*Advances in Technology Innovation, vol. x, no. x, 20xx, pp. xx-xx*

# Remaining Useful Life Prediction of Milling Tool Based on Improved PSO-MultiAM-BiLSTM

Xiaomei Ni[1], David Chua Sing Ngie[1,*], Wanzhen Wang[2], Miaomiao Xin[2], Qiu Man[2], Liangyu Tian[2], Jingzhe Sun[2]

[1]Faculty of Engineering, Universiti Malaysia Sarawak (UNIMAS), Sarawak, Malaysia

[2]School of Intelligent Manufacturing and Control Engineering, Qilu Institute of Technology, Shandong, China

Received 22 May 2025; received in revised form 11 August 2025; accepted 18 August 2025

DOI: https://doi.org/10.46604/aiti.2025.15175

## Abstract

To improve the accuracy of remaining useful life (RUL) prediction for milling tools, this study proposes an enhanced PSO-MultiAM-BiLSTM model integrating particle swarm optimization (PSO), multi-head attention mechanism (MultiAM), and bidirectional long short-term memory (BiLSTM). The model captures key information in input sequences, alleviating early feature attenuation in BiLSTM from "chain propagation." A logarithmic decreasing strategy adjusts PSO inertia weights, balancing global and local searches while optimizing BiLSTM parameters. Validated on the PHM2010 dataset, the model attains an average coefficient of determination of 0.97, with average root-mean-square error and mean absolute error of 0.062 and 0.045, improving prediction accuracy by 9.64% and 4.06% over MultiAM-BiLSTM and PSO-AM-BiLSTM, respectively. Such a result attests to the effective extraction of degradation features of tools and provides a valuable reference for predicting the RUL of milling tools.

**Keywords:** MultiAM, PSO algorithm, BiLSTM, RUL

## 1. Introduction

As a fundamental tool of industry, the computer numerical control (CNC) machine tool contributes significantly to industrial manufacturing. With the increasing demand for product quality, the stability of the machining process has become increasingly important. This section examines the research background of remaining useful life (RUL) prediction, relevant literature, research gaps, research objectives, and the structure of this study.

Accurate prediction of tool RUL can reduce downtime due to tool failure, avoid unnecessary tool changes, and extend tool life, which is crucial for improving productivity and saving production costs [1]. Traditional RUL prediction methods mainly include methods based on physical models and traditional machine learning. Physical-model-based methods formulate mathematical models by studying tool failure mechanisms, such as tool wear, whereas they face challenges in adapting to complex and changing production environments and in maintaining low prediction accuracy [2-3]. Traditional machine learning models are mainly trained by degraded data, such as the hidden Markov model, the Wiener process, and the Bayesian model. However, these methods suffer from the inability to capture nonlinear relationships and difficulty in maintaining the accuracy of prediction results [4-5].

With the rapid development of deep learning and neural networks, RUL prediction based on deep learning models is gradually gaining prominence. By integrating multi-sensor data, feature extraction techniques, and advanced deep learning methods, researchers can effectively predict RUL [6]. Convolutional neural network (CNN) demonstrates proficiency in

* Corresponding author. E-mail address: 21010377@siswa.unimas.my

automatically extracting spatial data features, thereby augmenting RUL prediction accuracy [7-9]. Recurrent neural network (RNN) and its variants, such as long short-term memory (LSTM) and bidirectional long short-term memory (BiLSTM), are pervasively employed in natural language processing owing to their ability to process temporal data, and have recently been applied to RUL prediction as well [10].

Wang et al. [11] utilized a stacked self-encoder (SAE) with LSTM to enhance model generalization under complex conditions. Similarly, Wang et al. [12] proposed a CNN-LSTM-PSO method, which incorporates a CNN, LSTM, and particle swarm optimization (PSO). This method utilizes CNNs to extract local features, LSTM to process temporal information, and PSO to optimize hyperparameters, thereby improving prediction performance. Yao et al. [13] proposed an LSTM-based deep transfer reinforcement learning network with strong cross-tool and cutting condition generalization. Liu et al. [2] used a parallel residual network (PResNet) and a stacked BiLSTM to improve the accuracy of the tool RUL.

As a popular metaheuristic, PSO optimizes models like backpropagation (BP), LSTM, least-squares support vector machine (LSSVM), and radial basis function (RBF) for RUL prediction. Research has shown that both the BP neural network and RBF neural network optimized by the PSO algorithm can more accurately capture data features [14]. Some researchers also used the PSO algorithm to optimize the parameters of the CNN-LSTM model. The experimental results showed that the method yields high accuracy and stability in tool wear prediction [12]. Some studies have applied the PSO algorithm to the optimization of the LSSVM. The experiments show that the optimized model outperforms the unoptimized model in terms of prediction performance [15-16].

In recent years, the application of attention mechanisms (AM) in RUL prediction has received much attention [17]. Wang and Zhang [18] proposed a sequence-to-sequence model combining the AM and monotonicity loss, which utilized the AM to enhance the model's focus on key time-series features and introduced a monotonicity loss function to enhance the physical consistency of the prediction results. Dong and Zhao [19] proposed a method for tool wear prediction that combines an enhanced AutoEncoder (AE) with a multi-head attention mechanism (MultiAM), which utilizes the AE for feature extraction and focuses on essential timing features in the input data through a MultiAM to improve the prediction accuracy.

While LSTM and its variants have advantages in time-series processing, and researchers have explored deep learning models, PSO-optimized models, and AMs to improve tool RUL prediction, shortcomings remain in addressing the key challenges of current frameworks. The specific research gaps are as follows:

(1) BiLSTM exhibits attenuation of early wear features as a consequence of the "chain propagation" mechanism. Current studies lack a dynamic weighting mechanism designed for the stage-specificity of RUL prediction, hindering the strengthening of key time-period features and thus restricting the accuracy and applicability of tool's RUL prediction.
(2) As a widely used metaheuristic algorithm, PSO currently exhibits deficiencies in optimizing model hyperparameters and managing critical details of the training process. This shortcoming directly leads to the inability to fully leverage the inherent performance potential of target models.
(3) The initial weights of the current PSO algorithm are generally optimized using linear diminishing optimization. This optimization approach causes the weights to drop rapidly in the early stage of the particle search, incurring the premature convergence of particles to a local optimal solution instead of finding the global optimal solution.

Based on the above analysis of the current tool RUL prediction limitations, this paper proposes a RUL prediction method built on a fusion approach. This method aims to address identified gaps, and the main objectives of the study are as follows:

(1) The proposed model introduces the MultiAM and enables parallel computing and learning different subspaces. It can dynamically capture the key information in the input sequence, effectively addressing the issue of information attenuation in long sequences and overcoming the limitations of the traditional "chain propagation" approach.

(2) When optimizing the model using the PSO algorithm, in addition to optimizing the model learning rate, the number of hidden layer units and the initial weights of the model are also optimized. This enables more comprehensive and precise optimization of key parameters that impact BiLSTM network performance.

(3) This model dynamically adjusts the inertia weight using a nonlinear function, thereby achieving a better balance of the global search capability and local search capability of the PSO algorithm. This adjustment optimizes the initial weights of the algorithm and enhances its ability to locate the global optimal solution during the parameter optimization process.

The subsequent structure of this article is as follows: In Section 2, relevant research methodologies are introduced, including fundamental principles of BiLSTM, PSO, and MultiAM. In Section 3, the improved PSO-MultiAM-BiLSTM model is presented in detail. In Section 4, results and analysis of RUL prediction are provided, encompassing the introduction of the PHM2010 dataset, feature extraction methods, experimental setups, and a comprehensive discussion of model performance. Finally, in Section 5, the conclusion synthesizes key findings of this study, underscores the advantages of the proposed model, and outlines potential directions for future research.

