

# Fatigue-Consistent Load Extrapolation Based on Tail-Weighted Histogram-Regularized LSTM

Yu Bai, Fei Meng\*

Science of Systems, University of Shanghai for Science and Technology, Shanghai, China

Email: \*feimeng@usst.edu.cn

**How to cite this paper:** Bai, Y. and Meng, F. (2026) Fatigue-Consistent Load Extrapolation Based on Tail-Weighted Histogram-Regularized LSTM. *World Journal of Engineering and Technology*, **14**, 172-186. <https://doi.org/10.4236/wjet.2026.141010>

**Received:** December 22, 2025

**Accepted:** January 25, 2026

**Published:** January 28, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

This study proposes a Tail-Weighted Histogram-Regularized LSTM (TWHR-LSTM) to extend 10s load signals into longer sequences while preserving fatigue characteristics. The method removes trends, normalizes the signal, and trains a sequence-to-sequence LSTM model using overlapping windows of 2000 data points. The model's loss function includes reconstruction error, histogram distance, variance regularization, and a penalty for large step changes to focus on extreme rainflow cycles. Compared to traditional parameter and KDE rainflow extrapolation methods, TWHR-LSTM better reproduces the tail of the load range spectrum, with a pseudo-damage ratio error of only 1.2%. It shows superior performance in maintaining fatigue characteristics and signal quality.

## Keywords

LSTM Sequence Modeling, Load Extrapolation, Rainflow Counting

## 1. Introduction

Fatigue load extrapolation is an important research topic in modern mechanical engineering. It plays a key role in predicting the service life of industrial equipment. In real-world applications, long-term load testing is often costly and time-consuming. Therefore, researchers aim to estimate full-life load histories using short-term data through various models and algorithms [1]-[3].

Traditional statistical methods have been widely used for load spectrum extrapolation. Common techniques include kernel density estimation (KDE), Weibull distribution, and extreme value theory (EVT). Yang *et al.* [4] improved KDE by using the DBSCAN algorithm to optimize bandwidth, which helped capture the tail distribution of load more accurately. Shen *et al.* [5] developed

a dual-spectrum extrapolation method that considered the sequence effects of loads in drilling systems. Yong Zhang *et al.* [6] analyzed metro vehicle loads and proposed a programmatic method to generate load spectra. Mei *et al.* [7] combined multibody dynamics with statistical modeling to improve pantograph load prediction. These traditional models are easy to apply and interpret [8]. However, they often fail to capture rare extreme loads that are critical for fatigue damage.

In recent years, deep learning and data-driven models have become powerful tools for load extrapolation and fatigue life prediction. Heng *et al.* [9] proposed a CNN-LSTM hybrid model to predict multiaxial fatigue life across different materials. Chen *et al.* [10] developed a frequency-domain neural network that improved the prediction of fatigue over multiple scales. Santos *et al.* [11] introduced a physics-guided neural network to estimate offshore wind turbine loads more accurately. Farid [12] combined artificial neural networks with Gaussian process regression to enable real-time fatigue prediction under random loading. Jia *et al.* [13] used an LSTM model to predict the fatigue life of reinforced concrete beams, showing strong learning ability even with complex materials. These studies show that deep learning models can better capture nonlinear relationships in fatigue data than traditional statistical methods. Other researchers have focused on data reconstruction and hybrid modeling approaches. Lei *et al.* [14] used generative adversarial networks (GANs) to reconstruct missing load data in structural health monitoring. Gibson *et al.* [15] applied Gaussian process regression to predict strain and fatigue damage distributions. Yang *et al.* [16] and Zheng *et al.* [17] used extreme value theory and the peaks-over-threshold method to capture rare load cycles more accurately. Dewa and Kepka [18] improved extrapolation accuracy for bus structural steel, and Yang *et al.* [19] proposed a hybrid time-domain extrapolation method for vehicle loads. These studies suggest that combining engineering knowledge with data-driven modeling can significantly improve load extrapolation accuracy.

Recent studies have also worked on improving the design of loss functions and applying statistical constraints in neural networks. Ustinova and Lempitsky [20] proposed the histogram loss to align the model's output distribution with the target data. Niu *et al.* [21] developed a tail-aware prediction framework using LSTM and the generalized extreme value (GEV) distribution to better predict rare events. Imani *et al.* [22] tested histogram loss in regression tasks and found that it improved both model stability and convergence. These methods help neural networks maintain consistency with the statistical characteristics of real fatigue data while avoiding overfitting. Some recent studies have also introduced advanced hybrid frameworks and application-specific deep neural networks. Shi *et al.* [23] built a load spectrum model for loader equipment to support fatigue life evaluation. Gan *et al.* [24] created a modular neural network, pretrained with uniaxial fatigue data, to predict multiaxial fatigue life. Wang *et al.* [25] proposed a multi-physics neural network for additively man-

ufactured superalloy parts, showing strong generalization ability across different materials. These works demonstrate that combining engineering knowledge, data-driven modeling, and deep learning enables high-accuracy and robust fatigue life predictions.

Despite the noticeable achievements in load spectrum compilation and extrapolation, current rainflow-based and statistical methods still face two key limitations: (1) Most parametric and KDE-based models focus on fitting the overall range-mean distribution, but they lack explicit mechanisms to control or reproduce extreme load cycles. (2) Many extrapolation strategies aim to match amplitude-level statistics; however, the generated time histories are typically assembled from sampled cycles without enforcing local smoothness or temporal consistency, making them less realistic in the time domain.

To overcome these challenges, this study proposes a Tail-Weighted Histogram-Regularized Long Short-Term Memory network (TWHR-LSTM).

