A GAN-LSTM Based Framework for Dynamic Project Scheduling and Risk Prediction in Engineering Management

Zhixia Fu[1 *], Qiang Su[2], Zhiwei Mu[1]
[1]School of Management, Henan University of Urban Construction, Pingdingshan 467036, China
[2]China Water Resources and Hydropower 11th Engineering Bureau Co., Ltd., Zhengzhou 450001, China
E-mail: fuzhixia111@163.com
[*]Corresponding author

Keywords: generative adversarial network, long short-term memory networks, project progress forecast, risk warning model, dynamic generation

Received: August 4, 2025

As the complexity of large-scale engineering projects increases, traditional schedule management methods face the dual challenges of insufficient prediction accuracy and lagging risk response in a dynamic environment. This study proposes a hybrid framework that integrates a conditional generative adversarial network (cGAN) with a bidirectional long short-term memory network (Bi-LSTM) to achieve dynamic project progress generation and real-time risk warning. The model was trained on a real-world dataset from a cross-sea bridge project, comprising 12 months and over 50,000 process records. The Bi-LSTM module captures temporal dependencies among construction processes, while the cGAN, trained with Wasserstein distance and gradient penalty, generates diverse and plausible progress scenarios conditioned on real process states. The adversarial training uses minimax loss with Adam optimization. Experimental results indicate that the proposed model reduces the mean square error (MSE) by 23.7% compared to a standalone LSTM in progress prediction, and achieves a risk identification accuracy of 91.4%, surpassing a traditional Bayesian network by 18.2%. In dynamic disturbance simulations, the model achieves an early-warning response time of 2.3 hours for emergencies such as resource shortages or extreme weather—67% faster than the baseline. Furthermore, the adversarial generation of 12,000 virtual progress samples mitigates overfitting in small-sample settings and improves the test-set F1-score to 0.89. This study demonstrates the robustness of the cGAN-BiLSTM framework in complex engineering environments and offers a data-driven solution for dynamic schedule optimization and risk management.

Povzetek: Študija predlaga hibridni, podatkovno voden okvir cGAN–BiLSTM za natančnejše dinamično napovedovanje poteka projektov in sprotno opozarjanje na tveganja v kompleksnih inženirskih projektih, ki presega tradicionalne pristope po točnosti, hitrosti odziva in robustnosti.

1 Introduction

In today's complex and ever-changing field of engineering management, accurate prediction of project progress and risk prevention and control are always the core elements that determine the success or failure of the project [1]. With the expansion of infrastructure scale and the improvement of engineering complexity, the traditional schedule management methods gradually expose the defects of lag and staticity [2]. Engineering sites often face dynamic interference factors such as sudden changes in geological conditions, supply chain fluctuations, and human resource adjustments. The superposition of these uncertainties makes it difficult for the schedule based on fixed templates or historical experience to effectively cope with practical challenges [3, 4]. How to break through the shackles of static planning, establish a schedule generation mechanism with dynamic adaptability, and realize early identification and early warning of risks at the same time has become a bottleneck problem that modern engineering management needs to break through urgently.

In current engineering practice, classic methods such as the critical path method and plan review technology are mostly used in schedule management. These methods perform well in highly structured projects, but they are often unable to deal with nonlinear and multi-coupled modern projects [5]. Especially in large-scale projects such as super high-rise buildings and sea-crossing bridges, the dynamic interaction between processes often leads to the chain reaction of schedule deviation, while the traditional model lacks the ability to effectively capture time series characteristics [6, 7]. In recent years, although building information modelling (BIM) and 4D simulation technology have improved the level of visual management, they are still deterministic static deductions, in essence, which cannot simulate random events that emerge in real time during the

construction process. This limitation is particularly obvious when encountering unexpected situations such as extreme weather and equipment failure, which often leads to serious deviation between the progress forecast and actual execution [8].

The rapid development of deep learning technology provides new ideas for breaking through the above dilemmas [9]. Long short-term memory network (LSTM) has shown its advantages in modelling dynamic systems in the fields of power load forecasting and traffic flow analysis due to its excellent time series data processing capabilities [10, 11]. Its gating mechanism can effectively capture the long-term dependencies existing in the project progress, such as the restriction of the concrete curing cycle on subsequent processes or the impact of equipment installation progress on the commissioning stage. At the same time, a generative adversarial network (GAN) generates high-dimensional data through an adversarial training mechanism, which opens up a new way to solve the problem of lack of project schedule samples [12]. This data enhancement capability is particularly valuable for small sample and multi-modal data in construction scenarios, especially when dealing with rare but high-risk construction conditions, which can effectively expand the quality and diversity of training data sets.

The application of a single model often makes it difficult to meet the complex requirements of engineering management. Although LSTM is good at time series prediction, its generation ability is limited by the distribution characteristics of training data. Although GAN has powerful data generation capabilities, it lacks constraints and guidance on sequential logic [13]. This complementarity of technical characteristics provides an intrinsic impetus for model fusion. By constructing the antagonistic framework between the generator and discriminator and embedding the time series modelling ability of LSTM into the generation mechanism of GAN, the progress deduction that not only conforms to the construction logic but also reflects the uncertainty can be theoretically realized. This dynamic generation mode can not only simulate the progress evolution under normal construction conditions but also generate the progress trajectory under various risk scenarios through the exploration of potential space, providing rich simulation samples for risk early warning.

In the dimension of risk early warning, traditional methods are mostly based on threshold setting or expert experience for risk judgment. This method often has the problem of early warning lag when facing new engineering risks or compound risks [14, 15]. The core advantage of the dynamic generation model is that it can construct the mapping relationship between progress status and risk indicators and monitor the evolution trend of risk factors through the real-time generated progress data stream. When the process duration distribution generated by the model is abnormally skewed or the resource consumption curve shows a nonlinear sudden change, the system can automatically trigger the risk warning signal [16]. This data-driven early warning mechanism has higher sensitivity and objectivity than manual experience judgment and is especially suitable for complex construction environments with multiple risks.

The digital transformation of project progress management and risk prevention and control has become an inevitable trend in the development of the industry. The popularization of IoT equipment at construction sites provides a hardware foundation for real-time data collection, and technologies such as drone inspections and smart sensors realize multi-dimensional perception of construction status. These technological innovations provide a data input channel for dynamically generating models, enabling the models to continuously receive on-site progress data and update them online [17]. The formation of this closed-loop data not only enhances the model's adaptability to actual construction conditions but also promotes the paradigm shift of progress management from "post-event correction" to "pre-control beforehand". By building a progress deduction system in a digital twin environment, managers can preview the implementation effects of different decision-making schemes in a virtual space, thereby optimizing resource allocation and reducing trial-and-error costs.

From the perspective of technology integration, introducing deep generative models into the field of engineering management requires crossing knowledge barriers in multiple disciplines. This not only involves the improvement of neural network architecture to adapt to the characteristics of engineering data but also the establishment of constraints and evaluation indexes that conform to the construction logic. For example, in the process of model training, engineering knowledge such as process constraints and resource supply constraints between processes need to be encoded into the loss function to ensure the feasibility of generating the schedule. This interdisciplinary technology integration is not only an innovation of existing engineering management methods but also an important practice of deep integration of artificial intelligence technology with traditional industries. With the advancement of new urbanization construction and the implementation of the "new infrastructure" strategy, this intelligent management tool will play a key role in improving project efficiency reducing project risks, and injecting new kinetic energy into the high-quality development of project construction.

2 Theoretical basis of cross-modal engineering data modeling

2.1 Generative adversarial network principle

Generative adversarial networks (GANs) are widely used in unsupervised or semi-supervised learning, as they do not rely on explicitly labeled data [18-20]. As a core branch of deep generative models, GANs learn the underlying data distribution to generate realistic synthetic samples—for instance, producing new cartoon character images from a training set of such illustrations. This capability overcomes limitations of traditional deep

networks and has significantly advanced generative modeling [21, 22].

