ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Development and Design of Intelligent Bots for an Effective Smart IIoT Systems using Generative AI Models

Li Wang [1,*]

[1] School of Education, Luoyang Vocational College of Culture and Tourism, Luoyang, Henan 471000, China

SUMMARY: *This research has developed an industrial robot which is driven by generative AI for intelligent IIoT environments, and thus it evaluates its effectiveness in the work of monitoring, fault handling assistance, and scheduling support. One closed-loop structural frame is built through the integration of sensor flow data, PLC/SCADA signals, incident recording documents, one vectorized document store, one knowledge graph, digital twin statuses, tool invocation, and human-participated middle inspection. The experiment platform is constituted by two CNC units, two robot work stations, one conveyor unit, and one compressor/environment node. Through a 30-day observation time frame, the system has collected 51,840 time-ordered data records, 60 alert events, 60 upkeep records, 420 working personnel test searches, and 54 arrangement cases. The results obtained by us indicate that the bot which we put forward has obtained 93.3% total accuracy, 92.9% task finishing rate, 93.8% tool calling success rate, and 90.8% complex query accuracy, meanwhile it lets the hallucination rate be maintained at 2.9%. In the evaluation of industrial value, the system that we put forward made the time of alarm explanation reduce by 54.8%, made MTTR decrease by 35.2%, made machine stop time lower by 34.1%, made the effective maintenance adoption rate rise to 95.0%, made total working time reduce by 17.2%, and made the number of coordination work cut by 45.1% when it is compared with the rule-based basic method. These result points out that the put forward industrial robot can promote both conversation reliability and working efficiency in intelligent IIoT situation. However, the method still depends on high-quality industrial knowledge resources, and further work is needed in edge deployment efficiency, long-term robustness, and security governance.*

KEYWORDS: *Smart IIoT; Generative artificial intelligence; Industrial bot; Digital twin; decision support*

1 Introduction

With the continued evolution of the industrial Internet, edge computing, industrial robotics, and intelligent sensing, manufacturing systems are moving beyond simple device connectivity and information visualization toward a new stage in which business coordination, state awareness, and autonomous response are equally important. Intelligent IIoT is no longer confined to connecting sensors, PLCs, MES platforms, and cloud services. Instead, it requires systems to detect anomalies in time, interpret shop-floor semantics, and support decision-making under continuous data inflow, frequent operating-condition fluctuations, and dynamically changing tasks. At the same time, rule-based assisting tools which are still broadly applied in traditional

*smile7502@163.com
https://doi.org/10.65102/is20261014
Published: 30 April, 2026

industrial automation mainly work by means of pre-set logic, fixed threshold values, and restricted instruction collections. Although these methods are still able to work under standardization conditions, they many times meet difficulties when information has the need to be cooperated among different kinds of equipment, production links, and user positions. Their restrictive points become especially obvious when they process hidden semantic meanings, long-distance dependence relations, and mutually opposing working goals. Therefore, many systems can get connection with data flows, but still cannot reach the requirement of providing strong industrial cognition and service intelligence. The emergence of generative artificial intelligence within smart manufacturing mirrors the continuously increasing requirement for more powerful abilities in depiction, inference, and interaction, therefore it also offers a novel technical base for changing industrial bots from passive response tools into active cooperative units [1].

From the perspective of application demand, information on the shop floor no longer comes from a single sensor stream. Equipment condition data, warning records, repair documents, craft knowledge, production arrangement limits, staff feedback, and environment parameters together form the decision background, hence they differ greatly in time dimension, data format, and semantic fineness. The increasing attention on combining big language models with IoT does not only come from their capability for answering questions. What is more important, such models give the possibility to put different kinds of industrial data into one unified semantic space, therefore supporting more natural expression of tasks, more flexible understanding of queries, and interactive support for decisions that is nearer to the real operation logic For shop-floor operators, maintenance engineers, and scheduling managers, this capability makes it possible to interact with the system in language that is much closer to field practice when discussing equipment health, process deviation, resource occupation, or job priority, while receiving responses grounded in both context and domain knowledge.Existing researches on industrial robots further put forward that generation-type auxiliary tools which are designed to promote worker experience can make information getting smoother, cut down the load of interface changing and manual looking, and change human–machine cooperation from form-based support to talk-led helping. This provides a direct engineering motivation for the present study [2, 4].

However, the insertion of generative AI into smart IIoT should not be understood as only placing a general function talking robot into a workshop. Current research works have already indicated that believable putting into use inside smart manufacturing is restricted firstly by explainability, responsibility, and response stability, especially when model results are utilized for equipment diagnosis, maintenance suggestions, and task coordination. If there is no traceable evidence and transparent reasoning processes, therefore these systems are not likely to obtain acceptance in actual industrial environments [3]. In the same moment, the manufacturing which takes human as center under the Industry 5.0 pattern more and more emphasizes high-frequency interaction between human, machines, digital systems and physical entities. Research on embodied AI and industrial robotics likewise indicates that future manufacturing units require more than models capable of generating text; they require agents that can perceive the environment, interpret tasks, reason under operational constraints, and trigger execuTab. actions [5, 6]. When we make the comparison with this demand, numerous present industrial assistants still have three obvious shortcomings. First, their comprehension for context still stays on the thin level, hence it becomes hard to carry out joint interpretation of the present query together with real-time device condition, maintenance past record, and task goals. Second, their ability to adapt to the changing working conditions of workshop sites is still not enough, therefore the quality of output can become worse when working conditions change suddenly, data turns incomplete, or many events take place at the same time. Third,

most systems remain at the level of explanation or recommendation and have not yet established a closed-loop process that connects perception, understanding, tool invocation, execution feedback, and result verification. Change words to say another thing, what industry places really need is not a speak interactive face, but a clever service point which can do effective coordinate under actual restriction conditions.

Under this background, the current study is pushed forward by service demands in intelligent IIoT situations, and researches the exploitation and planning of an industrial bot which uses generative AI. This research puts its emphasis on three actual problems: the difficulty of making multi-source different industrial information become one, the inconsistency between traditional question answering systems and real workshop on-site conditions, and the absence of closed-loop support for fault help and task cooperation. Because industrial running and upkeep scenarios rely extremely on professional technical terms, device working theories, and working procedures, former studies have lifted the effect of big language models in upkeep-centered question answering through the bringing in of domain knowledge bases, hence it displays that the capability of industrial robots cannot get improvement without ceaseless knowledge input and structural arrangement [7]. According to this standpoint, this research further combines real-time sensor flow data, equipment record files, knowledge-base search results, and digital twin situation states to construct a generative industrial robot framework for intelligent IIoT. By this method, this system is planned not only to understand users' questions, but also to generate more pointed answers through combining present workshop condition, past matters, and tools which can be invoked. The main contribution points of this research have three aspects. First of all, one generative AI industrial robot framework for the intelligent Industrial Internet of Things has been constructed, in which industrial conversation, knowledge advancement, circumstance perception and task execution are placed into one single unified service structure. Second, we build a multi-source coordination mechanism via the combination of sensor data, equipment logs, domain knowledge and digital twin conditions, therefore we push forward the improvement of industrial background understanding and the handling of complicated tasks. Third, the method which we have proposed is verified in a realistically limited semi-physical industrial environment from the aspects of question-answering quality, fault-dealing efficiency, task finishing rate, and system delay time, hence for the purpose of assessing its actual usability and deployment value in manufacturing assistance.