(1) The method introduces a tail-weighted sampling strategy combined with penalties on large step changes between adjacent points, allowing the model to pay more attention to high-amplitude segments and preserve them more accurately in the extrapolated results.

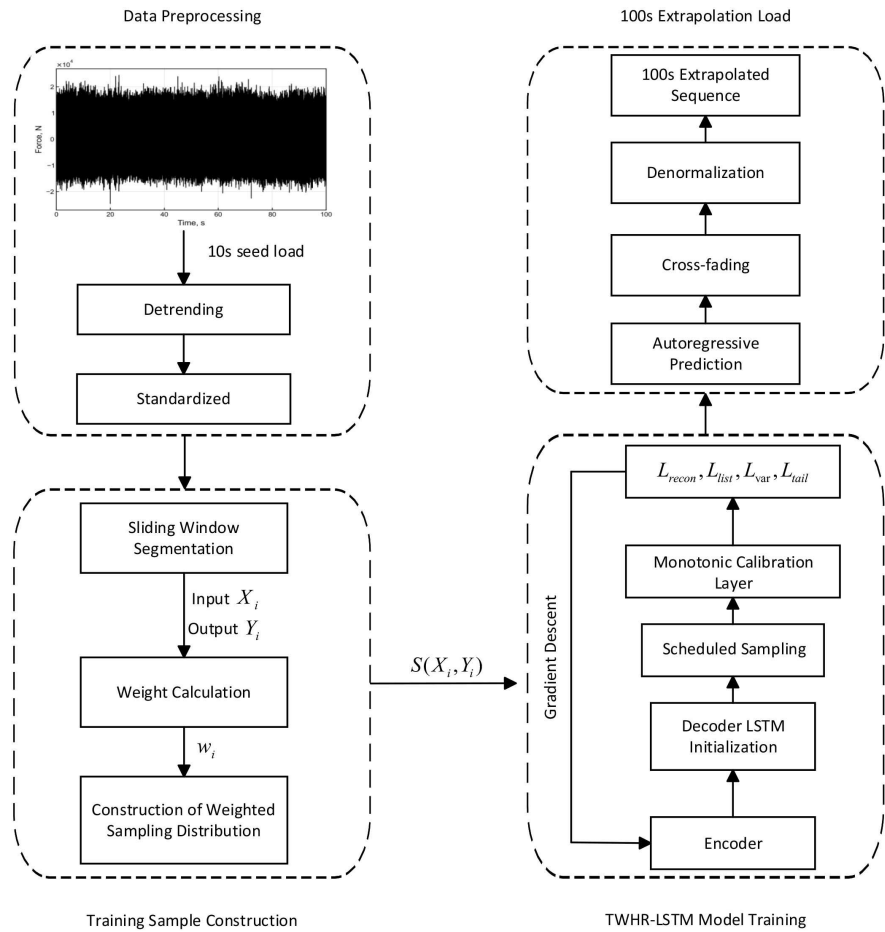
(2) A sequence-to-sequence LSTM framework is trained with pointwise reconstruction and variance regularization to capture temporal patterns, while a differentiable amplitude-histogram loss ensures that the generated distribution aligns with the original.

Together, these design elements improve both the fidelity of extreme events and the time-domain realism of the extrapolated load sequence.

## 2. The Method

In this work, a tail-weighted histogram-regularized long short-term memory network (TWHR-LSTM) is proposed to extrapolate a 10-second seed load into a 100 s load sequence. The raw force signal is first detrended to remove linear drift and then normalized. The resulting sequence is split chronologically into training and validation sets. Input-output pairs are created using overlapping sliding windows along the 10-second input, and a smooth tail-weighted sampling strategy is applied. This increases the sampling frequency for segments with large peak-to-peak ranges, while still retaining cycles of moderate amplitude.

The model adopts a sequence-to-sequence LSTM with an encoder-decoder architecture and a monotonic calibration layer. It is trained using a composite loss function that includes pointwise reconstruction error, a differentiable amplitude-histogram distance, variance regularization, and a tail penalty based on large local increments to reflect severe rainflow cycles. After training, the model operates in autoregressive mode, and short overlapping output blocks are combined using cross-fading to generate a continuous 100 s load sequence (see **Figure 1**).



**Figure 1.** Training TWHR-LSTM model and load extrapolation.

### 2.1. Data Preprocessing

The raw load signal is denoted as  $x(t)$ . A linear trend is removed using least-squares fitting:

$$x_{dt}(t) = x(t) - (\alpha t + \beta), \tag{1}$$

where  $\alpha$  and  $\beta$  are the fitted coefficients. This detrending step eliminates slow drift and constant offsets in the original signal. The detrended signal is then standardized:

$$z(t) = \frac{x_{dt}(t) - \mu}{\sigma}, \tag{2}$$

where  $\mu$  and  $\sigma$  are the sample mean and standard deviation, respectively. The normalized sequence  $z(t)$  is used for all subsequent processing. The 10s sequence is split chronologically: the earlier portion is used for training, while the later portion is reserved for validation.

This time-based split minimizes the risk of information leakage across subsets.

Crucially, a “train-from-scratch” protocol is adopted for each specific seed signal to capture its unique non-stationary dynamics. To prevent overfitting on the limited 10-second duration, the sliding window strategy acts as a data aug-

mentation technique. This forces the model to learn the underlying temporal transition rules shared across samples rather than memorizing the single long sequence.

