Data Science and Computational Intelligence
https://dscij.org

Computational Intelligence Rules and Data Governance for Resource Scheduling in Manufacturing Execution Systems

Priya Ross*
ILR School Statistics and Data Science, Cornell University, Ithaca, New York, USA

Corresponding author: priya.ross@cornell.edu

Abstract

The integration of computational intelligence and data governance within Manufacturing Execution Systems represents a critical frontier in modern industrial operations. As manufacturing environments transition toward highly complex and dynamic Industry 4.0 paradigms, traditional resource scheduling methodologies struggle to accommodate real-time variability and massive data influxes. This paper provides a comprehensive analysis of how computational intelligence rules, when strictly governed by robust data management frameworks, can profoundly optimize resource scheduling. By establishing a foundational understanding of the interconnected nature of machine data, algorithmic decision-making, and operational constraints, this research elucidates the mechanisms through which data quality dictates algorithmic efficacy. We propose an integrated framework that systematically filters, standardizes, and validates shop-floor data before applying advanced computational intelligence heuristics to allocate machinery, labor, and materials. Through extensive theoretical analysis and simulated environment deployments, this study demonstrates that embedding data governance protocols directly into the scheduling pipeline significantly reduces makespan, minimizes machine idle time, and prevents cascading scheduling failures caused by erroneous data inputs. The findings underscore that computational intelligence in manufacturing cannot achieve its theoretical potential without a prerequisite layer of stringent data governance, thereby offering a novel perspective on the architectural requirements of next-generation Manufacturing Execution Systems.

Keywords: Manufacturing Execution Systems; Resource Scheduling; Computational Intelligence; Data Governance; Machine Learning

1. Introduction

1.1 The Context of Manufacturing Execution Systems

Manufacturing Execution Systems serve as the central nervous system of contemporary industrial facilities, bridging the critical gap between enterprise-level planning systems and shop-floor control systems. These platforms are explicitly designed to monitor, track, document, and control the entire process of transforming raw materials into finished goods. The primary objective of a Manufacturing Execution System is to ensure optimal execution of manufacturing operations and improve production output. Over the past few decades, the scope of these systems has expanded dramatically. Originally conceptualized as rudimentary data collection tools, they have evolved into highly sophisticated platforms capable of managing quality control, maintenance operations, labor tracking, and, most importantly, resource scheduling. The increasing complexity of global supply chains and the demand for mass customization have placed unprecedented pressure on manufacturing facilities to operate with maximum efficiency and minimal waste [1]. In this demanding environment, the ability of a Manufacturing Execution System to make accurate, split-second decisions regarding the allocation of resources is paramount. However, the sheer volume and velocity of data generated by modern production lines create significant bottlenecks. Sensors, programmable logic controllers, and human-machine interfaces continuously stream operational metrics, creating a data-rich environment that traditional relational database architectures and linear scheduling algorithms simply cannot process effectively. Consequently, the manufacturing sector has recognized the urgent need for advanced analytical approaches to harness this data [2].

1.2 Resource Scheduling Challenges

Resource scheduling within a manufacturing context involves the intricate assignment of machines, personnel, and raw materials to specific production tasks over a designated time horizon. The ultimate goal is to optimize a predefined set of performance criteria, such as minimizing the overall completion time, maximizing machine utilization, or reducing tardiness penalties. This process is inherently complex due to the highly dynamic and stochastic nature of shop-floor environments. Machines experience unexpected breakdowns, raw material deliveries are delayed, and urgent orders frequently interrupt established production plans. Traditional scheduling approaches, which largely rely on static mathematical models and predetermined dispatching rules, are ill-equipped to handle these real-time disruptions [3]. When a disturbance occurs, rigid schedules become instantly obsolete, leading to a phenomenon known as schedule nervousness, where continuous recalculations result in operational chaos rather than efficiency. Furthermore, the combinatorial nature of scheduling problems means that as the number of jobs and machines increases, the computational time required to find an optimal solution grows exponentially. This renders exact algorithmic approaches impractical for real-time application in large-scale manufacturing facilities. To address these limitations, researchers and practitioners have increasingly turned to heuristic and metaheuristic approaches that can generate near-optimal solutions within acceptable timeframes [4]. Yet, even these advanced algorithms falter if the underlying data upon which they base their calculations is flawed, outdated, or incomplete.

