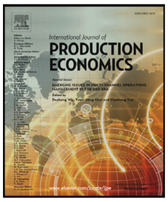
Int. J. Production Economics 291 (2026) 109841

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe



An intelligent Digital Twin based on machine learning for interpretable decision-making in manufacturing☆

Stefano Genetti, Giorgio Scarton, Marco Formentini, Giovanni Iacca*

University of Trento, Via Sommarive 9, Povo, 38123, Trento, Italy

ARTICLE INFO

Keywords:
Supply chain
Digital Twin
Decision-making
Production scheduling
Explainability
Large Language Models

ABSTRACT

In the context of Industry 4.0, several technologies converge to orchestrate improvements in business performance. Among these, Artificial Intelligence and Digital Twins stand out as some of the most promising. These two technologies are connected through the concept of intelligent Digital Twins (iDTs), which enhance standard Digital Twins with intelligent capabilities while keeping humans at the core of the process. One of the main obstacles to the broad adoption of iDTs in operations and supply chain management is the reliance on opaque AI models, which often limit trust and acceptability among operations experts and managers. To address this, it is critical to design iDTs that not only leverage the advanced capabilities of AI but also provide interpretable and actionable insights to stakeholders. In this paper, we present an action research in Adige Spa to develop an iDT framework for production scheduling. Our framework integrates interpretable machine learning techniques, employing evolutionary learning to produce decision trees that are transparent by design. Additionally, we incorporate Large Language Models to explain decision tree policies in natural language, enhancing user understanding. The framework also facilitates human interaction, allowing users to express preferences and guide the tree learning process. Results in a hybrid flow shop setting demonstrate that the proposed iDT framework delivers interpretable and effective decision-support policies while empowering users to influence and refine its outcomes, hence bridging the gap between AI-driven insights and real-world applicability.

1. Introduction

With the advent of Industry 4.0 (I4.0), the Digital Twin (DT) concept has become a cornerstone of modern operations and Supply Chain Management (SCM) (Brusset et al., 2025). Originally conceived as a digital representation of a physical entity to enhance prognostics and resource efficiency (Grieves and Vickers, 2017), DTs now encompass a wide range of applications, from cloud-based organizational models to dynamic process composition (Ivanov et al., 2022). Recent advances have extended their utility, integrating simulation and Artificial Intelligence (AI) tools to support real-time decision-making (Sharma et al., 2022; Brusset et al., 2022). Intelligent DTs (iDTs) have emerged within this domain, combining human and AI decision-making to address uncertainty, analyze complex systems, and guide stakeholders toward robust solutions (Zheng et al., 2022; Ivanov, 2023). Yet, much of the literature continues to treat DTs as primarily technological constructs, focusing on architectures or data integration, while offering limited insights into how they can be designed to facilitate actionable, trustworthy, and human-centric decision support in industrial contexts. Nevertheless, some organizations have already initiated this implementation process (Berti and Finco, 2022). For example, in 2019, DHL, in collaboration with its partner company Tetra Pak, introduced a warehouse DT to optimize logistics operations and better manage inventory and material flows (Pohl, 2019). This initiative demonstrated the potential of DT technology to improve operational efficiency, enhance decision-making, and foster seamless coordination across complex supply chain networks (Kinra et al., 2020).

Despite the promise of iDTs, their adoption faces notable challenges. In particular, in the context of SCM, most of the AI implementations rely on opaque (black-box) machine learning models (Baryannis et al., 2019), which often leads to resistance from stakeholders who demand transparency, especially when systems must comply with regulatory requirements (Commission, 2024) or undergo audits. In the literature, the gap regarding the role of humans in integrating AI to support SCM decisions has been widely recognized (Klumpp and Ruiner, 2022).

☆ This article is part of a Special issue entitled: ‘Digital Twins in Operations and Supply Chains’ published in International Journal of Production Economics.

* Corresponding author.

E-mail addresses: stefano.genetti@unitn.it (S. Genetti), giorgio.scarton@unitn.it (G. Scarton), marco.formentini@unitn.it (M. Formentini), giovanni.iacca@unitn.it (G. Iacca).

https://doi.org/10.1016/j.ijpe.2025.109841
Received 26 December 2024; Received in revised form 24 September 2025; Accepted 3 November 2025

Black-box algorithms are, in fact, perceived as difficult to test, validate, and trust before deployment (Hasija and Esper, 2022a). Consequently, there is a pressing need for methods that improve transparency and interpretability in AI-driven decision-making.

Two primary approaches have been explored in this context. The first approach, often referred to as *explainable AI* (XAI) (Kosasih et al., 2024), develops post-hoc explanations for otherwise opaque models (Barredo Arrieta et al., 2020; Longo et al., 2020). While XAI has gained popularity, its explanations still operate on inherently opaque models, which limits the extent to which stakeholders can fully trust and understand their inner workings (Ramachandram et al., 2025).

The second approach, referred to as *interpretable* AI (IAI) (Gunning and Aha, 2019), emphasizes intrinsic transparency (Zhou et al., 2024), designing models whose structure and reasoning are understandable to humans without the need for external explanation mechanisms. This category of methods can be especially valuable in dynamic and uncertain environments, such as SCM, where stakeholders require models that not only deliver accurate predictions but also justify and defend their recommendations. In this regard, integrating IAI methods with DTs offers a compelling opportunity to bridge this gap. A DT framework enhanced with IAI can, in fact, provide real-time simulation, optimization, and decision support while maintaining transparency and stakeholder trust.

Given these premises, this study focuses on a specific SCM problem, namely production scheduling, seeking to address the above-mentioned gap through the following research question:

> How can a simulation–optimization Digital Twin framework, integrated with interpretable AI methodologies, be developed to enhance production scheduling under conditions of uncertainty and variability?

To answer this question, we introduce a simulation–optimization DT framework that combines IAI methodologies, Large Language Models (LLMs), and human-AI interaction. Designed for a real-world supply chain optimization problem at a global leader in tool machine manufacturing, the framework leverages simulation to capture system stochasticity and IAI to provide transparent, actionable decision support, aligning with stakeholders' preferences for robust and interpretable tools. By incorporating an interactive user interface, the framework not only enables planners to test and evaluate production scenarios in a virtual environment but also bridges the gap between AI-driven insights and stakeholder understanding, fostering trust in the decision-making process. In this way, the DT moves beyond a passive digital replica toward an active decision-support capability, enabling planners to test scenarios, interpret AI-driven recommendations, and align outcomes with organizational priorities.

This study is grounded in action research (Coughlan and Coghlan, 2002; Bhamra et al., 2022; Panchal et al., 2024; Romano and Formentini, 2012), ensuring iterative collaboration with the partner company to refine the DT framework and enhance its usability. This approach allows for continuous feedback loops between theory and practice, addressing the persistent gap between conceptual models of DTs and their practical deployment in industrial settings (Berti and Finco, 2022). Furthermore, by experimenting with decision trees and LLMs to deliver transparent explanations, the framework demonstrates the importance of explainability and human-centric design as enablers of DT adoption.

Moreover, this study contributes to the literature on DT and production economics in different ways. First, at the *theoretical level*, it extends the conceptualization of DTs by framing them as human-in-the-loop (HITL) decision-support systems (Hines et al., 2025) that allow for combining simulation–optimization with interactive and explainable AI. This perspective moves beyond the dominant technology-centric view of DTs and emphasizes their role as organizational information processing capabilities that foster trust, adoption, and alignment with managerial decision-making. Second, at the *methodological level*, to the best of our knowledge, we are the first to develop and validate a novel DT framework that integrates IAI methodologies and LLMs to provide interpretable and user-friendly explanations of AI-driven recommendations. This integration addresses a critical barrier to the industrial uptake of advanced analytics, namely the opacity of complex AI models (Puthanveettil Madathil et al., 2025). Third, at the *practical level*, our proposed framework is implemented and refined through action research (Coughlan and Coghlan, 2002; Bagni et al., 2025) with a global tool machine manufacturer, offering actionable guidance for deploying DTs in production scheduling under uncertainty. Together, these contributions strengthen the bridge between academic research and industrial practice (Tortorella et al., 2023), positioning DTs as enablers of the transition from I4.0 to a more human-centric Industry 5.0 (Breque et al., 2021), and directly responding to the open challenges of theorizing, designing, and implementing DTs in operations and SCM (Brusset et al., 2025).

The structure of this paper is as follows: Section 2 reviews the relevant literature on DTs and their integration with AI for production scheduling. Section 3 presents the research methodology and workflow adopted for this study. Section 4 provides details on the intervention context and the production scheduling problem under analysis. Section 5 introduces the proposed framework and highlights its methodological contributions. Section 6 describes the development process of the framework, followed by an analysis of the results in Section 7. Section 8 discusses the implications of our findings, offering insights from both theoretical and managerial perspectives. Finally, Section 9 concludes the paper and suggests potential directions for future research.

2. Theoretical background

2.1. Digital twins

DTs have become central to I4.0, providing digital representations of physical systems that enable monitoring, simulation, and optimization (Biesinger et al., 2019; Zheng et al., 2022; Grieves and Vickers, 2017). While DTs were initially applied in domains such as aircraft prognostics and health management (Gockel et al., 2012; Seshadri and Krishnamurthy, 2017), they have since expanded across manufacturing and supply chains (Knapp et al., 2017; Wang et al., 2019; Ding et al., 2019; Ciano et al., 2021). According to Brusset et al. (2025), DTs in operations and SCM are characterized by four key features. First, a DT is a digital representation of physical operations or supply chains. Second, it relies on sensors and other technologies to connect to the real world and capture changes in real time. Third, it provides analytics to support managerial decision-making (Ivanov, 2024). Finally, it incorporates autonomous procedures to facilitate decision-making in industrial settings.

Indeed, a growing body of research illustrates DTs' potential to support decision-making and improve efficiency, for example, through categorization frameworks (Kritzinger et al., 2018; Sharma et al., 2022), integration with AI and big data (Qi and Tao, 2018; Min et al., 2019; Brusset et al., 2022), and case studies showcasing industrial applications (West et al., 2021; Onaji et al., 2022).

In supply chains, DTs are increasingly recognized as enablers of resilience and efficiency, with applications ranging from procurement and logistics to production planning (Ding, 2019; Semenov et al., 2020; Lee and Lee, 2021). Several studies confirm a fast-growing interest in DT-enabled supply chains, identifying key themes such as model design, integration, and quality control (Bhandal et al., 2022; Lam et al., 2023; Wang et al., 2024). Yet, despite this momentum, much of the literature remains conceptual, focusing on architectures or hypothetical scenarios, while practical implementations remain scarce (Berti and Finco, 2022). Moreover, many AI-driven approaches rely on opaque models, limiting adoption by practitioners seeking transparency and trust (Carter et al., 2023).

Brusset et al. (2025) emphasizes the need to theorize DT concepts and propose concrete methods and technologies for their design and implementation in supply chain and production contexts (Berti and Finco, 2022; Brusset et al., 2022). In particular, open challenges include clarifying how DTs enable predictive and prescriptive decision-making (Sharma et al., 2022), how human-machine interaction can support adoption (Ivanov, 2023; Kinra et al., 2020), and how DTs can be designed to enhance organizational capabilities.

2.2. Intelligent digital twins for production scheduling

AI has demonstrated significant potential in production scheduling, enhancing efficiency, adaptability, and responsiveness to dynamic industrial conditions (Del Gallo et al., 2023; Elbasheer et al., 2022). When coupled with AI, DT can optimize integrated process and production planning, as shown in deep RL and multi-agent frameworks applied in real-world manufacturing (Müller-Zhang et al., 2023; Bakopoulos et al., 2024). iDTs combine real-time system modeling, predictive and prescriptive analytics, and adaptive decision-making capabilities, enabling more autonomous and intelligent management of production and supply chain operations.

However, implementing iDTs remains challenging due to large-scale data management, integration with legacy systems, and the socio-technical complexity of industrial environments (Ivanov, 2023). Moreover, the opacity of AI models often limits adoption. HITL mechanisms address this by allowing decision-makers to guide, validate, and refine AI-driven recommendations while providing interpretable explanations, hence fostering trust and collaboration between humans and AI (Heydarbakian and Spehri, 2022; Yang, 2018; Mienye and Jere, 2024).

By embedding HITL processes within iDTs, users can actively contribute domain expertise to the AI reasoning loop, improving the relevance, reliability, and robustness of outputs. This combination of intelligent automation and human oversight creates adaptive, human-centric tools for production scheduling and SCM (Hines et al., 2025). With these capabilities, iDTs can thus balance AI-driven analytics with human intuition, supporting more informed, resilient, and transparent decision-making, and aligning with Industry 5.0 (Breque et al., 2021) principles of collaboration, sustainability, and human-centric operations.

3. Research methodology

To address the above-mentioned gap, we conducted an interventional study in a machine tool manufacturer. The intervention was derived from action research (Oliva, 2019) aimed at solving a real company problem while also contributing to scientific knowledge (Shani and Coghlan, 2021). The choice of this methodology for our empirical work has been driven by the possibility of addressing real research objectives within their natural setting (Schmidberger et al., 2009) while gathering firsthand information and creating a new workable solution. Being directly involved in the development of the solution allowed us to gather more information than the one usually achieved through other methodologies, such as case studies. Furthermore, we were able to validate our results in a real-world setting, increasing the validity of the findings and taking into consideration the criticalities of the industrial setting. Additionally, the use of action research approach allowed the company to directly implement the solution, providing practical results faster.

3.1. Research setting

The intervention took place in Adige Spa (BLM Group), a company based in Italy that is the world's leading manufacturer of high-precision, high-performance CNC laser cutting, tube bending, and shaping machines. More than half of the components used in the company's machines are outsourced, yet the company strives to maintain a lead time of just three weeks. To achieve this, long-term production scheduling is essential. Currently, production scheduling is predominantly human-driven, with planners developing schedules based on various factors, such as the expected machine delivery dates and the anticipated arrival of outsourced components. Additional considerations, including the availability of specialized workers and production capacity, are incorporated to ensure balanced operations and to avoid bottlenecks or disruptions in the production process.

The research originated as part of a broader strategic digitalization program undertaken by Adige Spa in 2017. Within this initiative, the company successfully implemented a Planning Management Assistant (PMA), which has been in use for over six years. The PMA was designed to synchronize real-time production with customer orders, ensuring the alignment between production capacity and specific customer requirements. However, the complexity, uncertainty, and interconnectedness of these factors highlighted the need for a digital solution capable of managing these elements effectively. First of all, Adige Spa adopts a multiple-sourcing strategy, which increases the complexity of goods procurement due to the number of supplier relationships that the company has to manage. Secondly, suppliers typically have visibility only on the current orders received by Adige Spa; on the other hand, the company already knows future demand, but keeps the opportunity to allocate future orders to their portfolio of suppliers. Adige Spa is aware of the benefits of information sharing with their suppliers; conversely, an excess of real-time information exchange may create "nervousness" in the supplier's behavior (Danese et al., 2013). In fact, suppliers may struggle to adapt to sudden shifts in demand, which could affect their ability to meet production schedules. This emphasizes the need for better synchronization and collaboration, enabling a more proactive and responsive SCM process.