## 2.2. Sliding Window Construction and Tail-Weighted Sampling

The normalized sequence  $z(t)$  is segmented into overlapping input-output pairs. For a given starting index  $i$ , the input window is defined as

$$\mathbf{X}_i = [z_i, z_{i+1}, \dots, z_{i+T_{in}-1}], \quad (3)$$

and the corresponding future window is

$$\mathbf{Y}_i = [z_{i+T_{in}}, \dots, z_{i+T_{in}+T_{out}-1}]. \quad (4)$$

The index  $i$  is advanced with a fixed stride. This produces a sequence of pairs  $\{(\mathbf{X}_i, \mathbf{Y}_i)\}$  from the 10s record. To emphasize regions with large fluctuations, a simple peak-to-peak range is computed for each combined window

$$\mathbf{W}_i = [\mathbf{X}_i, \mathbf{Y}_i] \quad (5)$$

and its range is calculated as

$$r_i = \max(\mathbf{W}_i) - \min(\mathbf{W}_i). \quad (6)$$

To increase the representation of high-variation segments during training, each window is assigned a weight

$$w_i = 1 + \gamma f(r_i), \quad (7)$$

where  $\gamma$  controls the maximum emphasis. The function  $f(\cdot)$  is a smooth curve that increases from 0 at the median to 1 near the upper quantiles of the range distribution. These weights are used in a weighted random sampler, allowing high-range windows to be sampled more frequently while still retaining moderate-range windows in the training set.

## 2.3. Sequence-to-Sequence LSTM with Distribution-Aware Loss

Each normalized load segment is expressed as an input-target pair  $(X_i, Y_i)$ , where  $X_i \in \mathbb{R}^{T_{in} \times 1}$  and  $Y_i \in \mathbb{R}^{T_{out} \times 1}$ . An encoder LSTM encodes  $X_i$  into a latent state, and a decoder LSTM generates the future sequence. During training, scheduled sampling is applied, where the decoder input is a mix of ground truth and past predictions, enabling the model to learn autoregressive continuation.

The specific architecture and training hyperparameters are detailed in **Table 1**. The encoder and decoder are both 2-layer LSTMs with 128 hidden units to balance model capacity with the limited data size. A dropout rate of 0.2 is applied to further mitigate overfitting. The model is trained using the Adam optimizer.

The decoder output is passed through a monotone calibration layer that applies a smooth, increasing mapping along the amplitude axis. This adjusts the marginal distribution while preserving the order of extreme values.

The loss function combines pointwise accuracy, distribution alignment, and tail modeling. The reconstruction term is the standard mean squared error:

$$L_{\text{recon}} = \|\hat{Y}_i - Y_i\|_2^2. \quad (8)$$

To align amplitude distributions, soft histograms  $p$  and  $q$  are computed from  $\hat{Y}_i$  and  $Y_i$ , and their cumulative distributions are compared via:

$$L_{\text{hist}} = \|\text{CDF}(p) - \text{CDF}(q)\|_1. \quad (9)$$

A variance penalty is added to match the global energy level of the predicted and target sequences. Additionally, large local increments approximate rainflow tail cycles. First-order differences are computed, and values above a high percentile are retained. The loss encourages slightly higher intensity in the prediction:

$$L_{\text{tail}} = (s_{\text{pred}} - 1.1s_{\text{tgt}})^2, \quad (10)$$

which introduces a mild bias toward more severe rare cycles. The total loss is a weighted sum:

$$L = w_{\text{recon}}L_{\text{recon}} + w_{\text{hist}}L_{\text{hist}} + \lambda_{\text{var}}L_{\text{var}} + \lambda_{\text{tail}}L_{\text{tail}}, \quad (11)$$

with weights scheduled to emphasize reconstruction in early training and distributional and tail accuracy in later stages (see **Table 1**).

## 2.4. Long-Horizon Load Extrapolation

After training, the model runs in an autoregressive way. It first takes the initial  $T_{\text{in}}$  samples from the normalized data as input, called  $\mathbf{X}^{(0)}$ . The model then gives the first predicted block:

$$\hat{\mathbf{Y}}^{(0)} = F(\mathbf{X}^{(0)}) \quad (12)$$

This predicted block becomes the next input:

$$\mathbf{X}^{(k+1)} = \hat{\mathbf{Y}}^{(k)}, k = 0, 1, \dots \quad (13)$$

The model repeats this step until it produces the full 100 s sequence. To avoid sudden changes between blocks, each output segment overlaps slightly with the next one. These overlaps help make the full sequence smooth. The final normalized sequence  $\hat{z}(t)$  is then converted back to physical units using the stored mean and standard deviation:

$$\hat{x}(t) = \hat{z}(t)\sigma_x + \mu_x \quad (14)$$

## 2.5. Pseudo-Damage Definition

This output can be used for rainflow counting and fatigue damage estimation. Because material parameters are unavailable, a relative measure is adopted [26]-[28]. Specifically, the pseudo-damage of each extrapolated 100 s load is compared with that of the ground-truth 100 s signal. A ratio close to 1 indicates high accuracy.

Pseudo-damage is computed from rainflow cycles  $(F_a, F_m, n)$ , where  $F_a$  is the amplitude,  $F_m$  is the mean load, and  $n$  is the cycle count. Goodman mean correction is applied to account for mean stress effects. The equivalent amplitude is

$$F_a^{eq} = \frac{F_a}{1 - \frac{F_m}{F_u}}$$

where  $F_u$  is the ultimate load limit. Cycle pseudo-damage is then accumulated as

$$D = \sum_i n_i (F_{a,i}^{eq})^m$$

where  $m$  is the S-N slope. To remove the influence of segment length, a per-time pseudo-damage rate is defined:

$$d = \frac{D}{T}, k_T = \frac{d_{ext}}{d_{true}}$$

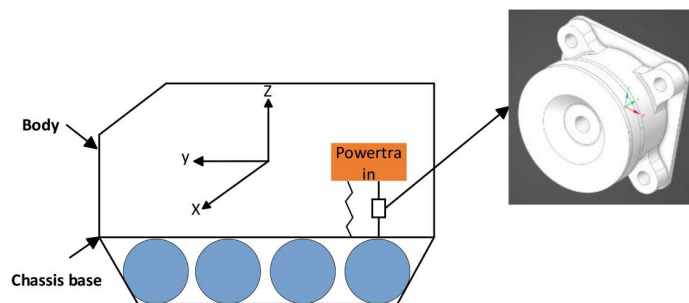
Finally, the accuracy metric is the ratio between extrapolated and true 100 s data:  $k_T \approx 1$  indicates that the extrapolated load preserves the fatigue severity of the true signal [29].