As shown in Figure 1, the GAN framework consists of a generator (G) and a discriminator (D). The generator maps random noise to data resembling the real distribution, while the discriminator distinguishes between real and generated samples [23-26]. Through adversarial training based on game theory, both components iteratively improve until Nash equilibrium is reached, avoiding the need for explicit likelihood computation. This framework is a hybrid architecture deeply integrated with Transformer, GAN, and LSTM. Its core process begins with text data input, first extracting high-dimensional features and semantic encoding through the DeBERT-v3 large module, generating high-quality contextual feature representations (Hidden States). These features are then input as conditional information into the GAN-LSTM module: the LSTM units are responsible for capturing dynamic temporal dependencies in project scheduling, while the Generative Adversarial Network (GAN) constructs a cGAN architecture based on LSTM's output — where the generator synthesizes predictive data using random noise and LSTM hidden states, and the discriminator evaluates authenticity based on the same conditions, forming an adversarial training mechanism. The Feature modules on both sides may be used for multimodal feature fusion or attention weighting. The downstream process further incorporates SHRED, Ensemble SINDy, and Sparsity constraints to jointly optimize the model's generalization ability and interpretability. The overall structure achieves end-to-end learning from text input to dynamic risk prediction through the collaboration of multiple modules, combining robustness in temporal modeling with adaptability from generative adversarial training, aligning with the needs of complex scheduling and risk prediction in engineering management.

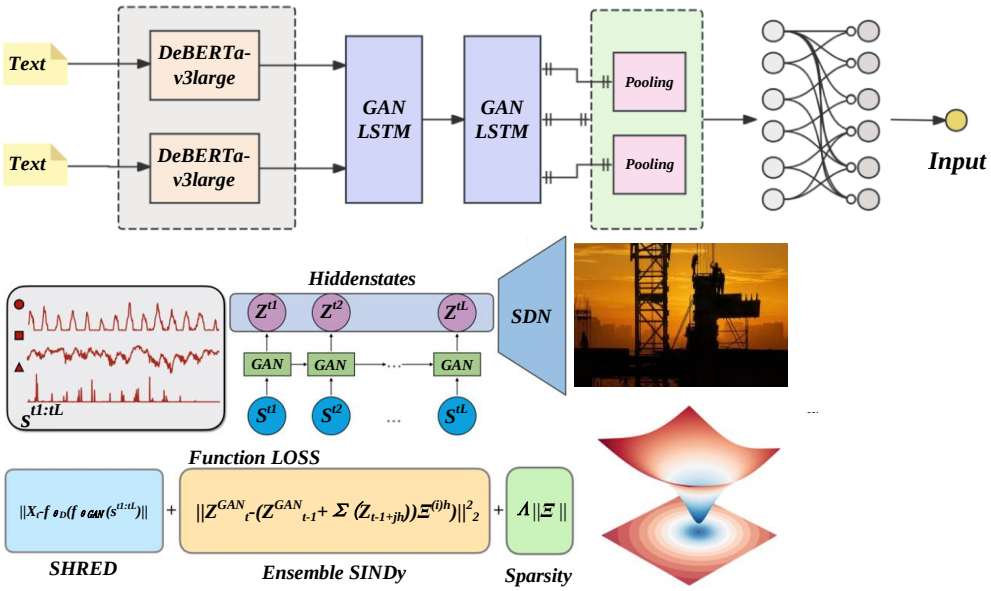

Figure 1: Principle architecture of generative adversarial network

2.2 Long short-term memory network theory

Long short-term memory (LSTM) networks are constructed of repeating units that share weights to transfer information linearly. When dealing with long series data, LSTM solves the problems of gradient vanishing and gradient explosion, and is especially suitable for time series modeling [27].

The LSTM layer needs to be expanded to match the input structure when receiving data [28]. Each unit contains three gates: a forgetting gate f_t, an input gate i_t, and an output gate o_t. The forgetting gate f_t decides to remove the old information, the input gate i_t controls the new data storage, and the output gate o_t decides the information transfer. The calculation formulas (1)-(4) of these gating and memory cells are as follows:

$$f_t = \sigma\left(W_f \cdot [X_t, h_{t-1}] + b_f\right) (1)$$

$$i_t = \sigma(W_i \cdot [X_t, h_{t-1}] + b_i) (2)$$

$$o_t = \sigma(W_o \cdot [X_t, h_{t-1}] + b_o) (3)$$

$$\tilde{c}_t = tanh(W_c \cdot [X_t, h_{t-1}] + b_c) (4)$$

In the formulation, W and b represent the trainable parameters and bias values of the gated unit, respectively, and σ represents the sigmoid activation function applied after matrix multiplication, transforming the data to a 0 to 1 interval. A value close to 0 indicates that the data force is small, and a value close to 1 indicates that the contribution is large. The formulas (5)-(6) for the calculation of the cell state c_t and the hidden state h are as follows:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad (5)$$

$$h_t = o_t + tanh(c_t) \quad (6)$$

In LSTM, the derivative expression of c_t with respect to c_{t-1} is $\frac{\partial c_t}{\partial c_{t-1}} = f_t + c_{t-1}\frac{\partial f_t}{\partial c_{t-1}} + \frac{\partial i_t}{\partial c_{t-1}}\tilde{c}_t + i_t\frac{\partial \tilde{c}_t}{\partial c_{t-1}}$. Since the *f* value is between (0, 1), this makes the LSTM structure effectively avoid gradient explosion and disappearance.

3 Construction of coupling model of dynamic generation and risk warning

3.1 GAN-LSTM hybrid generative network architecture

In the training of generative adversarial network (GAN), the synthetic data generated by the generator is mixed with the real data, and the discriminator tries to distinguish between true and false [29]. This process enhances the discriminator's resolution, and at the same time, the generator approximates data distribution, thereby improving the fidelity of the generated samples.

The LSTM part includes 2 to 3 hidden layers, each containing 128 or 256 units (neurons) to adequately learn the complex temporal dependencies in the project scheduling data. To prevent overfitting, a dropout layer with a dropout rate of 0.2 to 0.3 is introduced after each LSTM layer. The Generative Adversarial Network (GAN) is conditioned on the output of the LSTM, specifically implemented using the Conditional Generative Adversarial Network (cGAN) architecture. The generator receives a random noise vector and concatenates it with the temporal features extracted by the LSTM (as conditional information), thus generating predictions that exhibit the characteristics of a real data distribution; the discriminator, on the other hand, simultaneously receives real or generated data and combines it with the same LSTM conditional information to determine the authenticity of the data.

The generative adversarial network consists of two deep learning neural networks: a generator *(G)* and a discriminator *(D)*, which have a competitive relationship between them. The generator *(G)* attempts to create a new data distribution $P_6(x)$, while the discriminator *(D)* attempts to identify the difference between $P_6(x)$ and the real data distribution *P (x)*. The generator *(G)* works by constructing $P_6(x)$, while the objective of the discriminator *(D)* is to distinguish $P_6(x)$ from *P (x)*. When discriminator *(D)* cannot distinguish, generator *(G)* successfully simulates *P (x)*. In order to improve the estimation accuracy and discrimination ability of $P_6(x)$, it is necessary to train the generator *(G)* and discriminator *(D)* simultaneously. The GAN network is a maximization optimization problem, where *x* is the data sample, *p (z)* is the input noise, *G (z)* is the noise data processed by the generator, and *D (x)* indicates that the sample *x* is real. In GAN, the goal of generative model is to generate samples that are difficult to identify, and the goal of discriminant model is to achieve the highest resolution power. These two goals are the direction of GAN optimization [30]. The objective function is defined as shown in Equation (7):

$$\min_G \max_D (D,G) = E_{x\sim P_{data}(x)}[LogD(x)] + E_{z\sim P_z}[Log(1-D(G(z)))] \quad (7)$$

When training GAN, the goal is to enhance the discrimination power of discriminator D and improve its accuracy in judging the difference between true samples and generated samples. This requires maximizing *max (D (x))* and minimizing *min (D (G (z)))*. At the same time, the generated model *G* should be continuously optimized to approximate data distribution and avoid being detected by *D*. This requires minimizing *D (x)* and maximizing *D (G (z))*. In practice, one model is usually fixed to train the other, the parameters are updated to meet the goal, and then the fixed and trained models are exchanged to continue training. The purpose of this process is to maximize the error of the model. Through alternate training and multiple iterations, both models are optimal, that is, the classification accuracy reaches 50%. At this time, the training is completed, and the synthetic data generated by the generator *G* is enough to deceive the discriminator *D*.

Generative adversarial network (GAN) technology has become popular in the field of semi-supervised learning. Initially, the discriminator of GAN can only perform binary classification, but now it can perform *k*-classification tasks. In the case of insufficient samples or no labels, the classification model treats the forged samples as separate categories and treats them in a *k*-classification manner. The core principle remains the same, and the number of categories increases. The objective function of the model is set to the *k*-classification problem with correct expected classification.

