A Multi-Agent Collaboration-Based Edge Data Processing and Decision-Making Mechanism for the Industrial Internet of Things

Chunqi Jiao, Xuan Liu*

Harbin Institute of Information Technology, Harbin, 150431, China
**Corresponding author:m15174507563@163.com*

Abstract: *A large amount of data is produced at the edge of the Industrial Internet of Things, and it needs to be processed in real time and intelligently decided upon; thus, the latency and bandwidth limitations of traditional centralisation have become serious problems. The problems mentioned above will be solved by the new Distributed Processing Framework based on multi-agent cooperation. As an edge node, it is an independent decision-making unit in a multi-agent system, so for the purpose of heterogeneous data synchronization, distributed consensus algorithms are employed, and event-driven protocols are used to improve communication efficiency. Deep reinforcement learning will be employed to obtain the feature representation and dynamically schedule tasks online, and due to resource constraints, it has been optimised to have a small model size. Game theory and policy gradient methods are combined to form a cooperative decision-making scheme in a dynamic environment, and thus the local strategies will be in line with the general goal. The whole stack of the technical system for data interaction and intelligent decision-making in this paper will be introduced to support the application of edge intelligence in the Industrial Internet of Things.*

Keywords: *Multi-agent cooperation, Industrial Internet of Things, Edge computing, Deep Reinforcement Learning, Distributed Decision-making, Game Theory.*

Introduction

Near the edge of the Industrial Internet of Things, many kinds of high-speed, multi-modal, real-time data are constantly being generated, and how quickly this data can be processed directly affects the operation of essential applications such as production process monitoring and predictive maintenance. The industry is very complex and changing rapidly; there are limited resources at the edge and fluctuations in network conditions, so the former centralized mode cannot achieve the required millisecond response speed and high reliability. Sinking the computation to the edge will reduce the amount of data that needs to be transmitted, but if the operation mode of the nodes is isolated, collaboration among distributed resources will not be possible. A Multi-agent System is a good way to model distributed cooperative problem-solving, and by sharing information and coordinating policies, collective intelligence can be achieved. At the same time, deep reinforcement learning has also achieved good results in solving the problem of dynamic decision-making by learning online and adapting. To solve the problems of high interaction complexity, limited resources for model deployment and maintenance, and the issue of decision inconsistency in dynamic environments caused by multi-agent integration, this paper integrates multi-agent collaboration, deep reinforcement learning and game theory. The three stages should be done in an orderly manner; that is to say, an edge computing architecture, a distributed data processing model and a dynamic decision-making optimisation mechanism need to be established to provide a theoretical foundation for the intelligent development of the Industrial Internet of Things.

1. Multi-Agent Collaborative Edge Computing Architecture and Data Interaction Mechanism

1.1 Modeling of Multi-Agent Systems for Industrial Internet of Things Edge

Edge nodes in an Industrial Internet of Things (IIoT) system have many attributes; they are distributed in various places, have different resources, and can be accessed to different extents. Therefore, the demands of the new low-latency, high-reliability system cannot be met by the old centralised model. According to the theory of multi-agent systems, distributed models for edge computing can be constructed, and each edge node with computing and communication capabilities can be considered an agent that is autonomous in perception, decision-making and execution. These agents can sense their surroundings instantly and work alone based on the small amount of information they have collected; at the same time, they can also cooperate to solve all kinds of problems in the field of edge data processing. In the Modelling stage, it is necessary to specify the scope of the function and action set for the agents, define how their internal state will be changed, set up communication links with the outside world, and thus build an official model that is in line with changes in the industrial situation.

At the bottom of the model construction are the descriptions of inter-agent relations and the degree of task coupling. Given that there are many kinds of devices and various directions for data flow in the Industrial Internet of Things, the topological structure of the multi-agent system should show the logical connections and data dependencies among the physical devices. Agents and the relationships among them can be modelled as a dynamic graph in graph theory, the attributes of the nodes are the computing and storage resource capabilities of the agents, and the weights of the edges are communication quality and data flow characteristics. Based on the above, the state space and action space of the agents have been set up, and reward functions for various industrial applications have been defined. Therefore, although the agents have their own local goals, they can still help achieve the goal of the whole system and provide a starting point for coordinated data management[1].

1.2 Heterogeneous Data Aggregation and Synchronization Mechanism Based on Distributed Consensus

The data at the edge of the Industrial Internet of Things has many kinds, is high-dimensional, and is not organised. The sampling frequencies, time-stamp references and numerical scales of the data obtained from all the sensors are not the same. Together, they will produce an erroneous report and therefore make an error. A Distributed Consensus Mechanism will be employed to solve the problem mentioned above. After a few rounds of communication and agreement among the agents, they have reached a common understanding of the data semantics and time references without a central node. At a certain time, only the collected data can be used to obtain the corresponding features and timestamps, and then, after several rounds of communication among themselves, these feature vectors are distributed to reduce uncertainty or redundancy in the data.

