



OPEN

The construction of an integrated cloud network digital intelligence platform for rail transit based on artificial intelligence

Keke Wang¹, Xin Zhou² & Jianbo Guan³✉

This study presents the design and validation of a closed-loop control platform for rail transit construction. The platform integrates multi-source data, enables real-time prediction, and supports AI-driven scheduling, with strategy execution and feedback implemented via digital twins. A three-layer architecture is constructed, comprising edge sensing, cloud computing, and intelligent interaction. The system incorporates data fusion middleware, an AI decision engine, and a 3D digital twins module. The operational workflow follows the perception–fusion–prediction/optimization–execution/feedback loop: edge devices collect on-site status, cloud middleware integrates and serves the data, the AI engine performs prediction and scheduling optimization, and the digital twins layer validates strategies and dispatches execution to the front end. At the data modeling level, a Transformer-Encoder-based multimodal temporal fusion model is designed, and graph attention networks are employed for heterogeneous structure modeling. Apache Kafka and Flink handle streaming data to achieve high-frequency, low-latency processing. The intelligent analysis layer integrates a Spatio-Temporal Graph Convolutional Network for passenger flow and construction period prediction, a Shifted Window Transformer for image recognition, and the Proximal Policy Optimization (PPO) algorithm for task scheduling optimization. Field tests in an urban rail construction project show that the platform maintains 91.6% accuracy in passenger flow prediction under high-concurrency conditions and achieves 98.2% accuracy in image recognition. PPO-based scheduling reduces average task completion time by 27.4%. The system sustains an average response latency of 280 ms, peak throughput of 27,000 messages per second, and over 95% closed-loop execution success rate. These results indicate that the platform meets its design targets in prediction accuracy, response latency, and scheduling efficiency under real-world conditions, providing a foundation for informatization and intelligent upgrading in urban rail transit.

Keywords Rail transit, Data fusion, Intelligent scheduling, Digital twins platform

Urban rail transit construction has increasingly faced challenges that traditional management approaches struggle to address. As next-generation rail transit infrastructure evolves from “information integration” toward “intelligent closed-loop” systems, practical engineering applications still encounter difficulties^{1,2}. Cloud-edge-end architectures, which combine cloud and edge computing technologies, provide a foundation for efficient data acquisition, processing, and inference in rail transit systems^{3,4}. The integration of AI models with visualization engines advances digital twins toward real-time feedback and collaborative decision-making, while data-driven scheduling mechanisms support integrated operational control⁵.

To ensure conceptual clarity, this study defines key technical terms and their integration within the platform. The Cloud-Edge-Endpoint Collaboration architecture consists of three layers: centralized computation and global optimization in the cloud, real-time processing and response at edge nodes, and perception collection and preliminary analysis on terminal devices. The Cloud-Network-Data-Intelligence Integrated Platform is a multi-layer system built on a unified communication network, a data fusion middleware, and an intelligent analysis engine, connecting the entire workflow of data perception, processing, decision-making, and feedback. In this study, digital twins refer to a three-dimensional virtual model combining Building Information Modeling (BIM) and Geographic Information System (GIS), maintaining second-level bidirectional synchronization with

¹Ningbo Rail Transit Group Co., Ltd, Ningbo, China. ²Popsmart Technology (Zhejiang) Co., Ltd, Ningbo, China.
³Ningbo Card Technology Co., Ltd, Ningbo, China. ✉email: guanjb@nbmetro.com

the physical site for situational awareness, simulation, and strategy validation. The AI engine integrates deep learning, Graph Neural Network (GNN), and reinforcement learning models, supporting predictive analysis, pattern recognition, and scheduling optimization. The closed-loop feedback framework comprises perception, analysis, decision, and execution layers, ensuring that newly generated data after strategy execution flows back into the platform to refine models and decisions continuously. The platform links these components through the data fusion middleware and standardized interfaces, enabling cross-domain data sharing, cross-modal analysis, and cross-scenario scheduling, supporting high-concurrency prediction, rapid response, and dynamic optimization.

Despite these advances, three critical issues persist in practical deployment. First, data silos arise from heterogeneous devices and non-uniform interfaces, limiting information sharing⁶. Second, delayed decision feedback occurs because traditional management approaches primarily respond after incidents, preventing real-time prediction and coordinated action⁷. Third, weak system coupling results from the absence of unified platforms for multi-source data fusion, AI modeling, and three-dimensional (3D) visualization⁸. These problems become particularly significant in multi-disciplinary, high-risk, high-concurrency rail construction environments. This study addresses three key aspects. Data fusion unifies heterogeneous sources—including video, Internet of Things (IoT) sensors, BIM/GIS, and construction logs—into standardized, consistently accessible data services. Real-time prediction forecasts key indicators such as passenger flow and construction schedule deviations while maintaining second-level inference latency. Closed-loop scheduling integrates reinforcement learning-based optimization with the digital twins execution layer, enabling strategy validation and front-end execution, with execution success rate serving as a measurable metric.

The implemented platform includes: (i) a data fusion middleware based on Kafka/Flink and graph/relational databases; (ii) passenger flow and construction period prediction models using Spatio-Temporal Graph Convolutional Network (ST-GCN) and construction image recognition models using Shifted Window Transformer (Swin Transformer); (iii) a scheduling optimization engine based on Proximal Policy Optimization (PPO) integrated with the BIM + GIS digital twins execution layer.

The main contributions of this study are: (1) the design and implementation of a closed-loop architecture with quantifiable interfaces and latency indicators connecting perception, prediction, optimization, and execution feedback; (2) the selection and deployment of engineering-ready AI models tailored to task requirements, achieving high throughput and low latency; (3) performance validation in a real urban rail construction project, demonstrating that design targets—prediction accuracy, response latency, task completion time reduction, and execution success rate—are consistently achieved.

Literature review

Cloud computing and edge computing technologies have been applied in rail transit systems to enhance data processing and operational efficiency^{9,10}. Atanasov et al.¹¹ developed a cloud-based disaster recovery platform for metro signaling systems using OpenStack, achieving high-availability data backup and anomaly recovery. Lifelo et al.¹² proposed lightweight data processing modules deployed on edge nodes to support video-based early warning and remote monitoring in tunnel construction sites, which reduced bandwidth pressure and response latency. Zhukabayeva et al.¹³ demonstrated that a cloud-edge collaborative architecture improved real-time construction status collection. These studies collectively indicate that cloud-edge coordination enhances data processing efficiency and provides conditions for deploying intelligent decision systems.

AI has been gradually implemented in multiple key scenarios of rail transit^{14,15}. In scheduling optimization, Souza et al.¹⁶ established a deep reinforcement learning-based vehicle scheduling strategy, improving operational efficiency during metro congestion. For predictive maintenance, Li et al.¹⁷ employed convolutional neural networks and recurrent neural networks to perform multi-dimensional fault detection on track systems, achieving earlier warning times. In passenger flow analysis, Fan et al.¹⁸ used a Transformer architecture to predict pedestrian flow at metro transfer stations, enhancing prediction accuracy. These applications demonstrate the advantages of AI algorithms in handling complex temporal sequences, high-frequency dynamics, and multi-source data in rail transit applications.

Domestic digital and intelligent platforms generally focus on information integration. For example, Beijing Subway's Traffic Information Management System manages basic information and visualizes static data. In contrast, foreign platforms emphasize digital twins and system simulation. Siemens' Railigent platform integrates BIM and real-time operational data for asset health monitoring and remote operations^{19,20}. However, existing studies often lack closed-loop feedback for the full construction process, and decentralized issues persist in data fusion and AI modeling.

Recent research has increasingly focused on reinforcement learning and graph-based methods for rail transit optimization. In scheduling, Multi-Agent Deep Reinforcement Learning (MADRL) has been applied to coordinate departures and route management. Zhang et al.²¹ proposed an MADRL framework that improved timetable robustness and rolling stock utilization under congestion in simulated urban rail scenarios. For passenger flow prediction, methods combining spatio-temporal convolution/graph networks with Transformers have advanced performance for complex metro spatio-temporal dependencies. Dai et al.²² validated a decomposition-enhanced dual-Transformer model on urban rail passenger flow data, showing improved generalization for short-term forecasts. In the digital twins domain, a Digital Twins–Cyber-Physical Systems framework has been proposed for railway infrastructure, enabling real-time monitoring and predictive maintenance while integrating BIM/GIS with live data streams; however, deep coupling with online AI decision engines remains a challenge²³.