A DT emerged as a highly suitable approach, offering a comprehensive digital replica of the production system. This not only enhanced the production process management but also created momentum to extend the scope of its application to the entire supply chain. Consequently, a broader project was launched to develop a supply chain DT. Leveraging data collected through SAP, the company initiated the development of a DT of the production process, also including the supply chain, integrated with an AI-driven demand forecasting system to support simulations, and the ERP with selected suppliers.

Aligned with the real-time paradigm established through the development of the PMA, the DT enables the company to model, simulate, and optimize operations within a virtual environment, enabling planners to respond dynamically to changes, such as delays in component deliveries or fluctuations in demand. Additionally, simulations executed through the DT provide the ability to test and evaluate various production scenarios, analyze system behavior, and proactively identify potential issues. This facilitates informed, data-driven decision-making even under conditions of uncertainty.

The visibility of customer orders within Adige Spa can extend up to six months, especially when orders are placed with a late requested delivery date. This extended visibility allows for a more informed approach to production planning, though it also requires careful coordination and flexibility. Production capacity adjustments are made on a monthly basis, which helps to outline the overall production schedule. This monthly planning serves as a framework to balance available resources and demand. However, the production schedule is continuously revised on a weekly basis to adapt to the pace of production and the timely availability of external components. This frequent revision ensures that the plan remains dynamic and can accommodate

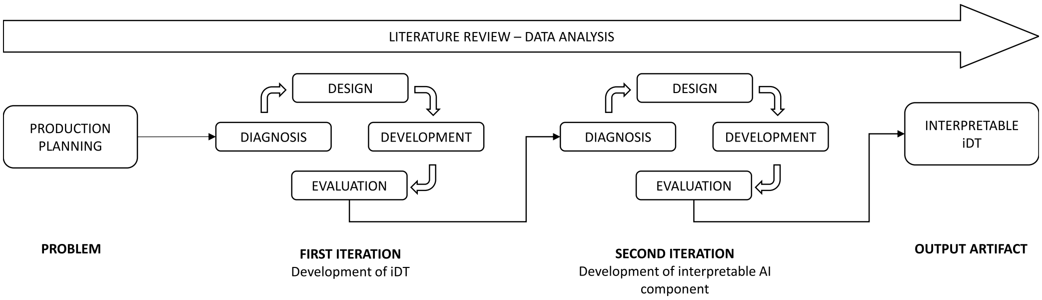


Fig. 1. Representation of the action research cycle composed of two iterations.

any unforeseen changes in production or supply. In some cases, customer orders may be shifted or delayed, or external components may not arrive on time. When this happens, there is a need for prompt intervention in the production scheduling to minimize disruptions. This flexibility is critical to avoid unnecessary storage costs or production inefficiencies. The ability to quickly adjust the production plan ensures that the company can remain agile, responding effectively to changes in both customer demands and supply chain dynamics.

In this context, a DT facilitates a deeper understanding of the system's behavior by enabling the visualization of complex interdependencies and bottlenecks. This is particularly critical in managing outsourced components, where unforeseen disruptions can cascade through the production schedule. By embedding planners' expertise into a DT, the company not only preserves and disseminates this knowledge across the organization but also enhances its planning capabilities with advanced analytics and AI-driven insights. This ensures continuity in decision-making, reduces the risk of disruption due to staff turnover, and fosters organizational resilience. Ultimately, the DT aligns long-term planning with the agility required in modern manufacturing, ensuring the company can meet its lead time goals while maintaining operational efficiency.

The DT infrastructure also enables real-time sharing of production changes at Adige Spa with its suppliers, allowing them to promptly adjust their capacity to accommodate unexpected shifts in demand. This not only benefits Adige Spa by providing a cohesive environment for monitoring the supply chain's status, but also incentivizes its suppliers to share their information, enhancing overall supply chain synchronization.

3.2. Research process

As illustrated in Fig. 1, the research adopted an iterative cycle comprising data collection, design, development, and evaluation. Throughout this process, the research team contributed with their expertise in DTs and process optimization, while the company provided critical insights into the production scheduling process and the domain knowledge required to address the specific scheduling problem. This collaborative approach ensured that the developed solution was both technically robust and aligned with the practical realities and challenges faced by the organization.

The study spanned six months and was conducted in two iterations. It began with meetings between the research team and key stakeholders from Adige Spa. Participants included the Operations Director (OD), who promoted the digitalization initiative; the Chief Information Officer (CIO), who shared insights into the company's data structure and digital architecture; and the production planning office staff, who provided hands-on insights into the processes and validated the results. The discussions highlighted the need for a solution capable of modeling the current state of the supply chain while offering simulation functionalities to integrate forecasted data (customer orders) and enhance mid-to-long-term production schedules. Additionally, to align with Industry 5.0 principles, the OD emphasized incorporating human-AI interaction, ensuring an HITL approach that complements decision-making.

Simultaneously, a comprehensive literature review was conducted to assess the current state-of-the-art in production scheduling. Internal company documents on existing production scheduling (detailed in Section 3.1) helped refine the problem model. Furthermore, historical data related to customer orders, including expected component arrival dates and delivery timelines, were shared. However, as noted by the CIO, data availability was limited due to a transition to a new ERP system in 2021, which hindered the proper migration of all required information.

To address this limitation, synthetic datasets were generated based on the structure of the provided data. These datasets were utilized during the first iteration to assess the potential of the AI solution in enhancing production scheduling and to establish a testing environment for integrating the solution into the DT framework. In the subsequent phase, the selected algorithms were tested on real-world data to validate their applicability and performance in a practical context. To further evaluate the results, the research team collaborated with the production planning office to analyze the system's effectiveness and usability. During these discussions, it became evident that improving the transparency and understandability of the system's logic was critical to fostering acceptance and adoption among the staff.

This need led to the second iteration, which focused on enhancing the interpretability of the system's recommendations. Decision trees were integrated for their transparency, and experiments with LLMs explored user-friendly explanations, bridging AI insights with stakeholder trust. During this iteration, results were progressively presented to the planners and the OD to validate their accuracy and gather feedback for refining the DT recommendations. Once the solution demonstrated satisfactory performance, discussions with the CIO were initiated to plan the industrialization of the system and its integration into the company's information systems environment.

In line with (Reason and Bradbury, 2001) view that action research should be assessed based on "quality rather than validity", our intervention was designed to meet the four criteria outlined in Levin (2003). First, the research delivered a practical solution that the company has successfully implemented. Second, the project addressed a real-world challenge by focusing on improving the production scheduling process and interpretability. Third, we worked closely with the company's team to develop a shared understanding, collaboratively interpreting information and constructing a unified perspective. Lastly, our active involvement throughout all stages of the development process underscored the strong partnership between the research team and the company.

4. Problem description and analysis

In this work, we considered a simplified segment of the supply chain of the aforementioned manufacturing company. The objective of the supply chain is to produce and deliver CNC machines for tube processing in response to customer demand. The company manufactures two series of machines: group_1 (comprising four types: m_1, m_2, m_3, m_4) and group_2 (comprising eight types: $m_5, \dots, m_{12}$), resulting in a total of twelve configurations. Each machine follows a specific sequence of production stages ($t_1, \dots, t_k$), requiring human resources with different competencies (r_1, r_2, r_3). These resources are available in sufficient numbers to work on multiple machines concurrently across 20 assembly areas. Production also depends on two semifinished components, c_1 and c_2, supplied externally. Completed machines are transported by trucks to logistics centers, with transportation time being a significant factor. Efficient scheduling of customer orders and human resource allocation on a daily basis is critical to optimizing the overall supply chain. To accurately model the situation in the DT environment and enable effective simulation, we represented the described manufacturing system as an instance of Hybrid Flow Shop Scheduling (HFS) (Ruiz and Vázquez-Rodríguez, 2010). The HFS is an operations management strategy focused on the optimal allocation of resources to tasks (or jobs) over time, with the goal of achieving specific performance objectives. In HFS, jobs pass sequentially through multiple production stages, each with parallel resources available. This configuration reflects real-world complexities better than traditional flow shop models, where each production stage is served by a single resource. In our context, jobs represent customer orders of CNC machines, each defined by a unique identifier, the machine type (m_t), a delivery deadline (dd), and the arrival dates of components c_1 (d_1) and c_2 (d_2). In our model, a single human resource can work on only one machine at a time. The production of a CNC machine can be started when an assembly area is available. Each assembly area remains occupied by a job until its completion, as preemption is not allowed. Given an input list of customer orders, the objective of the problem is to determine a schedule that optimizes performance according to a specific metric. Metrics commonly studied for HFS problems (Li et al., 2011) include the makespan (i.e., the total time required to complete all jobs), the total flow time (i.e., the sum of the time each job spends in the system), the total completion time (i.e., the sum of all job completion times), and the total tardiness (i.e., the sum of delays beyond job deadlines). This study focuses on minimizing the makespan, as this metric often reflects efficiencies across other measures. The HFS problem is known to be NP-hard, as demonstrated in Gupta (1988). This implies that finding an optimal solution in polynomial time is infeasible. Using the standard triplet notation for scheduling problems, as introduced in Graham et al. (1979), the described problem can be formulated as:

$$FHk, ((Pr_t)_{t=1}^{k}) | M_j | C_{\max}$$

where FH_k represents an HFS with k stages; $(Pr_t)_{t=1}^{k}$ indicates that each stage, $t = 1, \dots, t_k$, has r_t identical human resources available in parallel; M_j refers to eligibility constraints, such as the requirement that components c_1 or c_2 must be available before a job can proceed; $C_{\max}$ denotes that the objective is to minimize the makespan.

5. Methods

In the first iteration of the employed action research strategy (Fig. 1), we combined the simulation capabilities of a DT with various optimization methods widely recognized in the literature as state-of-the-art for process scheduling. These methods included a greedy heuristic, Optuna (Akiba et al., 2019), a Genetic Algorithm (Forrest, 1996), Ant Colony Optimization (Dorigo et al., 2006), and an RL technique, namely Proximal Policy Optimization (PPO) (Schulman et al., 2017). While these algorithms showed promising preliminary results in tackling the production planning problem of interest within the simulated environment, their opaque nature led to limited acceptance among company stakeholders, who were reluctant to trust or deploy in the field a system whose internal workings were hidden, difficult to maintain, and debug. Statements such as "*I'd like to know why the system proposes [...]*" or "*I cannot understand [...]*" by the planners and the OD were quite frequent during the meetings in which the results of the first iteration were described. This drove the second iteration of our intervention involving the interpretability of the outputs, which constitutes the core contribution of this research.

The proposed methodology, called IELDT (Interactive Evolutionary Learning Decision Trees), involves three phases:

- **Phase 1** employs an evolutionary algorithm to construct interpretable decision-making policies represented as decision trees.
- **Phase 2** translates these models into natural language descriptions for enhanced comprehensibility.
- **Phase 3** incorporates user feedback in an iterative process to refine and improve the tree-based models based on human preferences. A visual representation of the framework is provided in Fig. 2.

5.1. Evolutionary learning of interpretable decision trees

Our framework employs a flexible simulation–optimization methodology, integrating a simulation module with an evolutionary optimizer to generate interpretable decision-making policies structured as decision trees (Dhebar et al., 2020).

For the simulation module, we utilize AnyLogic (Company, 2024a), a widely used simulation tool, to create a DT of the real-world system of interest, accurately mapping all relevant details. This approach effectively simplifies the process of modeling uncertainty and stochasticity, which are inherent in industrial scenarios. For instance, in a supply chain system, transportation times might vary due to factors such as weather conditions, traffic congestion, or vehicle availability. Similarly, in a manufacturing context, the production capacity may not be in line with the requests, or qualified technicians may not be available at the right time, leading to delays and disruptions in the production process.

For the optimization component, this work builds upon the method proposed in Custode and Iacca (2023), which employs a two-level optimization scheme, implemented in Python, combining evolutionary computation and RL to train interpretable policies represented as decision trees. Notably, while the original work primarily focused on well-known yet somehow simplistic RL benchmark tasks from OpenAI (Brockman et al., 2016), it did not explore the application of this method in real-world deployment scenarios.

For the evolutionary component, the method uses Grammatical Evolution (Ryan et al., 1998) to evolve a population of pop_{size} candidate solutions (i.e., decision trees) across n_{gen} generations. Each solution in the population represents a candidate policy in the form of a tree and has a "genotype" (i.e., a numerical representation) encoded as a fixed-length list of g_l integers, where each integer is constrained to the range $[g_{\min}, g_{\max}]$.

The initial population is generated randomly, and the fitness of the corresponding solutions is evaluated (see below). At each generation, parent solutions are selected with tournament selection. In this method, t_{size} solutions are chosen randomly, and the solution with the best fitness (i.e., in our case, the lowest makespan) among them is selected as a parent. These selected parents are then combined using a two-point crossover operator (O'neill et al., 2003). This operation is applied to selected parent pairs with a uniform probability c_p. After crossover, mutation is applied to each offspring with a mutation probability m_p. For each "gene" (i.e., integer value) in a solution $\mathbf{x}$ on which mutation is applied, there is a probability m_p that it will be replaced with another random integer value between $g_{\min}$ and $g_{\max}$. The newly generated offspring solutions replace the current population using a steady-state selection replacement strategy (Custode and Iacca, 2023), where better solutions are preserved in the population, ensuring robust candidates persist across generations.

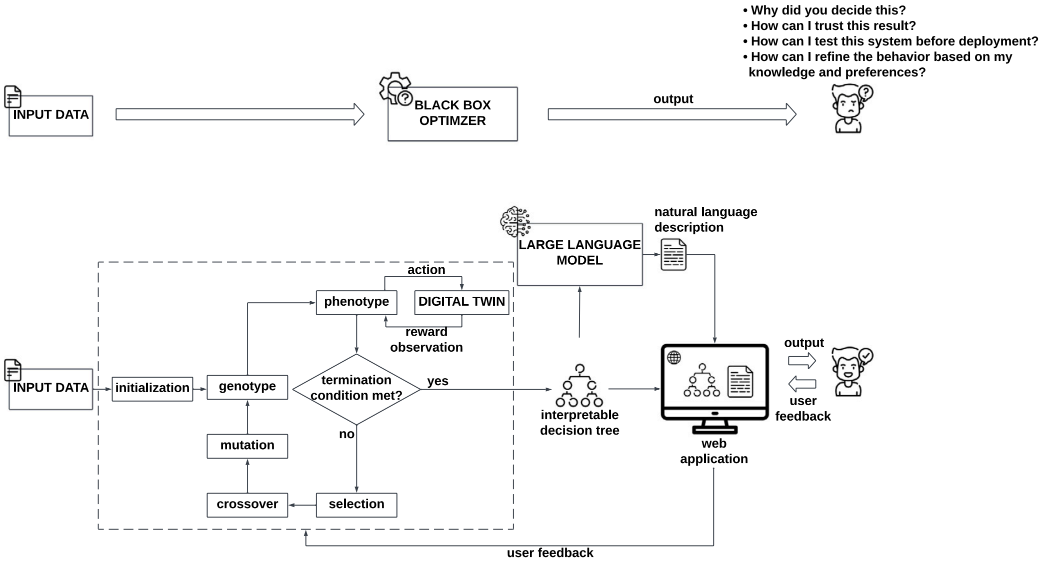


Fig. 2. Conceptual comparison between traditional scheduling approaches and the proposed IELDT framework. At the top, traditional methods use black-box models, where the internal workings are opaque to users, making trust, testing, and deployment challenging. Below is our IELDT framework, which combines evolutionary computation and reinforcement learning to train interpretable decision tree policies using DT simulations based on input data. Large Language Models generate natural language explanations of these policies. Outcomes are presented through an intuitive web application. Users can provide feedback based on the model's logic and performance, influencing the next optimization steps.

