

A deep learning approach for low-cycle fatigue life prediction under thermal–mechanical loading based on a novel neural network model

Yang Yang^{a,b}, Bo Zhang^{a,b}, Hao Wu^{a,b}, Yida Zhang^{a,b}, Hong Zhang^{a,b,*},
Yongjie Liu^{a,b}, Qingyuan Wang^{a,b,c}

^a Failure Mechanics and Engineering Disaster Prevention Key Laboratory of Sichuan Province, College of Architecture and Environment, Sichuan University, Chengdu 610065, China

^b Key Laboratory of Deep Underground Science and Engineering, Ministry of Education, Sichuan University, Chengdu 610065, China

^c School of Architecture and Civil Engineering, Chengdu University, Chengdu 610106, China

ARTICLE INFO

Keywords:

Low-cycle fatigue
Life prediction
Convolutional neural network
Transformer
Deep learning

ABSTRACT

In this study, a novel neural network model is designed to predict the low-cycle fatigue (LCF) life under thermal–mechanical loading. The model is composed of three sub-networks: (1) a Convolutional Neural Network (CNN), (2) a Transformer, and (3) a Fully Connected Neural Network (FCNN). These three sub-networks function as the Backbone, Neck, and Head of the model, respectively. The model, named ConTrans (an abbreviation for **C**onvolutio**n**al Neural Network-**T**ransformer), utilizes binarized hysteresis images as input. The predictive capability of the ConTrans model for LCF life has been validated using experimental data from four different materials. The results demonstrate that almost all predictions fall within the 2-factor band, confirming the model's accuracy and effectiveness.

1. Introduction

In practical engineering, mechanical components are usually subjected to repeated thermal–mechanical loading, which leads to fatigue deformation, particularly low-cycle fatigue (LCF). This can cause premature failure of components under loading at different temperatures. Establishing an effective LCF life prediction model is a vital technology to ensure the safe operation of key products and major facilities, which is of great strategic significance for the aerospace industry, transportation, and nuclear power plants, among other fields. The Coffin–Manson [1] relationship, commonly utilized in fatigue life prediction, described the association between plastic strain amplitude and LCF life. The energy-based fatigue life model [2–5] has the capability to integrate stress–strain and hysteresis loops for the determination of fatigue damage and the prediction of fatigue life. Lee et al. [6] developed a novel life prediction model using an energy-based fatigue damage parameter, the non-dimensional plastic strain energy density (PSED). Zhu et al. [7] introduced a novel energy equivalent damage parameter (EDP), which leverages uniaxial fatigue data for purpose of predicting fatigue life under multiaxial fatigue loading conditions. Gan et al. [8] proposed a new energy-based model to assess LCF life by using the plastic strain energy principle and the critical surface approach. Mandegarian et al. [9] established a fatigue failure criterion based

* Corresponding author at: Failure Mechanics and Engineering Disaster Prevention Key Laboratory of Sichuan Province, College of Architecture and Environment, Sichuan University, Chengdu 610065, China.

E-mail address: zzhanghong@scu.edu.cn (H. Zhang).

<https://doi.org/10.1016/j.engfracmech.2024.110239>

Received 17 December 2023; Received in revised form 15 May 2024; Accepted 11 June 2024

Available online 24 June 2024

0013-7944/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

on normalized energy, which computed the strain energy using off-axis stress and strain components. Gao et al. [10] substituted the maximum normal stress in the original Smith-Watson-Topper (SWT) model with the equivalent stress amplitude and then reinterpreted the SWT parameters as the equivalent strain energy density (ESED). Therefore, a half-life hysteresis loop is usually utilized in the fatigue life study as it contains stress–strain as well as energy information and is critical in life prediction. These models, however, are all based on physical knowledge and require prior in-depth study on the mechanical properties of the materials, while relying on a large amount of experimental data. In contrast, machine learning approaches can utilize existing data to automatically learn and extract the key factors affecting LCF life, leading to more general and accurate predictions.

Machine learning (ML) models have been widely used as a fatigue life prediction method, characterized by a good balance between computational efficiency and prediction accuracy. In the LCF domain, Xu et al. [11] integrated three characteristics of the Nickel-based superalloys and proposed four ML models to predict LCF life, and evaluated them by R^2 value. Duan et al. [12] developed ML models with crack extension rate, residual stress, and average strain as the inputs to predict the LCF life of 316 austenitic stainless steel. Long et al. [13] employed five ML models to predict LCF life of lead-free solders by considering loading, composition, and geometry factors.

Contrary to the above ML techniques, an advanced approach known as deep learning boasts a more robust representational capability [14]. Notably, the neural network is the most extensively utilized tool within this method. For example, Zhang et al. [15] formulated a method based on deep learning that enables the concurrent prediction of material fatigue life under creep, fatigue, and creep-fatigue loading conditions. Yang et al. [16] used a long short-term memory network (LSTM) to eliminate the traditional multiaxial fatigue life prediction model limitation of material and loading conditions. Sun et al. [17] utilized the generative adversarial network (GAN) model for the enhancement of multi-axis fatigue data, which resulted in a significant performance improvement of several machine learning models. For the prediction of gear contact fatigue life, Zhou et al. [18] proposed a hybrid physically-based and data-driven approach and a deep belief network model was developed to show the relationship between various parameters and fatigue life. Halamka et al. [19] proposed a novel hybrid physically-informed neural network in which the LSTM/GRU unit was combined with the fully-connected layer for extracting the damage parameters of the loading cycle.

Convolutional neural networks (CNN) and attention mechanisms are infrequently employed in the study of fatigue life. Kamiyama et al. [20] applied CNN to predict fatigue crack development with satisfactory results. Heng et al. [21] used CNN to extract features from the loading sequence and the LSTM captured the time series features from the output of CNN. Lastly, a fully connected layer was utilized for dimensional transformation. Sun et al. [22] proposed CNN models for hysteresis loop image recognition to predict multiaxial LCF life. However, this method encountered difficulties when handling complex loading conditions, such as thermal–mechanical loading. Regarding attention mechanisms, Yang et al. [23] used them to directly simulate a highly nonlinear mapping between multiaxial fatigue life complex loading conditions. Despite the popularity of the attention mechanism-based Transformer [24] in recent years, it has not been applied in the fatigue life prediction domain. In many conventional models, LCF life of a specific material is determined by several factors, such as the loading conditions, ambient conditions (like temperature), and material parameters, etc. Most of the deep learning approaches mentioned above are based on these factors, which are limited by the data obtainment and model training.

In this study, without requiring loading conditions and other factors of the material as additional inputs, a novel neural network model named ConTrans is developed to predict the LCF life from a more general input—the half-life hysteresis loop image. The model's accuracy is tested using experimental data from four materials subjected to thermal–mechanical loading. The proposed neural network model consists of three sub-networks: CNN, Transformer, and Full Connected Neural Network (FCNN), which act as the Backbone, Neck, and Head of the model respectively. The paper is organized as follows: (1) The detailed algorithm of the established model is introduced; (2) The framework and the determination of the model parameters are demonstrated in Section 3; (3) In Section 4, the model performance is evaluated and discussed; (4) The conclusion and outlook are presented in Section 5 and 6, respectively.

2. Algorithms

CNN [25] has gained significant traction in the field of Computer Vision (CV), owing to its superior capabilities in processing image information. On the other hand, the Transformer model [24], which serves as the cornerstone of the Generative Pre-trained Transformer (GPT) [26], has demonstrated exceptional performance in the field of Natural Language Processing (NLP).

In this study, a novel neural network model, named as **ConTrans** (Convolutional Neural Network-Transformer), is established that amalgamates the strengths of CNNs in local information processing and Transformers in global information processing. This fusion enables the model to establish a complex mapping relationship between hysteresis loop images and LCF life.

As depicted in Fig. 1, the ConTrans model comprises three key components: the Backbone (CNN), the Neck (Transformer), and the

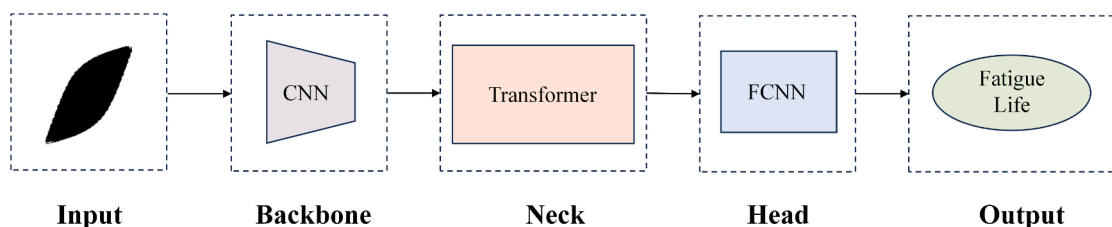


Fig. 1. The architecture diagram of ConTrans.