Most existing studies focus on local functionalities rather than integrated platform architectures with complete feedback mechanisms. This study addresses these gaps by combining AI models, digital twins, and real-time data channels to construct a closed-loop intelligent platform for rail transit construction. The study aims to improve intelligent development in both construction and operation/maintenance. The innovations of

this study include: (i) unified deployment of ST-GCN, Swin Transformer, and PPO-based reinforcement learning strategies within a Cloud-Edge-Endpoint collaborative architecture, forming a closed loop of prediction-recognition-scheduling; (ii) integration of a real-time feedback mechanism in the digital twins system, enabling bidirectional synchronization between virtual simulation and the physical site for online strategy evaluation and rapid feedback; (iii) provision of a cross-domain data fusion middleware to unify multi-modal heterogeneous data—including video, sensors, BIM, and construction logs—addressing current gaps in AI, digital twins, and cloud platform integration for real-time operation and task coordination.

Research model

To clarify the system composition and module interactions, this section presents the overall system architecture (Fig. 1), with data flows and control flows annotated in the figure caption. Sects. “[Platform Architecture Design](#)”–“[Digital Twins-driven Feedback Optimization System](#)” describe the core modules shown in the figure, including the data fusion middleware, intelligent decision engine, and digital twins platform, providing implementation details for each component.

The architecture in Fig. 1 comprises three layers: the edge perception and acquisition layer, the cloud data fusion and computation layer, and the intelligent application and interaction layer. Solid blue arrows indicate data flows from edge devices—including cameras, sensors, ticket gates, and industrial equipment—to cloud processing and applications. Dashed red arrows represent control flows from AI and decision modules to edge execution units. The system integrates cloud and edge middleware, AI inference, and dual rendering to support real-time perception, analysis, decision-making, and feedback.

Platform architecture design

An integrated “cloud-network-data-intelligence” platform is designed for rail transit based on a cloud-edge-end collaborative architecture. The platform consists of three hierarchical layers: the edge sensing and acquisition layer, the cloud data fusion and computation layer, and the intelligent application and interaction layer. These layers enable real-time processing of heterogeneous data, analytical decision-making, and visual presentation, while supporting three core capability modules that facilitate a closed-loop intelligent decision-making process^{24,25}.

The data fusion middleware establishes multi-source heterogeneous data channels and forms a unified data service system for construction management. It integrates multimodal data, including video surveillance, IoT sensors, and BIM models. Transformer-Encoder and GNN technologies are applied for feature alignment and

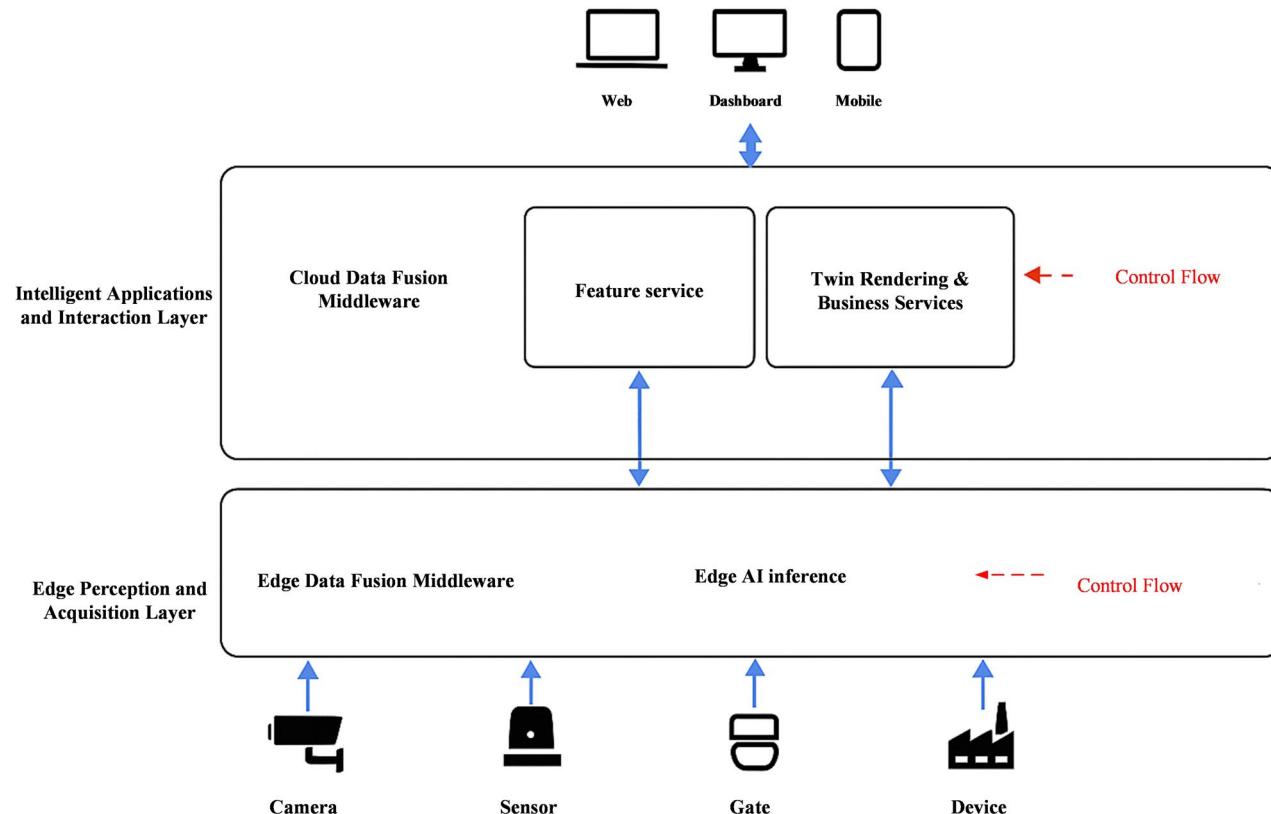


Fig. 1. Overall architecture of the cloud-network-data-intelligence integrated platform for rail transit.

correlation modeling. Fused data are stored in distributed and graph databases, providing a foundation for business modules and AI model invocation²⁶.

The intelligent decision engine incorporates AI algorithms for real-time analytical predictions, supporting applications such as construction progress forecasting, equipment risk identification, and scheduling optimization. The engine provides modular deployment and model management, supporting plug-and-play operation. Representative models include ST-GCN for passenger flow prediction, Swin Transformer for image recognition, and PPO for construction scheduling optimization²⁷.

The digital twins simulation platform utilizes BIM and GIS as base maps to construct multi-level spatial models. By integrating engineering status data with AI predictions, it generates dynamic visual management views. The platform supports equipment status linkage and construction process replay, providing high-precision modeling and interactive rendering through a hybrid Unity+Unreal engine. Macro-to-micro perspective switching enables an environment for informed decision-making.

To clarify system functionality, the three-layer architecture is further decomposed into components, interfaces, and data/control flows, specifying inputs, outputs, and integration protocols.

(1) Edge perception and acquisition layer.

Hardware/Terminals: industrial cameras, Unmanned Aerial Vehicle video terminals, personnel authentication ticket gates, vibration/displacement/temperature-humidity/dust sensors, and tunnel boring machine (TBM) or large equipment controllers.

Access Protocols:

- Sensors: message queuing telemetry transport (MQTT, QoS 1/2), modbus-transmission control protocol/remote terminal unit (TCP/RTU), open platform communications unified architecture (OPC-UA)
- Video: real-time streaming protocol (RTSP), GB28181, Real-time messaging protocol (RTMP)
- Local cache and resume transmission: structured query language lite, chrony synchronization; retry intervals for resume transmission: 5–30 s

Data Formats: JavaScript Object Notation (JSON)/Avro for structured data, H.264/H.265 for video streams, Parquet for batch uploads from the edge.

Security: Transport Layer Security (TLS); device-level authentication using appKey/secret + timestamp + signature; whitelisting and rate control.

(2) Cloud data fusion and computation layer.

Data Fusion Middleware:

- Message queue: Kafka (multi-partitioned, with topics organized by line/section/data domain)
- Stream processing: Flink (exactly-once semantics, checkpoint/savepoint, watermark synchronization)
- Storage: time series databases (e.g., InfluxDB, TDengine), graph databases (Neo4j), object storage service (MinIO), relational databases (DM/PG, PostgreSQL)
- Data governance: schema registry, data quality monitoring (missing/abnormal thresholds), data lineage and metadata management
- Service interfaces: representational state transfer, gRPC remote procedure call, GraphQL; unified authentication via OAuth 2.0 and JSON Web Token

Intelligent decision engine:

- Online inference services: ST-GCN for passenger flow and construction progress prediction, Swin Transformer for image recognition, PPO for scheduling optimization
- Feature services: feature views and sliding-window caching (60 s / 5 min / 30 min)
- Model management: version control, A/B testing, staged release, and rollback strategies

Digital twins simulation platform:

- Rendering engine: hybrid Unity + Unreal rendering
- BIM/GIS integration: loading industry foundation classes, city geography markup language, and oblique photogrammetry models
- Linkage: subscribing to fusion middleware/AI results to drive hierarchical 3D scene traversal and status highlighting, with reverse dispatch of control strategies

(3) Intelligent application and interaction layer.

Terminals: Web dashboards, operations and maintenance back-end, mobile clients (NBLink/Narrowband Link, enterprise WeChat H5).

