Big Data-Driven Agent Modeling and Dynamic Collaborative Optimization for Industrial Processes

Chunqi Jiao, Junwei Zhang*

Harbin Institute of Information Technology, Harbin, 150431, China
Corresponding author: zjw_1997@sina.cn

Abstract: *The deep integration of the Industrial Internet of Things and big data is driving a paradigm shift in industrial process modeling. Traditional mechanistic models struggle to capture nonlinearity and time-varying characteristics, while data-driven methods lack causal logic. To address this, this paper proposes a big data-driven framework for agent-based modeling and dynamic collaborative optimization in industrial processes. At the individual agent level, the framework constructs a structured representation integrating graph neural networks and knowledge graphs through spatiotemporal alignment of multi-source data and deep feature extraction. It incorporates physical information constraints to achieve mechanism-data hybrid modeling, and combines neural architecture search with reinforcement learning to generate autonomous decision-making agents. At the collaborative level, a dynamic topology and lightweight protocol based on process coupling degree is designed. Hierarchical reinforcement learning and game theory are utilized for task decomposition and conflict resolution, while federated learning and topology reconstruction enhance fault tolerance and self-healing capabilities. At the evolutionary level, an edge-cloud collaborative inference framework is established. Online learning and concept drift detection enable parameter adaptation, and multi-objective evolutionary algorithms combined with emergence monitoring regulate collective behavior. This research provides a theoretical framework and technical pathway for building industrial intelligent systems characterized by autonomous perception, collaborative decision-making, and continuous evolution.*

Keywords: *Big data-driven; Industrial processes; Agent modeling; Multi-agent collaboration; Dynamic collaborative optimization; Knowledge fusion; Edge-cloud collaboration; Evolutionary regulation*

Introduction

Operating within a ubiquitous sensing environment, industrial processes have seen a surge in data scale and complexity, offering the potential to overcome the limitations of traditional control methods. However, the core challenge remains extracting knowledge from multi-source heterogeneous data and transforming it into actionable decisions. While traditional mechanistic models struggle to adapt to dynamic operating conditions, purely data-driven methods are often limited in their generalization capabilities. Industrial agents, by integrating both mechanistic knowledge and data characteristics, are endowed with autonomous perception and decision-making abilities, and multi-agent networks are capable of managing large-scale coupled processes. This paper focuses on big data-driven agent modeling and dynamic collaborative optimization for industrial processes. It aims to construct a theoretical framework that spans from the construction of individual agent intelligence and multi-agent collaboration to real-time evolution. This framework encompasses deep feature extraction and knowledge fusion, collaborative architecture design, and edge-cloud collaborative regulation, thereby providing support for achieving autonomous operation in industrial processes.

1. Mechanisms of Data-Driven Agent Modeling for Industrial Processes

1.1 Feature Extraction and Representation of Industrial Processes Based on Multi-Source Heterogeneous Data

1.1.1 Spatiotemporal Alignment and Preprocessing of Multi-Modal Industrial Data

Within the Industrial Internet of Things environment, process data exhibits significant multi-source heterogeneity. Differences in sensor sampling frequencies, start times, and data formats must be addressed through alignment to eliminate spurious correlations. Therefore, a timestamp alignment method based on an event-triggered mechanism is established, which unifies high-frequency vibration signals, low-frequency temperature records, and discrete quality indicators onto a common time reference. In the spatial dimension, a graph matching algorithm is employed to correct spatial misalignments caused by measurement point offsets, ensuring that the data distribution aligns with the equipment topology. During the data cleaning phase, outlier detection and missing value imputation are integrated. A generative adversarial network is utilized to simulate the distribution under normal operating conditions, thereby reconstructing incomplete data resulting from sensor drift or communication packet loss. This process provides complete and aligned input tensors for subsequent feature extraction.

1.1.2 Deep Feature Extraction and Implicit Representation of Process States

Building upon data alignment, this step aims to extract deep features that reflect the essential changes within the process. To address the nonlinear and multi-scale characteristics of industrial time-series data, a multi-scale convolutional network based on an attention mechanism is constructed. This network extracts local fluctuation and global trend features in parallel, thereby capturing dynamic patterns at different temporal resolutions. A graph convolutional neural network is then employed to propagate information along material flow paths, learning the implicit state vectors of units under the influence of their neighborhoods. A contrastive learning strategy is introduced to pull samples from similar operating conditions closer together in the latent space while pushing those from different conditions apart, ensuring that the features possess both reconstruction capability and discriminative power for different process states. The resulting feature vectors serve as the perceptual interface for the agent, carrying a comprehensive representation of the industrial process's operational status.

