criteria detailed in Table 5, Figure 7 offers a visual comparison to emphasize the disparities between static re-planning and the DRL-based method. The DRL agent was far superior to the static methods in all the essential measures: it recovered the interrupted project more often, substantially decreased the delay of the project, and restored feasibility in fewer iterations. Such a graphical representation serves to confirm the agent’s capacity to adjust and interact appropriately with interruptions in changeable project situations.

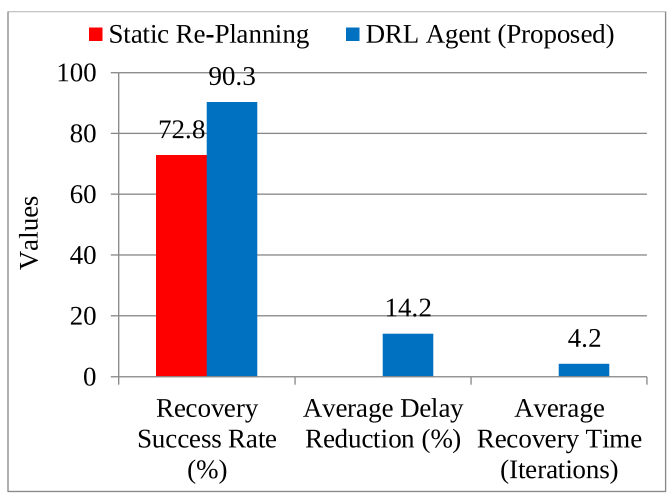

Fig. 7 Adaptive rescheduling performance: Static re-planning vs. DRL agent.

The DRL agent achieved a remarkable performance of 90% in restoring feasible schedules within five iterations of training and was also able to effectively respond to diverse disruption scenarios such as resource unavailability and task reordering. These results illustrate the DRL agent's capability to formulate adaptive recovery strategies through its interaction with the environment. A static re-planning, which always reacts in the same manner to disruptions, cannot be directly compared to an agent that modifies its operations based on the scenario context. The convergence at 60 episodes indicates the efficiency of learning. At the same time, the 90.3% recovery success rate is a very strong point for its practical application to be considered as a tool for project continuity in a dynamic environment.

4.5. Discussion

The individual performance of each module, as presented above, highlights the capabilities of each component. This section merges these results to assess the overall functioning of the system as a whole.

Table 6. Core contributions of each layer in the hybrid intelligent framework

System Layer	Primary Contribution
Machine Learning (ML)	Anticipates delays and supports informed pre-scheduling decisions
Metaheuristic Optimizer	Constructs efficient, constraint-compliant schedules
Simulation Module	Enhances robustness against stochastic variability
DRL Agent	Learns to adjust plans in real-time under disruptions reactively

The different layers of the system serve distinct but essential roles: Machine Learning provides prediction, metaheuristics generate good scheduling, simulation verifies the stability, and deep reinforcement learning enables adaptability. On its own, the proposed system forms a modular, extensible, and intelligent scheduling architecture. This system, which employs learning-based prediction, probabilistic simulation, and autonomous control, is a robust solution for unstable industrial environments such as construction, IT deployment, and manufacturing. Nevertheless, a few limitations have to be admitted. To illustrate, the DRL agent's performance depends on the reward function's design and the selection of state variables; thus, different project types may need different tunings. Besides, the robustness evaluation was based on synthetic simulation data; therefore, applying the same approach to real-world industrial datasets will serve as the next validation stage. The subsequent investigation might also consider the multi-agent reinforcement learning or attention-based models to raise the level of scalability and decision quality.

5. Conclusion and Future Work

This paper presents a hybrid intelligent scheduling system that integrates elements of machine learning, metaheuristic optimization, uncertainty modeling, and deep reinforcement learning in a single architecture for dynamic project environments. The system consists of delay prediction (using SVM, Random Forest, and LSTM), schedule optimization (using GA and PSO), robustness analysis (using Monte Carlo and fuzzy logic), and adaptive decision-making (using a Deep Q-Network agent). The results of the experiments show that the implementation of this modular and layered architecture significantly enhances adaptability, robustness, and, overall, the performance of project scheduling in uncertain environments. The hybrid GA–PSO optimizer was successful in shortening Makespan and improving constraint compliance. The fuzzy and Monte Carlo simulations validated the reliability of the produced schedules, while the DRL agent was very efficient in real-time schedule adjustments, thus it was able to achieve over 90% of recovery success during disruption scenarios.

Future work will focus on various ways of broadening this framework. For example, the use of real industrial datasets from sectors such as construction and IT would make the system more generalizable. Moreover, by mixing in more learning methods such as attention-based Neural Networks and Transformer Models, the accuracy of forecasting could be improved. Also, the reinforcement learning part could be extended to cater to multi-agent collaboration, and human-in-the-loop decision support may be added as well to give the scheduler more autonomy and make it more reliable.