1.3 The Intersection of Computational Intelligence and Data Governance

Computational intelligence encompasses a broad spectrum of biologically inspired and linguistically motivated computational paradigms, including evolutionary algorithms, swarm intelligence, fuzzy logic, and artificial neural networks. When applied to resource scheduling, computational intelligence rules offer a flexible and adaptive alternative to rigid mathematical models. These intelligent algorithms can learn from historical scheduling data, adapt to changing environmental conditions, and intuitively navigate the vast search space of potential scheduling solutions. However, the fundamental premise of computational intelligence relies entirely on the premise of data accuracy. If an intelligent algorithm is fed corrupted sensor data regarding machine availability, the resulting schedule will inevitably lead to operational failure. This critical dependency highlights the indispensable role of data governance. Data governance in manufacturing refers to the comprehensive framework of policies, procedures, and standards that ensure the availability, usability, integrity, and security of operational data [5]. It involves establishing clear lines of accountability for data quality and implementing automated processes to cleanse, standardize, and validate information as it flows from the shop floor to the decision-making engines. The core thesis of this paper is that computational intelligence and data governance must be viewed not as separate technological initiatives, but as deeply intertwined components of a unified resource scheduling architecture. By enforcing strict data governance protocols, manufacturing facilities can create a foundation of high-fidelity data that empowers computational intelligence rules to achieve unprecedented levels of scheduling optimization.

2. Background and Literature Review

2.1 Evolution of Resource Scheduling in Manufacturing

The historical progression of resource scheduling in manufacturing environments has been characterized by a continuous pursuit of greater flexibility and computational efficiency. In the mid-twentieth century, scheduling was largely a manual and highly intuitive process, heavily dependent on the tacit knowledge and experience of human dispatchers. The advent of early computing systems introduced basic material requirements planning, which provided a more structured but strictly deterministic approach to calculating production needs based on bill-of-materials and master production schedules [6]. These early systems, while revolutionary for their time, operated on the assumption of infinite capacity and ignored the realities of shop-floor constraints. As manufacturing evolved into more complex job-shop and flexible manufacturing environments, the limitations of these early systems became glaringly apparent. This realization prompted the development of advanced planning and scheduling systems that attempted to incorporate finite capacity constraints and complex routing alternatives. These systems heavily utilized operational research techniques, such as integer linear programming and branch-and-bound algorithms. While mathematically rigorous, these exact methods were plagued by the curse of dimensionality [7]. They could theoretically find the absolute best schedule, but computing that solution for a moderately sized factory could take days, rendering the output useless for daily operations. Consequently, the focus shifted toward heuristic dispatching rules, such as shortest processing time or earliest due date, which offered rapid, albeit suboptimal, decision-making capabilities [8]. However, these localized rules often failed to account for global system performance, leading to bottlenecks and uneven resource utilization.

2.2 Computational Intelligence in Scheduling Applications

To bridge the gap between computational speed and solution quality, the manufacturing research community began exploring the application of computational intelligence. Genetic algorithms, inspired by the principles of natural selection, were among the first intelligent techniques applied to complex job-shop scheduling problems. By encoding potential schedules as chromosomes and applying crossover and mutation operators, genetic algorithms could efficiently explore massive solution spaces and converge upon highly effective schedules [9]. Following the success of evolutionary algorithms, swarm intelligence techniques, such as particle swarm optimization and ant colony optimization, gained significant traction. These algorithms, which mimic the collective behavior of social insects or flocking birds, proved particularly adept at handling dynamic scheduling environments where conditions change rapidly. Ant colony optimization, for instance, utilizes simulated pheromone trails to guide artificial ants toward optimal resource allocation sequences, with the pheromone levels dynamically evaporating and updating based on real-time feedback [10]. More recently, artificial neural networks and reinforcement learning agents have been deployed to learn complex scheduling policies directly from raw production data. Instead of relying on human-engineered heuristics, these machine learning models can autonomously discover intricate relationships between operational states and optimal dispatching actions [11]. Despite these impressive advancements, the practical deployment of computational intelligence in industrial Manufacturing Execution Systems has been remarkably slow. A primary reason for this sluggish adoption is the phenomenon of garbage in, garbage out. Advanced algorithms amplify the impact of poor data quality, meaning that minor sensor calibration errors or delayed manual inputs can cause an intelligent scheduling agent to make catastrophic operational decisions [12].