Head (FCNN). The Backbone is designed to receive hysteresis loop images, which are then fed into the Neck in a unified tensor format. The Neck processes this information and passes it to the Head, which ultimately predicts the fatigue life. This innovative approach leverages the best of both worlds, promising enhanced performance in fatigue life prediction.

Based on the above analysis, ConTrans model is established in this study, including three important elements: (1) a CNN as Backbone, (2) a Transformer as Neck, and (3) a FCNN as Head. These three elements will be introduced in the following subsections, respectively.

2.1. Backbone: Convolutional neural network

A typical convolutional neural network is illustrated in Fig. 2, which is established with input layer, basic unit: ‘convolutional – activation- pooling’, and dense layer:

- (1) Input layer: This layer receives the original image data, and each image can be represented as a three-dimensional matrix: **(Channels, Height, Width)**, in which, Channels present the color channels of the image (red, green, blue). The grayscale image is composed of only black and white color, whose channel is 1.
- (2) Convolutional layer: This layer convolves the input image with a convolutional kernel to extract features from the image.
- (3) Activation layer: This layer introduces nonlinearity to the output of the convolutional layer by applying an activation function, which enables the network to learn complex features [27] and can be expressed as follows:

$$Y = f(X * W + b) \quad (1)$$

In which, f is the activation function (such as Rectified Linear Unit(ReLU) [28], Sigmoid). X is the input. W is the Filter or Kernel. b is the bias, and $*$ is convolution operation.

- (4) Pooling layer: This reduces the size of the feature maps output by the convolutional layer, thus reducing the amount of computation and the number of parameters and preventing overfitting, and can be expressed as follows:

$$Z = g(Y \downarrow s) \quad (2)$$

In which, Z is the output of pooling layer. g is the aggregation function. Y is the output of the convolutional layer. $\downarrow$ is the sampling operation and s is the window size. The global average pooling (GAP) layer, as referenced in [29], operates by averaging the entire feature map from the preceding layer to generate a single feature point. These feature points are then aggregated to construct the final feature vector.

- (5) Dense Layer: This layer synthesizes and classifies the features extracted from the previous layers and outputs the final prediction results.

$$O = h(Z \cdot W + b) \quad (3)$$

In which, O is the output of fully connected layer. h is the activation function. Z is the output of the previous layer. W is the weighting matrix. b is the bias, and $\cdot$ is matrix multiplication.

In this model, the CNN serves as Backbone, responsible for receiving images and learning their features.

2.2. Neck: Transformer

A Transformer model can be represented by Fig. 3, and generally consists of several vital components, which are introduced separately in the following as exemplified by the NLP task.

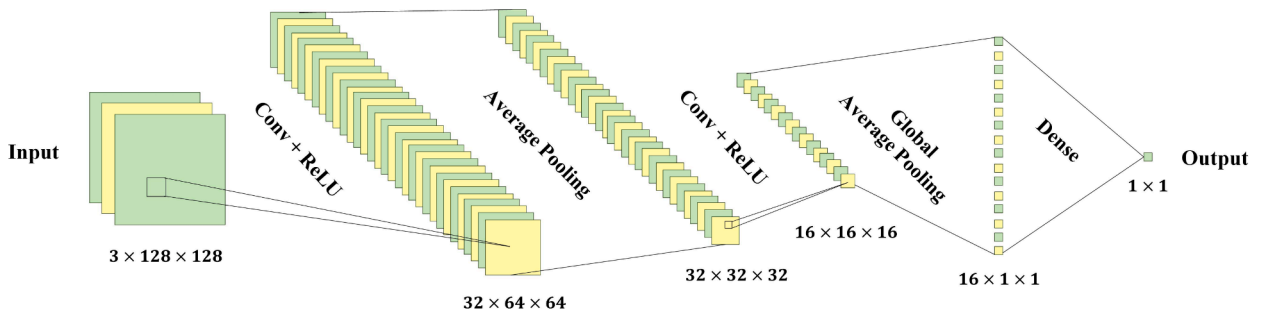


Fig. 2. Diagram of a typical Convolution Neural Network.

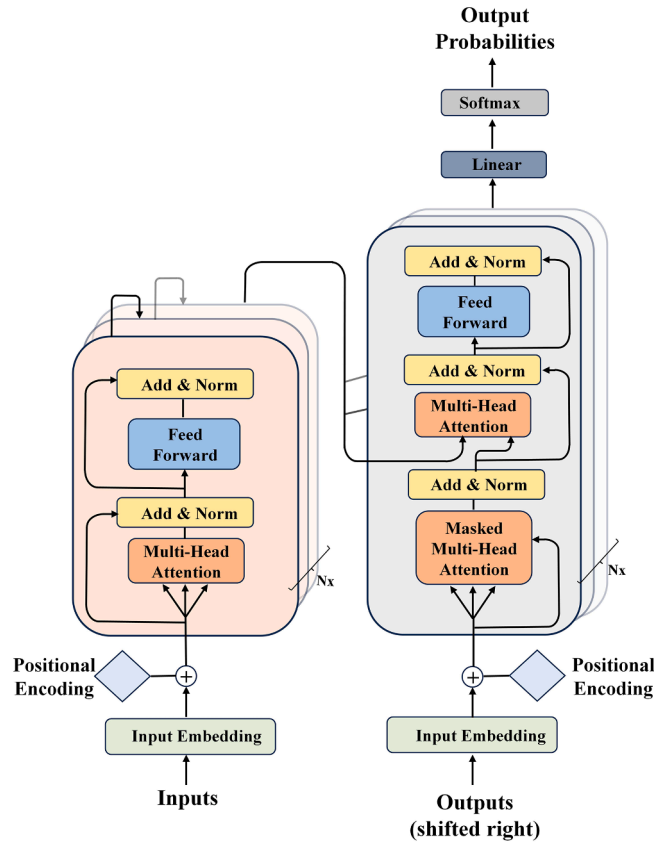


Fig. 3. The diagram of Transformer model.

2.2.1. Positional encoding

In NLP tasks, words are fed into the Transformer model with the addition of a Positional Encoding (PE). This is because a Transformer model, unlike the structure of RNN, does not directly utilize the sequential information of the words, instead utilizing global information. The role of positional encoding is to maintain the relative and solute position of the word in the sequence. The dimensions of the positional encoding and the word are identical, which allows for their direct summation.

In the original work of Transformer [24], the PE matrix is calculated as:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (4)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (5)$$

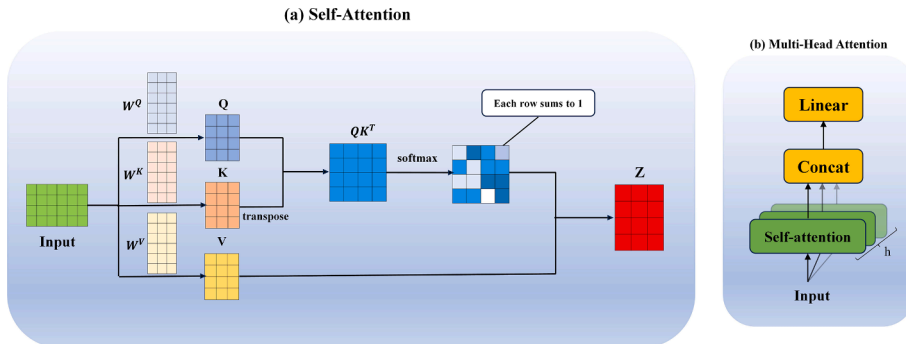


Fig. 4. Illustrations of Attention mechanism: (a) Self-Attention mechanism, (b) Multi-Head Attention mechanism.

where pos is the position of the word in the sentence. d is the dimension of the PE . $2i$ is the even dimension and $2i+1$ is the odd dimension. By adding the above PE with the word vectors, the input X to the Transformer is obtained.

2.2.2. Self-attention mechanism and multi-head attention mechanism

The architecture of Self-Attention [30] is illustrated in Fig. 4(a). For a better understanding, imagine a translation task where the input is a sentence with four words. Each row represents a word, and every word can be determined by a 1 by 6 vector. Three linear transformation matrices are employed in the computation: W_Q , W_K , W_V , which are randomly initialized at the beginning of training.

Initially, the input X is subjected to multiplication with the three matrices as follows:

$$Q = X \cdot W_Q \quad (6)$$

$$K = X \cdot W_K \quad (7)$$

$$V = X \cdot W_V \quad (8)$$

in which Q , K , V are Query, Key, and Value matrices, respectively. Notice that each row of X , Q , K , V represents a word.