5.1.1. Fitness evaluation

Fitness evaluation involves translating the individual genotype into a phenotype (i.e., a decision tree) using production rules defined by a problem-specific grammar $\mathcal{G}$ in Backus–Naur Form (BNF) (McCracken and Reilly, 2003) as detailed in Custode and Iacca (2023). The resulting tree consists of internal nodes, which evaluate conditions, and leaf nodes s, that contain probability distribution functions $p_s : A \rightarrow [0,1]$. These functions associate each possible action $a \in A$ with a probability of being selected at leaf node s. Initially, the action probabilities in the tree's leaves are assigned randomly. The decision tree serves as a policy π that interacts with the environment, modeled as a Markov decision process, over e training episodes. The policy processes a one-dimensional input feature vector, which encodes the current state of the decision-making problem. Internal nodes evaluate conditions on the input features, while the leaf nodes provide the action $a \in A$ to be taken. Actions are selected using an ϵ-greedy strategy, where the action with the highest probability $\arg\max_a p_s(a)$ is selected with probability $1-\epsilon$, and a random action is chosen with probability ϵ. During the interaction with the environment, feedback in the form of reward r is collected. This reward is the result provided by the simulation module according to the decision performed by the policy decision tree π, which, in our case, reflects the obtained makespan (or, more specifically, its negated value, as the reward is to be maximized). Throughout the training episodes, the tree's leaf nodes are updated using Bellman's equation for Q-learning (Watkins, 1989):

$$p_s(a) \leftarrow (1-\alpha)p_s(a) + \alpha(r + \gamma \max_{a'}\{p_s(a')\})$$

where $p_s(a)$ is the probability of the last action a taken at leaf s, α is the learning rate, γ is the discount factor, r is the observed reward, and a' denotes to the next action. Ultimately, the fitness f of a solution $\mathbf{x}$, encoding the policy π, is defined as the mean return across e episodes:

$$f(\mathbf{x}) = \frac{1}{e}\sum_{i=1}^{e} R_i(\pi)$$

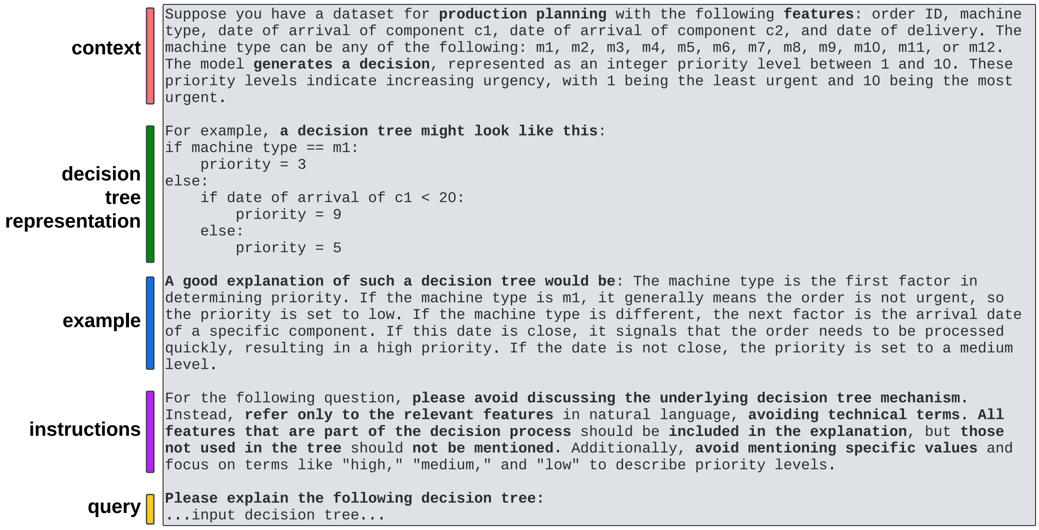
where $R_i(\pi)$ represents the total reward accumulated during the ith episode.

The Python-based optimizer and the Anylogic simulation module are integrated through the open-source package ALPypeOpt (Ecandell, 2024). This package enables the execution of simulations using the solution (and its corresponding policy) generated by the optimizer and facilitates the retrieval of corresponding simulation results.

5.2. Natural language explanation

While decision trees are inherently interpretable, as their hierarchical structure allows for straightforward tracing of decisions, certain aspects, specifically related to the domain or company dynamics, may be unclear to general users interacting with the system. This lack of clarity can potentially reduce understanding and trust in the model. To address this challenge and further enhance the interpretability of the decision trees, we propose generating natural language explanations that describe the decision-making process. To this end, we follow the methodology introduced in Ziems et al. (2023) and extended in Serafim et al. (2024), which utilizes LLMs to generate textual explanations of decision trees. The interaction with the LLM occurs exclusively through text, making prompt engineering critical for obtaining high-quality, coherent explanations. By leveraging well-structured prompts, we ensure that the generated descriptions are comprehensible and user-friendly. The key components of the prompt structure are as follows (see Fig. 3 for a complete example):

- *Context.* The prompt begins by outlining the context in which the decision-making policy is applied. It describes the features used in the decision tree, providing a brief explanation of each. Additionally, it defines the possible outcomes, i.e., the terminal values at the tree's leaves, along with their meanings.
- *Decision tree representation.* Since only textual inputs are used, the decision tree must be converted into a text-based representation. Previous research (Bhattacharya et al., 2023; Nam et al., 2024) has shown that LLMs perform well with code-like prompts. To exploit this, we developed an automated algorithm to transform the tree structure into a code-like format.
- *Example.* LLMs have demonstrated to give more appropriate responses when a demonstration is provided (Brown, 2020). To guide the model's output, we include a sample decision tree paired with a corresponding explanation. This example helps refine the LLM's responses by providing a clear reference for structure and style, improving homogeneity among the responses, and guiding the explanation toward a human-understandable description.



Suppose you have a dataset for **production planning** with the following **features**: order ID, machine type, date of arrival of component c1, date of arrival of component c2, and date of delivery. The machine type can be any of the following: m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, or m12. The model **generates a decision**, represented as an integer priority level between 1 and 10. These priority levels indicate increasing urgency, with 1 being the least urgent and 10 being the most urgent.

For example, **a decision tree might look like this:**

```
if machine type == m1:
    priority = 3
else:
    if date of arrival of c1 < 20:
        priority = 9
    else:
        priority = 5
```

A good explanation of such a decision tree would be: The machine type is the first factor in determining priority. If the machine type is m1, it generally means the order is not urgent, so the priority is set to low. If the machine type is different, the next factor is the arrival date of a specific component. If this date is close, it signals that the order needs to be processed quickly, resulting in a high priority. If the date is not close, the priority is set to a medium level.

For the following question, **please avoid discussing the underlying decision tree mechanism.** Instead, **refer only to the relevant features** in natural language, **avoiding technical terms. All features that are part of the decision process** should be **included in the explanation**, but **those not used in the tree** should **not be mentioned.** Additionally, **avoid mentioning specific values** and focus on terms like "high," "medium," and "low" to describe priority levels.

Please explain the following decision tree:
...input decision tree...

Fig. 3. Prompt structure constructed following the template outlined in Serafim et al. (2024).

- *Instructions.* Following effective prompt strategies (Brown, 2020; Serafim et al., 2024; Bubeck et al., 2023), we explicitly specify how the LLM should structure its response. For instance, we ask it to mention all features used in the decision tree, avoiding references to features not present, and refraining from using technical jargon or numerical values. By adhering to these guidelines, the LLM should generate more succinct and accessible descriptions.
- *Query.* Finally, the prompt concludes with a direct request for an explanation of the decision tree.

As detailed in Section 6, we employ multiple pretrained LLM models to generate a variety of descriptions. This allows us to emphasize different noteworthy aspects, providing users with a richer and more complete explanation.

5.3. Human-driven refinement

Periodically, after a set number of generations in the evolutionary process, the two best-performing decision trees from the current population are presented to the user together with their natural language descriptions generated by the LLMs. These results are provided through an intuitive, user-friendly web interface (Fig. 4). Through the interface, the domain expert can cast a vote for the decision tree that best aligns with their preferences. This vote directly influences the next generation by adjusting the selection pressure during the parent selection process, increasing the likelihood that the solution representing the chosen tree will be chosen as a parent for the subsequent generation. As a result, the algorithm is implicitly guided toward generating decision trees with structures similar to the one favored by the user. In this way, it is possible to exploit human knowledge to guide the automated learning of the algorithm while, at the same time, the domain expert may gain useful insights by observing the options proposed by the algorithm, increasing their own knowledge.

This human-AI interaction cycle can be repeated as many times as necessary, allowing the decision-maker to refine the proposed interpretable policy until they are satisfied with the result. An overview of the user interaction process is illustrated in Fig. 5.

6. Experimental setup

In this section, we provide an overview of the experimental design employed to evaluate the proposed framework.

Table 1
Hyperparameter configuration.

Parameter	Value	Description
g_l	100	Length of each individual genotype.
$g_{\min}$	0	Minimum value in genotype.
$g_{\max}$	40 000	Maximum value in genotype.
pop_size	20	Number of solutions in the population.
c_p	0.8	Probability of crossover between solutions.
m_p	0.6	Probability of mutation in a solution.
n_gen	100	Total number of generations.
t_size	2	Tournament size for parent selection.
α	0.001	Learning rate for Bellman's equation.
γ	0.05	Discount factor in Bellman's equation.
ϵ	0.05	Exploration rate for ϵ-greedy strategy.
e	5	Number of training episodes.

6.1. Computational setup

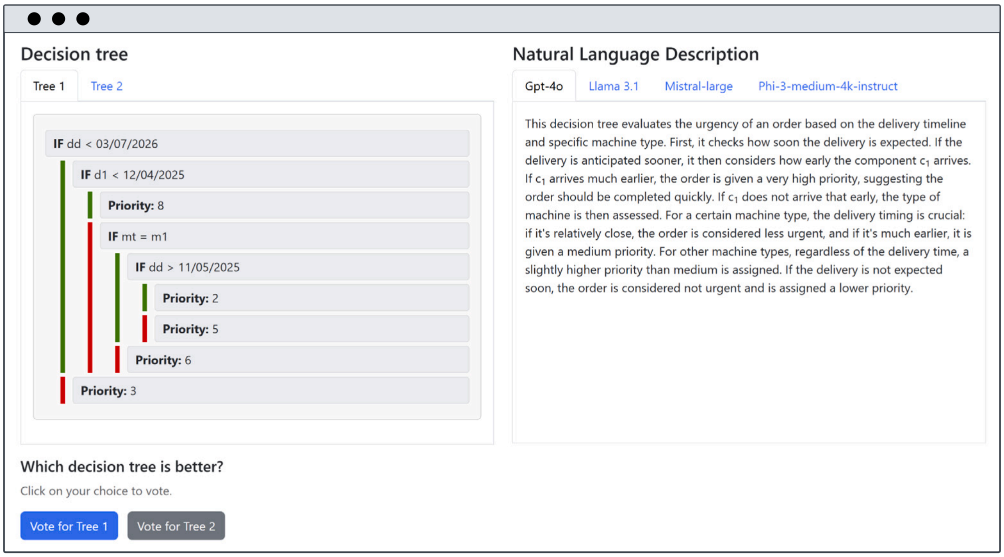
In this study, we utilized the free Personal Edition of AnyLogic 8.9.0. To ensure the solution could be implemented by the company without requiring specialized hardware, thereby minimizing implementation costs, the experiments described in this paper were conducted on a standard Windows 11 laptop. The system was equipped with a 14-core Intel i7-12700H @ 2.30 GHz and 32 GB of RAM.

6.2. Optimizer and simulation setup

At the core of IELDT lies the evolutionary learning algorithm described in Section 5.1. For the algorithm's hyperparameters, we employed empirically validated values tailored to the use case at hand. Table 1 summarizes the hyperparameter configuration. The evolutionary process was run for $n_{\text{gen}} = 100$ generations. User feedback, as described in Section 5.3, was integrated every 10 generations to guide optimization.

A key aspect of the algorithm's setup is the definition of a BNF grammar, which translates an individual genotype into its corresponding phenotype decision tree. To align with the peculiarities of the scheduling problem described in Section 4, the grammar $\mathcal{G}$ is defined as follows:

$$\begin{aligned}
\langle \text{bt} \rangle &::= \langle \text{if} \rangle \\
\langle \text{if} \rangle &::= \text{if } \langle \text{cond} \rangle \text{ then } \langle \text{a} \rangle \text{ else } \langle \text{a} \rangle \quad | \\
&\qquad \text{if } \langle \text{cond}_{\text{eq}} \rangle \text{ then } \langle \text{a} \rangle \text{ else } \langle \text{a} \rangle \\
\langle \text{a} \rangle &::= \text{leaf} \mid \langle \text{if} \rangle
\end{aligned}$$



Decision tree

- IF dd < 03/07/2026
  - IF d1 < 12/04/2025
    - Priority: 8
    - IF mt = m1
      - IF dd > 11/05/2025
        - Priority: 2
        - Priority: 5
      - Priority: 6
  - Priority: 3

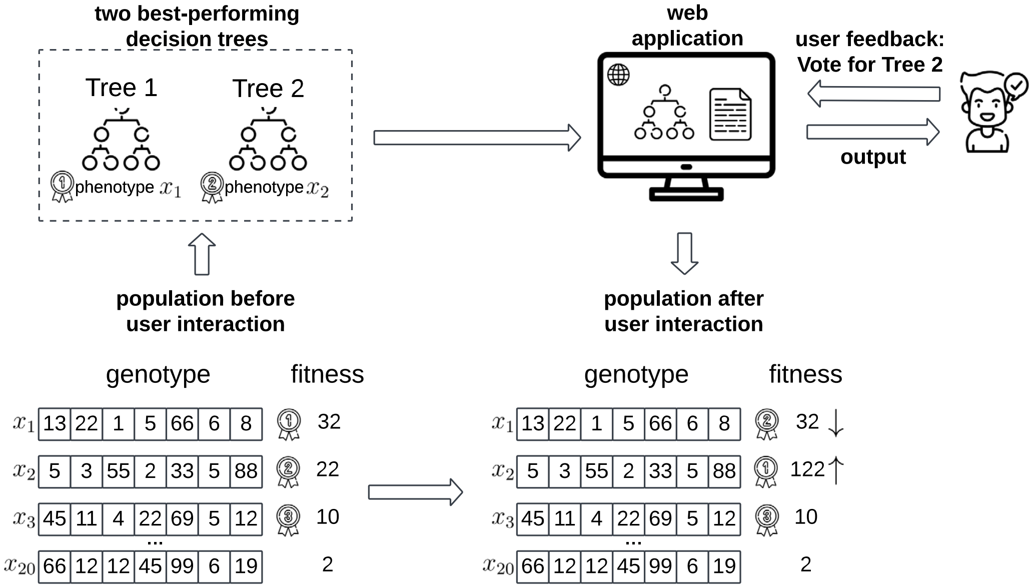
Natural Language Description

This decision tree evaluates the urgency of an order based on the delivery timeline and specific machine type. First, it checks how soon the delivery is expected. If the delivery is anticipated sooner, it then considers how early the component c_1 arrives. If c_1 arrives much earlier, the order is given a very high priority, suggesting the order should be completed quickly. If c_1 does not arrive that early, the type of machine is then assessed. For a certain machine type, the delivery timing is crucial: if it's relatively close, the order is considered less urgent, and if it's much earlier, it is given a medium priority. For other machine types, regardless of the delivery time, a slightly higher priority than medium is assigned. If the delivery is not expected soon, the order is considered not urgent and is assigned a lower priority.