2 Methods

2.1 Architecture Design of the Generative-AI-Driven Intelligent Bot for Smart IIoT

For industrial mechanical arms which are put in intelligent IIoT environments, the core key point is not merely to connect a large language model to industrial data connecting interfaces. What is required instead is a service framework capable of sustained operation around anomaly perception, equipment-state interpretation, maintenance recommendation, and production-task coordination. Information sources on the shop floor are inherently complex. They include not only continuous sensor streams such as temperature, current, and vibration, but also PLC/SCADA alarms, event logs, maintenance records, work orders, and order as well as resource states maintained in the MES. When these data sources are simply piled together in parallel, the system may receive large volumes of information without being able to form a semantic understanding consistent with actual operating conditions, let alone transform question-answering outputs into execuTab. industrial actions. For this reason, the bot

architecture proposed in this study is built on four principles: shop-floor states must be readable, industrial knowledge must be retrievable, task recommendations must be reviewable, and execution outcomes must be wriTab. back to the system. This design lets the robot not only know the abnormality of equipment, but also put its inference results into the actual work flow procedures. When we cope with the combined integration of IIoT, digital twins, and LLMs inside heterogeneous equipment surroundings, Ref. [8] points out that, in manufacturing environments where old assets exist together with intelligent devices, unified data visiting and state mapping are basic preconditions for system usability.

For the organization of the system architecture, the present study selects a representative industrial event: in the process of continuous processing, Machine Tool Unit A displays abnormal main shaft temperature increase which is accompanied by lagged cooling reaction, hence this hence disrupts the pace of downstream work activities and influences the completion of orders. In this kind of process, the changes of temperature and load inside the sensor time window can show the developing tendency of abnormal change; PLC label data and log documents record triggering trails on the control aspect and the event aspect; and working condition states in the MES reveal the way the abnormal situation spreads into the production organization. To this kind of situation, the system must first finish unified gathering of original data, then build a context packet which takes the present event as center, and finally let the bot combine knowledge and tools to produce explanations, judgments, and suggestions. By means of this design, the architecture no longer keeps a loose combination of abstract functions, but is continuously organized around one single operation thread: how a device abnormal situation is found, explained, checked, and dealt with in a closed cycle. The entire system working principle is shown in Fig. 1.

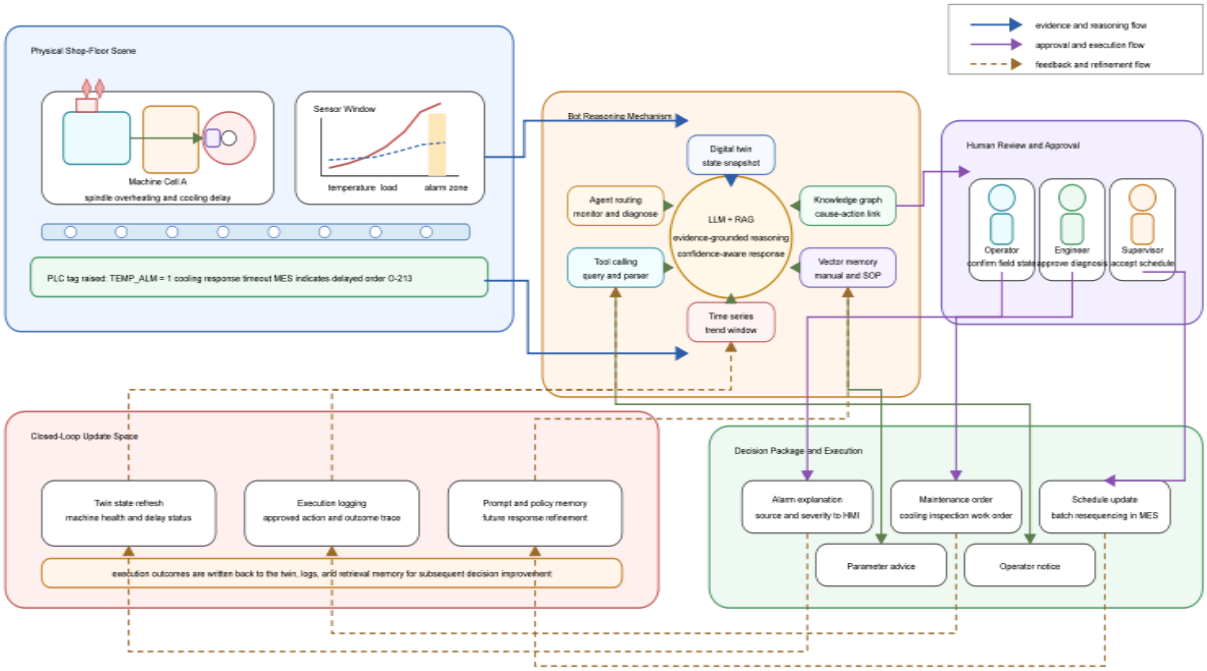


Figure 1: Diagram illustrating the scenario-based human-machine closed-loop mechanism of a generative AI-driven intelligent IIoT industrial bot.

Just as that which is shown in Fig. 1, the left part represents workshop on-site occurrence region, the middle part corresponds to robot logic deduction organization, the right part indicates manual check and execution region, and the lower part represents condition feedback and strategy update. On the left direction of Fig. 1, the scene on site of the actual working

workshop becomes the source of the original industrial proving materials. The machine tool assembly, transmission belt, and mechanical hand together compose the real working environment, in which main shaft overheat and late cooling form the starting occurrence. The sensing window has changes of temperature and load that are before and after the abnormal situation, hence it retains its time sequence continuity. The PLC label data, cooling over-time records, and MES delay messages displayed at the lower part further put control-layer conditions, event-layer tracks, and production-layer influences into the identical event background. The aim of this arrangement is to guarantee that the system input maintains multi-source proof consistency for one single fault occurrence, instead of handling data from different sources in a separated way. As to intelligent IIoT surroundings, this event-focused arrangement of inputs is thus more helpful for follow-up semantic explanation and task-level estimation.

The middle part of Fig. 1 corresponds to the core reasoning mechanism proposed in this study. This area is centered on the LLM+RAG module and is surrounded by six supporting nodes, namely digital twin state, knowledge graph, vector memory, time-series window, tool calling, and agent routing. Among them, the digital twin node provides mirrored information about current equipment health and production disturbance; the knowledge graph node represents the relations among components, faults, causes, and actions; the vector memory node retrieves maintenance manuals, standard operating procedures, and historical work-order fragments; the time-series node preserves trend evidence before and after the anomaly; the tool-calling node connects query, parsing, and work-order interfaces; and the agent-routing node dispatches requests to different subtask channels, such as monitoring interpretation, fault diagnosis, or maintenance assistance, according to task type. Under this kind of organization, the output which the LLM produces no longer relies on a single text input, but is instead generated under the common restrictions from real-time states, historical experience and structured knowledge. Ref. [9] has put forward that the combining of large language models with digital twins lets more effective use of global time information in manufacturing processes be got, thus it improves the interpretability of state explanation and interaction. This also provides an important rationale for allowing both time-window evidence and twin states to participate in the reasoning process in this study.

The right-hand part of Fig. Fig. 1 shows the flow of the human-in-the-loop checking and doing work. In this research work, the robot is not designed to bypass on-site workshop staff and directly carry out the issuance of critical operations. Instead, three human work roles are clearly kept: operation person, facility engineer, and management person. The working person mainly takes charge for making sure whether a true unusual situation exists on the working spot and what shape it shows itself. The equipment engineer carries out examination on whether the fault explanation accords with equipment mechanisms and therefore makes the decision on whether the maintenance recommendation ought to be accepted. The manager makes the determination that whether scheduling adjustments should be put into implementation according to the influences which are related to orders. After this mankind examination flow, the system can generate execuTab. results in the right-below region, which include alarm explanation, maintenance work forms, parameter suggestions, and arrangement renewals. This design preserves the strength of generative AI in understanding and organizing complex information, while maintaining clear boundaries for automated decision-making in high-risk industrial settings. In discussing integrated architectures for LLMs and digital twins, Ref. [10] likewise emphasizes that Industry 5.0-oriented systems should strengthen collaboration between humans and intelligent systems, together with review mechanisms and traceability. The human review nodes designed in this study follow exactly this principle.