Then, the matrix obtained by multiplying Q by the transpose of K is 4 by 4, which represents the relationship between every word. Subsequently, the output of the self-attention mechanism can be derived from Q , K and V , which are calculated as follows:

$$Z = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (9)$$

where softmax is an activation function that normalizes numerical vectors into probability distribution vectors with probabilities that sum to 1, which means correlation coefficients between words. K^T is the transpose matrix of K and d_k is the number of columns of K . Z is the output of self-attention, which represents the relation matrix of the input sentence.

Multi-Head Attention [24] is illustrated in Fig. 4(b), in which multiple parallel self-attentions are employed to extract different attention heads of input sentences simultaneously. Feed input X into each of the h different self-attentions, and one can get h Z_i , ($i=1,2,\dots,h$). The output Z of the multi-head attention is the concatenation of the h Z_i after processed by a linear layer. Multi-Head Attention allows for joint attention to information from different representation subspaces at different locations, which is essential in complex tasks.

Masked-Multi-Head-Attention [24] is a special attention mechanism. It allows the model to focus only on the previous position and ignore the subsequent positions when generating each position of the output sequence. That prevents the model from seeing future information and ensures the correctness and consistency of the output sequence.

2.2.3. Feed forward and add & norm

Feed Forward is a two-layer Full Connected Layer, the activation function of the first layer is $ReLU$, and the second layer does not have an activation function, and the corresponding formula is as follows:

$$\max(0, XW_1 + b_1)W_2 + b_2 \quad (10)$$

In which, X is the input of Feed Forward. W_1 , W_2 are weighting matrices and b_1 , b_2 are bias.

The purpose of the Feed Forward is to introduce non-linearity and allow the model to learn complex relationships between features, which are essential for tasks like language translation, text generation, and more.

Add is $(X + \text{Multi-HeadAttention}(X))$ or $(X + \text{FeedForward}(X))$, which is a type of residual connection that is usually applied in deep networks [31] to improve their performance. Add layer is illustrated in Fig. 5.

Norm is Layer Normalization [32], which speeds up convergence by converting the inputs of each layer of neurons to have the same variance. The Add & Norm Layer is calculated as follows:

$$\text{LayerNorm}(X + \text{Multi-HeadAttention}(X)) \quad (11)$$

$$\text{LayerNorm}(X + \text{FeedForward}(X)) \quad (12)$$

The purpose of the Add & Norm layer is to stabilize training and improve gradient flow during backpropagation.

2.2.4. The process of transformer

The Transformer [24] model, as illustrated in Fig. 3, utilizes an encoder-decoder architecture, where the encoder transforms the input sequence into a continuous expression, and the decoder generates an output sequence based on that expression. Taking the NLP

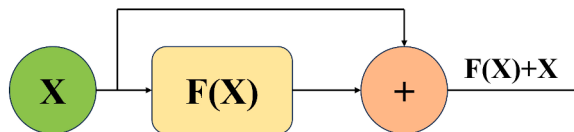


Fig. 5. Add Layer.

task as an example, the basic process of the Transformer is described as follows:

Firstly, each word of the input sequence is converted into a vector, which is obtained by summing the word embedding and the position's embedding. That preserves the semantic and sequential information of the words.

Then, these vectors are fed into an encoder, which consists of multiple identical layers, each containing a multi-head attention mechanism, residual connection, and feedforward neural network. The Multi-Head Attention mechanism calculates the correlation of each position in the input sequence with other positions and weights the input sequence according to these correlations. The Feed Forward performs a nonlinear transformation of the input sequence to improve the expressiveness of the model. Finally, the encoded information is obtained.

After that, the output of the encoder is fed into the decoder, the structure of the decoder is basically similar to that of the encoder, but the decoder has one more Masked Multi-Head Attention, which is mentioned in [Section 2.2.3](#). Again, after the Multi-Head Attention mechanism, residual connections, and Feed Forward, the decoded information is obtained. This study is a regression prediction task, so it does not need Masked Multi-Head Attention. Only Transformer Encoder module is employed in this model.

Eventually, the output of the decoder passes through a linear layer and a *softmax* layer, obtaining the word probability at each position, and then selects the output word according to a certain strategy.

In the work, Transformer Encoder module is employed as Neck in this model for further fusion of features.

2.3. Head: Full connected neural network

FCNN, as illustrated in [Fig. 6](#), is one of the most basic artificial neural network structures, also known as Multilayer Perceptron (MLP) [\[33,34\]](#).

In this work, FCNN is employed as the Head of the ConTrans model. The choice of FCNN as the Head of the model is primarily due to its ability to learn complex features of the input data. By connecting each neuron to all neurons in the previous and subsequent layers, the FCNN forms a densely connected structure that enables the network to learn and represent high-level features of the input data. In addition, FCNN is highly flexible and can be easily scaled to larger network structures, to handle more complex tasks. It is therefore suitable as a Head for various types of models.

3. Framework of the model

3.1. Diagram of ConTrans model

The comprehensive diagram of the ConTrans model is depicted in [Fig. 7](#).

Initially, the binarized hysteresis loop image, represented as a single channel, is input into the CNN, the backbone of ConTrans. This image traverses through N basic units of CNN. As introduced in [section 2.1](#), the features of the image are preliminarily learned through these network layers. It is mentioned that the final layer is the global average pooling layer. The global pooling layer ensures that a

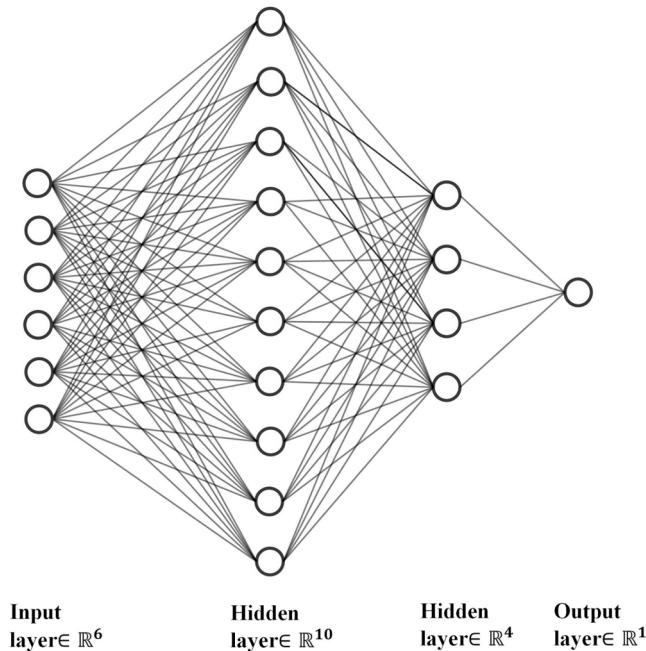


Fig. 6. Illustration of FCNN.

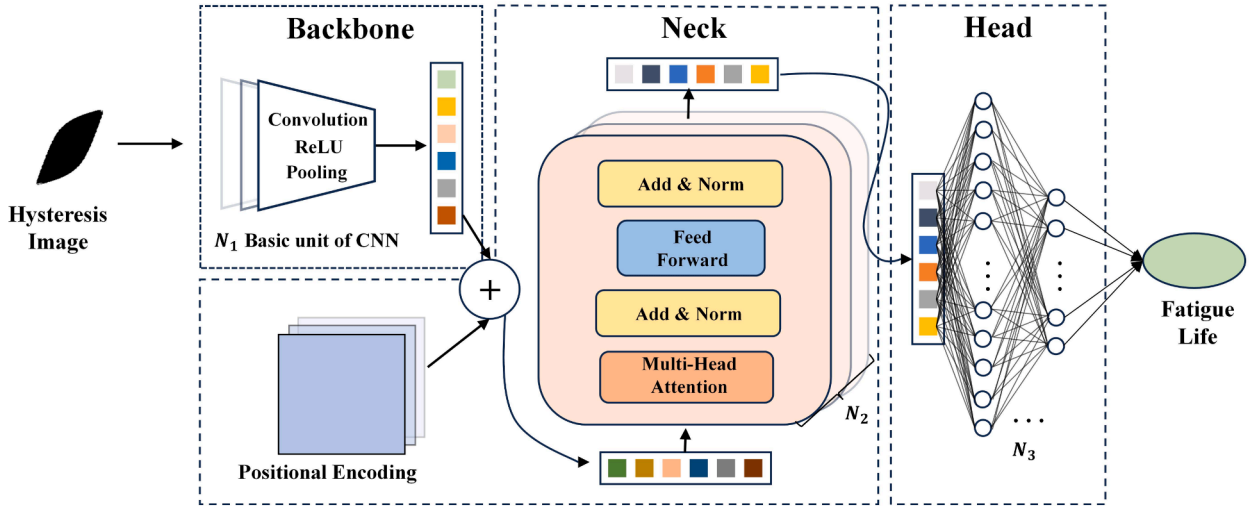


Fig. 7. The whole detailed diagram of ConTrans model: Input, Backbone, Neck, Head, Output.

feature tensor of uniform size is obtained, regardless of the initial image size, which is critical for the following feature learning. After CNN processing, the image data is transformed into a tensor that is utilized for further feature extraction in the subsequent neck network.