Which decision tree is better?

Click on your choice to vote.

Fig. 4. A screenshot of the web application featuring a navigation bar on the left, allowing users to access the two decision trees alongside their natural language descriptions, displayed on the right for each of the four LLMs considered. The interface includes two buttons for users to vote for the decision tree that best aligns with their preferences.



	genotype							fitness
x_1	13	22	1	5	66	6	8	32
x_2	5	3	55	2	33	5	88	22
x_3	45	11	4	22	69	5	12	10
...								
x_{20}	66	12	12	45	99	6	19	2

	genotype							fitness
x_1	13	22	1	5	66	6	8	32 ↓
x_2	5	3	55	2	33	5	88	122 ↑
x_3	45	11	4	22	69	5	12	10
...								
x_{20}	66	12	12	45	99	6	19	2

Fig. 5. Schematic representation of the user interaction process. At each interaction step, the two best-performing decision trees are presented to the user through the web application. The user's vote increases the fitness of the selected tree, thereby raising its probability of being chosen as a parent during tournament selection. Numerical values shown are illustrative only and do not correspond to actual experimental results.

$$
\begin{aligned}
\langle\text{cond}\rangle &::= \langle\text{input}_1\rangle\ \langle\text{comparison}\rangle\ \langle\text{const}_1\rangle \quad | \\
&\quad\ \ \langle\text{input}_2\rangle\ \langle\text{comparison}\rangle\ \langle\text{const}_2\rangle \quad | \\
&\quad\ \ \langle\text{input}_3\rangle\ \langle\text{comparison}\rangle\ \langle\text{const}_3\rangle \quad | \\
&\quad\ \ \langle\text{input}_4\rangle\ \langle\text{comparison}\rangle\ \langle\text{const}_4\rangle \\
\langle\text{cond}_{\text{eq}}\rangle &::= \langle\text{input}_0\rangle = \langle\text{const}_0\rangle \\
\langle\text{comparison}\rangle &::= < \mid > \\
\langle\text{const}_0\rangle &::= m_1 \mid m_2 \mid \ldots \mid m_{12} \\
\langle\text{const}_1\rangle &::= \text{integer value} \in [0, \max d_1] \\
\langle\text{const}_2\rangle &::= \text{integer value} \in [0, \max d_2] \\
\langle\text{const}_3\rangle &::= \text{integer value} \in [0, \max dd] \\
\langle\text{const}_4\rangle &::= \text{integer value} \in [0, N]
\end{aligned}
$$

In the grammar, the starting symbol ⟨bt⟩ expands into the ⟨if⟩ symbol, whose corresponding production rule is used to generate the internal nodes of the decision tree. These nodes define a condition, a true branch (evaluated if the condition is true), and a false branch (evaluated if the condition is false). The condition can be an inequality comparison (⟨cond⟩) or an equality comparison (⟨cond_{eq}⟩). The result of evaluating a logical statement (⟨a⟩) expands into either another internal node or a leaf node. Using this grammar, an individual genotype is translated into a phenotype decision tree which serves as a policy π for interacting with the training environment over e episodes. Fig. 6 provides an example of this genotype-to-phenotype translation.

In each episode, the policy is used to schedule N customer orders for CNC laser-cutting machines. Orders are processed one at a time, with each order represented as a five-dimensional feature vector. The features include:

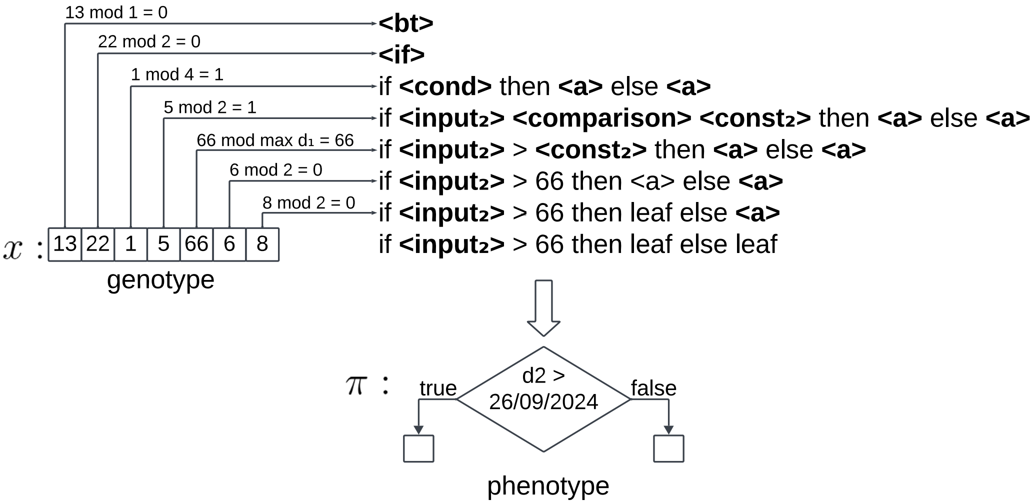


Fig. 6. Translation of an individual genotype to a phenotype decision tree using grammar $\mathcal{G}$. Note that with a starting date of July 22, 2024, the value 66 corresponds to September 26, 2024, i.e., 66 days later. Details about the procedure are provided in Custode and Iacca (2023).

- input_0, representing the machine type (m_t);
- input_1, representing the arrival date (d_1) of component c_1;
- input_2, representing the arrival date (d_2) of component c_2;
- input_3, representing the customer delivery date (dd);
- input_4, representing the number of remaining orders at the time of processing the current input.

Starting from the root node, the decision tree evaluates order features at internal nodes and traverses branches until reaching a leaf. The machine type feature (input_0) is used in equality comparisons, appearing on the right side of the equality production rule ($\langle \text{cond}_{\text{eq}} \rangle$), while the other features (const_1 to const_4) are used in inequality comparisons, appearing on the right side of the inequality production rule ($\langle \text{cond} \rangle$). Features input_1 through input_4 are compared to constants (const_1 to const_4) associated to their corresponding dataset's ranges. For instance, input_1 (d_1) is compared against values between 0 and the maximum d_1 in the dataset. This design choice simplifies the search space, hence facilitating algorithm convergence toward better solutions. At the last level of the decision tree, leaf nodes assign priority levels from 1 to 10 to input orders.

At the end of each episode, the policy π assigns a priority level to each of the N customer orders, reflecting their urgency in the production plan. These priority levels are converted into priority numbers by dividing the range $[0, N]$ into 10 equal intervals and randomly selecting a value from the corresponding interval for each level. This approach further reduces the search space and aids convergence. Using the assigned priority numbers, customer orders are sorted by priority to generate a permutation. However, this permutation does not include all design variables required to construct an HFS schedule. Prior studies (Urlings et al., 2010; Yu et al., 2018) highlight that optimizing all HFS design variables simultaneously results in an overly large solution space, making the search inefficient within feasible computational budgets. To address this, we adopt *list scheduling* (Oĝuz and Ercan, 2005; Rashidi et al., 2010; Pan et al., 2017) as a surrogate heuristic to derive the actual production plan from the priority-sorted order list. In this approach, assembly areas of the manufacturing floor are allocated to orders based on their priority as soon as the areas become available. When an area is freed and orders remain in the queue, the highest-priority order that has not yet entered production is assigned to the available space. Similarly, at the start of each production day, human resources are allocated to the CNC laser-cutting machines to produce according to their priority. The heuristic processes machines in descending priority order and checks whether required components (c_1 and/or c_2) are present or not needed, and whether sufficient human resources are available for the current production stage. If these conditions are met, resources are reserved; otherwise, the heuristic moves to the next job without delays. Since the objective is to optimize the makespan, all operations are initiated as early as possible.

The initial probabilities for a leaf node to assign different priority levels are randomized. As described in Section 5.1, these probabilities are refined iteratively based on feedback from the environment. The reward, derived from the makespan achieved at the end of a simulation, guides this adjustment. The simulation is executed using the DT developed in AnyLogic, which effectively models the relevant supply chain segment. This model captures key complexities and peculiarities of the system, including stochastic elements and uncertainties, hence providing an accurate representation of the real-world processes. For instance, transportation times for delivering completed CNC machines to logistics centers follow probability distributions, reflecting real-world variability and delays.

To prevent overfitting to input order sequences, the order list is randomly shuffled at the start of each simulated episode. This ensures the policy depends on actual order features rather than sequence-specific patterns.

6.3. Baseline (without human-driven refinement)

To assess the quality of IELDT, we compare it with a baseline IAI approach called ELDT. This latter shares the same optimization algorithm as IELDT, but it excludes user interaction during the optimization process. Instead, ELDT autonomously performs the two-level optimization scheme described in Section 5.1, refining an interpretable decision tree policy based solely on feedback from simulations with the DT. For a fair comparison, both methods use identical hyperparameter configurations, grammar definitions, and computational budgets.

6.4. Large language models

We conducted experiments to evaluate various prompt structures, iteratively refining them until achieving the final version shown in Fig. 3. The language models utilized in this study are:

- `gpt-4o` by OpenAI (OpenAI, 2024)
- `meta-llama-3.1-405B-instruct` by Meta (Meta, 2024)
- `mistral-large-2407` by Mistral AI (AI, 2024)
- `phi-3-medium-4k-instruct` by Microsoft (Face, 2024)

No fine-tuning or additional modifications were applied to the pre-trained models. Interactions with the models were conducted through their respective publicly available APIs.

6.5. Datasets

The specificity of the scenario under investigation precludes the availability of benchmark datasets. The proposed methodology was evaluated using four synthetically generated stochastic datasets (`d1`-`d4`) and validated on a real-world dataset (`d5`). The real-world dataset (`d5`) includes 345 customer orders for m_1 and m_5 laser-cutting machines from 2021 to 2024, provided by Adige Spa. These customer orders were made accessible to the DT through the SAP Manufacturing Integration and Intelligence (SAP MII) system (Company, 2024b), which provides an interface between SAP ERP and operational applications. For all experiments, the number of human resources was fixed as $r_1 = r_2 = r_3 = 5$. Dataset `d1` contains 100 customer orders with machine types (m_t) randomly assigned. The arrival dates (d_1, d_2) of components c_1 and c_2 are random values between 1 and 20 days from the simulation start. The delivery date (dd) is set to d_1 plus a random number of days between 20 and 50, reflecting the company's lead time. Dataset `d2` follows the same structure as `d1`, but is restricted to machines belonging to group_1. Dataset `d3` also mirrors `d1`, focusing exclusively on group_2 configurations. Dataset `d4` includes 100 orders with machine types (m_t) uniformly sampled from all available categories. The considered production period spans one year and is divided into five periods, each covering approximately 73 days. Orders are assigned to these periods randomly, with d_1 and d_2 set within the corresponding period's day range, and dd calculated as d_1 plus a random number of days between 20 and 50.

7. Results

Due to the stochastic nature of the proposed methodology, the reported results are based on experiments from 10 independent runs for both IELDT and ELDT.

The plots in Figs. 7 and 8 show the makespan (y-axis) as a result of the AnyLogic simulation, plotted against the number of evaluations (x-axis), which represents the number of executions of the DT simulation. Fig. 7 focuses on the stochastic datasets (`d1` to `d4`), while Fig. 8 reports the performance observed on the real-world dataset (`d5`). These results serve as a measure of production scheduling quality, with the policies generated by IELDT represented by the green curve and those from ELDT (the baseline without human-driven refinement) by the blue curve.

Fig. 9 displays the best decision tree policies (i.e., those that resulted in the lowest makespan in the DT simulation) obtained by the two algorithms across the 10 runs in the case of dataset `d5`. For the policy generated by IELDT (on the right), we provide the natural language explanations derived from the four LLMs, which are reported in Table 2.

The obtained results provide evidence of the effectiveness of the proposed IAI method in addressing the SCM problem under investigation. From prior research, currently under review, we have already established that ELDT achieves performance on par with, and in some cases better than, black-box optimizers such as Optuna (Akiba et al., 2019), a Genetic Algorithm (Forrest, 1996), Ant Colony Optimization (Dorigo et al., 2006), and PPO (Schulman et al., 2017) across various SCM tasks. These findings challenge the prevailing notion that there exists a trade-off between interpretability and performance, suggesting that interpretable decision-tree-based policies can deliver high performance while offering significant advantages in terms of understandability and trustworthiness (Zhou et al., 2024; Rudin, 2019).

When comparing ELDT with the IELDT variant introduced in this work, the plots in Fig. 7 and 8 show that IELDT performs comparably to ELDT across synthetic (Fig. 7) and real-world (Fig. 8) datasets. However, by incorporating human feedback into the optimization process, IELDT demonstrates faster convergence, achieving an up to 21% improvement (in terms of number of evaluations needed to achieve the same makespan, on average across 10 runs), as well as lower makespan in 3 out of 5 cases (including the real-world one).

To assess the statistical significance of such performance improvement, we conducted a statistical analysis applying the Wilcoxon test ($\alpha = 0.05$) on the number of evaluations required by IELDT, for each of the 10 runs on each dataset, to achieve the same solution quality achieved at convergence (identified as the earliest evaluation step at which its makespan reaches its lowest value and remains constant) by ELDT. This analysis reveals that the convergence improvement provided by IELDT is statistically significant ($\alpha < 0.05$) on two out of five datasets. While the remaining datasets do not meet the threshold for statistical significance, the p-values are still relatively small. For instance, on the real-world dataset `d5`, the p-value is 0.069.

Overall, these results underscore the importance of human-AI collaboration, allowing domain experts to actively (and effectively) guide the optimization process toward solutions that align more closely with their expertise, preferences, intuition, and practical considerations.