## 2. Methodology

This section introduces the architecture framework of the BiLSTM neural network and the MultiAM, focusing on the core components of these systems and their basic operation mechanisms. Additionally, the optimization workflow of the PSO algorithm is also presented.

### *2.1. BiLSTM neural networks*

LSTM can effectively capture long-range sequential dependencies and is widely used in tool life prediction. The key components of LSTM entail the cell state and several gating mechanisms, including the forget gate, input gate, and output gate. These gating mechanisms employ a sigmoid function to regulate the flow of information. The specific parameter update calculation steps are as follows:

(1) Calculate the forget gate $f_t$ and select the information to be forgotten:

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

$$i_t = \sigma\left(W_i \times \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

(2) Update the unit status:

$$\tilde{C}_t = \tanh\left(W_C \times \left[h_{t-1}, x_t\right] + b_C\right) \tag{3}$$

(3) Calculate the value of the candidate memory cell at the current moment $C_t$:

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{4}$$

(4) Compute the output gate $o_t$ and the hidden layer state at the current moment $h_t$:

$$o_t = \sigma\left(W_o \times \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t \times \tanh\left(C_t\right) \tag{6}$$

where $W_f$ and $W_i$ are the corresponding connection weight matrices; $b_f$, $b_i$ and $b_C$ are the bias vectors; $x_t$ is the input data at the current moment; $h_{t-1}$ is the hidden layer value at the previous moment; $\sigma$ is the sigmoid activation function and the value interval is [0,1]; $C_{t-1}$ is the last moment cell state; $C_t$ is the current cell state; and $\tilde{C}_t$ is the temporary cell candidate value.

Despite its ability to address the long-term dependency problem, LSTM utilizes only the past information of time-series data while ignoring future information. The BiLSTM extends the capability of LSTM, which can handle multivariate time-series data by simultaneously considering both forward and backward information of the series, along with the effects of multiple influencing factors on the prediction results. As such, it is suitable for multivariate time-series data in the tool cutting process, and its structure is shown in Fig. 1.

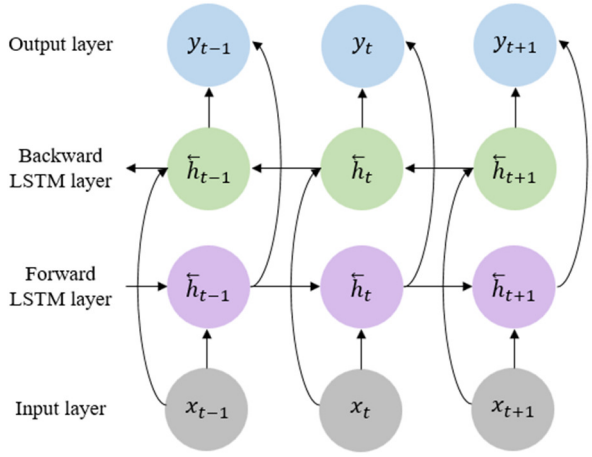


Fig. 1 Structure of BiLSTM

### 2.2. *PSO algorithm*

The PSO algorithm is a swarm intelligence-based optimization algorithm that finds more accurate results through the iterative evolution of the current solution [15]. PSO can expeditiously converge and find high-quality solutions in a shorter period. Therefore, it is widely used in the fields of hyperparameter optimization, feature selection, and neural network training. The specific steps in the process of the PSO algorithm search are as follows:

(1) Initialize the PSO algorithm, including the population size $N$, the end time, and other parameters. Subsequently, determine the velocity $v_i$ and position $x_i$ of each particle $i$ based on these factors:

$$v_i = \left(v_{i1}, v_{i2}, \ldots, v_{iD}\right),\ i = 1, 2, \ldots, N \tag{7}$$

$$x_i = \left(x_{i1}, x_{i2}, \ldots, x_{iD}\right),\ i = 1, 2, \ldots, N \tag{8}$$

(2) Evaluate the fitness of each particle using the objective function F[$i$];

(3) Update the individual optimal solution and the global optimal solution according to the following equations:

$$p_{best} = \left(p_{i1}, p_{i2}, \ldots, p_{iD}\right),\ i = 1, 2, \ldots, N \tag{9}$$

$$g_{best} = \left(g_{i1}, g_{i2}, \ldots, g_{iD}\right),\ i = 1, 2, \ldots, N \tag{10}$$

where $p_{best}$ is the individual optimal solution, indicating the initial position of each particle; $g_{best}$ is the global optimal solution, signifying the particle with the best initial fitness value.

(4) In updating the velocity and position of the particles, the following calculations are performed:

$$v_{id} = wv_{(i-1)d} + c_1 r_1 \left(p_{(i-1)d}\right) + c_2 r_2 \left(p_{gd} - x_{(i-1)d}\right) \tag{11}$$

$$x_{id} = x_{(i-1)d} + av_{(i-1)d} \tag{12}$$

where $w$ is the inertia factor; $c_1$ and $c_2$ are the acceleration coefficients; $r_1$ and $r_2$ are mutually independent stochastic functions between [0,1]; $\alpha$ is the constraint factor used to limit the speed.

(5) Loop the steps (2) to (4) until the optimization termination condition is satisfied, ending the optimization process.

### *2.3. Multiple attention mechanisms (AM)*

In recent years, AMs have become a much-anticipated research topic in the field of deep learning. Traditional AMs tend to rely on another relevant sequence to determine the allocation of attention when processing an input sequence. In this regard, the MultiAM introduces multiple independent attention computations to enhance the expressive power of the model. The core idea of the MultiAM is to map the input data into multiple low-dimensional subspaces and compute the attention independently in each subspace. Finally, these attention results are spliced and fused. The core idea of the MultiAM is shown in Fig. 2. Unlike the single-head AM, the MultiAM divides the $Q$ (query vector), $K$ (key vector), and $V$ (value vector) into $h$ parts (multiple heads) and computes the attention independently in these subspaces. The calculation formula is as follows:

$$MultiHead\left(Q,K,V\right)=Concat\left(head_1,head_2,\ldots,head_h\right)W_O \tag{13}$$