Following this, the image feature tensor, along with Positional Encoding, is fed into the Transformer Encoder module. As introduced in section 2.2, this module further integrates the global features using the Multi-Head Attention mechanism, Residual connections, and Feed-forward operations. The Multi-Head Attention mechanism calculates the correlation of each position with other positions in the input tensor processed by CNN. The Residual connections help to improve the learning performance. The Feed Forward layer introduces the non-linearity of the learned feature. In conclusion, the neck network of ConTrans model enhances the model's ability to capture complex patterns and dependencies in the data.

Finally, the feature tensor is input into the FCNN to predict the fatigue life. As introduced in section 2.3, this final step transforms the high-level features extracted and processed by the previous layers into a specific prediction, providing a direct, data-driven estimate of the LCF life.

This process, from initial input to final prediction, illustrates the seamless integration of different neural network architectures in the ConTrans model, each contributing to the accurate prediction of LCF life. The model's design ensures that it can handle varying image sizes, extract, and process complex features, and make accurate predictions, all while minimizing the need for human intervention. This makes ConTrans a robust and versatile tool for predicting LCF life from half-life hysteresis loop images.

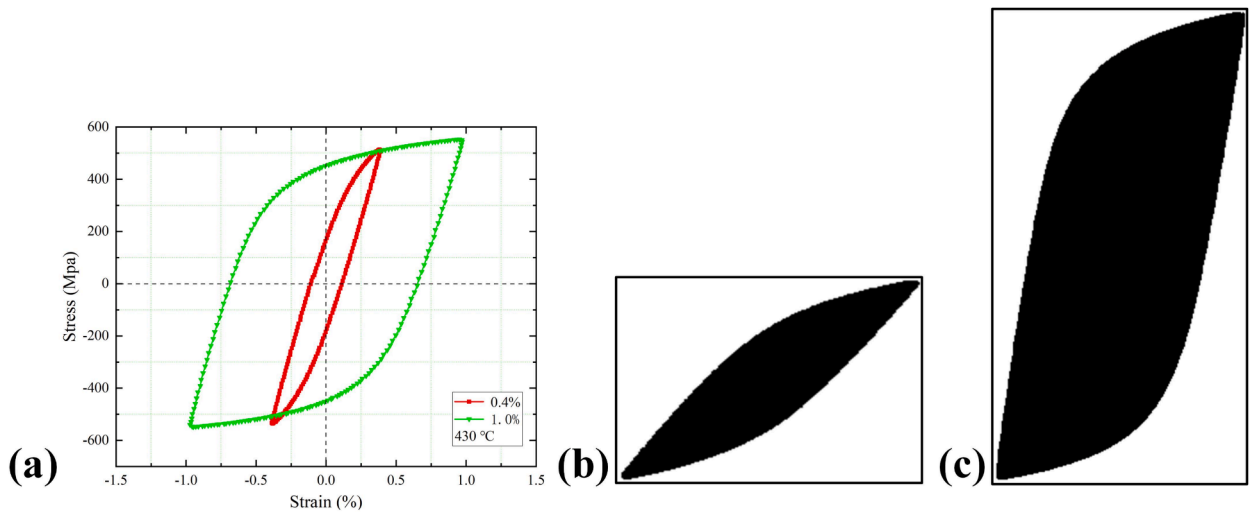


Fig. 8. (a) The experimental data with 0.4% and 1.0% strain amplitude under 430 °C. (b) and (c) The hysteresis image of data with 0.4% and 1.0%, respectively.

3.2. Data pre-processing

The hysteresis loops contain information on stress–strain and energy, which are usually utilized to predict fatigue life, such as the classical Ramberg-Osgood [35] relationship. Hysteresis loops can be easily obtained from stress-strain curves without complex mathematical processing. In this study, the half-life hysteresis loops are converted into image data, which are then used as input for the ConTrans model. This is because there is a stabilization period existing within the fatigue cycle, and the half-life hysteresis loops fall within this stabilization period. In this work, the hysteresis image is a binary image with only one-color channel and two possible values: black or white. The image size depends on the maximum stress and strain values in the experiment, so it can change for different cases.

To explain how the hysteresis image is processed, two examples of MarBN steel with different strain amplitudes (0.4 % and 1.0 %) at 430 °C, as displayed in Fig. 8. First, plot the hysteresis loop with strain on the x-axis and stress on the y-axis, as displayed in Fig. 8(a).

The area enclosed by the hysteresis loop is the strain energy. In consideration of the effectivity and reliability of the strain energy criterion [36,37], to emphasize the enclosed area of the hysteresis loop, a binary method is employed to process the hysteresis loop images [22]. The binary method is performed as follows:

$$P(x,y) = \begin{cases} 0, & (x,y) \in \Omega \\ 1, & (x,y) \notin \Omega \end{cases} \quad (13)$$

In which, (x,y) is the location of pixels. Ω is the boundary and interior region of the hysteresis loop. $P(x,y)$ is the pixel value at (x,y) . Thus, the pixels in region Ω set to 0, and which are black in the image. The pixels outside of the region Ω are set to 1, and which are white in the image. The hysteresis loops with 0.4 % and 1.0 % strain amplitude correspond to Fig. 8(b) and Fig. 8(c), respectively.

Image data quality depends on two main factors: scale and resolution. These two factors influence the amount of information in the image and the performance of the model [22]. The image scales of all samples are normalized to 0.1 % per inch in the stress direction and 100 MPa per inch in the strain direction. In this way, the number of pixels in the height and width directions of the hysteresis image are calculated as follows:

$$H = \frac{\max(stress) \times P}{2 \times S_{stress}} \quad (14)$$

$$W = \frac{\max(strain) \times P}{2 \times S_{strain}} \quad (15)$$

In which, H and W are the pixel number in the height and width direction, respectively. $\max(stress)$ and $\max(strain)$ are the maximum value of the stress and strain, respectively. S_{stress} and S_{strain} are the scale of stress and strain directions, which are set as 100 *Mpa/inch* and 0.1 %/*inch*, respectively. P is the image resolution.

Since convolutional neural networks are characterized by translation invariance, data augmentation of image data is a common means of processing data for image tasks in computer vision. In this study, data augmentation is employed to enhance the model performance. Additionally, the data values for fatigue life, which span a wide range and vary significantly in order of magnitude, require normalization prior to training. This normalization allows the model to converge more rapidly to an optimal solution [38]. For fatigue life N_f , scaling it using the logarithmic transformation $\log N_f$, has proven to be effective.

3.3. Hyperparameters

Prior to training the ConTrans model, it's necessary to adjust the hyperparameters that shape the model architecture, such as the number of basic units of CNN and the dimensions of the fully connected layers. These adjustments are made in accordance with the complexity of the practical case. Theoretically, the model's ability to model nonlinear mapping relationships improves with higher dimensionality and deeper layers. The hyperparameters involved in ConTrans are listed in Table 1.

For Item (1), the resolution P refers to the resolution of hysteresis loop image. If the resolution is too small, one pixel may

Table 1
Lists of hyperparameters.

Item	Symbol
(1) Resolution of the hysteresis image	P
(2) Dimensions of out-channels in CNN	d_c
(3) Size of kernel	s_k
(4) Size of pooling-kernel	s_p
(5) Size of global pooling-kernel	s_g
(6) Number of basic units of CNN	num_c
(7) Number of attention heads	h
(8) Number of Transformer Encoder layer	num_e
(9) Dimension of input for Transformer	d_m
(10) Dimensions of full connected layers	d_f
(11) Number of hidden layers	num_f

encompass a large range of data, leading to the distortion of the hysteresis loop. As the resolution increases, the images become smoother, and the shape becomes clearer. However, higher resolution incurs greater computational cost, necessitating a balance. In this work, the resolution is set as 50 pixels per inch (ppi).

Items (2) to (6) determine the complexity of the CNN (Backbone), which serves as the primary control hyperparameter. Items (7) to (9) and Items (10), (11) determine the Neck and Head of ConTrans, respectively. In theory, higher values of these hyperparameters make the model better at dealing with the nonlinear problem. But with that, the training process becomes more challenging and time-consuming.

3.4. Determination of model parameters

3.4.1. Model initialization

In contrast to hyperparameters, model parameters are derived from the learning process. These parameters, which require training (for instance, the linear transformation matrix), must be initialized. This study employs Xavier initialization [39], designed to maintain the variance of each layer's output close to that of its input. This approach helps to circumvent issues such as exploding or vanishing gradients. During the training phase, these values are continually updated via the backward propagation method to minimize the loss function, which is the discrepancy between the predicted fatigue life and the actual value.