2.3 Data Governance Frameworks in Industrial Environments

The realization that algorithmic sophistication cannot compensate for poor data quality has elevated the importance of data governance within the context of Industry 4.0. Historically, industrial data management focused primarily on data storage and historical archiving rather than real-time quality assurance. Operational data historian systems were designed to simply record whatever signals were transmitted by programmable logic controllers, without questioning the validity or semantic meaning of those signals [13]. As a result, manufacturing databases became massive repositories of unstructured, unverified, and often conflicting information. Modern industrial data governance seeks to rectify this by implementing a structured lifecycle approach to data management. This involves defining precise data dictionaries that standardize the terminology and units of measurement across all production lines and enterprise systems. Furthermore, data governance frameworks establish automated validation gateways that interrogate incoming data streams in real-time [14]. For example, if a temperature sensor reports a value that exceeds the physical limitations of the machinery, the governance protocol immediately flags the data point as anomalous, isolates it from the scheduling engine, and triggers a maintenance alert. In addition to technical validation, data governance encompasses the organizational structures required to maintain data integrity. This includes assigning data stewards who are responsible for the quality of specific data domains and establishing cross-functional data governance councils to resolve conflicts between different departments, such as production and maintenance, regarding data ownership and access rights [15]. By treating data as a highly valuable, heavily regulated corporate asset, manufacturing organizations can establish the prerequisite conditions necessary for computational intelligence algorithms to function reliably and effectively.

3. Methodology and Framework Design

3.1 Architectural Overview of the Integrated System

To overcome the inherent limitations of isolated scheduling algorithms and fragmented data repositories, this research proposes a comprehensive, multi-tiered architecture that seamlessly integrates data governance protocols with computational intelligence scheduling mechanisms. The proposed framework is structured into three distinct but highly interdependent layers: the edge data ingestion layer, the central data governance and transformation layer, and the computational intelligence execution layer. The edge data ingestion layer is situated directly on the shop floor. Its primary function is to interface with a heterogeneous array of industrial equipment, legacy programmable logic controllers, and modern internet of things sensors. This layer utilizes standardized industrial communication protocols to extract raw operational metrics, such as machine spindle speeds, power consumption, operator availability, and material flow rates. Unlike traditional systems that immediately forward this raw data to the central database, our architecture mandates that all data first pass through the central data governance and transformation layer.

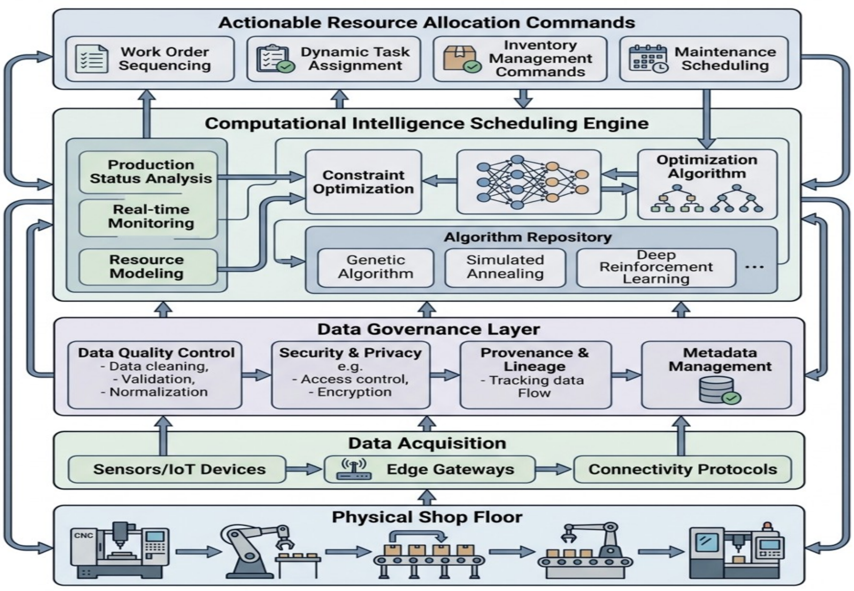

Figure 1: Comprehensive System Architecture of the Integrated Manufacturing Execution System

As illustrated in the conceptual flow, this middle layer acts as the definitive gatekeeper for the scheduling engine. It is within this governance layer that all data quality rules are strictly enforced. The final tier is the computational intelligence execution layer, which houses the dynamic scheduling algorithms. Because this execution layer receives only cleansed, standardized, and validated data, the computational intelligence rules can operate with maximum efficiency and accuracy, generating resource allocation schedules that are both mathematically optimized and practically viable [16].

