Journal of Electronic Research and Application, 2026, Volume 10, Issue 4
http://ojs.bbwpublisher.com/index.php/JERA
Online ISSN: 2208-3510
Print ISSN: 2208-3502

# Deep Learning Prediction Model for Dynamic Response of Bridge Cranes

**Guanlan Li[1]*, Zhaoqing Guan[1], Qiang Liu[2], Wenqing Yang[3], Siqun Ma[1]**

[1]Zhan Tianyou College of Dalian Jiaotong University(CRRC College), Dalian 116028, Liaoning, China
[2]Yantai Vocational College of Automotive Engineering, Yantai 265500, Shandong, China
[3]Zhengzhou Depot of China Railway Zhengzhou Group Co., Ltd. Zhengzhou 450000, Henan, China

***Corresponding author*:** Guanlan Li, 903131322@qq.com



**Abstract:** The transient dynamic response of a bridge crane's lifting mechanism is critical for operational safety and structural fatigue life. While traditional multi-body dynamics simulations offer high fidelity, their substantial computational cost hinders real-time analysis in digital twin applications. To overcome this bottleneck, this paper proposes a deep learning surrogate model based on a Long Short-Term Memory (LSTM) network for rapid prediction of the transient dynamics in a double-girder bridge crane's lifting system. First, a high-fidelity dynamic benchmark model incorporating wire rope flexibility and contact friction is developed in ADAMS. Second, a high-quality dataset of 400 samples is constructed via Latin Hypercube Sampling, covering variations in load, lifting height, speed, and acceleration. Third, a three-layer encoder-LSTM-decoder network is designed and trained using a cosine annealing learning rate schedule and the AdamW optimizer. Experimental results demonstrate that the proposed model achieves excellent prediction accuracy, with a normalized mean absolute error (NMAE) of 0.0431, a normalized root mean square error (NRMSE) of 0.0681, and an average peak relative error of 4.72%, meeting engineering requirements. Most notably, the prediction time is reduced from approximately 30 minutes per simulation to 300 milliseconds, representing a computational efficiency improvement by a factor of about 6000 compared to conventional dynamic simulation.

**Keywords:** Bridge crane; Transient dynamics simulation; Deep learning; Long short-term memory network

**Online publication:** May 21, 2026

## 1.Introduction

As core equipment in modern industrial production, port logistics, and construction, the operational safety and reliability of bridge cranes directly impact production efficiency, personnel safety, and economic benefits. With the ongoing advancement of intelligent technology and digital twins, their integration into engineering applications is becoming increasingly widespread. Within digital twin systems, basic functions such as real-time data mapping and visual monitoring typically rely on high-cost sensors and controllers for

effective implementation, while more advanced functions like real-time simulation and future prediction often necessitate even greater investments in hardware and software. However, applying digital twin systems to complex dynamic systems like cranes faces a significant technical bottleneck: high-fidelity physical simulation models are usually computationally intensive and time-consuming, making it difficult to meet the stringent requirements for millisecond-level synchronization, prediction, and decision feedback regarding crane operational status [1]. To address this challenge, various research approaches have been proposed. For example, Professor Van-Hai Trinh introduced a surrogate model method based on Karhunen-Loève expansion and neural networks for predicting the dynamic response of a bridge crane's main girder in a high-dimensional design space [2]. Zhang proposed a hybrid modeling method using a physics-informed neural network (PINN) to tackle the issues of massive real-time sensor data and complex dynamic coupling in operating tower cranes [3]. Building upon these efforts and focusing specifically on the dynamic response prediction of the rope system within the lifting mechanism, this paper proposes a dynamic data prediction method based on a Long Short-Term Memory (LSTM) deep neural network model.

## 2.ADAMS-based dynamic model

The double-girder bridge crane features a complex overall structure and operates under diverse working conditions. To streamline the computational and modeling efforts, this study focuses exclusively on the lifting mechanism as a decoupled subsystem. This focus allows for a dedicated investigation into the dynamic response characteristics of the ropes and the suspended load under various operational scenarios. The data generated from this analysis form the basis for establishing the prediction model. The basic parameters of the targeted lifting mechanism are summarized in **Table 1**, and its three-dimensional representation is provided in **Figure 1**.