1.2 Deep Knowledge Representation and Fusion Mechanisms for Complex Industrial Processes

1.2.1 Structured Knowledge Modeling Based on Graph Neural Networks and Knowledge Graphs

The prior knowledge of industrial processes is embedded within process flow diagrams, equipment design parameters, and operating procedures. Transforming this structured information into a computable format is central to knowledge representation. A process topology graph is constructed with equipment units as nodes and material and energy flows as edges. A graph attention mechanism is then utilized to dynamically learn the influence weights between adjacent nodes, enabling the model to comprehend the coupling relationships between upstream and downstream units. Knowledge graph technology is further introduced to store equipment attributes, process parameter thresholds, and causal relationships in the form of triples, thereby building an industrial knowledge base rich in semantics. Through graph embedding methods, these symbolic entities and relationships are mapped into continuous vectors, achieving a unified representation of topological structure and semantic information. This provides the agent with a mechanistic cognitive capability that extends beyond purely data-driven approaches.

1.2.2 Knowledge Fusion and Evolution Driven by Mechanism-Data Hybrid Models

A purely data-driven model is prone to overfitting, while a pure mechanistic model struggles to handle complex operating conditions. The fusion of these two approaches allows for complementary advantages. Therefore, a fusion framework based on a physics-informed neural network is constructed. Differential equations, such as those for mass and energy conservation, are embedded as soft constraints into the loss function to guide the model toward predictions that conform to physical laws. A Bayesian weighting method is then adopted to dynamically adjust the fusion weights based on the confidence levels of the data and the mechanistic model under specific operating conditions. When process disturbances occur, the data-driven component responds rapidly, while the mechanistic part provides a stable baseline. As operational data accumulates, the model continuously updates the parameter estimations of the mechanistic component, enabling the iterative evolution of knowledge.

1.3 Autonomous Construction and Decision Logic Generation of Industrial Agents

1.3.1 Task-Oriented Adaptive Generation of Agent Structures

Different industrial scenarios impose varying functional requirements on agents, making a fixed structure insufficient to balance versatility and efficiency. Therefore, neural architecture search technology is employed to automatically combine network layers within a predefined operator space, based on the task objectives and constraints, thereby generating an agent topology suitable for the current process. During the initialization phase, a meta-learning strategy is introduced to extract common parameters from a historical task library to serve as initial weights, endowing the newly constructed agent with rapid transfer capabilities. The structure generation process also considers computational resource limitations. When deployed on edge devices, network pruning and quantization compression are applied to ensure real-time performance requirements are met.

1.3.2 Evolution of Decision-Making Logic Based on Reinforcement Learning and Its Explainability Analysis

The core function of an agent is to generate optimal decisions based on the process state, and its logic is formed through interaction with the environment via reinforcement learning. The state space is defined as the latent vector output from the feature extraction layer, while the action space encompasses the setpoints of key process parameters. The reward function integrates indicators of energy efficiency, product quality, and operational stability. The Deep Deterministic Policy Gradient algorithm is employed for offline training within a simulation environment, enabling the agent to progressively approximate the optimal policy. A causal attention module is introduced into the decision-making process to attribute action selections to specific process variables. This module outputs a contribution heatmap for each decision moment, explaining why a particular parameter was adjusted at that specific time. By recording the decision-making trajectory alongside the corresponding process responses, a causal chain of the decision logic is constructed. This allows operators to trace the agent's intentions, thereby enhancing human-machine trust and laying an explainable foundation for practical industrial applications[1].