The construction of the synchronisation mechanism should be fast, and the communication overhead should be small. Given that the network of the industrial field has a small bandwidth and uneven channel conditions, a fully connected consensus algorithm will probably have a very large communication volume. Based on the trust-weighted consensus algorithm, the weight given to information from a neighbouring node in aggregation can be changed dynamically according to how reliable that node has been in the past. At the same time, a logical clock is used to give all units of data a common time address, and thus out-of-order data caused by transmission delay can be addressed. Therefore, based on the above, the multi-agent system can construct a common view of the status of the industrial process at the edge and provide a unified and reliable data foundation for high-level decision-making.

1.3 Communication Protocol and Load Balancing Strategy for Collaboration Processing Among Agents

The cooperation efficiency of a multi-agent system is determined by how much information is shared among the agents. However, the small network resources and changes in the topology of the Industrial Internet of Things environment are very difficult to address in the design of communication protocols. The previous round of broadcasting and central collection cannot meet the demands of real-time edge computing, so now a lightweight and flexible communication scheme needs to be designed. Based on the Design of the protocol, it is an event-driven communication model; that is to say, agents will only send information after detecting a significant change in their environment or when the confidence of their decision-making falls below a certain threshold to reduce unnecessary communication. To reduce the dimension of encoding for the transmitted state information and thus decrease the network load, compressed sensing can be used to organise the packets, and the necessary data will still be available.

The Design of the Load-Balancing Strategy should consider the computing power, available energy and the length of the current task queue for agents on both sides. In a distributed environment, if only the scheduling information available locally is used, it will be easy to have an uneven distribution of the load among nodes. A market mechanism or a game-theoretic model can be employed to have the agents themselves distribute the computational tasks, and according to the resource bids of the neighbouring nodes, the task allocation scheme will be changed dynamically. The control task needs to be dealt with promptly, so it has been given a high-priority preemptive schedule and must be completed on time; otherwise, the data will be old. Joint Optimisation of the Communication Protocol and Load Balancing will be carried out to ensure the good operation of the multi-agent system in resource-constrained edge environments and provide stable support for all kinds of industrial **applications.**

2. A Distributed Processing Model for Edge Data Based on Deep Reinforcement Learning

2.1 Online Extraction of Data Features and Methods for Multi-dimensional Information Fusion

The volume of Data Streams generated at the edge of the Industrial Internet of Things is large, non-stationary, and highly correlated. The old way of offline feature extraction with batch processing cannot meet the demands of real time. The online feature extraction mechanism needs to process and condense the raw signal data that have been collected in a short period of time, and then extract features that can describe the operating conditions of equipment or changes in the environment, etc. A sliding window and incremental learning method can be employed to track local changes in the feature buffer, and then the Fast Fourier Transform (FFT) and wavelet packet decomposition of the time-series data in the window can be used to obtain multi-dimensional features in both the time and frequency domains. A light convolutional autoencoder is employed to learn features and reduce the dimensionality of the representation for unstructured data, such as vibration signals and infrared thermography, automatically at the edge[2].

Many kinds of information fusion are carried out at both the feature and decision levels. The feature vectors of the various agents at the level of features are in different dimensions and have semantic gaps, so they need to be transformed into a common representation space by feature alignment and mapping. At each time step, different weights are given to the various dimensions of the feature, and more attention is paid to the important ones. The first judgments produced by each agent after local fusion at the decision level must be jointly approved. Anomalies caused by sensor noise or other reasons are dealt with using Bayesian inference or the theory of evidence. A layered fusion architecture can keep a large amount of data in its original form and reduce transmission and computation; thus, it will provide a good basis for intelligent decision-making in the next stage.

2.2 Dynamic Scheduling Mechanism for Distributed Processing Tasks Based on Deep Q-Networks

Given that there are many data processing tasks at the edge and their arrival times are irregular, a fixed-schedule task-scheduling method will not be practical for a long time. Deep Q-Networks are model-free reinforcement learning algorithms that learn how to choose the best plan by experiencing all sorts of situations and gaining knowledge about the distribution of problems and allocation of resources over a long time through interaction with the environment. At the level of the model, the state of each agent includes the length of its current task queue, available computational resources and load information of nearby nodes, and the set of actions is various scheduling options, such as local execution, task migration, collaboration, etc. Build a deep neural network to approximate the state-action value function, and then, given the current observation of the environment, choose an action from the schedule that maximizes the accumulated reward.