Interfaces: unified Application Programming Interface (API) Gateway and Single Sign-On.

Audit and Traceability: operation logs, model inference logs, and traceable workflow tracking.

Use Case A: Hard Hat Violation Detection and Closed-Loop Handling.

- (1) Acquisition: edge cameras stream via RTSP, edge nodes extract frames (3–5 frames per second (FPS)) and optionally perform lightweight detection;
- (2) Aggregation: video frame indices and metadata are sent to Kafka (topic: vision.events), while original streams are stored in object storage;
- (3) Inference: online inference using Swin Transformer (< 120 ms/frame) outputs bounding boxes, categories, and confidence scores;
- (4) Fusion: Flink rule engine correlates “no hard hat” alerts with personnel authentication and area access permissions;
- (5) Presentation: digital twins scene highlights the construction point and replays 10 s of evidence;
- (6) Handling: corrective work orders are generated and pushed to mobile clients; if necessary, edge-level voice alerts or access restrictions are issued;
- (7) Closed-Loop: after verification, work orders are automatically archived, and cases are added to the knowledge base.

End-to-End Latency Targets: $P50 < 300$ ms, $P95 < 600$ ms.

Use Case B: Passenger Flow and Construction Schedule Prediction–Driven Scheduling Optimization.

- (1) Data Window: the fusion middleware continuously constructs 60 s sliding-window features;
- (2) Prediction: ST-GCN outputs $T + 5/15/60$ min confidence intervals;
- (3) Decision: PPO generates scheduling actions (task assignment, equipment relocation, sequence adjustment) based on predicted load and resource constraints;
- (4) Simulation: digital twins sandbox executes multiple strategies to select the cost-optimal plan;
- (5) Execution: scheduling instructions are dispatched to the field and collaborative systems (progress, materials, safety);
- (6) Evaluation: execution results are fed back to update strategy value, forming daily/weekly evaluation dashboards.

Resilience and disaster recovery: Kafka multi-replication, Flink checkpoints (interval 1–5 min), cross-Availability Zone disaster recovery.

Backpressure and rate control: message backlog alerts, adaptive rate limiting, cold/hot partition rebalancing.

Observability: Prometheus + Grafana for throughput, latency, failure rates; Elasticsearch–Logstash–Kibana for logs and traceability.

Security and compliance: fine-grained Role-Based Access Control (RBAC), data masking/anonymization, key rotation, API auditing.

Deployment process: pre-production stress testing, staged release, rollback thresholds, and circuit-breaking strategies.

Multi-source data fusion mechanism

To unify the modeling of time series data during track construction, the platform employs a Transformer-Encoder architecture²⁸. The multimodal input data can be represented as Eq. (1):

$$X = \left[x_1^{(s)}, x_2^{(e)}, x_3^{(v)}, \dots, x_t^{(d)} \right] \in \mathbb{R}^{T \times D} \quad (1)$$

In Eq. (1), $x_t^{(s)}$, $x_t^{(e)}$, and $x_t^{(v)}$ denote the features of sensors, equipment, and video at time t , respectively, T refers to the length of the time window, and D denotes the feature dimension. For instance, with a sampling rate of 1 Hz and a 60-s window ($T=60$), X contains synchronized snapshots of all modalities over the past minute, such as equipment current or video frame density.

After embedding, the sequence representation is computed via multi-head self-attention:

$$\text{Attention}(Q, K, V) = \text{soft max} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2)$$

In Eq. (2), Q , K , and V are the query, key, and value vectors, and d_k stands for the key dimension. The output sequence is then projected into a unified semantic space through a linear layer for downstream prediction and recognition tasks²⁹. In cases of sudden construction noise spikes, the model assigns higher weights to recent features (e.g., equipment current peaks 10–20 s prior), enhancing the timeliness of abnormal event detection and correlation.

To model the complex relationships among personnel, equipment, sites, and the environment, the platform constructs a heterogeneous graph and applies a graph attention network (GAT). The heterogeneous graph is defined as: $G = (V, E)$, $V = \{v_i\}$, $E = \{(v_i, v_j, r)\}$. Here, nodes v_i represent entities such as personnel, equipment, and monitoring points; edges (v_i, v_j, r) refer to interactions or business associations; r indicates the type of interaction.

In GAT, the representation of each node is updated by aggregating neighbor information weighted by attention coefficients:

$$h'_i = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} Wh_j \right) \quad (3)$$

In Eq. (3), h_j represents neighbor nodes; W denotes a trainable weight matrix; $\mathcal{N}(i)$ represents the neighbor set of node i . The attention coefficient α_{ij} is calculated as Eq. (4):

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_j]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(a^T [Wh_i \| Wh_k]))} \quad (4)$$

Here, Wh_j represents the result of the linear transformation of node j 's features. For example, in a heterogeneous graph such as “person–ticket gate–equipment,” frequent interactions between a person and a ticket gate within a short window increase $\alpha_{(\text{person}, \text{gate})}$, enabling the model to infer potential risks such as shift-change congestion or abnormal card swiping. The GAT output enhances node context awareness and provides structured input for downstream fault propagation and risk prediction³⁰.

To satisfy the high throughput and low latency requirements of real-time construction site data, the platform implements a Kafka message queue and Apache Flink stream processing architecture. Edge nodes push data to Kafka, Flink processes it in real-time, and the results are dispatched to AI inference modules, databases, or the digital twins platform, enabling cloud-edge collaborative data channels.

Intelligent perception and prediction model

The ST-GCN, Swin Transformer, and PPO introduced in this section form the core online inference and policy optimization modules of the “Intelligent Decision Engine” in Fig. 1. Their inference outputs are returned in real time to the digital twins platform to support closed-loop control.

Model selection in the platform follows the principle of “task feature–model architecture alignment–deployment suitability.” For the three key tasks—spatio-temporal prediction, visual recognition, and scheduling optimization—ST-GCN, Swin Transformer, and PPO are selected for the following reasons.

(1) ST-GCN for spatio-temporal prediction.

Compared with one-dimensional temporal convolutional networks or gated recurrent convolutional networks, ST-GCN jointly models spatial topology and temporal evolution within a unified framework. This makes it suitable for rail construction, where construction points and equipment form complex spatial relationships that evolve over time.

(2) Swin transformer for visual recognition.

Compared with convolutional networks such as ResidualNetwork (ResNet), Swin Transformer applies shifted windows and hierarchical feature extraction. This approach preserves sensitivity to local details while capturing long-range dependencies. It is effective for detecting small objects, handling multi-scale variations in construction scenes, and maintaining robustness under conditions such as low light or occlusion.

(3) PPO for scheduling optimization.

Compared with conventional policy gradient methods (e.g., Advantage Actor-Critic) or deep Q-networks, PPO adopts clipped probability ratios for policy updates. This enhances stability and convergence in high-dimensional, continuous action spaces, making it well-suited for scheduling tasks across multiple construction zones under resource constraints.

By combining these three algorithms, the platform achieves a balance between prediction accuracy, model stability, and computational efficiency, meeting the collaborative requirements of rail transit construction.

To capture temporal variations and spatial correlations, the platform applies ST-GCN for tasks such as predicting construction progress, passenger flow, and workforce attendance³¹.

Construction points are represented as nodes, while construction processes or personnel flows are represented as edges in a graph $G = (V, E)$. Node features evolve into a spatiotemporal sequence: $X \in \mathbb{R}^{N \times T \times F}$, where N is the number of nodes, T is the number of time steps, and F represents the feature dimension (e.g., job intensity, personnel density, task delays). Spatiotemporal modeling is expressed as Eq. (5):

$$H^{(l+1)} = \sigma \left(\sum_{k=0}^K D_k^{-1/2} A_k D_k^{-1/2} H^{(l)} W_k \right) \quad (5)$$

In Eq. (5), A_k refers to the k -th order adjacency matrix; W_k represents a learnable parameter; σ is a nonlinear activation function; K is the temporal convolution kernel order; D_k indicates the dimension of the key vector; $H^{(l)}$ is the input representation at layer l .

Engineering Example: When progress in tunnel section A slows, $k=1$ captures the resulting efficiency decline and material buildup at adjacent section B, while $k=2$ captures disruptions in workflow further downstream.

For video and image data from surveillance, drones, and mobile uploads, the platform applies Swin Transformer models for recognition tasks³². The pipeline consists of follows these steps. Original images are divided into fixed-size patches. Each patch undergoes linear embedding into feature vectors $x_i \in \mathbb{R}^d$. Window-based multi-head attention and sliding window mechanisms capture local and global contextual relationships.

Final outputs provide classification or segmentation results for identifying safety helmet usage, equipment anomalies, and regulatory violations in work zones.