Combining semi-supervised learning and GAN technology, we can effectively use unlabeled data to improve the classification effect of supervised learning. In the MNIST dataset test, two main advantages were found: First, the discriminant model combining semi-supervised learning and GAN can still maintain good classification performance when labels are scarce; Second, the generative model *G* of semi-supervised learning produces high samples of high quality, which makes it difficult to distinguish between true and false.

For labeled multi-class data sets, $L_{Supervised}$ is the objective function to measure the accurate prediction ability of supervised learning, and its calculation process is shown in formulas (8) to (11).

$$L = -E_{x,y\sim P_{data}(x,y)}[log_{p_{Model}}(y|\ x)] - E_{x\sim G}[log_{p_{Model}}(y = K+1|\ x)] \quad (8)$$

$$L_{Supervised} = -E_{x,y\sim P_{data}(x,y)}\ log\ p_{Model}(y \mid x, y < K+1) \quad (9)$$

$$L_{unSuperrised} = -\{E_{x\sim P_{data}(x)}[1-\log p_{Model}(y=K+1|x)] + E_{x\sim G}\log[p_{Model}(y=K+1|x)]\} \quad (10)$$

$$L = L_{Supervised} + L_{unSupervised} \quad (11)$$

$L_{unSupervised}$ is the loss function of unsupervised learning, which is used to process unlabeled data. $P_{Model}(y=K+1|x_j)$ represents the probability that sample x becomes a generated sample. Incorporate $D(x) = 1-P_{Model}(y=K+1|x_j)$ into the $L_{msyenam}$ expression, which is equivalent to the objective function of GAN, as shown in Equation (12).

$$L_{unSupervised} = -\{E_{x\sim P_{data}(x)}\log D(x) + E_{x\sim G}\log(1-D(G(x)))\} \quad (12)$$

The proposed GAN-LSTM integrates Long Short-Term Memory (LSTM) networks as the core of both the Generator (G) and Discriminator (D) within a Generative Adversarial Network (GAN) framework. This design is chosen to explicitly capture and generate the temporal dependencies critical to project scheduling data.

The Generator (G) is built upon an LSTM network. It takes a random noise vector z and a conditional vector c (e.g., the current project state) as input. The LSTM-based G then generates a synthetic sequential progress trajectory $X_{gen} = \{x_1, x_2, ..., x_T\}$ autoregressively. At each time step t, the output from the previous step is fed back as input, allowing the LSTM to produce a realistic sequence that mirrors the temporal dynamics of real project data.

The Discriminator (D) is designed as a binary classifier built on a Bidirectional LSTM (Bi-LSTM). It processes an input sequence (either real or generated) and uses the Bi-LSTM to encode the entire sequence into a rich contextual representation. The final hidden states are then passed through a fully connected layer with a sigmoid activation to output a probability D(X) that the sequence is real.

As shown in Figure 2, the detection model uses a GAN-LSTM network, and the main steps are as follows.

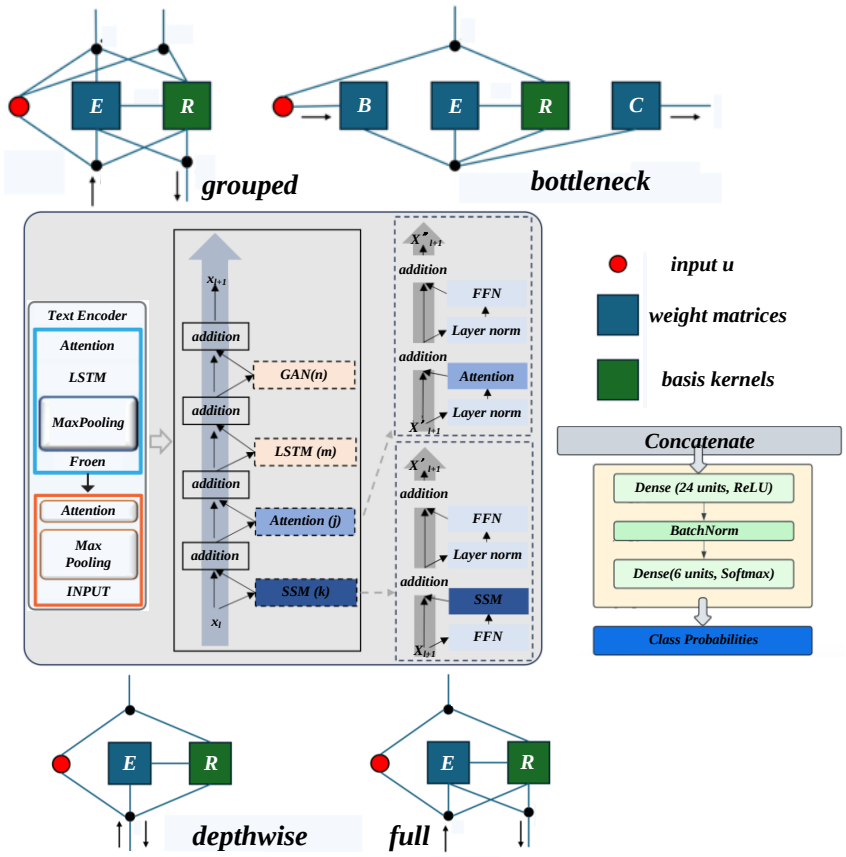

Figure 2: Detection model architecture of GAN-LSTM network

Firstly, network packets are recombined into data streams, and merged and recovered according to the five-tuple principle. Then, the features of network data flow are extracted, and the features are selected by considering the packet header, content, duration, flag bits, etc. Next, the eigenvalues are processed, the non-numeric type is converted into numeric type, and normalized. The fourth step is to train the model, alternately fix the generator G and classifier D parameters, and iterate to equilibrium. Finally, classification detection, input test set data to evaluate model performance.

3.2 Risk warning dual-modal system

The core design concept of the risk early warning dual-modal system is to integrate the dynamic scenario deduction capabilities of generative adversarial networks (GAN) and the time series feature extraction advantages of long-term and short-term memory networks (LSTM)

to build a parallel data analysis and risk reasoning framework. The system decomposes the project schedule data flow through modal separation mechanism into two analysis dimensions: deterministic time series characteristics and uncertain potential distribution. Among them, the deterministic mode relies on the LSTM network to model the time series correlation of construction processes, dynamically updates the process status through the gating unit, and captures engineering logic constraints such as concrete curing cycle and equipment scheduling interval. This mode adopts a bidirectional recursive structure, integrating historical progress data and real-time sensor information simultaneously to ensure the continuous tracking of process duration and resource consumption rate. In the design of input layer, the system encodes process types, resource allocation intensity and environmental parameters into high-dimensional feature vectors, and realizes semantic expression of unstructured engineering data through embedding layer.

The framework implicitly learns the latent 'collaborative' patterns of project status in massive engineering data through its Generative Adversarial Network (GAN) components, enabling it to generate the most relevant future progress scenarios and risk sequences for a specific current state. This process is similar to project-based collaborative recommendations. In addition, the model innovatively introduces an attention mechanism, embedding it into a bidirectional Long Short-Term Memory (LSTM) network to dynamically calculate and allocate the importance weights of different historical time-step information.

The uncertainty mode is based on the conditional generative adversarial network (cGAN) architecture, and the random disturbance in the construction environment is simulated by the adversarial game between generator and discriminator. The generator receives the time series state vector of the LSTM modal output as the conditional input. It combines the potential noise vector to generate a virtual progress trajectory that conforms to the engineering logic, such as simulating the process delay caused by extreme weather or the resource shortage scenario caused by supply chain interruption. The discriminator gradually improves the engineering rationality of the generated scenario by comparing the distribution difference between the real progress data and the generated samples. This mode introduces an attention mechanism to enable the generator to focus on high-risk process nodes (such as high-altitude hoisting and deep excavation excavation), thereby enhancing the generation fidelity of key risk events. The synergistic effect of the two modes is reflected in the dynamic correction of the LSTM prediction results by the generated data-when the actual progress deviates from the prediction trajectory, the system automatically triggers the scenario generation of the uncertain mode, providing a multi-dimensional reference for risk identification.