Changes in the operating conditions of the industrial site will affect the schedule, so they need to be updated in time. There is a sudden rise in data or a failure of an edge node, but the old rule-based schedule cannot be updated in time. Agents that have been trained with Deep Q-Networks can change their behaviour according to the knowledge they have acquired in the past. To deal with the problem of time variation in the training data, experience replay is employed to train the agent so that it can handle all kinds of abnormal situations based on previously recorded scheduling cases. At the same time, a dual-network structure is also used to avoid overestimation of the value function and improve the stability of the scheduling decision. After many rounds of policy iteration, a distributed scheduling method for large industrial areas has been developed to improve resource utilisation and meet the deadline requirements of tasks in a multi-agent system[3].

2.3 Model Lightweighting and Inference Acceleration Strategies Under Resource Constraints of Edge Nodes

The problem is that the deep reinforcement learning model is too computationally expensive and does not have enough resources at the edge node, so some small modifications need to be made to the model during deployment. Reduce the quantity of knowledge by employing the technology of knowledge distillation. Train a simplified student network to imitate the behaviour of a large teacher network, and then reduce the scale of the core decision-making part in Deep Q-Network so that it can be run at the edge. At the end of distillation, the aim is to obtain a good decision from the teacher model; more importantly, we hope to retain the generalisation ability of the teacher network across different regions of the state space so that the student network can also make reasonable judgments in new industrial situations.

The Construction of Inference Acceleration Methods should be based on the specific characteristics of the hardware and the structure of the algorithm. ARM architecture and FPGAs are often used in edge devices, so to reduce both the memory consumption and computation time of the convolutional and fully connected layers in the network, operator fusion and quantization have been employed to convert the 32-bit floating-point parameters to an 8-bit fixed-point format. To solve the problem of long-term dependence on processing, state caching for recurrent neural networks is used to avoid recomputing the history. Deploy the model and then add adaptation to dynamically change the inference precision based on the current load of the edge node; that is, when resources are scarce, a lighter model will be used for simple operation, and when resources are abundant, the full model will be employed to improve the accuracy of decision-making. All the above optimisation can help the large-scale deep reinforcement learning model operate normally in resource-constrained environments at the edge and provide a stable algorithm for intelligent data processing in the Industrial Internet of Things.

3. Collaborative Decision-making and Strategy Optimisation Mechanisms in Dynamic Environments

3.1 Modeling of Multi-agent Collaborative Decision-making and Solution Based on Game Theory

The environment at the edge of the Industrial Internet of Things is dynamic and uncertain, and there are many strategic dependencies in the decision-making process of several agents. The old way of optimisation cannot deal with the conflict and coordination of individual rationality and the public interest. Game theory is a general way to solve problems of distributed decision-making. Model the decision-making process of each agent as that of a game player, let the strategy space be the set of possible actions, and define a payoff function that shows how much the decision result contributes to the quality of task completion. Generally speaking, the goals of the agents in cooperative data processing have two parts: cooperation (for example, working together to ensure the stability of the system) and competition (to obtain limited computational and communication resources). Therefore, the problem will be shown more clearly in the form of a game. Potential game theory has the property of a pure-strategy Nash equilibrium and is easy to obtain with distributed learning algorithms, so it is a good way to model cooperative decision-making in edge intelligence. From the above, it can be seen that the actual demand for data processing in practice has been met.

The Game Model should have a good strategy and work in real time. The State Space of the industrial environment is very large, and since the payoff function is not known beforehand, the traditional analytical solution method cannot be used. Joint application of distributed reinforcement learning and game theory can be employed to achieve the above goal; that is, agents iteratively adjust their strategies based on the results of the previous actions they have taken and the rewards obtained, and gradually reach a Nash equilibrium. At this time, a damper will be added to avoid a large revision of the plan, and an experience pool will be used to store information about previous problems for learning by example. When there are sudden changes in the environment, agents can quickly modify their own behaviour according to the online observations of the strategies used by other agents, and thus the whole system can maintain good cooperation in all kinds of operating conditions[4].