The downward section of Fig. 1 is corresponding to the system's closed-loop renewal space. After the execution is finished, new equipment status and task-disturbance information are

written back to the digital twin state refreshing module. Actions which have got human approval, together with their results, are written down in the execution log, while prompt preferences and strategy preferences that can promote the quality of later responses are collected and stored in the memory module. By this means, the system does not regard the processing of one individual event as the final end point. Instead, by means of state write-back, record accumulation, and strategy revision, the next round of reasoning is based upon the updated facts of shop-floor. The closed-loop construction which is displayed in Fig. 1 basically links abnormal perception, evidence collection, inference production, human verification, and execution writing return into one continuous process. This permits the robot to carry out three functions in intelligent IIoT settings—interpretation, help, and coordination—and thus also gives a unified structure foundation for the afterward experimental measurement of question-answering quality, fault-processing efficiency, task finishing rate, and system delay time.

2.2 Experimental Setup, Industrial Scenarios, and Baseline Methods

For evaluating the practical application ability of the put forward industrial robot in intelligent IIoT environments, experiments were arranged on a small-scale experiment intelligent manufacturing production line. This platform is composed by two CNC process-cutting units, two robot work positions, one conveying belt unit, and one air compressor and environment observation node. All devices have been connected to a unified data channel via edge gateways, and they have kept working for 30 days, with a sampling interval which is 5 min. This platform maintains the characteristics of discrete type manufacturing, including cooperation among multiple devices and the coexistence of different source data, meanwhile it also includes common working procedures such as equipment state monitoring, fault processing, and task adjustment. Therefore it is very suiTab. for becoming the validation environment of this study. As for the demands of predictive maintenance under IIoT situations for state drawing and continual watching, distributed digital twin frameworks have been proven to be fit for supporting this kind of experiment arrangement, and the platform design that this work uses is in accordance with that thought direction [11].

With regard to the arrangement of the system, this research uses a structure which is composed of edge-end data collection, private cloud searching and inference, and application-end operation and record keeping. The edge level takes charge of protocol adaptation, time stamp alignment, and buffer-type forwarding. The cloud stratum carries time-sequence queries, log analysis, document getting, knowledge graph visiting, and big model reasoning calculation. The application layer has received the output results from alarm explanation, maintenance suggestion, work order creation and arrangement adjustment. This kind of disposition not only guarantees sTab. data access in the process of experiment, but also causes it to be possible to observe how retrieval augmentation and task invocation produce influences on latency and response quality. At the same moment, the security and coordination merits of the edge–private cloud structure in industrial situations supply a feasible technical route for the carrying out of this research [12]. Our experiment platform, three working task flows, and baseline contrast plan are all shown in this Fig. 2.

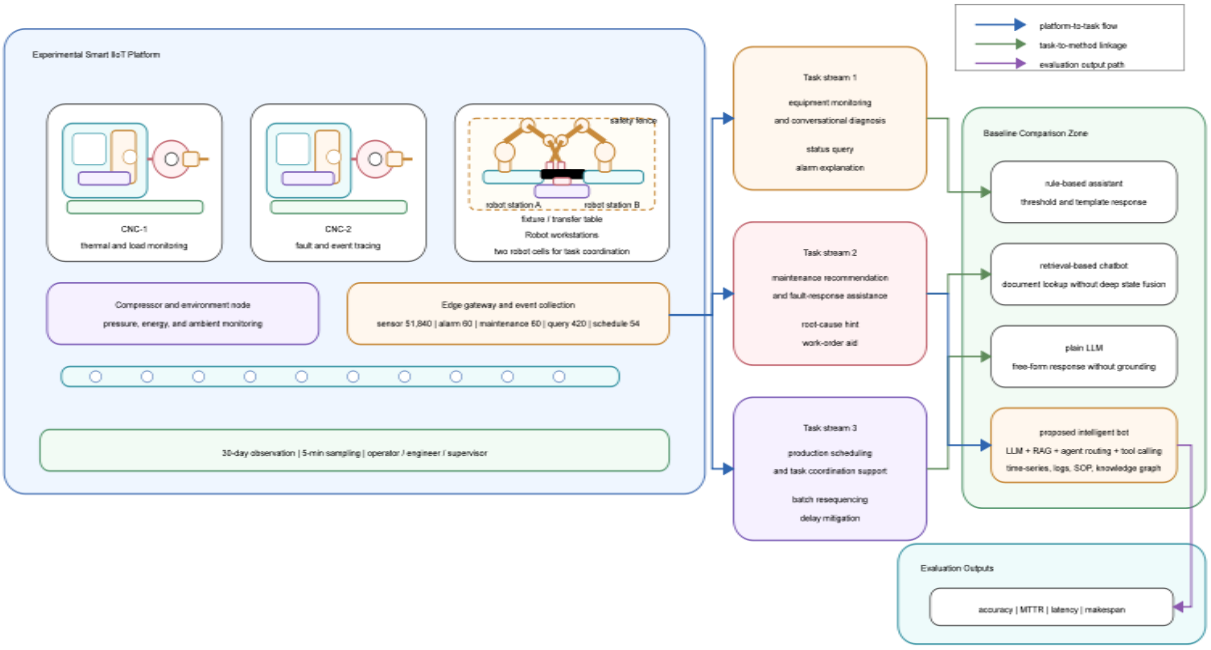


Figure 2: Diagram illustrating the experimental platform, three types of task flows and the baseline comparison mechanism.

Such as it is shown in Fig. 2, on the left hand side it shows the small-size intelligent IIoT experiment production line, which contains two CNC units, two robot work stations, one convey unit, one air compressor and environment node, and also the edge gateway and event collection module. The middle section puts forward three task flows, which correspond separately to equipment monitoring and conversation-based diagnosis, maintenance suggestion and fault-response help, and production arrangement and task-coordination support. The right-hand side displays four comparison baseline methods, hence the lower portion summarizes the integrated evaluation results. On the basis of this platform, in this research, there was a total of 51840 time-series records, 60 alarm events, 60 maintenance logs, 420 operator test instructions, 54 scheduling scenarios got organized. The query-style samples were split into training, verification, and test sets according to a proportion of 60%/20%/20%, therefore obtaining 252/84/84 samples, respectively. The samples for event handling were cut into 70%/15%/15%, and the scheduling scenes were separated into 36/9/9. For the work of equipment maintenance and fault diagnosis, equipment entities, component information, fault reasons, processing operations, and work-order terms were further arranged into fine-grained knowledge relationships, so that the stability of fault semantic expression and context retrieval can be promoted. This situation keeps consistent with the result which gotten from researches that study the LLM-aided building of fine-grained fault knowledge graphs [13].

The three task situations each undertake different verification goals. The first sort, equipment watch-keeping and dialog-based judgment, is utilized to assess the system's ability in condition inquiry, abnormal explanation, and process follow-up. The second category, maintenance recommendation and fault-response assistance, is designed to assess fault-clue localization, maintenance recommendation generation, and work-order support. The third category, production scheduling and task-coordination support, is used to examine batch rescheduling, delay mitigation, and resource coordination. To demonstrate the practical gains of the proposed method in industrial contexts, four comparison methods were included in the experiments: a rule-based dashboard assistant, a conventional retrieval-based chatbot, a plain LLM without industrial grounding, and the proposed intelligent bot. To speak more specifically, the rule-based assistant depends on fixed threshold values and pre-set rules; the chatbot which

is based on retrieval only carry out document getting and template-type answer making; the pure large language model generates answers straight from given prompts, having no access to industrial situation states or enhancement of information retrieval; and the intelligent robot which we put forward has the integration of time-series windows, logs, SOPs, a knowledge graph, tool calling, and agent routing. To CNC diagnostic work especially, the cooperation between knowledge graph and language model can greatly enhance relation restrictions and mutual explanation ability in the process of diagnosis. Because of this aspect, the knowledge-strengthened setup is kept in the second task type category[14].

With respect to prompt organization and agent scheduling, the system first identifies the task according to user role, query intent, and target equipment, and then dispatches the request to the monitoring-interpretation agent, maintenance-assistance agent, or scheduling-support agent. Evidence is subsequently extracted from the time-series window, event logs, maintenance documents, and knowledge graph, after which SQL queries, event parsing, work-order generation, and scheduling interfaces are invoked as required by the task. The experimental data, task partitioning, and baseline configurations are listed in Tab. 1.