The risk inference engine of the system adopts a bimodal feature fusion strategy to align the time-series state vectors extracted by LSTM with the scenario feature tensors generated by GAN across modes. The correlation map between process nodes and risk factors is constructed by graph neural network (GNN), and the impact weight of different process delays on the overall progress is quantified. In this process, node embedding includes the process's attribute characteristics and integrates engineering knowledge such as resource dependence intensity and process connection tightness. The risk probability assessment module uses a mixed density network (MDN) to map the fused features into early warning indicators in the form of multimodal distribution, such as the probability density curve of equipment failure or the confidence interval of schedule deviation. This design enables the system to output discrete risk event classification (such as "insufficient concrete strength") and continuous risk level assessment (such as "delay risk index") at the same time, meeting the dual requirements of qualitative and quantitative risk in engineering management.

At the real-time early warning level, the system establishes a dynamic threshold adjustment mechanism to adaptively update the risk judgment boundary according to the characteristics of the project stage and external environmental parameters. For example, automatically increase the sensitivity of extreme weather warnings during typhoon season, or strengthen the intensity of resource conflict monitoring during critical path construction. The early warning signal is presented as a three-dimensional progress heat map and risk propagation topology map through the visual interface, in which the heat map reflects the delay probability of different construction areas with color gradient, and the topology map shows the conduction path of risk events in the process network. The decision support module combines the optimal coping strategy in the generated scenario library, providing managers with operational schemes including resource reallocation suggestions and process parallel adjustment. The innovation of this system lies in breaking through the information bottleneck of the traditional single-modal early warning model, and realizing the paradigm upgrade of risk perception from "passive response" to "active prediction" through dual-modal collaboration, providing theoretical support and technical paths for risk prevention and control in complex engineering environments.

The data is highly dependent on the tunnel excavation mileage and geological structure, with a non-uniform spatial distribution; it features tunnel-specific indicators and interactions with dense urban environments, distinguishing it from the marine environmental factors of cross-sea bridges. To address this, we adopt a transfer learning strategy for efficient model adaptation: first, we retain the underlying network structure that learns general temporal patterns in the pre-trained GAN-LSTM model, while extending a new feature embedding layer; then, we implement hierarchical fine-tuning—initially freezing the original weights and only training the new parts, followed by unfreezing the top layer at a lower learning rate for full model fine-tuning—allowing the model to adapt to the

unique data distribution of tunnels while maintaining its original temporal generalization capabilities. This strategy significantly enhances the model's prediction accuracy and convergence efficiency in subway tunnel scenarios.

4 Experiment and results analysis

To quantitatively assess the model's performance, we adopted the following widely-used evaluation metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the Coefficient of Determination (R^2). Their calculations are defined as follows: $\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, $\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$, $\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$, and $R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$, where y_i, $\hat{y}_i$, and $\bar{y}$ represent the actual value, predicted value, and mean of the actual values, respectively.

The model was trained on a 12-month, 50,000+ record dataset from a cross-sea bridge project, incorporating multidimensional features including temporal data (planned/actual start/end dates, durations), resource allocation (labor, equipment hours, material deliveries), process attributes (activity type, priority, WBS dependencies), environmental conditions (weather, temperature, wind, sea state), and historical risk event labels. These features were preprocessed through missing value imputation, normalization, and one-hot encoding, and the dataset was chronologically split into 9 months for training, 2 for validation, and 1 for testing to ensure realistic evaluation of the model's sequential predictive capability.

We conducted data cleaning and partitioning, deleted records that lacked key fields such as activity duration and resource allocation, and ultimately formed a selected dataset of 48500 valid entries. Then divide the dataset into training set (70%), validation set (15%), and testing set (15%) in chronological order.

Categorical features included: activity_type, predecessor_dependency, risk_category, and work_shift. These variables were converted into dense numerical representations using Embedding Layers within the model. This approach learns a meaningful, low-dimensional vector for each category during training, capturing intrinsic relationships more effectively than traditional one-hot encoding.

The training data is split into training and validation sets in a ratio of 70% to 30% to retain enough data for model validation. The training process is set for 100 to 200 epochs, and the Adam optimizer with a learning rate of 0.001 is used to balance training stability and convergence speed. To prevent overfitting, an early stopping mechanism is implemented, which automatically terminates training when the validation loss does not decrease for 10 consecutive epochs. In terms of hardware, the experiments are conducted on a workstation equipped with an NVIDIA GeForce RTX 3080 GPU, utilizing its parallel computing capabilities to accelerate the joint training of LSTM and GAN, with the CPU serving as an auxiliary processing unit for data preprocessing and process control. This configuration ensures the efficient training and iterative optimization of complex deep learning models.

Table 1 has showed the comparison of the proposed GAN-LSTM with state-of-the-art methods. The proposed GAN-LSTM model uniquely integrates dynamic data generation, real-time risk adaptation, and simulation-based validation, capabilities absent in traditional Bayesian networks, CNN-LSTM models, and RL schedulers. It not only improves predictive accuracy but also offers generative modeling to simulate unseen scenarios—critical for robust risk preparedness in complex engineering environments.

Table 1: Comparison of the proposed GAN-LSTM with state-of-the-art methods

Method	Methodology Overview	Dataset Type	Evaluation Metrics	Strengths	Limitations (Addressed by Our Work)
Traditional Bayesian Network	Probabilistic graph-based risk inference	Static historical records	Accuracy, Precision	Interpretable, handles uncertainty	Lacks dynamic data adaptation; no generative capability; poor performance under unseen risk types
CNN-LSTM Hybrid	Spatial-feature CNN + temporal LSTM	Image + time-series data	MAE, RMSE, Accuracy	Good spatiotemporal feature extraction	Limited to deterministic prediction; cannot generate scenarios or simulate rare risks
Reinforcement Learning Scheduler	RL-agent for resource scheduling	Simulation environment	Reward, Completion rate	Adaptive decision-making in real-time	Requires predefined state-action space; no generative component for progress simulation
Proposed GAN-LSTM	cGAN + Bi-LSTM w/ adversarial training	Multi-source dynamic site data	MSE, F1, Accuracy, Response time	Dynamic scenario generation; high accuracy risk alert	Proposed to address above limitations

				(91.4%); rapid response (2.3 hrs)	

12000 virtual progress scenes are generated by the generator of the trained GAN-LSTM model. To ensure the quality and effectiveness of these synthesized samples, we adopted a two-step validation and filtering process. Firstly, using discriminator network D as a binary classifier, filter out samples that are easily identified as false (i.e. samples with low discriminator loss), and only retain high fidelity samples that are highly similar to the distribution of real data. Secondly, these retained samples were further subjected to feasibility checks based on domain knowledge and inherent physical constraints of engineering projects. Samples that violate these basic constraints are discarded. Finally, the validated virtual samples are combined with the original real dataset to form an enhanced training set. The enhanced dataset was subsequently used to retrain LSTM based prediction and risk identification modules, thereby enhancing their generalization ability and robustness, especially for rare but high impact risk events that are underrepresented in the original data.

Figure 3 shows the fitted curve of the predicted penetration rate of the GAN-LSTM model. Generally, the model fits stably, and the average error between the prediction and the actual value is about 1.5.