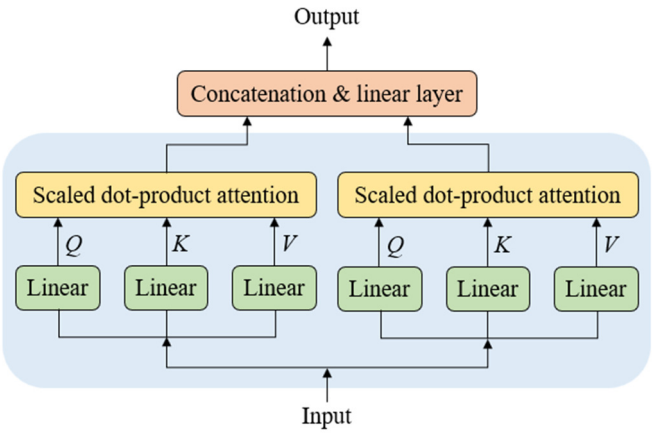


Fig. 2 Core idea of the MultiAM

### *2.4. Improved PSO-MultiAM-BiLSTM model*

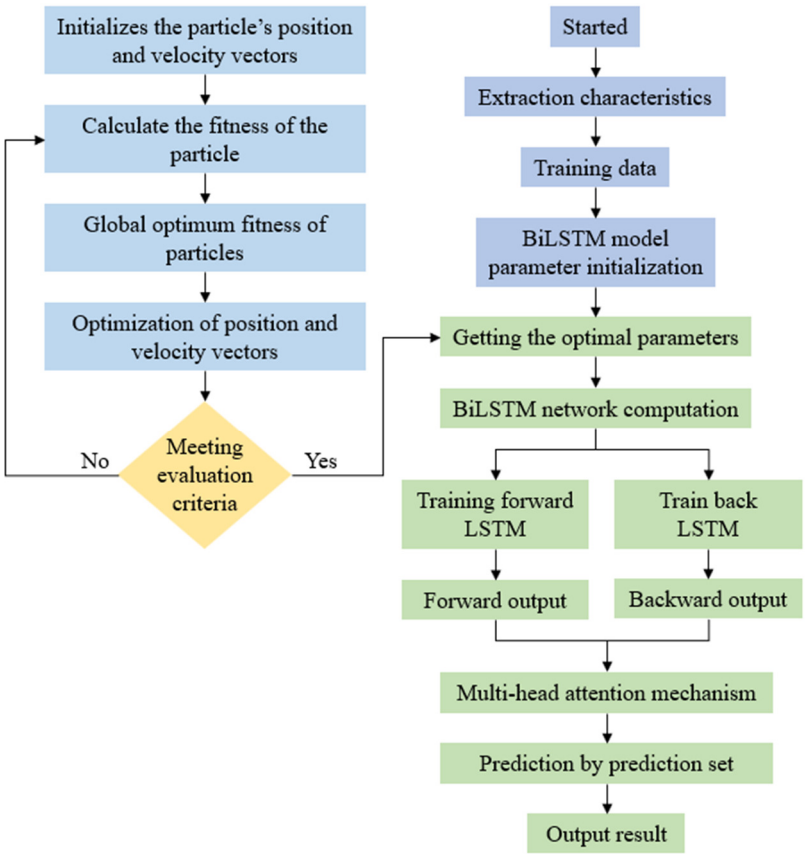


Fig. 3 PSO-MultiAM- BiLSTM training flowchart

Based on the aforementioned theory, this research proposes a RUL prediction model based on the improved PSO-MultiAM-BiLSTM. The inertia weights of the PSO algorithm in this model are dynamically adjusted through a nonlinear function, and the PSO algorithm optimizes key parameters of the BiLSTM network, such as the number of hidden layers and the learning rate. The correlation weights between features are computed using a MultiAM that enables the BiLSTM model to dynamically focus on the important information in the input sequence. The process of model realization is shown in Fig. 3.

The specific steps in the RUL prediction process of the proposed PSO-MultiAM-BiLSTM are as follows:

(1) When the BiLSTM network executes the BP algorithm for gradient computation, the output value of each neuron is derived by concatenating the hidden states from forward and backward computations. A MultiAM is incorporated to enhance label localization and classification, enabling the recognition of discrepancies between different monitoring data and facilitating the extraction of deeper potential information from the data. To mitigate overfitting, a dropout rate of 0.25 is employed. Finally, the data is passed through two fully connected layers with 10 and 1 units, respectively, before being fed into the regression layer.

(2) The PSO algorithm is initialized by adopting the initial root-mean-square error (RMSE) of each particle as the local optimum. The initial parameters of the model are configured as listed in Table 1, including the initial learning rate, the number of hidden layer units, and initial weights. These parameter settings are determined based on extensive trial experiments.

Table 1 PSO-MultiAM-BiLSTM model parameters

| Network parameter | Initial value |
|---|---|
| Initialization weights | [0.4,1] |
| Oblivion gate bias | 1 |
| Input gate bias (progression, pacemaker = 0.1) | [0,0.8] |
| Output gate bias (progression, pacemaker = 0.1) | [0,0.8] |
| Number of iterations | 200 |
| Population size | 10 |
| Learning factor $c1$ | 1.5 |
| Learning factor $c2$ | 1.5 |
| Number of hidden layer units m (progression, pacemaker = 10) | [10,300] |
| Learning rate r | [0.01,0.15] |

(3) In the PSO optimization process, a logarithmic decreasing strategy is introduced to dynamically adjust the inertia weights. This strategy gradually reduces the inertia weights during the iterative process, effectively balancing the global exploration and local search capabilities of the algorithm and improving its optimization performance. The formula for the logarithmic decreasing inertia weights is given in the formula below. The RMSE is employed as the fitness function, and the optimal learning rate and number of hidden layers of the BiLSTM model are determined through iterative search, attaining efficient optimization of model parameters. Additionally, the Adam optimizer is utilized to optimize the model weights, ensuring expeditious convergence during training and further enhancing the training efficiency and prediction performance of the model. The specific formula is as follows:

$$\omega(t) = \omega_{\min} + \left(\omega_{\max} - \omega_{\min}\right) \times \frac{\ln\left(T_{\max} - t + 1\right)}{\ln\left(T_{\max}\right)} \tag{14}$$

where $\omega(t)$ is the inertia weight at iteration $t$ times; $\omega_{\min}$ is the minimum value of the inertia weights, which is reached in the later stages of the iteration and enhances the local search capability of the algorithm; $\omega_{\max}$ is the maximum value of the inertia weights, ensuring strong global exploration capability at the beginning of the iteration; $T_{\max}$ is the maximum number of iterations of the algorithm; and $t$ is the current iteration number.

## 3. Dataset and Feature Extraction

This section introduces the source of the dataset, experimental conditions, and data preprocessing methods. Regarding the experimental conditions, key details include the equipment used and milling parameters. In terms of data preprocessing, it mainly covers original signal processing and feature extraction, including calculation formulas of time-domain features, frequency-domain features, and time-frequency domain features.

### *3.1. Introduction of experimental data*

The dataset from the PHM2010 Data Challenge was used for this experimental validation [1]. The RFM760 CNC machine tool was selected as the experimental platform, with a three-flute tungsten carbide ball-end milling cutter as the cutting tool, to cut the stainless steel (RC52) material. The specific experimental equipment and milling parameters are shown in Table 2.

Table 2 Experimental equipment and milling parameters

| Equipment type | Experimental equipment | Milling parameters | Parameterization |
|---|---|---|---|
| Machine type | Roders RFM760 | Spindle speed | 10,400 r/min |
| Cutters | Ball tungsten carbide milling cutter | Feed rate | 1,555 mm/min |
| Wear measurement equipment | LEICA MZ12 microscope | Amount of tools per travel | 0.001 mm |
| Workpiece materials | Stainless steels HRC52 | Cutting width | 0.125 mm |
| Data acquisition equipment | NI DAQ data acquisition card | Depth of cut | 0.2 mm |
| Boosters | Kistler charge amplifier | Sampling frequency | 50 kHz |
| Force sensor | Three-way force gauge | Milling method | Milling with no problem |
| Vibration sensors | Triaxial acceleration sensor | Cooling method | Dry cut |

To comprehensively monitor the state of the milling cutter in the machining process, a variety of equipment, such as force-measuring instruments, acceleration sensors, and acoustic emission sensors, was used to collect the original time-domain signals, such as cutting force, vibration, and acoustic emission. These signals were sampled at a frequency of 50 kHz to ensure the accuracy and integrity of the data. In the experiment, each tool traveled 108 mm in the *X*-direction, and each tool traveled 315 times to fully collect the cutting monitoring signals under different conditions [15]. Through an in-depth analysis of these signals, the dynamic characteristics of the milling machining process can be better understood, and effective data can be provided for the subsequent training and optimization of the neural network model.