**Table 1.** Parameter setting.

Category	Parameter	Value
Data Specification	Sampling Rate	2000 Hz
Sliding Window	Input, Output	2000 points
	Stride	200 points
Training	Layers, Hidden Units	2, 128
	Dropout Rate, Epochs	0.2, 200
Loss Weights	$\lambda_{var}, \lambda_{tail}$	0.1, 0.1
		0.7, 0.3 (0 - 50)
	$W_{recon}, W_{hist}$	0.5, 0.5 (51 - 150)
		0.3, 0.7 (151 - 200)

### 3. Experimental Data Source and Processing

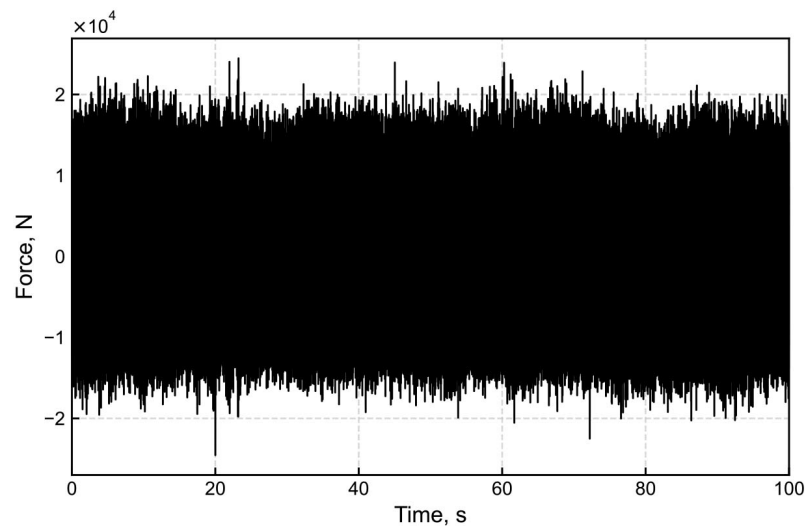
Experimental data used in this study were collected from field tests on the powertrain mounting system of a heavy-duty vehicle. The powertrain was elastically connected to the vehicle body at four mounting points (see **Figure 2**). Triaxial accelerometers were installed at each mount to record vibrations in three directions at a sampling rate of 2000 Hz. All signals were synchronously recorded using a dynamic acquisition system.



**Figure 2.** Heavy-duty vehicle powertrain mounting system.

Engineering experience indicated that the left-rear mount is more prone to fatigue damage during operation. Therefore, this study focuses on the dynamic force signal in the z-direction at that location. The test was conducted under a steady vehicle speed of 30 km/h, and the total recording time was 100 seconds.

The raw signal included high-frequency noise, so a low-pass filter with a 500 Hz cutoff was applied to improve the signal-to-noise ratio while preserving relevant content. For load spectrum analysis, rainflow counting was used to remove cycles that contribute little to fatigue. Specifically, cycles with amplitudes below 10% of the maximum load were discarded. A full 100-second load-time history was then reconstructed using linear interpolation (see **Figure 3**).



**Figure 3.** 100 s force data at 30 km/h.

To evaluate different extrapolation methods, a validation framework was designed. Starting from the first 10 seconds of measured data, each method generated an extrapolated 100 s sequence. These results were compared with the original measured 100 s signal for performance assessment.

It is important to acknowledge that the proposed method is condition-specific. Since the spectral characteristics and amplitude distributions of powertrain vibrations vary significantly with vehicle speed, the model requires retraining when applied to a different operating condition (e.g., 60 km/h). However, given the lightweight architecture of the TWHR-LSTM, retraining on a new 10-second seed is computationally efficient, allowing for rapid adaptation to various steady-state scenarios.

#### 4. Extrapolation Methods

This section presents three extrapolation strategies that extend a 10 s load signal to 100 s. All share the same detrended 10 s seed and produce signals with identical sampling rate and target duration. The three approaches tested reflect different philosophies of modeling. Two classical rainflow methods model cycle statistics and

render synthetic turning points as a means of indirect time series reconstruction. The proposed TWHR-LSTM learns continuation directly in the time domain, with explicit attention to tail enhancement and distribution matching.

#### **4.1. Parametric Rainflow Extrapolation**

This approach starts by extracting rainflow cycles from the 10 s seed. It fits a Weibull-3P distribution to the cycle ranges and a Gaussian distribution to the cycle means. Their statistical relationship is maintained using a Gaussian Copula. Based on these models, new cycles are sampled and turned into alternating turning points. A synthetic load waveform is then built using linear interpolation, guided by observed half-cycle lengths. While the method is easy to interpret, its accuracy is limited by the assumptions of the fitted distributions, especially in the upper tail [30].

#### **4.2. KDE Rainflow Extrapolation**

This method models use a non-parametric 2-D kernel density estimator. Synthetic cycles are drawn from the estimated distribution, trimmed by percentiles, and smoothly amplified in the tail using a continuous gain function. The reconstruction of turning points follows the same steps as in the parametric approach. While KDE offers more flexibility in modeling complex distributions, it may lack constraints when handling extreme values.

