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 Model of Industrial Process Modelling is being revised in light of the deep integration of Industrial Internet of Things and Big Data. The previous mechanical model is too simple and not very general; therefore, data-driven methods cannot provide causal explanations. Therefore, the framework of Big Data-driven Agent-based Modelling and Dynamic Collaborative Optimisation in Industrial Processes is put forward here. Graph Neural Networks and Knowledge Graphs are combined at the level of a single agent to organise the data and integrate information from all places and times in order to learn deep features. Constraints on the physical information have been added for the construction of the mechanism-data hybrid model, and reinforcement learning is used to develop an autonomous decision-making agent. Change the topology and lightweight protocol of collaboration according to the degree of process coupling. Hierarchical Reinforcement Learning and Game Theory are used to divide the problem and resolve conflicts; Federated Learning and Topology Reconstruction are employed to improve fault tolerance and self-healing. The Development of the Stage for the Edge-Cloud Collaborative Inference Framework has begun. Online learning and concept drift detection are used to change the parameters, and then multi-objective evolutionary algorithms with emergence monitoring control the behaviour of the group. Provide the theoretical basis and technical path for the construction of intelligent industrial systems with autonomous perception, cooperative decision-making and continuous development.*

Keywords: *Big Data-driven, Industrial Processes, Agent Modeling, Multi-agent Collaboration, Dynamic Collaborative Optimization, Knowledge Fusion, Edge-Cloud Collaboration, Evolutionary Regulation.*

Introduction

With the development of ubiquitous sensing, many areas in the industrial process have been equipped with sensors, and a large amount of data has been collected. The main problem is how to obtain knowledge from all sides and turn it into actual behaviour. Although the old mechanism model cannot be used to explain changes in operation, it is not entirely divorced from data. Industrial agents have combined mechanistic knowledge and data characteristics to develop the ability of perception and autonomous decision-making, and multi-agent networks can manage large-scale coupled processes. Research on Big Data-driven Agent Modelling and Dynamic Collaborative Optimisation of Industrial Processes. The purpose is to construct a theory of intelligence and cooperation for individuals and groups, and also to study how this theory has changed over time. The above framework has the advantages of good feature extraction and knowledge fusion, collaboration architecture design, regulation of edge-cloud cooperation, etc., and can achieve autonomous operation of industrial processes.

1. Mechanisms of Data-Driven Agent Modeling in 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

The data in the Industrial Internet of Things comes from many places and is varied. Change the difference in sensor sampling frequency, start time and data format, etc., to avoid false correlations. Therefore, a timestamp-synchronisation method based on events has been employed to align the high-frequency vibration signals, low-frequency temperature data and discrete quality indicators at the same time. A graph matching algorithm is used to correct for the spatial misalignment caused by an offset in the measurement point and ensure that the distribution of data is in line with the equipment topology. At the same time, there will be outliers and missing values in the data. A Generative Adversarial Network is employed to model the distribution of normal operating conditions, and then the missing data caused by sensor drift or loss of communication packets is reconstructed. Then all the input tensors for feature extraction will be ready and aligned.

1.1.2 Deep Feature Extraction and Implicit Representation of Process States

Based on the alignment of the data, rich features that show the general trend in the process will be extracted at this time. Industrial time series data are usually irregularly sampled in time and have different scales, so a multi-scale convolutional network with an attention mechanism is used. Collect information from all levels in a particular area and around the world, and then find some patterns. A Graph Convolutional Network is then used to spread information along the paths of material flow and learn the latent state vectors of units under the influence of their neighbours. Contrastive learning can be employed to bring samples from the same operating environment closer in the latent space and push those from different environments further apart; thus, both feature reconstruction and discrimination can be carried out at any time in the process. The resulting feature vectors are the perception interfaces of the agent, and they contain all the information of the operating conditions in the industrial process.

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 previous knowledge of the industrial process includes process flow diagrams, equipment design parameters, operating procedures, etc. First, we need to organise the knowledge in a way that is computable. A process topology graph is built, and the nodes are equipment units and the edges are the flow of material and energy. Then, a graph attention mechanism is employed to weigh the different weights of adjacent nodes and obtain the coupling strength between the upper and lower units. Knowledge graph technology is used to store the attributes of equipment, thresholds for process parameters and causal relationships in the form of triples, and thus build an industrial knowledge base with rich semantics. Graph embedding methods can be used to convert the symbolic entities and relationships in the graph into continuous vectors for joint representation of topology and semantics. This is a kind of knowledge for the agent that cannot be obtained from data.