### *3.2. Data preprocessing*

To allow the model to extract more accurate and effective information, the original signal requires the following preprocessing steps. First, the intense vibration signal generated by the tool cutting is truncated. Subsequently, the original signal is subjected to wavelet threshold processing for noise reduction. Feature extraction plays an important role in data preprocessing. To obtain more comprehensive information about the remaining service life of the tool, 30 features from the time domain, frequency domain, and time-frequency domain were extracted from the raw signal data of each channel. After completing the feature engineering work, the original datasets C1, C4, and C6 correspond to a feature space of size ($315 \times 6 \times 30$).

(1) Time domain features

In the time domain, a total of 18 features, including the absolute mean, maximum value, root mean square, root amplitude, skewness, kurtosis, shape factor, pulse factor, Teager energy, skewness factor, crest factor, clearance coefficient, activity, energy entropy, fluidity, information entropy, Willison amplitude, and kurtosis factor, are mainly extracted, and the specific calculation formulas are presented in Table 3.

Table 3 Time domain feature calculation formula

| Feature | Formula | Feature | Formula |
|---|---|---|---|
| Absolute mean | $\frac{1}{N}\sum_{i=1}^{N}\lvert y_i\rvert$ | Skewness factor | $\frac{\frac{1}{n}\sum_{i=1}^{n}(x_1-\bar{x})^3}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_1-\bar{x})^2}}$ |
| Maximum value | $max(x_1, x_2, \cdots, x_n)$ | Crest factor | $\frac{x_{peak}}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert}$ |
| Root mean square | $\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2}$ | Clearance coefficient | $\frac{Max}{MAV}$ |
| Root amplitude | $\sqrt{Amplitude}$ | Activity | $(RMS)^2$ |
| Skewness | $\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^3$ | Energy entropy | $-\sum_{i=1}^{N}p_i In(p_i)$ |
| Kurtosis | $\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu}{\sigma}\right)^4$ | Fluidity | $\frac{\int_0^{\infty} f\times P(f)df}{\int_0^{\infty}P(f)df}$ |
| Shape factor | $\frac{RMS}{MAV}$ | Information entropy | $-\sum_{i=1}^{N}p(x_i)In[p(x_i)]$ |
| Pulse factor | $\frac{x_{peak}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}$ | Willison amplitude | $\frac{1}{K}\sum_{i=1}^{K}\lvert x_{i+1}-x_i\rvert$ |
| Teager energy | $E_i=x_i^2-x_{i-1x_i+1}$ | Kurtosis factor | $\frac{\frac{1}{n}\sum_{i=1}^{n}(x_1-\bar{x})^4}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_1-\bar{x})^2}}-3$ |

(2) Frequency domain features

The four features of center of gravity frequency (FC), mean square frequency (MSF), root mean square frequency (RMSF), and frequency variance (VF) were extracted mainly by fast Fourier transform (FFT) in the frequency domain, and the specific calculation formula is shown in Table 4.

Table 4 Frequency domain feature calculation formula

| Feature | Formula |
|---|---|
| FC | $\frac{\int_0^{+\infty} fS(f)df}{\int_0^{+\infty}S(f)df}$ |
| MSF | $\frac{\int_0^{+\infty} f^2S(f)df}{\int_0^{+\infty}S(f)df}$ |
| RMSF | $\sqrt{\frac{\int_0^{+\infty} f^2S(f)df}{\int_0^{+\infty}S(f)df}}$ |
| VF | $\frac{\int_0^{+\infty}(f-FC)^2S(f)df}{\int_0^{+\infty}S(f)df}$ |

(3) Time and frequency domain features

The db3 (Dobesi limit phase) wavelet transform is mainly used in the time-frequency domain to extract features and analyze the data of both low and high frequency parts, and the wavelet tree depth used is 3. The wavelet packet transform is reconstructed to analyze the features of different frequency bands. The eight node coefficients are parametrised as the eight features extracted in the time-frequency domain.

## 4. Results and Discussion

In this study, the sample set was divided into three sets of experiments, and two of the three tool labeling samples were used for training and the other for testing in each set of experiments. The detailed setup of the experimental groups is shown in Table 5.

Table 5 Training and test set setup

| Experiment No. | Training set | Test set |
|---|---|---|
| E1 | C1, C4 | C6 |
| E2 | C1, C6 | C4 |
| E3 | C4, C6 | C1 |

The particle fitness convergence curves of the presented model across the E1, E2, and E3 experimental groups are presented in Figs. 4-6. As observed in the figures, the fitness value stabilizes after an average of 23 iterations. The optimal learning rates and numbers of hidden layer units for the models corresponding to the three datasets at this stage are listed in Table 6.