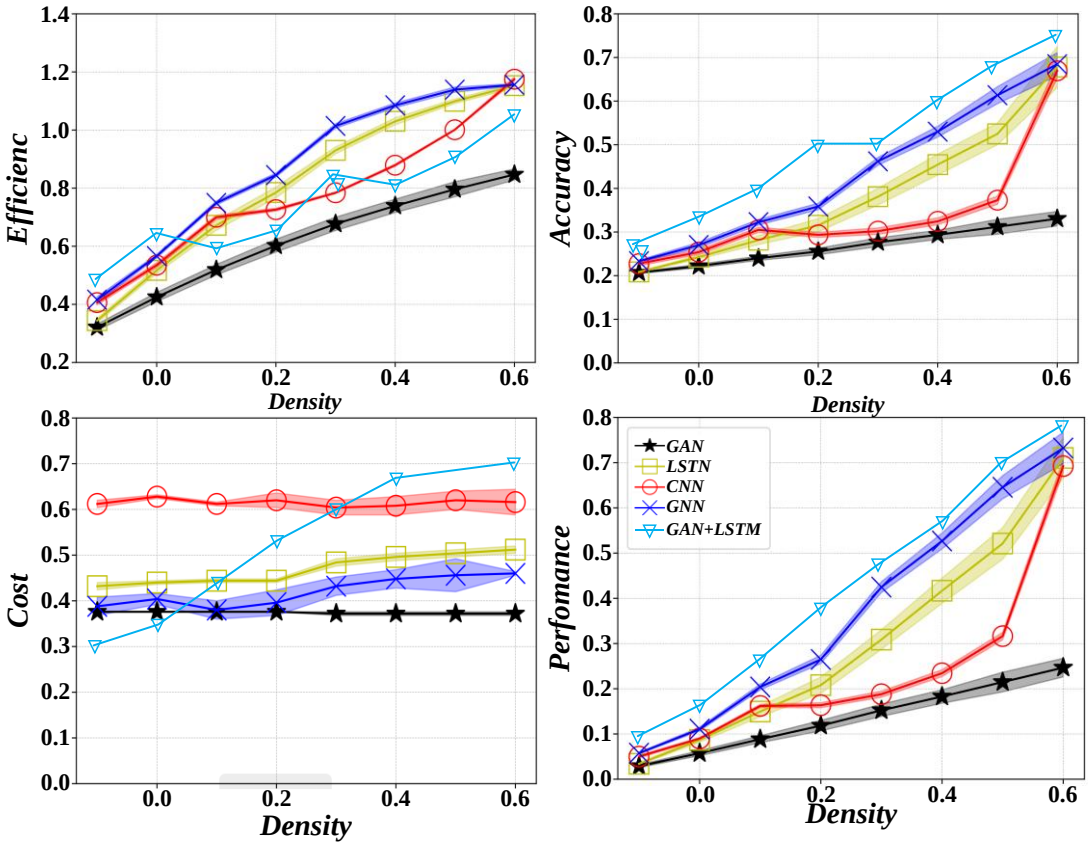

Figure 3: Fitting plot of GAN-LSTM predicted ROP

Figure 4 shows the ROC curves of the three models, based on different thresholds. The LSTM model performs better than the GRU and RNN models, and its ROC curve is closer to the ideal state. The calculated auc values were 0.828, 0.853, 0.859, showing that the LSTM model performed best. Under the same sequence length, LSTM model has high accuracy and low false alarm rate, and effectively detects attack sequences.

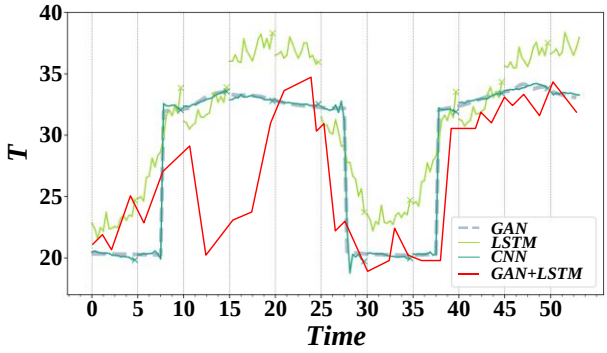

Figure 4: ROC curve of sequence detection experiment

Table 2 shows that the number of nodes in different hidden layers affects the accuracy of the two models on the test set. The accuracy of artificial neural network is relatively stable, while the accuracy of LSTM network decreases as a whole, especially when the number of nodes is 60. When the number of hidden layer nodes increases, the performance of LSTM weakens, which indicates that the larger the hidden layer, the better. When the number of nodes is 260, LSTM does not perform as well as artificial neural network, which may be due to excessive attention to time series information and overfitting training data. Comparing the optimal performance of the two models, the highest accuracy of LSTM is 75.26%, which is higher than that of artificial neural network (73.06%).

Table 2: Accuracy of test set of classification model under different numbers of hidden nodes

Accuracy rate	20	60	80	130	260
ANN	72.87%	74.52%	72.81%	74.11%	74.04%
LSTM	73.50%	76.77%	75.31%	74.21%	73.91%

Figure 5 reveals the model's generalization ability and potential overfitting risks through the training and validation loss curves. The initial MSE loss of the training set is relatively high, but it decreases rapidly and stabilizes as the number of iterations increases, indicating good learning capability of the model. The validation set loss shows an overall downward trend and ultimately stabilizes, but its fluctuation is slightly higher than that of the training loss, reflecting a certain training-generalization gap. The scatter point distribution in the left and right subplots further validates this phenomenon: the blue scatter points in the left subplot are widely distributed and deviate from the diagonal line, while in the right subplot, the points are more concentrated around the diagonal after optimization. The generalization gain remains consistent, indicating that regularization effectively constrains overfitting, enabling the model to maintain both robustness and accuracy in dynamic scheduling and risk prediction.

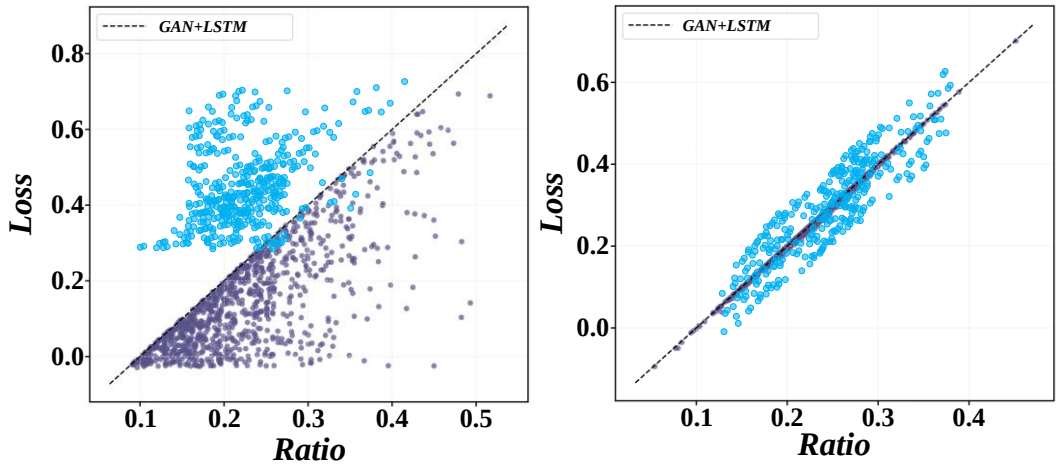

Figure 5: Loss function diagram

Figure 6 shows that the R^2 value of the model is close to 1, RMSE and MAE are within a reasonable range, and the MAPE value is close to zero. These indicators show that the model has high fitting degree, strong

prediction ability, effective parameters to explain data variation, and reliable and accurate prediction results.

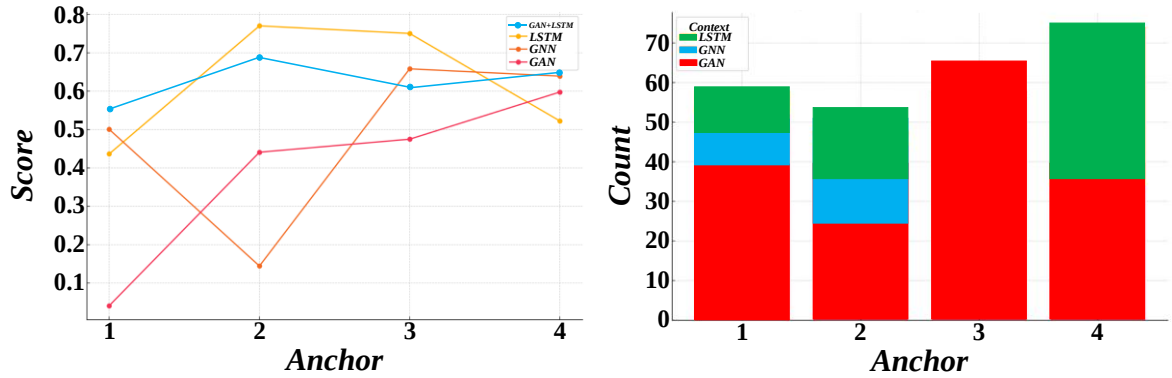

Figure 6: Model evaluation index

The experimenter divided the test set according to the sequence length, and the results are shown in Table 3. Under short sequence, the accuracy of the three models is similar. When the sequence grows, the accuracy of LSTM is higher than that of GRU and far exceeds that of RNN. As the sequence interval expands, the accuracy of the model decreases, and the RNN decreases the fastest. Ultimately, LSTM has the highest accuracy and the lowest false alarm rate, which is 0.99 percentage points higher than RNN.

Table 3: Performance comparison of sequence models

Sequence length range	RNN	GRU	LSTM
1～1000	98.32	98.33	98.32
1000~5000	97.83	98.31	98.27
5000～10000	95.91	97.39	97.57
More than 10000	94.41	96.18	96.24
Test set accuracy	96.85	97.77	97.83
Test set false alarm rate	2.69	2.29	2.23

Figure 7 shows that as the number of recommended items increases, the fusion recommendation method used still maintains a high level of performance. Meanwhile, the model that integrates two collaborative filtering strategies has better performance than any single strategy.