Table 1: Experimental data, task segmentation and baseline method configurations.

Project	Configuration Details
Experimental Platform	2 CNC Machines, 2 Robot Workstations, 1 Set of Conveyor Units, 1 Compressor, and Environmental Monitoring Node
Operating Cycle	30 days
Sampling Interval	5 minutes
Time-Series Data	51,840 entries
Alarm Events	60 entries
Maintenance Logs	60 entries
Operator Test Instructions	420 entries
Scheduling Scenarios	54 groups
User Roles	Operator, Engineer, Supervisor
Task Scenario 1	Equipment Monitoring and Conversational Diagnosis
Task Scenario 2	Maintenance Recommendation and Fault-Response Assistance
Task Scenario 3	Production Scheduling and Task Coordination Support
Query Sample Division	252 / 84 / 84
Event Sample Division	70% / 15% / 15%
Scheduling Scenario Division	36 / 9 / 9
Baseline 1	Rule-Based Dashboard Assistant
Baseline 2	Conventional Retrieval-Based Chatbot
Baseline 3	Plain LLM Without Industrial Grounding
Baseline 4	Proposed Intelligent Bot

Tab. 1 summarises the platform composition, sample size, number of scenarios, dataset segmentation and performance boundaries of the four baseline groups, to facilitate a consistent comparison in the subsequent results section. Given that predictive maintenance systems are highly sensitive to response times and distributed processing capabilities, this paper records both event-level processing latency and recovery effectiveness in maintenance and fault resolution tasks, thereby providing a basis for subsequent analyses of MTTR, response latency and task completion rates [15].

2.3 Performance Metrics and Statistical Evaluation

To ensure that the evaluation results reflect model performance, system operational quality and industrial application benefits simultaneously, this paper establishes an indicator system comprising three dimensions: Bot intelligence metrics, IIoT system metrics and industrial value metrics. Statistical analysis is conducted using significance tests, confidence intervals, ablation experiments and robustness tests. The 420 test commands are primarily used for the statistical analysis of bot intelligence metrics; the 60 alarm and maintenance events are used to analyse system response and repair effectiveness; and the 54 scheduling scenarios are used to analyse coordination efficiency and task completion quality. All four sets of methods were run under the same test set, with the same role permissions and task constraints.

With respect to Bot intelligence measurement indexes, this paper gives definitions to intent recognition accuracy, response correctness, hallucination rate, task completion rate, and tool invocation success rate. These norm standards are employed to evaluate the correctness of work distribution, the degree that answers correspond with artificial marking results, the proportion of content that does not match on-site conditions or knowledge proofs, the ability that complete operable results can be outputted, and the effective callback rate for SQL searches, log analysis, work sheet interfaces, and arrangement interfaces. The correctness of responses and the rate of task completion are evaluated by means of two-person annotation and then re-checking. With regard to IIoT system measuring indexes, this article gives definitions to end-to-end delay time, P95 response time, fault check accuracy, fault check recall rate, F1 value, warning response time, and packet/calculation extra cost to describe response time effectiveness, tail latency, abnormal check quality, warning processing speed, and extra communication and calculation extra cost. With respect to industrial value measuring norms, this paper gives definitions to MTTR, downtime reduction, schedule efficiency, operator workload score and maintenance decision consistency for the description of mean time to repair, the decrease of downtime, arrangement efficiency, operator work burden and the consistency degree between system proposals and engineer decisions.

Table 2: Performance evaluation metrics and statistical analysis plan.

Metric Level	Metric Name	Meaning	Statistical Granularity	Statistical Method
Bot Intelligence	Intent Recognition Accuracy	Proportion of correctly classified request intents	Query Level	Mean, 95% CI, ANOVA
Bot Intelligence	Response Correctness	Degree of agreement between response and labeled answer	Query Level	Mean, t-test / Mann–Whitney U
Bot Intelligence	Hallucination Rate	Proportion of outputs inconsistent with actual states or knowledge evidence	Query Level	Proportion, 95% CI
Bot Intelligence	Task Completion Rate	Proportion of providing complete execuTab. results	Query Level	Proportion, ANOVA
Bot Intelligence	Tool Invocation Success Rate	Proportion of valid returns from tool invocations	Query Level	Proportion, 95% CI
IIoT System	End-to-End Latency	Duration of a single request for a complete response	Query Level	Mean, P95, Significance Test
IIoT System	P95 Response Time	Tail response characteristics under high load	Query Level	Quantile Comparison
IIoT System	Precision / Recall / F1	Quality of anomaly event recognition	Event Level	Mean, 95% CI
IIoT System	Alarm Response Time	Time from alarm occurrence to provision of mitigation advice	Event Level	Mean, t-test / U-test
IIoT System	Packet / Compute Overhead	Additional communication and computational resource consumption	Request Level	Mean, ANOVA
Industrial Value	MTTR	Mean Time to Repair	Event Level	Mean, 95% CI
Industrial Value	Downtime Reduction	Proportion of reduction in downtime	Event Level	Proportion Comparison
Industrial Value	Schedule Efficiency	Efficiency of task coordination and completion after scheduling	Scenario Level	Mean, ANOVA
Industrial Value	Operator Workload Score	Score representing operator workload	Role Level	Mean, U-test
Industrial Value	Maintenance Decision Consistency	Degree of agreement between system suggestions and engineer decisions	Event Level	Proportion, 95% CI
Additional Experiments	Ablation Study	Assessing the contribution of individual modules	Multiple Granularity	Module-wise Comparison
Additional Experiments	Robustness Under Missing/Noisy Data	Testing stability under missing and noisy conditions	Multiple Granularity	10%/20%/30% Perturbation Comparison

The experiment data and statistics analysis method are given in Tab. 2. The continuous variables we had were firstly carried out normality testing by us; when data satisfy the normal distribution, they are presented as mean ± standard deviation, and group comparisons are carried out by utilizing t-tests; When data do not accord with normal distribution, they are shown as median and interquartile range, and the Mann–Whitney U test is utilized by us. We have carried out analysis on the overall differences among the four baseline methods by means of one-way analysis of variance (ANOVA), and 95% confidence intervals are given in the report. Furthermore, the ablation research and the robustness experiment have been done by us to discuss the contributions of key components and the system's stability in the situations of 10%, 20% and 30% data loss or noise interference, respectively.

3 Results and Discussion

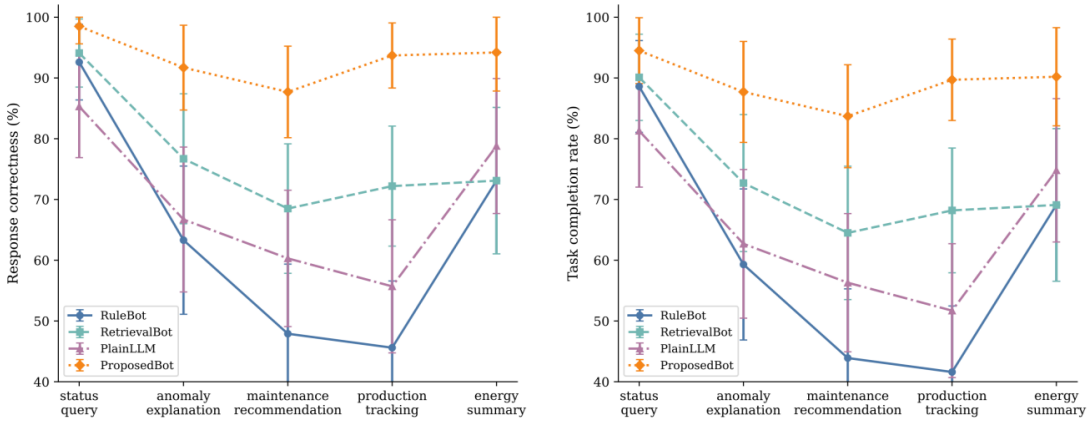
3.1 Network structure identification performance under synthetic benchmarks

The overall results of the different methods in dialogue understanding and decision support tasks are shown in Tab. 3.