#### **4.3. TWHR-LSTM Sequence Extrapolation**

The proposed TWHR-LSTM learns to extend the load signal directly in the time domain. It uses sliding windows to create input-target pairs, giving more weight to segments with large amplitudes. During training, the model minimizes a combination of losses: pointwise reconstruction error, a differentiable histogram loss to match amplitude distributions, and a tail-focused penalty to reinforce extreme transitions. After training, it produces the full 100-second signal through recursive prediction, with cross-fading used to blend adjacent segments smoothly.

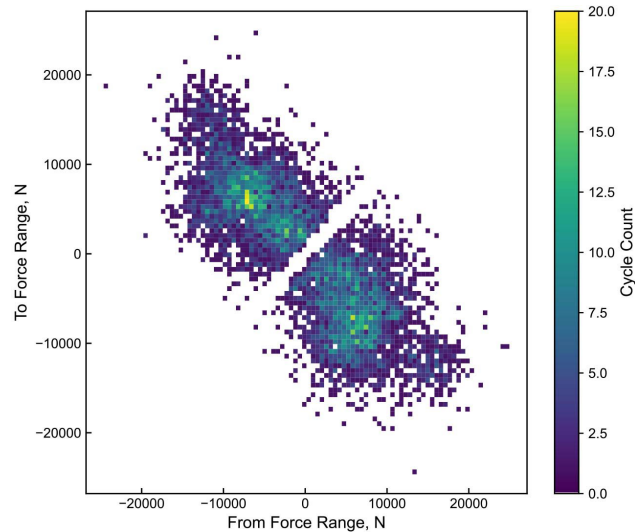
### **5. Experimental Results**

This section compares three extrapolation strategies. The comparison focuses on distribution-oriented metrics: the rainflow from-to matrix, the range exceedance curve, and the pseudo-damage ratio with respect to the measured 100 s record. These together reflect whether or not an extrapolated signal captures both small-cycle behavior and the tail-dominated cycles which dominate fatigue.

#### **5.1. Rainflow from-to Matrix Comparison**

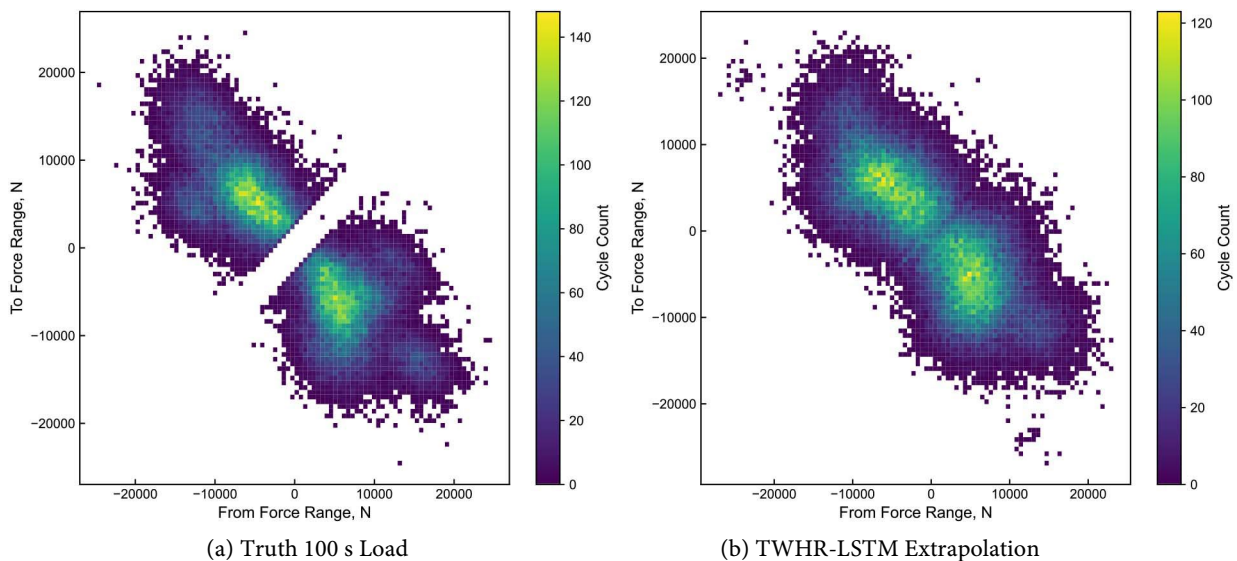
To evaluate the three extrapolation methods, all load signals are first processed by rainflow counting. The rainflow matrix describes the joint distribution of cycle “from-to” values and their occurrence counts. And it is widely used in fatigue analysis to check whether the cycle pattern of a reconstructed signal is consistent with that of the reference load.

**Figure 4** shows the rainflow matrix of the 10s seed load. Because the duration is short, the cycles are concentrated in several local clusters. Based on this 10s seed, three 100 s load histories are generated: parametric rainflow extrapolation, KDE-based non-parametric extrapolation, and the proposed TWHR-LSTM method.



**Figure 4.** Rainflow from-to matrix of the 10s seed load.

**Figure 5** shows the rainflow matrices of the three extrapolated loads, compared to the truth 100 s load. The parametric rainflow method captures the main cycle region but appears overly smooth and underestimates extreme cycles due to its Weibull-Gaussian assumptions. The KDE-based method maintains the overall shape but shows fragmented and noisy density in the large-amplitude areas. In contrast, the proposed TWHR-LSTM accurately reproduces the two main clusters, the diagonal transition band, and the tail cycles with much greater detail and precision.