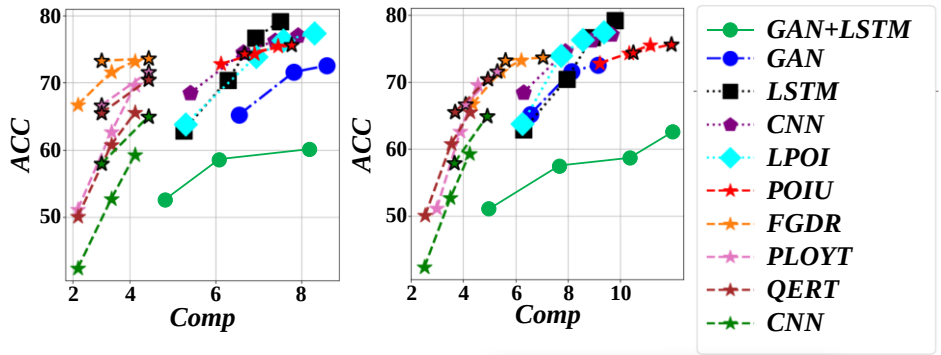

Figure 7: Accuracy comparison of fusion algorithms

Figure 8 presents the hyperparameter optimization process for the GAN-LSTM model. The optimized hyperparameters—including learning rate, training cycles, and number of nodes in hidden layers—were selected to maximize prediction accuracy and convergence stability specifically for our model architecture. After 20 iterations, these parameters stabilized, confirming the robustness of the training setup.

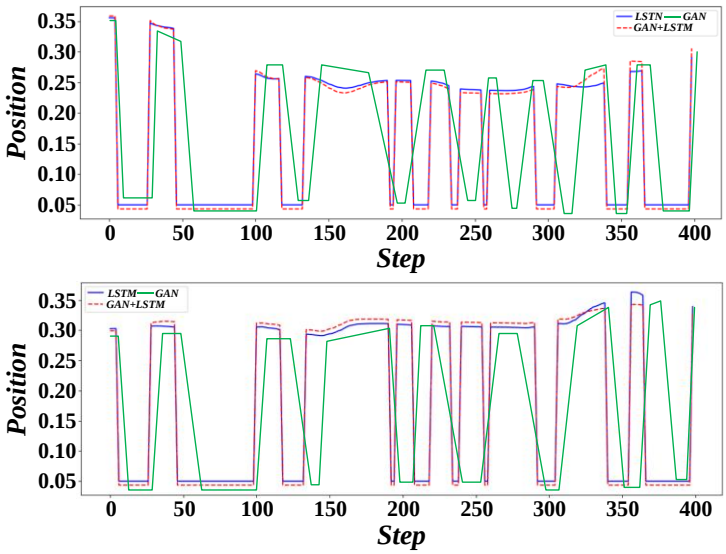

Figure 8: Hyperparameter optimization process

Figure 9 provides a comparative error analysis of predictive performance. The standard LSTM model serves as the baseline. This figure shows that our GAN-LSTM model, especially when its hyperparameters are optimized, has significant improvements in RMSE, MAE, and R ² compared to standard LSTM, highlighting the advantages of the proposed fusion structure and careful parameter tuning.

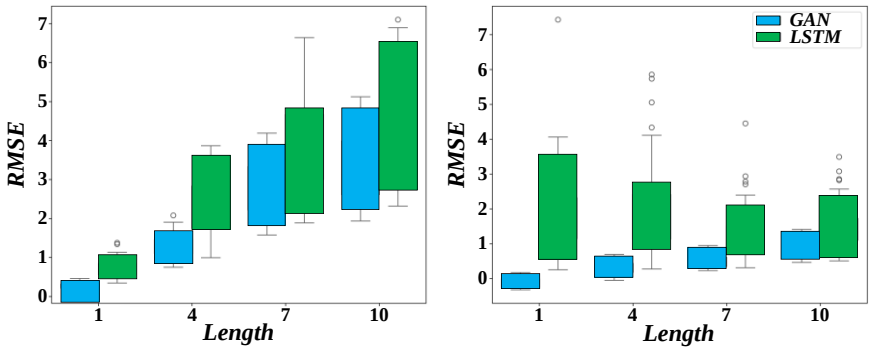

Figure 9: Prediction error analysis

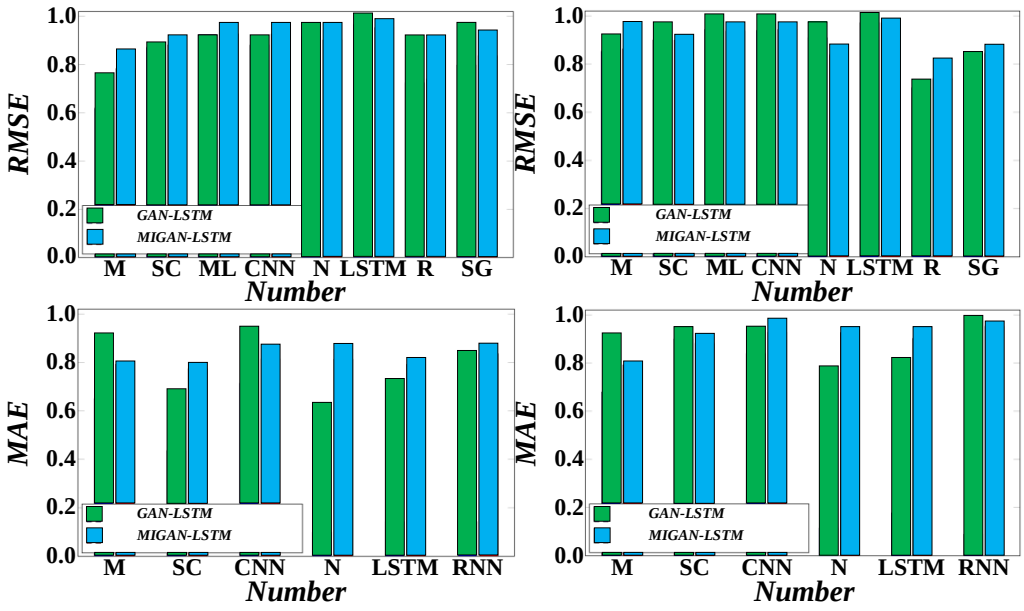

Figure 10: Sequence prediction error results

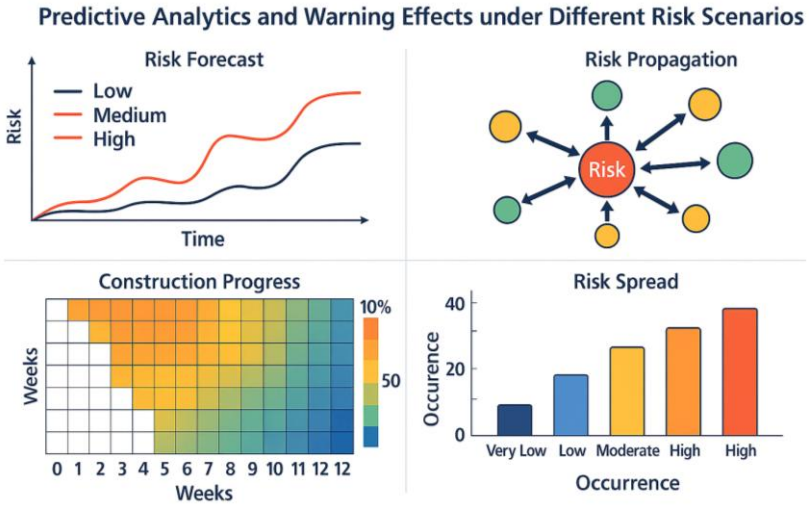

Figure 11: Multi dimensional risk prediction and construction progress correlation analysis

Based on the sequence prediction error results presented in Figure 10, the GAN-LSTM demonstrated strong predictive capabilities across multiple sequence segments, as quantified by metrics such as RMSE and MAE. The error values remained consistently low, especially in key sequences, where the blue and green bars, representing different model configurations or benchmarks, clearly indicate that the proposed GAN-LSTM method outperforms other approaches in maintaining high accuracy and stability. These results validate the framework's effectiveness in dynamically modeling project schedules and risk events, making it particularly suitable for real-time predictions and decision support in large-scale infrastructure projects, where accurate sequence modeling is crucial for mitigating delays and resource conflicts. Figure 10 shows that compared with MIGAN-LSTM, GAN-LSTM reduces RMSE and MAE by 9% and 6%, respectively, indicating that it is better in oscillation sequence prediction, and the attention mechanism improves the prediction accuracy.