Table 3: Comparison of the performance of different methods in dialogue understanding and decision support tasks.

Method	Overall accuracy	Task completion rate	Tool success rate	Hallucination rate	Mean latency (s)	P95 latency (s)	Complex query accuracy
RuleBot	66.2%	64.3%	54.8%	0.2%	1.40	2.30	46.7%
RetrievalBot	77.9%	72.4%	77.1%	5.7%	2.55	3.44	70.4%
PlainLLM	70.7%	69.0%	44.3%	12.1%	3.59	4.45	57.9%
ProposedBot	93.3%	92.9%	93.8%	2.9%	3.06	4.02	90.8%

In Tab. 3, the ProposedBot that we put forward has obtained 93.3%, 92.9%, 93.8% and 90.8% separately on the four core targets of total accuracy, task finish rate, tool success ratio and complex query accuracy, with a corresponding hallucination ratio that is 2.9%, hence it overall surpasses the other three methods. RuleBot shows a superiority on average delay time, but its correct rate for complex queries is just 46.7%; PlainLLM has got strong ability of language generation, but its rate of hallucination achieves 12.1%; Compared with PlainLLM, RetrievalBot has certain promotion on fact consistency, but its ability that finishes tasks in complicated industry scenes still has limitation. On the whole, the bringing in of time windows, event recording documents, SOP obtaining, knowledge graphics and tool calling has brought about obvious enhancements to this system's answer correct rate, task finishing and output dependability. The response correct degree and task finish rates among different task kinds are displayed in Fig. 3.

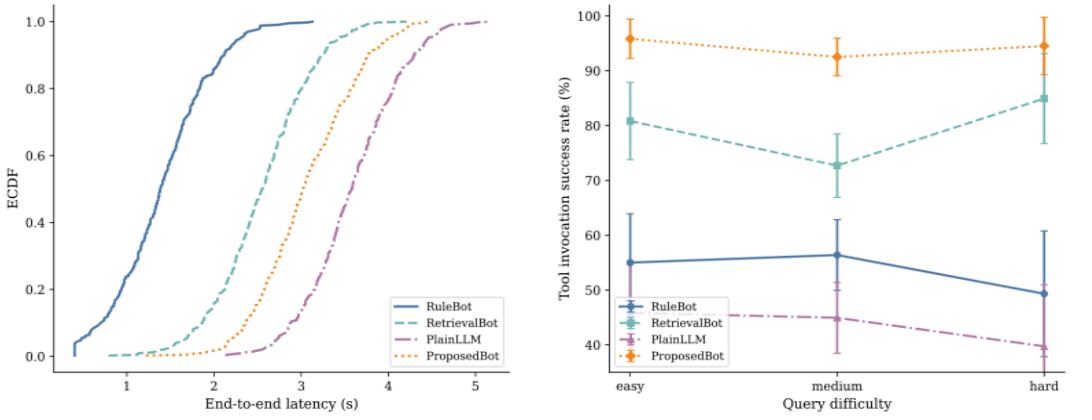


(a) Response correctness with 95% confidence intervals

(b) Task completion rate with 95% confidence intervals

Figure 3: Performance evaluation of dialogue understanding and task completion.

Fig. 3(a) shows the response correctness of each method across six task categories, along with their 95% confidence intervals, whilst Fig. 3(b) shows the corresponding task completion rates. In Fig. 3, the performance gap between the four methods is relatively limited for status queries, whereas the differences are more pronounced across the three task categories of anomaly explanation, maintenance recommendation and production tracking. Taking production tracking as an example, ProposedBot achieves a response correctness of 93.7%, whilst RetrievalBot, PlainLLM and RuleBot achieve 72.2%, 55.7% and 45.6% respectively; as for the suggestions of maintenance, therefore, ProposedBot also holds the topmost rate of task completion. These outcome show that when a inquiry contains real-time condition, past records, procedure restrictions and following actions at the same time, therefore neither rule matching nor simple document searching alone can produce a whole, consistent and operable answer. Jeon et al. [16] noted in their research on machine tool monitoring dialogue that the ability to explain equipment status is significantly enhanced when real-time data retrieval is integrated with generative models; Colabianchi et al. [17] evaluation of digital assistants for assembly manufacturing also indicates that industrial settings place greater emphasis on a system's ability to reduce the burden of retrieval, verification and decision-making. The task-specific differences illustrated in Fig. 3 are consistent with the conclusions of the aforementioned studies. The distribution of response latency and tool invocation results are shown in Fig. 4.



(a) End-to-end latency ECDF

(b) Tool invocation success rate with 95% confidence intervals

Figure 4: Validation of response latency and tool invocation capabilities.

Fig. 4(a) uses an ECDF to illustrate the end-to-end latency distributions of different methods, whilst Fig. 4(b) presents the tool invocation success rates and their 95% confidence intervals for different query difficulties. In Fig. 4(a), the RuleBot curve is generally furthest to the left, indicating the fastest response speed; although ProposedBot incorporates retrieval, proxy scheduling and tool invocation, its latency distribution remains significantly better than that of PlainLLM, with a P95 latency of 4.02 s, which is within an accepTab. range. Fig. 4(b) shows that as query difficulty increases from ‘easy' to ‘hard', the tool invocation success rates of all four methods decline; however, ProposedBot consistently maintains the highest level, with a relatively smaller decrease. This indicates that industrial context enhancement and tool chain design do not cause significant system instability, but rather improve information integration capabilities and execution consistency under complex tasks. Huang et al. [18] noted in their research on flexible manufacturing scheduling that the effectiveness of large models in complex tasks depends on the coordinated organisation of state, resource, and action information; The authors Wang and other researchers [19] also in their research about vision-language cooperation for human-centred manufacturing have emphasised that the value of industrial agents is constructed on the continuous combination of understanding, execution, and feedback. The outcomes within Fig. 4 have confirmed this cognition.

A research on mistake cases further finds out the performance boundaries of various kinds of methods. The main problem of RuleBot is that its answers frequently stay on the level of thresholds and labels, hence it causes difficulty in connecting MES states with task influences; the deviation of PlainLLM is mainly displayed in offering generalized recommendations which do not accord with log records; Although RetrievalBot has the ability to obtain related document segments, it still cannot appropriately combine the connections among the current situation, past occurrences, and following operations. The mistakes that exist in the ProposedBot are on the whole gathered in maintenance situations in which log clues are not complete, or similar faults exist together; These usually show themselves as changes in the order of mistake possible items, though in most situations it still gives usable judgment guiding rules and next step suggestions. According to what is shown in Tab. 3, Fig. 3 and Fig. 4, the method that this paper puts forward exhibits good whole performance on the three aspects of dialogue comprehension, mission fulfillment and decision assistance.

3.2 Fault Diagnosis, Maintenance Assistance, and Scheduling Effectiveness

Across 60 event-level handling samples and 54 scheduling scenarios, the four methods demonstrated clear differences in fault identification, maintenance assistance and task coordination; the results are shown in Tab. 4. To ensure consistency in measurement, fault identification capability is characterised by the response correctness of the anomaly explanation task; the effectiveness of maintenance recommendations is characterised by the proportion of incidents resolved through closed-loop resolution following the implementation of recommendations; and scheduling support capability is measured by makespan and the number of coordination instances.

Table 4: Comparison of the industrial value of fault handling and scheduling support.