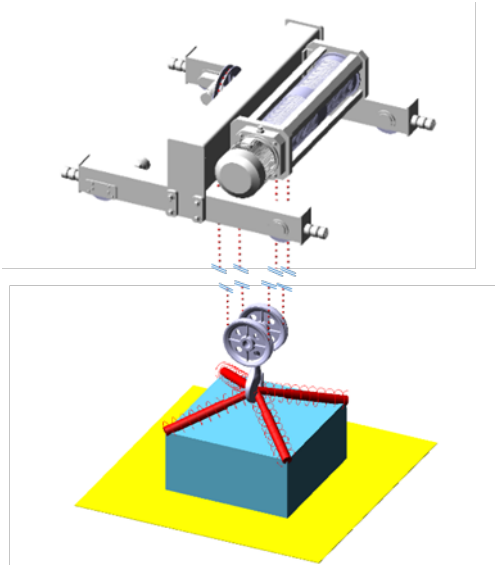


**Figure 1.** 3D model of the lifting mechanism.

**Table 1.** Basic parameters of the lifting mechanism

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Rated load | 10 t | Steel wire rope specification | 6x19W + FC-16.0 |
| Lifting speed - Low | 4 m/min | Drum size | Ø304.5×1106 |
| Lifting speed - Medium | 7.6 m/min | Rated lifting height | 7.2 m |
| Lifting speed - High | 16 m/min | Number of sheaves | 3 |

## 2.1. Discretized rope model

To accurately simulate the dynamic behavior of a flexible rope, the rope is discretized into a series of units connected via bushing forces. While such discrete units are typically modeled as cylinders, cylindrical geometries often suffer from poorer numerical convergence in friction calculations compared to their spherical counterparts. Given that the interaction between the pulley and the rope involves static friction, and the detailed frictional conditions are not the primary focus of this study, spherical units are adopted in this model to enhance computational efficiency. Nevertheless, to ensure the simulation accurately captures the rope's dynamic response, it remains essential to derive the correct geometric parameters for these discrete units. Following Chinese National Standard GB 8918-2006, Steel Wire Ropes for Important Purposes, a three-dimensional model of the steel wire rope was created, from which the mass and moments of inertia along each principal axis were measured. The resulting parameters for the rope's discrete units are listed in **Table 2**, and the corresponding three-dimensional model is illustrated in **Figure 2**.

**Figure 2.** 3D model of the lifting mechanism.

The anisotropic stiffness parameters of the bushing are approximated using **Equation (1)**, and the anisotropic damping values are taken as 1% of the stiffness values based on empirical data:

$$\begin{cases} K_{11} = \dfrac{EA}{L} \\ K_{22} = K_{33} = \dfrac{GA}{L} \\ K_{44} = \dfrac{G\pi D^4}{32L} \\ K_{55} = K_{66} = \dfrac{E\pi D^4}{64L} \end{cases} \tag{1}$$

where E is the Young's modulus of the material, G is the shear modulus, A is the cross-sectional area of the discrete element, L is its height, and D is the equivalent diameter.

**Table 2.** Basic parameters of the lifting mechanism

| Parameters | Value | Parameters | Value |
|---|---|---|---|
| Density | 7860 kg/m$^3$ | Poisson's ratio | 0.28 |
| Quality | 0.56 kg | Moment of Inertia - x | 47.289 kg·mm$^2$ |
| Height | 50 mm | Moment of Inertia - y | 47.289 kg·mm$^2$ |
| Young's modulus | 6.02E+05 Mpa | Moment of Inertia - z | 1.476 kg·mm$^2$ |

## 2.2. Contact and constraint

In ADAMS, the joint constraints and force elements between components are defined based on practical conditions, and the driving motion is imposed on the revolute joint of the drum. In this study, the lifting attachment is modeled as a hook. During actual operation, the load is suspended from the hook by a rope. Since the rope is not the primary focus of the simulation, and in most scenarios only its axial force needs to be considered, it is simplified as a spring-damper element that transmits only axial tension and compression forces. To ensure the simulation reflects real-world conditions, the damper parameters are set to match the stiffness and damping of a rope segment whose total length equals that of the discretized rope units in the model.