2. Dynamic Collaborative Architecture for Multi-Agent Systems in Industrial Processes

2.1 Topology Structure and Communication Protocol Design for Distributed Agent Networks

2.1.1 Dynamic Network Topology Generation Based on Industrial Process Coupling Strength

The network topology of a multi-agent system must align with the inherent coupling relationships within the industrial process to support efficient collaborative decision-making. By analyzing the transfer entropy and mutual information among process variables, the coupling strength between different production units is quantified in terms of material flow, energy exchange, and information transfer. This quantification serves as the basis for the connection weights between agents. A graph attention network is then employed to dynamically evaluate the impact of changing operating conditions on this coupling strength. When the load of a specific unit is adjusted or the status of a piece of equipment changes, the topology evolves accordingly. Areas with enhanced coupling automatically establish high-bandwidth communication links, while connections in weakly correlated regions are simplified. This process results in a dynamic network characterized by a coexistence of sparsity and local density. This adaptive topology generation mechanism ensures the necessary information exchange for collaboration while significantly reducing communication overhead and the risk of network congestion.

2.1.2 Lightweight Communication Protocol and Data Exchange Mechanism for Real-Time Collaboration

Industrial process control imposes strict constraints on communication latency, making it difficult for traditional protocol stacks to meet millisecond-level response requirements. Therefore, a lightweight publish-subscribe protocol based on the Data Distribution Service (DDS) architecture is designed. Tailored topic filtering and Quality of Service (QoS) policies are customized according to the characteristics of industrial data, transmitting only key information such as state changes or decision gradients. A compact binary serialization format is adopted to encode the data exchanged between agents, thereby compressing the data frame length. The Precision Time Protocol (PTP) is introduced to ensure that distributed nodes maintain a unified time base, allowing each agent to receive neighbor states with comparable timestamps. This mechanism supports both high-frequency periodic collaboration and event-triggered asynchronous communication, providing a reliable data channel for real-time joint decision-making by multiple agents.

2.2 Task-Driven Dynamic Division of Labor and Collaboration Mechanisms for Multi-Agent Systems

2.2.1 Hierarchical Decomposition of Global Optimization Objectives and Local Task Allocation

To drive distributed collaboration, plant-level performance indicators must be translated into executable local subtasks for each agent. A hierarchical reinforcement learning framework is therefore adopted. In this framework, an upper-level coordinator decomposes the global objective, such as energy consumption per unit of product, into phased contribution targets for agents in different areas based on the current operating conditions. Subsequently, the lower-level agents adjust their individual strategies according to these targets. An attention mechanism is employed to evaluate the marginal impact of each agent on the global objective, enabling the dynamic allocation of task weights and thereby directing resources toward critical process links. Furthermore, a gradient aggregation method with differential privacy protection is introduced. This method facilitates cross-node feedback on task execution performance while preserving the privacy of local data, consequently ensuring that local optimization directions consistently align with the global optimum.

2.2.2 Multi-Agent Collaboration Strategy and Conflict Resolution Based on Game Theory

When multiple agents make independent decisions, overall system performance may degrade due to resource competition or conflicting objectives. Therefore, multi-agent collaboration is modeled as a potential game, where a potential function is constructed to reflect the global collaborative benefit. Consequently, each agent is guided by the gradient of this potential function while pursuing its individual rewards. A distributed iterative best-response algorithm is then employed, enabling each agent to observe the actions of its neighbors and update its own strategy, gradually converging towards a Nash equilibrium. To address potential conflicts, a contribution evaluation mechanism based on the Shapley value is introduced. This mechanism imposes penalty terms on individual behaviors that harm overall collaboration, thereby encouraging agents to achieve a balance between individual rationality and collective rationality, ultimately realizing harmonious collaboration under multi-objective constraints[2].

2.3 Multi-Agent Collaborative Fault Tolerance and Self-Healing Strategies under Uncertain Operating Conditions

2.3.1 Distributed Anomaly Detection and Isolation Based on Federated Learning

Faults in industrial processes often emerge locally and propagate rapidly, necessitating early detection and isolation at the distributed level. Therefore, a federated learning framework is employed to train local anomaly detection models for each agent, such as a Long Short-Term Memory autoencoder, which is used to reconstruct sensor time-series data and calculate residuals. Each agent only uploads its model parameters to neighboring nodes. By comparing the consistency of the parameter update directions through a consensus protocol, agents can identify anomalous individuals whose parameters deviate from the group, thereby locating the fault source. Upon detecting an anomaly, the communication link with the faulty agent is immediately severed. Simultaneously, neighboring nodes are notified to disregard information published by that agent. This action prevents the propagation of erroneous decisions, forming the first line of defense for fault tolerance.