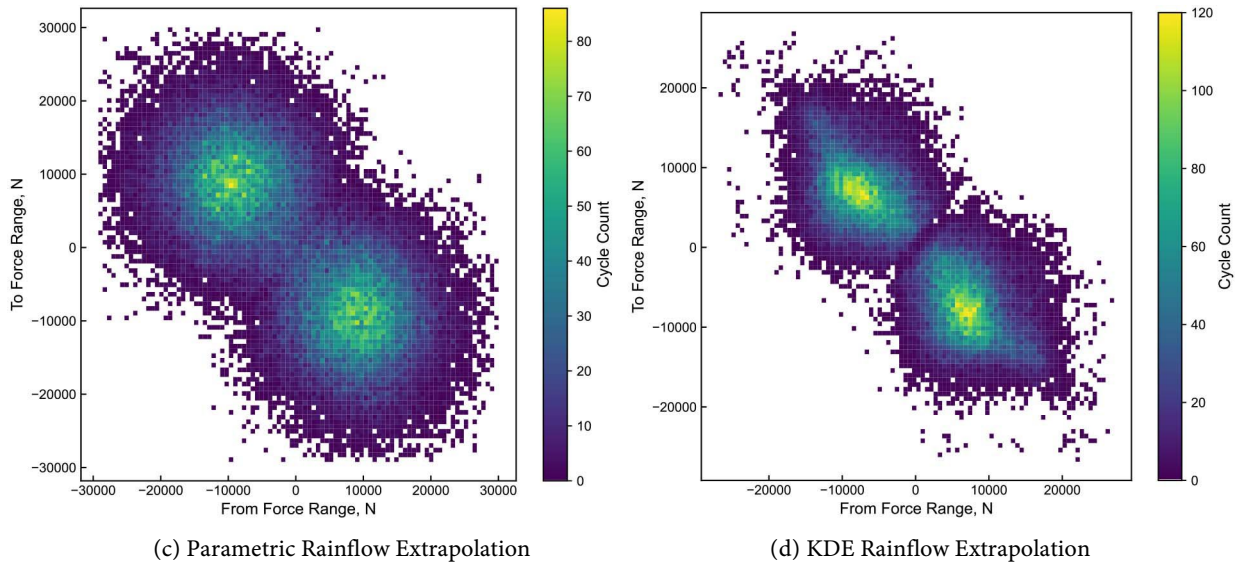


Figure 5. Extrapolation vs truth—100 s load.

The large-range and low-probability cycles of this method are the closest to the ground-truth matrix. This is attributed to tail-weighted sampling and histogram-based loss, guiding the network to match the true cycle distribution rather than only minimizing pointwise error.

### 5.2. Comparison of Range-Frequency Characteristics

To further investigate how each method performs in the high-range region, a range exceedance curve is plotted for each case, as shown in Figure 6. This curve sorts all rainflow ranges from largest to smallest and plots how often each range occurs.

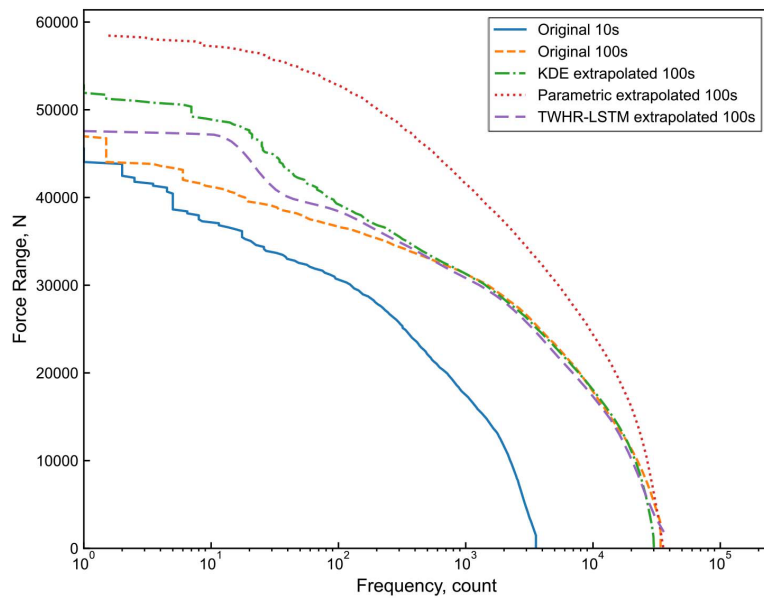


Figure 6. Range exceedance curve.

This kind of plot is especially important for fatigue analysis, because large-range cycles contribute most to total damage. As a result, accurately capturing the upper tail of the distribution is critical.

The parametric method clearly overestimates the extreme ranges. This happens because the fitted Weibull-3P distribution has a heavy tail, which leads to an unrealistically long upper end. The KDE-based method performs better around the middle of the range but still shows a clear mismatch in the top 5 - 10%, where smoothing blurs sharp peaks and weakens the tail.

In contrast, the proposed TWHR-LSTM closely matches the truth 100 s distribution across the entire range, including the tail.

### 5.3. Pseudo-Damage-Based Accuracy Evaluation

All three extrapolation methods—parametric rainfall extrapolation, KDE-based extrapolation, and the proposed TWHR-LSTM—are evaluated under same procedure. The resulting pseudo-damage ratios for the three methods are summarized in **Table 2**, providing a unified and fatigue-oriented comparison of extrapolation accuracy.