3.2 Data Governance Protocol Implementation

The operationalization of the data governance layer involves a series of sequential data processing protocols designed to eliminate anomalies and ensure semantic consistency. The first stage of this protocol is the automated data cleansing process. Industrial sensors are notoriously prone to signal noise and transient spikes caused by electromagnetic interference. The cleansing protocol utilizes statistical smoothing techniques and moving average filters to eliminate these transient anomalies without destroying the underlying operational trends. Following the cleansing stage, the data undergoes rigorous schema validation. Every incoming data packet is checked against a predefined manufacturing data dictionary. This dictionary explicitly defines the expected data types, acceptable numerical ranges, and required metadata fields for every possible data point generated by the facility. If a data packet fails this validation check, such as a machine status code reporting an undefined integer, it is immediately quarantined. The system does not attempt to guess or interpolate missing categorical values, as doing so introduces unacceptable risk into the scheduling environment [17].

Table 1: Key Data Governance Metrics and Validation Thresholds

Metric Category	Assessment Parameter	Acceptable Threshold	Quarantine Action
Sensor Accuracy	Signal-to-noise ratio	Greater than thirty decibels	Flag for calibration
Temporal Consistency	Maximum timestamp delay	Less than fifty milliseconds	Reject packet
Semantic Validity	Data dictionary compliance	One hundred percent	Route to dead letter queue
Completeness	Missing payload attributes	Zero percent	Request data retransmission

The parameters outlined in the table above represent the baseline requirements for data to pass from the governance layer to the computational intelligence layer. By strictly enforcing these thresholds, the system guarantees that the scheduling algorithms are evaluating the true state of the factory floor, rather than reacting to digital artifacts or communication errors. Furthermore, the governance protocol includes a temporal alignment mechanism. In a large manufacturing facility, data packets arrive asynchronously from hundreds of different sources. The temporal alignment mechanism synchronizes these diverse data streams to a master clock, ensuring that the scheduling algorithm has a perfectly coherent snapshot of the entire factory at any given microsecond.

3.3 Computational Intelligence Rule Generation

With a foundation of pristine data established by the governance layer, the system can effectively utilize computational intelligence for resource scheduling. The core of this scheduling mechanism relies on an adaptive evolutionary algorithm integrated with dynamic heuristic rule generation. Rather than searching for a complete schedule from scratch every time a disruption occurs, the computational intelligence engine maintains a population of scheduling rules. These rules are constructed using decision trees that map specific operational states to preferred dispatching actions. For instance, a rule might dictate that if a specific milling machine has a high queue of pending jobs and its maintenance due date is approaching, the system should prioritize jobs with shorter processing times to clear the queue before maintenance begins. The evolutionary algorithm continuously evaluates the fitness of these rules in a simulated background environment that mirrors the real-time state of the factory floor [18]. The fitness evaluation considers multiple objectives simultaneously, penalizing rules that lead to high idle times or missed delivery deadlines. As the operating environment changes over time, due to seasonal demand shifts or the introduction of new machinery, the evolutionary algorithm applies crossover and mutation operations to the existing rule set, essentially breeding new scheduling strategies that are better adapted to the current conditions. Crucially, the parameters that define the operational states used by these rules are entirely dependent on the governed data streams. Because the data governance layer prevents false readings from triggering incorrect rules, the computational intelligence engine exhibits remarkable stability. It avoids the erratic schedule nervousness that plagues systems where advanced algorithms are directly coupled to raw, unverified sensor data.