2.3.2 Self-Healing Mechanism through Dynamic Topology Reconstruction and Function Migration

After some agents fail or are isolated, the system must rapidly reorganize to maintain essential functions. Therefore, the remaining agents, based on current connectivity and task requirements, employ a distributed topology optimization algorithm to renegotiate connections and generate alternative collaborative subnets, effectively filling the structural voids left by the failed nodes. The control functions of the failed agents are subsequently migrated to neighboring agents with similar functionalities via transfer learning. These neighboring agents load pre-backed-up standby models and assume control of the corresponding loops, thereby achieving a seamless switching. This entire process is driven by a distributed consensus protocol without the need for a central coordinator, ensuring that the system maintains robust operation under uncertain conditions such as equipment failure or communication interruption, and progressively recovers to an optimal collaborative state.

3. Real-Time Optimization and Evolutionary Regulation of Industrial Process Agent Models

3.1 Real-Time Inference and Computation Migration Methods Based on Edge-Cloud Collaboration

3.1.1 Inference Task Partitioning and Edge-Cloud Collaborative Computation Based on Latency Constraints

Industrial process control imposes strict limitations on inference latency. Processing solely in the cloud struggles to meet millisecond-level response requirements, while complete local deployment is constrained by the computational capacity of edge nodes. Therefore, a task partitioning mechanism is constructed. This mechanism segments the deep neural network model layer by layer. Based on the computational complexity and data volume of each layer, shallow feature extraction components are placed on edge nodes, while deep abstract inference components are offloaded to the cloud. A linear programming method is then employed, aiming to minimize end-to-end latency while considering the computational capabilities of both edge and cloud, as well as network transmission rates, to determine the optimal partitioning point. During the collaborative computation process, edge nodes process sensor data streams in real time and upload compressed intermediate feature tensors to the cloud. The cloud completes the subsequent inference and returns the resulting decisions to the edge, thereby achieving a balance between accuracy and latency.

3.1.2 Context-Aware Computation Migration Decision-Making Mechanism

The resource status of edge nodes and the network environment change dynamically over time, making a fixed migration strategy prone to service quality degradation. Therefore, a context-aware module is established to monitor in real time the CPU utilization, memory occupancy, remaining battery power of edge nodes, as well as the signal strength and bandwidth fluctuations of the wireless network. Deep reinforcement learning is then utilized to train a migration decision-making agent. This agent takes the current context information as the state input and outputs binary actions determining whether to migrate and which computational tasks to migrate. The reward function integrates latency penalties, energy consumption penalties, and cloud computing costs. This guides the agent to automatically reduce migration when resources are strained or the network is congested, and to fully utilize cloud computing capacity during idle periods. This mechanism endows collaborative computation with environmental adaptability, thereby maintaining stable real-time inference performance under varying operating conditions.

3.2 Online Model Updating and Parameter Adaptation Driven by Operational Data Streams

3.2.1 Incremental Model Updating and Knowledge Fusion Based on Online Learning

Industrial processes continuously generate new data, making offline retraining costly and unable to respond to changes in a timely manner. Therefore, an online learning framework is adopted. Stochastic gradient descent and its variants are utilized to perform single-step updates upon the arrival of new samples, allowing the model parameters to evolve along the direction of the instantaneous gradient. Elastic weight consolidation is introduced to identify parameters crucial to model performance on historical data. During updates, a smaller learning rate is applied to these parameters to avoid catastrophic forgetting. New knowledge is integrated incrementally into the existing model. Simultaneously, knowledge distillation is employed to transfer the behavior of the old model to the new one. This ensures that the model retains its ability to recognize historical patterns while absorbing characteristics of new operating conditions, thereby achieving the continuous accumulation of knowledge[3].

3.2.2 Concept Drift Detection in Industrial Processes and Dynamic Adjustment of Model Parameters

Factors such as equipment aging and changes in raw material batches can cause concept drift in the data distribution of industrial processes, rendering a fixed model ineffective. Therefore, an unsupervised concept drift detector is constructed to monitor changes in the distribution of latent space features. Statistical measures like Maximum Mean Discrepancy or Hotelling's statistic are employed to quantify the degree of deviation between the current data and a historical baseline distribution. When significant drift is detected, a model parameter adaptation mechanism is triggered. In the case of sudden drift, the last few layers of the model are immediately reset and quickly fine-tuned using recent data. For gradual drift, a sliding window-weighted online learning approach is adopted, assigning higher weights to newer samples. This parameter adjustment process forms a closed loop with the detection mechanism, ensuring that the model consistently aligns with the current process dynamics and maintains control accuracy.