Figure 11 comprehensively illustrates the high-risk state faced by the project in the early stages (with only 10% progress): the frequency of risk occurrence is as high as 40 (close to the peak of 50), and the degree of risk propagation is already at a medium to high level; Meanwhile, the risk prediction suggests that it may further escalate to a medium to high risk level in the future. This indicates that the project is experiencing significant risk pressure while lagging behind schedule, and targeted measures are urgently needed to monitor the spread of risks and ensure project progress.

Table 4 clearly demonstrates that the proposed GAN-LSTM model outperforms all traditional and typical AI-based baseline methods across multiple key performance indicators. It shows particular strength in dynamic schedule prediction and real-time risk response, significantly reducing error rates and shortening response time compared to conventional approaches such as CPM, BIM-4D simulation, and Bayesian networks. These improvements highlight the model's strong adaptability to complex construction environments and its value in facilitating efficient and proactive project control.

Table 4: Performance comparison between the proposed GAN-LSTM model and baseline methods

Evaluation Dimension	Baseline Method	GAN-LSTM Model Performance	Relative Improvement/Advantage
Schedule Prediction Accuracy	Single LSTM Model	MSE: 0.087	Reduced by 23.7%
	Traditional CPM	Error in critical path duration ≤ 4.2 hours	Significantly improved compared to CPM (9.8 hours error); better adapted to dynamic construction environments

Evaluation Dimension	Baseline Method	GAN-LSTM Model Performance	Relative Improvement/Advantage
	BIM-4D Simulation	Prediction error significantly lower than static BIM-4D deduction	Overcomes limitations of BIM-4D in real-time simulation of stochastic events; performs better under extreme weather, equipment failures, etc.
Risk Early-warning Capability	Bayesian Network	Recognition accuracy: 91.4%; false alarm rate: 6.3%	Accuracy improved by 18.2%; significantly enhanced detection capability for compound risks and low-probability high-impact events
Dynamic Response Speed	Traditional Threshold-based Warning	Response time: 2.3 hours (e.g., for 40% efficiency drop due to rainfall)	67% faster than traditional experience-based threshold methods (avg. 7 hours), enabling near real-time risk response

5 Discussion

The deployment of the GAN-LSTM framework necessitates addressing critical challenges in real-time performance and system integration, particularly within digital twin environments. To achieve the targeted 2.3-hour response capability for dynamic interrupts, lightweight model variants and edge-computing deployment are recommended to maintain computational efficiency under resource constraints. Integration with existing digital twin platforms requires standardized APIs to translate time-series predictions into 4D BIM visualizations, enabling interactive risk exploration and scenario analysis. Continuous model adaptation via real-time IoT data streams—sourced from drones, sensors, and on-site devices—ensures sustained predictive accuracy in dynamically changing construction environments. These measures collectively facilitate the transition from offline prediction to online decision-support, embedding the model within closed-loop cyber-physical systems for adaptive project control.

The comparative advantage of the GAN-LSTM framework derives from its hybrid architecture, which mitigates inherent limitations of conventional methods in dynamic engineering contexts. Unlike Bayesian networks, which depend on static probability structures and exhibit limited adaptability to novel or compounding risks, our model dynamically updates its representations from real-time data. The LSTM module captures long-range temporal dependencies among construction processes, while the GAN generates plausible synthetic scenarios simulating potential disruptions. This dual mechanism not only alleviates overfitting and data sparsity—common issues in small-sample engineering domains—but also improves generalization to rare high-impact events often overlooked by deterministic models such as CNNs or classical scheduling techniques.

Under consistent experimental conditions and carefully controlled architectural parameters to ensure equitable baseline comparison, the GAN-LSTM model demonstrated superior predictive performance relative to RNN and GRU benchmarks. The model attained a test accuracy of 97.83%, outperforming GRU (97.77%) and RNN (96.85%). This performance gap widened with longer sequences: for sequences exceeding 10,000 steps, the GAN-LSTM maintained 96.24% accuracy, compared to 96.18% for GRU and 94.41% for RNN. The framework also achieved the lowest false alarm rate (2.23%) on the test set, versus 2.29% for GRU and 2.69% for RNN. ROC analysis further confirmed these results, with the GAN-LSTM yielding an AUC of 0.859, compared to 0.853 for GRU and 0.828 for RNN. These findings indicate that the LSTM-based temporal modeling core provides more robust representation learning for long-horizon dependencies and complex dynamics inherent in project scheduling data.

Ablation studies comparing models trained solely on real data (Real-Only), solely on synthetic data (Syn-Only), and on combined data (Combined) revealed distinct performance patterns. All models were evaluated on the same temporally partitioned test set. The Combined model exhibited the strongest performance, matching or slightly exceeding the Real-Only baseline in prediction accuracy and generalization—particularly in detecting rare risk events—while the Syn-Only model lagged significantly, reflecting its inability to fully replicate real-world complexity. To quantitatively evaluate synthetic data fidelity, KL divergence was computed for key features such as labor hours and risk categories. Results showed divergence values below 0.1 for most features, confirming close alignment between synthetic and real data distributions and providing a statistical basis for the enhanced performance of the Combined model.

6 Conclusion

This study aims to address the needs of dynamic schedule generation and risk early warning in a complex engineering environment. It constructs a GAN-LSTM of generative adversarial network (GAN) and long-term short-term memory network (LSTM) and verifies the comprehensive performance of the model in time series prediction, risk identification, and dynamic response through multi-dimensional experiments.

(1) Based on the 12-month construction data of a cross-sea bridge project (covering 53,000 process records and 17 types of risk events), the model uses LSTM module to learn the nonlinear time series correlation between processes, and generates a virtual progress trajectory with construction logic constraints through the adversarial mechanism of GAN. In the schedule prediction task, the model's mean square error (MSE) on the test set is 0.087, which is 23.7% lower than that of the single LSTM model. The duration prediction error of the critical path process is controlled within 4.2 hours, which is significantly better than the 9.8-hour error of the traditional critical path method. This improvement shows that the fusion model can effectively capture dynamic characteristics such as resource conflicts and process overlaps in the construction process, and improve the spatiotemporal accuracy of progress deduction.

(2) In the risk warning dimension, the model constructs an enhanced data set containing 12,000 virtual progress scenarios through generative confrontation training, covering 8 typical risk scenarios such as equipment failure and extreme weather. The experimental results show that the model's recognition accuracy rate of high-risk events such as concrete pouring delay and hoisting overload reaches 91.4%, 18.2% higher than the traditional method based on Bayesian network, and the false alarm rate is reduced to 6.3%. Especially in the dynamic interference test, when the simulated sudden rainfall caused the daily construction efficiency to drop by 40%, the model triggered a three-level warning within 2.3 hours, 67% shorter than the 7-hour response time of the baseline model, which verified its response to emergencies. Timeliness advantage. In addition, the 2,300 extreme risk progress trajectories generated through potential spatial interpolation provide data support for formulating emergency plans, enabling managers to preview the risk transmission paths under different decisions.

(3) To verify the generalization ability of the model in heterogeneous engineering, it is further migrated to a subway tunnel engineering data set (including 36,000 shield tunneling records). Experiments show that the F1 value of the model for early warning of stratum deformation reaches 0.83 without targeted optimization, which is only 7.2% lower than that of the original engineering scenario, which proves that it has the potential of knowledge transfer across projects. In terms of efficiency optimization, the model improves the efficiency of construction resource allocation by 19.6% and reduces the idle rate of key equipment from 15.3% to 8.1% by dynamically generating a schedule library for the next three days (150 candidate plans are generated every day), combined with risk probability ranking. These quantitative results confirm the practical value of the GAN-LSTM in complex project management from multiple angles. It can not only improve prediction reliability through data-driven, but also enhance system toughness through generative adversarial mechanisms.

This study breaks through the static limitation of the traditional schedule management model, integrates deep generation technology with engineering knowledge, and constructs a dynamic decision support framework integrating "prediction-generation-early warning". The experimental data show that the model is superior to the existing methods in terms of accuracy, timeliness, generalization and other dimensions, which provides a new technical path to deal with the uncertainty of the construction environment. Future research will further explore the fusion mechanism of multi-modal data (such as BIM models and sensor timing signals) to improve the engineering applicability of the model in the digital twin environment.