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

A model based only on data will be overfitted, and a purely mechanistic model cannot cover all operating conditions. The two ways are combined to have some good attributes. Thus, the foundation of the fusion framework will be a Physics-Informed Neural Network (PINN). Differential equations for the conservation of mass and energy are added as soft constraints to the loss function to help the model produce results in line with the laws of physics. Then, based on the confidence in the data and the mechanistic model at different times, a Bayesian weighting method is employed to change the fusion weights dynamically. There will be some interruptions in the process, but the data-driven part will respond promptly, and the mechanistic part will provide a stable foundation. Based on the accumulation of operation data, the model will continuously adjust the parameter estimation of the mechanistic part and learn iteratively.

1.3 Autonomous Construction and Decision Logic Generation for Industrial Agents

1.3.1 Task-Oriented Adaptive Generation of Agent Structures

Agents in different industrial environments have various functional requirements, so a single Structure will be neither general-purpose nor very efficient. Neural Architecture Search can be used to automatically combine network layers in a particular operator space according to the goals and constraints of a problem to obtain an agent topology that is suitable for the current situation. At the beginning of training, a meta-learning method is used to learn general parameters from an old set of tasks, and based on this, the initial values of the current agent are set to accelerate learning. There is also a problem of construction computer power. Since the edge devices are real-time systems, network pruning and quantization compression have been employed.

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

The goal of an agent is to select the most advantageous action in a given state based on experience, and the rules for doing so are learned via reinforcement learning. The state space is the latent vector output of the feature extraction layer, and the action space is the setpoints of the key process parameters. The three parts of the reward function are energy efficiency, product quality and operational stability. DDPG is employed to train the agent in a simulation environment offline, and then the best policy is iteratively optimised. A Causal Attention Module is added to the decision-making stage to assign different weights to the various process factors. A contribution heatmap for each decision time is output by the module to show why a particular parameter was changed at that time. Record the path of decision-making and the corresponding process responses to build a causal chain of the decision logic. Therefore, the operator will know why the agent acts and build trust in the machine by providing reasons for the action in real factories[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 Coupling Strength of Industrial Processes

The Network Topology of a multi-agent system should be in line with the natural coupling relationships of the industrial process to achieve good joint decision-making. By analysing the transfer entropy and mutual information of process variables, one can quantify the degree of coupling among different production units in terms of material flow, energy exchange and information transfer. Quantification will be employed to set the connection weights of the agents. Then, a graph attention network is employed to vary the weight of the coupling strength based on all kinds of operating conditions. When the load of a particular unit is changed or the condition of a piece of equipment is modified, so too is the topology. The good coupling area is a high-bandwidth communication link, so no connection needs to be made in the weakly correlated region. Therefore, the resulting network is sparse, but there are some regions of high concentration. Based on the topology generation method above, it will be more convenient for the parties to cooperate and reduce communication costs and network congestion.

2.1.2 Light Communication Protocol and Data Exchange Mechanism for Real-time Collaboration

Given that the communication delay of industrial process control is small, the traditional protocol stack is not suitable for this. Therefore, a small Publish-Subscribe protocol based on the Data Distribution Service (DDS) architecture will be developed. According to the attributes of the industrial data, a custom topic filter and a Quality of Service (QoS) policy are used to send only the necessary information, such as changes in state or decision gradients. A short binary serialization format is employed to encode the data sent by the agents and reduce the length of the data frame. To have a common clock for all the distributed nodes and to be able to obtain neighbour information at roughly the same time, PTP is used. Therefore, both high-frequency periodic collaboration and event-driven asynchronous communication can be supported, and a stable data channel for real-time joint decision-making among many agents will be established.

2.2 Task-driven Dynamic Division of Labour and Collaboration Mechanisms for Multi-agent Systems

2.2.1 Hierarchical Decomposition of the Global Optimization Objective and Local Task Allocation

To promote distributed cooperation, the general goals of the plant should be broken down into several local goals for each agent. Therefore, the chosen structure is hierarchical reinforcement learning. The coordinator breaks down the general goal of reducing energy consumption per unit of product into several steps for the different departments according to their current operating conditions. Then the lower-level agents will change their own plans in light of this goal. Attention is paid to how much each agent contributes to the overall goal, so different weights are given to the problems, and more attention is paid to the important links in the process. A Gradient Aggregation Method with Differential Privacy is Used. In this way, one can collect feedback on the execution results of tasks from all nodes without exposing their local data, and thus the direction of local optimisation will be in line with that of the whole.