The loss function adopts cross-entropy and demonstrates strong robustness in addressing small-target detection and inter-class imbalance problems, as defined in Eq. (6):

$$\mathcal{L}_{\text{focal}} = -\alpha_t (1 - p_t)^\gamma \log(p_t) \quad (6)$$

In Eq. (6), $\mathcal{L}_{\text{focal}}$ represents the objective loss function for model training in classification tasks; α_t is the weighting factor for category t , which mitigates the impact of class imbalance; p_t denotes the model's predicted probability for the current sample category t ; γ denotes the modulation parameter that reduces the contribution of well-classified samples and enhances the training focus on hard-to-classify samples.

Engineering Example: In hard-hat detection, the proportion of “not worn” instances is significantly lower than that of “worn.” By assigning $\alpha_{\text{not worn}} > \alpha_{\text{worn}}$ and setting $\gamma \approx 2$, the recall of rare violation cases can be substantially improved.

Scheduling optimization problems involving multi-zone tasks, resource constraints, and dynamic priority shifts remain challenging for traditional heuristic methods in high-dimensional, nonlinear state spaces^{33,34}. The proposed platform addresses this challenge through a PPO-based reinforcement learning algorithm designed for resource scheduling strategy modeling.

The scheduling problem is formulated as a Markov Decision Process. The state space S incorporates personnel, equipment, task queues, and construction time windows. The action space A represents scheduling decisions, such as task allocation, equipment relocation, and work sequencing. The reward function $R(s, a)$ integrates construction efficiency, energy consumption, task completion rates, and risk avoidance. The policy network $\pi_\theta(a|s)$ utilizes deep neural networks to generate action distributions for optimal scheduling decisions³⁵.

The optimization objective of PPO is to maximize the clipped surrogate function, expressed in Eq. (7):

$$\mathcal{L}^{\text{CLIP}}(\theta) = E_t [\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_t)] \quad (7)$$

In Eq. (7), $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, \hat{A}_t refers to the estimated advantage value, and ε represents the clipping threshold.

Engineering Example: During a conflict between “TBM advancement” and “equipment maintenance,” if the action “temporarily pause maintenance and resume advancement” leads to a measurable reduction in delays and resource conflicts, then $\hat{A}_t > 0$. Under the clipped constraint, PPO increases the likelihood of selecting this action while preventing drastic deviations caused by incidental rewards.

Digital twins-driven feedback optimization system

To advance rail transit engineering from visual monitoring toward intelligent feedback, the platform constructs a closed-loop optimization system centered on BIM + GIS integration, a digital twins engine, and AI-driven perception and feedback. This system supports a comprehensive management cycle encompassing status monitoring, prediction, simulation, scheduling, and feedback.

Based on BIM models and 3D GIS maps, the platform enables entity-level modeling and status mapping of multiple targets, including construction sites, sections, lines, and equipment³⁶. By integrating IoT devices, video collection nodes, and the data fusion middleware, it implements two-way synchronization between physical entities and the cloud platform. Real-time status reports update BIM/GIS model parameters and the visual layer. In cases of abnormal conditions, the platform can issue control instructions to edge devices to support closed-loop handling^{37,38}.

The system employs a Unity + Unreal hybrid architecture to construct a high-fidelity digital twins engine. This engine integrates building structure models, terrain layers, and dynamic environmental data to enable dynamic simulation of engineering processes, multi-source data visualization, key event deduction, emergency drills, and the simulation of construction plans and resource allocation strategies.

To achieve effective integration of BIM and GIS data, the platform adopts an integration strategy based on spatial indexing and semantic mapping. The BIM model is based on the Industry Foundation Classes standard format, which includes detailed semantic information of building components and construction processes; GIS data is based on the CityGML format, which expresses urban geographic spatial information and hierarchical terrain structure. During the integration process, the platform addresses the differences in coordinate reference, scale accuracy, and spatial overlap between “wall”, “ground”, and geographic feature classification by aligning with spatial bounding boxes and mapping semantic labels (such as “wall”, “ground”, and geographic feature classification). For spatial overlapping conflicts (component models overlapping with terrain), priority-based conflict elimination rules and triangulation reconstruction methods are used for optimization. This achieves a 1:1 mapping effect consistent with the real environment structure in the virtual model.

The digital twins system is integrated with the data fusion middleware and intelligent decision engine to form a perception–analysis–optimization–feedback loop. Real-time status data are collected in the perception phase. Risk prediction and schedule deviation detection are performed in the analysis phase. The evolution of the construction site is reproduced in the simulation phase, while potential system behaviors are evaluated in the decision-making phase. In the feedback phase, recommended strategies are transmitted to edge devices, completing closed-loop control at the platform level.

AI decision interpretability and trustworthiness design

Given the safety-critical nature of rail transit systems, the platform integrates interpretability and trustworthiness mechanisms into the design and deployment of AI models (ST-GCN, Swin Transformer, and PPO). These mechanisms enable stakeholders to understand, validate, and trust system-generated decisions.

(1) Interpretability methods

For the ST-GCN model, a spatio-temporal attention weight-based visualization method is applied to highlight the nodes and time periods that contribute most to predictions, helping engineers identify key influencing factors. For the Swin Transformer model, Gradient-weighted Class Activation Mapping (Grad-CAM) is used to generate heatmaps that reveal the regions of focus during image recognition, thereby supporting verification of the decision basis. For PPO scheduling policies, key state-action pairs and their associated rewards are recorded, and the decision-making process is replayed in the digital twins simulation, allowing users to visually inspect the rationale behind the strategies.

(2) Trustworthiness assurance

During model training and inference, uncertainty estimation is incorporated to provide confidence intervals or confidence scores for outputs, reflecting the stability of decisions. Human review interfaces are embedded into AI prediction and scheduling modules, particularly for high-risk tasks such as equipment speed limits or emergency response operations, thereby implementing an “AI + human” joint confirmation mechanism. In addition, all model inputs, outputs, and intermediate inference records are archived as traceable audit logs, ensuring compliance with safety regulations and supporting post-event analysis.

Through these measures, the platform not only enhances model performance but also ensures interpretability and trustworthiness in high-risk rail construction environments. It provides a reliable foundation for large-scale engineering deployment.

Experimental design and performance evaluation

Datasets collection

To support the training and evaluation of diverse intelligent models, this study constructs a comprehensive multi-source dataset covering construction processes, operational equipment, personnel behaviors, image and video data, and scheduling logs. The primary sources include project management data and construction logs provided by a municipal rail transit company, along with train signaling data, equipment operation logs, and scheduling records collected in collaboration with the Signaling Experiment Center of China Railway Rolling Stock Corporation. Additional data comes from edge video capture terminals, IoT sensing devices deployed at project sites, and user behavior logs with system response records during platform prototype operation. The complete dataset reaches 18 terabytes (TB), covering the period from January 2021 to December 2023. It spans 8 metro lines, 122 construction sections, and more than 3,000 workers, thereby providing rich spatiotemporal heterogeneity and authentic representation of engineering scenarios.

To support the modeling tasks introduced in Sect. “[Research Model](#)”, the platform performs preprocessing, annotation, and structured distribution of raw data. Three specialized sub-datasets are constructed for prediction modeling, visual recognition, and scheduling optimization. Video and image data are acquired through edge AI gateways at standardized 1080p resolution, with keyframes manually annotated through multiple expert rounds, maintaining accuracy above 98%. Structured and log data are extracted automatically from construction management platforms, equipment monitoring systems, and scheduling subsystems, with timestamp consistency checks. Sensor data are transmitted via the MQTT protocol to Kafka clusters, while Flink systems perform unified format conversion and quality inspection, discarding records with more than 15% missing values. All data undergo sensitive information desensitization and anonymization to ensure compliance and safe release for research use.

To promote reproducibility, an anonymized sample dataset is publicly released under strict de-identification protocols. It includes representative subsets such as passenger flow time series, construction logs, equipment IoT logs, image annotations, scheduling requests, and streaming performance measurements. A data dictionary and detailed usage protocol are also provided to facilitate experimental replication. The sample dataset and accompanying documentation are available as a public attachment (rail platform sample dataset.xlsx) in the [supplementary material](#).

Note: For compliance and privacy reasons, the released dataset contains synthetic or anonymized data. Its statistical properties remain consistent with the real data, while ensuring that no personally identifiable information is disclosed.

Experimental environment

The experimental system of the platform is deployed on a Huawei Cloud Stack 8.0 private cloud environment, establishing a full-process operational architecture with edge-cloud-end collaborative capabilities. The system supports closed-loop workflows covering front-end data acquisition, cloud-based model training, and terminal control feedback. To enable efficient training and inference of complex AI models, the platform is equipped with high-performance computing resources and mainstream AI software frameworks. The detailed configurations are presented in Table 1.