In ADAMS, a modified friction model is used, which accounts for static friction, Coulomb friction, and viscous friction, thereby providing a more accurate description of the transition from static to kinetic friction [4]. To bring the simulation closer to real-world results, the contact stiffness is calculated based on Hertz contact theory using **Equation (2)** [5]:

$$\begin{cases} k = \left| \frac{4}{3} E \cdot \sqrt{\frac{R_1 R_2}{R_1 + R_2}} \right| \\ \frac{1}{E} = \frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \end{cases} \tag{2}$$

where E is the equivalent elastic modulus of the two contacting objects, $R_1$ and $R_2$ are the radii of curvature at the contact position, and $E_1$, $\nu_1$, $E_2$, $\nu_2$ are the elastic moduli and Poisson's ratios of the two objects, respectively.

## 2.3. Simulation of typical operating conditions

Dynamic simulations were conducted under typical rated and overload conditions. For each condition in ADAMS, the initial state was configured either by direct modification or via a brief pre-simulation. The driving input for each condition is defined as a product of a maximum magnitude and a STEP function, which provides a smooth, twice-differentiable transition between constant values, a suitable form for crane dynamics. The drive/brake function profiles for low, medium, and high speeds are presented in **Figure 3**. **Figure 4** shows the velocity variation curve and the rope dynamic response within 3 seconds of lifting from rest on the ground under the rated load condition.

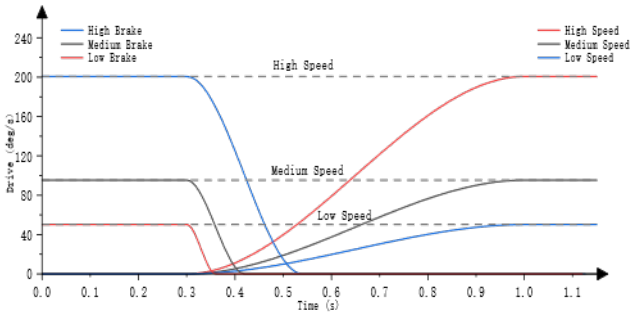
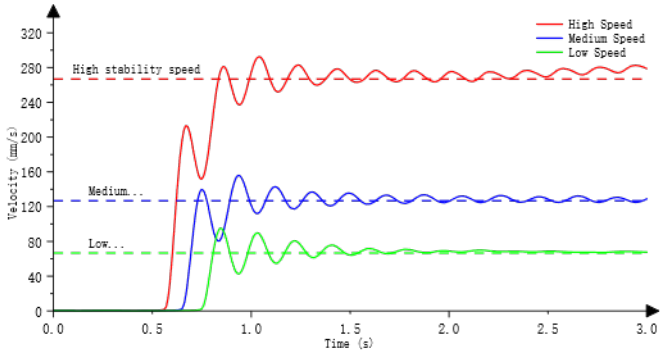
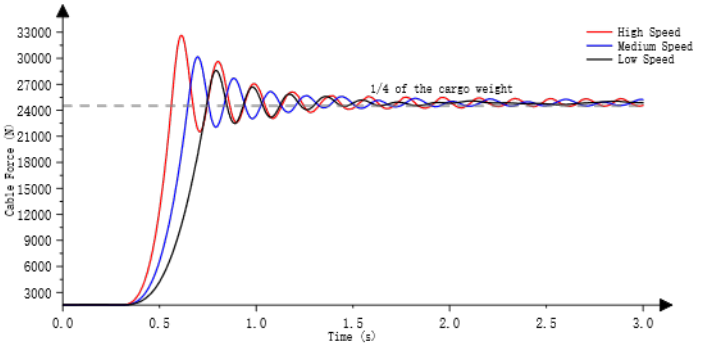