Metric	Rule-based baseline	RAG-based baseline	Plain LLM baseline	ProposedBot
Fault diagnosis accuracy (%)	63.3	76.7	66.7	91.7
Alarm explanation time (min)	24.11 ± 2.64	17.40 ± 3.19	21.06 ± 3.01	10.90 ± 2.80
MTTR (min)	192.98 ± 15.09	164.68 ± 17.21	176.72 ± 16.19	125.11 ± 17.41
Downtime (min)	203.34 ± 21.16	173.83 ± 18.75	187.83 ± 20.60	133.95 ± 19.78
Effective adoption rate (%)	73.3	75.0	73.3	95.0
Makespan (min)	607.02 ± 28.36	552.82 ± 27.14	575.35 ± 26.87	502.37 ± 27.91
Coordination count (times)	7.36 ± 0.89	5.42 ± 0.83	6.08 ± 0.86	4.04 ± 0.89

In Tab. 4, the fault check exact rate of the ProposedBot reaches 91.7 percent, which represents a promotion of 28.4 percentage points compared to the rule-based baseline (63.3%), hence it has an increment of 15.0 and 25.0 percentage points over the retrieval-enhanced baseline and the general-use big-model baseline, separately. The corresponding alarm explanation time was cut down to 10.90 ± 2.80 minutes, which is a 54.8% decrease when compared with the rule-based baseline's 24.11 ± 2.64 minutes, and also has decreases of 37.4% and 48.2% when compared with the retrieval-enhanced baseline and the general large-scale model baseline, in that order. This shows that when state evidence, log traces and task interfaces are put together, the system can obviously cut down waiting and confirmation periods in the explanation and response stages after anomaly detection is done. The combining of control logic limits and generating models lets the system become more fit for execution-directed industrial works, which is consistent with the constraint-based generating method for industrial control that is stressed by MetaIndux-PLC [20].

Results from maintenance assistance further prove that the ProposedBot which we put forward not only gives remediation suggestions in a quicker speed but also brings more sTab. closed-loop repairing results. Its MTTR is situated at 125.11±17.41 min, which represents the decreasing magnitudes of 35.2%, 24.0% and 29.2% when making comparison with the rule-based baseline, retrieval-enhanced baseline and general large-scale model baseline, respectively; the average stopping time is 133.95 plus or minus 19.78 minutes, which represents decreases of 34.1%, 22.9% and 28.7% when compared with the three baseline groups, respectively. From the perspective of the rate of effective acceptance of suggestions, ProposedBot reaches 95.0%, therefore the other three groups lie between 73.3% and 75.0%. This difference shows that the suggestions given by the system are not only ‘for reference', but thus can directly finish the maintenance work cycle in most of the accident cases [21]. Investigation of AI-driven digital twins inside the Industrial Internet of Things (IIoT) holds that when state reflectors are put together with past data, the system has a better ability to produce decision-making value which puts stress on recovery and reduction of machine stop time; the promotes in MTTR and work halt that are showed in this paper confirm this opinion.

The outcome about arrangement assistance also displays obvious industrial advantages. The time consumption of ProposedBot was 502.37 ± 27.91 minutes, which gives a 17.2% decrease when compared with the rule-based scheduling baseline of 607.02 ± 28.36 minutes, and gives a 9.1% and 12.7% decrease when compared with the retrieval-enhanced and general generative baselines, respectively. At the same time, the average quantity of coordination occurrences dropped from 7.36 ± 0.89 in the rule-based baseline method to 4.04 ± 0.89, which stands for a reduction of 45.1%; when we compare with the retrieval-enhanced and general generative baseline methods, the reductions are 25.5% and 33.6% respectively. In the practical applications of real world, this metric can be considered as a substitute for the strength of human involvement, hence it shows that the system can cut down repetitive coordination and the

second-time adjustments when it deals with equipment abnormalities which overflow to cause the breaking of scheduling. Fig. 5 gives out the relative performance outlines of the different industry value measuring indicators.

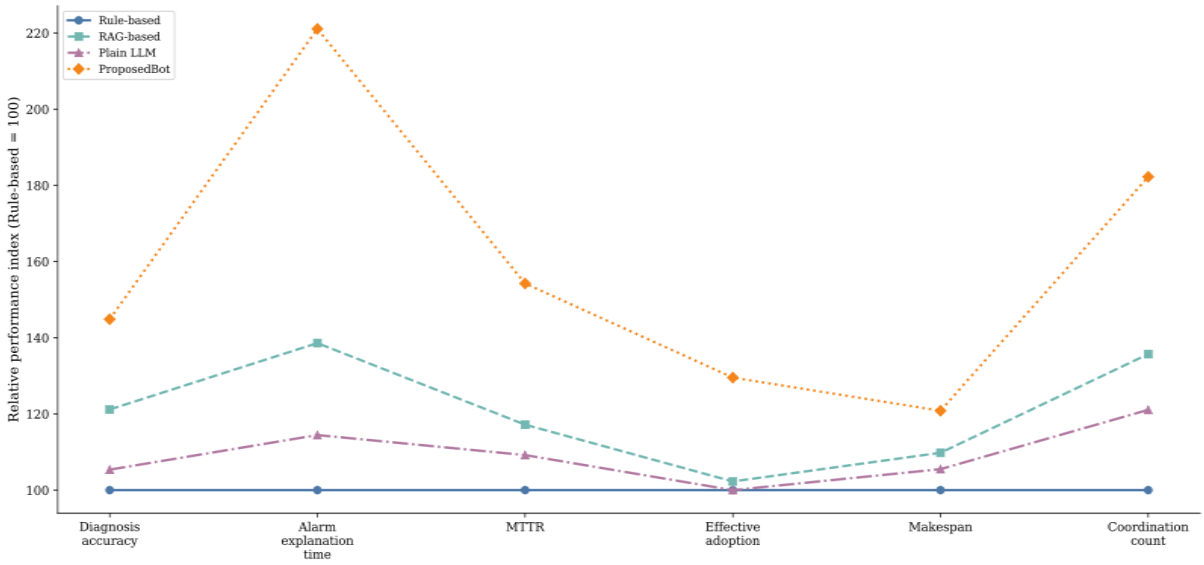


Figure 5: Relative performance profile of the industrial value of fault handling and dispatch support.

In Fig. 5, after we build a relative performance index which takes baseline as 100, the ProposedBot has shown the biggest promotion in three targets: alarm explanation time, MTTR and count of coordination cases; it also keeps a steady superiority in fault identification correctness, effective use rate of suggestions and total working time. The worth of strengthened big language models in complicated IoT interaction chains lies mainly in promoting the uniformity of interface invocations and task advancement [22]; the overall characteristic graph displayed in Fig. 5 is in accordance with this cognition.

From the results, the retrieval-enhanced baseline outperforms the rule-based method on most metrics, indicating that document-level evidence is already capable of improving certain industrial responses; the general-purpose large model outperforms the rule-based baseline in scheduling tasks, but remains unsTab. in fault identification and closed-loop maintenance; the advantages of ProposedBot are primarily reflected in its ability to organise real-time status, event logs, knowledge relationships and execution interfaces into a continuous decision-making chain, thereby achieving better overall performance across the three stages of fault diagnosis, maintenance handling and scheduling support.

3.3 Robustness, Explainability, and Deployment Implications in Smart IIoT

The usability of this system not merely relies on the accuracy of each individual question-answering reply, but also on whether core modules really operate as they are designed, whether the system can keep sTab. when data is not complete, and whether it can produce continuous industrial profits after it is put into use. Fig.s 6 to 8 give further outcomes from three angles: module contribution degrees, anti-interference ability for incomplete data, and advantages when putting into use. The result of module ablation is exhibited in Fig. 6.