It is important to emphasize once more that the interpretable nature of the decision tree generated through our framework is key for enabling effective human-AI interaction. As illustrated in Fig. 9, our interpretable framework provides transparent and explainable policies, while demonstrating proficiency in addressing the company's production scheduling challenges. While black-box optimization processes could in principle incorporate HITL interventions, their opaque internal workings typically restrict user feedback on the final results and performance metrics, rather than the algorithmic logic that achieved them. In contrast, with IELDT, users can trace causal relationships and fully understand the model's decision-making logic and performance, which in turn fosters deeper insight and more meaningful collaboration, and simplifies deployment through reduced testing and easier debugging. Beyond prioritizing input orders, the decision tree can also enhance stakeholder understanding by potentially revealing hidden correlations and patterns, informing strategic decisions, and deepening insights into system dynamics. This approach encourages greater adoption of AI solutions by addressing the barriers posed by the opaque nature of conventional machine learning and optimization methods, which often hinder widespread implementation in industrial settings.

Finally, Table 2 highlights the ability of LLMs to generate clear, concise descriptions of the decision-making process, validating the prompt-generation strategy detailed in Section 5.2. The iterative dialogue facilitated by these models between AI systems and human experts fosters trust and engagement, refining policies to better align with human expertise and operational needs.

As a side note, AnyLogic has proven effective for developing the DT of the manufacturing floor and supply chain. Using the HFS framework, it enabled us to create detailed simulations of operations with a level of detail that would be impractical using purely analytical methods or traditional modeling approaches. Its intuitive graphical interface and 3D visualization capabilities empowered the company experts to actively participate in the design and analysis processes, mapping relevant supply chain details accurately. Additionally, visualizing the proposed production plans in a virtual environment allowed the stakeholders to develop an accurate intuition of their impact on the overall supply chain comprehensively.

8. Discussion

This study underscores the importance of aligning AI technologies with stakeholder needs, emphasizing transparency, usability, and human-AI collaboration. Our findings highlight in particular how IAI and simulation–optimization frameworks can bridge the gap between theoretical advancements in iDTs and their practical application in SCM (Hasija and Esper, 2022b). Building upon the results presented in the previous section, we now discuss the main implications of the introduced iDT framework in the context of SCM.

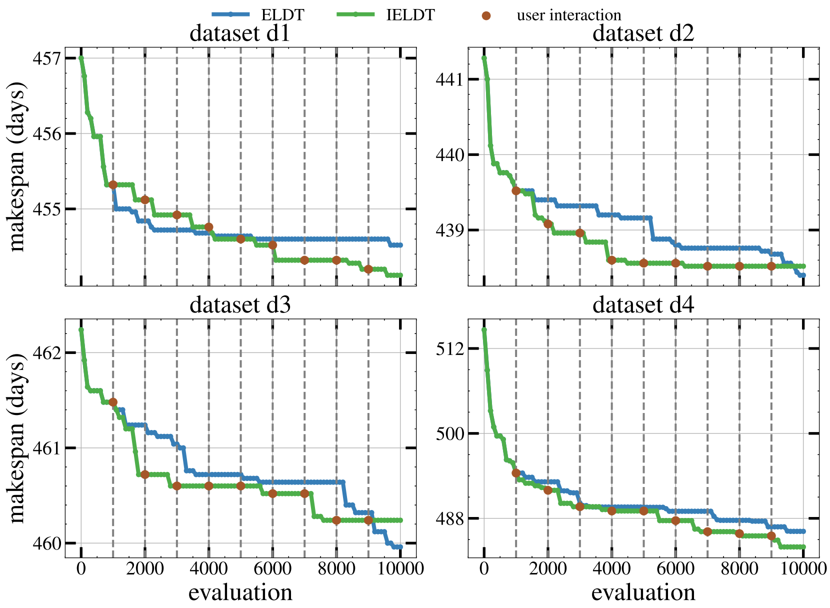


Fig. 7. Makespan (in days) (y-axis) versus simulation runs (x-axis) achieved by the best decision-making policy obtained using ELDT (blue curve) and IELDT (green curve) on the stochastic datasets `d1` to `d4`. Vertical dashed lines indicate user interactions during the optimization process, with green dots showing the makespan at the interaction points on the IELDT curve. The curves represent averages over 10 independent runs.

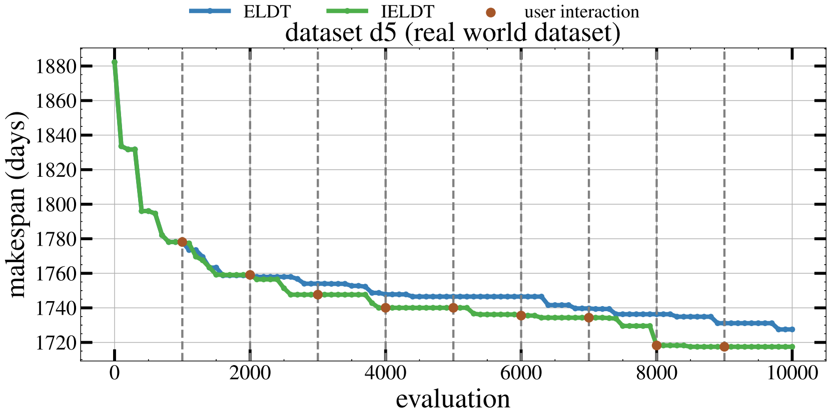


Fig. 8. Makespan (in days) (y-axis) versus simulation runs (x-axis) achieved by the best decision-making policy obtained using ELDT (blue curve) and IELDT (green curve) on the real-world dataset `d5`. Vertical dashed lines indicate user interactions during the optimization process, with green dots showing the makespan at the interaction points on the IELDT curve. The curves represent averages over 10 independent runs.

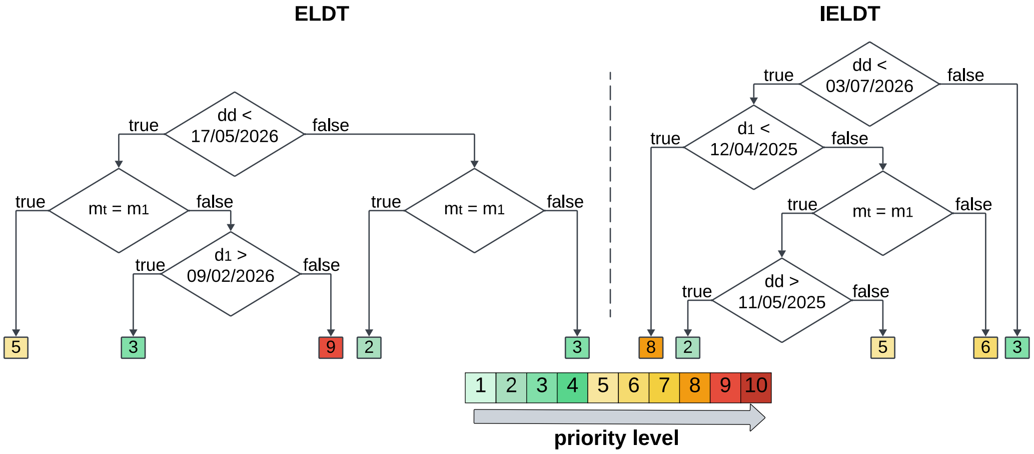


Fig. 9. Interpretable decision tree policies generated by `eldt` (left) and `ieldt` (right) on dataset `d5`. Diamonds represent internal nodes with logical conditions, and squares represent leaf nodes. Leaf node colors range from light (low order priority) to dark red (high order priority), reflecting scheduling prioritization in the simulation model.

Table 2
Descriptions generated by the four LLMs for the decision tree produced by IELDT shown in Fig. 9 (right).

gpt-4o. This decision tree evaluates the urgency of an order based on the delivery timeline and specific machine type. First, it checks how soon the delivery is expected. If the delivery is anticipated sooner, it then considers how early the component c_1 arrives. If c_1 arrives much earlier, the order is given a very high priority, suggesting the order should be completed quickly. If c_1 does not arrive that early, the type of machine is then assessed. For a certain machine type, the delivery timing is crucial: if it is relatively close, the order is considered less urgent, and if it is much earlier, it is given a medium priority. For other machine types, regardless of the delivery time, a slightly higher priority than medium is assigned. If the delivery is not expected soon, the order is considered not urgent and is assigned a lower priority.
meta-llama-3.1-405B-instruct. Looking at the priority assignment, it is clear that timing plays a significant role. Orders with an imminent delivery and early arrival date of component c_1 are considered highly urgent. However, if the arrival date of component c_1 is not as early, the type of machine comes into play. Machines of a certain type tend to have lower to moderate priority, especially if their delivery is not as rushed. On the other hand, machines of a different type generally have a moderate priority level. Orders with a more relaxed delivery timeline are assigned a lower priority.
mistral-large-2407. When planning the production, the date of delivery is the first factor we consider. If the delivery date is relatively close, we take a closer look at the date of arrival of c_1. If c_1 is expected to arrive soon, the priority is set high, as both the delivery and key component arrival are imminent. If the c_1 arrival is not as urgent, we then consider the machine type. For machine type m_1, we further evaluate the delivery date. If the delivery date is quite close, the priority is set low, as there is still some time. However, if the delivery date is more urgent, the priority is set to a medium level. For other machine types, the priority is set slightly higher than the medium level, as these types might have different urgency considerations compared to machine type A. If the delivery date is not as pressing, the priority is set low since there is ample time for production scheduling.
phi-3-medium-4k-instruct. For this decision tree, the order's priority level depends on a few key details: the delivery date, the component c_1 arrival date, and the machine type. If the order's delivery date was relatively recent, the tree then checks how long ago c_1 arrived. If this happened quite a while ago, the tree looks at the machine type to further inform the priority. When the machine type is m_1, it seems like this type of machine does not require immediate attention, provided the delivery date is quite some time in the past. However, if it is a fairly recent delivery date, then regardless of the machine type, the priority is usually higher. On the other hand, if the delivery date is older than a month or so, the priority tends to be lower, but it still depends a bit on the machine type and c_1 arrival date. Overall, the delivery date typically has a strong direct impact on determining the order's priority, but this may be influenced by the c_1 arrival date and the type of machine being used.

8.1. Performance implications

At this stage, quantitative indicators are still under development since more time is required to effectively assess the results and preliminary tests are currently underway at Adige Spa. However, it is possible to provide several qualitative insights and implications that clearly validate the quality of the proposed approach, while monitoring the current behavior of the system. The implementation of our iDT framework supports planners in production scheduling by enabling them to make informed decisions about the prioritization of orders. The recommendations generated by the framework take into account not only the current and past orders but also incorporate data about the forecasted demand provided by the marketing office, providing a comprehensive view of the production landscape.

By integrating forecasted order data into the proposed iDT, it may be possible to complement the framework proposed in Park et al. (2021). Indeed, our framework enables the simulation of mid-term supply chain behaviors, empowering planners to prioritize production while also having a long-term perspective, allowing for strategic decision-making. This is particularly valuable as it would allow for more proactive, rather than reactive, scheduling decisions. For instance, in addition to considering immediate customer demands, planners can use our framework to anticipate potential future demand fluctuations. This foresight can enhance the production scheduling process by enabling decision-makers to develop strategies based on projected supply and demand dynamics. By utilizing simulated scenarios, planners can assess the outcomes of different actions in various conditions, whether under high demand, low demand, or other supply chain disruptions.

As confirmed by the OD, in a best-worst case analysis, planners can use this framework to model and simulate supply chain behaviors under various demand scenarios. For example, if there is a sudden surge in demand, the iDT can help planners identify how to adjust production schedules and resources accordingly to meet these demands, while simultaneously identifying potential bottlenecks or resource shortages. Similarly, during periods of low demand, planners can adjust their operations to minimize idle resources, optimize capacity, and avoid overproduction. These capabilities enable the adjustment of production capacity to accommodate fluctuations, reducing the risk of both underproduction and overproduction, which could lead to costly inefficiencies (e.g., due to high storage costs) or disruptions.

Moreover, the insights generated by the proposed iDT framework can be shared with suppliers, fostering better coordination across the supply chain. By providing suppliers with access to estimated demand and expected delivery times, the framework can enable suppliers to plan their production and inventory strategies more effectively. This improved alignment between the company and its suppliers can help reduce lead times, minimize disruptions, and optimize inventory management. In particular, sharing forecasted demand data with suppliers can reduce the likelihood of stockouts or inventory surplus, contributing to smoother operations and better responsiveness to market changes. Furthermore, by promoting the exchange of AI-driven and DT-enhanced insights along the supply chain, our framework can facilitate the coordinated use of these technologies across organizational boundaries. This aligns with the principles highlighted in Klumpp and Ruiner (2022), advocating for the integration of AI and DT solutions to achieve holistic and synchronized SCM.

Additionally, this collaborative approach can play a pivotal role in the development of new supplier contracts or agreements. By incorporating the data and insights provided by the iDT into contract terms, both the company and its suppliers can gain greater flexibility in production scheduling. This flexibility facilitates better alignment between the production capacities and the timelines between the two parties, ensuring a more resilient supply chain. Moreover, this collaboration can also help avoid the financial consequences of underproduction or overproduction.

8.2. Performance in case of rescheduling

As seen earlier, our proposed interactive, simulation-based optimization framework, IELDT, produces interpretable scheduling policies in the form of decision trees. The policy optimization is performed directly on the actual set of customer orders available at execution time, combined with the real-time state of the production floor and supplier ecosystem. By initializing the DT directly from the real system, we avoid the manual parameter setup typically required in traditional simulation-based approaches (Wooley et al., 2023). Furthermore, the resulting policy is robust for scheduling the set of orders, even under minor deviations. Nonetheless, real-world execution may diverge from the initially expected conditions during order processing.

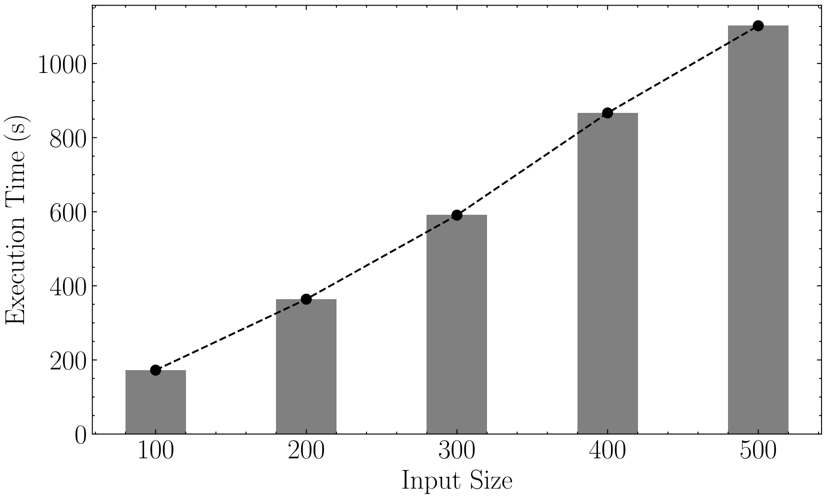


Fig. 10. Computation time (in seconds) for rescheduling as a function of the number of customer orders. Reported values exclude user interaction time to focus on the computational effort of the optimization process.