3.4.2. Loss function

From trying a few of mainstream loss functions (such as mean absolute error, mean square error, and mean absolute relative error), it is found that the loss function that blended L_1 Loss and L_2 Loss is the most efficient and can avoid vanishing gradient problems, and the Loss Function can be expressed as the following equation,

$$L_1 = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (16)$$

$$L_2 = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (17)$$

$$\text{Loss function} = \omega_1 \cdot L_1 + \omega_2 \cdot L_2 \quad (18)$$

where y_i is the true value of the i th sample, $\hat{y}_i$ is the predicted value of the i th sample, and n is the total number of samples. and ω_2 are the weights of L_1 Loss and L_2 Loss, respectively, which are set at 0.8 and 0.2 before training in the study.

3.4.3. Optimizer

The RAdam optimizer [40] equipped with a dynamic learning rate and weight decay, is utilized to enhance the learning process. Learning rate decay [41,42] is an optimization strategy that dynamically modifies the learning rate based on the progression of training. The objective of learning rate decay is to employ a larger learning rate during the initial stages of training to expedite convergence, and a smaller learning rate in the later stages to enhance the model's accuracy and stability. In this study, the initial and minimum learning rates are set at 1×10^{-4} and 5×10^{-8} , respectively, with a decay factor of 0.5. It is found that weight decay [41], a regularization technique, is particularly effective in preventing overfitting for prediction tasks in this study. RAdam, an optimization algorithm based on Adaptive Moment Estimation (Adam) [43], dynamically adjusts the learning rate throughout the training process. Unlike Adam, RAdam introduces a variable correction factor to counteract the variance bias in Adam, allowing the learning rate to reflect the gradient's signal-to-noise ratio more accurately.

3.4.4. Activation function

Activation functions are crucial for neural networks to model complex nonlinear phenomena and learn efficiently. The choice of activation function depends on the research task. Generally, it is recommended to utilize a *sigmoid* and *tanh* function in classification task, and a *ReLU* function [28] (i.e., $\text{ReLU}(x) = \max(0, x)$) is a more universal choice. However, in this work, it is found that *LeakyReLU* function [44] has better performance than the *ReLU* function. And the *LeakyReLU* function is adopted as activation function of Backbone (CNN) and Head (FCNN). The detailed methodology of *LeakyReLU* function is as follows:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \lambda x, & x < 0 \end{cases}, \lambda \in (0, 1) \quad (19)$$

in which, λ is slope parameter and set as 0.01 in this work. For the history dependent problem, the Gaussian error linear unit (*GELU*) [45] may be the most effective. In this study, the *GELU* is adopted in Neck (Transformer):

$$\text{GELU}(x) = 0.5x \left(1 + \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{\sqrt{2}}} e^{-t^2} dt \right) \quad (20)$$

In summary, the parameters of ConTrans model are listed in Table 2.

3.5. Training process

The procedure of model training is shown in the Fig. 9, which is interpreted as follows:

Firstly, the hysteresis loop is binarized. Before starting the training, the model is initialized. Then, the hysteresis image is fed into the ConTrans model, followed by forward propagation to compute the loss value. After evaluating the loss value, backward propagation is performed while the gradient is zeroed and the learning rate and weights decay. The model gradient decreases and converges gradually until the loss value is stable. Finally, the predicted fatigue life is obtained.

4. Case study and discussion

4.1. Experimental data

In the current section, the performance of ConTrans is evaluated by predicting the LCF life of four materials, including (1) case A: heat-resistant steel of MarBN steel [46] for double phases structure including body-centered cubic (BCC) and face-centered cubic (FCC), (2) case B: nickel-base superalloys of MAR M247 for FCC, (3) case C: Additive manufacturing 304 steel (AM 304) [47] for FCC and case D: 2024 aluminum alloy (2024AA) collected from this literature [48]. The corresponding temperatures and strain amplitudes for the four materials are shown in Table 3–6. The experimental data contains 15 samples of MarBN in Table 3, 31 samples of M247 in Table 4, 24 samples of AM 304 in Table 5, and 20 samples of 2024AA in Table 6.

4.2. Experimental procedure

Strain-controlled LCF tests were carried out at different temperatures. Cylindrical specimens were fabricated from all materials, with gauge sections of 5 mm and 6 mm in diameter and a dimension tolerance of 0.02 mm. The detailed geometry of the specimen is shown in Fig. 10. At least two specimens were used for testing at different temperatures at each strain level and to keep the temperature constant, the specimens were incubated for 20 min before testing. Tests were performed using symmetrical triangular waveforms at strain ratio $R = -1$ and strain amplitude control mode. A clamp-on extensometer with an axial 12 mm measuring length was used to monitor strain during fatigue tests. Besides, the stress and strain are obtained by the force sensor and extensometer during the cyclic loading, and the hysteretic loop at each cycle is recorded with 400 data points. The schematic of the hysteretic loop is presented in Fig. 11.

4.3. Performance

The performance of the ConTrans model is compared with that of the CNN and Transformer models evaluated by four cases: A. MarBN, B. M247, C. AM304, and D. 2024AA, with the hyperparameters for each model detailed in Table 7. Values of hyperparameters in four cases. It should be noted that the CNN and Transformer models are configured with the same hyperparameters as the ConTrans model. The prediction results for all three models are illustrated in Fig. 12. Upon analyzing the errors in the predictions, it is observed that the ConTrans model yields higher accuracy compared to the CNN and Transformer models, with all predictions falling within a scatter band of twice the standard deviation.

The performance of the ConTrans model is further evaluated by calculating the L1 loss and L2 loss between the predicted results and the experimental data, as shown in Table 7. Whereas, the performance of three different models (CNN, Transformer, and ConTrans) with two different loss functions is shown, with the ends of the box representing 25 % and 75 % of the results. The whiskers are the minimum and maximum values, respectively. For a well-performing model, the box-and-line plot should have a lower mean and a narrower sign range. From Fig. 13, it is obvious that the ConTrans model performs best, as it has the narrowest range and the lowest average loss value.

4.4. Discussion

The performance of the ConTrans model was evaluated using experimental data from four materials subjected to thermal–mechanical loading: MarBN, MAR M247, AM 304, and 2024AA. The results demonstrated that the CNN model is capable of extracting features from the hysteresis images, while the Transformer model can capture the temporal dependencies of the hysteresis loops. However, the ConTrans model has superior accuracy and lower loss compared to the CNN and Transformer models, with most predictions falling within a scatter band of twice the standard deviation. Furthermore, the ConTrans model integrates the strengths of both the CNN and Transformer models, outperforming them in LCF life prediction for all four materials. It indicates that the ConTrans model is not sensitive to the structure of the material (BCC and FCC).

Table 2
Parameters of ConTrans model.

Processing for input data	Processing for output data	Model initialization	Optimizer	Loss Function	Activation Function
Binary method	logarithmic	Xaiver	RAdam	L_1, L_2 Loss	LeakyReLU, GELU

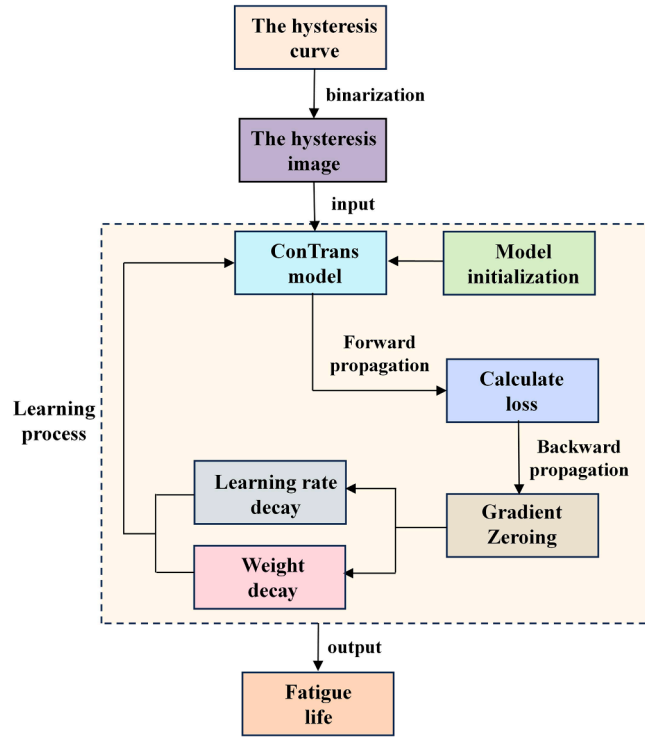


Fig. 9. The procedure of model training.

Table 3

The experimental data of MarBN steel under different temperature.