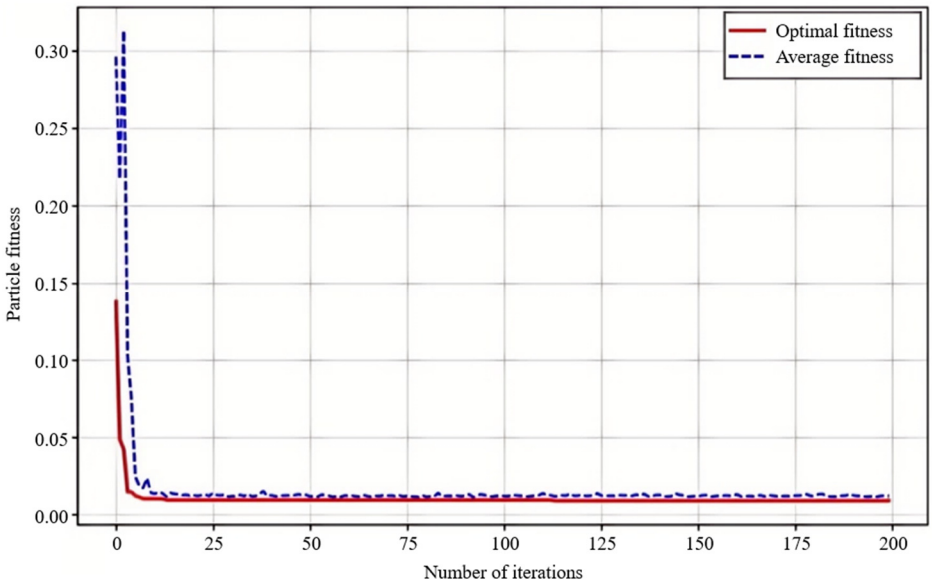


Fig. 4 PSO particle fitness curves for group E1

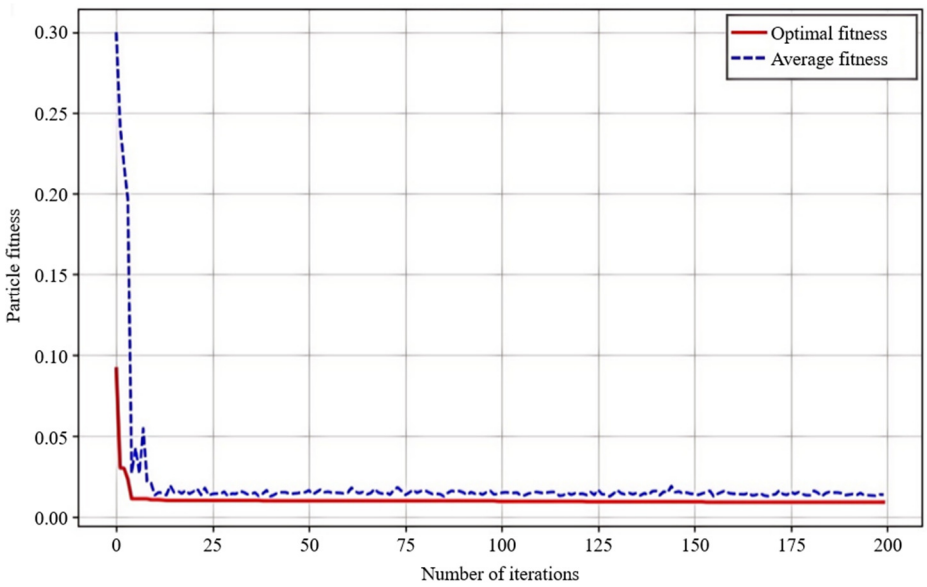


Fig. 5 PSO particle fitness curves for group E2

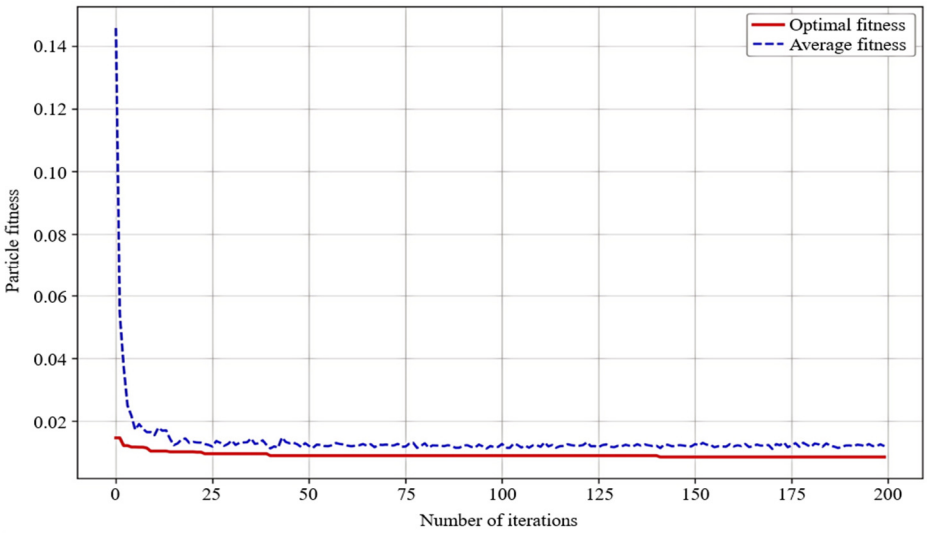


Fig. 6 PSO particle fitness curves for group E3

Table 6 Optimal parameters of the model with different datasets

| Experiment No. | E1 | E2 | E3 |
|---|---|---|---|
| Optimal number of hidden layer units | 18 | 20 | 20 |
| Optimal learning rate | 0.013 | 0.015 | 0.02 |
| Optimal weight | 0.46 | 0.51 | 0.42 |

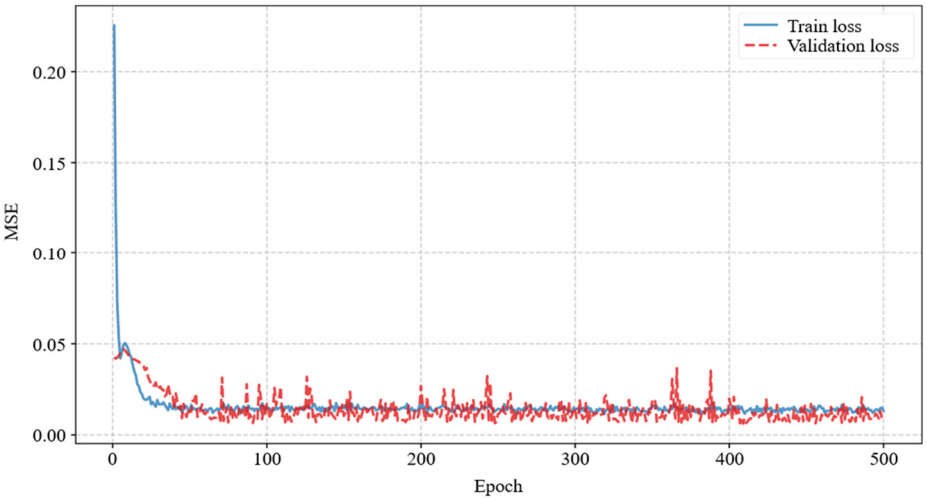


Fig. 7 Train-Validation Loss of E1

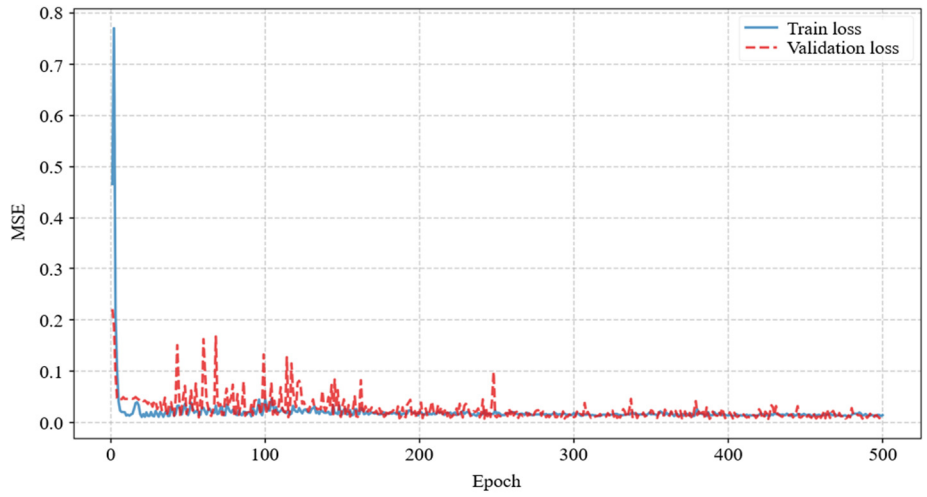


Fig. 8 Train-Validation Loss of E2

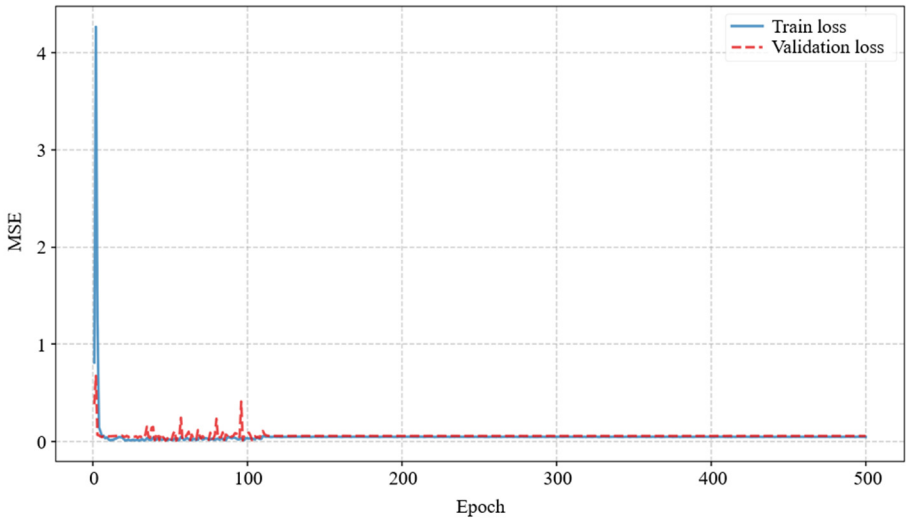


Fig. 9 Train-Validation Loss of E3

The obtained train_loss and val_loss are shown in Figs. 7-9. As can be observed from the figures, with the increase in the number of training epochs, the value of the loss function continuously decreases until convergence, which indicates that the model can effectively learn the extracted signal data features.