To address this, our HITL framework can be rerun on the updated system state, which essentially includes the current list of customer orders. This rescheduling operation can be triggered either periodically (e.g., every one to two weeks) or in response to major disruptions that affect production plans. This generates a new policy tailored to the current situation, helping maintain reliability over time. This approach is particularly suitable for the company under study, which manufactures CNC laser cutting machines with relatively long lead times, making online policy adjustments both feasible and effective. We note that in environments with very short production cycles, where rescheduling may be needed on an hourly basis, seamless application of our framework would be more challenging, though this limitation does not affect our use case.

To provide practitioners with practical evidence, we evaluated the computational effort required for rescheduling. Testing rescheduling with real-world production data was not feasible within the scope of this study, since the company has only recently begun systematic data collection, and its historical records are limited, heterogeneous, and insufficient for meaningful analysis. We therefore conducted synthetic experiments to assess scalability. Fig. 10 reports the computation time required by our approach as a function of the number of orders to be scheduled (ranging from 100 to 1000). These experiments reflect a realistic rescheduling setting, where the decision-maker optimizes the policy on the current list of active orders, showing how the required time scales with input size. For clarity, the reported results exclude user interaction time, which is highly variable and depends on the individual decision-maker; including it would not allow for a meaningful or generalizable evaluation. The plot shows that the execution time of the rescheduling process scales linearly with the number of orders. Even for long input lists of up to 500 customer orders, the computation time remains within practical bounds for real-world use. In the case of Adige Spa, where the number of orders rarely exceeds 400, the approach proves well-suited to operational needs.

8.3. Reflections on implementation and action research

In collaboration with Adige Spa, a reflection on the overall action research process was carried out, considering both operational and managerial perspectives. Three key factors influencing the process were identified. First, the strong push by the OD toward implementing new digital technologies played a crucial role. This initiative not only provided the necessary momentum for the project but also facilitated the involvement of a diverse range of stakeholders within the company, from operational staff to senior managers. This multi-level engagement ensured a comprehensive understanding of the challenges and opportunities associated with the proposed iDT framework and enabled access to information and data collection. Second, the need to adapt the company's existing technological architecture and data management processes emerged as a significant factor. The information-intensive requirements of the iDT framework necessitated the development of tailored solutions to address data integration and quality. This adaptation process was instrumental in aligning the iDT framework with the company's digital ecosystem. Third, the emphasis on fostering human-AI collaboration underscored the importance of creating an interpretable and user-friendly system. By incorporating HITL principles, the framework not only improved decision-making processes but also facilitated trust and adoption among staff. This focus on interpretability ensured that the recommendations provided by the iDT were accessible and actionable, bridging the gap between advanced AI capabilities and practical usability.

The implementation into the production environment of the iDT is still in progress; we are currently planning the next phases of the process and reflecting on the potential integration of further elements into the framework.

Fig. 11 summarizes the IELDT framework's key impacts on the organization at supply chain, strategic, and operational levels.

8.4. Reflections on explainability and interpretability

The distinction between interpretability and explainability is quite subtle, and there is still little consensus on it. Following definitions from (Gunning and Aha, 2019) and Barredo Arrieta et al. (2020), we adopt the view that explainability refers to post-hoc techniques that clarify a model's decisions, without necessarily explaining the inner workings of the model's internal logic. Methods like SHAP (M. and Su-In, 2017), LIME (Ribeiro et al., 2016), or saliency maps (Simonyan et al., 2013), offer this kind of explanation by highlighting features that contribute to a model's outcome. For example, recent work (Fischer et al., 2024) applied XAI methods, namely DeepSHAP (Avanti Shrikumar and Kundaje, 2017) and Input x Gradient (Kokhlikyan et al., 2020), to explain the decisions of deep RL agents in real-world production contexts.

Interpretability, on the other hand, is a structural property of models whose internal decision-making process can be directly understood and replicated by humans. For example, decision trees or rule-based systems are typically more interpretable because one can more easily follow and understand their decision-making steps compared to

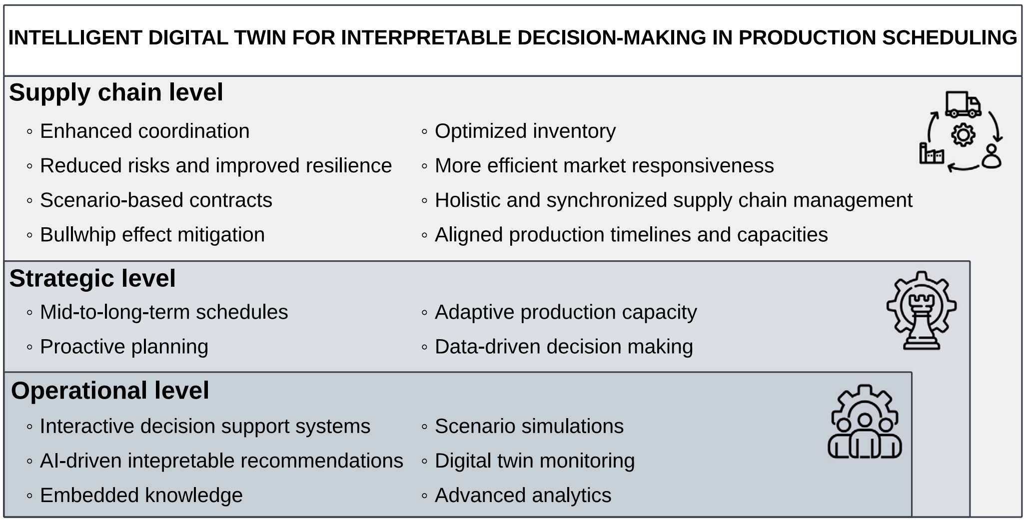


Fig. 11. Overview of the IELDT framework's impact on the organization across the supply chain, strategic, and operational levels.

complex models like neural networks. Notably, these two properties are not inherently in conflict. While explainability does not necessarily imply interpretability, an interpretable model is also inherently explainable. Our approach, for instance, offers both: the core decision-making process is represented through an interpretable decision tree, and we provide post-hoc explainability through natural language translations of this process. In this framework, the decision tree is the central component, while the LLM acts as an interface to present the decision-making process to human users.

The key contribution of our work lies in the integration of evolutionary search, decision trees, RL, and LLMs to create an end-to-end pipeline for generating and explaining schedules. To clarify the scope of our contributions, we highlight three key levels of explainability in our framework:

1. how the scheduling policy is generated;
2. how the policy produces a schedule;
3. how the decision process can be communicated in natural language.

This distinction is possible because our approach goes beyond traditional scheduling methods, such as optimization techniques based on exact methods or metaheuristics, which directly generate a schedule (typically in the form of a table), offering limited insights into the underlying decision logic. In contrast, our method learns an explicit scheduling policy: an interpretable decision tree optimized via evolutionary search and RL, which not only generates the schedule (point 2) but can also be translated into understandable explanations via the LLM (point 3). On the other hand, we acknowledge that our work does not aim to explain the evolutionary process itself (point 1). Indeed, the interpretability of evolutionary algorithms, particularly understanding the behavior of genetic operators like crossover and mutation, remains an open research challenge. Recent work (Zhou et al., 2024) has begun to explore this intersection between Evolutionary Computation and XAI, highlighting the complexity and long-term nature of the problem. While we recognize its importance, such an investigation lies beyond the scope of our current contribution.

8.5. The impact of large language models descriptions

At the beginning of the second iteration of our action research intervention with Adige Spa (see Fig. 1), the initial version of our simulation-based interpretable AI solution produced only a final scheduling policy represented as a decision tree. At this stage, the system lacked interactive components and explanatory features. During early interactions with company stakeholders from several divisions including production planners and supplier managers, we observed that they spontaneously attempted to interpret the logic of the tree. For some branches, decisions were perceived as intuitive and aligned with domain knowledge. For others, however, stakeholders struggled to justify the model's outputs or reached divergent interpretations. In essence, while the decision tree already provided an interpretable structure, its accessibility was limited by technical notation, the cognitive effort required to follow multiple branches, and the absence of contextual explanations. These observations were reflected in stakeholder feedback, for example:

> *"It's transparent in theory, but I still have to stop and translate the abbreviations. It slows me down."*

> *"I can see the rules, but some branches are hard to follow. I'm not sure why the model is making that choice here."*

These empirical observations led the research team to integrate key components of IELDT with web-based visualizations and interactive features, moving toward an HITL optimization approach. In this direction, the integration of the LLM-based explanations proved particularly valuable in reducing usage barriers. The enhanced interface undoubtedly made the system more user-friendly compared to the raw decision tree alone. The LLM generated natural-language descriptions of the decision logic, translating technical tree structures and abbreviations (e.g., "dd" into "delivery date") into familiar terms for practitioners. Because IELDT is an interactive optimization framework, relying solely on the decision tree structure during the evolutionary process proved less engaging and less informative than a richer interface. By combining visualization with textual explanations, the system offered a more user-friendly and appealing interaction, supporting more effective HITL decision-making. Of course, the LLM's role was not to influence nor alter the optimization process, but rather to act as an explanatory layer that bridges the gap between technical model outputs and human reasoning. The interpretable decision tree scheduling policy and the interactive simulation-based optimization supported by DT capabilities remain the core of the framework, while the LLM improves accessibility and usability.

To assess the impact the added LLM-generated explanations, we conducted a qualitative evaluation with five production experts. In individual meetings, all participants consistently reported that the LLM-generated explanations enabled them to understand the model's behavior more easily and quickly than when only the decision tree was presented. Below, we report some representative quotes that illustrate these findings.

> "*The text explanations helped me immediately understand what the tree was doing. I don't need to decode the abbreviations anymore.*"

> "*Before, the tree looked complicated. Now the explanations make it feel much easier to use.*"

> "*Now I can follow the reasoning without guessing. The descriptions feel closer to how we actually talk about scheduling.*"

This feedback confirms that the LLM component improved human interaction with the system, addressing a real industrial challenge while contributing to the broader understanding of LLMs' role in production planning. Ultimately, this innovative integration of generative AI within an optimization framework further highlights how transparency, usability, and human-centered design are critical for adoption in I4.0 contexts. In this context, our case study provides a valuable reference for both researchers and practitioners seeking to enhance the accessibility of AI systems in industrial environments.

8.6. *Theoretical implications*

This study advances the theoretical understanding of iDTs by framing them as HITL decision-support systems (Hines et al., 2025) that integrate IAI methods and LLMs to address the dual challenges of uncertainty and interpretability in SCM (Baryannis et al., 2019; Sadeghi R. et al., 2024). Unlike prior DT implementations that often rely on opaque AI models, our framework emphasizes transparency, enabling stakeholders to understand, trust, and act upon system recommendations. By embedding IAI directly into the decision-making loop rather than relying solely on post-hoc explainability methods (Kosasih et al., 2024), this study provides a pathway for developing DTs that are both intelligent and interpretable, fostering adoption and alignment with managerial decision-making.

The integration of LLMs enhances the usability of iDTs by generating user-friendly explanations for complex optimization outputs, bridging the gap between sophisticated AI reasoning and human decision-makers (Zheng et al., 2022). The incorporation of human feedback within optimization, as demonstrated in IELDT, underscores the critical role of HITL collaboration in guiding AI-driven processes toward solutions that reflect domain expertise, practical considerations, and organizational objectives.

Furthermore, this study challenges the conventional trade-off between interpretability and performance by showing that transparent, HITL-enabled iDTs can deliver high-quality optimization outcomes, extending prior theoretical discussions on AI adoption in SCM (Klumpp and Ruiner, 2022). Also, the real-world implementation of the framework in a global tool machine manufacturer (Del Gallo et al., 2023; Coughlan and Coghlan, 2002; Bagni et al., 2025) demonstrates the applicability of these theoretical insights, illustrating how iDTs can operationalize human-centric decision-making in complex, uncertain production environments.

Collectively, these contributions extend the conceptualization of DTs beyond purely technological artifacts, positioning them as organizational information processing instruments that support trust, adoption, and effective managerial decision-making. By embedding HITL mechanisms, integrating IAI and LLMs, and validating the framework in practice, this research provides a foundation for transitioning from I4.0 toward a more human-centric Industry 5.0.

8.7. *Managerial implications*

From a managerial perspective, the developed framework provides actionable guidance for production scheduling in industries characterized by high outsourcing, variability, and uncertainty. By integrating simulation capabilities with IAI methods within an iDT, the framework enables planners to evaluate multiple production scenarios under stochastic conditions, thereby reducing risks associated with delays in component deliveries or sudden demand fluctuations (Hasija and Esper, 2022a; Zheng et al., 2022). Importantly, the system delivers interpretable, actionable recommendations, helping overcome organizational resistance to AI and DT adoption by fostering trust and confidence in decision-making outputs.

The framework also encodes planners' expertise into a digital format, preserving institutional knowledge and ensuring continuity in decision-making even in the face of workforce turnover. This HITL approach enhances operational resilience, allowing organizations to maintain robust production planning capabilities despite personnel changes.

Interactive capabilities, enabled by LLMs, further enhance usability by generating natural language explanations of the decision-making process. This functionality broadens the applicability of the framework across diverse organizational roles, including non-technical stakeholders, supporting engagement, comprehension, and trust in AI-driven recommendations. The simulation environment, implemented in AnyLogic, creates a comprehensive DT that accurately reflects supply chain dynamics, allowing managers to visualize and intuitively understand the impact of proposed production plans.

Finally, by providing a real-world implementation of an AI-enhanced iDT for production scheduling, this study addresses a gap in the literature, where practical case studies remain scarce (Del Gallo et al., 2023). The framework not only demonstrates the tangible operational benefits of combining simulation, AI, and HITL interaction but also serves as a potential roadmap for other companies considering similar technological adoption. In doing so, it lowers perceived barriers, supports evidence-based decision-making, and encourages the diffusion of human-centric, iDT technologies across industrial contexts, contributing to the broader transition toward Industry 5.0.

9. Conclusions and future research directions

DTs have emerged as a key player within the Industry 4.0 and Industry 5.0 realms, evolving from their initial role as mirroring simulations of physical systems to sophisticated tools for advanced data analytics and data-driven decision-making. Integrating AI into DT frameworks offers great potential for optimizing complex processes, yet the "black-box" nature of most AI algorithms has, insofar, limited the adoption of these methods in industrial systems, due to concerns about trustworthiness and maintainability.

Motivated by this limited adoption of AI in this space, in this work we proposed an interpretable simulation–optimization DT framework, called IELDT. Under the action research paradigm, we applied IELDT to a real-world supply chain scheduling problem for a global leader in CNC laser-cutting machine production. While action research projects are inherently context-specific and are not designed to produce universal knowledge (Coughlan and Coghlan, 2002), the findings from this study provide valuable contributions to scientific research and demonstrate the broader potential of combining DTs with IAI in similar contexts.