**Figure 3.** Typical operating condition driving curve.




**(a) Cargo** speed.




**(b) Cable** force.

**Figure 4.** Typical lifting condition response curve.

## 3. Deep learning prediction model

To process sequential data and capture temporal dependencies, Recurrent Neural Networks (RNN) were introduced. However, due to structural limitations, standard RNNs often struggle with the vanishing and exploding gradient problems during training on long sequences. As an enhanced variant of RNN, the LSTM network effectively mitigates these issues, enabling the learning of long-term dependencies [4]. Consequently, LSTM is particularly well-suited for dynamic response prediction scenarios characterized by complex nonlinear temporal relationships.

## 3.1. Introduction to LSTM

As shown in **Figure 5**, the LSTM cell introduces three gating mechanisms, the forget gate, input gate, and output gate, to regulate information flow. The forget gate decides which prior cell state information to discard; the input gate governs the storage of new information; and the output gate determines the subsequent hidden state. These gates collectively form the core of LSTM's ability to capture long-range dependencies. In the figure, Forget, Input, and Output represent the forget, input, and output gates, respectively, with the update functions $f_t$, $i_t$, and $o_t$ given by Equation (3):

$$\begin{cases} f_t = \sigma\left(W_f \cdot \left[h_{t-1}, X_t\right] + b_f\right) \\ i_t = \sigma\left(W_i \cdot \left[h_{t-1}, X_t\right] + b_i\right) \\ O_t = \sigma\left(W_o \cdot \left[h_{t-1}, X_t\right] + b_o\right) \end{cases} \tag{3}$$

where: σ represents the sigmoid activation function; $W_f$, $W_i$, $W_o$ correspond to the weight matrices of the three gates, and $b_f$, $b_i$, $b_o$ correspond to the biases; $X_t$ is the input value, $h_{t-1}$ is the hidden state of the previous time step, and $[h_{t-1}, X_t]$ is a combined vector.

Furthermore, in **Figure 5**, tanh denotes the hyperbolic tangent activation function, $h_t$ represents the output state at the current time step, $C_t^{'}$ is the candidate memory for the cell state, $C_t$ is the cell state at the current time step, and ⊙ denotes the Hadamard product (element-wise product). Their expressions are given by **Equation (4)**:

$$\begin{cases} C_t^{'} = tanh\left(W_c \cdot \left[h_{t-1}, X_t\right] + b_c\right) \\ C_t = f_t \odot C_{t-1} + i_t \odot C_t^{'} \\ h_t = O_t \odot tanh C_t \end{cases} \tag{4}$$

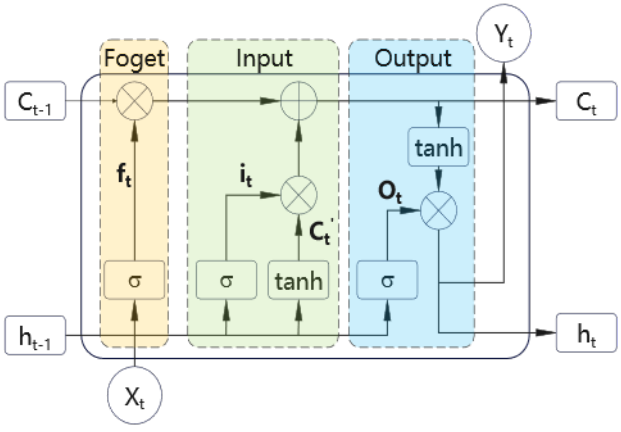



**Figure 5.** Structure of the LSTM cell.

## 3.2. Prediction model construction

A three-hidden-layer network is constructed based on the LSTM cell and the input-output characteristics of our dataset, as illustrated in **Figure 6**. The model comprises: an encoder layer (a fully connected layer) that projects the input features into a latent space to generate initial sequential representations; an LSTM layer that processes these sequential representations, where a dropout rate of 0.2 is applied to the forward pathways of the neurons' output gates to regularize the model and mitigate overfitting; and a decoder

layer (another fully connected layer) that maps the LSTM outputs to the final feature sequence.