3.2 Local Strategy Generation Based on Environmental Perception and Maintenance of Global Objective Consistency

Decision-making in a multi-agent system at the edge is based on each agent's observation of the local environment, and the overall goal (such as real-time performance, data integrity or energy efficiency of data processing) is generally the sum of the results of actions taken by all agents. The local strategy generation should be in the form of actions, and the overall goal is to be achieved indirectly. A deep neural network can learn a representation of the state for high-dimensional observations, and the output layer is set up as a strategy probability distribution so that the agent can choose an action from a continuous action space. In light of changes in the environment, to enhance the adaptability of the local strategy, an attention mechanism has been added to dynamically weigh different sensor data or the state information of neighbouring agents; thus, it is possible to improve the interpretability and selectivity of the strategy generation process.

Due to poor communication or other reasons, there will be deviations from the general goals at the local level. Add a consistency-maintenance term to the objective function of the local strategy in the learning stage and then use an auxiliary loss function or a consensus constraint. In addition to maximising the cumulative reward of the agent itself, it is also necessary to minimise the difference between its own strategy and that of neighbouring agents, or ensure that the joint action meets some global constraint conditions. With the development of the industry in recent years, agents have locally optimised a sequence of decisions for the future rolling horizon according to the concept of Model Predictive Control and have exchanged predictions with their neighbours to correct for local deviations through negotiation. To achieve the general goal and avoid concentrating at one place, a distributed negotiation method will be used.

3.3 Evaluation of the Effectiveness and Adaptation of Collaborative Decision-Making Based on Policy Gradients

All kinds of assessments of the effect of cooperative decision-making should be conducted, and the indices include the quality of task completion and the amount of resources used, etc. Policy Gradient Methods Update Policy Parameters Directly. Collect samples of interaction trajectories and calculate the gradient of cumulative rewards to solve the problem of high-dimensional continuous action spaces. Centralized training and decentralized execution can be employed in multi-agent systems to obtain global information during the training stage for the purpose of evaluating the value of joint actions, and then a more precise policy gradient signal is given to each agent. The main critic network is used in the actual construction to take the observations and actions of all agents as input and output an evaluation value for the current joint policy. Based on the above evaluation values, the actor networks of all agents will be updated. It is not stationary in training and is fully distributed at the time of use.

The Adaptive Optimisation Mechanism will be modified in light of changes in the industrial environment over time. If the efficiency of decision-making has dropped or there have been changes in the statistical properties of the environment, then the learning rate, exploration noise and network structure will be automatically adjusted to return to a good state. Meta-learning can help an agent quickly learn how to solve a new problem based on only a few examples of a similar problem and thus reduce the learning time in a new environment. Add some intrinsic motivation terms to encourage the agents to explore areas of the state space that have not been explored much and avoid getting stuck in local optima. The results of the effectiveness evaluation are sent back to the high-level task scheduling module, and then the priority and resource allocation of the tasks are adjusted; thus, a closed-loop adaptive system of perception, decision-making and optimisation has been built.

Conclusion

A Framework for Multi-agent Collaborative Data Processing and Decision-Making at the Edge in Industrial Internet of Things has been put forward in this paper. Coordination among all the nodes is carried out in the form of multi-agent modelling at the architecture level, and to solve the problems of data synchronisation and interaction efficiency, distributed consensus mechanisms and event-driven protocols have been employed. Deep Reinforcement Learning is employed to obtain features online and dynamically organise tasks at the model level, and then knowledge distillation and model quantization are carried out to meet the constraints of the edge. Game Theory and Policy Gradient methods are used together to form a dynamic cooperation mechanism at the decision-making level that considers both individual rationality and the general goals of the system. In the future, research will integrate federated learning and multi-agent systems to improve generalization under privacy constraints; explore a cloud-edge-device collaborative optimisation architecture; design scalable communication topologies for large-scale situations; introduce explainable artificial intelligence to increase the transparency of decision-making, etc.

Fund Projects

This paper is the "14th Five-Year Plan" project of the 2025 Collaborative Education Society: Research on the "Post-Course-Competition-Certificate-Innovation" Education Mechanism for Software Engineering Majors Driven by AI and Emerging Engineering Education (kt2025100118390856044)

References

[1] Tang Yue. "Design and Implementation of a Real-Time Data Processing System for the Industrial Internet of Things Based on Edge Computing." Yangtze River Information Communication, 38.07(2025): 225-227.

[2] Li Hao. Research on Joint Offloading and Caching Based on Multi-Agent and Dual Time Scales in the Industrial Internet of Things. 2025. Guilin University of Electronic Technology, MA thesis.

[3] Du Xiaojun. "Research on Optimization Strategies for Intelligent Control Systems Based on the Industrial Internet of Things." Software, 45.10(2024): 30-32.

[4] He Jingjian. Design and Implementation of an IoT Data Platform Based on Thing Models and Rule Engines. 2022. Automation Research and Design Institute of Metallurgical Industry, MA thesis.