References

[1] Sagar Gaur, "Understanding the Importance of Project Planning and Scheduling in Indian Construction Projects," *Journal of Positive School Psychology*, vol. 6, no. 3, pp. 3535-3544, 2022. [Google Scholar] [Publisher Link]

[2] Mandeep Singh, and Shashi Bhushan, "CS Optimized Task Scheduling for Cloud Data Management," *International Journal of Engineering Trends and Technology*, vol. 70, no. 6, pp. 114-121, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Gebrehana Derbe et al., "Scientometric Review of Construction Project Schedule Studies: Trends, Gaps and Potential Research Areas," *Journal of Civil Engineering and Management*, vol. 26, no. 4, pp. 343-363, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sofia Bento et al., "Artificial Intelligence in Project Management: Systematic Literature Review," *International Journal of Technology Intelligence and Planning*, vol. 13, no. 2, pp. 143-163, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ianire Taboada et al., "Artificial Intelligence Enabled Project Management: A Systematic Literature Review," *Applied Sciences*, vol. 13, no. 8, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Wang Wei, and Muhammad Ehsan Rana, "Software Project Schedule Management using Machine Learning & Data Mining," *International Journal of Scientific & Technology Research*, vol. 8, no. 9, pp. 1385-1389, 2019. [Google Scholar] [Publisher Link]

[7] Shahadat Uddin, Sirui Yan, and Haohui Lu, "Machine Learning and Deep Learning in Project Analytics: Methods, Applications and Research Trends," *Production Planning & Control*, vol. 36, no. 7, pp. 873-892, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[8] Aritra Pal, Danny Murguia, and Campbell Middleton, "AI-Powered Application for Construction Schedule Management using Natural Language," *Proceedings of the 2025 European Conference on Computing in Construction*, Porto, Portugal, pp. 1-8, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sarra Dahmani, Oussama Ben-Ammar, and Aïda Jebali, "Resilient Project Scheduling using Artificial Intelligence: A Conceptual Framework," *APMS: IFIP International Conference on Advances in Production Management Systems*, Nantes, France, pp. 311-320, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Shishir Tewari, and Ashitosh Chitnis, "Leveraging Graph based Machine Learning to Analyze Complex Enterprise Data Relationships," *International Journal of Novel Research and Development (IJNRD)*, vol. 6, no. 11, pp. 55-75, 2021. [Google Scholar] [Publisher Link]

[11] Fernando Acebes et al., "Stochastic Earned Duration Analysis for Project Schedule Management," *Engineering*, vol. 9, pp. 148-161, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Alvaro Cuadros, and David Ramirez, "Improvement of Risk Management in the Project Scheduling of Road Construction Projects-Case Study," *Procedia Computer Science*, vol. 239, pp. 767-772, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Kosmas Alexopoulos et al., "Deep Reinforcement Learning for Selection of Dispatch Rules for Scheduling of Production Systems," *Applied Sciences*, vol. 15, no. 1, pp. 1-26, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Shakib Zohrehvandi, and Roya Soltani, "Project Scheduling and Buffer Management: A Comprehensive Review and Future Directions," *Journal of Project Management*, vol. 7, no. 2, pp. 121-132, 2022. [CrossRef] [Publisher Link]

[15] Ali Behnam, Toby Harfield, and Russell Kenley, "Construction Management Scheduling and Control: The Familiar Historical Overview," *MATEC Web of Conferences*, vol. 66, pp. 1-5, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Maziyar Khadivi et al., "Deep Reinforcement Learning for Machine Scheduling: Methodology, the State-of-the-Art, and Future Directions," *Computers & Industrial Engineering*, vol. 200, pp. 1-45, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mira Vrbaski, Miodrag Bolic, and Shikharesh Majumdar, "Deep Reinforcement Learning Solution for Scheduling Critical Notifications in a Digital Twin Cluster," *Research Square*, pp. 1-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Sudheer Mangalampalli et al., "DRLBTSA: Deep Reinforcement Learning based Task-Scheduling Algorithm in Cloud Computing," *Multimedia Tools and Applications*, vol. 83, no. 3, pp. 8359-8387, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Chao Zhang, Max Juraschek, and Christoph Herrmann, "Deep Reinforcement Learning-Based Dynamic Scheduling for Resilient and Sustainable Manufacturing: A Systematic Review," *Journal of Manufacturing Systems*, vol. 77, pp. 962-989, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Valma Prifti, "Optimizing Project Management using Artificial Intelligence," *European Journal of Formal Sciences and Engineering*, vol. 5, no. 1, pp. 29-37, 2022. [Google Scholar] [Publisher Link]

[21] Shadi Salimimoghadam et al., "The Rise of Artificial Intelligence in Project Management: A Systematic Literature Review of Current Opportunities, Enablers, and Barriers," *Buildings*, vol. 15, no. 7, pp. 1-32, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Victor Hugo L. Lopes et al., "Deep Reinforcement Learning-Based Scheduling for Multiband Massive MIMO," *IEEE Access*, vol. 10, pp. 125509-125525, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mazen A. Al-Sinan, Abdulaziz A. Bubshait, and Zainab Aljaroudi, "Generation of Construction Scheduling through Machine Learning and BIM: A Blueprint," *Buildings*, vol. 14, no. 4, pp. 1-18, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] R. Mamatha, and K.G. Suma, "Role of Machine Learning in Software Project Management," *Journal of Physics: Conference Series: International Conference on Physics and Energy 2021 (ICPAE 2021)*, Kancheepuram, India, vol. 2040, no. 1, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Goran Avlijaš, "Examining the Value of Monte Carlo Simulation for Project Time Management," *Management*, vol. 24, no. 1, pp. 11-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[26] Yi Shuiran, and Huang Fei, "AI-Enabled Strategic Planning: Opportunities and Constraints in Automating Corporate Decision-making Processes," *Journal of Information Systems Engineering and Management*, vol. 10, no. 4, pp. 516-523, 2025. [CrossRef] [Publisher Link]

[27] Murat Pasa Uysal, and Ersin Akturk, "A Systemic Approach to Machine Learning Project Management," *IEEE Engineering Management Review*, vol. 53, no. 5, pp. 87-108, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Tetiana Portovaras et al., "Current Trends and Future Prospects in Business Management Analysis Integration," *Multidisciplinary Reviews*, vol. 7, 2024. [CrossRef] [Google Scholar] [Publisher Link]