During training, we use the AdamW optimizer along with a non-restarting cosine annealing learning rate scheduler (CosineAnnealingLR). The scheduler has a period of 200 epochs and a minimum learning rate of $1\times10^{-6}$. This strategy ensures smooth convergence by preventing oscillations from excessively large learning rate changes and mitigating underfitting caused by an overly small learning rate.

Other key hyperparameters are configured as follows: the loss function is the mean squared error (MSE); the batch size is 20; the maximum number of epochs is 200; and the initial learning rate is $5\times10^{-4}$. Additionally, an early stopping mechanism with a patience of 20 epochs is applied, which halts training if no improvement in validation loss is observed for 20 consecutive epochs.

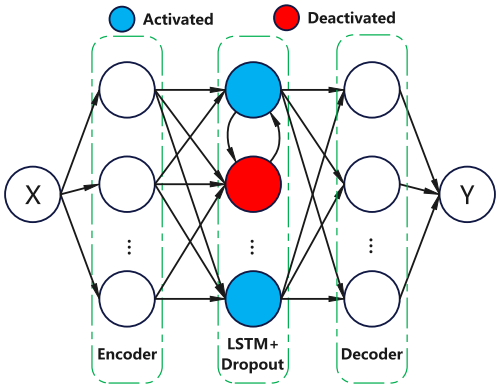



**Figure 6.** Schematic diagram of the model structure.

## 3.3. Dataset construction

A dependable and sufficiently large dataset is essential for reliable deeplearning model training. Our dataset is constructed through highprecision dynamic simulations that vary key initial conditions (load height, load weight, initial hoisting velocity, and hoisting acceleration) to reflect diverse crane operating states. Each simulation, configured with a 3s duration and a 0.02s fixed time step, outputs three characteristic responses: cable force, load position, and load velocity.

Postprocessing addresses the nonuniform time steps occasionally introduced by the Adams solver: spline interpolation is applied to all exported data to resample them onto a consistent time sequence. Python is then used for batch processing, aligning the feature sequences of every sample to identical time points. The processed data for each sample, comprising input features (Time, Load, Height, Vel0, Acc0) and output features (Cable_Force, Freight_Position, Freight_Velocity), are assembled into a CSV file following the prescribed column order, thereby yielding one complete dataset sample.

To obtain a deep learning model with stronger generalization capability and higher prediction accuracy, it is essential to construct a dataset with stronger representativeness and broader coverage. This essentially requires acquiring a sufficient number of sampling points within the parameter space spanned by multiple dimensions of initial conditions, and ensuring that these sampling points cover the entire multi-dimensional space as uniformly as possible. As early as 1979, M.D. McKay and R.J. Beckman proposed a random sampling method called Latin Hypercube Sampling (LHS). Its principle is to use the probability interval (0,1) to solve the inverse functions of different distributions, thereby obtaining sample points that cover the entire sampling region. This method is characterized by high sampling efficiency and good robustness [5]. Therefore, this study adopts the LHS method for sampling.

Crane operations are composed of four basic modes: accelerating hoisting, hoisting braking, accelerating lowering, and lowering braking. Even complex actions like inching lowering can be represented as rapid switches between accelerating lowering and lowering braking, while other maneuvers are likewise decomposable into these modes. **Table 3** lists the value ranges of the four variable initial conditions used in the Adams simulations for each mode. The acceleration levels (low, medium, high) correspond to the gear parameters introduced earlier, and the driving function for a simulation is defined by the chosen initial velocity and acceleration level.