Temperature	Strain amplitude (%)	Fatigue life (N_f)
25 °C	0.4	7730
	0.5	2585
	0.8	724
	1.0	442
	1.5	152
430 °C	0.4	1140
	0.5	522
	0.8	294
	1.0	318
	1.5	132
630 °C	0.4	1865
	0.5	1668
	0.8	771
	1.0	315
	1.5	117

As for various traditional fatigue life prediction methods, the Coffin-Manson model [1] has been prevalent in the past several years. The Basquin formula [49], which was combined with the Coffin-Manson model, was proposed to form a relation between total strain and fatigue life, as Eq. (21) presents.

$$\frac{\Delta \epsilon}{2} = \frac{\Delta \epsilon_e}{2} + \frac{\Delta \epsilon_p}{2} = \frac{\sigma'_f}{E} (2N_f)^b + \epsilon'_f (2N_f)^c \quad (21)$$

The elastic strain $\Delta \epsilon_e$ and plastic strain ϵ_p can be logarithmically shifted respectively. The σ'_f and ϵ'_f respectively denotes the fatigue strength coefficient and fatigue ductility coefficient, and the b and c describes the fatigue strength exponent and fatigue ductility exponent. Indeed, these parameters can be obtained by curve fitting, which somehow is complex to handle the data. What's more, some studies [50–53] demonstrated that the Basquin-Manson-Coffin model has a bad accuracy under extremely low-cycle fatigue regions due to an alternative failure mechanism under large strain reversals. Later, a novel fatigue life prediction model was proposed by Kuroda [54] to consider the coupled effect of different damage mechanisms, and the corresponding formulas are as follows:

Table 4

The experimental data of MAR-M 247 superalloy steel under different temperature.

Temperature	Strain amplitude (%)	Fatiguelife (N_f)	Temperature	Strainamplitude (%)	Fatiguelife (N_f)
500 °C	0.199	77,758	800 °C	0.198	31,675
	0.296	9142		0.297	1198
	0.474	900		0.298	1587
	0.542	497		0.299	544
	0.614	74		0.399	93
	0.677	53		0.499	362
	0.870	5		0.668	4
	0.879	10			
700 °C	0.295	2580	900 °C	0.199	49,793
	0.297	1420		0.296	1518
	0.298	1295		0.297	1427
	0.299	4026		0.298	2455
	0.682	2		0.299	1324
	0.877	10		0.499	219
	0.977	37		0.681	33
				0.768	7
				0.777	23

Table 5

The experimental data of AM 304 under different temperature.

Temperature	Strain amplitude (%)	Fatigue life 1 (N_f)	Fatigue life 2 (N_f)
25 °C	0.4	7970	8775
	0.5	2370	4990
	0.6	1915	2410
	0.8	259	237
300 °C	0.4	6700	7540
	0.5	1656	4605
	0.6	1635	1175
	0.8	570	138
650 °C	0.4	710	580
	0.5	515	315
	0.6	244	202
	0.8	54	82

Table 6

The experimental data of 2024AA under different temperature.

Temperature	Strain amplitude (%)	Fatigue life (N_f)
20 °C	0.35	58,965
	0.5	9576
	0.8	202
	1.0	69
	2.0	7
100 °C	0.5	10,744
	0.8	353
	1.0	102
	1.5	15
	2.0	6
200 °C	0.5	6044
	0.8	302
	1.0	73
	1.5	27
	2.0	17
300 °C	0.5	382
	0.8	159
	1.0	116
	1.5	71
	2.0	47

$$D_t = \frac{\varepsilon_{pmax}}{\varepsilon_f} \quad (22)$$

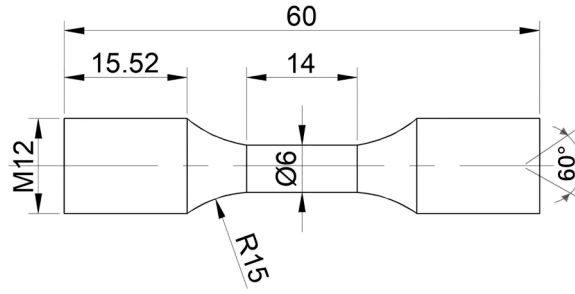


Fig. 10. The size of the fatigue sample (units: mm).

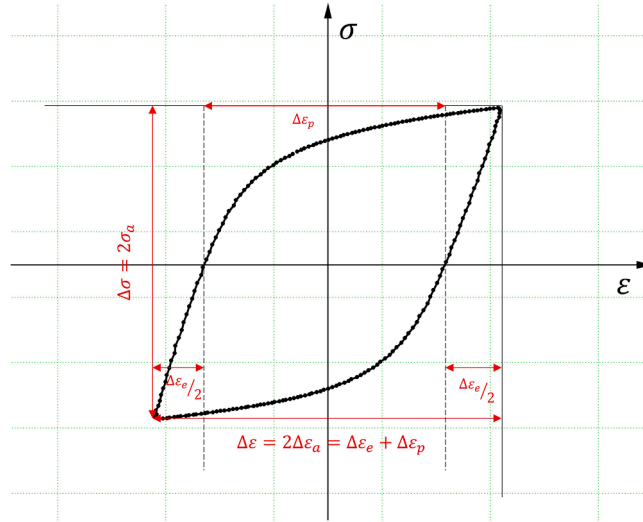


Fig. 11. The schematic diagram of the hysteretic loop.

Table 7

Values of hyperparameters in four cases.

Case	d_c	s_k	s_p	s_g	num_c	h	num_e	d_m	d_f	num_f
A	64	3	2	1	3	8	2	64	64	2
B	128	3	2	2	3	8	2	128	64	2
C	128	3	2	2	3	8	3	128	128	2
D	128	3	2	1	3	8	3	64	64	2

$$D_f = \frac{N}{N_f} = 4N \left(\frac{\Delta \varepsilon_p}{2\varepsilon_f} \right)^{a'} \quad (23)$$

$$D_c = \frac{\Delta \varepsilon_p N_f^{a'}}{C} \quad (24)$$

$$\frac{\varepsilon_{pmax}}{\varepsilon_f} + 4N_f \left(\frac{\Delta \varepsilon_p}{2\varepsilon_f} \right)^{a'} + \frac{\Delta \varepsilon_p N_f^{a'}}{C} = 1 \quad (25)$$

In this model, the damage of metal is divided into three parts, including a monotonic tensile damage part D_t , a ductility exhaustion damage part D_f and a microcrack propagation damage part D_c under small strain reversals. In Eq. (25), the fatigue life is obtained by the three part, where ε_{pmax} represents the maximum tensile plastic strain; ε_f denotes monotonic fracture strain; a' , a' and C describe material parameters. Obviously, the obtainment of these parameters is more difficult than that of Basquin-Manson-Coffin relation.

Inspired by the energy-based fatigue life prediction approach, the proposed ConTrans model offers a novel approach to predict LCF life from half-life hysteresis loop images, which can be easily obtained from the stress-strain data, eliminating the need for additional