The specific values of the regression metrics calculated by different algorithms are shown in Table 7. Concerning performance, the proposed method demonstrates significant advantages in predicting the RUL of milling cutters. Its average RMSE is 0.062, and the mean absolute error (MAE) is 0.045, both of which are lower than those of the other three models. The coefficient of determination ($R^2$) of the proposed method is 0.97, and its prediction accuracy is also higher than that of the other three models, indicating that this model has strong capabilities in data fitting and feature capturing.

Table 7 Comparison of evaluation indicators of different model prediction results

| Model | E1 | | | E2 | | | E3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | RMSE | MAE | $R^2$ | RMSE | MAE | $R^2$ | RMSE | MAE | $R^2$ |
| BiLSTM | 0.098 | 0.073 | 0.83 | 0.091 | 0.079 | 0.83 | 0.093 | 0.074 | 0.83 |
| MultiAM-BiLSTM | 0.073 | 0.057 | 0.88 | 0.071 | 0.057 | 0.89 | 0.072 | 0.059 | 0.89 |
| PSO-AM-BiLSTM | 0.067 | 0.047 | 0.946 | 0.065 | 0.049 | 0.93 | 0.065 | 0.051 | 0.93 |
| PSO-MultiAM-BiLSTM | 0.063 | 0.047 | 0.97 | 0.062 | 0.045 | 0.97 | 0.061 | 0.043 | 0.98 |

However, as shown in Table 8, the improved performance is accompanied by an increase in training costs. The total training time of the PSO-MultiAM-BiLSTM model is 57.6 minutes, which is significantly longer than that of the BiLSTM model (12.3 minutes). Nevertheless, in manufacturing scenarios with strict requirements for prediction accuracy, its accurate predictions can effectively prevent production failures caused by abnormal tool wear. The long-term value greatly outweighs the investment in short-term training time, yielding a more reliable technical solution for tool health management in intelligent production lines.

Table 8 Comparison of the computational cost of different models

| Model | Total training time |
|---|---|
| BiLSTM | 12.3 min |
| MultiAM-BiLSTM | 45.6 min |
| PSO-AM-BiLSTM | 50.5 min |
| PSO-MultiAM-BiLSTM | 57.6 min |

To compare the effect of each model, Figs. 10-12 show the prediction results of the PSO-MultiAM-BiLSTM, PSO-BiLSTM, and BiLSTM models for groups E1, E2, and E3. In the data processing session, a linear transformation is applied to the raw data of the RUL to unify the range of values to the interval [0, 1]. After normalizing the RUL of the tool on

the *y*-axis, the value of 0 represents that the tool life is exhausted, the value of 1 represents that the tool is in a new state, and the intermediate values reflect that the tool is in different degrees of use. The normalization process eliminates the influence of the original data due to the differences in the scale and range of values, and provides a more intuitive picture of the variation of the remaining tool life with the number of cuts. In this graph, the *x*-axis represents the number of cuts, and the y-axis represents the corresponding normalized value of the RUL.

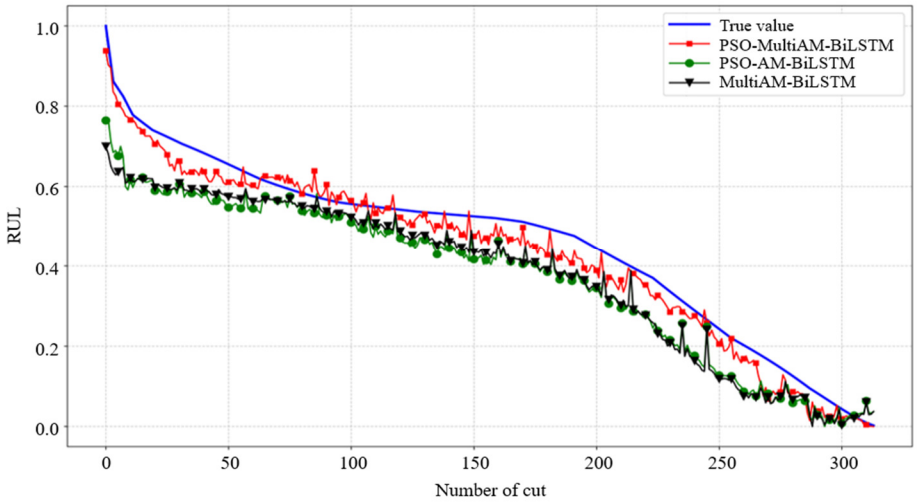


Fig. 10 Prediction effect on group E1

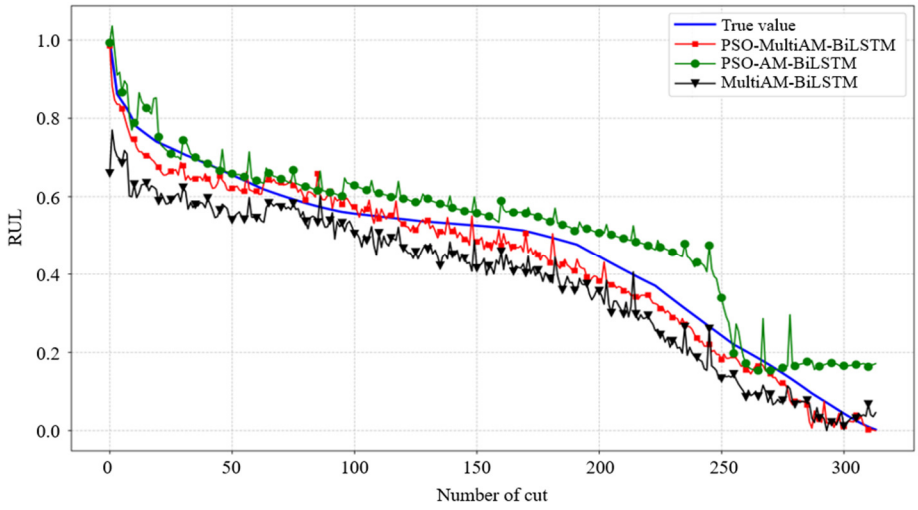


Fig. 11 Prediction effect on group E2

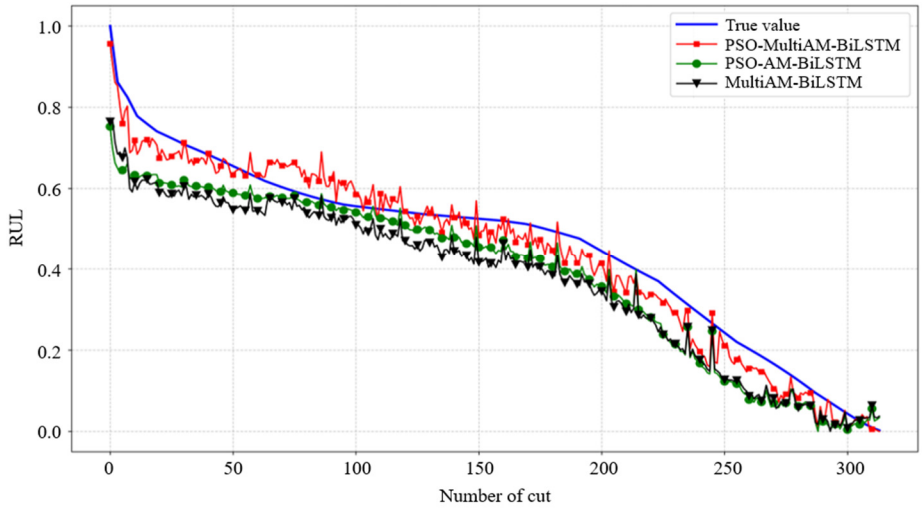


Fig. 12 Prediction effect on group E3

As depicted in Figs. 10-12, due to the lack of PSO hyperparameter optimization, the MultiAM-BiLSTM model is not well adapted to the data distribution, and its prediction accuracy and stability are inferior to those of the model with PSO optimization. When comparing the PSO-MultiAM-BiLSTM model with the PSO-AM-BiLSTM model, the prediction curve of the PSO-AM-BiLSTM model fluctuates significantly, particularly within the 50-200 cutting interval, indicating that the basic AM is less effective than the MultiAM in extracting features under complex working conditions. Its prediction stability is also slightly worse. In other words, the life curve of the PSO-Multiam-bilSTM model is closer to the real life curve, indicating that this model is more accurate in predicting the remaining life of tool wear and reflects the advantages of PSO in optimizing hyperparameters and the AM in focusing on key features. This demonstrates that the proposed method can more accurately identify the degree of tool degradation.