Category	Configuration/version description
Graphics processing unit (GPU) acceleration	8 × Nvidia A100 80GB (NVLink Interconnection)
Central processing unit and memory	256 cores Intel Xeon Gold 6338 + 2TB Double Data Rate 4 Error-Correcting Code memory
Storage system	Distributed Solid-State Drive storage (≥ 4 gigabytes (GB)/s), and hierarchical management of hot and cold data
Network environment	100Gb InfiniBand Interconnect + Jetson Edge AI node deployment (12 units)
Operating system	EulerOS + Kernel-based Virtual Machine virtualization
AI framework	PyTorch 2.2, TensorFlow 2.14, Deep Graph Library 1.1.2, PyTorch Geometric
Data processing	Apache Flink 1.17, Kafka 3.5, Neo4j 5.8 graph database
Visual simulation	Unity 2022.3 + Unreal Engine 5.2 (For digital twins engines)

Table 1. Experimental configuration.

Model type	Parameter	Value
ST-GCN	Number of convolutional layers	4
	Temporal window length	60 s
	Adjacency matrix order (K)	3
Transformer-encoder	Number of attention heads	8
	Embedding dimension d_{model}	512
	Dropout rate	0.1
Swin transformer	Patch size	4×4
	Sliding window size	7×7
	Batch size	32
PPO scheduling model	Learning rate	0.00025
	Discount factor γ	0.95

Table 2. Core parameter configuration.

Parameters setting

Based on the characteristics of the actual dataset and the experimental objectives, this study systematically configures and optimizes the hyperparameters of each sub-model. All parameter settings are determined through cross-validation and empirical adjustment, while overfitting is prevented by employing an early-stopping mechanism. The core parameter configurations of the main models are presented in Table 2.

All models are trained on an Nvidia A100 GPU with the Adam optimizer. A dynamic learning rate adjustment strategy is applied by monitoring validation loss.

To ensure reproducibility, this study provides additional details for the Transformer-Encoder and ST-GCN models. All models are trained on an Nvidia A100 server (40 GB memory) with a batch size of 64, using the Adam optimizer with an initial learning rate of 0.0001 and a weight decay of 0.0005. The dataset is divided into training, validation, and test sets in a 7:2:1 ratio, ensuring balanced representation across different time periods and construction zones.

For ST-GCN, the temporal window is set to 60 s, which is identified as optimal through cross-validation over candidate values {30, 45, 60, 90}. This configuration achieves a balance between capturing short-term fluctuations and extracting long-term patterns. For the Transformer-Encoder, the configuration includes 8 attention heads, a hidden dimension of 512, and 4 layers, which provides a validated trade-off between computational efficiency and model performance.

To support real-time data processing in high concurrency track construction scenarios, this platform adopts a Kafka + Flink combination architecture in the data channel module and performs engineering level configuration on key parameters.

Kafka: According to the logic of dividing railway lines and work areas, set up data flow channels for different construction areas corresponding to topics. Each topic is configured with 8–16 partitions, supporting horizontal expansion and parallel consumption. The producer has enabled compression (Snappy) and batch sending (BatchSize = 32KB) strategies to optimize throughput.

Flink: In the stream computing module, the key operators set the parallelism to 16–32 and use the Checkpoint mechanism to write states every 30 s to improve the system's fault tolerance. Simultaneously enable Event Time semantics and out of order buffering mechanism (Allowed Lateness = 3s) to ensure temporal consistency.

This configuration has been validated in platform throughput testing (Sect. “[System Throughput And Response Performance Testing](#)”) to support peak traffic of 30,000 + msgs/s, meeting the real-time processing requirements in high-density construction scenarios.

Performance evaluation

This section evaluates the platform's effectiveness and engineering applicability across four dimensions:

- (1) Prediction and recognition accuracy (Sect. “[Evaluation Of The Predictive Performance Of The Model](#)”): measured using mean squared error (MSE), mean absolute error (MAE), accuracy, recall, and F1-score.
- (2) Data channel and inference latency/throughput (Sect. “[System Throughput And Response Performance Testing](#)”): quantified in messages per second (messages/s) and milliseconds (ms).
- (3) Scheduling optimization effectiveness and stability (Sect. “[Strategy Optimization And Scheduling Efficiency Evaluation](#)”): assessed using task completion time (minutes), conflict and delay rates (%), and stability indicators.
- (4) Digital twins rendering and interaction performance (Sect. “[Performance Test Of 3d Simulation Engine](#)” and Scalability/Elasticity (Sect. “[Elastic Testing Of System Performance In Multi-Scenario Deployments](#)”): evaluated in terms of frames per second (FPS), interaction latency (ms), stability (%), and performance variations under concurrent load conditions.

All figures and tables specify units and statistical criteria in their captions.

Evaluation of the predictive performance of the model

Passenger flow variations during construction peak periods and schedule delays on key paths were predicted using subway entrance/exit flow data and construction plan logs. The predictive performance of Long Short-Term Memory (LSTM), GRU, and eXtreme Gradient Boosting (XGBoost) is compared. All reported results are expressed as the mean \pm standard deviation (SD) from three independent experiments. Evaluation metrics include mean squared error (MSE, (persons/min)²), mean absolute error (MAE, persons/min), accuracy, recall, and F1-score.

For image- and video-based tasks, both binary and multi-class classification problems are constructed. Swin Transformer (the proposed model), ResNet-50, and You Only Look Once version 5 (YOLOv5) are compared using accuracy, recall, and F1-score as performance metrics. Figure 2 presents the comparative results across tasks, with error bars denoting SD.

As illustrated in Fig. 2, for passenger flow prediction, ST-GCN (0.947 ± 0.005) achieves approximately 8.3% higher accuracy than LSTM (0.864 ± 0.006), while also substantially reducing MSE (0.083 ± 0.004 vs. 0.129 ± 0.006). This demonstrates the advantage of incorporating spatio-temporal structural modeling. In image recognition tasks, the Swin Transformer outperforms ResNet-50 and YOLOv5 across accuracy, recall, and F1-score, effectively handling complex scenes characterized by dense structures and large variations in object sizes.

System throughput and response performance testing

To evaluate the operational efficiency of the integrated rail transit platform in multi-source data processing and intelligent scheduling feedback, this study examines the throughput capacity and response latency of its five core modules. The testing scenario simulates the real data load during peak construction periods, including sensor data streams, video frame streams, and scheduling request flows. All performance tests are conducted under identical hardware conditions, with each test repeated three times. The reported values are mean results, and standard deviations are shown as error bars. The outcomes are presented in Fig. 3.

As illustrated in Fig. 3, the Kafka and Flink modules demonstrate the highest throughput and the lowest latency, fully meeting the real-time processing requirements of multi-source data during peak periods. The ST-GCN and Swin Transformer modules also maintain stable performance under high GPU utilization. The PPO scheduling feedback engine exhibits relatively higher latency in handling complex tasks; however, its availability remains above 95%, ensuring reliable multi-task scheduling in rail transit scenarios.

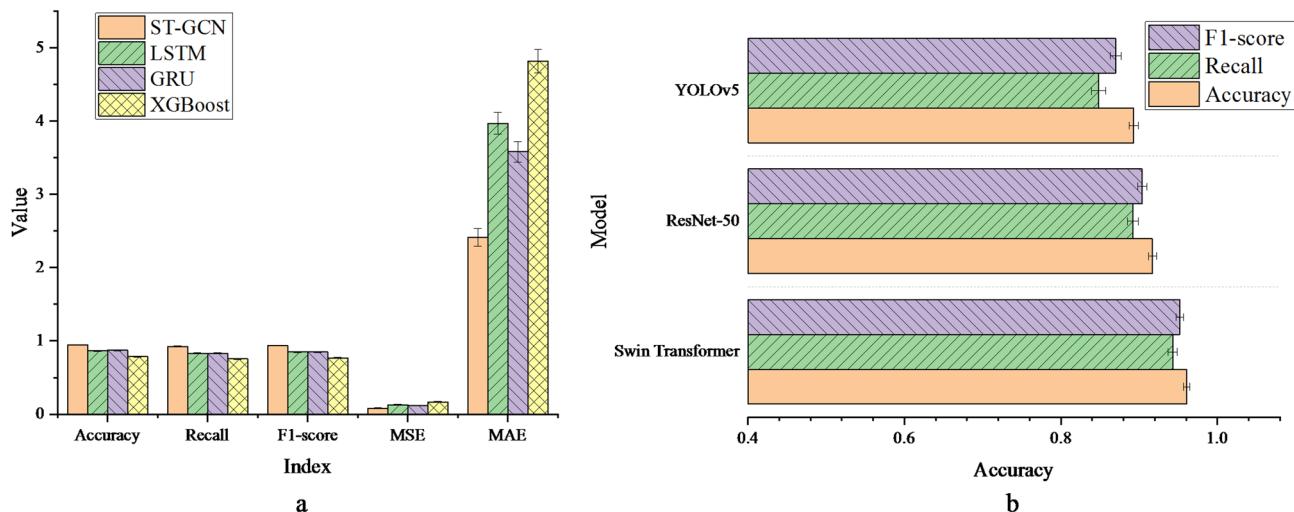


Fig. 2. Comparison of the prediction and recognition performance across models in different tasks.