The DT, developed using AnyLogic, captures the complexities of the target supply chain, including the uncertainties and stochastic behaviors inherent in industrial operations. Central to the framework is a simulation-driven optimization process that combines evolutionary computation with RL to iteratively refine decision-making policies represented as interpretable decision trees. The optimization process leverages the DT's capabilities to simulate scenarios based on customer order data. This enables the IAI algorithm to progressively refine decision tree policies by evaluating their performance against predefined metrics. To further enhance user comprehension and engagement, the framework employs prompt engineering templates from recent literature to query LLMs through APIs to generate intuitive natural language explanations of the decision tree policies. A key contribution of this research is the HITL approach, which empowers users to assess the decision trees not only based on their performance but also on their internal workings (an advantage not available with traditional black-box optimizers) and alignment with domain expertise. Through the user interface, stakeholders provide feedback by voting on decision trees that best reflect their knowledge and preferences. This judgment is then incorporated into subsequent optimization iterations, creating an iterative simulation-optimizer-human loop that progressively enhances both the solution quality and its alignment with human intuition.

The proposed proof-of-concept was experimentally validated on both synthetic datasets and real-world data provided by Adige Spa. Notably, results demonstrated that the integration of human feedback accelerates, in most cases, convergence toward better solutions compared to an optimizer operating without human intervention. This underscores the significant benefits of combining human expertise with AI-driven optimization, particularly in addressing uncertainty, managing complexity, and fostering stakeholder trust. Our findings highlight the potential of IAI-driven iDTs in enabling human-centric, explainable AI applications in SCM. Ultimately, this research underscores the importance of further exploring IAI methodologies in the context of DTs to enhance data-driven decision-making, improve user trust, and establish robust, collaborative solutions for complex industrial challenges.

Limitations and future works. One of the main limitations of the current work stems from the use of an evolutionary computation method: meta-heuristics, including evolutionary algorithms and swarm intelligence methods, are known to typically demand substantial computational resources and time to find optimal solutions. This is particularly true when the decision space is large, as the algorithms' convergence can be slow. Additionally, the inherent randomness of evolutionary algorithms can introduce instability in the results, especially under different initialization conditions, leading to significant variation in the generated decision trees. This variability may raise concerns about the feasibility of applying these methods in real industrial scenarios, where rapid decision-making and stable outputs are often critical. However, we believe that incorporating human feedback could be a valuable strategy to mitigate these issues: by aligning the output of these algorithms with human preferences and domain knowledge, we can help stabilize the results and ensure the decision trees reflect the intended goals. In this regard, it is worth noticing that, while interactive evolutionary computation (IEC) (Wang and Pei, 2024) has been widely applied to creative tasks such as product, art, and sound generation, to our knowledge, none have combined IEC with DT simulations for enhanced decision-making, where alignment with human preferences can in fact lead to stabler results. On the other hand, it is also important to note that, in the context of the manufacturing company featured in our case study, convergence time is less of an issue due to the long lead times associated with CNC laser cutting machines, which reduces the need for rapid decision-making.

Furthermore, the current framework is limited by its basic user interaction features and shallow analytics for evaluating interpretable policies in DT simulations. Future enhancements should focus on providing richer performance insights and enabling users to interact more deeply with decision trees by modifying their values (e.g., priority levels on leaves or thresholds in conditions) and structure (e.g., by removing, adding, or modifying nodes or whole branches). This would require a form of Lamarckian evolution (Whitley et al., 1994) to map phenotype changes back to the genotype space. On the same aspect, another promising direction would be to refine the system to gather feedback from multiple users via independent web page access. Integrating insights from various domain experts into the decision model could, in fact, enhance its effectiveness and robustness (a sort of "wisdom of the crowd").

The current LLM prompt-engineering strategy uses fixed templates that provide context, highlight key features, and offer decision tree explanation examples. A valuable enhancement could allow users to input their own examples of decision tree explanations, reducing configuration parameters, improving user engagement, and aligning explanations more closely with user preferences for potentially better interpretability. To further improve this approach, it will also be important to identify which LLMs (among the four used in this paper, and possibly others) perform best in generating natural language descriptions of decision tree policies. This requires collecting a dataset of decision tree-explanation pairs and evaluating the models using recent metrics for explanation quality (Nauta et al., 2023). Selecting the best-performing LLM would simplify the interface by presenting users with a single, high-quality description.

Finally, future efforts will focus on expanding the framework's application to other areas within the company, evaluating its performance across various contexts. Given the methodology's inherent flexibility, we expect its successful deployment in areas beyond production scheduling.

CRediT authorship contribution statement

Stefano Genetti: Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Giorgio Scarton:** Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Marco Formentini:** Writing – review & editing, Supervision, Conceptualization. **Giovanni Iacca:** Writing – review & editing, Supervision, Resources, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used OpenAI ChatGPT and Grammarly in order to automatically identify and correct grammatical errors, refine syntactically incorrect sentences, and enhance the overall quality of the writing. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the content of the published article.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. We thank Alberto Longobardi, Operation Director at Adige Spa (BLM Group) and Adige Sys (BLM Group), and Annamaria Croci, Chief Information Officer at BLM Group. Their support in defining the company's needs and structure, organizing insightful feedback sessions with key stakeholders, and providing essential data has been instrumental in ensuring this research delivers meaningful contributions.

Data availability

The data that has been used is confidential.

References

AI, M., 2024. Large Enough. URL https://mistral.ai/news/mistral-large-2407/.

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2623–2631. http://dx.doi.org/10.1145/3292500.3330701.

Avanti Shrikumar, P.G., Kundaje, A., 2017. Learning important features through propagating activation differences. In: International Conference on Machine Learning. pp. 3145–3153, URL https://proceedings.mlr.press/v70/shrikumar17a.

Bagni, G., Filho, M.G., Finne, M., Thürer, M., 2025. Design science research in operations management: is there a single type? Prod. Plan. Control 36 (6), 789–807. http://dx.doi.org/10.1080/09537287.2024.2310230, arXiv:https://doi.org/10.1080/09537287.2024.2310230.

Bakopoulos, E., Siatras, V., Mavrothalassitis, P., Nikolakis, N., Alexopoulos, K., 2024. Digital-twin-enabled framework for training and deploying AI agents for production scheduling. In: Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through AI. Springer Nature, pp. 147–179. http://dx.doi.org/10.1007/978-3-031-46452-2_9.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 58, 82–115. http://dx.doi.org/10.1016/j.inffus.2019.12.012.

Baryannis, G., Dani, S., Antoniou, G., 2019. Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. Future Gener. Comput. Syst. 101, 993–1004. http://dx.doi.org/10.1016/j.future.2019.07.059.

Berti, N., Finco, S., 2022. Digital Twin and Human Factors in Manufacturing and Logistics Systems: State of the Art and Future Research Directions. IFAC-PapersOnLine 55 (10), 1893–1898. http://dx.doi.org/10.1016/j.ifacol.2022.09.675, URL https://linkinghub.elsevier.com/retrieve/pii/S2405896322019930.

Bhamra, R., Hicks, C., Small, A., García-Villarreal, E., 2022. Value, product delivery strategies and operational performance in the medical technology industry. Int. J. Prod. Econ. 245, http://dx.doi.org/10.1016/j.ijpe.2021.108399.

Bhandal, R., Meriton, R., Kavanagh, R.E., Brown, A., 2022. The application of digital twin technology in operations and supply chain management: a bibliometric review. Supply Chain Manag.: Int. J. 27 (2), 182–206. http://dx.doi.org/10.1108/SCM-01-2021-0053.

Bhattacharya, P., Chakraborty, M., Palepu, K.N., Pandey, V., Dindorkar, I., Rajpurohit, R., Gupta, R., 2023. Exploring Large Language Models for Code Explanation. arXiv preprint arXiv:2310.16673. URL http://arxiv.org/abs/2310.16673.

Biesinger, F., Meike, D., Kraß, B., Weyrich, M., 2019. A digital twin for production planning based on cyber-physical systems: A case study for a cyber-physical system-based creation of a Digital twin. Procedia CIRP 79, 355–360. http://dx.doi.org/10.1016/j.procir.2019.02.087.

Breque, M., De Nul, L., Petridis, A., 2021. Industry 5.0 - Towards a Sustainable, Human-Centric and Resilient European Industry. Publications Office of the European Union, http://dx.doi.org/10.2777/308407.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W., 2016. Openai gym. http://dx.doi.org/10.48550/arXiv.1606.01540, arXiv preprint arXiv:1606.01540.

Brown, T.B., 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165. URL http://arxiv.org/abs/2005.14165.

Brusset, X., Kinra, A., Drake, M., 2025. Theorising, Designing, and Implementing Digital Twins in Operations and Supply Chains. http://dx.doi.org/10.2139/ssrn.5354298, Available at SSRN (July 16, 2025).

Brusset, X., Torre, D.L., Broekaert, J., 2022. Algorithms, Analytics, and Artificial Intelligence. In: The Digital Supply Chain. Elsevier, pp. 93–110. http://dx.doi.org/10.1016/B978-0-323-91614-1.00006-X.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., et al., 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712. URL http://arxiv.org/abs/2303.12712.

Carter, A., Imtiaz, S., Naterer, G., 2023. Review of interpretable machine learning for process industries. Process. Saf. Environ. Prot. 170, 647–659. http://dx.doi.org/10.1016/j.psep.2022.12.018.

Ciano, M.P., Pozzi, R., Rossi, T., Strozzi, F., 2021. Digital twin-enabled smart industrial systems: a bibliometric review. Int. J. Comput. Integr. Manuf. 34 (7–8), 690–708. http://dx.doi.org/10.1080/0951192X.2020.1852600.

Commission, E., 2024. Artificial Intelligence Act: Proposal for a regulation of the European Parliament and of the Council laying down harmonized rules on Artificial Intelligence. Off. J. Eur. Union URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206.

Company, T.A., 2024a. AnyLogic: Simulation modeling software tools & solutions for business. URL https://www.anylogic.com/.

Company, T.S., 2024b. Sap. URL https://www.sap.com/.

Coughlan, P., Coghlan, D., 2002. Action research for operations management. Int. J. Oper. Prod. Manage. 22 (2), 220–240. http://dx.doi.org/10.1108/01443570210417515.

Custode, L.L., Iacca, G., 2023. Evolutionary learning of interpretable decision trees. IEEE Access 11, 6169–6184. http://dx.doi.org/10.1109/ACCESS.2023.3236260.

Danese, P., Romano, P., Formentini, M., 2013. The impact of supply chain integration on responsiveness: The moderating effect of using an international supplier network. Transp. Res. Part E: Logist. Transp. Rev. 49 (1), 125–140, URL https://linkinghub.elsevier.com/retrieve/pii/S1366554512000774.

Del Gallo, M., Mazzuto, G., Ciarapica, F.E., Bevilacqua, M., 2023. Artificial Intelligence to solve production scheduling problems in real industrial settings: Systematic literature review. Electronics 12 (23), http://dx.doi.org/10.3390/electronics12234732.

Dhebar, Y., Deb, K., Nageshrao, S., Zhu, L., Filev, D., 2020. Interpretable-AI policies using evolutionary nonlinear decision trees for discrete action systems. arXiv preprint arXiv:2009.09521. URL http://arxiv.org/abs/2009.09521.

Ding, Y., 2019. Brief analysis about digital twin supply chain model and application. Ind. Eng. Innov. Manag. 2 (1), 14–23. http://dx.doi.org/10.23977/ieim.2019.21003.

Ding, K., Chan, F.T., Zhang, X., Zhou, G., Zhang, F., 2019. Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors. Int. J. Prod. Res. 57 (20), 6315–6334. http://dx.doi.org/10.1080/00207543.2019.1566661.

Dorigo, M., Birattari, M., Stutzle, T., 2006. Ant colony optimization. IEEE Comput. Intell. Mag. 1 (4), 28–39. http://dx.doi.org/10.1109/MCI.2006.329691.

Ecandell, M., 2024. An open source library for connecting AnyLogic models with Python-based black box optimization packages. URL https://github.com/MarcEscandell/ALPypeOpt.git.

Elbasheer, M., Longo, F., Nicoletti, L., Padovano, A., Solina, V., Vetrano, M., 2022. Applications of ML/AI for decision-intensive tasks in production planning and control. Procedia Comput. Sci. 200, 1903–1912. http://dx.doi.org/10.1016/j.procs.2022.01.391.

Face, H., 2024. microsoft/Phi-3-medium-4k-instruct. URL https://huggingface.co/microsoft/Phi-3-medium-4k-instruct.

Fischer, D., Hüsener, H.M., Grumbach, F., Vollenkemper, L., Müller, A., Reusch, P., 2024. Demystifying reinforcement learning in production scheduling via explainable AI. arXiv preprint arXiv:2408.09841. URL http://arxiv.org/abs/2408.09841.

Forrest, S., 1996. Genetic algorithms. ACM Comput. Surv. 28 (1), 77–80. http://dx.doi.org/10.1145/234313.234350.

Gockel, B., Tudor, A., Brandyberry, M., Penmetsa, R., Tuegel, E., 2012. Challenges with structural life forecasting using realistic mission profiles. In: AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. American Institute of Aeronautics and Astronautics, pp. 1–11. http://dx.doi.org/10.2514/6.2012-1813.

Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.R., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: Annals of Discrete Mathematics, vol. 5, Elsevier, pp. 287–326. http://dx.doi.org/10.1016/S0167-5060(08)70356-X.

Grieves, M., Vickers, J., 2017. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: Transdisciplinary Perspectives on Complex Systems. Springer, pp. 85–113. http://dx.doi.org/10.1007/978-3-319-38756-7_4.

Gunning, D., Aha, D.W., 2019. DARPA's Explainable Artificial Intelligence Program. AI Mag. 40 (2), 44–58. http://dx.doi.org/10.1609/aimag.v40i2.2850.

Gupta, J.N., 1988. Two-stage, hybrid flowshop scheduling problem. J. Oper. Res. Soc. 39 (4), 359–364. http://dx.doi.org/10.2307/2582115.

Hasija, A., Esper, T.L., 2022a. In artificial intelligence (AI) we trust: A qualitative investigation of AI technology acceptance. J. Bus. Logist. 43 (3), 388–412. http://dx.doi.org/10.1111/jbl.12301.

Hasija, A., Esper, T.L., 2022b. In artificial intelligence (AI) we trust: A qualitative investigation of AI technology acceptance. J. Bus. Logist. 43 (3), 388–412. http://dx.doi.org/10.1111/jbl.12301.

Heydarbakian, S., Spehri, M., 2022. Interpretable Machine Learning to Improve Supply Chain Resilience, An Industry 4.0 Recipe. IFAC-PapersOnLine 55 (10), 2834–2839. http://dx.doi.org/10.1016/j.ifacol.2022.10.160.

Hines, P., Magnani, F., Mula, J., Sanchis, R., 2025. Toward lean industry 5.0: a human-centered model for integrating lean and industry 4.0 in an automotive supplier. Int. J. Prod. Res. 1–24. http://dx.doi.org/10.1080/00207543.2025.2551238.