In the accelerating hoisting mode, the relationship between the initial velocity and the gear selection is as follows:

(1) when the initial velocity $V_0 \in (0,5.4)$m/min, the acceleration gear can be low, medium, or high;

(2) when $V_0 \in (5.4,9.6)$m/min, the acceleration gear can be medium or high;

(3) when $V_0 \in (9.6,16)$m/min, the acceleration gear can only be high.

**Table 3.** The range of input feature values in different modes

| Mode | Height | Load | Initial velocity | Acceleration gear |
|---|---|---|---|---|
| **Accelerated Hoisting** | 0.5~6m | 0~10T | 0~16m/min | (low, medium, high) |
| **Hoist Braking** | 0.5~6m | 0~10T | 0~16m/min | high |
| **Accelerated Lowering** | 0.5~6m | 0~10T | 0~5.4m/min | low |
| **Lowering Braking** | 0.5~6m | 0~10T | 0~5.4m/min | high |

The correlation between initial velocity and allowable acceleration gear in the acceleratinghoisting mode necessitates a revised sampling strategy that aligns with practical operating norms. Safety regulations often require lowspeed operation for heavy loads or near personnel, thereby increasing the likelihood of lowgear usage. To encode this prior knowledge, the velocity domain is partitioned into three intervals with an assumed usageprobability ratio of **5:3:2**; the corresponding samplenumber ratio is thus **50:30:20**. For the two dimensions of height and load, the sampling number is set to N = 100. Respectively, the sample numbers corresponding to the low, medium, and high gears are preset as $n_1$, $n_2$, and $n_3$. A sampling distribution that satisfies the above requirements is shown in **Table 4**.

**Table 4.** Sample distribution for accelerated hoisting mode

| Number of samples | Low speed | Medium speed | High speed | Total |
|---|---|---|---|---|
| Low gear | 30 | 10 | 10 | 50 |
| Medium gear | \ | 20 | 10 | 30 |
| High gear | \ | \ | 20 | 20 |
| Total | 30 | 30 | 40 | 100 |

A Latinhypercube sampler (implemented via SciPy's qmc submodule) is applied independently to each of the four operatingmode distributions. With different random seeds, 100 points are drawn per distribution, producing **400 distinct initialcondition sets** (**Figure 7**). Each set is used to configure the Adams dynamic model, and simulations are executed followed by consistent postprocessing, yielding **400 samples** of the characteristic response data. The full dataset is then split into training, validation, and test sets in an **8:1:1** ratio, resulting in 320 training samples, 40 validation samples, and 40 test samples.

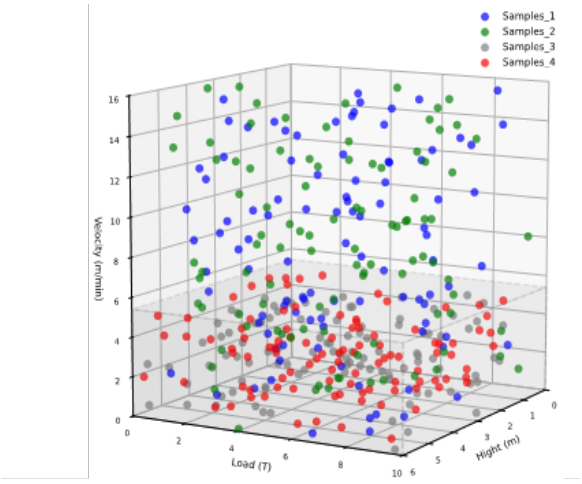



**Figure 7.** Distribution of sampling points.

## 3.4. Model training and evaluation

Since the input and output feature data of the samples have different dimensions and units, their direct use may easily lead to exploding or vanishing gradient problems. Therefore, before model training, the dataset is first shifted to zero the starting points of all sample output features, thereby eliminating prediction errors caused by the initial values. Subsequently, normalization is applied to mitigate gradient update imbalances arising from different data scales. In this model, the Min-Max Normalization method is adopted to scale all data to the [0, 1] interval. The mean squared error (MSE) is used as the evaluation metric during model training, and its expression is as **Equation (5)**:

$$S_{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{5}$$

where: $y_i$ represents the true value, $\hat{y}_i$ represents the predicted value, and n represents the total number of samples.