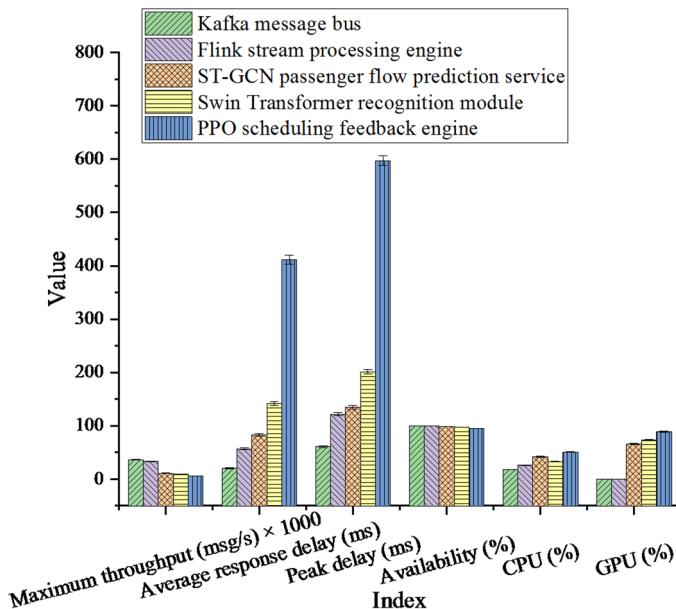


Fig. 3. Evaluation of throughput and latency performance of each core module.

Metric	Proposed platform	Siemens railigent	Difference
Maximum throughput (msg/s × 1000)	33.2	29.7	12%
Average response latency (ms)	57	67	-15%
Platform availability (%)	99.95	99.94	≈ 0
Average prediction accuracy (%)	94.7	92.3	2.60%

Table 3. Comparison of the proposed platform with Siemens Railigent.

To further assess the platform's overall performance against industry standards, Siemens Railigent is used as a benchmark. Under identical data throughput and latency testing conditions, the comparative results of overall processing capability, response performance, and availability are summarized in Table 3.

The results in Table 3 indicate that, with the aid of edge computing optimizations, the proposed platform achieves approximately 12% higher throughput than the benchmark, reduces average response latency by 15%, and maintains comparable availability. Moreover, it improves average prediction accuracy by 2.6%, highlighting its superior capability in managing large-scale, real-time rail transit operations.

Strategy optimization and scheduling efficiency evaluation

Testing is conducted under uniform task sets and resource constraints, simulating typical scheduling scenarios including tunnel excavation, sleeper laying, and cable routing across five task categories. Performance is evaluated along five key dimensions: average task completion time (minutes), resource conflict occurrence rate (%), task delay rate (%), normalized average reward, and scheduling stability measured by the Coefficient of Variation (CV), where lower values indicate higher stability. All scheduling performance tests are executed under identical hardware and task conditions, repeated three times, with standard deviations used to plot error bars. Figure 4 presents a comparison of efficiency and stability among the scheduling optimization models across multiple work zones.

As shown in Fig. 4, the PPO-based scheduling model consistently outperforms rule-based scheduling across all five evaluated dimensions. Specifically, the average task completion time decreases by approximately 20%, resource conflict rate and task delay rate are reduced by 60.9% and 67.7%, respectively, the normalized average reward increases by 35.6%, and scheduling stability (CV) improves by 46.1%. These results demonstrate that the PPO model achieves superior efficiency and stability in complex multi-zone tasks, making it particularly well-suited for real-time task coordination and conflict management on construction sites.

Performance test of 3D simulation engine

To validate the platform's modeling and interaction capabilities in complex construction scenarios, multi-dimensional performance testing is conducted on the Unity + Unreal hybrid rendering engine integrated within the digital twins platform. The evaluation considers five typical simulation scenarios: shield tunneling, process evolution, material handling, emergency drills, and plan deduction. Testing metrics include loading time (seconds), average frame rate (FPS), interaction response latency (milliseconds), peak GPU utilization (%), and

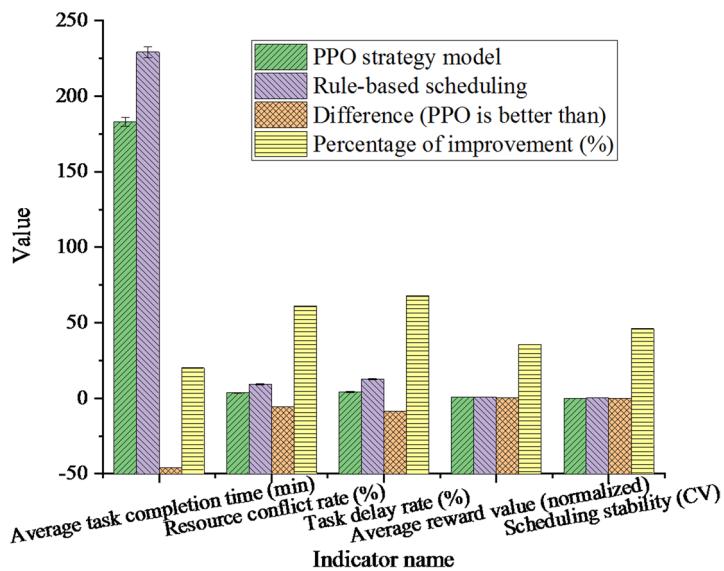


Fig. 4. Efficiency and stability comparison of scheduling optimization models across multiple work zones.

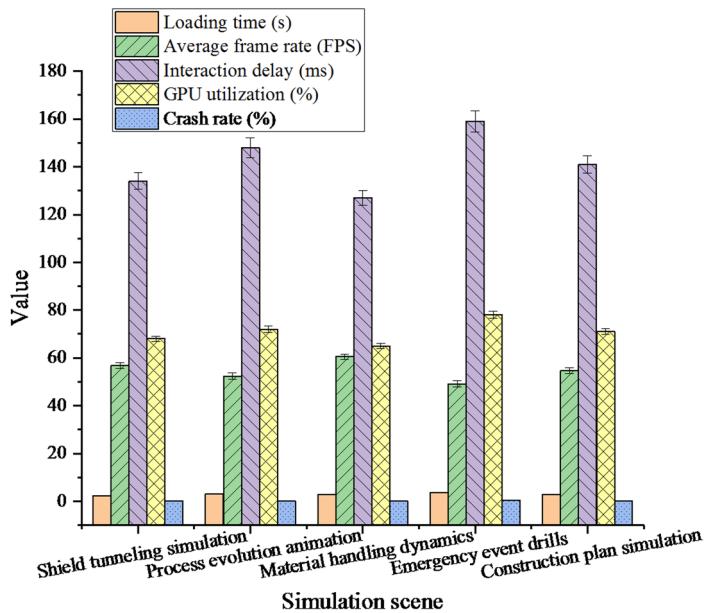


Fig. 5. Performance test results of the 3D engine across typical simulation scenarios.

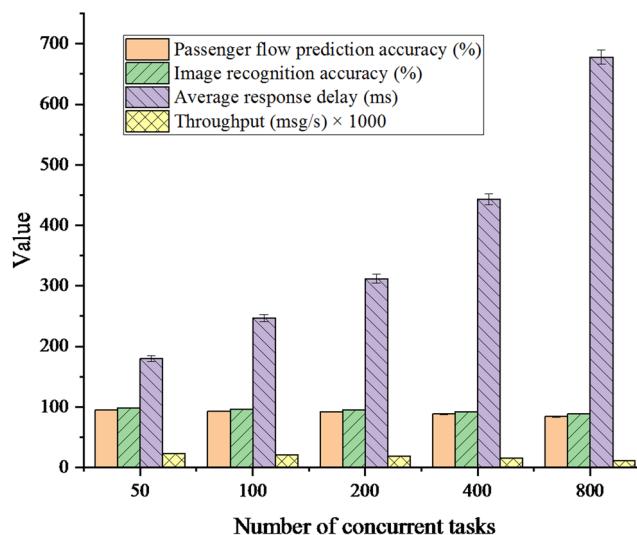
system crash/abnormality rate (%). Each scenario is tested independently three times. Results are presented in Fig. 5.

As shown in Fig. 5, the platform maintains an average frame rate above 50 FPS across all scenarios, with interaction latency below 160 ms. Average loading time does not exceed 3 s, and peak GPU utilization ranges between 65–78%. System stability exceeds 99.7%, with crash rates $\leq 0.3\%$, indicating smooth operation and high reliability under complex construction visualization tasks.

To achieve real-time 1:1 mapping between physical entities and virtual twins, the platform integrates IoT data synchronization middleware with the hybrid rendering engine. On-site sensor data are transmitted via MQTT and OPC-UA protocols to edge nodes and then pushed to the cloud simulation engine with a refresh cycle ≤ 300 ms. Latency is managed through priority queues, batch compression, and sliding-window smoothing. In case of missing or anomalous data, the system triggers automatic retransmission or rolls back to the most recent consistent snapshot, ensuring synchronization between the virtual scene and the physical environment.