**Table 2.** Pseudo-damage ratios and range count ( $m = 8$ ).

Methods	Count	Value $k_r$
Original	33972	1
Parametric	35612	1.120
KDE	33093	1.281
TWHR-LSTM	35466	1.012

The pseudo-damage ratios in **Table 2** show that TWHR-LSTM gives the smallest error. The  $k_r = 1.012$  means only about 1.2% overestimation, while the parametric method overestimates by 12% and KDE by 28.1%. Meanwhile, the TWHR-LSTM cycle count 35466 remains close to the original 33972, indicating better fatigue-severity preservation.

## 6. Conclusions

This paper investigated data-driven extrapolation of road load spectra from a short 10 s seed load to an equivalent 100 s sequence. Two conventional rainfall-based approaches were used as baselines: one parametric method using Weibull-distributed ranges, Gaussian-distributed means, and a Gaussian copula; and one non-parametric KDE-based method. Both methods can reproduce the overall range histogram, but they are sensitive to tail fitting and copula assumptions.

A tail-weighted histogram-regularized LSTM (TWHR-LSTM) was then proposed. The network is trained in a sequence-to-sequence setting with sliding windows, using tail-aware sampling, a differentiable histogram loss to enforce amplitude consistency, and a penalty on large step changes to preserve rare, high-amplitude cycles.

Rainflow from-to matrices and range exceedance curves show that TWHR-LSTM better preserves the shape of the tail region. Pseudo-damage per unit time confirms this trend: the proposed method yields a pseudo-damage ratio closest to the real 100 s reference. Overall, the results indicate that TWHR-LSTM can provide a more balanced load extrapolation, capturing extreme events without sacrificing global statistical consistency.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Cheng, X., Han, T., Yang, P. and Zhang, X. (2021) Compilation of Load Spectrum for 5MN Metal Extruder Based on Long Short-Term Memory Network. *Applied Sciences*, **11**, Article No. 9708. <https://doi.org/10.3390/app11209708>
- [2] Duran, J.A.R., Bermudez, P.R. and Francisco, A.S. (2023) Extrapolation of Real Service Loads to Longer Periods. *Acta Mechanica*, **235**, 1751-1759. <https://doi.org/10.1007/s00707-023-03810-0>
- [3] Shi, C., Sun, Z. and Feng, G. (2023) Research on the Method of Compiling the Normalized Fatigue Load Spectrum of Motorized Transportation. In: Yadav, S., *et al.*, Eds., *Recent Advances in Applied Mechanics and Mechanical Engineering*, Springer, 333-340. [https://doi.org/10.1007/978-981-99-2375-5\\_33](https://doi.org/10.1007/978-981-99-2375-5_33)
- [4] Yang, X., Zhou, X., Wan, B. and Fu, Y. (2023) Load Spectra Extrapolation by Bandwidth-Optimized Kernel Density Estimation Based on DBSCAN Algorithm. *Journal of Vibration Engineering & Technologies*, **12**, 1445-1456. <https://doi.org/10.1007/s42417-023-00919-3>
- [5] Shen, Y., Li, Y., Wang, J. and Wang, H. (2024) A Double Load Spectrum Extrapolation for Drill Pipe of Rotary Drilling Rig Considering Load Sequence. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, **238**, 5902-5920. <https://doi.org/10.1177/09544062231221623>
- [6] Su, Y.Z., *et al.* (2025) Research on Program Load Spectrum for Key Components of a Metro Vehicle Body. *Frontiers in Applied Mathematics and Statistics*, **11**, Article ID: 1556150. <https://doi.org/10.3389/fams.2025.1556150>
- [7] Mei, G., Luo, Q., Qiao, W., Huang, Z., Lu, J. and Wang, J. (2022) Study of Load Spectrum Compilation Method for the Pantograph Upper Frame Based on Multi-Body Dynamics. *Engineering Failure Analysis*, **135**, Article ID: 106099. <https://doi.org/10.1016/j.engfailanal.2022.106099>
- [8] Chavoshi, S.Z., Booker, J., Bradford, R. and Martin, M. (2021) A Review of Probabilistic Structural Integrity Assessment in the Nuclear Sector and Possible Future Directions. *Fatigue & Fracture of Engineering Materials & Structures*, **44**, 3227-3257. <https://doi.org/10.1111/ffe.13572>
- [9] Heng, F., Gao, J., Xu, R., Yang, H., Cheng, Q. and Liu, Y. (2023) Multiaxial Fatigue Life Prediction for Various Metallic Materials Based on the Hybrid CNN-LSTM Neural Network. *Fatigue & Fracture of Engineering Materials & Structures*, **46**, 1979-1996. <https://doi.org/10.1111/ffe.13977>
- [10] Chen, S., Zhou, X. and Bai, Y. (2025) A Frequency Domain Enhanced Multi-View Neural Network Approach to Multiaxial Fatigue Life Prediction for Various Metal Materials. *International Journal of Fatigue*, **190**, Article ID: 108620.