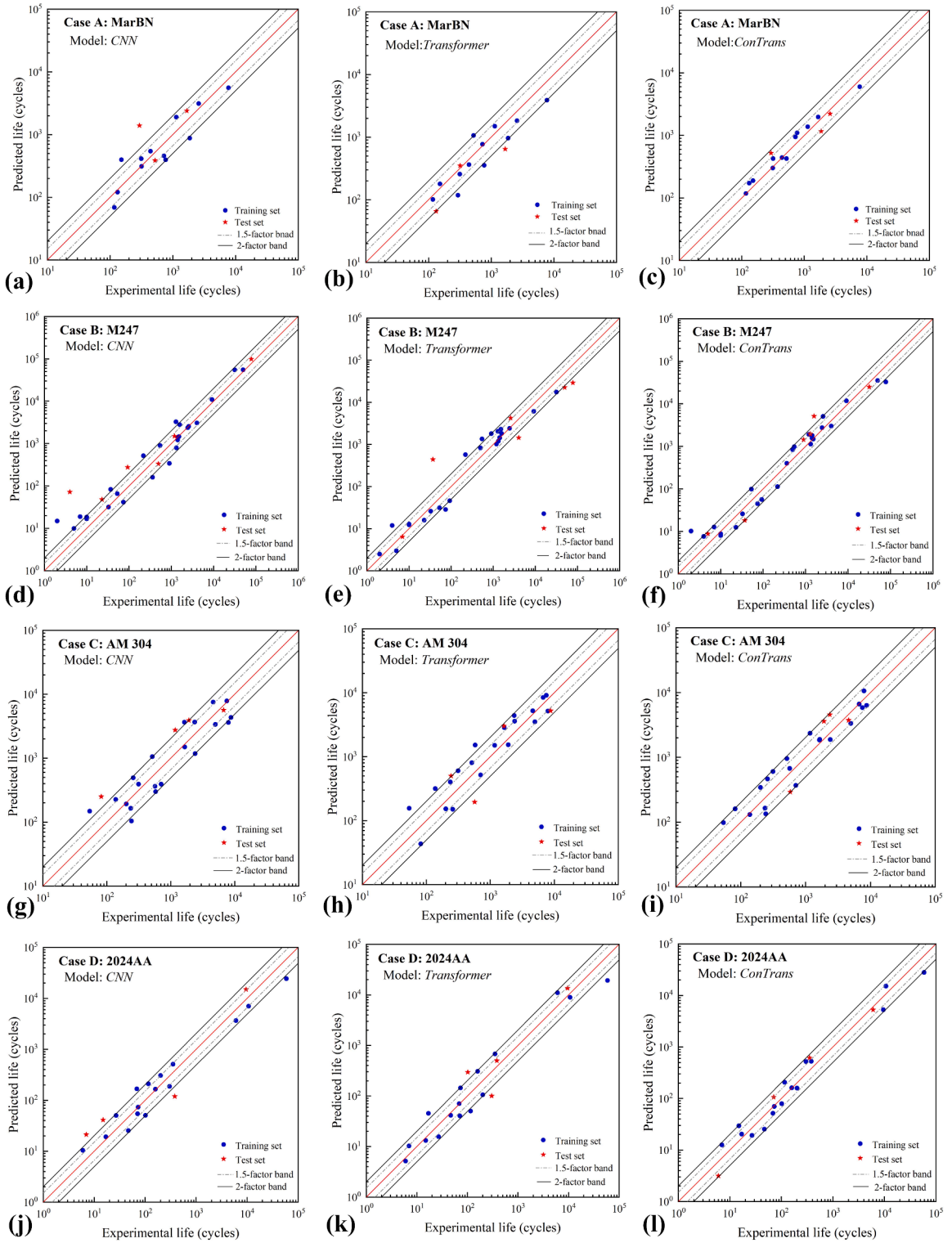


Fig. 12. The comparison of experimental and predicted life of CNN model, Transformer model and ConTrans model.

inputs such as loading conditions and material damage factors. Traditional models are constrained by the researcher's comprehension of the studied phenomenon and the precision of the obtained material parameters. In contrast, the proposed ConTrans model offers a novel approach to predict LCF life from half-life hysteresis loop images, which can be easily obtained from the stress-strain data,

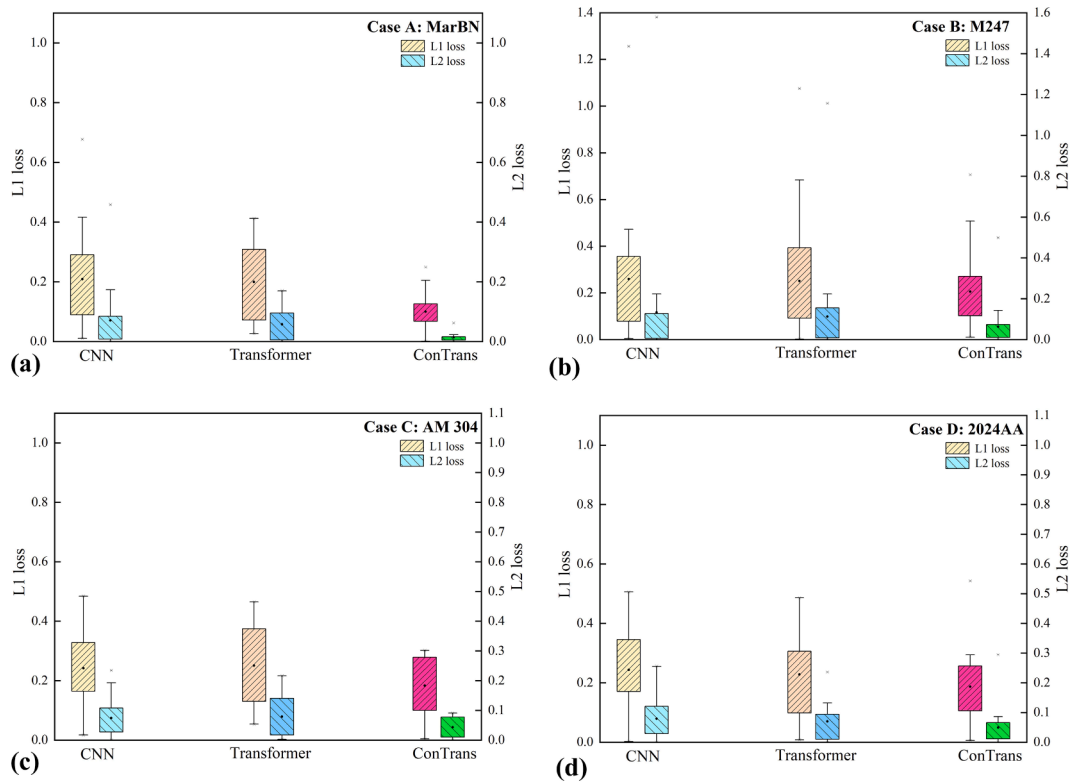


Fig. 13. Model performance of CNN, Transformer and ConTrans model.

eliminating the need for additional inputs such as loading conditions and material damage factors. What's more, the ConTrans model can be applied to a variety of loading conditions and materials, theoretically, provided that the hysteresis images of the loading process are available. However, it's admitted that the ConTrans model requires training with a dataset specific to a certain material to obtain model parameters that are only valid for that material.

The ConTrans model shows promise as a tool for LCF life prediction, providing rapid and accurate results without the need for additional information about the loading conditions and damage factors of the material, which could prove beneficial for material design, optimization, and testing, as well as for failure analysis and prevention.

5. Conclusion

In this study, a novel neural network model, named ConTrans, is established for LCF life prediction from binarized hysteresis images. ConTrans model consists of three components: (1) a CNN as the Backbone, (2) a Transformer as the Neck, and (3) a FCNN as the Head. The model leverages the local information processing ability of CNN and the global information fusion ability of Transformer to extract and learn mechanical information such as stress-strain relationships and energy information from hysteresis images. The model is validated by four materials (MarBN steel, MAR-M247, AM 304, and 2024AA) under thermal-mechanical loading. The results demonstrate that the ConTrans model has excellent LCF life prediction capability, and can be adjusted according to the specific case under study to achieve higher accuracy. Moreover, the quantity and quality of the fatigue data used are also important factors that affect the model performance.

6. Outlook

The proposed ConTrans model is a novel and effective method for predicting the LCF life from the half-life hysteresis loop image, without requiring any additional inputs such as loading conditions and damage factors. The model has been validated on different structures of materials with thermal-mechanical loading and has shown high accuracy and robustness. It is an attempt to introduce the advanced Transformer model into the field of fatigue life prediction for the first time, and at the same time combines the knowledge in the fields of CV and NLP. However, there are still several challenges and limitations that need to be addressed in the future work. For example, the model performance may depend on the quality and quantity of the training data, and the generalization ability of the model to other materials and loading scenarios needs to be further tested. Moreover, the interpretability of the model is not well understood, and the mechanism behind the image-to-life mapping is still unclear. Therefore, future work will focus on improving the data collection and augmentation, exploring the transfer learning and domain adaptation techniques, and developing the visualization

and analysis tools for the model. These efforts will enhance the applicability and reliability of the ConTrans model for the LCF life prediction of engineering materials.

Another promising direction for the future work is to incorporate physical knowledge into the deep learning models, which can improve the physical consistency and robustness of the predictions. A recent algorithm known as Physics-informed Neural Network (PINN) [55] is a class of algorithms that seamlessly integrates data and abstract mathematical operators. Making learning algorithms physically informed amounts to introducing appropriate observational, learning, or inductive biases that can guide the learning process to identify physically consistent solutions. The mentioned patterns of learning algorithms biasing towards physically consistent solutions are not mutually exclusive, but can effectively be combined to produce a very wide range of hybrid approaches for building physically-informed learning models. Currently, PINN is effectively used in many fields, details of which can be referenced in the reviews [56–59]. In the future, the idea of physics informed can also be borrowed to the field of LCF life prediction under thermal–mechanical loading. This would enable the ConTrans model to leverage the prior knowledge of the material behavior and the fatigue mechanism and to reduce the dependence on the data availability and quality.

CRedit authorship contribution statement

Yang Yang: Writing – original draft, Visualization, Validation, Methodology, Data curation. **Bo Zhang:** Writing – review & editing, Project administration. **Hao Wu:** Investigation. **Yida Zhang:** Data curation. **Hong Zhang:** Writing – review & editing, Project administration. **Yongjie Liu:** Funding acquisition. **Qingyuan Wang:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 12272245, No. 11832007, No.12332012). The authors are sincerely grateful and supported by the fund of the State Key Laboratory of Long-life High-Temperature Materials (No. DECSKL202102).