To assess the engine's effectiveness in construction decision-making, a usability study is conducted with 10 construction managers and 5 project schedulers from different work zones. Evaluation criteria include interface usability, information clarity, decision-support value, and overall satisfaction, rated on a 1–5 Likert scale

Evaluation metric	Mean score (\pm SD)	Improvement vs. No assistance
Interface usability	4.6 \pm 0.3	–
Information clarity	4.4 \pm 0.4	–
Decision-support value	4.5 \pm 0.5	–
Overall satisfaction	4.7 \pm 0.2	–
Decision time reduction (%)	–	18.2 \pm 3.5
Decision error rate reduction (%)	–	21.5 \pm 4.1

Table 4. Usability and decision-support evaluation.**Fig. 6.** Changes in system performance with different deployment scales.

Concurrent tasks	Prediction accuracy (%)—this platform	Prediction accuracy (%)—railigent	Throughput (msg/s $\times 10^3$)—this platform	Throughput (msg/s $\times 10^3$)—railigent
50	94.7	94.2	23	22.1
200	91.6	90.1	18.9	17.2
800	84.1	80.3	11.6	9.5

Table 5. Multi-scenario deployment performance comparison.

(5 = highest). Participants' task plan adjustment time and error rates are also compared with and without digital twins assistance. Results are summarized in Table 4.

The survey results demonstrate that the digital twins engine is highly effective in interface interaction and information presentation. In task scheduling scenarios, it significantly reduces decision-making time and error rates, confirming its practical value and potential for broader deployment in rail construction management.

Elastic testing of system performance in multi-scenario deployments

The platform's performance under varying concurrent task loads is evaluated to assess scalability and elasticity. Tests are conducted under identical hardware and network conditions, with concurrent task volumes of 50, 100, 200, 400, and 800, each repeated three times. Metrics recorded include passenger flow prediction accuracy, image recognition accuracy, average response latency, and throughput, with SD used for error bars, as shown in Fig. 6.

As illustrated in Fig. 6, when concurrent tasks do not exceed 200, both prediction and recognition accuracy remain above 91%, with latency maintained below 350 ms. Under extreme load conditions (800 concurrent tasks), prediction accuracy decreases to $84.1\% \pm SD_1$, image recognition accuracy decreases to $88.9\% \pm SD_2$, latency rises to $678 \pm SD_3$ ms, and throughput drops to $11.6 \pm SD_4 \times 10^3$ msg/s. These results demonstrate the platform's scalability and the predictable performance degradation under high concurrency.

To further evaluate elasticity across deployment scales, Siemens Railigent is tested under identical conditions for comparison. Results are summarized in Table 5.

Under high-concurrency scenarios (≥ 400 tasks), the platform exhibits smaller reductions in prediction and recognition accuracy and a more gradual throughput decline compared with Railigent. These results underscore the advantages of edge computing and distributed scheduling architectures in supporting scalable and elastic performance for multi-scenario deployments. The main reason for performance degradation under extremely high concurrency conditions (800 concurrent tasks) is the dual bottleneck of computation and communication. On the one hand, GPU resource utilization is approaching its upper limit ($> 90\%$), and the competition for video memory between inference tasks leads to a decrease in batch processing efficiency of the model. On the other hand, the network bandwidth between edge nodes and the cloud becomes saturated during high-frequency data transmission, causing an increase in message queue (Kafka) waiting time, which in turn affects overall throughput and response latency. In addition, some scheduling requests experience microsecond level blocking during the resource lock competition phase, resulting in a non-linear increase in overall system latency in high concurrency scenarios. These phenomena indicate that in larger scale deployments in the future, platform scalability can be further improved by introducing mechanisms such as multi GPU parallel inference, bandwidth adaptive compression, and asynchronous task queues.

Security and privacy assurance

To guarantee data security and privacy compliance during the engineering deployment of the integrated rail transit platform, multiple protective measures are implemented across system design and operation.

Data transmission security: All data channels employ TLS 1.3/Secure Sockets Layer encryption, preventing man-in-the-middle attacks and eavesdropping during transmission.

Data storage security: Critical operational data, including sensor streams, construction logs, and image/video data, are encrypted using Advanced Encryption Standard (AES)-256 symmetric encryption. Periodic key rotation is implemented to mitigate potential leakage risks.

Access control: RBAC combined with OAuth 2.0 authentication ensures that only authorized users and devices can access specific resources and services.

Potential threats and corresponding mitigation measures in practical applications include:

- **Adversarial attacks:** Input validation modules intercept anomalous inputs for image recognition and predictive models, while adversarial training is applied to enhance model robustness.
- **Data leakage risks:** Log auditing and anomaly detection continuously monitor access behaviors, triggering alerts and isolation procedures upon detecting suspicious activity.
- **Network coverage limitations:** Local caching and offline inference mechanisms are deployed at critical nodes to maintain service continuity during temporary data transmission failures.

This security framework ensures that the platform maintains high-performance real-time operation while preserving data confidentiality, integrity, and availability, fully complying with safety and regulatory requirements in the rail transit sector.

Discussion

The experimental results demonstrate that the platform achieves strong performance across prediction, recognition, response, and scheduling tasks, benefiting from structured spatiotemporal modeling, hierarchical visual attention mechanisms, and low-latency data streams integrated with strategy optimization tailored for rail transit engineering. ST-GCN and Swin Transformer show clear advantages in capturing temporal-spatial dependencies and interpreting complex visual scenarios, respectively, although their performance can fluctuate under low-quality inputs or network latency. The Kafka + Flink data pipeline and PPO-based scheduling strategy ensure stability and real-time responsiveness for multi-task operations, but large-scale deployments remain resource-intensive.

These findings highlight several future optimization directions, including enhancing robustness under poor-quality data, implementing adaptive allocation of computing resources, and improving cross-scenario transferability. The platform demonstrates strong generalizability across diverse physical and operational environments. In high-altitude railway scenarios, climatic conditions and network stability may impact real-time data collection and transmission. In ultra-dense urban subway environments, complex signal interference and high-concurrency multi-source data streams increase demands on system throughput and latency. By leveraging modular deployment, edge computing, and adaptive data flow control, the platform maintains high prediction accuracy and system stability under these varying conditions, demonstrating its transfer potential and engineering applicability.

Despite successful validation in real urban rail projects, deployment challenges remain. Network coverage limitations can degrade real-time performance, regulatory differences may constrain data collection and storage strategies, and heterogeneous edge devices can introduce compatibility issues. To mitigate these risks, offline caching and local inference mechanisms alleviate network instability, encryption and access control enforce data compliance, and standardized interface protocols enhance cross-device compatibility. These measures substantially reduce deployment risks and improve adaptability across engineering contexts.

Several limitations are identified. First, low-quality video inputs—such as severe compression, poor illumination, or camera shake—can reduce image recognition accuracy, a common challenge in complex construction environments. Second, hardware failure or overload at edge nodes may cause response delays or task interruptions; redundancy and automatic failover mitigate these risks, but real-time performance can still be affected. Third, under extreme high-concurrency scenarios with fluctuating network conditions, some decline in prediction and recognition accuracy is unavoidable, highlighting the need for more efficient model compression and adaptive inference strategies. Finally, cross-regional deployment requires addressing

differences in regulations, standards, and data security policies, which influence both technical implementation and subsequent maintenance.

Future work will focus on enhancing model robustness, implementing early anomaly detection and warning systems, and developing a unified security and data governance framework to support cross-domain collaboration.

Conclusion

Research contribution

This study presents the design of a rail transit digital-intelligent integration platform based on a “cloud–edge–end” architecture. By integrating models such as ST-GCN, Swin Transformer, and PPO, the platform achieves a full-process closed loop encompassing data acquisition, intelligent analysis, and digital twins–driven feedback. The proposed cross-modal data fusion and digital twins–based strategy optimization methods substantially enhance passenger flow prediction accuracy, construction schedule efficiency, and system response performance. These contributions provide both theoretical support and practical demonstration for the informatization and intelligent management of rail transit engineering projects.

Future works and research limitations

Although the platform developed in this study demonstrates strong performance and scalability in practical engineering applications, several limitations remain. The image recognition module relies heavily on edge computing resources, which constrains deployment on low-power devices. The PPO-based scheduling optimization model exhibits variable convergence speed under ultra-large task scales, highlighting the need for multi-agent learning mechanisms to enhance collaborative efficiency. Additionally, the digital twins engine is currently at an early stage of autonomous reasoning and feedback control.