- <https://doi.org/10.1016/j.ijfatigue.2024.108620>
- [11] Santos, F.N., D'Antuono, P., Robbelein, K., Noppe, N., Weijtjens, W. and Devriendt, C. (2023) Long-Term Fatigue Estimation on Offshore Wind Turbines Interface Loads through Loss Function Physics-Guided Learning of Neural Networks. *Renewable Energy*, **205**, 461-474. <https://doi.org/10.1016/j.renene.2023.01.093>
- [12] Farid, M. (2022) Data-Driven Method for Real-Time Prediction and Uncertainty Quantification of Fatigue Failure under Stochastic Loading Using Artificial Neural Networks and Gaussian Process Regression. *International Journal of Fatigue*, **155**, Article ID: 106415. <https://doi.org/10.1016/j.ijfatigue.2021.106415>
- [13] Jia, M., Zhou, C., Pei, X., Xu, Z., Xu, W. and Wan, Z. (2025) Fatigue Life Prediction of CFRP-FBG Sensor-Reinforced RC Beams Enabled by LSTM-Based Deep Learning. *Polymers*, **17**, Article No. 2112. <https://doi.org/10.3390/polym17152112>
- [14] Lei, X., Sun, L. and Xia, Y. (2020) Lost Data Reconstruction for Structural Health Monitoring Using Deep Convolutional Generative Adversarial Networks. *Structural Health Monitoring*, **20**, 2069-2087. <https://doi.org/10.1177/1475921720959226>
- [15] Gibson, S.J., Rogers, T.J. and Cross, E.J. (2023) Distributions of Fatigue Damage from Data-Driven Strain Prediction Using Gaussian Process Regression. *Structural Health Monitoring*, **22**, 3065-3076. <https://doi.org/10.1177/14759217221140080>
- [16] Yang, Y., Zhang, J., Zhang, Z., Hu, X., Zhang, Y. and Lin, W. (2025) Time-Domain Extrapolation of Vehicle Durability Load Spectrum Based on Load Decomposition and POT Model Threshold Preference. *Mechanical Systems and Signal Processing*, **224**, Article ID: 112043. <https://doi.org/10.1016/j.ymsp.2024.112043>
- [17] Zheng, G., Liao, Y., Chen, B., Zhao, S. and Wei, H. (2024) Multi-Axial Load Spectrum Extrapolation Method for Fatigue Durability of Special Vehicles Based on Extreme Value Theory. *International Journal of Fatigue*, **178**, Article ID: 108014. <https://doi.org/10.1016/j.ijfatigue.2023.108014>
- [18] Dewa, R.T. and Kepka, M. (2021) Improved Extrapolation Method for the Fatigue Damage of Bus Structural Steel under Service Loading. *Journal of Mechanical Science and Technology*, **35**, 4437-4442. <https://doi.org/10.1007/s12206-021-0914-4>
- [19] Yang, Y., Peng, L., Lin, W., Zhang, J. and Zhang, Z. (2025) Improved Time-Domain Hybrid Extrapolation Method for Vehicle Durability Load Spectrum Based on Load Component Decomposition. *Measurement*, **245**, Article ID: 116660. <https://doi.org/10.1016/j.measurement.2025.116660>
- [20] Ustinova, E. and Lempitsky, V. (2016) Learning Deep Embeddings with Histogram Loss. *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, 5-10 December 2016, 4177-4185.
- [21] Niu, H., Murray, S., Jaber, F., Heidari, B. and Duffield, N. (2025) Tail-Aware Forecasting of Precipitation Extremes Using STL-GEV and LSTM Neural Networks. *Hydrology*, **12**, Article No. 284. <https://doi.org/10.3390/hydrology12110284>
- [22] Imani, E., Luedemann, K., Scholnick-Hughes, S., *et al.* (2024) Investigating the Histogram Loss in Regression.
- [23] Shi, X., Guo, W., Wang, J., Li, G. and Lu, H. (2025) Compilation of Load Spectrum of Loader Working Device and Application in Fatigue Life Prediction. *Sensors*, **25**, Article No. 5585. <https://doi.org/10.3390/s25175585>
- [24] Gan, L., Wang, A., Zhong, Z. and Wu, H. (2024) Multiaxial Fatigue Life Prediction Based on Modular Neural Network Pretrained with Uniaxial Fatigue Data. *Engineering Computations*, **42**, 2314-2335. <https://doi.org/10.1108/ec-11-2023-0852>
- [25] Wang, H., Li, B., Lei, L. and Xuan, F. (2024) Multi-Physics Information-Integrated

- Neural Network for Fatigue Life Prediction of Additively Manufactured Hastelloy X Superalloy. *Virtual and Physical Prototyping*, **19**, e2368652.  
<https://doi.org/10.1080/17452759.2024.2368652>
- [26] Bao, K., Dai, J., Liu, Y. and Zhang, Q. (2021) Analysis of Virtual Fatigue Life of Welding Structure of Tracked Vehicle Based on Load Spectrum. *Journal of Physics: Conference Series*, **1986**, Article ID: 012049.  
<https://doi.org/10.1088/1742-6596/1986/1/012049>
- [27] Guo, Y., Li, L., Gui, J. and Hu, S. (2025) Neural Network Approaches for Real-Time Fatigue Life Estimation by Surrogating the Rainflow Counting Method. *International Journal of Fatigue*, **197**, Article ID: 108941.  
<https://doi.org/10.1016/j.ijfatigue.2025.108941>
- [28] Yan, X., Zhang, J., Zhang, J., Wu, Y., Zhang, J. and Xu, L. (2025) Methodology for Compiling Torque Load Spectra of Tractor Power Take-Off Shafts Based on Nonlinear Damage Accumulation. *Engineering Fracture Mechanics*, **314**, Article ID: 110685.  
<https://doi.org/10.1016/j.engfracmech.2024.110685>
- [29] Luo, J., Wang, S., Liu, X. and Geng, S. (2021) Fatigue Life Prediction of Train Wheel Shaft Based on Load Spectrum Characteristics. *Advances in Mechanical Engineering*, **13**. <https://doi.org/10.1177/1687814021992153>
- [30] Adam, M.A., Karam Singh, S.S., Abdullah, S. and Md. Jedi, M.A. (2025) Strain-Based Fatigue Reliability Assessment of an Automobile's Lower Arm for Various Road Load Conditions Due to Limited Experimental Data. *Jurnal Kejuruteraan*, **37**, 79-96.  
[https://doi.org/10.17576/jkukm-2025-37\(1\)-07](https://doi.org/10.17576/jkukm-2025-37(1)-07)