References

- [1] Tavernelli J, Coffin L. Experimental support for generalized equation predicting low cycle fatigue. *ASME J Basic Eng* 1962.
- [2] Mroziński S. Energy-based method of fatigue damage cumulation. *Int J Fatigue* 2019;121:73–83.
- [3] Mroziński S, Topoliński T. New energy model of fatigue damage accumulation and its verification for 45-steel. *J Theor Appl Mech* 1999;37:223–40.
- [4] Skelton R, Vilhelmsen T, Webster G. Energy criteria and cumulative damage during fatigue crack growth. *Int J Fatigue* 1998;20:641–9.
- [5] Smith K. A stress-strain function for the fatigue of metals. *J Mater* 1970;5:767–78.
- [6] Lee K-O, Hong S-G, Lee S-B. A new energy-based fatigue damage parameter in life prediction of high-temperature structural materials. *Mater Sci Engng A* 2008;496:471–7.
- [7] Zhu H, Wu H, Lu Y, Zhong Z. A novel energy-based equivalent damage parameter for multiaxial fatigue life prediction. *Int J Fatigue* 2019;121:1–8.
- [8] Gan L, Wu H, Zhong Z. Use of an energy-based/critical plane model to assess fatigue life under low-cycle multiaxial cycles. *Fatigue Fract Engng Mater Struct* 2019;42:2694–708.
- [9] Mandegarian S, Taheri-Behrooz F. A general energy based fatigue failure criterion for the carbon epoxy composites. *Compos Struct* 2020;235:111804.
- [10] Gao F, Xie L, Liu T, Song B, Pang S, Wang X. An equivalent strain energy density model for fatigue life prediction under large compressive mean stress. *Int J Fatigue* 2023;177:107899.
- [11] Xu L, Zhang R, Hao M, Xiong L, Jiang Q, Li Z, et al. A data-driven low-cycle fatigue life prediction model for nickel-based superalloys. *Comput Mater Sci* 2023;229:112434.
- [12] Duan H, Cao M, Liu L, Yue S, He H, Zhao Y, et al. Prediction of 316 stainless steel low-cycle fatigue life based on machine learning. *Sci Rep* 2023;13:6753.
- [13] Long X, Lu C, Su Y, Dai Y. Machine learning framework for predicting the low cycle fatigue life of lead-free solders. *Engng Fail Anal* 2023;148:107228.
- [14] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015;521:436–44.
- [15] Zhang X-C, Gong J-G, Xuan F-Z. A deep learning based life prediction method for components under creep, fatigue and creep-fatigue conditions. *Int J Fatigue* 2021;148:106236.
- [16] Yang J, Kang G, Liu Y, Kan Q. A novel method of multiaxial fatigue life prediction based on deep learning. *Int J Fatigue* 2021;151:106356.
- [17] Sun X, Zhou K, Shi S, Song K, Chen X. A new cyclical generative adversarial network based data augmentation method for multiaxial fatigue life prediction. *Int J Fatigue* 2022;162:106996.
- [18] Zhou C, Wang H, Hou S, Han Y. A hybrid physics-based and data-driven method for gear contact fatigue life prediction. *Int J Fatigue* 2023;107763.
- [19] Halamka J, Bartošák M, Španiel M. Using hybrid physics-informed neural networks to predict lifetime under multiaxial fatigue loading. *Engng Fract Mech* 2023;289:109351.
- [20] Kamiyama M, Shimizu K, Akiniwa Y. Prediction of low-cycle fatigue crack development of sputtered Cu thin film using deep convolutional neural network. *Int J Fatigue* 2022;162:106998.
- [21] Heng F, Gao J, Xu R, Yang H, Cheng Q, Liu Y. Multiaxial fatigue life prediction for various metallic materials based on the hybrid CNN-LSTM neural network. *Fatigue Fract Engng Mater Struct* 2023;46:1979–96.
- [22] Sun X, Zhou T, Song K, Chen X. An image recognition based multiaxial low-cycle fatigue life prediction method with CNN model. *Int J Fatigue* 2022.

- [23] Yang J, Kang G, Kan Q. A novel deep learning approach of multiaxial fatigue life-prediction with a self-attention mechanism characterizing the effects of loading history and varying temperature. *Int J Fatigue* 2022;162:106851.
- [24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Adv Neural Inf Proces Syst* 2017;30.
- [25] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2015:3431–40.
- [26] Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training, 2018.
- [27] Ramachandran P, Zoph B, Le QV. Searching for activation functions, arXiv preprint arXiv:1710.05941, 2017.
- [28] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision* 2015:1026–34.
- [29] Lin M, Chen Q, Yan S. Network in network, arXiv preprint arXiv:1312.4400, 2013.
- [30] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473, 2014.
- [31] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016. p. 770–8.
- [32] Ba JL, Kiros JR, Hinton GE. Layer normalization, arXiv preprint arXiv:1607.06450, 2016.
- [33] Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model. *Adv Neural Inf Proces Syst* 2000;13.
- [34] Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw* 2015;61:85–117.
- [35] Ramberg W, Osgood WR. Description of stress-strain curves by three parameters, in, 1943.
- [36] Chen X, Xu S, Huang D. A critical plane-strain energy density criterion for multiaxial low-cycle fatigue life under non-proportional loading. *Fatigue Fract Engng Mater Struct* 1999;22:679–86.
- [37] Zhou K, Sun X, Shi S, Song K, Chen X. Machine learning-based genetic feature identification and fatigue life prediction. *Fatigue Fract Engng Mater Struct* 2021;44:2524–37.
- [38] Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT press 2016.
- [39] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks, in: *Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, 2010, pp. 249–256.
- [40] Liu L, Jiang H, He P, Chen W, Liu X, Gao J, Han J. On the variance of the adaptive learning rate and beyond, arXiv preprint arXiv:1908.03265, 2019.
- [41] Smith LN. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay, arXiv preprint arXiv:1803.09820, 2018.
- [42] Xu Z, Dai AM, Kemp J, Metz L. Learning an adaptive learning rate schedule, arXiv preprint arXiv:1909.09712, 2019.
- [43] Kingma DP, Ba J. Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.
- [44] Xu B, Wang N, Chen T, Li M. Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853, 2015.
- [45] Hendrycks D, Gimpel K. Gaussian error linear units (gelus), arXiv preprint arXiv:1606.08415, 2016.
- [46] Zhang H, Wang Q, Gong X, Wang T, Pei Y, Zhang W, et al. Comparisons of low cycle fatigue response, damage mechanism, and life prediction of MarBN steel under stress and strain-controlled modes. *Int J Fatigue* 2021;149:106291.
- [47] Wang Q, Liu M, Zou T, Jiang Y, Gao Z, Pei Y, et al. The cyclic deformation behavior and microstructural evolution of 304L steel manufactured by selective laser melting under various temperatures. *Mater Sci Engng A* 2024;891:145949.
- [48] Falkowska A, Tomczyk A, Seweryn A. The effect of elevated temperature on LCF damage growth in 2024AA–experiment and modeling. *Engng Fail Anal* 2024;158:108015.
- [49] Oh B. The exponential law of endurance tests, in: *Proc Am Soc Test Mater* 1910:625–30.
- [50] Delprete C, Rosso C. Residual life estimation under low-cycle and thermo-mechanical fatigue conditions: proposal of a dedicated numerical code, in: *Engineering Systems Design and Analysis*. American Society of Mechanical Engineers; 2014.
- [51] Delprete C, Sesana R. Proposal of a new low-cycle fatigue life model for cast iron with room temperature calibration involving mean stress and high-temperature effects. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 2019;233:5056–73.
- [52] Hai L, Ban H, Yang X, Shi Y. Low-cycle fatigue behaviour of hot-rolled titanium-clad bimetallic steel. *Int J Mech Sci* 2023;254:108443.
- [53] Zou T, Liu M, Wang Q, Jiang Y, Wu H, Zhang H, et al. New approach to low-cycle fatigue lifetime prediction for deep-rectangular notched components with finite residual thickness: Experiment and simulation. *Int J Fatigue* 2024;108380.
- [54] Kuroda M. Extremely low cycle fatigue life prediction based on a new cumulative fatigue damage model. *Int J Fatigue* 2002;24:699–703.
- [55] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 2019;378:686–707.
- [56] Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J Sci Comput* 2022;92:88.
- [57] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. *Nature Reviews Physics* 2021;3:422–40.
- [58] Von Rueden L, Mayer S, Beckh K, Georgiev B, Giesselbach S, Heese R, et al. Informed machine learning—a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Trans Knowl Data Engng* 2021;35:614–33.
- [59] Willard J, Jia X, Xu S, Steinbach M, Kumar V. Integrating physics-based modeling with machine learning: a survey, arXiv preprint arXiv:2003.04919, 1 (2020) 1–34.