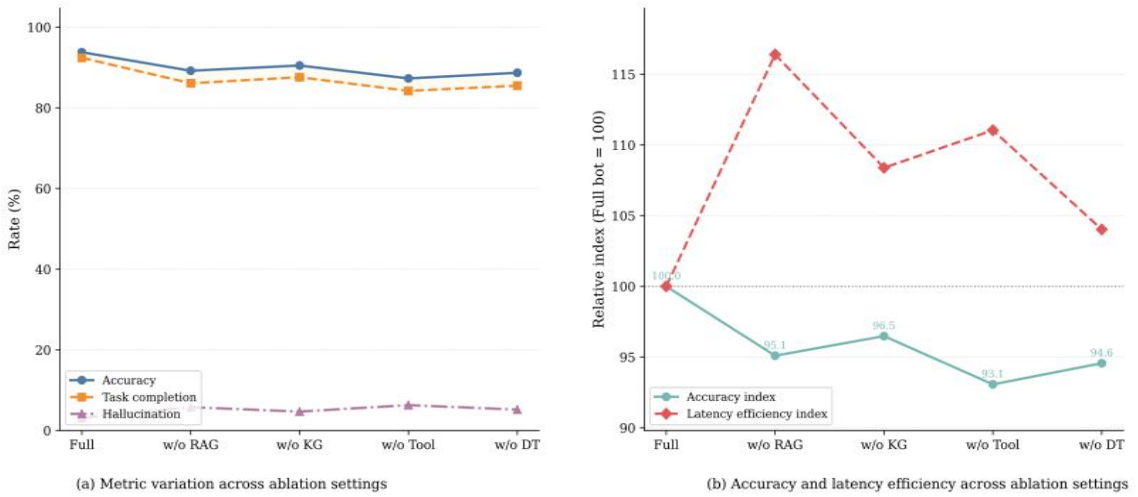


Figure 6: Module ablation results and accuracy–latency trade-off validation plot.

In the Fig. 6(a), the accurate rate, task finished proportion and illusion occurrence rate of the full system are 93.8%, 92.4% and 3.1% separately. After removing RAG, the accuracy drops to 89.2%, the task completion rate falls to 86.1%, and the hallucination rate rises to 5.8%; after removing the knowledge graph, accuracy and task completion rate dropped to 90.5% and 87.6% respectively; after removing tool calling, accuracy further decreased to 87.3%, task completion rate fell to 84.2%, whilst the hallucination rate rose to 6.3%, representing the worst performance among all ablation settings; After we get rid of digital-twin feedback, accuracy and task completion rate are respectively 88.7% and 85.5 percent. These outcomes show that fetch strengthening and tool calling are the most crucial elements for system working effect; the first one directly has influence on the strength of evidence restriction, hence the second one decides whether the responding can be connected to the question, analysis and doing sequence; the knowledge graph and digital-twin feedback, from another perspective, have greater function on expressing fault connections and keeping the continuousness of states. Fig. 6(b) further illustrates that whilst some ablation settings resulted in lower P95 latency, accuracy simultaneously declined, thus failing to achieve a superior overall trade-off. The P95 latency of the complete system is 3.62 s, which falls within an accepTab. range whilst maintaining the highest accuracy, indicating that the architecture proposed in this paper strikes a sound balance between precision and timeliness. Robustness results are shown in Fig. 7.

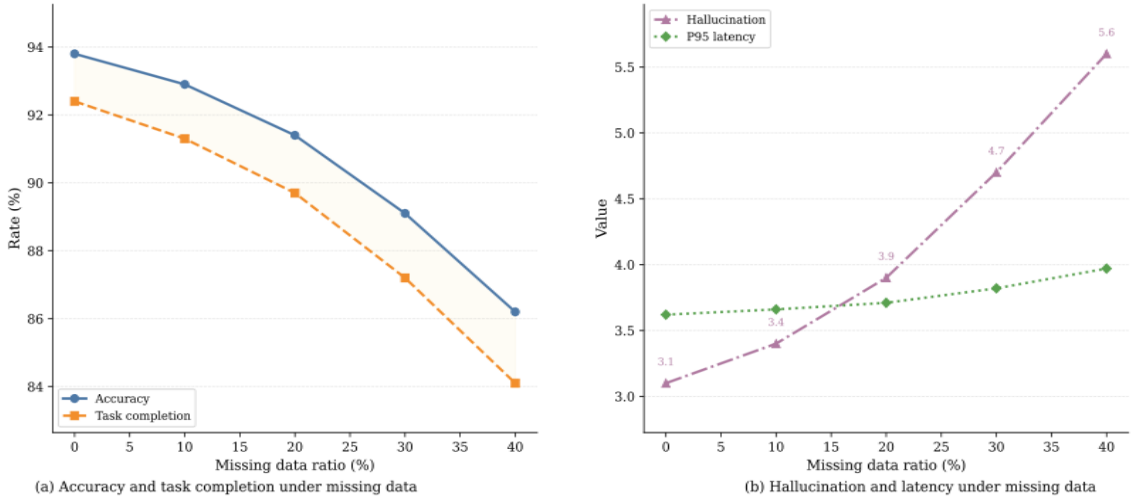


Figure 7: Robustness verification under missing data conditions.

In Fig. 7(a), as the missing data ratio increases from 0% to 40%, accuracy decreases from 93.8% to 86.2%, and the task completion rate decreases from 92.4% to 84.1%. Under missing data conditions of 10% and 20%, accuracy remains at 92.9% and 91.4%, whilst the task completion rate remains at 91.3% and 89.7%, indicating that the system exhibits good tolerance to moderate levels of missing information; When the missing data ratio exceeds 30%, the decline in performance accelerates, indicating that the complementary relationship between multi-source evidence has been significantly weakened. Fig. 7(b) shows the changes in the hallucination rate and P95 latency. As the missing data ratio increases, the hallucination rate rises from 3.1% to 5.6%, and the P95 latency increases from 3.62 s to 3.97 s. Although the increases are not drastic, they still indicate that incomplete evidence simultaneously affects both output reliability and inference efficiency. It can thus be seen that the method proposed in this paper maintains relatively sTab. performance under general data loss conditions; however, if there is a prolonged high rate of data loss in the field, it is still necessary to combine sensor redundancy, log compensation and state backwriting mechanisms to further enhance the system's resilience. The results of the industrial deployment are shown in Fig. 8.

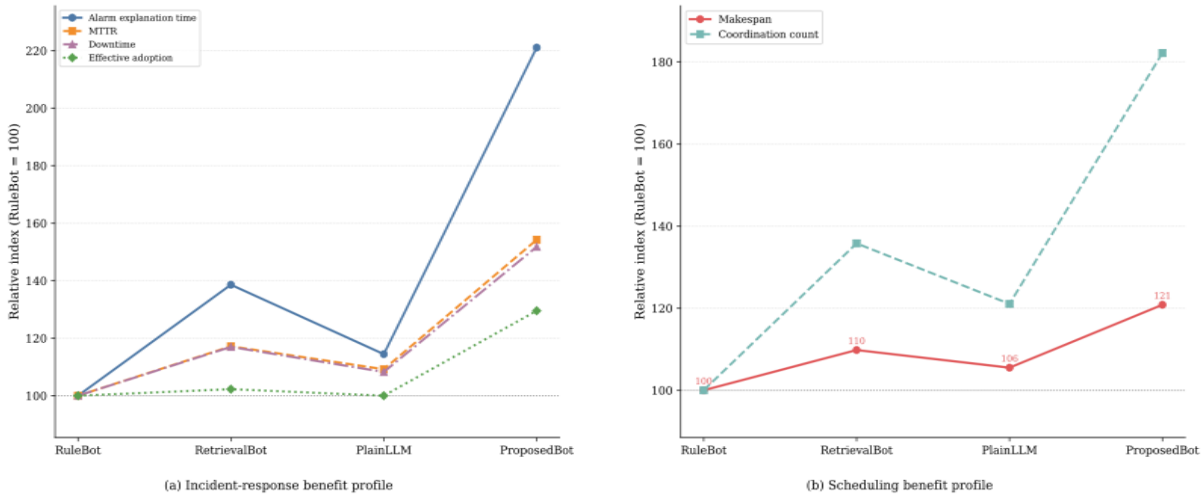


Figure 8: Benefit profile for intelligent IIoT deployment.