2.2.2 Multi-agent Cooperation and Conflict Resolution Strategy Based on Game Theory

If the many agents act independently, there will be competition for resources and conflicts of interest in the entire system. Therefore, multi-agent cooperation can be expressed in the form of a potential game by adding a potential function that shows the total gain of cooperation. Therefore, each agent will move in the direction of the gradient of this potential function to achieve its own goal. Then a distributed iterative best-response algorithm is employed; that is to say, each agent observes the actions of its neighbours and repeatedly updates its own strategy to reach a Nash equilibrium. To avoid disputes, the Shapley value method will be employed to divide the contributions. Therefore, it will be difficult for individual behaviour to harm the whole group, and the agents will strive to balance their own interests with the needs of the community to achieve harmonious public goods[2].

2.3 Multi-agent Collaborative Fault Tolerance and Self-healing Strategies in Uncertain Operating Conditions

2.3.1 Distributed Anomaly Detection and Isolation Based on Federated Learning

Faults in the industrial process are often localised and spread rapidly; therefore, they must be detected and dealt with in time. Therefore, a local anomaly detection model, such as a Long Short-Term Memory autoencoder, is trained for each agent in the federated learning framework, and then the residual is obtained by reconstructing the sensor time-series data. Only pass the model parameters to the next node of each agent. Since all the parameters in the consensus algorithm should change in the same direction, any parameter with a different direction is likely an outlier that has caused an error. All communication problems with the agent will be dealt with and rectified promptly. Simultaneously, neighboring nodes are notified to disregard information published by that agent. This will stop the spread of bad decisions and set up the first line of defence for fault tolerance.

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

If some agents fail or go out of sight, correct them promptly and continue to work normally. Given the current connectivity and task requirements of these agents, a distributed topology optimisation algorithm is used to add new links and form alternative cooperative subnets in the event that there is a structural defect caused by a node failure. Then, transfer learning is employed to transfer the control functions of the failed agents to nearby agents that have similar functions. The next agent will load the pre-backed-up standby model and smoothly take over the control of that loop. The whole process is carried out by a distributed consensus algorithm and does not have a central coordination unit; therefore, even if some equipment fails or there are communication problems, the system will still operate normally and gradually reach a good cooperative state.

3. Real-Time Optimisation 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 Partitioning of Inference Tasks and Edge-Cloud Collaborative Computation based on Latency Constraints

The Industrial Process Control has a small inference delay. Only cloud computing will not be able to achieve the millisecond response time requirement, and full local deployment is limited by the computational power of the edge node. Therefore, a division-of-labour plan will be established. Divide the Layers of the deep neural network model into groups. Both the computational complexity and the amount of data for the individual layers are relatively large, so the shallow feature extraction part has been placed at the edge node, and the deep abstraction inference part is offloaded to the cloud. Then, with the help of linear programming, we will minimize the total delay at the edge and cloud under the constraints of computation capacity at the edge and cloud, network transmission speed, etc., and find an optimal division point. At the same time, the edge node will collect the sensor data and compress it, and then send some features to the cloud for joint processing. The cloud will perform inference and return the result to the edge; thus, it will be both accurate and fast.

3.1.2 Context-Aware Computation Migration Decision-Making Mechanism

The resources at the edge node and the network environment are changing all the time, so the original fixed migration plan will no longer meet the service quality requirements. Therefore, a context-aware module is added to constantly monitor the CPU utilisation, free memory and remaining battery power of the edge node, changes in signal strength and bandwidth of the wireless network, etc. Deep Reinforcement Learning will be employed to train the migration decision-making agent next. The state of the agent is the current context information, and it outputs binary actions to determine whether to move and which computational tasks to move. Latency, Energy Consumption and Cloud Computing Costs are all in the penalty function. Therefore, if the resources are insufficient or there is network congestion, the agent will automatically decrease migration and use the full capacity of off-peak cloud computing. The above way can deal with changes in the environment and continue to operate normally in all kinds of situations.