To delve into the advantages of the deep learning hybrid model, four comparison models, including RNN [20], CNN-BiLSTM [21], and BiGRU [22], are employed for experiments, and the results are shown in Table 9. The findings indicate that the PSO-AM-BiLSTM model performs best among all compared models, yielding the lowest RMSE and MAE values. This validates the superiority of combining PSO, MultiAM, and BiLSTM. PSO optimizes hyperparameters to adapt the model to the data distribution, while MultiAM focuses on key temporal features to jointly enhance prediction performance.

Table 9 Comparative Experiment Results Evaluation

| Model | MAE | RMSE |
|---|---|---|
| RNN [20] | 4.1336 | 4.0227 |
| CNN_BiLSTM [21] | 0.1781 | 0.2943 |
| BiGRU [22] | 0.078 | 0.097 |
| PSO-MultiAM-BiLSTM | 0.049 | 0.0564 |

The PSO module in the network structure is replaced with the genetic algorithm (GA) and differential evolution (DE) for comparison, and the experimental results are illustrated in Figs. 13-15. Among the prediction results of each model, the one optimized by PSO exhibits a higher degree of fit with the actual value curve, indicating the highest prediction accuracy. Models optimized through GA and DE show the most significant deviation from true values when the number of cutting operations ranges between 100 and 200. This suggests that tool wear rate changes are complex at this stage, imposing high requirements on model feature extraction and optimization capabilities. However, PSO demonstrates stronger search ability and is more suitable for handling such complex variations. Thus, the PSO-MultiAM-BiLSTM model outperforms other models in predicting the RUL of milling. It is particularly effective in addressing intricate changes in tool wear during cutting processes, providing robust support for practical applications in manufacturing scenarios.

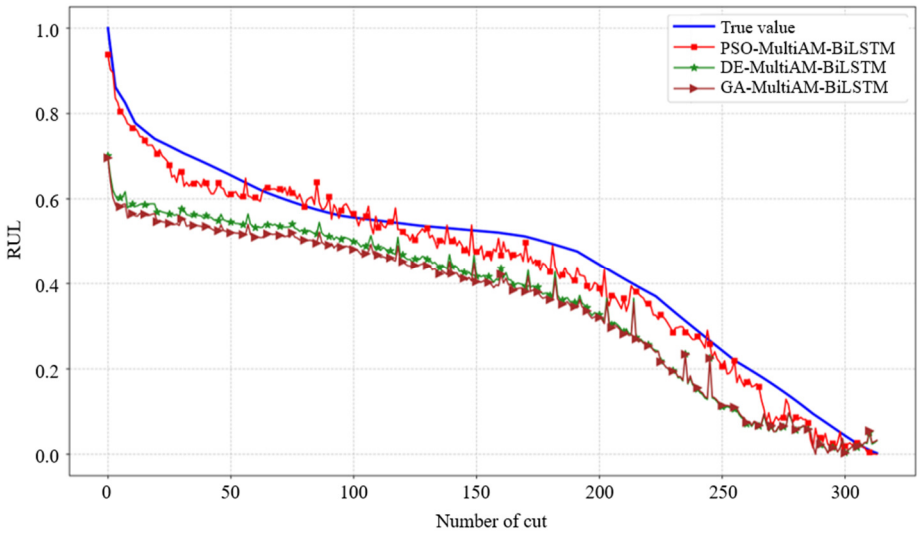


Fig. 13 The predictive effects of different optimization algorithms on the E1 group

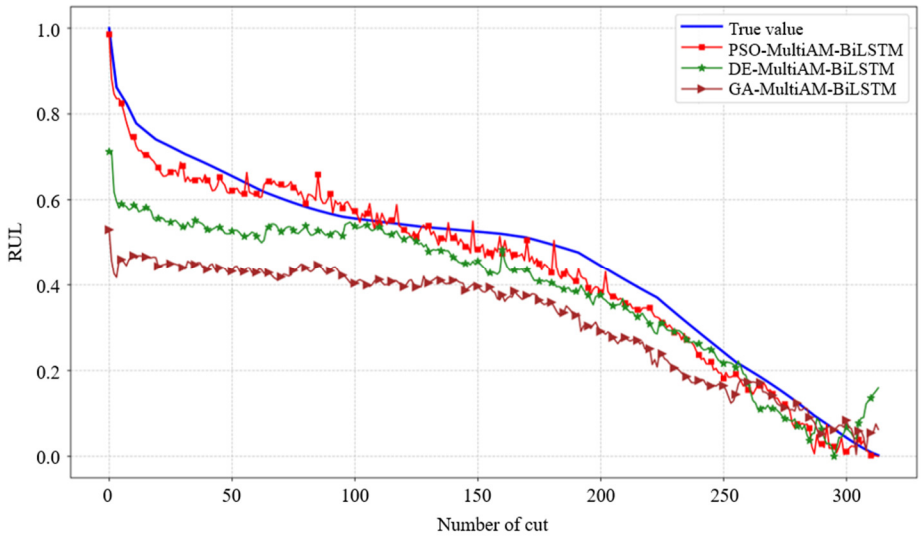


Fig. 14 The predictive effects of different optimization algorithms on the E2 group

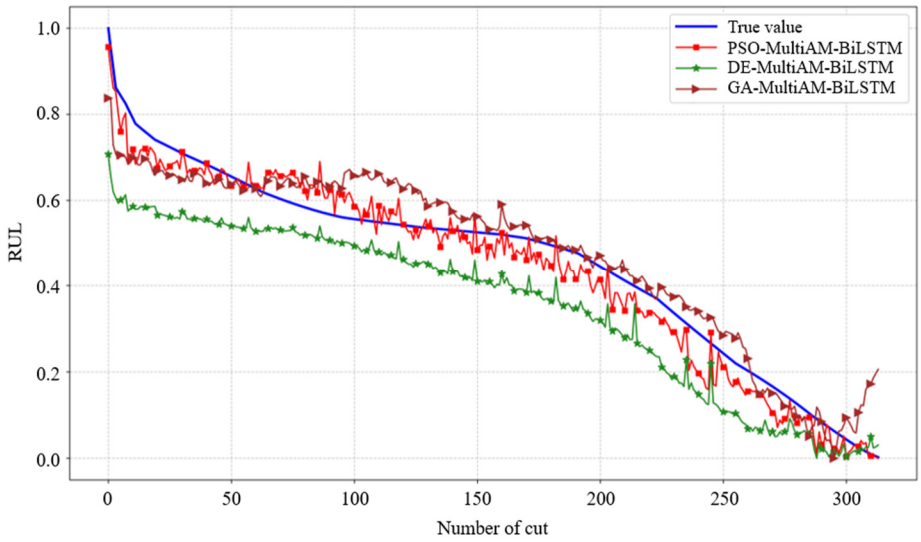


Fig. 15 The predictive effects of different optimization algorithms on the E3 group

## 5. Conclusion

This study proposes an improved PSO-MultiAM-BiLSTM model for accurate RUL prediction of milling tools, addressing limitations in existing deep learning and optimization frameworks. By integrating a MultiAM with BiLSTM, the model dynamically captures key features in long time-series data, mitigating early critical feature attenuation caused by "chain propagation" in traditional BiLSTM. PSO is applied to optimize BiLSTM parameters, including hidden layer units, learning rate, and initial weights, using a logarithmic decreasing strategy for inertia weights, balancing global exploration and local search capabilities, and avoiding premature convergence to local optima. Experimental validation on the PHM2010 dataset demonstrates the superiority of the proposed model in predicting tool RUL. The main findings are summarized as follows:

(1) The proposed PSO-MultiAM-BiLSTM model effectively mitigates early critical feature attenuation in long time-series data and enhances feature discrimination across different wear stages.