After 200 epochs of iteration, the model trained at the 185th epoch is selected as the best model, achieving the lowest validation loss of 0.0371 and a training loss of 0.0429. The changes in training loss, validation loss, and learning rate during the training process before reaching the best model are shown in **Figure 8**.

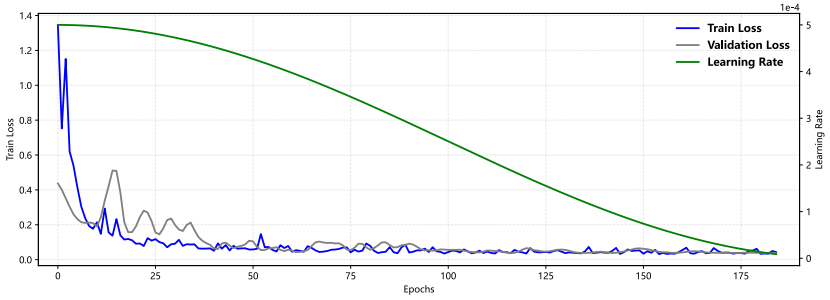



**Figure 8.** Loss and learning rate change curves.

The best model is used to perform forward propagation (without gradient computation) on the test set samples, which are free of data leakage, to obtain predicted values. These predicted values are then compared with the ground truth values in the test set to evaluate the prediction accuracy of the model. Since the three

output features have different dimensions and units, two evaluation metrics based on range normalization are adopted to comprehensively assess the prediction performance: the Normalized Root Mean Square Error (NRMSE) and the Normalized Mean Absolute Error (NMAE). Their calculation formulas are as **Equation (6)**:

$$\begin{cases} S_{NRMSE} = \dfrac{\sqrt{\dfrac{1}{n}\sum_{i=0}^{n}\left(y_i - \hat{y}_i\right)^2}}{|\, y_{max} - y_{min}\,|} \\ S_{NMAE} = \dfrac{\dfrac{1}{n}\sum_{i=0}^{n}\left|y_i - \hat{y}_i\right|}{|\, y_{max} - y_{min}\,|} \end{cases} \tag{6}$$

where: n represents number of time steps， $y_{max}$ represents maximum value of the feature, $y_{min}$ represents minimum value of the feature, $y_i$ represents predicted value at the i-th time step, $\hat{y}_i$ represents actual value at the i-th step.

Comparison results of selected test samples are shown in **Figure 9**. Based on the comparison between predicted and ground truth values for the three features across all 40 samples in the test set, the prediction accuracy of the model is evaluated as follows: the average peak relative error is 4.72%. This metric is particularly meaningful for the force and velocity features, indicating that the model possesses a good capability for peak value prediction. The average NMAE value is 0.0431, implying that the mean absolute error accounts for only about 4.31% of the data fluctuation range, demonstrating that the model has successfully learned the overall variation patterns of the dynamic responses for each feature. The average NRMSE value is 0.0681. Since the root mean square error is sensitive to outliers that deviate significantly from the true values, this metric reflects the good stability of the model's prediction capability.

Finally, a script is used to perform 50 rounds of forward propagation tests, and the average time from model loading to the completion of forward propagation is recorded and compared with the duration of dynamic simulation. The dynamic model employed in this paper takes approximately 30 minutes to complete one multi-body dynamics simulation calculation, whereas the trained LSTM model takes less than 300 ms to complete one calculation. In this scenario, the computational efficiency is directly improved by a factor of approximately 6000.