Future research directions include: (1) developing lightweight model architectures and edge-collaborative inference mechanisms to improve deployment capabilities on resource-constrained nodes; (2) incorporating causal reasoning and knowledge graph technologies to enhance interpretability and generalization for complex construction events; (3) advancing the intelligent evolution of the digital twins system to enable dynamic prediction of multi-objective states and automated strategy generation.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author Jianbo Guan on reasonable request via e-mail guanjb@nbmetro.com.

Received: 16 June 2025; Accepted: 19 November 2025

Published online: 29 November 2025

References

1. Yang, D., Cui, E., Wang, H. & Zhang, H. EH-edge—an energy harvesting-driven edge IoT platform for online failure prediction of rail transit vehicles: A case study of a cloud, edge, and end device collaborative computing paradigm. *IEEE Veh. Technol. Mag.* **16**(2), 95–103 (2021).
2. Sun, Y., Zhang, C., Chang, B., Zhu, Q., Zhang, R., Wang, X., & Ren, Y. (2024, November). Multi-level Intelligent Operation and Maintenance Platform of Digital Station Based on Cloud Edge-End Architecture. In: International Joint Conference on Energy, Electrical and Power Engineering (pp. 115–125). Singapore: Springer Nature Singapore.
3. Guo, Q., Xu, Z., Yuan, J. & Wei, Y. Cloud-edge collaboration-based task offloading strategy in railway IoT for intelligent detection. *Wireless Netw.* **31**(2), 1361–1376 (2025).
4. Gong, T., Zhu, L., Yu, F. R. & Tang, T. Edge intelligence in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **24**(9), 8919–8944 (2023).
5. Trigka, M. & Dritsas, E. Edge and cloud computing in smart cities. *Future Internet* **17**(3), 118 (2025).
6. Xu, Y., Feng, D., Zhao, M., Sun, Y. & Xia, X. G. Edge intelligence empowered metaverse: Architecture, technologies, and open issues. *IEEE Network* **37**(6), 92–100 (2023).
7. Zhang, Y., Yu, H., Zhou, W. & Man, M. Application and research of IoT architecture for end-net-cloud edge computing. *Electronics* **12**(1), 1 (2022).
8. Gao, Z. et al. Cloud-edge-end integrated artificial intelligence based on ensemble learning. *Comput. Commun.* **2025**, 108103 (2025).
9. Zhang, Q. et al. Cloud-edge-end-based aircraft assembly production quality monitoring system framework and applications. *J. Manuf. Syst.* **75**, 116–131 (2024).
10. Fog-Cloud, A. E. M. H. Intelligent transportation. *Urban Mob. Chall. Intell. Transport. Syst.* **11**, 315 (2025).
11. Atanasov, I., Dimitrova, D., Pencheva, E. & Trifonov, V. Railway cloud resource management as a service. *Future Internet* **17**(5), 192 (2025).
12. Lifelo, Z., Ding, J., Ning, H. & Dhelim, S. Artificial intelligence-enabled metaverse for sustainable smart cities: Technologies, applications, challenges, and future directions. *Electronics* **13**(24), 4874 (2024).
13. Zhukabayeva, T., Zholsbayeva, L., Karabayev, N., Khan, S. & Alnazzawi, N. Cybersecurity solutions for industrial internet of things–edge computing integration: Challenges, threats, and future directions. *Sensors* **25**(1), 213 (2025).
14. Ghosh, S. K., Raha, A., Raghunathan, V. & Raghunathan, A. Partnner: Platform-agnostic adaptive edge-cloud dnn partitioning for minimizing end-to-end latency. *ACM Transac. Embed. Comput. Syst.* **23**(1), 1–38 (2024).
15. Benmerar, T. Z. et al. Towards establishing intelligent multi-domain edge orchestration for highly distributed immersive services: A virtual touring use case. *Clust. Comput.* **27**(4), 4223–4253 (2024).
16. Souza, D. et al. Architectural trends in collaborative computing: approaches in the internet of everything era. *Future Internet* **16**(12), 445 (2024).
17. Li, Q., Cui, L., Wang, Q., Guo, A. & Yuan, H. Construction and application of an agent-based intelligent operation and maintenance system for UAV. *Drones* **9**(4), 309 (2025).
18. Fan, K. et al. MSIAP: A dynamic searchable encryption for privacy-protection on smart grid with cloud-edge-end. *IEEE Transac. Cloud Comput.* **11**(2), 1170–1181 (2021).

19. Zhang, C. et al. Fed4UL: A cloud–edge–end collaborative federated learning framework for addressing the non-IID data issue in UAV logistics. *Drones* **8**(7), 312 (2024).
20. Xu, R. et al. A secure interconnected autonomous system architecture for multi-domain IoT ecosystems. *IEEE Commun. Mag.* **62**(7), 52–57 (2024).
21. Zhang, L., Yan, Y. & Hu, Y. Dynamic flexible scheduling with transportation constraints by multi-agent reinforcement learning. *Eng. Appl. Artif. Intell.* **134**, 108699 (2024).
22. Dai, G. et al. Tucker decomposition-enhanced dynamic graph convolutional networks for crowd flows prediction. *ACM Transac. Intell. Syst. Technol.* **16**(1), 1–19 (2025).
23. Rolofs, G., Wilking, F., Goetz, S. & Wartzack, S. Integrating digital twins and cyber–physical systems for flexible energy management in manufacturing facilities: A conceptual framework. *Electronics* **13**(24), 4964 (2024).
24. Whitmore, J., Mehra, P., Hastings, O. & Linford, E. Improving real-time performance of autonomous driving systems with edge computing. *Int. J. Manag. Sci. Res.* **8**(5), 35–41 (2025).
25. Kliestik, T., Kral, P., Bugaj, M. & Durana, P. Generative artificial intelligence of things systems, multisensory immersive extended reality technologies, and algorithmic big data simulation and modelling tools in digital twin industrial metaverse. *Equilib. Q. J. Econ. Econ. Policy* **19**(2), 429–461 (2024).
26. Qi, H. et al. Multi-compression scale DNN inference acceleration based on cloud-edge-end collaboration. *ACM Transac. Embed. Comput. Syst.* **23**(1), 1–25 (2024).
27. Wang, Y. et al. A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects. *IEEE Internet Things J.* **10**(17), 14965–14987 (2023).
28. Yin, Z., Luan, T. H., Cheng, N., Hui, Y., & Wang, W. (2022). Cybertwin-enabled 6G space-air-ground integrated networks: Architecture, open issue, and challenges. arXiv preprint [arXiv:2204.12153](https://arxiv.org/abs/2204.12153).
29. Rocha, A. et al. Edge AI for internet of medical things: A literature review. *Comput. Electr. Eng.* **116**, 109202 (2024).
30. Ding, K. & Fan, L. Q. AML-based web-twin visualization integration framework for DT-enabled and IIoT-driven manufacturing system under I4. 0 workshop. *J. Manuf. Syst.* **64**, 479–496 (2022).
31. Arcas, G. I., Cioara, T., Anghel, I., Lazea, D. & Hangan, A. Edge offloading in smart grid. *Smart Cities* **7**(1), 680–711 (2024).
32. Jiao, S., Wang, H. & Luo, J. SRA-E-ABCO: Terminal task offloading for cloud-edge-end environments. *J. Cloud Comput.* **13**(1), 58 (2024).
33. Aslam, A. M. et al. Metaverse for 6G and beyond: The next revolution and deployment challenges. *IEEE Internet Things Mag.* **6**(1), 32–39 (2023).
34. Wen, H. et al. The digital twins for mine site rescue environment: Application framework and key technologies. *Process Saf. Environ. Prot.* **186**, 176–188 (2024).
35. Baidya, T. & Moh, S. Comprehensive survey on resource allocation for edge-computing-enabled metaverse. *Comput. Sci. Rev.* **54**, 100680 (2024).
36. Jebamikyoush, H., Li, M., Suhas, Y. & Kashef, R. Leveraging machine learning and blockchain in E-commerce and beyond: Benefits, models, and application. *Discov. Artif. Intell.* **3**(1), 3 (2023).
37. Wu, Y., Zhang, K. & Zhang, Y. Digital twin networks: A survey. *IEEE Internet Things J.* **8**(18), 13789–13804 (2021).
38. Ji, B., Zhang, J., Wang, Y., Liu, J. & Wang, W. Achievable rate maximization for IRS-assisted system with relay cooperation of IoT network. *IEEE Trans. Consum. Electron.* **70**(1), 1020–1028 (2023).

Author contributions

Keke Wang: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation Xin Zhou: methodology, software, validation, formal analysis Jianbo Guan: writing—review and editing, visualization, supervision, project administration, funding acquisition.

Funding

This work was supported by the Key Technological Innovation Program of Ningbo City under Grant No.2023Z220.

Declarations

Competing interests

The authors declare no competing interests.

Ethics approval

This article does not contain any studies with human participants or animals performed by any of the authors. All methods were performed in accordance with relevant guidelines and regulations.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-29732-6>.

Correspondence and requests for materials should be addressed to J.G.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025