Fig. 8(a) compares the differences between the four groups of methods in terms of alarm resolution time, MTTR, downtime and the rate of effective adoption of recommendations, using relative indices. In Fig. 8, the ProposedBot that we put forward shows that the benefits on incident-response metrics are most concentrated, with alarm solution time cut down by 54.8% when compared with the rule-based baseline, MTTR cut down by 35.2%, downtime cut down by 34.1%, and the effective use rate of recommendations goes up by 29.6%. These promotions prove that the system's real value does not only lie in "how many answers are right", but it is in its capability to give acceptable, operable, production-recovering suggestions as fast as possible after fault is found. Fig. 8(b) gives a relative contrast on makespan and coordination number. The advantages that ProposedBot has on scheduling performance are likewise obvious, thus the makespan is decreased by 17.2% hence the coordination count is decreased by 45.1% when compared with the rule-based baseline. This result reveals that when equipment abnormal conditions are spread to the production organization layer, the system can effectively cut down redundant coordination work and secondary adjustment works, therefore holding local equipment problems inside a smaller scope [23-25]. Through combining Fig.s 6 to 8, we can discover that the interpretability of the method we put forward mainly comes from the decomposability of module contributions, its robustness can be proved by its capability that it

keeps performance under medium data loss situations, and its deployment value is focused on the synchronous promotion of fault solution efficiency and task cooperation efficiency. For the intelligent IIoT application situations, putting the sTab. running of RAG and tool calling in the first place is of the greatest importance; Knowledge graph and digital twin feedback are more appropriate for further promoting state continuity and relational expression abilities after the system has entered the continuous operation stage. From the angle of engineering putting into practice, on the condition that the working place has basic data gaining quality and interface usable status, the method which is put forward in this paper has already shown good potential of being arranged and used.

4 Conclusion

(1) This present paper puts forward a closed-loop framework for generative AI industry mechanical devices which are designed for intelligent IIoT, it integrates time-sequence data, event records, knowledge graphs, digital twin conditions and tool calling mechanisms into one combined system. It also brings in a man-machine cooperation flow that includes operators, engineers and managers, therefore it makes possible the comprehensive disposal of equipment supervision, malfunction help and mission support.

(2) Experimental outcomes prove that the system shows good effectiveness in equipment monitoring, fault help and scheduling support situation. The ProposedBot which we put forward has obtained a total accuracy of 93.3%, a task finish rate of 92.9% and a tool successful rate of 93.8%; The explanation time of alarms got a reduction of 54.8%, MTTR got a reduction of 35.2%, and makespan got a reduction of 17.2%, which shows this method not only promotes the quality of dialogue understanding but also promotes response efficiency and the collaboration inside industrial working processes.

(3) This present dissertation still possesses certain restrictive shortcomings. This system is on the heavy dependence of high-quality industrial knowledge bases, standard logs and steady interfaces; On the conditions of edge deployment, the problems of computational costs, long-time running stability and safety management still need further assessments. In the future, the research work can place emphasis on continuous knowledge renewing, light-weight edge reasoning and safety-strengthened deployment.

About the Author

Li Wang was born in Henan, China,in 1975. She studied in Zhengzhou University and received her bachelor's degree in 1999. From 2014 to 2016, she studied in Henan Normal University and received her Master's degree in 2016. Currently, she works in Luoyang Vocational College of Culture and Tourism. She has published multiple papers. Her research interests are included Artificial Intelligence and Machine Learning.

References

[1] Kusiak, A. (2025). Generative artificial intelligence in smart manufacturing. Journal of Intelligent Manufacturing, 36(1), 1-3.

[2] Zong, M., Hekmati, A., Guastalla, M., et al. (2025). Integrating large language models with internet of things: applications. Discover Internet of Things, 5, 2.

[3] Abhilash, M., Luo, X., Liu, Q., et al. (2024). Towards next-gen smart manufacturing systems: the explainability revolution. npj Advanced Manufacturing, 1, 8.

[4] Kiangala, K. S., & Wang, Z. (2025). A generative pre-trained transformer industrial bot to improve operators' working experience in a small Industry 5.0 factory. The International Journal of Advanced Manufacturing Technology, 136, 3525-3541.

[5] Xu, J., Sun, Q., Han, Q. L., et al. (2025). When embodied AI meets Industry 5.0: Human-centered smart manufacturing. IEEE/CAA Journal of Automatica Sinica, 12(3), 485-501.

[6] Fan, H., Liu, X., Fuh, J. Y. H., et al. (2025). Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics. Journal of Intelligent Manufacturing, 36, 1141-1157.

[7] Wang, H., & Li, Y. F. (2023). Large language model empowered by domain-specific knowledge base for industrial equipment operation and maintenance. In 2023 5th International Conference on System Reliability and Safety Engineering (SRSE) (pp. 474-479).

[8] Gautam, A., Aryal, M. R., Deshpande, S., et al. (2025). IIoT-enabled digital twin for legacy and smart factory machines with LLM integration. Journal of Manufacturing Systems, 80, 511-523.

[9] Sun, Y., Zhang, Q., Bao, J., et al. (2024). Empowering digital twins with large language models for global temporal feature learning. Journal of Manufacturing Systems, 74, 83-99.

[10] Chen, C., Zhao, K., Leng, J., et al. (2025). Integrating large language model and digital twins in the context of Industry 5.0: Framework, challenges and opportunities. Robotics and Computer-Integrated Manufacturing, 94, 102982.

[11] Abdullahi, I., Longo, S., Samie, M., et al. (2024). Towards a distributed digital twin framework for predictive maintenance in Industrial Internet of Things (IIoT). Sensors, 24(8), 2663.

[12] Al-Hawawreh, M., & Hossain, M. S. (2024). Digital twin-driven secured edge-private cloud Industrial Internet of Things (IIoT) framework. Journal of Network and Computer Applications, 226, 103888.

[13] Liao, X., Chen, C., Wang, Z., et al. (2025). Large language model assisted fine-grained knowledge graph construction for robotic fault diagnosis. Advanced Engineering Informatics, 65, 103134.

[14] Liu, Y., Zhou, Y., Liu, Y., et al. (2025). Intelligent fault diagnosis for CNC through the integration of large language models and domain knowledge graphs. Engineering, 53, 311-322.

[15] Alabadi, M., Habbal, A., Guizani, M., et al. (2024). An innovative decentralized and distributed deep learning framework for predictive maintenance in the Industrial Internet of Things. IEEE Internet of Things Journal, 11(11), 20271-20286.

[16] Jeon, J., Sim, Y., Lee, H., et al. (2025). ChatCNC: Conversational machine monitoring via large language model and real-time data retrieval augmented generation. Journal of Manufacturing Systems, 79, 504-514.

[17] Colabianchi, S., Costantino, F., Sabetta, N., et al. (2024). Assessment of a large language model based digital intelligent assistant in assembly manufacturing. Computers in Industry, 162, 104129.

[18] Huang, J., Teng, Y., Liu, Q., et al. (2025). Leveraging large language models for efficient scheduling in human-robot collaborative flexible manufacturing systems. npj Advanced Manufacturing, 2, 47.

[19] Wang, T., Fan, J., Zheng, P., et al. (2024). An LLM-based vision and language cobot navigation approach for human-centric smart manufacturing. Journal of Manufacturing Systems, 75, 299-305.

[20] Ren, L., Wang, H., Dong, J., et al. (2025). MetaIndux-PLC: A control logic-guided LLM for PLC code generation in industrial control systems. Applied Soft Computing, 184, 113673.

[21] Bolbotinović, Ž., Milić, S. D., Janda, Ž., et al. (2025). AI-powered digital twin in the industrial IoT. International Journal of Electrical Power & Energy Systems, 167, 110656.

[22] Wang, J., Yu, L., Luo, X., et al. (2024). LLMIF: Augmented large language model for fuzzing IoT devices. In 2024 IEEE Symposium on Security and Privacy (SP) (pp. 881-896).

[23] Wen, S., Li, F., Zhuang, W., et al. (2025). Leveraging large language models for human-machine collaborative troubleshooting of complex industrial equipment faults. Advanced Engineering Informatics, 65, 103235.

[24] Xiao, C., Liu, X., Wulamu, A., et al. (2025). KG-SR-LLM: Knowledge-guided semantic representation and large language model framework for cross-domain bearing fault diagnosis. Sensors, 25(18), 5758.

[25] Palma, G., Cecchi, G., Rizzo, A., et al. (2025). Large language models for predictive maintenance in the leather tanning industry: Multimodal anomaly detection in compressors. Electronics, 14(10), 2061.