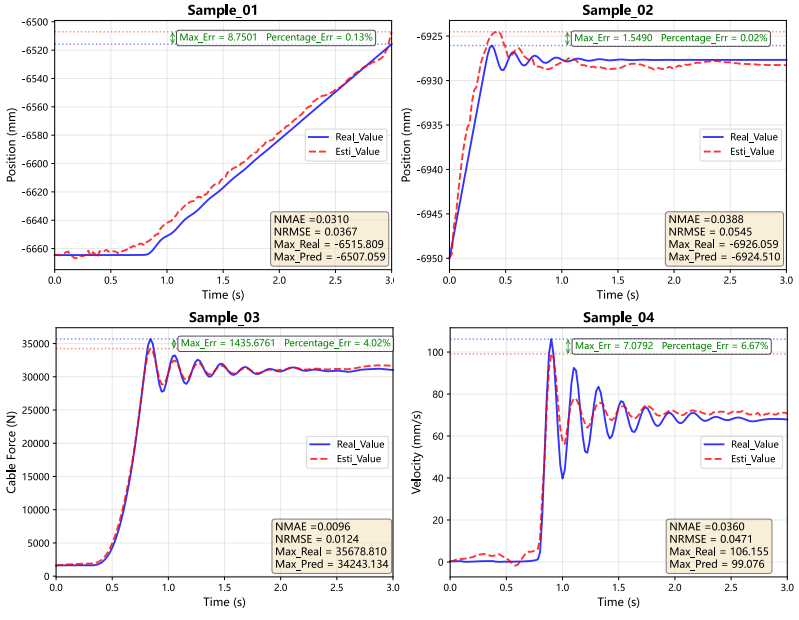



**Figure 9.** Comparison of predicted data for selected test samples.

## 4. Conclusion

In this study, to address the computational efficiency issue in predicting the transient dynamic response of the hoisting mechanism of a double-beam bridge crane, a deep learning prediction model based on an LSTM neural network was successfully constructed and validated. This model enables a transition from traditional numerical simulation based on multibody dynamics to efficient data-driven prediction. A comparison between the dynamic simulation results and the LSTM model predictions demonstrates that the proposed model possesses both high prediction accuracy and remarkable computational efficiency. However, several aspects of this study still require further improvement. For instance, the model was trained without using data collected from the real-world operating environment of the crane. Consequently, although the model predictions are close to the numerical simulation results, they may still deviate from the actual measured values. Furthermore, during actual operation, the hoisting mechanism of the crane not only performs lifting and lowering actions but also involves acceleration and deceleration motions in the forward/backward and lateral directions. These motions induce more complex dynamic response variations, which the current model is incapable of predicting. Therefore, in future research, using data collected from real-world environments, combined with dynamic model data validated through comparison with real-world measurements, and adopting more appropriate deep learning model architectures, it is expected to develop a prediction model with stronger generalization ability, higher prediction accuracy, and better computational efficiency.

## Funding

The authors sincerely thank the 2025 Dalian Jiaotong University College Students Innovation and Entrepreneurship Training Program (Grant No. X202510150007S) for funding this research.

## Disclosure statement

The author declares no conflict of interest.

## References

[1] Zhang Y, Zhao Z, Zhao C, et al., 2025, A Fast Calculation Method for Electromagnetic Field based on Deep Operator Network. High Voltage Engineering, 51(3): 1484–1494.

[2] Liu F, Chen H, 2025, Response Study of Tower Crane and Elastic Plate under Dynamic Complicated Excitation. PLoS One, 20(6): e0324745.

[3] Sun J, Ke D, Xu J, et al., 2026, A PINN-DDPG Fusion Evaluation Method for Dynamic Regulation Potential of Building Thermal Load, Proceedings of the CSEE, 1–12.

[4] Chen J, Hao X, Yu Q, 2012, Dynamic Analysis of Parallel Mechanism with Clearance Joints. Journal of Zhengzhou University (Engineering Science), 33(1): 98–101.

[5] Feng L, Wang J, Liu F, et al., 2018, Dynamic Simulation of Edge Loading Caused by Separation of Ceramic Hip Joint Ball Head and Acetabular Cup. Chinese Journal of Tissue Engineering Research, 22(7): 985–990.