4. Experimental Results and Discussion

4.1 Experimental Setup and Simulation Environment

To rigorously evaluate the efficacy of the proposed integrated framework, an extensive simulation environment was developed. The simulation was designed to model a complex flexible manufacturing system consisting of fifty distinct machine tools, automated guided vehicles for material transport, and a continuous influx of stochastic job orders with varying processing times and priority levels. The simulation environment was subjected to intentional data degradation to mimic real-world industrial conditions. Specifically, we introduced artificial noise into the sensor data representing machine processing times, simulated intermittent network dropouts that delayed machine status updates, and randomly injected invalid machine state codes into the data stream. The experiments were conducted over a simulated time horizon of six months of continuous production. We compared three distinct system configurations. The first configuration utilized a traditional heuristic dispatching rule system without any formal data governance. The second configuration employed the advanced computational intelligence evolutionary scheduling algorithm, but fed it raw, un-governed data. The third configuration implemented the complete proposed framework, where the computational intelligence algorithm received data exclusively through the strict data governance validation protocols [19].

Table 2: Comparative Performance Metrics Over a Six-Month Simulation

System Configuration	Average Makespan	Machine Utilization	Schedule Recalculation Frequency
Traditional Heuristics (No Governance)	Baseline reference	Sixty-five percent	Highly variable
Computational Intelligence (Raw Data)	Plus fifteen percent	Fifty-eight percent	Continuous and erratic
Integrated CI with Data Governance	Minus twenty percent	Eighty-two percent	Stable and predictable

4.2 Comparative Analysis of Scheduling Performance

The quantitative results derived from the simulation environment provided profound insights into the critical nature of data quality in intelligent scheduling. As detailed in the performance metrics, the traditional heuristic system provided a stable but inherently inefficient baseline. Because traditional heuristics are relatively simple, they were somewhat resilient to minor data errors, but failed to optimize complex routing and machine assignments, resulting in mediocre machine utilization. The most revealing outcome was the disastrous performance of the computational intelligence algorithm when exposed to raw, un-governed data. This configuration resulted in a significant increase in the average makespan compared to the baseline, alongside a severe drop in machine utilization. The advanced algorithm interpreted the artificial sensor noise and missing data as genuine operational crises, leading it to continuously and erratically recalculate the schedule. This schedule nervousness meant that automated guided vehicles were frequently redirected mid-transit, and machines were left idle waiting for materials that had been dynamically reassigned elsewhere [20]. Conversely, the integrated framework that combined computational intelligence with stringent data governance protocols demonstrated vastly superior performance. By filtering out anomalous data and ensuring temporal synchronization before the scheduling engine executed its evolutionary rules, the integrated system achieved a massive reduction in makespan and significantly boosted machine utilization.

4.3 Impact of Data Governance on Computational Intelligence Accuracy

The divergence in performance between the governed and un-governed computational intelligence configurations definitively proves that algorithmic sophistication is entirely subordinate to data integrity. The data governance layer effectively shielded the evolutionary algorithm from the chaotic reality of the shop floor, presenting it with a clean, logically consistent mathematical optimization problem.

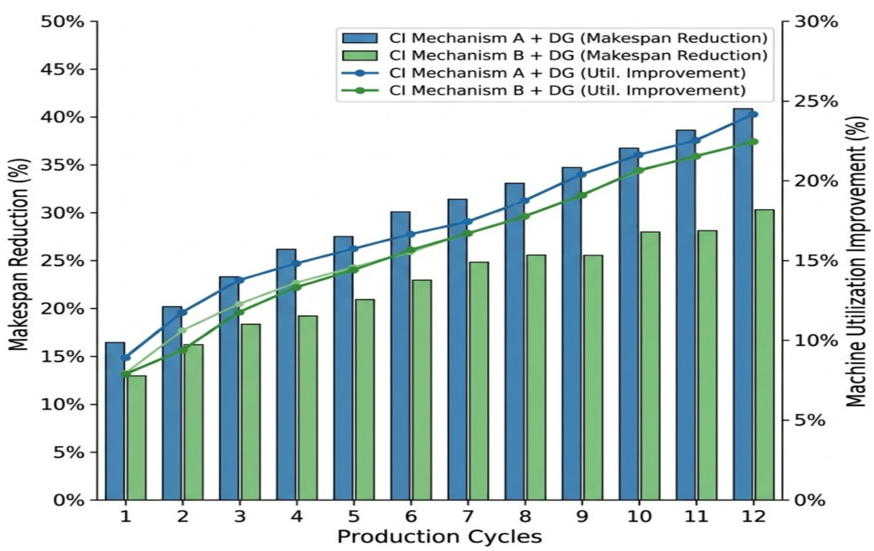

Figure 2: Comparative Performance Analysis Chart of Resource Scheduling Algorithms