3.3 Evolution of Agent Behavior and Emergence Regulation under Multi-Objective Constraints

3.3.1 Self-Organization of Agent Strategies Based on Multi-Objective Evolutionary Algorithms

Industrial process optimization typically involves multiple conflicting objectives, such as energy consumption, production output, and product quality. To address this, the decision policy of each agent is encoded as a parameter vector, and a multi-objective evolutionary algorithm framework is constructed to evaluate the quality of these policies based on Pareto dominance relations. During each generation of evolution, offspring strategies are generated through simulated binary crossover and polynomial mutation. Elite individuals are then selected using non-dominated sorting and crowding distance metrics. Throughout this evolutionary process, agents continuously explore the frontier of the objective space, thereby developing diverse behavioral patterns. Environmental selection pressure is introduced by dynamically weighting the priority of objectives according to the current operating conditions. This guides the population to converge towards regions that align with production requirements, ultimately achieving strategy self-organization under multi-objective constraints[4].

3.3.2 Monitoring and Feedback Regulation of Global Emergent Behaviors

Local interactions among multiple agents can potentially lead to unpredictable global emergent behaviors, such as resonance, oscillations, or resource contention. To address this, a global monitoring layer is constructed. It utilizes a graph neural network to aggregate the states and actions of individual agents, thereby learning the distribution of emergent patterns under normal operating conditions. Anomalous emergence is identified by calculating the deviation between the real-time aggregated features and this normal distribution. Once undesirable emergence is detected, a regulatory mechanism is triggered. This involves either adjusting the reward function of each agent through soft reward shaping to suppress action tendencies leading to the anomaly, or adding a global cost term within a distributed constrained optimization framework, prompting agents to consider the overall system impact during local decision-making. This feedback loop guides the macroscopic behavior towards a desired collaborative state without undermining the agents' autonomy[5].

Conclusion

This paper has constructed a theoretical framework for big data-driven agent modeling and dynamic collaborative optimization in industrial processes. At the individual agent level, by employing spatiotemporal alignment of multi-source data and deep feature extraction, a structured representation integrating graph neural networks and knowledge graphs has been established. Physical information constraints have been introduced to achieve mechanism-data hybrid modeling, and neural architecture search has been combined with reinforcement learning to generate autonomous decision-making agents. This addresses the challenge of adaptability under complex operating conditions. At the group level, a dynamic topology and lightweight protocol based on process coupling strength have been designed. Hierarchical reinforcement learning and game theory have been utilized for task decomposition and conflict resolution. Furthermore, federated learning and topology reconstruction have enhanced fault tolerance and self-healing capabilities under uncertain conditions. At the evolutionary level, an edge-cloud collaborative inference framework has been established. Online learning and concept drift detection enable parameter adaptation, while multi-objective evolutionary algorithms combined with emergence monitoring regulate collective behavior. Future research will focus on deepening the integration of causal reasoning, aligning human and machine intentions, analyzing complex emergence patterns, and conducting engineering deployment validation.

References

[1] Chen, L. (2025). Data-Driven Soft Sensor Methods for Industrial Processes: Research and Application (PhD dissertation). Beijing University of Chemical Technology.
[2] Yu, Q. X., et al. (2025). A review of data-driven fault diagnosis methods for large-scale industrial processes. Chinese Journal of Engineering, 47(04), 780-793.
[3] Mu, M. (2024). Research on Data-Driven Process Monitoring Methods for Industrial Processes (PhD dissertation). Lanzhou University of Technology.
[4] Zhao, C. H., et al. (2022). Preface to the special issue on intelligent modeling and control methods

and applications for complex industrial processes. Control Engineering of China, 29(04), 577-580.
[5] Liu, Q., et al. (2018). A perspective on data-driven operational monitoring and self-optimization for industrial processes. Acta Automatica Sinica, 44(11), 1944-1956.