(2) PSO optimization of BiLSTM parameters improves model performance by balancing global and local search capabilities, enhancing overall optimization effectiveness.

(3) Experimental results demonstrate high prediction accuracy, with an average $R^2$ of 0.97, RMSE of 0.062, and MAE of 0.045, showing an improvement of 9.64% over MultiAM-BiLSTM and 4.06% over PSO-AM-BiLSTM.

(4) Comparative experiments replacing PSO with GA or DE indicate that PSO exhibits stronger adaptability to complex tool wear rate variations, especially within the 100-200 cutting operation range.

The results of the study are highly informative for RUL prediction in milling machining. The method can also be conveniently applied to RUL prediction of cutting tools in other real machining processes, such as turning and drilling. The validation of this method is currently limited to a single tool parameter; future work can expand experimental samples and apply the approach to different RUL prediction scenarios.

## Acknowledgments

This work is supervised by professors from Universiti Malaysia Sarawak. This work is also supported by the research program of Qilu Institute of Technology (No.: QIT23NN038).

## Conflicts of Interest

The authors declare no conflict of interest.

## References

[1] J. Karandikar, T. Schmitz, and S. Smith, "Physics-Guided Logistic Classification for Tool Life Modeling and Process Parameter Optimization in Machining," Journal of Manufacturing Systems, vol. 59, pp. 522-534, 2021.

[2] X. Liu, S. Liu, X. Li, B. Zhang, C. Yue, and S. Y. Liang "Intelligent Tool Wear Monitoring Based on Parallel Residual and Stacked Bidirectional Long Short-Term Memory Network," Journal of Manufacturing Systems, vol. 60, pp. 608-619, 2021.

[3] N. K. Singh, R. K. Rathore, A. K. Sinha, and H. Narayan, "Multiobjective Optimization of Process Parameter Subjected to End Milling Process of AA7075 Alloy Through TOPSIS Method," AIP Conference Proceedings, vol. 3111, no. 1, article no. 060004, 2024.

[4] D. Han, J. Yu, and D. Tang, "An HDP-HMM Based Approach for Tool Wear Estimation and Tool Life Prediction," Quality Engineering, vol. 33, no. 2, pp. 208-220, 2020.

[5] N. Brili, M. Ficko, and S. Klančnik, "Tool Condition Monitoring of the Cutting Capability of a Turning Tool Based on Thermography," Sensors, vol. 21, no. 19, article no. 6687, 2021.

[6] L. Wu, K. Sha, Y. Tao, B. Ju, and Y. Chen, "A Hybrid Deep Learning Model as the Digital Twin of Ultra-Precision Diamond Cutting for In-Process Prediction of Cutting-Tool Wear," Applied Sciences, vol. 13, no. 11, article no. 6675, 2023.

[7] W. X. Yan, W. Pin, and L. He, "Reliability Prediction of CNC Machine Tool Spindle Based on Optimized Cascade Feedforward Neural Network," IEEE Access, vol. 9, pp. 60682-60688, 2021.

[8] P. Ünal, B. U. Deveci, and A. M. Özbayoğlu, "A Review: Sensors Used in Tool Wear Monitoring and Prediction," International Conference on Mobile Web and Intelligent Information Systems, pp. 193-205, 2022.

[9] C. Shi, B. Luo, S. He, K. Li, H. Liu, and B. Li, "Tool Wear Prediction via Multidimensional Stacked Sparse Autoencoders with Feature Fusion," IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5150-5159, 2020.

[10] Q. An, Z. Tao, X. Xu, M. El Mansori, and M. Chen, "A Data-Driven Model for Milling Tool Remaining Useful Life Prediction with Convolutional and Stacked LSTM Network," Measurement, vol. 154, article no. 107461, 2020.

[11] M. Wang, J. Zhou, J. Gao, Z. Li, and E. Li, "Milling Tool Wear Prediction Method Based on Deep Learning under Variable Working Conditions," IEEE Access, vol. 8, pp. 140726-140735, 2020.

[12] S. Wang, Z. Yu, G. Xu, and F. Zhao, "Research on Tool Remaining Life Prediction Method Based on CNN-LSTM-PSO," IEEE Access, vol. 11, pp. 80448-80464, 2023.

[13] J. Yao, B. Lu, and J. Zhang, "Tool Remaining Useful Life Prediction Using Deep Transfer Reinforcement Learning Based on Long Short-Term Memory Networks," The International Journal of Advanced Manufacturing Technology, vol. 118, no. 3-4, pp. 1077-1086, 2022.

[14] B. Li and X. Tian, "An Effective PSO-LSSVM-Based Approach for Surface Roughness Prediction in High-Speed Precision Milling," IEEE Access, vol. 9, pp. 80006-80014, 2021.

[15] Y. Ge, J. Zhang, G. Song, and K. Zhu, "An Effective LSSVM-Based Approach for Milling Tool Wear Prediction," The International Journal of Advanced Manufacturing Technology, vol. 126, no. 9-10, pp. 4555-4571, 2023.

[16] J. W. Li, C. B. Liu, H. Guo, and N. Lyu, "Tool Life Prediction Based on PSO-RBF Neural Network," Computer Systems Applications, vol. 31, no. 1, pp. 309-314, 2022. (In Chinese)

[17] P. Wei, R. Li, X. Liu, H. Gao, M. Dai, Y. Zhang, et al., "Research on Tool Wear State Identification Method Driven by Multi-Source Information Fusion and Multi-Dimension Attention Mechanism," Robotics and Computer-Integrated Manufacturing, vol. 88, article no. 102741, 2024.

[18] G. Wang and F. Zhang, "A Sequence-to-Sequence Model with Attention and Monotonicity Loss for Tool Wear Monitoring and Prediction," IEEE Transactions on Instrumentation and Measurement, vol. 70, article no. 3525611, 2021.

[19] C. Dong and J. Zhao, "An Augmented AutoEncoder with Multi-Head Attention for Tool Wear Prediction in Smart Manufacturing," IEEE Access, vol. 12, pp. 79128-79137, 2024.

[20] G. Sateesh Babu, P. Zhao, and X. L. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life," International Conference on Database Systems for Advanced Applications, pp. 214-228, 2016.

[21] C. Zhao, X. Huang, Y. Li, and M. Yousaf Iqbal, "A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction," Sensors, vol. 20, no. 24, article no. 7109, 2020.

[22] B. Li, M. Han, X. Zhang, H. Luan, and S. Sun, "Research on the Prediction Method of Remaining Tool Life Based on GASF-LSTM-Attention," 10th International Conference on Computer and Communications, pp. 1422-1427, 2024.