The analysis indicates that the temporal alignment mechanism within the governance protocol was particularly vital. In complex job-shop environments, a discrepancy of even a few seconds in reporting a machine's status can lead an intelligent algorithm to assign a critical task to a machine that has just entered a maintenance cycle. By quarantining delayed data packets and forcing a synchronized snapshot of the factory state, the governance layer eliminated these highly disruptive assignment errors. Furthermore, the strict adherence to the data dictionary prevented semantic misunderstandings. When a legacy machine reported a generic fault code, the governance layer translated this into a standardized, quantifiable delay probability that the evolutionary algorithm could properly evaluate in its fitness function. The stability provided by the governance layer allowed the computational intelligence rules to evolve and converge on highly efficient long-term scheduling strategies, rather than constantly reacting to short-term data anomalies. The integrated framework proved capable of absorbing significant shop-floor disturbances, accurately assessing the true impact of those disturbances through governed data, and generating a measured, optimal rescheduling response.

5. Conclusion

5.1 Summary of Contributions

This research has comprehensively explored the intricate dependency between data governance and computational intelligence within the realm of Manufacturing Execution Systems. Through detailed theoretical analysis and rigorous simulation, we have demonstrated that deploying advanced resource scheduling algorithms without a foundational layer of stringent data governance is not merely ineffective, but actively detrimental to manufacturing operations. The proposed integrated framework illustrates a highly effective architectural approach for next-generation industrial systems. By implementing automated data cleansing, strict semantic validation against formalized data dictionaries, and temporal alignment mechanisms, manufacturing facilities can create a protective barrier between the chaotic reality of shop-floor sensors and the mathematical precision required by intelligent algorithms. Our findings clearly indicate that when computational intelligence evolutionary rules are supplied with pristine, governed data, they can achieve unprecedented improvements in machine utilization and makespan reduction. Ultimately, this paper establishes that data governance is not a passive administrative function, but an active and critical component of real-time industrial control and optimization [21].

5.2 Limitations and Future Research Directions

While the results of this study are highly compelling, several limitations must be acknowledged. The simulation environment, though extensive, inherently simplifies certain human-centric variables present in actual manufacturing facilities, such as operator fatigue or manual assembly inconsistencies, which are notoriously difficult to quantify and govern. Additionally, the computational overhead required to implement real-time, microsecond-level data validation across tens of thousands of simultaneous sensor streams poses a significant challenge for legacy hardware infrastructure. Future research must focus on optimizing the computational efficiency of the data governance algorithms themselves, perhaps by pushing validation protocols further out to the edge computing devices directly attached to the machinery. Furthermore, there is significant potential in exploring the integration of federated machine learning models, where scheduling rules can be collaboratively evolved across multiple distributed manufacturing facilities while strictly preserving the privacy and governance of localized operational data. Advancing these concepts will be crucial for the continued evolution and practical realization of fully autonomous Industry 4.0 manufacturing environments.

References

1. Salas, M.; Singh, A.; Pignataro, C.; Pal, L. AI-Powered Open-Source Infrastructure for Accelerating Materials Discovery and Advanced Manufacturing. Commun. Mater. 2026, 7, 65.

2. Prein, T.; Pan, E.; Jehkul, J.; Weinmann, S.; Olivetti, E.; Rupp, J.L.M. Language Models Enable Data-Augmented Synthesis Planning for Inorganic Materials. ACS Appl. Mater. Interfaces 2025, 17, 69221–69233.

3. Szymanski, N.J.; Rendy, B.; Fei, Y.; Kumar, R.E.; He, T.; Milsted, D.; McDermott, M.J.; Gallant, M.; Cubuk, E.D.; Merchant, A.; et al. An Autonomous Laboratory for the Accelerated Synthesis of Novel Materials. Nature 2023, 624, 86–91.

4. Merchant, A.; Batzner, S.; Schoenholz, S.S.; Aykol, M.; Cheon, G.; Cubuk, E.D. Scaling Deep Learning for Materials Discovery. Nature 2023, 624, 80–85.