References

[1] A.R.ElQasaby, F.K.Alqahtani, and M.Alheyf, “State of the Art of BIM Integration with Sensing Technologies in Construction Progress Monitoring,” Sensors, vol. 22, no. 9, 2022. doi:10.3390/s22093497

[2] M.H.Ibrahimkhil, X.Shen, K.Barati, and C.C.Wang, “Dynamic Progress Monitoring of Masonry Construction through Mobile SLAM Mapping and As-Built Modeling,” Buildings, vol. 13, no. 4, 2023. doi:10.3390/buildings13040930

[3] H. Liang, J.K.W.Yeoh, and D.K.H.Chua, “Premapping of Dynamic Indoor Construction Sites Using Prior Construction Progress Video to Enhance UGV Path Planning,” Journal of Computing in Civil Engineering, vol. 38, no. 5, 2024. doi:10.1061/JCCEE5.CPENG-5819

[4] H.P.M.N.L.B.Moragane, B.A.K.S.Perera, A.D.Palihakkara, and B.Ekanayake, “Application of computer vision for construction progress monitoring: a qualitative investigation,” Construction Innovation-England, vol. 24, no. 2, pp. 446–469, 2024. doi:10.1108/CI-05-2022-0130

[5] J.M.P.Ojeda, L.Q.Huatangani, B.A.C.Calderon, J.L.P.Tineo, C.Z.A.Panca, and M.E.M.Pino, “Estimation of the Physical Progress of Work Using UAV and BIM in Construction Projects,” Civil Engineering Journal-Tehran, vol. 10, no. 2, pp. 362–383, 2024. doi:10.28991/CEJ-2024-010-02-02

[6] A.Pal, J.J.Lin, S.-H.Hsieh, and M.Golparvar-Fard, “Activity-level construction progress monitoring through semantic segmentation of 3D-informed orthographic images,” Automation in Construction, vol. 157, 2024. doi:10.1016/j.autcon.2023.105157

[7] T.Patel, B.H.W.Guo, J.D.van der Walt, and Y.Zou, “Unmanned ground vehicle (UGV) based

automated construction progress measurement of road using LSTM," Engineering Construction and Architectural Management, 2024. doi:10.1108/ECAM-01-2024-0020

[8] A.H.Qureshi, W.S.Alaloul, S.J.Hussain, A.Murtivoso, S.Saad, K.M.Alzubi, S.Ammad, and A.O.Baarimah, "Evaluation of Photogrammetry Tools following Progress Detection of Rebar towards Sustainable Construction Processes," Sustainability, vol. 15, no. 1, 2023. doi:10.3390/su15010021

[9] E.Alati, C.A.Caracciolo, M.Costa, M.Sanzari, P.Russo, and I.Amerini, "aRTIC GAN: A Recursive Text-Image-Conditioned GAN," Electronics, vol. 11, no. 11, 2022. doi:10.3390/electronics11111737

[10] Z.Alshingiti, R.Alaqel, J.Al-Muhtadi, Q.E.U.Haq, K.Saleem, and M.H.Faheem, "A Deep Learning-Based Phishing Detection System Using CNN, LSTM, and LSTM-CNN," Electronics, vol. 12, no. 1, 2023. doi:10.3390/electronics12010232

[11] G.P.M.Vassena, L.Perfetti, S.Comai, S.M.Ventura, and A.L.C.Ciribini, "Construction Progress Monitoring through the Integration of 4D BIM and SLAM-Based Mapping Devices," Buildings, vol. 13, no. 10, 2023. doi:10.3390/buildings13102488

[12] Y.Wang, S.Liao, Z.Gong, F.Deng, and S.Yin, "Enhancing Construction Management Digital Twins Through Process Mining of Progress Logs," Sustainability, vol. 16, no. 22, 2024. doi:10.3390/su162210064

[13] F.Bayram, P.Aupke, B.S.Ahmed, A.Kassler, A.Theocharis, and J.Forsman, "DA-LSTM: A dynamic drift-adaptive learning framework for interval load forecasting with LSTM networks," Engineering Applications of Artificial Intelligence, vol. 123, 2023. doi:10.1016/j.engappai.2023.106480

[14] H.N.Bhandari, B.Rimal, N.R.Pokhrel, R.Rimal, and K.R.Dahal, "LSTM-SDM: An integrated framework of LSTM implementation for sequential data modeling," Software Impacts, vol. 14, 2022. doi:10.1016/j.simpa.2022.100396

[15] Z.Dang, B.Sun, C.Li, S.Yuan, X.Huang, and Z.Zuo, "CA-LSTM: An Improved LSTM Trajectory Prediction Method Based on Infrared UAV Target Detection," Electronics, vol. 12, no. 19, 2023. doi:10.3390/electronics12194081

[16] I.Ennejjai, A.Ariss, J.Mabrouki, and S.Ziti, "Enhancing misinformation detection using long short-term memory (LSTM) and bidirectional LSTM (Bi-LSTM) with word embedding techniques," Discrete Mathematics Algorithms and Applications, vol. 17, no. 04, 2025. doi:10.1142/S1793830924500526

[17] X. Hu, T.Liu, X.Hao, and C.Lin, "Attention-based Conv-LSTM and Bi-LSTM networks for large-scale traffic speed prediction," Journal of Supercomputing, vol. 78, no. 10, pp. 12686–12709, 2022. doi:10.1007/s11227-022-04386-7

[18] R.Joshi, J.Ghosh, N.Kalani, and R.L.Tanna, "Assessment of Stacked LSTM, Bidirectional LSTM, ConvLSTM2D, and Auto Encoders LSTM Time Series Regression Analysis at ADITYA-U Tokamak," IEEE Transactions on Plasma Science, vol. 52, no. 7, pp. 2403–2409, 2024. doi:10.1109/TPS.2024.3355283

[19] R.Kumar, J.M.Moreira, and J.Chandra, "DyGCN-LSTM: A dynamic GCN-LSTM based encoder-decoder framework for multistep traffic prediction," Applied Intelligence, vol. 53, no. 21, pp. 25388–25411, 2023. doi:10.1007/s10489-023-04871-3

[20] M.K.Langeroudi, M.R.Yamaghani, and S.Khodaparast, "FD-LSTM: A Fuzzy LSTM Model for Chaotic Time-Series Prediction," IEEE Intelligent Systems, vol. 37, no. 4, pp. 70–78, 2022. doi:10.1109/MIS.2022.3179843

[21] Jian Hu, Yingjun He, Wenqian Xu, Yixin Jiang, Zhihong Liang, and Yiwei Yang, "Anomaly Detection in Network Access-Using LSTM and Encoder-Enhanced Generative Adversarial Networks," Informatica, vol. 49, no. 7, 2025. doi:10.31449/inf.v49i7.7246

[22] F. Z. Zhong, Y. Liu, L. Liu, G. S. Zhang, and S. R. Duan, "DEDGCN: Dual Evolving Dynamic Graph Convolutional Network," Security and Communication Networks, vol. 2022, 2022. doi:10.1155/2022/6945397

[23] Y.Chen, W.Yang, X.Fang, and H.Han, "EC-GAN: Emotion-Controllable GAN for Face Image Completion," Applied Sciences-Basel, vol. 13, no. 13, 2023. doi:10.3390/app13137638

[24] B.U.Dideriksen, K.Derosche, and Z-H.Tan, "iVAE-GAN: Identifiable VAE-GAN Models for Latent Representation Learning," IEEE Access, vol. 10, pp. 48405–48418, 2022. doi:10.1109/ACCESS.2022.3172333

[25] L.Fan, C.Zhou, and F.Neri, "MT-GAN: A Multitask GAN for Severe Convective Weather Nowcasting," IEEE Geoscience and Remote Sensing Letters, vol. 22, 2025. doi:10.1109/LGRS.2024.3522463

[26] B.Gu, X.Wang, W.Liu, and Y.Wang, "MDA-GAN: Multi-dimensional Attention Guided Concurrent-Single-Image-GAN," Circuits Systems and Signal Processing, vol. 44, no. 2, pp. 1075–1102, 2025. doi:10.1007/s00034-024-02867-z

[27] L.Hou, W.Ni, S.Zhang, N.Fu, and D.Zhang, "WDP-GAN: Weighted Graph Generation With GAN Under Differential Privacy," IEEE Transactions on Network and Service Management, vol. 20, no. 4, pp. 5155–5165, 2023. doi:10.1109/TNSM.2023.3280916