Ivanov, D., 2023. Intelligent digital twin (iDT) for supply chain stress-testing, resilience, and viability. Int. J. Prod. Econ. 263, http://dx.doi.org/10.1016/j.ijpe.2023.108938.

Ivanov, D., 2024. Conceptualisation of a 7-element digital twin framework in supply chain and operations management, volume=62. Int. J. Prod. Res. 6, 2220–2232. http://dx.doi.org/10.1080/00207543.2023.2217291, URL https://www.tandfonline.com/doi/full/10.1080/00207543.2023.2217291.

Ivanov, D., Dolgui, A., Sokolov, B., 2022. Cloud supply chain: Integrating Industry 4.0 and digital platforms in the "Supply Chain-as-a-Service". Transp. Res. Part E: Logist. Transp. Rev. 160, http://dx.doi.org/10.1016/j.tre.2022.102676.

Kinra, A., Hald, K.S., Mukkamala, R.R., Vatrapu, R., 2020. An unstructured big data approach for country logistics performance assessment in global supply chains. Int. J. Oper. Prod. Manage. 40 (4), 439–458. http://dx.doi.org/10.1108/IJOPM-07-2019-0544.

Klumpp, M., Ruiner, C., 2022. Artificial intelligence, robotics, and logistics employment: The human factor in digital logistics. J. Bus. Logist. 43 (3), 297–301. http://dx.doi.org/10.1111/jbl.12314.

Knapp, G., Mukherjee, T., Zuback, J., Wei, H., Palmer, T., De, A., DebRoy, T., 2017. Building blocks for a digital twin of additive manufacturing. Acta Mater. 135, 390–399. http://dx.doi.org/10.1016/j.actamat.2017.06.039.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., Reblitz-Richardson, O., 2020. Captum: A unified and generic model interpretability library for pytorch. arXiv preprint arXiv:2009.07896. URL https://arxiv.org/abs/2009.07896.

Kosasih, E.E., Papadakis, E., Baryannis, G., Brintrup, A., 2024. A review of explainable artificial intelligence in supply chain management using neurosymbolic approaches. Int. J. Prod. Res. 62 (4), 1510–1540. http://dx.doi.org/10.1080/00207543.2023.2281663.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., 2018. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 51 (11), 1016–1022. http://dx.doi.org/10.1016/j.ifacol.2018.08.474.

Lam, W.S., Lam, W.H., Lee, P.F., 2023. A bibliometric analysis of digital twin in the supply chain. Mathematics 11 (15), 3350. http://dx.doi.org/10.3390/math11153350.

Lee, D., Lee, S., 2021. Digital twin for supply chain coordination in modular construction. Appl. Sci. 11 (13), 5909. http://dx.doi.org/10.3390/app11135909.

Levin, M., 2003. Action research and the research community. Concepts Transform. 8 (3), 275–280. http://dx.doi.org/10.1075/cat.8.3.08lev.

Li, X., Chehade, H., Yalaoui, F., Amodeo, L., 2011. A new method coupling simulation and a hybrid metaheuristic to solve a multiobjective hybrid flowshop scheduling problem. In: Conference of the European Society for Fuzzy Logic and Technology. pp. 1082–1089. http://dx.doi.org/10.2991/eusflat.2011.33.

Longo, L., Goebel, R., Lecue, F., Kieseberg, P., Holzinger, A., 2020. Explainable Artificial Intelligence: Concepts, applications, research challenges and visions. In: Machine Learning and Knowledge Extraction, vol. 12279, Springer International Publishing, pp. 1–16. http://dx.doi.org/10.1007/978-3-030-57321-8_1.

M., L.S., Su-In, L., 2017. A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, vol. 30, pp. 4768–4777, URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

McCracken, D.D., Reilly, E.D., 2003. Backus-Naur Form (BNF). In: Encyclopedia of Computer Science. John Wiley and Sons Ltd., pp. 129–131, URL https://dl.acm.org/doi/abs/10.5555/1074100.1074155.

Meta, 2024. Introducing Llama 3.1: Our most capable models to date. URL https://ai.meta.com/blog/meta-llama-3-1/.

Mienye, I.D., Jere, N., 2024. A survey of decision trees: Concepts, algorithms, and applications. IEEE Access 12, 86716–86727. http://dx.doi.org/10.1109/ACCESS.2024.3416838.

Min, Q., Lu, Y., Liu, Z., Su, C., Wang, B., 2019. Machine Learning based Digital Twin Framework for Production Optimization in Petrochemical Industry. Int. J. Inf. Manage. 49, 502–519. http://dx.doi.org/10.1016/j.ijinfomgt.2019.05.020, URL https://linkinghub.elsevier.com/retrieve/pii/S0268401218311484.

Müller-Zhang, Z., Kuhn, T., Antonino, P.O., 2023. Towards live decision-making for service-based production: Integrated process planning and scheduling with Digital Twins and Deep-Q-Learning. Comput. Ind. 149, http://dx.doi.org/10.1016/j.compind.2023.103933.

Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., Myers, B., 2024. Using an LLM to help with code understanding. In: IEEE/ACM International Conference on Software Engineering. pp. 1–13. http://dx.doi.org/10.1145/3597503.3639187.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., Van Keulen, M., Seifert, C., 2023. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable AI. ACM Comput. Surv. 55 (13s), 1–42. http://dx.doi.org/10.1145/3583558.

Oĝuz, C., Ercan, M.F., 2005. A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. J. Sched. 8, 323–351. http://dx.doi.org/10.1007/s10951-005-1640-y.

Oliva, R., 2019. Intervention as a research strategy. J. Oper. Manage. 65 (7), 710–724. http://dx.doi.org/10.1002/joom.1065.

Onaji, I., Tiwari, D., Soulatiantork, P., Song, B., Tiwari, A., 2022. Digital twin in manufacturing: conceptual framework and case studies. Int. J. Comput. Integr. Manuf. 35 (8), 831–858. http://dx.doi.org/10.1080/0951192X.2022.2027014.

O'neill, M., Ryan, C., Keijzer, M., Cattolico, M., 2003. Crossover in grammatical evolution. Genet. Program. Evolvable Mach. 4, 67–93. http://dx.doi.org/10.1023/A:1021877127167.

OpenAI, 2024. Hello GPT-4o. URL https://openai.com/index/hello-gpt-4o/.

Pan, Q.-K., Ruiz, R., Alfaro-Fernández, P., 2017. Iterated search methods for earliness and tardiness minimization in hybrid flowshops with due windows. Comput. Oper. Res. 80, 50–60. http://dx.doi.org/10.1016/j.cor.2016.11.022.

Panchal, G., Clegg, B., Koupaei, E.E., Masi, D., Collis, I., 2024. Digital transformation and business intelligence for a SME: systems thinking action research using PrOH modelling. Procedia Comput. Sci. 232, 1809–1818. http://dx.doi.org/10.1016/j.procs.2024.02.003.

Park, K.T., Son, Y.H., Noh, S.D., 2021. The architectural framework of a cyber physical logistics system for digital-twin-based supply chain control. Int. J. Prod. Res. 59 (19), 5721–5742. http://dx.doi.org/10.1080/00207543.2020.1788738.

Pohl, D., 2019. DHL supply chain partners Tetra Pak to implement its first digital twin warehouse in Asia Pacific. URL https://group.dhl.com/en/media-relations/press-releases/2019/dhl-supply-chain-partners-tetra-pak-implement-first-digital-twin-warehouse-asia-pacific.html.

Puthanveettil Madathil, A., Luo, X., Liu, Q., Walker, C., Madarkar, R., Qin, Y., 2025. A review of explainable artificial intelligence in smart manufacturing. Int. J. Prod. Res. 1–44. http://dx.doi.org/10.1080/00207543.2025.2513574.

Qi, Q., Tao, F., 2018. Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. IEEE Access 6, 3585–3593. http://dx.doi.org/10.1109/ACCESS.2018.2793265.

Ramachandram, D., Joshi, H., Zhu, J., Gandhi, D., Hartman, L., Raval, A., 2025. Transparent AI: The Case for Interpretability and Explainability. http://dx.doi.org/10.48550/arXiv.2507.23535, arXiv preprint arXiv:2507.23535.

Rashidi, E., Jahandar, M., Zandieh, M., 2010. An improved hybrid multi-objective parallel genetic algorithm for hybrid flow shop scheduling with unrelated parallel machines. Int. J. Adv. Manuf. Technol. 49, 1129–1139. http://dx.doi.org/10.1007/s00170-009-2475-z.

Reason, P., Bradbury, H., 2001. Handbook of Action Research. Sage, http://dx.doi.org/10.4135/9781848607934.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. "Why should I trust you?" Explaining the predictions of any classifier. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. http://dx.doi.org/10.18653/v1/N16-3020.

Romano, P., Formentini, M., 2012. Designing and implementing open book accounting in buyer-supplier dyads: A framework for supplier selection and motivation. Int. J. Prod. Econ. 137 (1), 68–83. http://dx.doi.org/10.1016/j.ijpe.2012.01.013.

Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1 (5), 206–215. http://dx.doi.org/10.1038/s42256-019-0048-x.

Ruiz, R., Vázquez-Rodríguez, J.A., 2010. The hybrid flow shop scheduling problem. European J. Oper. Res. 205 (1), 1–18. http://dx.doi.org/10.1016/j.ejor.2009.09.024.

Ryan, C., Collins, J.J., Neill, M.O., 1998. Grammatical evolution: Evolving programs for an arbitrary language. In: Genetic Programming: First European Workshop. pp. 83–96. http://dx.doi.org/10.1007/BFb0055930.

Sadeghi R., K., Ojha, D., Kaur, P., Mahto, R.V., Dhir, A., 2024. Explainable artificial intelligence and agile decision-making in supply chain cyber resilience. Decis. Support Syst. 180, http://dx.doi.org/10.1016/j.dss.2024.114194.

Schmidberger, S., Bals, L., Hartmann, E., Jahns, C., 2009. Ground handling services at European hub airports: Development of a performance measurement system for benchmarking. Int. J. Prod. Econ. 117 (1), 104–116. http://dx.doi.org/10.1016/j.ijpe.2008.10.006.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. URL http://arxiv.org/abs/1707.06347.

Semenov, Y., Semenova, O., Kuvataev, I., 2020. Solutions for digitalization of the coal industry implemented in UC Kuzbassrazrezugol. In: E3S Web of Conferences, vol. 174, EDP Sciences, p. 01042. http://dx.doi.org/10.1051/e3sconf/202017401042.

Serafim, P.B., Crescenzi, P., Gezici, G., Cappuccio, E., Rinzivillo, S., Giannotti, F., 2024. Exploring Large Language Models Capabilities to Explain Decision Trees. In: HHAI Hybrid Human AI Systems for the Social Good. IOS Press, pp. 64–72. http://dx.doi.org/10.3233/FAIA240183.

Seshadri, B.R., Krishnamurthy, T., 2017. Structural health management of damaged aircraft structures using digital twin concept. In: AIAA/AHS Adaptive Structures Conference. pp. 1–13. http://dx.doi.org/10.2514/6.2017-1675.

Shani, A.B.R., Coghlan, D., 2021. Action research in business and management: A reflective review. Action Res. 19 (3), 518–541. http://dx.doi.org/10.1177/1476750319852147.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., Calinescu, A., 2022. Digital Twins: State of the art theory and practice, challenges, and open research questions. J. Ind. Inf. Integr. 30, http://dx.doi.org/10.1016/j.jii.2022.100383.

Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. http://dx.doi.org/10.48550/arXiv.1312.6034, arXiv preprint arXiv:1312.6034.

Tortorella, G.L., Saurin, T.A., Hines, P., Antony, J., Samson, D., 2023. Myths and facts of industry 4.0. Int. J. Prod. Econ. 255, 108660. http://dx.doi.org/10.1016/j.ijpe.2022.108660.

Urlings, T., Ruiz, R., Serifoglu, F.S., 2010. Genetic algorithms with different representation schemes for complex hybrid flexible flow line problems. Int. J. Metaheuristics 1 (1), 30–54. http://dx.doi.org/10.1504/IJMHeur.2010.033122.

Wang, J., Li, X., Wang, P., Liu, Q., 2024. Bibliometric analysis of digital twin literature: A review of influencing factors and conceptual structure. Technol. Anal. Strat. Manag. 36 (1), 166–180. http://dx.doi.org/10.1080/09537325.2022.2026320.

Wang, Y., Pei, Y., 2024. A comprehensive survey on interactive evolutionary computation in the first two decades of the 21st century. Appl. Soft Comput. 164, 111950. http://dx.doi.org/10.1016/j.asoc.2024.111950.

Wang, J., Ye, L., Gao, R.X., Li, C., Zhang, L., 2019. Digital Twin for rotating machinery fault diagnosis in smart manufacturing. Int. J. Prod. Res. 57 (12), 3920–3934. http://dx.doi.org/10.1080/00207543.2018.1552032.

Watkins, C.J.C.H., 1989. Learning from Delayed Rewards (Ph.D. thesis). King's College, URL https://www.cs.rhul.ac.uk/~chrisw/thesis.html.

West, S., Stoll, O., Meierhofer, J., Züst, S., 2021. Digital twin providing new opportunities for value co-creation through supporting decision-making. Appl. Sci. 11 (9), http://dx.doi.org/10.3390/app11093750.

Whitley, D., Gordon, V.S., Mathias, K., 1994. Lamarckian evolution, the Baldwin effect and function optimization. In: Parallel Problem Solving from Nature. pp. 5–15. http://dx.doi.org/10.1007/3-540-58484-6_245.

Wooley, A., Silva, D.F., Bitencourt, J., 2023. When is a simulation a digital twin? A systematic literature review. Manuf. Lett. 35, 940–951. http://dx.doi.org/10.1016/j.mfglet.2023.08.014.

Yang, F.-J., 2018. An implementation of naive Bayes classifier. In: International Conference on Computational Science and Computational Intelligence. IEEE, pp. 301–306. http://dx.doi.org/10.1109/CSCI46756.2018.00065.

Yu, C., Semeraro, Q., Matta, A., 2018. A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. Comput. Oper. Res. 100, 211–229. http://dx.doi.org/10.1016/j.cor.2018.07.025.

Zheng, X., Lu, J., Kiritsis, D., 2022. The emergence of cognitive digital twin: vision, challenges and opportunities. Int. J. Prod. Res. 60 (24), 7610–7632. http://dx.doi.org/10.1080/00207543.2021.2014591.

Zhou, R., Bacardit, J., Brownlee, A.E.I., Cagnoni, S., Fyvie, M., Iacca, G., McCall, J., van Stein, N., Walker, D.J., Hu, T., 2024. Evolutionary Computation and Explainable AI: A roadmap to understandable intelligent systems. IEEE Trans. Evol. Comput. 1–16. http://dx.doi.org/10.1109/TEVC.2024.3476443.

Ziems, N., Liu, G., Flanagan, J., Jiang, M., 2023. Explaining tree model decisions in natural language for network intrusion detection. rXiv preprint arXiv:2310.19658. URL http://arxiv.org/abs/2310.19658.