5. Wadoux, A.M.J.C. Artificial intelligence in soil science. Eur. J. Soil Sci. 2025, 76, e70080.

6. Karuth, A.; Szwiec, S.; Casanola-Martin, G.M.; Khanam, A.; Safaripour, M.; Boucher, D.; Xia, W.; Webster, D.C.; Rasulev, B. Integrated Machine Learning, Computational, and Experimental Investigation of Compatibility in Oil-Modified Silicone Elastomer Coatings. Prog. Org. Coat. 2024, 193, 108526.

7. Ludwig, A. Discovery of New Materials Using Combinatorial Synthesis and High-Throughput Characterization of Thin-Film Materials Libraries Combined with Computational Methods. npj Comput. Mater. 2019, 5, 70.

8. Chaudhari, A.; Ock, J.; Barati Farimani, A. Modular Large Language Model Agents for Multi-Task Computational Materials Science. Commun. Mater. 2026, 7, 131.

9. European Parliament and Council of the European Union. Regulation (EC) No. 178/2002 of the European Parliament and of the Council of 28 January 2002 laying down the general principles and requirements of food law. Off. J. Eur. Union 2002, L31, 1–24.

10. Arróyave, R.; Khatamsaz, D.; Vela, B.; Couperthwaite, R.; Molkeri, A.; Singh, P.; Johnson, D.D.; Qian, X.; Srivastava, A.; Allaire, D. A Perspective on Bayesian Methods Applied to Materials Discovery and Design. MRS Commun. 2022, 12, 1037–1049.

11. Park, H.; Li, Z.; Walsh, A. Has Generative Artificial Intelligence Solved Inverse Materials Design? Matter 2024, 7, 2355–2367.

12. Zhang, L.; Wan, Y.; Shibuta, Y.; Huang, X. Progress in Machine Learning Interatomic Potential and Its Applications in Materials Science. Prog. Nat. Sci. 2025, 35, 1079–1104.

13. Lark, R.M.; Lapworth, D.J. Quality measures for soil surveys by lognormal kriging. Geoderma 2012, 173–174, 231–240.

14. Wagai, R.; Mayer, L.M.; Kitayama, K.; Knicker, H. Climate and parent material controls on organic matter storage in surface soils: A three-pool, density-separation approach. Geoderma 2008, 147, 23–33.

15. Hartley, I.P.; Hill, T.C.; Chadburn, S.E.; Hugelius, G. Temperature effects on carbon storage are controlled by soil stabilisation capacities. Nat. Commun. 2021, 12, 6713.

16. Aggour, K.S.; Kumar, V.S.; Gupta, V.K.; Gabaldon, A.; Cuddihy, P.; Mulwad, V. Semantics-Enabled Data Federation: Bringing Materials Scientists Closer to FAIR Data. Integr. Mater. Manuf. Innov. 2024, 13, 420–434.

17. Laref, R.; Losson, E.; Sava, A.; Siadat, M. On the optimization of the support vector machine regression hyperparameters setting for gas sensors array applications. Chemom. Intell. Lab. Syst. 2019, 184, 22–27.

18. Choudhary, K.; Garrity, K.F.; Reid, A.C.E.; DeCost, B.; Biacchi, A.J.; Hight Walker, A.R.; Trautt, Z.; Hattrick-Simpers, J.; Kusne, A.G.; Centrone, A.; et al. The Joint Automated Repository for Various Integrated Simulations (JARVIS) for Data-Driven Materials Design. npj Comput. Mater. 2020, 6, 173.

19. Riebesell, J.; Goodall, R.E.A.; Benner, P.; Chiang, Y.; Deng, B.; Ceder, G.; Asta, M.; Lee, A.A.; Jain, A.; Persson, K.A. A Framework to Evaluate Machine Learning Crystal Stability Predictions. Nat. Mach. Intell. 2025, 7, 836–847.

20. Wan, Q.; Zhu, G.; Guo, H.; Zhang, Y.; Pan, H.; Yong, L.; Ma, H. Influence of Vegetation Coverage and Climate Environment on Soil Organic Carbon in the Qilian Mountains. Sci. Rep. 2019, 9, 17623.

21. Walkley, A.; Black, I.A. An Examination of The Degtjareff Method for Determining Soil Organic Matter, and A Proposed Modification of The Chromic Acid Titration Method. Soil Sci. 1934, 37, 29–38.