3.2 Online Model Updating and Parameter Adaptation Based on Streams of Operational Data

3.2.1 Incremental Model Updating and Knowledge Fusion Based on Online Learning

As more and more new data are produced in the industrial process continuously, offline retraining is too expensive and cannot keep up with changes in time. Therefore, the above structure is used online. Stochastic Gradient Descent and its variants are used to update the model parameters only after a new sample has been obtained, and then the model parameters are moved in the direction of the current gradient. Elastic Weight Consolidation is used to find out which factors in the historical data affect the performance of the model. A small learning rate is used to update the above parameters so that there is no catastrophic forgetting. New knowledge is gradually added to the old knowledge. Simultaneously, knowledge distillation is employed to transfer the behavior of the old model to the new one. Therefore, the model will remember the previous pattern and gradually learn the new changes in the operating environment; it can continue to improve over time[3].

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

Aging equipment and changes in the batch of raw materials can cause concept drift in the data distribution of industrial processes, and thus the original model will no longer be accurate. Therefore, an unsupervised concept drift detector is used to find changes in the distribution of latent space features. The two are Maximum Mean Discrepancy and Hotelling's T-squared statistic, and they show how far the current data is from the historical base distribution. If the drift is too large, then change the model parameters. If there is a sudden drift, the last few layers of the model will be immediately reset and rapidly retrained with the new data. A sliding-window weighted online learning method is used to give more weight to the most recent samples when there is slow drift. Due to the change in the detection method, this parameter will be adjusted accordingly to keep up with the changes in the production process and maintain stable operation of the model.

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

3.3.1 Self-organization of Agent Strategies based on Multi-objective Evolutionary Algorithms

The general aim of industrial process optimisation is to reduce energy consumption and improve production efficiency and quality. Therefore, the decision policy of each agent can be represented by a parameter vector, and a framework of multi-objective evolutionary algorithms is employed to evaluate the quality of these policies based on Pareto dominance. At each generation of evolution, the offspring strategies are obtained by binary crossover and polynomial mutation. Then, from the top ones, select according to non-dominated sorting and crowding distance. At the same time, they have also been continuously exploring the limits of the goal area and showing all sorts of behaviour. Environmental Selection Pressure is used to change the weight given to different goals according to the current operating environment. It will motivate the people to move to areas with good production conditions and achieve strategy self-organisation for all goals[4].

3.3.2 Monitoring and Feedback Regulation of Global Emergent Behaviour

Many local interactions among agents can result in all kinds of unpredictable global emergent phenomena, such as resonance, oscillation and resource competition. The World-Monitoring Layer will be added at this end. A Graph Neural Network is used to collect the states and actions of the individual agents, learn the distribution of emergent patterns in normal operation, and so on. Anomaly detection is based on how far the current aggregated features are from the distribution of normal data. If it is not in good order, there will be a correction. One can change the reward function of each agent with soft reward shaping to reduce the motivation for an abnormal action, or a global cost term can be added in a distributed constrained optimisation framework so that agents consider the overall effect of their actions locally. The above feedback loop will help everyone work together in harmony and maintain the independence of the individuals[5].

Conclusion

A Foundation for Big Data-driven Agent Modelling and Dynamic Cooperative Optimisation of Industrial Processes has been laid in this paper. At the level of individual agents, Spatiotemporal alignment of multi-source data and deep feature extraction have been used to construct an organised representation that combines graph neural networks and knowledge graphs. Constraints on the physical information have been added to construct a mechanism-data hybrid model, and neural architecture search combined with reinforcement learning has been employed to develop autonomous decision-making agents. It can solve the problems in many places at the same time. A Dynamic Topology and a Lightweight Protocol Based on the Process Coupling Strength at the Group Level have been developed. Hierarchical Reinforcement Learning and Game Theory have been applied to task decomposition and conflict resolution. Federated Learning and Topology Reconstruction have increased the fault tolerance and self-healing ability of uncertain environments. Inference is being carried out in the form of edge-cloud cooperation now. Online learning and concept drift detection are used to change the parameters, and then multi-objective evolutionary algorithms with emergence monitoring are applied to regulate group behaviour. In the future, we will strengthen the integration of causal reasoning, coordinate human and machine intentions, explore all kinds of emergence patterns, and carry out 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.